Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Μεταπτυχιακή Διατριβή
στα Πληροφοριακά Συστήματα

Αποτίμηση της Καταλληλότητας Χρήσης Αντικειμενοστρεφών Μετρικών για τη Δημιουργία Μοντέλων Πρόβλεψης Σφαλμάτων σε Λογισμικό Ανοικτού Κώδικα

Χρήστος Φιλιππόπουλος

Επιβλέπων Καθηγητής
Μιχάλης Ξένος

Αύγουστος 2012

Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Αποτίμηση της Καταλληλότητας Χρήσης Αντικειμενοστρεφών Μετρικών για τη Δημιουργία Μοντέλων Πρόβλεψης Σφαλμάτων σε Λογισμικό Ανοικτού Κώδικα

Χρήστος Φιλιππόπουλος

Επιβλέπων Καθηγητής
Μιχάλης Ξένος

Η παρούσα μεταπτυχιακή διατριβή υποβλήθηκε
προς μερική εκπλήρωση των απαιτήσεων για απόκτηση

μεταπτυχιακού τίτλου σπουδών
στα Πληροφοριακά Συστήματα

από τη Σχολή Θετικών και Εφαρμοσμένων Επιστημών
του Ανοικτού Πανεπιστημίου Κύπρου

Αύγουστος 2012

Περίληψη

Η παρούσα μεταπτυχιακή διατριβή εντάσσεται στην ευρύτερη περιοχή της Τεχνολογίας Λογισμικού και πιο συγκεκριμένα καταπιάνεται με το περισπούδαστο θέμα της Ποιότητας Λογισμικού. Το λογισμικό ανοικτού κώδικα τα τελευταία χρόνια έχει γίνει απαραίτητο τόσο στην καθημερινότητα εκατομμυρίων χρηστών όσο και στις εταιρίες που επενδύουν όλο και περισσότερο σε έργα ανοικτού κώδικα. Το μοντέλο ανάπτυξης του λογισμικού ανοικτού κώδικα έχει μεγάλες διαφορές σε σχέση με τις παραδοσιακές μεθόδους ανάπτυξης εμπορικού λογισμικού, αφού στηρίζεται κυρίως στη συμμετοχή εθελοντών. Έτσι, προκύπτει άμεσα η ανάγκη να μελετηθεί η ποιότητα του πηγαίου κώδικά τους και να αναζητηθούν τρόποι για τη βελτίωσή της. Πολύ σημαντικός παράγοντας που μπορεί να επηρεάσει αρνητικά την ποιότητα ενός λογισμικού είναι η ύπαρξη σφαλμάτων. Επιπρόσθετα, ο αντικειμενοστρεφής τρόπος ανάλυσης και σχεδίασης είναι πλέον στις ημέρες μας ο κυρίαρχος τρόπος ανάπτυξης εφαρμογών με τη συντριπτική πλειοψηφία των έργων ανοικτού κώδικα να τον εφαρμόζουν. Προκύπτει λοιπόν το ερώτημα: *υπάρχει τρόπος να προβλέψουμε τα σφάλματα σε λογισμικό ανοικτού κώδικα που ακολουθεί το αντικειμενοστρεφές παράδειγμα;*

Σκοπός της έρευνάς μας είναι η διερεύνηση της απάντησης στο παραπάνω ερώτημα αποτιμώντας εμπειρικά την καταλληλότητα της χρήσης αντικειμενοστρεφών μετρικών για τη δημιουργία μοντέλων πρόβλεψης σφαλμάτων. Με τη βοήθεια του εργαλείου ckjm υπολογίστηκαν δεκαεπτά αντικειμενοστρεφείς μετρικές σε επίπεδο κλάσης για τον πηγαίο κώδικα του λογισμικού ανοικτού κώδικα jEdit. Στη συνέχεια έγινε αντιστοίχιση των σφαλμάτων του με τις κλάσεις του πηγαίου κώδικα που χρειάστηκαν διόρθωση. Τα παραπάνω δεδομένα αποτέλεσαν την είσοδο σε μοντέλα πρόβλεψης σφαλμάτων που κατασκευάστηκαν με τη χρήση τόσο παραδοσιακών στατιστικών τεχνικών στο R (γραμμική και λογιστική παλινδρόμηση), όσο και των νεότερων αλλά πολλά υποσχόμενων τεχνικών μηχανικής μάθησης στο WEKA (δέντρο απόφασης και τεχνητό νευρωνικό δίκτυο). Η αποτίμηση κάθε μοντέλου έγινε με χρήση της διασταυρωμένης επικύρωσης σε 10 μέρη και περιλάμβανε μεταξύ άλλων την αναλογία σωστών θετικών προβλέψεων, την ακρίβεια, την ανάκληση και την περιοχή κάτω από την καμπύλη ROC. Το μοντέλο που προέκυψε από την εφαρμογή της δυαδικής πολλαπλής λογιστικής παλινδρόμησης με επιλογή προς τα εμπρός είχε την καλύτερη επίδοση, ενώ του τεχνητού νευρωνικού δικτύου ακολούθησε δεύτερο με μικρή όμως διαφορά.

Summary

This thesis refers to the scientific field of Software Engineering and more specifically is dealing with the profound issue of Software Quality. Open source software has become necessary in every day life for millions of users but also for companies that keep investing resources in open source projects. The development model for open source software has great differences compared to the traditional development methods of commercial software, as it mainly depends on volunteers' contribution. That's the reason we need to focus extensively on the quality of its source code, as well as it's essential to seek methods for improving its software quality. A major factor that could negatively affect the software quality is the presence of faults. Object oriented analysis and design is nowadays the main method for applications' development, a method that most of the open source projects are currently using. So an important question arises: *Can faults be predicted in object-oriented open source projects?*

The main intention of this research is to examine the above matter, empirically assessing the suitability of using object oriented metrics in order to create fault predicting models. Using the tool ckjm, we calculated seventeen object oriented metrics on class level, for the source code of the open source software jEdit. Afterwards, the faults were corresponded with the source code classes that needed to be corrected. The above data constituted the input in fault prediction models constructed by using traditional statistical techniques in R (linear and logistic regression), and the more modern and promising machine learning algorithms in WEKA (decision tree and artificial neural network). Each model was validated by using 10-fold cross-validation technique and among others we calculated the true positive rate, the precision, the recall and the area under the ROC curve (AUC). The model created by applying binary multiple logistic regression with stepwise forward selection provided the best performance, while artificial neural network followed second with a slight distance.

Ευχαριστίες

Η εκπόνηση μιας διπλωματικής διατριβής πέρα από την απόκτηση πολύτιμων επιστημονικών γνώσεων και δεξιοτήτων, προσφέρει και πολλά άλλα έμμεσα οφέλη. Ένα από αυτά είναι ότι σε παρακινεί να συζητήσεις τα ερευνητικά σου ερωτήματα, τις δυσκολίες που αντιμετωπίσεις αλλά και να ακούσεις γνώμες για τα αποτελέσματα της έρευνας σου από πολλούς ανθρώπους. Αλλά και σε ψυχολογικό επίπεδο, όταν όλα μοιάζουν δύσκολα και σκοντάφτεις από τον ένα τοίχο στον άλλο μέχρι να βρεις εκείνη την χαραμάδα για να προχωρήσεις ένα βηματάκι παραπέρα την έρευνα σου, τότε πάλι αυτό σε βοηθάει να βρεις τα όρια σου αλλά και να εκτιμήσεις τους ανθρώπους που πραγματικά είναι κοντά σου και σε στηρίζουν.

Πρώτα από όλους θα ήθελα να ευχαριστήσω τον επιβλέποντα της μεταπτυχιακής διατριβής κύριο Μιχάλη Ξένο, Αναπληρωτή Καθηγητή του Τμήματος Πληροφορικής στο Ελληνικό Ανοικτό Πανεπιστήμιο, με τον οποίο η συνεργασία μας ξεκίνησε το 2006 στα πλαίσια της θεματικής ενότητας ΠΛΗ42 «Ειδικά Θέματα Τεχνολογίας Λογισμικού» και συνεχίστηκε με την εκπόνηση της πτυχιακής μου εργασίας ένα χρόνο αργότερα. Έμελλε να συναντηθούμε πάλι το 2010 στο Ανοικτό Πανεπιστήμιο της Κύπρου στα πλαίσια της θεματικής ενότητας ΠΛΣ61 «Σχεδίαση και Ανάλυση Λογισμικού» και να συνεχίσουμε την συνεργασία μας με την εκπόνηση της παρούσας διπλωματικής διατριβής. Τον ευχαριστώ ολόθερμα για την εμπιστοσύνη που μου έδειξε κατά την εκπόνηση της, την καθοδήγηση, τις πολύτιμες συμβουλές του αλλά και για την αμέριστη ενθάρρυνση που μου πρόσφερε όταν τα χρονικά περιθώρια στένευαν. Το ευχάριστο κλίμα, η εποικοδομητική συνεργασία μας και το γεγονός ότι η ενασχόληση με θέματα ποιότητας λογισμικού δεν ήταν απλά μια "εργασία" αλλά μία πραγματικά ευχάριστη ενασχόληση, συνέβαλλε τα μέγιστα για την επιτυχή ολοκλήρωση της παρούσας διπλωματικής διατριβής.

Ευχαριστώ την πτυχιούχο Πληροφορικής Κατερίνα Ραμουτσάκη για την βοήθεια στην απόδοση εξειδικευμένων όρων από τα Αγγλικά στα Ελληνικά, τις επισημάνσεις στις πρόχειρες εκδόσεις των κειμένων αλλά και για την γενικότερη ψυχολογική υποστήριξη. Τον φίλο και συμφοιτητή Γιάννη Βενετικίδη που όλα αυτά τα χρόνια στο ΑΠΚΥ μου πρόσφερε αμέριστη συμπαράσταση στα απίστευτα ξενύχτια που κάναμε για την παράδοση όλων (μα πραγματικά όλων!) των εργασιών. Ευτυχώς που υπάρχει και το Skype και δεν άφηνε ο ένας τον άλλο να χαλαρώσει και να κοιμηθεί. Φυσικά δεν θα μπορούσε να αποτελεί εξαίρεση και η διπλωματική διατριβή!

Ευχαριστώ τους αγαπητούς μου συνάδελφους στη Διεύθυνση Διαφάνειας και Ηλεκτρονικής Διακυβέρνησης της Περιφέρειας Δυτικής Ελλάδας που με ενθάρρυναν συνεχώς σε αυτή την προσπάθεια παρά το γεγονός ότι χρειάστηκε να λείψω για μεγάλο χρονικό διαστήματα και επωμίστηκαν το βάρος της απουσίας μου. Ιδιαίτερα αισθάνομαι την ανάγκη να ευχαριστήσω τον Προϊστάμενο της Διεύθυνσης Διαφάνειας και Ηλεκτρονικής Διακυβέρνησης Δημήτρη Ανεστόπουλο. Η συνεργασία μας ξεκίνησε τον Ιανουάριο του 2011 με την εφαρμογή του προγράμματος "Καλλικράτη" και με όλα τα τεράστια προβλήματα που τον συνόδευαν αλλά με την καθοδήγηση του και την στάση του βοήθησε τα μέγιστα για την αυτονόμηση της λειτουργίας του Τμήματος Πληροφορικής της Π.Ε. Αιτωλοακαρνανίας και τη συστηματική μου ενασχόληση με θέματα που άπτονται της επιστήμης της Πληροφορικής. Τον ευχαριστώ για την έγκριση όλων των αδειών που ζήτησα το 2012 ώστε να αφιερωθώ στο παρόν πόνημα, τις εύστοχες επισημάνσεις στις αρχικές εκδόσεις των κείμενων και τις ουσιαστικές συζητήσεις που είχαμε κατά την πορεία της παρούσας έρευνας. Πάντα μια συζήτηση με τον Δημήτρη έχει να σου προσφέρει και μια νέα οπτική γωνία, πολλές φορές μάλιστα να σε κάνει να αναρωτιέσαι γιατί δεν το σκέφτηκες πιο πριν, αφού η λύση ήταν μπροστά σου!

Φιλιππόπουλος Χρήστος

Μεσολόγγι, Αύγουστος 2012

Περιεχόμενα

Κατάλογος Εικόνων

Κατάλογος Σχημάτων

Κατάλογος Πινάκων

Κεφάλαιο 1
Εισαγωγή

Το κεφάλαιο ξεκινάει με το κίνητρο που μας οδήγησε να ερευνήσουμε το συγκεκριμένο θέμα. Ο μεγαλύτερος αριθμός ερευνών σχετικά με την ποιότητα του λογισμικού, όπως και τα περισσότερα μοντέλα ποιότητας λογισμικού στηρίζονται στο μοντέλο ανάπτυξης του κλειστού κώδικα και αγνοούν τις ιδιαιτερότητες της ανάπτυξης λογισμικού ανοικτού κώδικα. Αν και τα έργα ανοικτού κώδικα έχουν να επιδείξουν ένα μεγάλο αριθμό από επιτυχημένα προγράμματα που χρησιμοποιούνται από εκατομμύρια χρήστες καθημερινά, η ακαδημαϊκή κοινότητα δεν φαίνεται να έχει ασχοληθεί επισταμένα με τα σημαντικά θέματα της αξιολόγησης του πηγαίου κώδικα τους. Αυτή η έλλειψη ήταν το βασικό κίνητρο για την έναρξη της παρούσας έρευνας. Μετά παρουσιάζουμε τα ερευνητικά ερωτήματα που θα προσπαθήσουμε να απαντήσουμε στην διπλωματική διατριβή. Το κύριο ερώτημα είναι αν μπορούμε να προβλέψουμε τα σφάλματα σε έργα ανοικτού κώδικα με χρήση αντικειμενοστρεφών μετρικών. Στη συνέχεια γίνεται μια αναφορά στις σημαντικότερες μελέτες σχετικά με την πρόβλεψη σφαλμάτων στο λογισμικό. Η εξέταση τους γίνεται με χρονολογική σειρά και για κάθε έρευνα αναφέρουμε τα δεδομένα που χρησιμοποίησαν, τις μεθόδους που εφάρμοσαν για την κατασκευή των μοντέλων πρόβλεψης και τα σημαντικότερα αποτελέσματα που ανέφεραν. Τέλος, δίνεται μια αναλυτική περιγραφή για κάθε ένα από τα υπόλοιπα κεφάλαια της μεταπτυχιακής διατριβής ώστε ο αναγνώστης να αποκτήσει μια πρώτη εικόνα για την οργάνωση της παρούσας έρευνας.

1.1 Κίνητρο

Το λογισμικό ανοικτού κώδικα άρχισε να έρχεται στο προσκήνιο προς τα τέλη της δεκαετίας του '90 και σιγά σιγά κατάφερε να αλλάξει τον τρόπο με τον οποίο αντιλαμβάνονται την ανάπτυξη του λογισμικού τόσο οι εταιρίες παραγωγής όσο και οι προγραμματιστές [026]. Η επιτυχία αυτή σε συνδυασμό με τον μεγάλο βαθμό εισχώρησης του λογισμικού ανοικτού κώδικα σε πάρα πολλές εκφάνσεις της ζωής μας, έχει κεντρίσει το ενδιαφέρον της ακαδημαϊκής κοινότητας σε πάρα πολλούς τομείς που ξεφεύγουν από τα στενά όρια της πληροφορικής και άπτονται τομέων όπως οι οικονομικές επιστήμες, η κοινωνιολογία αλλά και η πολιτική επιστήμη.

Πλέον η αξία του αναγνωρίζεται ακόμη και από μεγάλες παραδοσιακές εταιρίες όπως είναι η IBM και η HP, που έχουν υιοθετήσει αρκετές πρακτικές του. Όμως η συντριπτική πλειοψηφία των εταιριών που αναπτύσσουν λογισμικό λειτουργούν με το μοντέλο του κλειστού κώδικα δηλαδή διανέμουν μόνο τα εκτελέσιμα αρχεία, ενώ στον πηγαίο κώδικα του προγράμματος δεν έχει κανένας πρόσβαση εκτός αυτών που εργάζονται στην ανάπτυξη του [027]. Αντίθετα στα έργα ανοικτού λογισμικού κεντρικό ρόλο έχουν η εθελοντική συνεργατική ανάπτυξη του λογισμικού χωρίς τους περιορισμούς ενός εταιρικού περιβάλλοντος και η ελεύθερη διάθεση του πηγαίου κώδικα σε όλους τους ενδιαφερόμενους [028]. Παρά το γεγονός ότι αυτές οι ιδέες υπάρχουν από τότε που γράφτηκαν τα πρώτα προγράμματα για υπολογιστές, η μορφή του λογισμικού ανοικτού κώδικα που γνωρίζουμε σήμερα ξεκίνησε το 1984 από τον Richard Stallman με το GNU Project και τον οργανισμό Free Software Foundation που το υποστήριζε [029].

Οι περισσότερες μελέτες σχετικά με την ποιότητα του λογισμικού, όπως και τα περισσότερα μοντέλα ποιότητας λογισμικού στηρίζονται στο μοντέλο ανάπτυξης του κλειστού κώδικα και αγνοούν τις ιδιαιτερότητες της ανάπτυξης λογισμικού ανοικτού κώδικα. Αν και τα έργα ανοικτού κώδικα έχουν να επιδείξουν ένα τεράστιο αριθμό από επιτυχημένα προγράμματα που χρησιμοποιούνται από εκατομμύρια χρήστες καθημερινά, η ακαδημαϊκή κοινότητα δεν φαίνεται να έχει ασχοληθεί επισταμένως με τα σημαντικά θέματα, της αξιολόγησης του πηγαίου κώδικά τους και της εξεύρεσης τρόπων για την βελτίωση της ποιότητάς τους εν γένει. Η έλλειψη αυτή αποτέλεσε και το κίνητρο για την έναρξη της παρούσας έρευνας. Ένας παράγοντας που μπορεί να επηρεάσει αρνητικά την ποιότητα ενός λογισμικού είναι η ύπαρξη σφαλμάτων. Η έγκαιρη πρόγνωση της πιθανής ύπαρξης σφάλματος στον πηγαίο κώδικα που υποβάλλει κάποιος από την ομάδα ανάπτυξης μπορεί να συμβάλλει στη βελτίωση της ποιότητας του λογισμικού που διατίθενται από τα έργα ανοικτού κώδικα, αφού οι προσπάθειες εντοπισμού σφαλμάτων πριν τη διάθεση του πηγαίου κώδικα μπορούν να επικεντρωθούν στο συγκεκριμένο σημείο.

1.2 Ερευνητικά Ερωτήματα

Μία από τις σημαντικότερες εργασίες που πραγματοποιείται κατά τη διάρκεια της ανάπτυξης ενός λογισμικού και πιο εντατικά πριν την επίσημη δημόσια διανομή του, είναι ο έλεγχος για σφάλματα. Επειδή όμως οι πόροι που μπορούν να διατεθούν για τον έλεγχο είναι πεπερασμένοι, για την επίτευξη της καλύτερης δυνατής ποιότητας θα πρέπει να ελεγχθούν τα κομμάτια του πηγαίου κώδικα που έχουν τις περισσότερες πιθανότητες να έχουν σφάλματα. Έρευνες [030, 031] έχουν δείξει ότι η κατανομή των σφαλμάτων ακολουθεί λίγο πολύ το νόμο του Παρέτο, δηλαδή ότι περίπου το 80% των σφαλμάτων υπάρχουν στο 20% του πηγαίου κώδικα. Αυτή η παρατήρηση οδηγεί και στην πρώτη ερευνητική μας ερώτηση:

Γιατί κάποια κομμάτια πηγαίου κώδικα έχουν μεγαλύτερη ροπή προς τα σφάλματα;

Η απάντηση σε αυτή την ερώτηση μπορεί να μας βοηθήσει να κατανοήσουμε τη φύση των σφαλμάτων και να τα αποφύγουμε μελλοντικά. Δυστυχώς, δεν υπάρχει μια καθολική απάντηση καθώς κάθε έργο λογισμικού φαίνεται να έχει τους δικούς του ιδιαίτερους παράγοντες που καθορίζουν αν ένα κομμάτι κώδικα έχει ροπή προς τα σφάλματα ενώ κάποιο άλλο αντίστοιχα δεν έχει [032]. Τουλάχιστον, μπορούμε να προσπαθήσουμε ώστε να ανιχνεύσουμε εκείνα τα χαρακτηριστικά ενός συγκεκριμένου έργου λογισμικού που αυξάνουν την τάση προς τα σφάλματα και να εστιάσουμε σε αυτά κατά τον έλεγχο του πηγαίου κώδικα. Αυτό οδηγεί στη δεύτερη ερευνητική μας ερώτηση:

Μπορούμε να προβλέψουμε την πιθανότητα σφάλματος σε ένα κομμάτι κώδικα;

Αρχικά κάποιος θα μπορούσε να υποθέσει ότι για να απαντήσουμε στο παραπάνω ερώτημα αρκεί να εξετάσουμε το ιστορικό των σφαλμάτων και να βρούμε ποιά κομμάτια κώδικα είχαν στον παρελθόν περισσότερα σφάλματα. Όμως το λογισμικό εξελίσσεται συνεχώς με αναδόμηση (refactoring) του πηγαίου κώδικα και με προσθήκες νέων λειτουργιών. Οπότε το ιστορικό μπορεί να μας φανερώσει ποιά κομμάτια πηγαίου κώδικα που είχαν σφάλματα στο παρελθόν διορθώθηκαν αλλά όχι ποια κομμάτια πηγαίου κώδικα θα παρουσιάσουν σφάλματα στο μέλλον. Για αυτό το λόγο, χρειαζόμαστε τη δημιουργία μοντέλων πρόβλεψης σφαλμάτων που θα μπορούν να εφαρμοστούν τόσο σε νέα έργα όσο και σε αυτά που εξελίσσονται. Οι εσωτερικές μετρικές λογισμικού έχουν χρησιμοποιηθεί στο παρελθόν με επιτυχία για την πρόβλεψη σφαλμάτων σε εμπορικό λογισμικό [005, 017, 022] ή σε λογισμικό που έχει αναπτυχθεί από φοιτητές [014, 015, 016]. Παρά τη μεγάλη εξάπλωση των έργων ανοικτού κώδικα, αναλογικά

λίγες έρευνες έχουν προσπαθήσει να βελτιώσουν την ποιότητα τους μέσω της πρόβλεψης σφαλμάτων στον πηγαίο κώδικα τους [018, 019, 020]. Όμως, επειδή το μοντέλο ανάπτυξης του λογισμικού ανοικτού κώδικα έχει μεγάλες διαφορές σε σχέση με τις παραδοσιακές μεθόδους ανάπτυξης εμπορικού λογισμικού [028], χρειάζεται επιβεβαίωση της ικανότητας πρόβλεψης των μετρικών και για την περίπτωση των έργων ανοικτού κώδικα. Οδηγούμαστε λοιπόν στο καίριο ερευνητικό ερώτημα που θα προσπαθήσει να απαντήσει η παρούσα έρευνα :

Μπορούμε να προβλέψουμε τα σφάλματα σε έργα ανοικτού κώδικα με χρήση μετρικών;

Στα χρονικά πλαίσια εκπόνησης μιας μεταπτυχιακής διατριβής θα ήταν αδύνατο να ελέγξουμε την παραπάνω ερώτηση στη γενική της περίπτωση, συνεπώς έγιναν κάποιες επιλογές και τέθηκαν κάποιοι περιορισμοί. Όμως, δόθηκε ιδιαίτερη έμφαση ώστε να μην γίνουν επιλογές που θα ελάττωναν σημαντικά την αντιπροσωπευτικότητα των αποτελεσμάτων:

- Ο αντικειμενοστρεφής τρόπος ανάλυσης και σχεδίασης είναι πλέον στις ημέρες μας ο κυρίαρχος τρόπος ανάπτυξης εφαρμογών, με τη συντριπτική πλειοψηφία των έργων ανοικτού κώδικα να τον εφαρμόζουν. Οπότε επιλέξαμε να εκτιμήσουμε την καταλληλότητα της χρήσης αντικειμενοστρεφών μετρικών για τη δημιουργία μοντέλων πρόβλεψης σφαλμάτων στις κλάσεις του πηγαίου κώδικα.

- Δεν υπάρχει εύκολος και αυτοματοποιημένος τρόπος που να αντιστοιχεί τις διορθώσεις των σφαλμάτων ενός προγράμματος ανοικτού κώδικα με τις αντίστοιχες κλάσεις που μεταβλήθηκαν. Η χειροκίνητη αντιστοίχηση είναι χρονοβόρα, οπότε επιλέχθηκε ένα λογισμικό ανοικτού κώδικα μεσαίου μεγέθους λίγων εκατοντάδων κλάσεων.

- Υπάρχει μία πληθώρα από γλώσσες προγραμματισμού που ακολουθούν το αντικειμενοστρεφές παράδειγμα, όπου οι C++ και Java είναι οι πιο δημοφιλείς επιλογές. Δεδομένου ότι η C++ δεν είναι "καθαρή" αντικειμενοστρεφής γλώσσα προγραμματισμού και έχει χρησιμοποιηθεί στις περισσότερες σχετικές μελέτες, αποφασίσαμε ως γλώσσα προγραμματισμού να εξετάσουμε την περίπτωση της Java.

- Υπάρχουν πολλές μετρικές που έχουν προταθεί στην βιβλιογραφία [033, 034, 035, 036, 037, 038, 039, 040, 041] για τη μέτρηση της ποιότητας του πηγαίου κώδικα ενός λογισμικού. Δύσκολο να εξεταστούν όλες. Η επιλογή των αντικειμενοστρεφών μετρικών έγινε με γνώμονα τα αποτελέσματα σχετικών ερευνών για την πρόβλεψη σφαλμάτων και τη διαθεσιμότητα εργαλείων ανοικτού κώδικα που τις υποστηρίζουν.

1.3 Επισκόπηση Βιβλιογραφίας

Οι περισσότερες προσεγγίσεις στην πρόβλεψη σφαλμάτων του λογισμικού συνήθως χρησιμοποιούν μετρικές και το ιστορικό των σφαλμάτων των προηγουμένων ή της προηγούμενης έκδοσης του. Αν ένα σφάλμα αναφερθεί κατά τη διάρκεια των δοκιμών ή και μετά την δημόσια διανομή του λογισμικού τότε το ή τα κομμάτια του πηγαίου κώδικα που χρειάζονται τροποποίηση σημειώνονται με το ένα αλλιώς με το μηδέν. Έτσι, για να συλλέξουμε τα απαραίτητα δεδομένα χρειάζονται ένα σύστημα διαχείρισης πηγαίου κώδικα (version control systems) όπως είναι το CVS ή το SVN, ένα σύστημα καταχώρησης και αναφοράς σφαλμάτων (bug tracking system) όπως είναι το Bugzilla [060] και ένα εργαλείο για τη συλλογή μετρικών από τον πηγαίο κώδικα του προγράμματος. Για την μοντελοποίηση χρησιμοποιούμε τις μετρικές λογισμικού ως ανεξάρτητες μεταβλητές και το ιστορικό σφαλμάτων ως εξαρτημένη μεταβλητή. Οι τεχνικές που χρησιμοποιούνται για την κατασκευή των μοντέλων κατά πλειοψηφία είναι κλασσικές στατιστικές μέθοδοι αλλά τα τελευταία χρόνια έχουν ξεκινήσει να χρησιμοποιούνται τεχνικές μηχανικής μάθησης. Στη συνέχεια παρουσιάζουμε με χρονολογική σειρά τις πιο σημαντικές μελέτες σχετικά με την πρόβλεψη σφαλμάτων στο λογισμικό .

Οι Lanubil et al. [042] συνέκριναν μοντέλα που προέκυψαν από ανάλυση κυρίων συνιστωσών (principal component analysis), διακρίνουσας ανάλυσης (discriminant analysis), λογιστική παλινδρόμηση (logistic regression) και πολυεπίπεδα τεχνητά νευρωνικά δίκτυα (multilayer neural networks) για την πρόβλεψη σφαλμάτων σε είκοσι εφτά ακαδημαϊκά έργα λογισμικού που υλοποιήθηκαν στο Πανεπιστήμιο του Bari. Χρησιμοποίησαν έντεκα μετρικές που περιλάμβαναν μεταξύ άλλων αυτές του Halstead [061, 062], McCabe [063], μετρικές ροής πληροφορίας (information flow metrics) των Henry και Kafura [064]. Καμιά από τις μεθόδους που χρησιμοποίησαν δεν έδωσε αποδεκτά αποτελέσματα αφού δεν μπορούσαν να ξεχωρίσουν μεταξύ των κομματιών κώδικα που είχαν σφάλματα με αυτά που δεν είχαν. Οι Khoshgoftaar et al [043] εφάρμοσαν τεχνητά νευρωτικά δίκτυα και διακρίνουσα ανάλυση σε ένα μεγάλο τηλεπικοινωνιακό σύστημα γραμμένο στη γλώσσα προγραμματισμού PROTEL που αποτελούνταν από περίπου δεκατρία εκατομμύρια γραμμές πηγαίου κώδικα. Λόγο του μεγάλου μεγέθους ως ελάχιστο κομμάτι κώδικα που εξετάστηκε ήταν το σύνολο των αρχείων πηγαίου κώδικα που υλοποιούσε μια συγκεκριμένη λειτουργία. Τα αποτελέσματα του νευρωνικού δικτύου ήταν αρκετά κανοποιητικά και το μοντέλο ενσωματώθηκε στο λογισμικό της εταιρίας EMELALD Software [065]. Οι Evett et al [044] δημιούργησαν ένα μοντέλο ποιότητας που βασιζόταν στο γενετικό προγραμματισμό (genetic programming) σε ένα στρατιωτικό σύστημα

μεταβίβασης πληροφοριών. Χρησιμοποίησαν οκτώ επίπεδα μετρικών που αποτελούνταν από τις μετρικές Halstead, McCabe και γραμμές πηγαίου κώδικα LOC. Οι επιδόσεις των αποτελεσμάτων έγιναν με την μέθοδο της διατάξιμης αποτίμησης (ordinal evaluation) και τα αποτελέσματα ήταν αρκετά ενθαρρυντικά.

Ο Kaszycki [045] μαζί με τις εσωτερικές μετρικές προϊόντος χρησιμοποίησε και μετρικές διαδικασίας όπως είναι π.χ. η εμπειρία του προγραμματιστή, προκειμένου να εμπλουτίσει με περισσότερα στοιχεία τα μοντέλα πρόβλεψης σφαλμάτων. Ως μέθοδος αποτίμησης των αποτελεσμάτων επιλέχτηκε το ποσοστό σωστών θετικών προβλέψεων (TP rate) και σωστών αρνητικών προβλέψεων (TN rate). Όταν στα μοντέλα περιλαμβάνονταν οι μετρικές της διαδικασίας τότε η απόδοση τους βελτιωνόταν κατά ένα σημαντικό βαθμό. Παρ' όλα αυτά, η χρήση μετρικών διαδικασίας συνεπάγεται ότι θα πρέπει τα μοντέλα να αναπροσαρμόζονται όταν πραγματοποιούνται οργανωτικές ή άλλες σημαντικές αλλαγές στην εταιρία που αναπτύσσει το λογισμικό. Ο Denaro [046] με την χρήση της λογιστικής παλινδρόμησης σε λογισμικό διαμόρφωσης σήματος σε κεραίες δημιούργησε μοντέλα πρόβλεψης σφαλμάτων των συστατικών (components) του. Ο συντελεστής καλής προσαρμογής R^2 (goodness of fit coefficient) χρησιμοποιήθηκε για τη σύγκριση των μοντέλων που προέκυψαν και βρέθηκε να υπάρχει συσχέτιση μεταξύ των εσωτερικών μετρικών και της πρόβλεψης σφαλμάτων.

Επίσης, οι Emam et al [048] έκαναν χρήση της λογιστικής παλινδρόμησης για την κατασκευή μοντέλων για την πρόβλεψη σφαλμάτων στις κλάσεις ενός εμπορικού προγράμματος γραμμένου στην γλώσσα προγραμματισμού Java. Επιλέχτηκαν να υπολογιστούν δύο μετρικές από τη συλλογή μετρικών CK [039], οκτώ από τη συλλογή του Briand [066] και ο αριθμός γραμμών του πηγαίου κώδικα της κλάσης. Ανέφεραν ότι το βάθος του δέντρου της κληρονομικότητας (depth of inheritance tree) και η φυγόκεντρη σύζευξη (efferent coupling) είναι οι πιο χρήσιμες μετρικές για την πρόβλεψη σφαλμάτων στις κλάσεις του προγράμματος. Οι Khoshgoftaar et al [049] υπολόγισαν είκοσι οκτώ μετρικές και εφάρμοσαν τις τεχνικές SPRINT [067] και CART [068] σε ένα μεγάλο τηλεπικοινωνιακό σύστημα για την ποιοτική κατηγοριοποίηση του πηγαίου κώδικα του. Υπολόγισαν τα σφάλματα πρώτου (Type I error), δευτέρου τύπου (Type II error) και τη συνολική σωστή κατηγοριοποίηση (correctly classified instances). Συνολικά το δέντρο απόφασης του SPRINT ήταν καλύτερο αφού είχε μικρότερο σφάλμα πρώτου τύπου και ήταν πιο εύρωστο (robust) αλλά ήταν αρκετά πιο πολύπλοκο από αυτό της τεχνικής CART. Οι Denaro et al [050] εφάρμοσαν λογιστική παλινδρόμηση σε ένα εμπορικό λογισμικό του βιομηχανικού τομέα με είσοδο τέσσερις από τις έξι μετρικές της συλλογής CK. Κατέληξαν στο συμπέρασμα ότι καμιά από τις μετρικές που εξέτασαν δεν

συσχετίζονταν με τα σφάλματα του λογισμικού και επιπλέον το μοντέλο που προέκυψε από την πολλαπλή λογιστική παλινδρόμηση δεν πρόσφερε κανένα πλεονέκτημα σε σχέση με την χρησιμοποίηση της μετρικής των γραμμών του πηγαίου κώδικα LOC. Το γεγονός της μη συσχέτισης καμίας αντικειμενοστρεφής μετρικής με τα σφάλματα μπορεί να οφείλεται στην ιδιαιτερότητα του συγκεκριμένου λογισμικού που προερχόταν από ένα σύστημα γραμμένο στην διαδικαστική γλώσσα προγραμματισμού C που αναβαθμίστηκε σε αντικειμενοστρεφή. Σε μια άλλη έρευνα τους οι Denaro et al. [051] χρησιμοποίησαν λογιστική παλινδρόμηση με μετρικές σε επίπεδο συνάρτησης στο λογισμικό εξυπηρετητή παγκόσμιου ιστού (web server) Apache 1.3 και Apache 2.0 για την πρόβλεψη σφαλμάτων στα συστατικά του. Η αξιολόγηση των μοντέλων έγινε με τον συντελεστή καλής προσαρμογής R^2, την πληρότητα (completeness) των συστατικών με σφάλματα και την ορθότητα (correctness) των συστατικών που προβλέφθηκαν ως προβληματικά. Επίσης, έδειξαν ότι η λογιστική παλινδρόμηση με την χρήση της διασταυρωμένης επικύρωσης (cross validation) είναι μια αποτελεσματική προσέγγιση για την πρόβλεψη σφαλμάτων στο λογισμικό όταν δεν υπάρχουν πολλά δεδομένα.

Οι Mahaweerawat et al. [052] αρχικά χρησιμοποίησαν ένα πολυεπίπεδο τεχνητό νευρωνικό δίκτυο αισθητήρα (perceptron) που εκπαιδεύεται με την μέθοδο της πίσω διάδοσης του λάθους (error back propagation) για να προσδιορίσουν ποιές κλάσεις έχουν σφάλματα και στη συνέχεια εφάρμοσαν ακτινική βάση συνάρτησης (radial basis function) για την κατηγοριοποίηση τους σε τύπους σφαλμάτων. Για την αποτίμηση των αποτελεσμάτων χρησιμοποίησαν την ακρίβεια (accuracy), σφάλματα πρώτου και δευτέρου τύπου. Με την παραπάνω μεθοδολογία κατάφεραν να εντοπίσουν επιτυχώς γύρω στο 90% των κλάσεων που είχαν σφάλματα. Οι Koru και Liu [053] διερεύνησαν την επίδραση του μεγέθους ενός συστατικού στην πρόβλεψη σφαλμάτων με την χρησιμοποίηση του J48 [008] και Kstar αλγορίθμων [069]. Ο αρμονικός διαιρέτης (F-measure) χρησιμοποιήθηκε για την αποτίμηση της επίδοσης των μοντέλων και υπολογίστηκαν μετρικές σε επίπεδο συνάρτησης και κλάσεις στις δημόσιες συλλογές δεδομένων της NASA. Την καλύτερη επίδοση είχαν ο J48 και οι μπεϋσιανοί ταξινομητές (bayesian classifiers) ενώ το τεχνητό νευρωνικό δίκτυο και οι μηχανές διανυσμάτων υποστήριξης (support vector machines) δεν τα πήγαν ιδιαίτερα καλά. Κατέληξαν, ότι ο αλγόριθμοι μηχανικής μάθησης του WEKA [070] δεν είναι ιδιαίτερα ενθαρρυντικοί αλλά ούτε και τελείως απογοητευτικοί. Οι Gyimothy et al [019] μελέτησαν αντικειμενοστρεφείς μετρικές για την πρόβλεψη σφαλμάτων με τη χρησιμοποίηση γραμμικής παλινδρόμησης, λογιστικής παλινδρόμησης, δέντρων απόφασης και τεχνητών νευρωνικών δικτύων στο λογισμικό ανοικτού κώδικα Mozilla [071]. Υπολογίστηκαν μετρικές σε επίπεδο κλάσης και για την απόδοση των μοντέλων που δημιουργήθηκαν χρησιμοποίησαν την

πληρότητα, την ορθότητα και την ακρίβεια. Κατέληξαν ότι η σύζευξη μεταξύ των κλάσεων είναι πολύ χρήσιμη για την πρόβλεψη σφαλμάτων, ενώ αντίθετα το βάθος του δέντρου κληρονομικότητας; και ο αριθμός των άμεσων απογόνων μιας κλάσης δεν πρέπει να χρησιμοποιούνται. Οι Zhou και Leung [023] προσπάθησαν να προβλέψουν δύο κατηγορίες σφαλμάτων στη συλλογή δεδομένων KC1 της NASA. Χρησιμοποίησαν λογιστική παλινδρόμηση, μπεϋσιανούς ταξινομητές, τυχαία δάση (random forests) και του κοντινότερου γείτονα (nearest neighbor). Ανάφεραν ότι τα σφάλματα χαμηλής σπουδαιότητας μπορούν να προβλεφθούν πιο εύκολα σε σχέση με αυτά που είναι υψηλής σπουδαιότητας. Επίσης, οι άμεσοι απόγονοι μιας κλάσης δεν βοηθούν στην πρόβλεψη σφάλματος, ενώ οι υπόλοιπες μετρικές της συλλογής CK μπορούν να είναι χρήσιμες στην πρόβλεψη σφαλμάτων στις κλάσεις του προγράμματος.

Ο Boetticher [054] χρησιμοποίησε τις δημόσιες συλλογές της NASA και εφάρμοσε τον αλγόριθμο J48 και απλοϊκά μοντέλα Bayes. Τα δεδομένα χωρίστηκαν σε τρία μέρη τα δεδομένα εκπαίδευσης (training set), τα δεδομένα ελέγχου με καλούς γείτονες (nice neighbors test set) και τα δεδομένα ελέγχου με κακούς γείτονες (nasty neighbors test set). Στους καλούς γείτονες το ποσοστό επιτυχίας ήταν 94% και στους κακούς 20%. Ο Pai [024] χρησιμοποίησε γραμμική παλινδρόμηση, παλινδρόμηση poisson και λογιστική παλινδρόμηση για να υπολογίσει την κατανομή της δεσμευμένης πιθανότητας (conditional probability) των κόμβων μπεϋσιανού δικτύου και μετά χρησιμοποίησε αυτά τα δίκτυα για να υπολογίσει την πιθανότητα σφάλματος στις κλάσεις από την δημόσια συλλογή της NASA. Οι μετρικές της συλλογής CK και ο αριθμός γραμμών πηγαίου κώδικα χρησιμοποιήθηκαν ως είσοδοι. Υποστήριξε ότι οι WMC, CBO, RFC και οι LOC είναι πολύ χρήσιμες μετρικές για την πρόβλεψη σφαλμάτων και πρότεινε ένα μπεϋσιανό μοντέλο που συνδύαζε τις μετρικές προϊόντος με μετρικές διαδικασίας.

Οι Tomaszewski et al [055] μελέτησαν λογισμικό που αναπτύχθηκε στην Ericsson. Χρησιμοποίησαν τη γνώμη ειδικών και πολλαπλή γραμμική παλινδρόμηση. Έντεκα ειδικοί πρόβλεψαν τα συστατικά του λογισμικού που μπορεί να είχαν σφάλματα. Μετά κατασκευάστηκαν στατιστικά μοντέλα και έγινε σύγκριση των αποτελεσμάτων τους με αυτά των ειδικών. Τα στατιστικά μοντέλα είχαν καλύτερες επιδόσεις από τις προβλέψεις των ειδικών. Οι Bibi et al [056] εφάρμοσαν την παλινδρόμηση μέσω ταξινόμησης (regression via classification) για να εκτιμήσουν τον αριθμό των σφαλμάτων με ένα διάστημα εμπιστοσύνης (confidence interval). Δεδομένα από μια μεγάλη εμπορική τράπεζα χρησιμοποιήθηκαν και η αποτίμησης της απόδοσης των μοντέλων έγινε με βάση το μέσο απόλυτο σφάλμα (mean absolute error). Η παλινδρόμηση μέσω ταξινόμησης είχε καλύτερα αποτελέσματα από τις κλασσικές μεθόδους παλινδρόμησης. Οι Riquelme et al [057] χρησιμοποίησαν πέντε δημόσιες συλλογές δεδομένων

από το αποθετήριο PROMISE [072] και δύο τεχνικές ταξινόμησης, δέντρο απόφασης C4.5 και απλοϊκά μοντέλα Bayes. Ανέφεραν ότι οι τεχνικές εξισορρόπησης (balancing techniques) βελτιώνουν την περιοχή κάτω από την καμπύλη ROC (AUC) αλλά όχι και το συνολικό ποσοστό σωστής ταξινόμησης των δεδομένων. Οι Chang et al [058] πρότειναν μια προσέγγιση κατά την οποία η πρόβλεψη σφαλμάτων βασιζόταν σε κανόνες συσχέτισης (association rules) για την ανακάλυψη των προτύπων σφαλμάτων. Τα αποτελέσματα στα οποία κατέληξαν ήταν άριστα. Το πλεονέκτημα αυτής της μεθόδου είναι ότι τα πρότυπα σφάλματα μπορούν να χρησιμοποιηθούν και σε μια τυχαία ανάλυση για να βρεθούν οι αιτίες των σφαλμάτων. Οι Marcus et al [074] πρότειναν ένα νέο μέτρο συνεκτικότητας (cohesion) που το ονόμασαν εννοιολογική συνεκτικότητα των κλάσεων C3 (conceptual cohesion of classes). Έδειξαν ότι ο συνδυασμός των μετρικών της δομικής και της εννοιολογικής συνεκτικότητας μπορεί να οδηγήσει σε καλύτερα αποτελέσματα από ότι η χρήση μόνο των δομικών μετρικών για την δημιουργία μοντέλων πρόβλεψης σφαλμάτων. Για την εμπειρική επαλήθευση των παραπάνω χρησιμοποιήθηκαν δύο λογισμικά ανοικτού κώδικα το WinMerge [075] και το Mozilla [071]. Η εκτίμηση της απόδοσης των μοντέλων βασίστηκε στην ακρίβεια, την πληρότητα και την ορθότητα. Τα αποτελέσματα από τη χρησιμοποίηση της λογιστικής παλινδρόμησης έδειξαν ότι η μετρική C3 προβλέπει καλύτερα από πολλές άλλες μετρικές συνεκτικότητας.

Οι Turhan et al [059] μελέτησαν είκοσι πέντε έργα ενός τηλεπικοινωνιακού συστήματος και δημιούργησαν μοντέλα από τα δημόσια δεδομένα της NASA. Χρησιμοποίησαν τον αλγόριθμο του κοντινότερου γείτονα για να κατασκευάσουν μεταβλητές πρόβλεψης από τις είκοσι εννέα μετρικές που υπολόγισαν. Ανέφεραν ότι τουλάχιστον το 70% των σφαλμάτων μπορεί να βρεθεί με την επιθεώρηση μόνο του 6% του πηγαίου κώδικα με απλά μοντέλα Bayes. Οι Olague et al [020] διερεύνησαν τις επιδόσεις τριών συλλογών μετρικών σε έξι εκδόσεις του προγράμματος ανοικτού κώδικα Rhino [073]. Για την κατασκευή των μοντέλων χρησιμοποιήθηκε η στατιστική τεχνικής της λογιστικής παλινδρόμησης. Οι συλλογές μετρικών που συνέκριναν ήταν οι CK,. MOOD και QMOOD. Ο συντελεστής συσχέτισης Spearman χρησιμοποιήθηκε για να βρεθούν οι συσχετίσεις μεταξύ των μετρικών και η εκτίμηση των μοντέλων έγινε με βάση την ακρίβειά τους. Κατέληξαν ότι οι μετρικές CK και QMOOD είναι πολύ χρήσιμες για την πρόβλεψη σφαλμάτων, ενώ αντίθετα οι MOOD δεν βοηθούν καθόλου. Επιπλέον, ανάφεραν ότι τα μοντέλα που κατασκευάζονται με τη λογιστική παλινδρόμηση είναι χρήσιμα για πρόβλεψη σφαλμάτων σε λογισμικό που αναπτύσσεται με εύκαμπτες (agile) μεθοδολογίες και ότι οι μετρικές WMC, RFC, CIS και NOM είναι οι πιο χρήσιμες από όλες τις άλλες για την πρόβλεψη των σφαλμάτων.

1.4 Περίγραμμα Μεταπτυχιακής Διατριβής

Ως το σημείο αυτό έχουμε αναλύσει στον αναγνώστη το κίνητρο που μας οδήγησε να ασχοληθούμε με το συγκεκριμένο θέμα, τα ερευνητικά ερωτήματα που θα προσπαθήσουμε να απαντήσουμε στα πλαίσια της μεταπτυχιακής διατριβής και κάναμε μία σύντομη επισκόπηση όλων των σημαντικότερων σχετικών ερευνών από τα μέσα της δεκαετίας του '90 μέχρι και σήμερα. Στη συνέχεια σκιαγραφούμε τα περιεχόμενα των επόμενων κεφαλαίων:

Θεωρητικό Υπόβαθρο (Κεφάλαια 2 και 3): Στο δεύτερο κεφάλαιο ασχολούμαστε με το λογισμικό ανοικτού κώδικα. Κάνουμε μια σύντομη ιστορική αναδρομή και ξεκαθαρίζουμε τις διαφορές "Ελεύθερου Λογισμικού" με το "Λογισμικό Ανοικτού Κώδικα", δίνοντας τα κριτήρια που πρέπει να πληροί ένα πρόγραμμα για να ανήκει στη μία ή και στην άλλη κατηγορία. Προχωράμε αναδεικνύοντας τις ιδιαιτερότητες της ανάπτυξης των έργων ανοικτού κώδικα σε σχέση με τον παραδοσιακό κύκλο ζωής ενός λογισμικού. Το κεφάλαιο ολοκληρώνεται με μια σύντομη αναφορά στα πλεονεκτήματα και στα μειονεκτήματα του λογισμικού ανοικτού κώδικα αλλά και πεδίων ενδιαφέροντος πέρα από τα στενά πλαίσια της πληροφορικής. Το τρίτο κεφάλαιο αφορά την ποιότητα λογισμικού και ξεκινάει αναφέροντας πέντε σημαντικές οπτικές γωνίες που μπορεί κανείς να την προσεγγίσει. Μετά δίνουμε τους πιο σημαντικούς ορισμούς της ποιότητας και εξηγείται γιατί είναι δύσκολο να την μετρήσουμε. Αυτή η δυσκολία δημιούργησε και την ανάγκη για τον ορισμό μοντέλων και προτύπων για την ποιότητα λογισμικού. Αυτά τα μοντέλα έχουν ως στόχο να αποδομήσουν την ποιότητα σε επιμέρους χαρακτηριστικά που ονομάζονται παράγοντες ποιότητας. Αυτοί οι παράγοντες μπορούν να αναλυθούν σε περαιτέρω χαρακτηριστικά που στη συνέχεια μπορούν να μετρηθούν. Παρουσιάζουμε τα κυριότερα μοντέλα ποιότητας για τις παραδοσιακές μεθόδους ανάπτυξης λογισμικού και αναφέρουμε κάποια μοντέλα που έχουν δημιουργηθεί ειδικά για το λογισμικό ανοικτού κώδικα. Δίνονται οι ορισμοί της μέτρησης, της μετρικής, γίνεται η διάκριση εσωτερικών και εξωτερικών μετρικών και το κεφάλαιο ολοκληρώνεται με μια εκτενή αναφορά στις εσωτερικές μετρικές όπου έχουν χωριστεί σε δύο μεγάλες κατηγορίες. Η πρώτη αφορά τις λεγόμενες παραδοσιακές μετρικές λογισμικού που αναπτύχτηκαν κατά κύριο λόγο για τις διαδικαστικές γλώσσες προγραμματισμού. Η δεύτερη αφορά τις αντικειμενοστρεφείς μετρικές που όπως φανερώνει και το όνομα τους αφορούν αποκλειστικά τις αντικειμενοστρεφείς γλώσσες προγραμματισμού. Ο αναγνώστης που έχει βασικές γνώσεις στα συγκεκριμένα γνωστικά αντικείμενα μπορεί να προσπεράσει τα δύο αυτά κεφάλαια, αφού ο σκοπός τους είναι εισαγωγικός για την ανάπτυξη του απαραίτητου υπόβαθρου κατανόησης που είναι απαραίτητο για τη συνέχεια.

Μεθοδολογία - Επιλογές (Κεφάλαιο 4)**:** Ξεκινάμε μια σύγκριση των κυριότερων συλλογών αντικειμενοστρεφών μετρικών και ορίζουμε συγκεκριμένα κριτήρια που πρέπει να έχει μια μετρική για να επιλεγεί ως ανεξάρτητη μεταβλητή στα μοντέλα μας. Συνεχίζουμε με μια μεγάλη έρευνα για να βρούμε εργαλεία που να υποστηρίζουν τις συγκεκριμένες μετρικές. Οι προϋποθέσεις που θέσαμε για τα εργαλεία αυτά ήταν: α) να είναι δωρεάν διαθέσιμα, β) να μπορούν να υπολογίσουν τις μετρικές στην γλώσσα προγραμματισμού Java και γ) να αναφέρουν τα αποτελέσματα σε επίπεδο κλάσης. Το πιο πλήρες εργαλείο για τους σκοπούς μας αναδείχτηκε το ckjm στην επεκταμένη του έκδοση που υποστηρίζει δέκα εννέα διαφορετικές μετρικές (μια περιγραφή όλων των εργαλείων που αξιολογήθηκαν στα πλαίσια της έρευνας μας δίνεται στο παράρτημα Ε). Στη συνέχεια αναζητήσαμε ένα πρόγραμμα ανοικτού κώδικα που να είναι γραμμένο στην Java, να είναι μεσαίου μεγέθους και να υπάρχουν διαθέσιμα τα σφάλματα που διορθώθηκαν σε κάθε έκδοση του. Τελικά επιλέξαμε την έκδοση 3.2 του προγράμματος jEdit που είναι ένα δημοφιλές πρόγραμμα διόρθωσης κειμένου (text editor) ειδικά για προγραμματιστές. Κατεβάσαμε τον πηγαίο κώδικα της έκδοσης 3.2 του jEdit και υπολογίσαμε τα αποτελέσματα των μετρικών χρησιμοποιώντας το εργαλείο ckjm extented. Υπολογίσαμε χειροκίνητα τη μέγιστη και τη μέση κυκλωματική πολυπλοκότητα για κάθε κλάση του πηγαίου κώδικα, αφού η έξοδος του ckjm δίνει την κυκλωματική πολυπλοκότητα σε επίπεδο συνάρτησης κάθε κλάσης, Μετά χρησιμοποιήσαμε το πρόγραμμα ανοικτού κώδικα BugInfo για να συλλέξουμε όλα τα σφάλματα της έκδοσης 3.2 του jEdit από τις καταχωρήσεις του ιστορικού (log files) του συστήματος διαχείρισης κώδικα SVN. Η αυτοματοποιημένη διαδικασία του BugInfo δεν εγγυάται την πληρότητα των σφαλμάτων που συλλέγει οπότε αναγκαστήκαμε να κάνουμε διορθώσεις που προήλθαν από τη χειροκίνητη αντιστοίχιση της διόρθωσης λαθών με τα συγκεκριμένα αρχεία κλάσεων του πηγαίου κώδικα που διορθώθηκαν. Έτσι ολοκληρώθηκε η διαδικασία συλλογής των δεδομένων, αφού είχαμε τα επεξεργασμένα αποτελέσματα των μετρικών από το ckjm extented για κάθε κλάση αντιστοιχισμένα με τον αριθμό λαθών από τη δημόσια διάθεση της έκδοσης 3.2 του προγράμματος jEdit μέχρι να βγει η νέα του έκδοση 4.0.

Έχουμε φτάσει πλέον στο σημείο που θα πρέπει να επιλέξουμε τα μοντέλα που θα χρησιμοποιηθούν. Ένα μοντέλο ουσιαστικά είναι μια συνάρτηση της μορφής `Y = f(X)`, όπου `X` είναι το διάνυσμα των ανεξάρτητων μεταβλητών και `Y` το διάνυσμα των εξαρτημένων μεταβλητών. Για τα μοντέλα μας η εξαρτημένη μεταβλητή Y ήταν ο αριθμός των σφαλμάτων ή η ύπαρξη σφάλματος σε μια κλάση και οι ανεξάρτητες μεταβλητές `X` ήταν οι μετρικές που συλλέξαμε για την συγκεκριμένη κλάση. Έγινε εκτεταμένη αναζήτηση στη βιβλιογραφία προκειμένου να επιλέξουμε τις τεχνικές μοντελοποίησης που θα χρησιμοποιήσουμε. Καταλήξαμε

σε δύο μεθόδους στατιστικής ανάλυσης και δύο μεθόδους μηχανικής μάθησης. Αναζητήθηκαν και τα αντίστοιχα εργαλεία που υποστηρίζουν τις συγκεκριμένες μεθόδους με μόνα κριτήρια να είναι ανοικτού κώδικα και να υπάρχει μεγάλη κοινότητα χρηστών στην περίπτωση που θα χρειαζόμασταν κάποιας μορφής υποστήριξη. Επιλέξαμε το πρόγραμμα R για τις στατιστικές μεθόδους και το WEKA για τη μηχανική μάθηση:

- **Στατιστική Ανάλυση**: Για την εύρεση του αριθμού των σφαλμάτων σε μία κλάση χρησιμοποιήσαμε απλή παλινδρόμηση για να εξετάσουμε τη σχέση κάθε μετρικής ξεχωριστά με τον αριθμό των σφαλμάτων. Μετά πολλαπλή βηματική παλινδρόμηση με επιλογή μεταβλητών προς τα εμπρός (forward stepwise selection) για να διερευνήσουμε τη συνδυασμένη επίδραση τους στην πρόβλεψη του αριθμού των σφαλμάτων. Η δεύτερη μέθοδος που εφαρμόσαμε είναι η δυαδική λογιστική παλινδρόμηση για την πρόβλεψη της ύπαρξης σφάλματος σε μία κλάση. Με απλή λογιστική προσπαθήσαμε να απομονώσουμε την επίδραση κάθε μετρικής ξεχωριστά και στη συνέχεια με την πολλαπλή βηματική λογιστική παλινδρόμηση με επιλογή μεταβλητών προς τα εμπρός εξακριβώσαμε την συνδυαστική τους επίδραση για την πρόβλεψη σφαλμάτων. Για την εφαρμογή των στατιστικών μεθόδων χρησιμοποιήσαμε το πρόγραμμα ανοικτού κώδικα R και για τις γραφικές παραστάσεις το SPSS (Τα script που δημιουργήσαμε για τα μοντέλα στο R δίνονται στο παράρτημα Α και τα αναλυτικά αποτελέσματα τους για περεταίρω έρευνα στο παράρτημα Β).

- **Μηχανική Μάθηση**: Τα δέντρα απόφασης (decision trees) είναι πολύ χρήσιμα γιατί το αποτέλεσμα τους μπορεί να αναπαρασταθεί γραφικά και να εξαχθούν κανόνες της μορφής "**AN** X **TOTE** Y". Τα τεχνητά νευρωνικά δίκτυα χρησιμοποιούνται σε ένα ευρύτατο φάσμα επιστημονικών περιοχών για πρόβλεψη από τα οικονομικά μέχρι και την ιατρική. Στα μειονεκτήματα τους θα μπορούσε κάποιος να αναφέρει ότι δεν μας παρέχουν κάποια ερμηνεία για το φαινόμενο που μελετάται και δεν υπάρχει κάποιος κανόνας για να ανακαλύψουμε τη βέλτιστη αρχιτεκτονική τους. Χρειάζονται αρκετές δοκιμές και ποτέ δεν ξέρουμε αν κάποια άλλη αρχιτεκτονική θα έδινε καλύτερα αποτελέσματα από αυτή που έχουμε επιλέξει. Για τα δέντρα απόφασης χρησιμοποιήσαμε στο WEKA τον αλγόριθμο J48 που είναι μια μικρή παραλλαγή του C4.5 και για τα νευρωνικά δίκτυα τον αλγόριθμο MultilayerPerceptron που είναι ένα πολυεπίπεδο τεχνητό νευρωνικό δίκτυο αισθητήρα που εκπαιδεύεται με την μέθοδο της οπίσθιας διάδοσης του λάθους (error back propagation). Όλα τα αναλυτικά αποτελέσματα για τις μεθόδους μηχανικής μάθησης υπάρχουν στο παράρτημα Γ.

Πειραματικά Αποτελέσματα - Ανάλυση (Κεφάλαιο 5)**:** Τα αποτελέσματα μας αρχίζουν με περιγραφική στατιστική για τις κλάσεις και τις μετρικές που αφορούν το πρόγραμμα jEdit έκδοση 3.2. Μετά εξετάζουμε τις πιθανές συσχετίσεις μεταξύ των μετρικών με τη χρήση του συντελεστή Spearman. Η πρώτη μέθοδος που χρησιμοποιούμε για την κατασκευή μοντέλων είναι η απλή γραμμική παλινδρόμηση όπου διερευνούμε κατά πόσο κάθε μετρική ξεχωριστά μπορεί να μας βοηθήσει να προβλέψουμε τον αριθμό των λαθών που χρειάστηκαν διόρθωση σε μια κλάση. Μετά χρησιμοποιούμε βηματική πολλαπλή γραμμική παλινδρόμηση όπου η προσθήκη των ανεξάρτητων μεταβλητών γίνεται με την επιλογή προς τα μπρος (stepwise forward selection). Η δεύτερη μέθοδος μας είναι η δυαδική λογιστική παλινδρόμηση όπου τώρα η εξαρτημένη μεταβλητή είναι δυαδική με τιμές την ύπαρξη ή όχι προβλήματος σε μια κλάση. Με την έννοια πρόβλημα εννοούμε στα πλαίσια του πειράματος μας την ύπαρξη ενός ή περισσότερων σφαλμάτων στη συγκεκριμένη κλάση. Συνεχίζουμε με δύο μεθόδους που ανήκουν στη γενικότερη κατηγορία της μηχανικής μάθησης, το δέντρο απόφασης και το τεχνητό νευρωνικό δίκτυο, όπου πάλι εξετάζουμε κατά πόσο προκύπτουν μοντέλα που προβλέπουν επιτυχώς την ύπαρξη προβλήματος σε μια κλάση. Για την αποτίμηση κάθε μοντέλου κατηγοριοποίησης που κατασκευάσαμε, επιλέξαμε να χρησιμοποιηθεί η διασταυρωμένη επικύρωση σε 10 μέρη (10-fold cross validation). Το μοντέλο που προέκυψε από την εφαρμογή της δυαδικής πολλαπλής λογιστικής παλινδρόμησης με επιλογή προς τα μπρος έδωσε τα καλύτερα αποτελέσματα στην πρόβλεψη σφαλμάτων, όμως με μικρή διαφορά από αυτά του τεχνητού νευρωνικού δικτύου. Σχολιάζουμε τα αποτελέσματα του καλύτερου μοντέλου κάθε διαφορετικής μεθόδου μοντελοποίησης που εφαρμόσαμε και γίνεται μια σύγκριση της απόδοσης τους με τις πιο σημαντικές παρόμοιες μελέτες. Ιδιαίτερα αναλυτική συζήτηση γίνεται για τη συλλογή μετρικών CK όπου συγκρίνουμε τα δικά μας αποτελέσματα με αυτά των πιο σημαντικών αντίστοιχων ερευνών.

Συμπεράσματα - Επεκτάσεις (Κεφάλαιο 6)**:** Τα κυριότερα συμπεράσματα της έρευνας μας μπορούν να συνοψιστούν στα εξής:

1. Οι αντικειμενοστρεφείς μετρικές δίνουν ικανοποιητικά αποτελέσματα σε μοντέλα για την έγκαιρη πρόγνωση σφαλμάτων σε προγράμματα ανοικτού κώδικα.

2. Οι παραδοσιακές μετρικές και ειδικότερα ο αριθμός των γραμμών του πηγαίου κώδικα LOC και η μέγιστη κυκλωματική πολυπλοκότητα μιας κλάσης CC_MAX συνδέονται θετικά και στατιστικά σημαντικά (σε επίπεδο σημαντικότητας 1%) με την ύπαρξη

σφάλματος σε μια κλάση. Αντίθετα, τα μοντέλα μηχανικής μάθησης επέδειξαν πολύ χαμηλή ικανότητα πρόβλεψης ύπαρξης σφάλματος σε μια κλάση.

3. Οι στατιστικές τεχνικές έδωσαν μοντέλα με καλύτερη προσαρμοστικότητα από αυτές της μηχανικής μάθησης.

4. Δεν υπάρχει μια συλλογή μετρικών που να δίνει από μόνη της ανώτερα αποτελέσματα. Τα καλύτερα μοντέλα είχαν ως είσοδο όλες τις μετρικές ή ένα υποσύνολο αυτών με μετρικές από διάφορες συλλογές.

5. Η συλλογή των μετρικών CK επιβεβαίωσε τη σημαντικότητα της αφού οι τέσσερις (WMC, DIT, CBO και RFC) από τις έξι μετρικές της συλλογής, βρέθηκαν να επηρεάζουν θετικά και στατιστικά σημαντικά (σε επίπεδο σημαντικότητας 1%) την πιθανότητα ύπαρξης σφάλματος σε μια κλάση.

Η μεταπτυχιακή διατριβή ολοκληρώνεται με προτάσεις για περαιτέρω έρευνα:

- **Περισσότερα Προγράμματα**: Για πιο γενικά συμπεράσματα θα πρέπει να εξεταστούν περισσότερα προγράμματα ανοικτού κώδικα και μάλιστα το ιδανικό θα ήταν να μην επιλεγούν τυχαία αλλά μέσω μιας σχετικής μεθοδολογίας που θα πρέπει να περιλαμβάνει συγκεκριμένα κριτήρια επιλογής και στόχους.

- **Διαφορετικές Γλώσσες Προγραμματισμού**: Η διερεύνηση της καταλληλότητας των αντικειμενοστρεφών μετρικών για την πρόβλεψη σφαλμάτων και σε άλλες γλώσσες προγραμματισμού πέρα από την Java που χρησιμοποιήσαμε αλλά και την C++ για την οποία υπάρχουν ανάλογες εργασίες.

- **Περισσότερες Μετρικές**: Υπολογίστηκαν δεκαεπτά αντικειμενοστρεφείς μετρικές με το εργαλείο ανοικτού κώδικα ckjm extended. Όμως, υπάρχουν εκατοντάδες αντικειμενοστρεφείς μετρικές που έχουν προταθεί στην βιβλιογραφία και που πιθανώς θα μπορούσαν να βελτιώσουν την απόδοση των μοντέλων πρόβλεψης σφαλμάτων.

- **Μοντέλα Εκτίμησης Προσπάθειας**: Μια ενδιαφέρουσα επέκταση στην παρούσα έρευνα θα ήταν επιπρόσθετα της εκτίμησης για την ύπαρξη σφάλματος σε μια κλάση, να υπάρχει εκτίμηση και της προσπάθειας που θα απαιτηθεί για την διόρθωση του.

- **Κατηγοριοποίηση Σφαλμάτων**: Η έρευνα μας δίνει την ίδια βαρύτητα σε όλα τα σφάλματα του λογισμικού. Όμως, μεγαλύτερη αξία έχουν τα σφάλματα που κρίνονται σημαντικά π.χ. που μπορούν να οδηγήσουν στο απότομο τερματισμό της εφαρμογής. Μια πιο ολοκληρωμένη προσέγγιση θα μπορούσε να λαμβάνει υπ' όψιν της και την σοβαρότητα του κάθε σφάλματος και να κάνει αντίστοιχες προβλέψεις.

- **Δημόσιο Αποθετήριο Δεδομένων**: Η δημιουργία και λειτουργία ενός δημόσιου αποθετηρίου δεδομένων ειδικά για μετρικές προγραμμάτων ανοικτού λογισμικού θα έδινε τεράστια ώθηση σε αντίστοιχες ερευνητικές προσπάθειες.

- **Εργαλείο Συλλογής Στοιχείων Σφαλμάτων**: Για τη διευκόλυνση της προσπάθειας δημιουργίας μοντέλων για την πρόβλεψη σφαλμάτων είναι επιτακτική ανάγκη τόσο ο ορισμός αξιόπιστων διαδικασιών για την καταχώρηση ή για την διόρθωση των σφαλμάτων από τους προγραμματιστές όσο και νέων εργαλείων που θα αυτοματοποιήσουν τη συλλογή αυτής της πολύτιμης πληροφορίας.

- **Περισσότεροι Αλγόριθμοι Μηχανικής Μάθησης**: Η επιλογή των πιο αποδοτικών αλγορίθμων μηχανικής μάθησης για ένα πρόβλημα δεν είναι κάτι εύκολο και απαιτούνται πολλοί πειραματισμοί για την εξεύρεση της πιο αποδοτικής λύσης. Πέρα από τους δύο αλγόριθμους μηχανικής μάθησης που εφαρμόσαμε στην παρούσα έρευνα, υπάρχουν πολλοί άλλοι για τους οποίους θα μπορούσε να εκτιμηθεί η χρησιμότητα τους.

Κεφάλαιο 2
Λογισμικό Ανοικτού Κώδικα

Πολλές φορές γίνεται σύγχυση μεταξύ των όρων "Ελεύθερο Λογισμικό" και "Λογισμικό Ανοικτού Κώδικα" που χρησιμοποιούνται στις περισσότερες περιπτώσεις ως ταυτόσημοι. Στην πρώτη ενότητα του κεφαλαίου ξεκαθαρίζουμε τις διαφορές τους και δίνουμε τα απαραίτητα κριτήρια με βάση τους επίσημους ορισμούς τους που πρέπει να πληρεί ένα πρόγραμμα για να ανήκει στο λογισμικό ελεύθερου ή του ανοικτού κώδικα. Στην δεύτερη ενότητα προσπαθούμε να αναδείξουμε τις ιδιαιτερότητες που υπάρχουν στο μοντέλο ανάπτυξης του λογισμικού ανοικτού κώδικα σε σχέση με τον παραδοσιακό κύκλο ζωής ενός πληροφοριακού συστήματος που αποτελείται από τον προγραμματισμό, την ανάλυση, την σχεδίαση, την υλοποίηση και την υποστήριξη. Αναφέρουμε αρκετά μοντέλα ανάπτυξης που έχουν αναφερθεί στην βιβλιογραφία και σκιαγραφούμε την αντιστοίχιση του παραδοσιακού κύκλου ζωής με αυτό των έργων ανοικτού κώδικα. Στην τελευταία ενότητα παρουσιάζουμε τα κυριότερα πλεονεκτήματα τους αλλά γίνεται και μια αναφορά στα πιθανά μειονεκτήματα ή προβλήματα μπορεί να έχει η υιοθέτηση ενός λογισμικού ανοικτού κώδικα από μια εταιρία ή έναν μεγάλο οργανισμό. Κλείνουμε το κεφάλαιο με μια γρήγορη αναφορά σε τομείς ενδιαφέροντος σχετικά με τα θέματα που χρήζουν μελέτης και από άλλες επιστήμες, πέραν της πληροφορικής ή των άμεσα αυτής συγγενών επιστημών. Πολύ ενδιαφέροντες μελέτες για το λογισμικό ανοικτού κώδικα μπορεί να βρει κανείς σε έρευνες οικονομικών επιστημών, κοινωνιολογίας, ανθρωπολογίας κλπ

2.1 Ιστορική Αναδρομή και Ορισμοί

Ο όρος "Λογισμικό Ανοικτού Κώδικα" (Open Source Software) είναι νεότερος από αυτόν του "Ελεύθερου Λογισμικού" (Free Software). Οι δυο όροι αν και έχουν πολλά κοινά χαρακτηριστικά και πολλές φορές χρησιμοποιούνται χωρίς διακρίσεις, δεν είναι απολύτως ταυτόσημοι. Ελεύθερο λογισμικό υπήρχε από τα πρώτα χρόνια της πληροφορικής, αφού οι προγραμματιστές μοίραζαν το λογισμικό που είχαν αναπτύξει και μάλιστα δεν ήταν σπάνια πρακτική να βοηθάει ο ένας τον άλλον σε δύσκολα προβλήματα συνεισφέροντας πηγαίο κώδικα. Όμως, τα επόμενα χρόνια με το σταδιακό διαχωρισμό του λογισμικού από το υλικό άρχισε να δημιουργείται μια νέα αγορά όπου στο εμπορικό λογισμικό διανέμεται μόνο η εκτελέσιμη μορφή του προγράμματος, ενώ πρόσβαση στον πηγαίο κώδικα του είχαν μόνο τα στελέχη της εταιρίας ανάπτυξης.

Η πιο καθοριστική πρωτοβουλία στο χώρο του ελεύθερου λογισμικού ήταν η δημιουργία του Ιδρύματος Ελεύθερου Λογισμικού (Free Software Foundation) στις 4 Οκτωβρίου του 1985 από τον Richard Stallman [029]. Μία από τις πρώτες ενέργειες του Ιδρύματος FSF ήταν να δημιουργήσει τον ορισμό του ελεύθερου λογισμικού που δεν μένει σταθερός αλλά αναθεωρείται κάθε φορά που προκύπτει ανάγκη. Στην τρέχουσα έκδοσή του ως ελεύθερο λογισμικό ορίζεται αυτό που παρέχει στον χρήστη του προγράμματος τις παρακάτω τέσσερις ελευθερίες [111]:

- **Ελευθερία 0:** Να μπορεί να χρησιμοποιήσει το πρόγραμμα για οποιοδήποτε σκοπό.

- **Ελευθερία 1:** Να μπορεί να μελετήσει πώς λειτουργεί το πρόγραμμα, και να μπορεί να το μεταβάλλει με βάση τις δικές του απαιτήσεις. Η πρόσβαση στον πηγαίο κώδικα είναι μια προϋπόθεση για αυτή την ελευθερία.

- **Ελευθερία 2:** Να μπορεί να διανέμει ελευθέρα αντίγραφα του προγράμματος, ώστε να μπορεί να βοηθήσει όποιον το χρειάζεται.

- **Ελευθερία 3:** Να μπορεί να αναδιανέμει τα αντίγραφα με τις βελτιώσεις σε άλλους. Κάνοντας κάτι τέτοιο βοηθάει όλη την κοινότητα ώστε να ευεργετηθεί από τις αλλαγές. Η πρόσβαση στον πηγαίο κώδικα είναι προϋπόθεση για αυτή την ελευθερία.

Στα τέλη της δεκαετίας του '90 άρχισε να χρησιμοποιείται άλλος ένας όρος, αυτός του "Λογισμικού Ανοικτού Κώδικα" που τελικά οδήγησε και στη δημιουργία ενός μη κερδοσκοπικού οργανισμού με την ονομασία Πρωτοβουλία Ανοικτού Κώδικα (Open Source Initiative) τον

Φεβρουάριο του 1998 από τους Bruce Perens και Eric Raymond [097]. Το κίνημα του ελεύθερου λογισμικού έχει ως κίνητρο την ηθική σχετικά με το λογισμικό και τις ελευθερίες που έχουν να κάνουν με αυτό, ενώ το κίνημα του ανοικτού κώδικα επικεντρώνεται κυρίως σε πιο πρακτικά θέματα και δεν ασχολείται με θέματα που αφορούν σε αρχές και δικαιώματα που έχουν οι χρήστες. Όπως και στην περίπτωση του ελεύθερου λογισμικού, υπάρχει ένας επίσημος ορισμός που έχει αναθεωρηθεί αρκετές φορές όλα αυτά τα χρόνια. Για να χαρακτηριστεί ένα λογισμικό ως ανοικτού κώδικα θα πρέπει να πληρεί τα παρακάτω δέκα κριτήρια [112]:

1. **Ελεύθερη Αναδιανομή:** Η άδεια δεν πρέπει να περιορίζει κάποιον από το να πουλήσει ή να δώσει το λογισμικό ως μέρος ενός άλλου λογισμικού.

2. **Πηγαίος Κώδικας:** Το πρόγραμμα πρέπει να περιέχει τον πηγαίο κώδικα.

3. **Παραγόμενο Λογισμικό:** Πρέπει να επιτρέπονται αλλαγές ή παράγωγα προϊόντα.

4. **Ακεραιότητα Πηγαίου Κώδικα του Συγγραφέα:** Η άδεια μπορεί να περιορίζει τον πηγαίο κώδικα από την αναδιανομή σε άλλη τροποποιημένη έκδοση.

5. **Καμία Διάκριση Εναντίων Προσώπων ή Ομάδων:** Η άδεια χρήσης δεν πρέπει να βλάπτει κανένα άτομο ή ομάδα ατόμων.

6. **Καμία Διάκριση ως προς τα Πεδία της Χρήσης:** Πρέπει να επιτρέπονται όλες οι πιθανές χρήσεις του προγράμματος από την άδεια χρήσης του.

7. **Διανομή της Άδειας:** Τα δικαιώματα που απορρέουν από το πρόγραμμα θα πρέπει να έχουν εφαρμογή και σε όποιον χρησιμοποιεί το πρόγραμμα.

8. **Η Άδεια δεν Πρέπει να Είναι Συγκεκριμένη για ένα Λογισμικό:** Τα δικαιώματα που απορρέουν από το πρόγραμμα δεν πρέπει να εξαρτώνται από το αν το πρόγραμμα είναι μέρος ενός πακέτου λογισμικού.

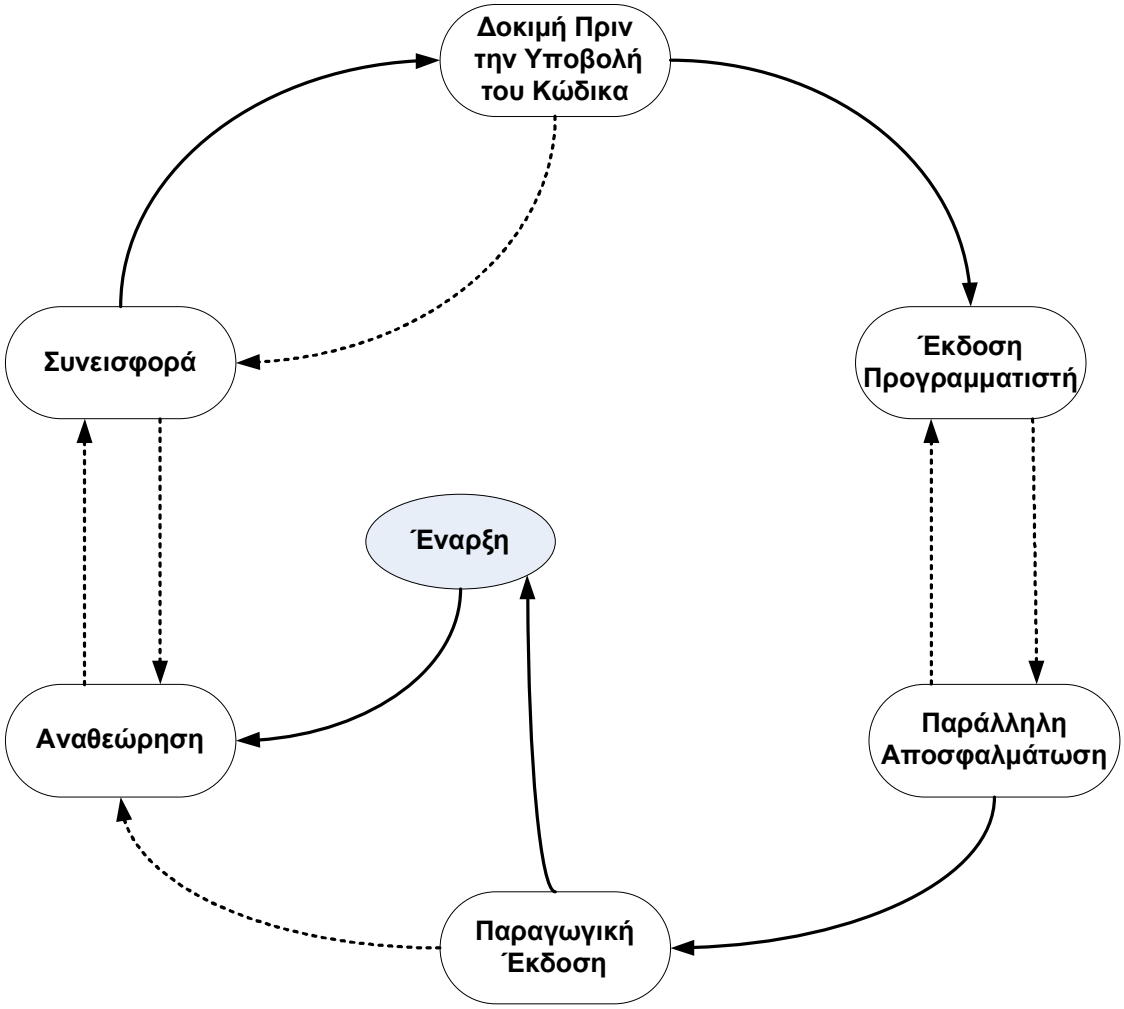
9. **Η Άδεια δεν Πρέπει να Περιορίζει Άλλο Λογισμικό:** Η άδεια δεν πρέπει να βάζει κανένα περιορισμό όσο αφορά άλλο λογισμικό που διανέμεται μαζί με αυτό.

10. **Η Άδεια θα Πρέπει να Είναι Ουδέτερη Τεχνολογίας:** Κανένας όρος της άδειας χρήσης δεν πρέπει να εξαρτάται από μία συγκεκριμένη τεχνολογία.

2.2 Μοντέλα και Χαρακτηριστικά Ανάπτυξης

Υπάρχουν αρκετές διαφορές μεταξύ της διαδικασίας ανάπτυξης των έργων λογισμικού ανοικτού κώδικα και των παραδοσιακών μεθόδων που ακολουθούσαν μέχρι πρόσφατα οι περισσότερες εταιρείες εμπορικού λογισμικού. Ο κύκλος ζωής ανάπτυξης πληροφοριακών συστημάτων (system development life cycle) των παραδοσιακών μεθόδων περιλαμβάνει γενικές φάσεις όπου όλες οι εργασίες του έργου μπορούν να οργανωθούν όπως είναι ο προγραμματισμός εργασιών, η ανάλυση, η σχεδίαση, η υλοποίηση και η υποστήριξη [105]. Κατά αντιστοιχία τα περισσότερα έργα ανοικτού κώδικα επιδεικνύουν μεταξύ τους αρκετά κοινά χαρακτηριστικά όπως είναι η παράλληλη ανάπτυξη, η αναθεώρηση ομότιμων (peer review), η παρακίνηση για την εμπλοκή των χρηστών (user feedback) και τη συμβολή των προγραμματιστών, η παράλληλη αποσφαλμάτωση του πηγαίου κώδικα, η προσέλκυση ιδιαίτερα ταλαντούχων προγραμματιστών και οι γρήγοροι ρυθμοί δημοσιοποίησης νέων εκδόσεων. Υπήρξαν διάφορες προσπάθειες για την δημιουργία μοντέλων που θα περιέγραφαν την διαδικασία ανάπτυξης των έργων ανοικτού κώδικα και έκαναν την αντιστοίχιση με τις παραδοσιακές τεχνικές.

Ο Jorgensen [106] ανέπτυξε ένα μοντέλο που αποτελεί μία αναλυτική περιγραφή των δραστηριοτήτων που είναι απαραίτητες για την υποστήριξη των διαδικασιών που ακολουθούν τα έργα ανοικτού κώδικα και είναι εμπνευσμένο από τον κύκλο ζωής των αλλαγών του έργου Free BSD. Αυτό το μοντέλο περιλαμβάνει τον κώδικα, την αναθεώρηση (review), τη δοκιμή πριν την υποβολή του κώδικα (pre-commit test), την έκδοση προγραμματιστή (development release), την παράλληλη αποσφαλμάτωση και την παραγωγική έκδοση (production release). Αν και το παραπάνω μοντέλο έχει γίνει αποδεκτό από διάφορους ερευνητές [096, 107] ως ένα γενικότερο πλαίσιο για τις δραστηριότητες σε έργα ανοικτού κώδικα, παρουσιάζει και κάποιες αδυναμίες. Η βασικότερη εξ αυτών είναι ότι όταν εφαρμόζουμε το μοντέλο σε ένα έργο ανοικτού κώδικα, δεν μας εξηγεί με ικανοποιητικό τρόπο πώς και πού συμβαίνουν οι δραστηριότητες όπως είναι π.χ. ο προγραμματισμός, η ανάλυση και η σχεδίαση. Ο Wynn [108] επέκτεινε τον κύκλο ζωής ανάπτυξης προσθέτοντας τη φάση της ωρίμανσης, στην οποία το έργο εισέρχεται όταν το έργο καταφέρει να φτάσει ένα κρίσιμο αριθμό από χρήστες και προγραμματιστές που δεν μπορεί να υποστηρίξει λόγω διαχειριστικών και άλλων περιορισμών. Όμως το μοντέλο που πρότειναν οι Roets et al [109] είναι αυτό που επεκτείνει πραγματικά όλα τα προηγούμενα μοντέλα και καταφέρνει σε ικανοποιητικό βαθμό να αντιστοιχίσει τις φάσεις του παραδοσιακού κύκλου ανάπτυξης πληροφοριακών συστημάτων με αυτές της ανάπτυξης των έργων ανοικτού κώδικα και παρουσιάζεται στο σχήμα 2.1.



Σχήμα 2.1: Μοντέλο Κύκλου Ζωής για Έργα Ανοικτού Λογισμικού

Η αντιστοίχιση που προκύπτει έχει κωδικοποιηθεί στον πίνακα 2.1. Η έναρξη ενός έργου ανοικτού κώδικα συνδυάζει τρεις φάσεις του παραδοσιακού κύκλου, τον προγραμματισμό εργασιών, την ανάλυση και τη σχεδίαση. Η σχεδίαση είναι η πιο σημαντική δραστηριότητα γιατί αν γίνει σωστά όλοι οι προγραμματιστές μετά δουλεύουν σε ένα ξεκάθαρα ορισμένο σκοπό. Η φάση της υλοποίησης του παραδοσιακού κύκλου αντιστοιχεί στις δραστηριότητες της αναθεώρησης, της συνεισφοράς, των δοκιμών πριν την υποβολή του κώδικα και την έκδοση για τον προγραμματιστή. Όσο περισσότεροι χρήστες αρχίζουν να εμπλέκονται στη διαδικασία του έργου ανοικτού κώδικα τότε η διεξαγωγή των παράλληλων δοκιμών και νέες διαφορετικές εκδόσεις μας οδηγούν στην παραγωγική έκδοση του λογισμικού που αντιστοιχεί στη φάση της υποστήριξης του παραδοσιακού κύκλου. Ένα άλλο μοντέλο που έχει συζητηθεί πάρα πολύ είναι του Raymond [028] που περιγράφεται στο κλασσικό πλέον κείμενο του "The Cathedral and the Bazaar". Η παραδοσιακή ανάπτυξη λογισμικού παρουσιάζεται σαν την διαδικασία ανέγερσης ενός ναού, όπου η εργασία γίνεται από έναν αρχιτέκτονα που δουλεύει απομονωμένος χωρίς την συμμετοχή των υπολοίπων που συμμετέχουν στο έργο. Αντίθετα, ο τρόπος ανάπτυξης των έργων ανοικτού κώδικα θυμίζει παζάρι εξαιτίας της ανοικτής και ανοργάνωτης φύσης του.

<table>
<tr><th>Παραδοσιακός Κύκλος Ανάπτυξης</th><th>Κύκλος Ανάπτυξης Λογισμικού Ανοικτού Κώδικα</th></tr>
<tr><td>Προγραμματισμός Εργασιών</td><td rowspan="3">Έναρξη</td></tr>
<tr><td>Ανάλυση</td></tr>
<tr><td>Σχεδίαση</td></tr>
<tr><td rowspan="4">Υλοποίηση</td><td>Αναθεώρηση</td></tr>
<tr><td>Συνεισφορά</td></tr>
<tr><td>Δοκιμές πριν την Υποβολή του Κώδικα</td></tr>
<tr><td>Έκδοση Προγραμματιστή</td></tr>
<tr><td rowspan="2">Υποστήριξη</td><td>Παράλληλη Αποσφαλμάτωση</td></tr>
<tr><td>Παραγωγική Έκδοση</td></tr>
</table>

Πίνακας 2.1: Αντιστοίχιση Παραδοσιακού Κύκλου Ζωής με Κύκλο Ζωής Λογισμικού Ανοικτού Κώδικα

Ένα πιο πρόσφατο μοντέλο που στηρίζεται περισσότερο στη δομή της ομάδας ανάπτυξης αλλά και των υπολοίπων ατόμων που συμμετέχουν ενεργά σε ένα έργο ανοικτού κώδικα, έχει προταθεί από τους Crowston και Howison [110]. Σε αυτό το μοντέλο η κοινότητα ενός έργου λογισμικού παρομοιάζεται με ένα "κρεμμύδι" που έχει τρία διαφορετικά επίπεδα με το πιο μικρό να βρίσκεται στον πυρήνα του. Στο εσωτερικό επίπεδο βρίσκεται η βασική ομάδα ανάπτυξης του έργου, στο επόμενο επίπεδο βρίσκεται μια μεγαλύτερη ομάδα ατόμων που συνεισφέρουν πηγαίο κώδικα και στο τελευταίο εξωτερικό επίπεδο μια ακόμα μεγαλύτερη ομάδα ατόμων που δεν συνεισφέρουν κώδικα αλλά κάνουν μία σειρά από άλλες χρήσιμες δραστηριότητες όπως είναι οι δοκιμές των νέων εκδόσεων, οι αναφορές σφαλμάτων, η συγγραφή της τεκμηρίωσης του προγράμματος κλπ. Οι ομάδες δεν είναι σταθερές αλλά υπάρχει μετακίνηση ατόμων μεταξύ των διαφορετικών επιπέδων π.χ. ένα άτομο μπορεί να ανέβει στο επόμενο επίπεδο με αξιοκρατικές διαδικασίες που μπορεί να περιλαμβάνουν ψηφοφορίες ή την επίτευξη κάποιου δύσκολου στόχου όπως είναι η επίλυση ενός πολύ δύσκολου σφάλματος της εφαρμογής.

2.3 Πλεονεκτήματα, Μειονεκτήματα και Πεδία Έρευνας

Το λογισμικό ανοικτού κώδικα έχει πάρα πολλά χαρακτηριστικά που του προσφέρουν σημαντικά πλεονεκτήματα σε σχέση με το εμπορικό λογισμικό. Θα ήταν ιδιαίτερα δύσκολο να γίνει μία εξαντλητική αναφορά τους, ξεχωρίσαμε μερικά ως πιο σημαντικά:

- **Πρόσβαση στον Πηγαίο Κώδικα:** Καθένας μπορεί να χρησιμοποιήσει τον πηγαίο κώδικα είτε για λόγους εκπαίδευσης είτε για να τον χρησιμοποιήσει σε άλλο λογισμικό ως εξωτερικό συστατικό (component) είτε για να τον τροποποιήσει και να τον χρησιμοποιήσει σύμφωνα με τις ανάγκες του.

- **Γρήγορη Ανανέωση Εκδόσεων:** Η ανάμειξη πάρα πολλών προγραμματιστών στην ανάπτυξη του λογισμικού και η συνεχής ανάδραση από τους χρήστες, έχουν οδηγήσει τα περισσότερα έργα ανοικτού κώδικα να έχουν έναν μικρό κύκλο ανάπτυξης πριν βγει μια νέα έκδοση. Αφού δεν υπάρχει μία εταιρία που πιθανόν να καθυστερούσε μία νέα έκδοση προκειμένου να μεγιστοποιήσει τα κέρδη της, δεν υπάρχει κανένας λόγος να μην υπάρχει συνεχής ενημέρωση νέων εκδόσεων [096].

- **Μεγάλη Κοινότητα Υποστήριξης:** Όλοι οι εμπλεκόμενοι στη διαδικασία ανάπτυξης του λογισμικού από το στενό πυρήνας της ομάδας ανάπτυξης μέχρι και τους απλούς χρήστες έχουν ιδιαίτερα αναπτυγμένη την κουλτούρα για τη διάχυση της γνώσης και της αλληλοβοήθειας. Στα περισσότερα έργα ανοικτού κώδικα παρέχεται εγχειρίδιο χρήσης, αναλυτική ηλεκτρονική βοήθεια σε μορφή wiki με παραδείγματα και άλλες ηλεκτρονικές υπηρεσίες υποστήριξης που προσφέρονται από εθελοντές όπως είναι οι λίστες email, φόρουμ συζητήσεων, σύστημα καταχώρησης σφαλμάτων κλπ.

- **Χωρίς Κόστος Απόκτησης:** Η χρησιμοποίηση του τόσο από μεμονωμένους ιδιώτες όσο και στα πλαίσια ενός οργανισμού ή μιας εταιρίας δεν συνεπάγεται κανένα επιπλέον κόστος προμήθειας για αυτούς. Αντίθετα, η αγορά ενός αντίστοιχου εμπορικού λογισμικού μπορεί να κοστίσει μέχρι και αρκετές χιλιάδες ευρώ.

- **Συνεργατική και Κατανεμημένη Ανάπτυξη:** Η προσέγγιση που ακολουθείται για την ανάπτυξη είναι η συνεργατική αφού δεν υπάρχει ηγέτης αλλά μία ομάδα που αποφασίζει για όλα τα κρίσιμα ζητήματα. Όλοι οι εμπλεκόμενοι στο έργο δεν συναντιούνται καν κατ' ιδίαν αλλά επικοινωνούν ηλεκτρονικά και είναι διασκορπισμένοι σε όλο τον κόσμο [097].

Παρά τα παραπάνω πλεονεκτήματα, υπάρχουν όπως είναι φυσικό και κάποια μειονεκτήματα που θα πρέπει να τα έχουμε υπ' όψιν μας όταν αναφερόμαστε σε λογισμικό ανοικτού κώδικα. Η σκοπιά με την οποία προσεγγίζουμε το συγκεκριμένο θέμα αφορά κυρίως την υιοθέτηση του από εταιρίες, οργανισμούς ή δημόσιες υπηρεσίες:

- **Αξιολόγηση Λογισμικού:** Υπάρχουν πάρα πολλά έργα ανοικτού λογισμικού με αποτέλεσμα να είναι πολύ δύσκολη διαδικασία η έρευνα για την αναζήτηση όλων των έργων ανοικτού λογισμικού που μπορούν να υποστηρίξουν με αποτελεσματικό τρόπο μία ανάγκη. Ακόμα πιο δύσκολη είναι η αξιολόγησή τους ως προς εξειδικευμένα θέματα όπως είναι η ασφάλεια, οι επιδόσεις τους, η ευχρηστία τους κλπ

- **Βιωσιμότητα Λογισμικού:** Τα έργα ανοικτού κώδικα στηρίζονται στην εθελοντική συμμετοχή όλων των εμπλεκομένων, κάτι που δεν είναι πάντα εξασφαλισμένο. Υπάρχουν πάρα πολλά προγράμματα ανοικτού κώδικα όπως ο φυλλομετρητής Firefox, οι εφαρμογές γραφείου OpenOffice κλπ που χρησιμοποιούνται από εκατομμύρια χρήστες, μεγάλες εταιρίες και γενικότερα υπάρχει μία τεράστια κοινότητα που τα υποστηρίζει. Ωστόσο, υπάρχουν και πάρα πολλές άλλες περιπτώσεις έργων ανοικτού λογισμικού που είτε δεν είναι ευρέως γνωστά είτε η ανάπτυξη τους έχει σταματήσει. Έτσι, δεν είναι καθόλου εύκολο πλην κάποιων συγκεκριμένων περιπτώσεων να γνωρίζει κάποιος τη βιωσιμότητά τους [098].

- **Έλλειψη Επίσημης Υποστήριξης:** Πολλές φορές υπάρχει η απαίτηση όπως π.χ. σε ανταγωνιστικά εταιρικά περιβάλλοντα ή μεγάλους οργανισμούς να παρέχεται γραπτή υποστήριξη για τα πληροφοριακά συστήματα και μάλιστα με συγκεκριμένες ρήτρες αν οι όροι αυτής δεν ικανοποιηθούν. Δυστυχώς όμως, κανένα λογισμικό ανοικτού κώδικα δεν παρέχει εγγύηση καλής λειτουργίας ούτε ρήτρες για το χρόνο αποκατάστασης κάποιου σημαντικού κενού ασφαλείας ή σφάλματος.

- **Κόστος Λειτουργίας:** Η χρησιμοποίηση λογισμικού ανοικτού κώδικα δεν σημαίνει ότι δεν υπάρχουν και άλλα κόστη που θα πρέπει να αναλάβει αυτός που χρησιμοποιεί το λογισμικό. Πρόσθετα κόστη μπορεί να είναι η παραμετροποίηση του λογισμικού για τις ιδιαιτερότητες του οργανισμού, η ανάπτυξη πρόσθετης λειτουργικότητας που είναι απαραίτητη για τις ανάγκες του, τα κόστη για την εξειδικευμένη εκπαίδευση του προσωπικού που θα το υποστηρίξει ή των χρηστών που θα το χρησιμοποιήσουν, αγορά ή αναβάθμιση υφιστάμενου εξοπλισμού πληροφορικής κλπ

Σχήμα 2.2: Τομείς Ενδιαφέροντος για τα Έργα Ανοικτού Κώδικα

Τα τελευταία δέκα χρόνια υπάρχει μεγάλη δραστηριότητα από διάφορες επιστήμες σχετικά με το φαινόμενο των έργων ανοικτού κώδικα [099]. Στο σχήμα 2.2 παρουσιάζουμε μία γραφική αναπαράσταση για να τονίσουμε το γεγονός ότι τα θέματα με τα οποία ασχολείται η κάθε επιστήμη δεν είναι ξεχωριστά από αυτά των άλλων αλλά διασταυρώνονται σε κάποια σημεία. Μεγάλος αριθμός μελετών προσπαθεί να απαντήσει το ερώτημα: γιατί οι προγραμματιστές συνεισφέρουν στα έργα ανοικτού κώδικα; Στα οικονομικά είναι πολύ γνωστό το πρόβλημα του "ελεύθερου επιβάτη" (free rider) δηλαδή κάθε προγραμματιστής θα μπορούσε να περιμένει να αναπτύξει κάποιος άλλος το λογισμικού (αφού διατίθεται ελεύθερα) και μετά να το χρησιμοποιήσει αυτός. Ο Lerner και Tirole [100] χρησιμοποιώντας οικονομική θεωρεία πρότειναν ένα μοντέλο όπου οι προγραμματιστές χρησιμοποιούν τις ικανότητες που ανέπτυξαν με τη συμμετοχή τους σε έργα ανοικτού κώδικα, για να βελτιώσουν το βιογραφικό τους και έτσι να διεκδικήσουν καλύτερη δουλειά.

Οι Hippel και Krogj [101] συνδυάζοντας θεωρίες από τις οικονομικές επιστήμες και την κοινωνιολογία δημιούργησαν ένα μοντέλο "ιδιωτικότητας-συλλογικότητας" για τα κίνητρα της καινοτομίας. Υποστήριξαν ότι οι προγραμματιστές συνεισφέρουν στη δημόσια καινοτομία γιατί αποκομίζουν ιδιωτικά οφέλη που σχετίζονται με αυτή τη διαδικασία της καινοτομίας. Σε αυτά τα οφέλη μπορεί να περιλαμβάνονται η διασκέδαση, η φήμη, η ευχαρίστηση, η μάθηση, η

αναγνώριση από τους άλλους κλπ. Άλλο ένα ενδιαφέρον πεδίο έρευνας που χρειάζεται περισσότερη προσοχή είναι αν τελικά οι εταιρίες ανταγωνίζονται ή συνεργάζονται με τα έργα ανοικτού κώδικα. Υπάρχουν ενδιαφέρουσες εργασίες από την επιστήμη της διοίκησης για το πώς το λογισμικό ανοικτού κώδικα θα επηρεάσει την στρατηγική των εταιριών που αναπτύσσουν εμπορικό λογισμικό για να το αντιμετωπίσουν [102] ή για το πώς θα μεταβάλει τη στρατηγική που ακολουθούν οι πελάτες όταν επιλέγουν λογισμικό ή τεχνολογικές πλατφόρμες [103]. Οι Dahlander και Magnusson [104] μελέτησαν αρκετές περιπτώσεις όπου εταιρίες εμπορικού λογισμικού συνεργάστηκαν επιτυχώς και δημιούργησαν καλές σχέσεις με κοινότητες εθελοντών προγραμματιστών.

Κεφάλαιο 3
Ποιότητα Λογισμικού

Η ποιότητα γενικότερα αλλά και η ποιότητα λογισμικού ειδικότερα είναι αδύνατο να οριστούν με ένα τρόπο που να μην είναι σχετικά αφηρημένος, με αποτέλεσμα να δημιουργούνται πολλά πρακτικά ζητήματα προς επίλυση. Η πρώτη ενότητα ξεκινάει με πέντε διαφορετικές οπτικές γωνίες που μπορεί να προσεγγιστεί η ποιότητα και στη συνέχεια γίνεται αναφορά στους πιο αντιπροσωπευτικούς ορισμούς. Η ανάγκη για τον επιμερισμό των συστατικών στοιχείων της ποιότητας σε μετρήσιμα μεγέθη μας οδηγεί στην δεύτερη ενότητα, δηλαδή στο να εξετάσουμε τα πιο σημαντικά πρότυπα και μοντέλα ποιότητας λογισμικού. Σκοπός αυτών των μοντέλων είναι να αποδομήσουν την έννοια της ποιότητας σε επιμέρους χαρακτηριστικά που να υπάρχει η δυνατότητα μέτρησής τους. Ειδική αναφορά γίνεται και για μοντέλα που έχουν αναπτυχθεί για το λογισμικό ανοικτού κώδικα. Έτσι έχουμε παρουσιάσει όλο το απαραίτητο γνωστικό υπόβαθρο για την τελευταία ενότητα του κεφαλαίου που διαπραγματεύεται τις μετρήσεις και τις μετρικές. Δίνονται οι ορισμοί της μέτρησης, της μετρικής και γίνεται η διάκριση μεταξύ εσωτερικών και εξωτερικών μετρικών. Γίνεται μια εκτενής αναφορά στις εσωτερικές μετρικές όπου έχουν χωριστεί σε δύο μεγάλες κατηγορίες. Η πρώτη αφορά τις λεγόμενες παραδοσιακές μετρικές λογισμικού που αναπτύχθηκαν κατά κύριο λόγο για τις διαδικαστικές γλώσσες προγραμματισμού. Η δεύτερη αφορά τις αντικειμενοστρεφείς μετρικές που όπως φανερώνει και το όνομά τους αφορούν αποκλειστικά τις αντικειμενοστρεφείς γλώσσες προγραμματισμού.

3.1 Ορισμοί για την Ποιότητα

Μέχρι το πρόσφατο παρελθόν όταν συνέβαινε ένα λάθος στο υπόλοιπο ενός τραπεζικού λογαριασμού ή το σύστημα ελέγχου των φαναριών κατέρρεε, οι άνθρωποι κατηγορούσαν γενικά τον "υπολογιστή", χωρίς να γίνεται καμία διάκριση μεταξύ του υλικού και του λογισμικού. Τα τελευταία χρόνια με την εξάπλωση της χρήσης του λογισμικού σχεδόν σε όλες τις πτυχές της ζωής μας, ο σύγχρονος άνθρωπος έχει αρχίσει να συνειδητοποιεί τη μεγάλη σημασία της ποιότητας ενός λογισμικού για την καθημερινότητά του. Πλέον η ποιότητα ενός λογισμικού είτε αφορά ένα πρόγραμμα στον υπολογιστή είτε οποιοδήποτε άλλο καταναλωτικό προϊόν (π.χ. αυτοκίνητο, κινητό τηλέφωνο κλπ) αποτελεί ένα βασικό κριτήριο που καθορίζει τις επιλογές του κάθε καταναλωτή. *Όμως τι ακριβώς εννοούμε όταν λέμε "Ποιότητα Λογισμικού";*

Υπάρχουν πολλές προσεγγίσεις στην έννοια της ποιότητας. Μία από τις πιο σημαντικές είναι αυτή του David Garvin που μελέτησε τον τρόπο με τον οποίο η ποιότητα εκλαμβάνεται σε διάφορους επιστημονικούς κλάδους όπως είναι τα οικονομικά, το μάρκετινγκ, η φιλοσοφία κλπ. Κατέληξε στο ότι η ποιότητα είναι μία σύνθετη και πολύπλευρη έννοια που μπορεί να περιγραφεί από πέντε διαφορετικές οπτικές γωνίες [113]:

1. **Αφηρημένη:** Η ποιότητα είναι κάτι που μπορεί να αναγνωριστεί αλλά δεν μπορεί να οριστεί. Αυτή η οπτική γωνία είναι κοντά στον ορισμό του "ιδεατού" από τον Πλάτωνα. ή της "μορφής" του Αριστοτέλη [114].

2. **Χρηστική:** Αφορά την ποιότητα από την πλευρά του χρήστη. Επικεντρώνεται στα χαρακτηριστικά που πρέπει να έχει το προϊόν για να ικανοποιήσει τις ανάγκες του χρήστη και έτσι συνδέεται στενά με τη χρηστικότητα του λογισμικού.

3. **Κατασκευαστική:** Ο κατασκευαστής του προϊόντος επικεντρώνεται στην ποιότητα του προϊόντος κατά την παραγωγή και μετά την αγορά από τον πελάτη. Ουσιαστικά επικεντρώνεται στο να είναι από την αρχή το προϊόν "όπως πρέπει" για να μην υπάρχουν κόστη για διορθώσεις στην παραγωγή ή στον πελάτη μετά την παράδοση.

4. **Προϊόντος:** Σε αντίθεση με τις προηγούμενες οπτικές γωνίες που θεωρούν τα εσωτερικά χαρακτηριστικά του προϊόντος έμφυτα, η οπτική του προϊόντος ασχολείται με την αξιολόγηση αυτών των χαρακτηριστικών. Η προσέγγιση αυτή έχει υιοθετηθεί από αρκετούς που αξιολογούν λογισμικό με χρήση εσωτερικών μετρικών, με τη λογική

ότι η μέτρηση και ο έλεγχος των εσωτερικών χαρακτηριστικών του λογισμικού μπορεί να οδηγήσει σε καλύτερη εμπειρία χρήσης. Οι έρευνες έχουν δείξει ότι ενώ οι κακές εσωτερικές μετρήσεις μπορεί να οδηγήσουν κατά κανόνα και σε κακές εξωτερικές μετρήσεις, το αντίθετο όμως δεν είναι καθόλου δεδομένο δηλαδή ένα λογισμικό που έχει ικανοποιητικές εσωτερικές μετρικές δεν είναι απαραίτητο ότι θα έχει και ικανοποιητικές εξωτερικές μετρικές [115].

5. **Αξίας:** Η ποιότητα είναι συνάρτηση του χρηματικού ποσού που είναι διατεθειμένος να πληρώσει ο χρήστης για την αγορά του προϊόντος. Η συνεισφορά αυτής της οπτικής έχει να κάνει με τη διαχείριση της εξισορρόπησης μεταξύ της ποιότητας του προϊόντος και του κόστους που απαιτείται για να επιτευχτεί αυτή. Πολλές φορές οι χρήστες είναι διατεθειμένοι να ανταλλάξουν λίγη μείωση στην ποιότητα με μία χαμηλότερη τιμή.

Αν και η παραπάνω ανάλυση βοηθά αρκετά στο να κατανοήσουμε την ποιότητα, δεν έχουμε απαντήσει πλήρως στο ερώτημα τι είναι η ποιότητα, αφού την περιγράψαμε μεν αλλά δεν την ορίσαμε. Υπάρχουν αρκετοί ορισμοί στην βιβλιογραφία που μπορεί να μελετήσει κανείς ώστε να αποκτήσει μία καλύτερη αντίληψη για το τι εννοούμε όταν λέμε ποιότητα:

- **ISO 8402:** Ποιότητα είναι το σύνολο των χαρακτηριστικών μιας οντότητας που τις αποδίδουν την ικανότητα να ικανοποιεί εκφρασμένες και συνεπαγόμενες ανάγκες [116].

- **American Society for Quality:** Ποιότητα είναι η συλλογή χαρακτηριστικών και ιδιοτήτων του προϊόντος που σχετίζονται με τη δυνατότητα του να εκπληρώνει τις ζητούμενες ανάγκες των πελατών [117].

- **Phillip Cosby:** Ποιότητα είναι η συμμόρφωση με τις απαιτήσεις των χρηστών [118].

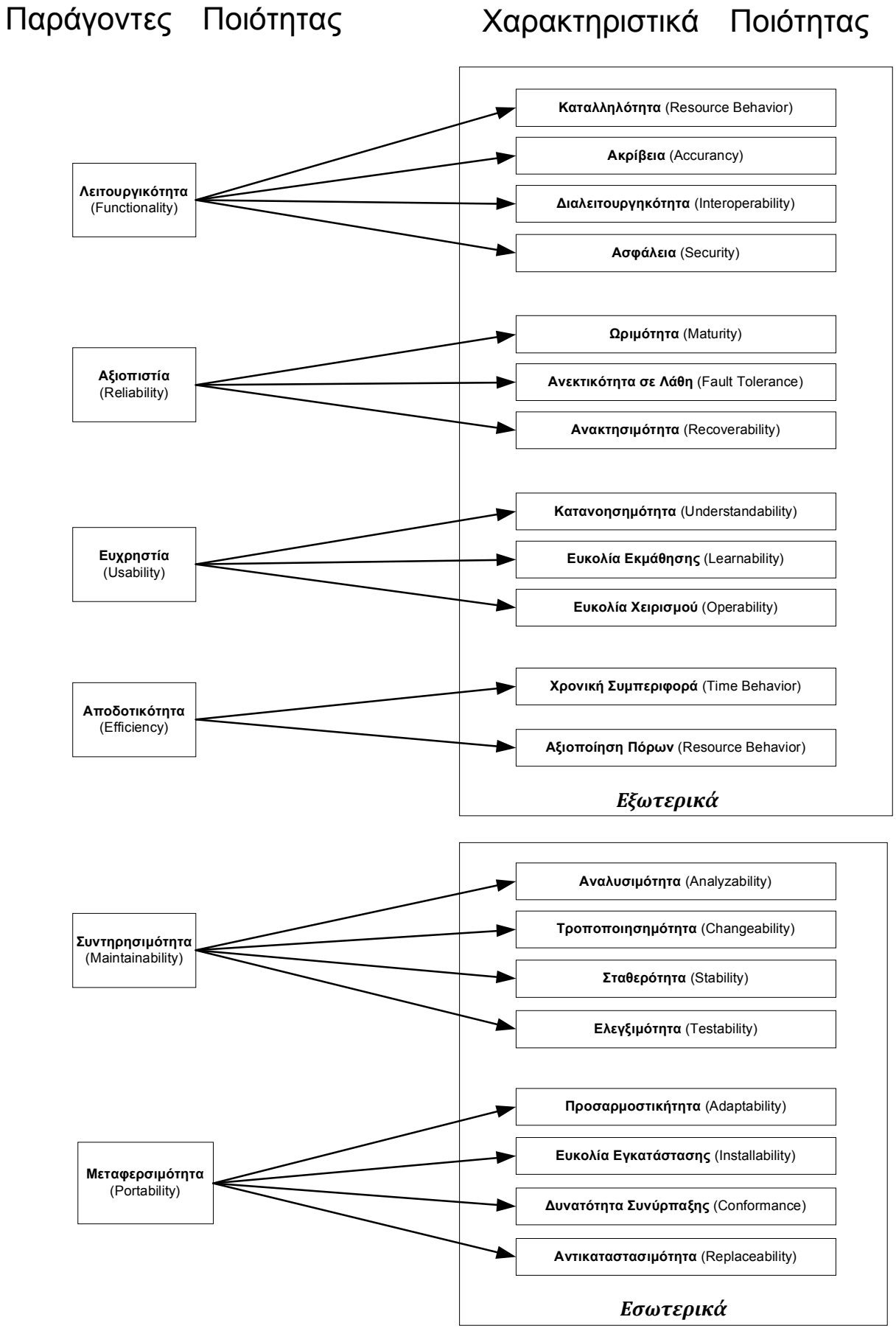
- **Joseph Juran:** Ποιότητα είναι η καταλληλότητα προς χρήση [119].

Παρατηρούμε ότι όλοι οι ορισμοί δεν είναι ιδιαίτερα συγκεκριμένοι οπότε και εναποτίθεται σε αυτόν που ενδιαφέρεται για την ποιότητα λογισμικού να τους εφαρμόσει πρακτικά σε ένα έργο λογισμικού. Προκύπτει δηλαδή η ανάγκη για τον επιμερισμό της ποιότητας σε επιμέρους μετρήσιμα χαρακτηριστικά ώστε να μπορεί να ελεγχθεί. Αυτό το κενό έρχονται να καλύψουν τα μοντέλα κα πρότυπα ποιότητας λογισμικού που παρουσιάζουμε στην επόμενη ενότητα. Δίνουμε έμφαση σε αυτά που αναφέρονται στην ποιότητα λογισμικού ανοικτού κώδικα.

3.2 Μοντέλα και Πρότυπα Ποιότητας Λογισμικού

Ο αφηρημένος τρόπος με τον οποίο ορίζεται η ποιότητα οδήγησε τους ερευνητές της ποιότητας λογισμικού να αναζητήσουν τρόπους με τους οποίους θα επιτύγχαναν την αντιστοίχιση του όρου ποιότητα με επιμέρους χαρακτηριστικά που θα μπορούσαν να μετρηθούν με κάποιο τρόπο. Αυτά επικράτησε να ονομάζονται παράγοντες ποιότητας (quality factors) και κάθε ένα από αυτά συνήθως περιλαμβάνει μία ομάδα από επιμέρους χαρακτηριστικά για τα οποία υπάρχει τρόπος να μετρηθούν. Βασικός σκοπός κάθε μοντέλου ποιότητας λογισμικού είναι να περιλαμβάνει παράγοντες ποιότητας που να έχουν μεταξύ τους τη χαμηλότερη δυνατή επικάλυψη και η συνολική σύνθεση τους να καλύπτει με ικανοποιητικό τρόπο όλες τις πλευρές της ποιότητας του λογισμικού. Το πρώτο μοντέλο ποιότητας λογισμικού που παρουσιάστηκε το 1977 από τον McCall [120] ο οποίος χώρισε τους παράγοντες ποιότητας σε κριτήρια, τα οποία στη συνέχεια θα μπορούσαν να μετρηθούν άμεσα με μετρικές. Συνολικά ορίστηκαν στο μοντέλο 11 παράγοντες ποιότητας, 25 κριτήρια και 41 μετρικές. Οι τιμές των μετρικών έπρεπε να προκύπτουν από απαντήσεις της μορφής "ναι" ή "όχι", σε ένα ερωτηματολόγιο που θα περιλάμβανε ερωτήσεις για όλα τα κριτήρια του μοντέλου. Εξακολουθεί να χρησιμοποιείται σε κάποιες επιχειρήσεις γιατί είναι αρκετά αναλυτικό και προσαρμοσμένο στην ομάδα υλοποίησης του λογισμικού.

Λίγο αργότερα το 1978 παρουσιάστηκε άλλο ένα μοντέλο ποιότητας λογισμικού από τον Boehm [121] που είχε παρόμοια ιεραρχική δομή με αυτό του McCall. Αντί για δύο επίπεδα το μοντέλο έχει τρία επίπεδα, όπου στο πρώτο επίπεδο είναι οι πρωταρχικές χρήσεις (primary uses), στο επόμενο οι ενδιάμεσες κατασκευές (intermediate constructs) και στο τελευταίο οι πρωτογενείς κατασκευές (primitive constructs). Αυτές οι πρωτογενείς κατασκευές μπορούν να μετρηθούν με μετρικές λογισμικού και δεν χρειάζεται να συμπληρωθεί κανένα ερωτηματολόγιο. Πολύ αργότερα το 1991 δημιουργήθηκε το πρότυπο ISO 9126 που αποτελεί ένα μοντέλο ποιότητας λογισμικού και το οποίο εξελίχθηκε σε διεθνές πρότυπο από τον οργανισμό ISO [122]. Έχει δανειστεί κάποια χαρακτηριστικά και από τα δύο προηγούμενα μοντέλα που αναφέραμε, αλλά χρησιμοποιεί διαφορετική ορολογία. Παρουσιάζεται στο σχήμα 3.1 και αποτελείται από έξι παράγοντες ποιότητας που αναλύονται σε επιμέρους χαρακτηριστικά ποιότητας. Σημαντικές διαφορές σε σχέση με τα άλλα δυο μοντέλα είναι ότι κάθε χαρακτηριστικό ανήκει αυστηρά μόνο σε ένα παράγοντα ποιότητας. Η οπτική του είναι κυρίως από την πλευρά του χρήστη αφού τα περισσότερα χαρακτηριστικά αφορούν εξωτερικά χαρακτηριστικά και δίνει πολύ μικρότερη βαρύτητα στην ομάδα ανάπτυξης. Τέλος, το γεγονός ότι δεν ορίζονται συγκεκριμένες μετρικές για την μέτρηση κάθε χαρακτηριστικού αλλά ότι πρέπει να οριστεί το συγκεκριμένο μοντέλο που θα ακολουθηθεί στην πράξη, οδήγησαν στην μη υιοθέτηση του από πολλούς οργανισμούς.



Σχήμα 3.1: Το Διεθνές Πρότυπο Ποιότητας Λογισμικού ISO 9126

Τα μοντέλα που έχουμε εξετάσει μέχρι τώρα αναφέρονται στις παραδοσιακές μεθόδους ανάπτυξης λογισμικού και όπως είδαμε στην ενότητα 2.2 υπάρχουν αρκετές διαφορές με τις μεθόδους ανάπτυξης λογισμικού ανοικτού κώδικα. Έτσι, λόγω των ιδιαιτεροτήτων που παρουσιάζει το λογισμικό ανοικτού κώδικα τα παραδοσιακά μοντέλα ποιότητας δεν είναι κατάλληλα για να χρησιμοποιηθούν από την κοινότητα ενός έργου λογισμικού. Προκύπτει λοιπόν ανάγκη για την τροποποίηση των υφισταμένων ή τη δημιουργία νέων μοντέλων ειδικά για τα έργα ανοικτού κώδικα. Παρά το γεγονός ότι έχουν γίνει αρκετές προσπάθειες προς αυτή την κατεύθυνση, δεν φαίνεται να υπάρχει κάποιο μοντέλο που να ξεχωρίζει ή που να καλύπτει όλες τις περιπτώσεις. Μία αντιπροσωπευτική αλλά όχι εξαντλητική λίστα είναι η παρακάτω:

- **Μοντέλο Ωριμότητας Ανοικτού Κώδικα** (Open Source Maturity Model) [123]: Η βασική υπόθεση που γίνεται σε αυτό το μοντέλο είναι ότι η ποιότητα ενός έργου λογισμικού είναι ανάλογη με την ωριμότητά του. Αυτή αναλύεται σε έξι συνιστώσες και κάθε μια από αυτές έχει ένα συγκεκριμένο βάρος. Για να εκτιμήσουμε ένα έργο ανοικτού κώδικα το μόνο που χρειάζεται είναι να βαθμολογήσουμε τις συνιστώσες και το τελικό αποτέλεσμα είναι το σταθμισμένο άθροισμα των βαθμολογιών. Αν και είναι απλό στην εφαρμογή, το μεγάλο του μειονέκτημα είναι ότι δεν λαμβάνει υπόψη τον πηγαίο κώδικα.

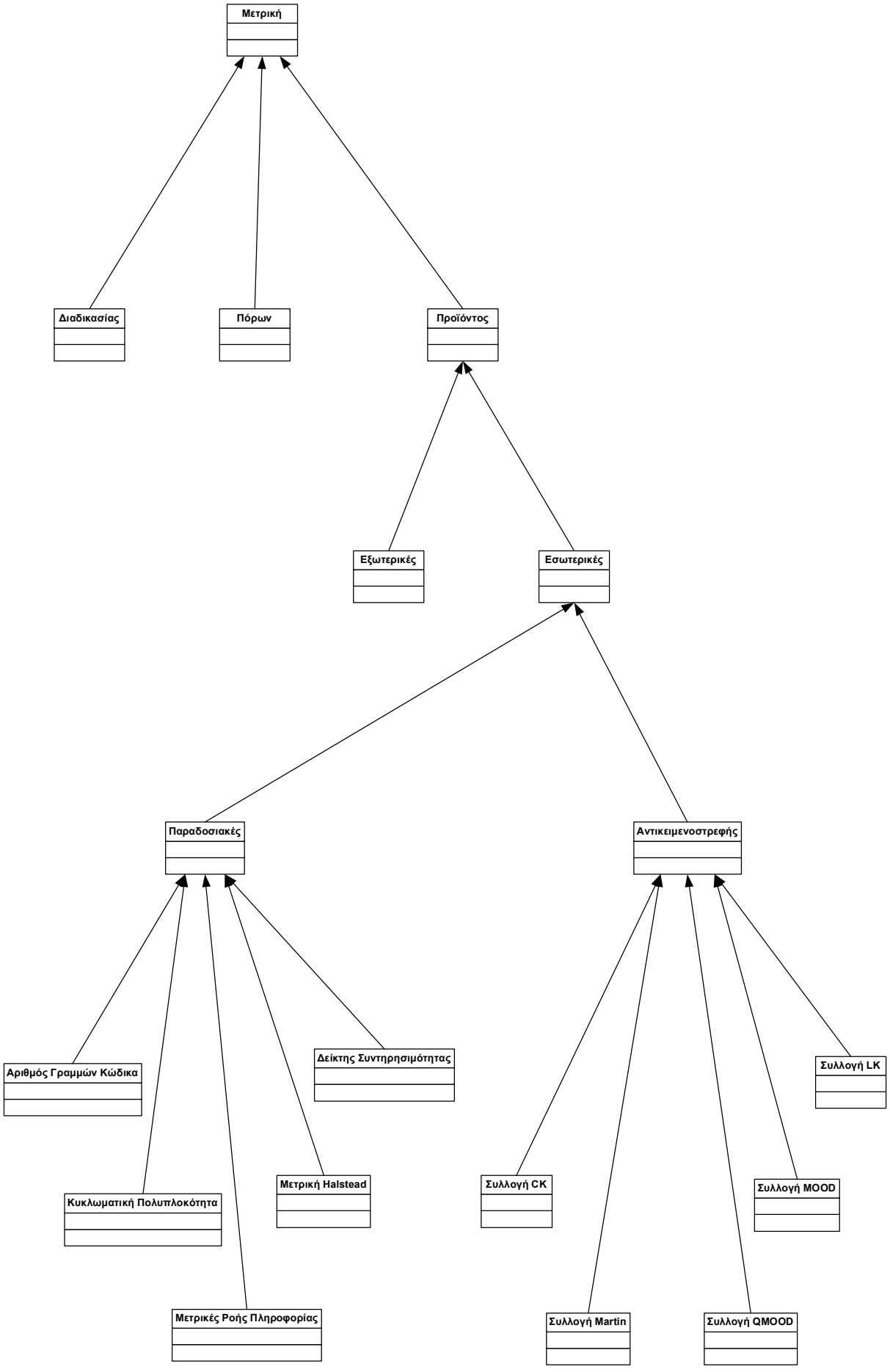
- **Κατάταξη Ετοιμότητας Ανοικτής Εργασίας** (Open Business Readiness Rating) [124]: Η διαδικασία της αξιολόγησης περιλαμβάνει αρχικά τον προσδιορισμό μιας εφαρμογής αναφοράς και μέσω αυτής μιας σειράς χαρακτηριστικών τα οποία θα πρέπει να διαθέτουν οι υπό αξιολόγηση εφαρμογές. Στην συνέχεια η αξιολόγηση γίνεται με την εκτίμηση της βαθμολογίας για κάθε ένα χαρακτηριστικό και η τελική βαθμολογία βγαίνει από το σταθμισμένο άθροισμα (με βάση τα βάρη που έχουν δοθεί σε κάθε χαρακτηριστικό) όλων των επιμέρους βαθμολογιών. Μειονέκτημα του είναι ότι η διαδικασία είναι υποκειμενική και δεν προσφέρεται για αυτοματοποίηση.

- **Αξιολόγηση και Επιλογή Λογισμικού Ανοικτού Κώδικα** (Qualification and Selection of Open Source Software) [125]: Είναι ένα επαναληπτικό μοντέλο τεσσάρων σταδίων. Το πρώτο αφορά τον καθορισμό των κριτηρίων αξιολόγησης. Το δεύτερο περιλαμβάνει την συλλογή συγκεκριμένων στοιχείων για την κοινότητα του υπό αξιολόγηση έργου. Το τρίτο τον ορισμό των κριτηρίων επιλογής με βάση της ανάγκες του χρήστη. Το τέταρτο είναι η εύρεση του λογισμικού που ικανοποιεί τους περιορισμούς και τις ανάγκες του χρήστη. Αν και υπάρχει λογισμικό για τη διευκόλυνση της διαδικασίας το μοντέλο δεν είναι ιδιαίτερα ευέλικτο και είναι σχετικά πολύπλοκη η εφαρμογή της διαδικασίας.

3.3 Μετρήσεις και Μετρικές Ποιότητας Λογισμικού

Ο αντικειμενικός σκοπός των μετρήσεων δεν είναι η μέτρηση των οντοτήτων αλλά η μέτρηση των χαρακτηριστικών των οντοτήτων. Έτσι, αφού δεν μπορούμε να μετρήσουμε άμεσα την ποιότητα λογισμικού μετράμε συγκεκριμένα χαρακτηριστικά της και από τις μετρήσεις βγάζουμε χρήσιμα συμπεράσματα για την ποιότητα. Αυτά τα χαρακτηριστικά περιγράφηκαν στην προηγούμενη ενότητα και σε αυτή θα δούμε συγκεκριμένες μετρικές για την ποιότητα λογισμικού. Σύμφωνα με τους Fenton και Pfleeger [145] μετρική είναι μία εμπειρική αντιστοίχιση ενός αριθμού ή συμβόλου σε μία οντότητα με στόχο να χαρακτηριστεί ένα συγκεκριμένο χαρακτηριστικό της οντότητας αυτής. Οι τιμές στις μετρικές αντιστοιχούνται μέσω των μετρήσεων. Μια διάκριση των μετρικών [146] είναι σε διαδικασία που αναφέρεται σε οποιαδήποτε διαδικασία σχετική με το λογισμικό, πόρων που αναφέρεται σε προσωπικό, αναλώσιμα κλπ και προϊόντος που αναφέρεται στο ίδιο το λογισμικό. Μάλιστα έχει επικρατήσει οι μετρικές προϊόντος να λέγονται και μετρικές λογισμικού. Αυτές χωρίζονται σε εξωτερικές και σε εσωτερικές. Οι εξωτερικές είναι αυτές που μετράνε πώς αλληλεπιδρά το λογισμικό με τους χρήστες αλλά και γενικότερα με το ευρύτερο περιβάλλον του (οπότε υπάρχει ένα θέμα υποκειμενικότητας της μέτρησης). Εσωτερικές είναι αυτές που μπορούν να μετρηθούν με άμεσο τρόπο από τα χαρακτηριστικά του λογισμικού. Στη συνέχεια θα ασχοληθούμε μόνο με εσωτερικές μετρικές όπου στο σχήμα 3.2 μοντελοποιούμε την παραπάνω ανάλυση με χρήση διαγράμματος σε UML.

Οι εσωτερικές μετρικές λογισμικού ξεκίνησαν να χρησιμοποιούνται από τις αρχές της δεκαετίας του '70 για λογισμικό που έχει γραφεί σε διαδικαστικές γλώσσες προγραμματισμού. Όμως από τα μέσα της δεκαετίας του '90 άρχισε να επικρατεί το αντικειμενοστρεφές παράδειγμα ανάπτυξης εφαρμογών επειδή πρόσφερε πολλές ευκολίες όπως είναι η γρήγορη ανάπτυξη με την αρχιτεκτονική που βασίζεται σε ανεξάρτητα συστατικά (components), η δυνατότητα επαναχρησιμοποίησης λειτουργιών που αυξάνουν την ποιότητα σχεδίασης, τα ολοκληρωμένα περιβάλλοντα ανάπτυξης (integrated development environment) κλπ. Αυτή η τάση δημιούργησε νέες προκλήσεις για όλους τους εμπλεκόμενους στη διαδικασία παραγωγής λογισμικού αφού οι "παραδοσιακές" μετρικές που δημιουργήθηκαν για το παραδοσιακό παράδειγμα ανάπτυξης δεν φαίνονταν να είναι ικανοποιητικές και για το αντικειμενοστρεφές παράδειγμα ανάπτυξης. Στην πρώτη υποενότητα θα ασχοληθούμε με τις παραδοσιακές μετρικές αφού είναι οι πρώτες που δημιουργήθηκαν και επιπλέον μπορούν να χρησιμοποιηθούν σε λογισμικό που είναι υλοποιημένο σε αντικειμενοστρεφή γλώσσα προγραμματισμού. Στη δεύτερη ενότητα γίνεται μία σύντομη παρουσίαση των πιο γνωστών αντικειμενοστρεφών συλλογών για μετρικές.



Σχήμα 3.2: Κατηγοριοποίηση των Μετρικών με Διάγραμμα UML

3.3.1 Παραδοσιακές Μετρικές

Με τον όρο "παραδοσιακές" μετρικές εννοούμε όλες τις μετρικές που χρησιμοποιήθηκαν τα πρώτα χρόνια και αφορούσαν ή είχαν νόημα κυρίως σε προγράμματα που είχαν υλοποιηθεί με διαδικαστικές γλώσσες προγραμματισμού. Αυτό δεν σημαίνει βέβαια ότι κάποιος δεν μπορεί να τις χρησιμοποιήσει σε προγράμματα που έχουν αναπτυχθεί με αντικειμενοστρεφείς γλώσσες.

3.3.1.1 Αριθμός Γραμμών Πηγαίου Κώδικα (LOC)

Η μέτρηση των γραμμών του πηγαίου κώδικα είναι μια απλή και αξιόπιστη μετρική και όλα τα εργαλεία μετρικών μπορούν να τον υπολογίσουν. Σημαντικό είναι να οριστούν εξ αρχής ποιοί κανόνες θα χρησιμοποιηθούν για την καταμέτρηση του αριθμού των γραμμών ώστε να είναι εύκολη η σύγκριση των αποτελεσμάτων που προκύπτουν κάθε φορά. Το ινστιτούτο τεχνολογίας λογισμού (SEI) του πανεπιστημίου Carnegie-Mellon έχει δημοσιεύσει ένα πλαίσιο [126] με το οποίο μπορεί να οριστεί μια λίστα ελέγχου με την οποία γίνεται σαφές τι μετράμε και τι δεν μετράμε κάθε φορά. Οι διάφορες μελέτες χρησιμοποιούν διαφορετικούς κανόνες για τη μέτρηση των γραμμών ανάλογα με τους σκοπούς που εξυπηρετεί μια μέτρηση. Ας δούμε μερικές χρήσιμες παραλλαγές αυτής της μετρικής:

- **Αριθμός Γραμμών Κώδικα** (Lines of Code - LOC): Είναι ο συνολικός αριθμός γραμμών του κώδικα. Μερικές φορές ως LOC αναφέρεται η μέτρηση που προκύπτει αφού αφαιρεθούν οι κενές γραμμές και οι γραμμές σχολίων.

- **Μη Σχολιασμένος Αριθμός Γραμμών** (Non Commented Lines of Code - NCLOC): Είναι ο αριθμός γραμμών του κώδικα που δεν περιλαμβάνει τα σχόλια και την τεκμηρίωση του.

- **Σχολιασμένος Αριθμός Γραμμών** (Commented Lines of Code - CLOC): Είναι ο αριθμός γραμμών του κώδικα που αφορούν αποκλειστικά σχόλια που γίνονται σε αυτών.

- **Εκτελέσιμες Εντολές** (Executable Statements - ES): Λαμβάνονται υπόψη μόνο οι εκτελέσιμες εντολές του πηγαίου κώδικα, δηλαδή εξαιρούνται οι διάφορες επικεφαλίδες, οι ορισμοί των διαφόρων δομών δεδομένων κ.τ.λ.

- **Παραδοτέος Αριθμός Εντολών** (Delivered Source Instructions - DSI): Ο αριθμός των εντολών που τελικά θα παραδοθούν στον πελάτη δηλαδή δεν περιλαμβάνει κώδικα που γράφτηκε για τις δοκιμές, τα αρχικά πρότυπα κ.τ.λ.

Αν και κάποιος θα περίμενε ότι όσο αυξάνεται το μέγεθος θα αυξάνεται και η πυκνότητα των λαθών που αυτός περιέχει, εμπειρικές μελέτες [003, 127] έχουν δείξει ότι η σχέση μεταξύ των λαθών και του μεγέθους του κώδικα είναι σχεδόν γραμμική. Με άλλα λόγια η πυκνότητα των λαθών αναμένουμε να είναι σχετικά σταθερή όσο αυξάνεται το μέγεθος του πηγαίου κώδικα. Άρα ερχόμαστε το ερώτημα που προκύπτει με βάση τα παραπάνω, αφού ο συνολικός αριθμός ελαττωμάτων είναι γραμμικός ως προς το συνολικό μέγεθος του κώδικα, υπάρχει κάποιο όριο για κάθε τμήμα (module) του; Μελέτες [002, 003] έχουν δείξει ότι η πυκνότητα των λαθών είναι μικρότερη όταν το τμήμα κάθε κώδικα είναι μεταξύ 200 και 750 γραμμών κώδικα και αυτό είναι ανεξάρτητο από τη γλώσσα που χρησιμοποιείται. Μάλιστα για να βρεθεί σε κάθε περίπτωση ο ιδανικός αριθμός προτείνεται μια μεθοδολογία [002] όπου αυτός μπορεί να βρεθεί με την ανάλυση των ιστορικών δεδομένων από παλιότερα έργα που έχουν περατωθεί.

3.3.1.2 Κυκλωματική Πολυπλοκότητα (Cyclomatic Complexity)

Είναι η πιο γνωστή μετρική για τον προσδιορισμό της πολυπλοκότητας ενός τμήματος κώδικα και προτάθηκε το 1976 από τον McCabe [063]. Ουσιαστικά πρόκειται για μια απαρίθμηση του αριθμού των σημείων απόφασης στο κομμάτι του κώδικα που μετράται. Το θεωρητικό υπόβαθρο της κυκλωματικής πολυπλοκότητας είναι ότι όσο περισσότερα μονοπάτια υπάρχουν σε ένα τμήμα κώδικα τόσο πιο πολύπλοκος είναι αυτός και τόσο πιο δύσκολα μπορεί να κατανοηθεί ή και να ελεγχθεί.

V(G)	Τύπος Διαδικασίας	Ρίσκο
1- 4	Απλή διαδικασία	Μικρό
5 - 6	Καλά δομημένη και σταθερή διαδικασία	Μικρό
11 - 20	Περισσότερο πολύπλοκη διαδικασία	Μέτριο
21 - 50	Μια πολύπλοκη διαδικασία	Υψηλό
50 -	Μια διαδικασία που είναι αδύνατον να ελεγθεί και πολύ δύσκολο να κατανοηθεί	Πολύ Υψηλό

Πίνακας 3.1: Συσχέτιση Κυκλωματικής Πολυπλοκότητας και Ρίσκου Διαδικασίας

Βασίζεται στη θεωρία γράφων όπου για ένα γράφο G μπορεί να υπολογιστεί η κυκλωματική του πολυπλοκότητα V(G) μετρώντας τον αριθμό των ανεξάρτητων διαδρομών μέσα στο πρόγραμμα μετρώντας έτσι τον ελάχιστο αριθμό ελέγχων που πρέπει να γίνουν προκειμένου να ελεγχθούν

όλες οι εντολές του προγράμματος. Ο υπολογισμός του V(G) μπορεί να γίνει με δυο ισοδύναμους τρόπους [128]:

- **V(G) = e - n + 2**: όπου e είναι ο αριθμός των ακμών του γράφου G και n είναι ο αριθμός των κόμβων του γράφου G.

- **V(G) = bd + 1**: όπου bd είναι ο αριθμός των δυαδικών αποφάσεων στον γράφο ελέγχου G. Στην περίπτωση που υπάρχουν n τρόποι απόφασης σε ένα κόμβο τότε αυτοί μετράνε στον τύπο σαν n -1 δυαδικές αποφάσεις.

Από τα παραπάνω είναι προφανές ότι όσο μεγαλύτερη είναι η κυκλωματική πολυπλοκότητα τόσο πιο δύσκολο είναι ο κώδικας να ελεγχθεί και να συντηρηθεί. Όμως, από ποια τιμή και μετά θα πρέπει να αρχίσουμε να ανησυχούμε για ένα τμήμα κώδικα; Στον πίνακα 3.1 παρουσιάζουμε μια σύνοψη των τιμών της μετρικής που φαίνεται να συμφωνούν οι περισσότερες μελέτες [129] και ο αριθμός 10 είναι ένα καλό όριο για μια διαδικασία. Όμως σε αυτό δεν συμφωνούν όλοι π.χ. η NASA προτείνει ότι σαν ιδανικό μέγιστο αριθμό κυκλωματικής πολυπλοκότητας αντί του 10 το 20 [001]. Σε κάθε περίπτωση θα πρέπει να έχουμε υπόψη μας ότι αριθμός μεγαλύτερος του 20 στην κυκλωματική πολυπλοκότητα θα πρέπει να μας ανησυχήσει, ενώ αριθμός μεγαλύτερος του 50 μπορεί να είναι πηγή μεγάλων προβλημάτων.

3.3.1.3 Μετρικές Halstead (Halstead's Metrics)

Το 1977 ο Maurice Halstead παρουσίασε μια σειρά μετρικών που είχαν ως σκοπό να καθορίσουν ένα ποσοτικό μέτρο της πολυπλοκότητας που να βασίζεται στους τελεστές και τα έντελα του πηγαίου κώδικα [061]. Ορίστηκαν εξισώσεις για τη δυσκολία, την προσπάθεια, τον όγκο κ.α. του προγράμματος που ονομάστηκαν η "επιστήμη του λογισμικού" και δίνονται από τους παρακάτω τύπους [062]:

- **Μήκος Προγράμματος** (Program Length): Είναι το σύνολο όλων των τελεστών και εντέλων του προγράμματος N = N1 + N2

- **Μέγεθος Λεξιλογίου** (Vocabulary Size): Είναι το σύνολο των μοναδικών τελεστών και εντέλων του προγράμματος n = n1 + n2

- **Όγκος Προγράμματος** (Program Volume): Είναι η πληροφορία του προγράμματος μετρημένη σε bits και είναι ίσος με V = N * log2(n)

- **Επίπεδο Δυσκολίας** (Difficulty Level): Είναι μια αναλογία μεταξύ του συνολικού αριθμού των τελεστών στο πρόγραμμα και του αριθμού των μοναδικών τελεστών D = (n1/2) * (N2/n2)

- **Επίπεδο Προγράμματος** (Program Level): Είναι το αντίστροφο της ροπής του προγράμματος για σφάλματα και δίνεται από τον τύπο L = 1 / D

- **Προσπάθεια Υλοποίησης** (Effort to Implement): Είναι μια συνάρτηση του όγκου του προγράμματος και της δυσκολίας του, οπότε δίνεται από τον τύπο E = V * D

- **Χρόνος Υλοποίησης** (Time to Implement): Είναι αναλογικός της απαιτούμενης προσπάθειας και έχουν γίνει αρκετές μελέτες για τον εμπειρικό καθορισμό του. Ο Halstead πρότεινε την διαίρεση της προσπάθειας υλοποίησης δια 18 για να μας δώσει τον χρόνο υλοποίησης σε δευτερόλεπτα T = E/18

- **Αριθμός Παραδομένων Λαθών** (Number of Delivered Bugs): Συσχετίζεται η συνολική πολυπλοκότητα του κώδικα με τον αριθμό λαθών από τον τύπο B = (E ^ 2/3) / 3000.

Όπου στους παραπάνω τύπους είναι:

n1 = ο αριθμός των διακριτών τελεστών του προγράμματος

n2 = ο αριθμός των διακριτών εντέλων του προγράμματος

N1 = ο συνολικός αριθμός των εμφανίσεων τελεστών στο πρόγραμμα

N2 = ο συνολικός αριθμός των εμφανίσεων των εντέλων στο πρόγραμμα

Στο σημείο αυτό θα πρέπει να τονίσουμε ότι ο τρόπος ορισμού των τελεστών και των εντέλων είναι μεγάλης σημασίας για την εφαρμογή της μεθόδου. Βέβαια αυτοί οι κανόνες δεν είναι μοναδικοί και πιο μεγάλη σημασία έχει να οριστεί ο τρόπος μέτρησης τους και αυτός να είναι συνεπής. Μιας και οι μετρικές αυτές υπολογίζονται αφού ο πηγαίος κώδικας έχει γραφεί μπορεί να μην μας δίνουν κάποια πρόβλεψη για την προσπάθεια αλλά είναι πολύ καλοί για την πρόβλεψη της προσπάθειας που θα χρειαστεί για την συντήρηση του πηγαίου κώδικα. Σε γενικές γραμμές δεν έχει αποδεδειχθεί ως ένα καλύτερο μέτρο πρόβλεψης από το αριθμό γραμμών του κώδικα που είναι πιο απλός [128],όμως συμπεριλάβαμε τη συγκεκριμένη μετρική λόγω της μεγάλης σημασίας της και της μεγάλης διάδοσης που έχει γνωρίσει.

3.3.1.4 Μετρικές Ροής Πληροφορίας (Information Flow Metrics)

Αυτές οι μετρικές έχουν σαν σκοπό να μετρήσουν την ροή της πληροφορίας από και προς τα διάφορα τμήματα κώδικα (modules). Θεωρητικά τουλάχιστον μια μεγάλη ροή πληροφορίας δείχνει μια έλλειψη συνοχής στη σχεδίαση και προκαλεί μεγαλύτερη πολυπλοκότητα. Αρχικά προτάθηκαν από τους Henry και Kafura [064] και έχουν εξελιχθεί στο πρότυπο IEEE 982.2 [130]. Χρησιμοποιούν ένα συνδυασμό των εισροών πληροφορίας (FanIn), εκροών πληροφορίας (FanOut) και του μήκους (Length) για να υπολογίσουν έναν αριθμό πολυπλοκότητας για μια διαδικασία. Αντίστοιχα, ορίζεται η πολυπλοκότητα ροής πληροφορίας (Information Flow Complexity - IFC) για ένα τμήμα κώδικα ως εξής:

$$IFC = (FanIN * FanOut) ^ 2 \quad (3.1)$$

Όπου

FanIN = ο αριθμός των τοπικών ροών σε ένα τμήμα κώδικα + ο αριθμός των δομών δεδομένων που χρησιμοποιούνται ως είσοδοι

FanOut = ο αριθμός των τοπικών εκροών σε ένα τμήμα κώδικα + ο αριθμός των δομών δεδομένων που χρησιμοποιούνται ως έξοδοι

Το πότε έχουμε ροή και πότε εκροή στο πηγαίο κώδικα ορίζεται αυστηρά στο πρότυπο IEEE 982.2 και δεν τα επαναλαμβάνουμε εδώ. Ένας μεγάλος δείκτης πολυπλοκότητας πληροφορίας σε μια διαδικασία μπορεί να σημαίνει ένα ή περισσότερα από τα παρακάτω:

- Περισσότερες από μια συναρτήσεις (Έλλειψη συνοχής).
- Ένα πιθανό σημείο κακής απόδοσης του συστήματος (Ένας πολύ μεγάλος όγκος πληροφορίας που κινείται).
- Μεγάλη πολυπλοκότητα μιας συνάρτησης (Έλλειψη συνοχής).
- Ένα καλό υποψήφιο για αναδόμηση (refactoring) ή απλοποίηση.
- Ένα καλό υποψήφιο για εκτεταμένες δοκιμές καλής λειτουργίας.

Μελέτες έχουν δείξει ότι η ροή πληροφοριών μπορεί να είναι ένας καλός δείκτης για την πρόβλεψη προβληματικών ή πολύπλοκων τμημάτων κώδικα [131] μεν αλλά το FanIN δεν φαίνεται να έχει συσχέτιση με την πρόβλεψη λαθών ή με την υποκειμενική άποψη ενός ειδικού σχετικά με την πολυπλοκότητα μιας διαδικασίας. Αυτό είναι λογικό γιατί ένα υψηλό FanIN μπορεί να προέρχεται από την συχνή κλήση κάποιων ρουτινών που απλά είναι δείγμα μιας καλής σχεδίασης και επαναχρησιμοποίησης. Θα παραμείνουμε στον ορισμό του FanIN που προτείνει το IEEE 982.2 αλλά θα πρέπει να έχουμε υπόψη μας ότι για τον ορισμό αυτών των μετρικών το FanIN στην μετρική IFC μπορούν να ορίζεται μόνο από τις δομές δεδομένων που διαβάζονται και να αφαιρεθεί ο αριθμών των ρουτινών που καλούνται.

3.3.1.5 Δείκτης Συντηρησιμότητας (Maintainability Index)

Ο δείκτης Συντηρησιμότητας (MI) είναι μία τιμή για την εκτίμηση της σχετικής συντηρησιμότητας του πηγαίου κώδικα. Η δημιουργία αυτού του σύνθετου δείκτη που βασίζεται στις μετρικές του αριθμού γραμμών πηγαίου κώδικα, τις μετρικές Halstead και την κυκλωματική πολυπλοκότητα έγινε το 1991 από τους Oman και Hargemeister [132]. Κατά την διάρκεια της δεκαετίας του '90 έγινε μεγάλη πρόοδος στη βελτίωση του από την σύμπραξη του πανεπιστημίου του Idaho και της βιομηχανίας (π.χ. HP) και χρησιμοποιήθηκε σε μεγάλα συστήματα του στρατού και της βιομηχανίας για παραπάνω από δέκα χρόνια [133]. Σκοπός ήταν να οριστεί μια μετρική που ο προγραμματιστής θα μπορούσε να χρησιμοποιήσει καθώς συντηρούσε τον πηγαίο κώδικα ώστε να μπορεί να ελέγξει αν οι αλλαγές που κάνει έχουν ως αποτέλεσμα τη βελτίωση ή όχι της συντηρησιμότητας του. Υπάρχουν δύο εκδόσεις του όπου στην μια συμπεριλαμβάνονται τα σχόλια (MI) και στη δεύτερη δεν συμπεριλαμβάνονται (MIwoc). Για την ακρίβεια έχουμε τις παρακάτω τρεις μετρικές [134]:

- **MIwoc**: Δείκτης Συντηρησιμότητας χωρίς σχόλια
- **MIcw**: Βάρος Δείκτη Συντηρησιμότητας σχολίων
- **MI**: Δείκτης Συντηρησιμότητας

Που δίνονται από τους παρακάτω τύπους:

$$MIwoc = 171 - 5.2 * \ln(aveV) - 0.23 * aveG - 16.2 * \ln(aveLOC) \quad (3.2)$$

$$MIcw = 50 * \sin(\sqrt{2.4 * perCM}) \quad (3.3)$$

$$MI = MIwoc + MIcw \quad (3.4)$$

Όπου:

aveV = μέσος Halstead όγκος V ανά τμήμα κώδικα

aveG = μέση επεκταμένη κυκλωματική πολυπλοκότητα ανά τμήμα κώδικα

aveLOC = μέσος αριθμός γραμμών ανά τμήμα κώδικα

perCM = μέσο ποσοστό γραμμών σχολίων ανά τμήμα κώδικα

Δείκτης MI	Συντηρησιμότητα
> 85	Πολύ καλό επίπεδο Συντηρησιμότητας του πηγαίου κώδικα
65 - 85	Μέτριο επίπεδο συντηρησιμότητας του πηγαίου κώδια που όμως δεν έχει ιδιαίτερα προβλήματα
< 65	Όχι καλό επίπεδο Συντηρησιμότητα. Μάλιστα σε πολύ κακό κώδικα μπορούν να προκύψουν και αρνητικές τιμές.

Πίνακας 3.2: Ερμηνεία Τιμών Δείκτη Συντηρησιμότητας

Σχετικά με τις τιμές του δείκτη συντήρησιμότητας έχουν επικρατήσει οι τιμές που παρουσιάζονται στον πίνακα 3.2 [135]. Οι Welker και Oman προτείνουν τρεις πολύ πρακτικούς κανόνες με βάση τους οποίους μπορεί κάποιος να διαλέγει ποιον από τους δύο παραπάνω τύπους πρέπει να χρησιμοποιήσει. Παραθέτουν τρία κριτήρια, και αν ένα από τα τρία αληθεύει τότε συνιστάται η χρήση του τύπου με τις τρεις μετρικές, αλλιώς είναι απαραίτητο να χρησιμοποιηθεί ο τύπος με τις τέσσερις [136]:

1. Τα σχόλια δεν ταιριάζουν ακριβώς με τον κώδικα που σχολιάζουν. Ακόμα και αν τα σχόλια παίζουν σημαντικό ρόλο για τους προγραμματιστές, όντας μη συγχρονισμένα προκαλούν προβλήματα και δυσκολεύουν τη συντήρηση του κώδικα.

2. Η ύπαρξη μεγάλων μπλοκ τυποποιημένων σχολίων, όπως αυτά που εισάγουν συστήματα διαχείρισης πηγαίου κώδικα των εταιρειών.

3. Η ύπαρξη κομματιών κώδικα, τα οποία, αν και περιέχουν εκτελέσιμες εντολές, δεν αποτελούν γραμμές κώδικα αλλά απλά έχουν σημειωθεί ως σχόλια.

3.3.2 Αντικειμενοστρεφείς Μετρικές

Οι μετρικές που παρουσιάζονται στη συνέχεια αφορούν αποκλειστικά λογισμικό που έχει αναπτυχθεί με αντικειμενοστρεφείς γλώσσες προγραμματισμού. Έχει γίνει επιλογή των πιο γνωστών συλλογών για μετρικές για όλα τα δυνατά επίπεδα μέτρησης δηλαδή αυτό της κλάσης, του πακέτου ή ολόκληρου του συστήματος.

3.3.2.1 Συλλογή Μετρικών CK (Chidamber and Kemerer Suite)

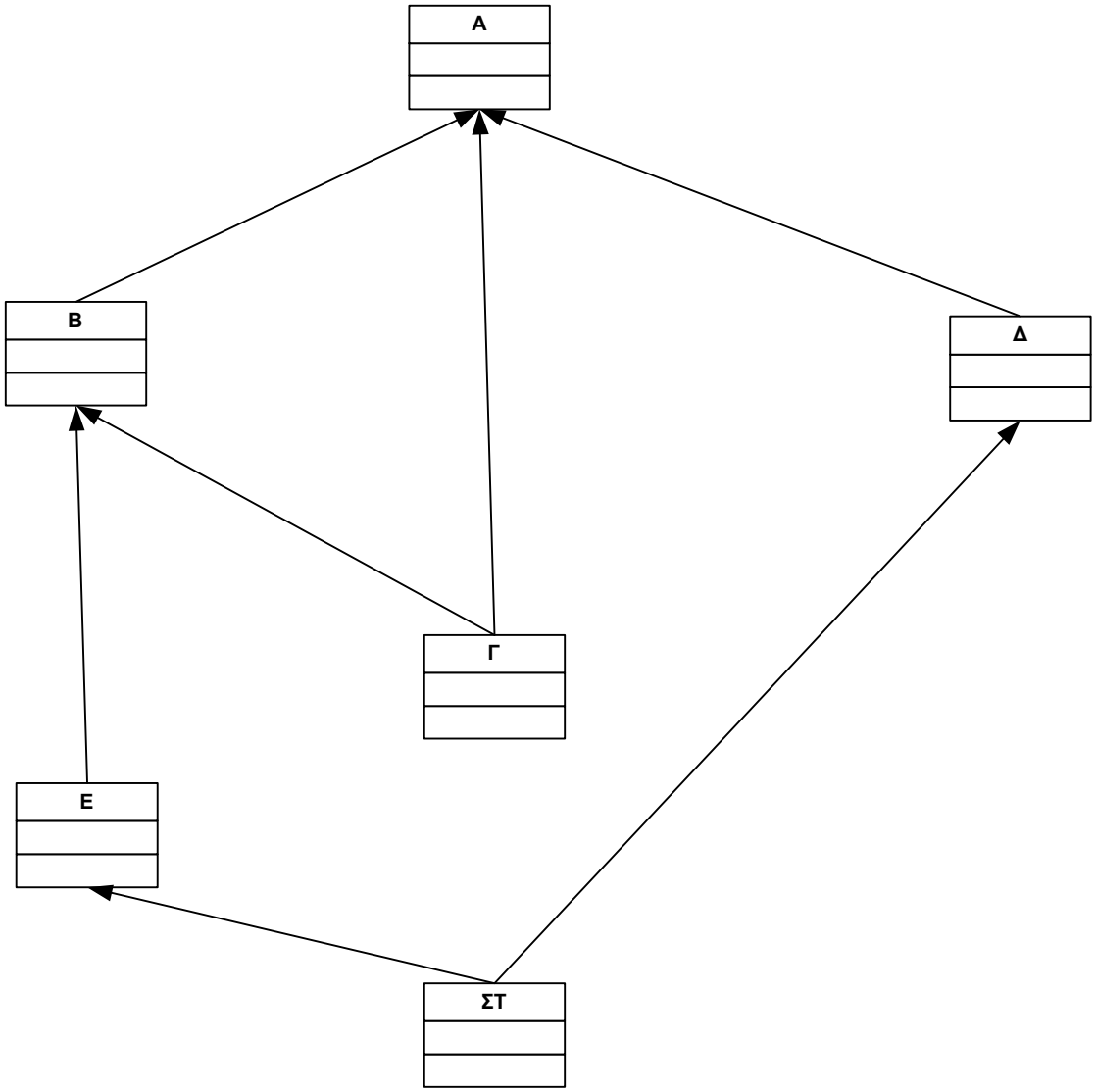
Είναι η πιο γνωστή συλλογή αντικειμενοστρεφών μετρικών και έχει χρησιμοποιηθεί σε εκατοντάδες έρευνες. Αποτελείται από τις παρακάτω έξι μετρικές [039]:

Σταθμισμένες Μέθοδοι ανά Κλάση (Weighted Methods per Class - WMC): Ισούται με το άθροισμα της πολυπλοκότητας κάθε μεθόδου της κλάσης. Δηλαδή αν η κλάση C έχει $m_1, m_2, \ldots, m_n$ μεθόδους και αντίστοιχα η πολυπλοκότητα κάθε μεθόδου είναι $c_1, c_2, \ldots, c_n$ τότε:

$$WMC_C = \sum_{i=1}^{n} c_i \quad (3.5)$$

Δεν καθορίζεται πώς υπολογίζεται η πολυπλοκότητα κάθε μεθόδου. Αυτή μπορεί να είναι η κυκλωματική πολυπλοκότητα ή λόγω της κληρονομικότητας επειδή είναι δύσκολο να αποτιμηθεί η κυκλωματική πολυπλοκότητα πολλοί ερευνητές για κάθε μέθοδο της κλάσης ορίζουν την πολυπλοκότητα της ίση με την μονάδα. Ο αριθμός των μεθόδων και η πολυπλοκότητά τους είναι ένας λογικός δείκτης της προσπάθειας που απαιτείται για την υλοποίηση και τον έλεγχο μιας κλάσης. Επίσης, κλάσεις με μεγάλο αριθμό μεθόδων είναι πιθανό να αφορούν περισσότερο συγκεκριμένη λειτουργικότητα της εφαρμογής και έτσι να μειώνεται η πιθανότητα για την επαναχρησιμοποίηση τους.

Βάθος Δέντρου Κληρονομικότητας (Depth of Inheritance Tree - DIT): Ισούται με το μέγιστο μονοπάτι από τη ρίζα του δέντρου κληρονομικότητας προς τον κόμβο που αναπαριστά την κλάση. Όσο αυξάνει το βάθος του δένδρου, οι κλάσεις χαμηλών επιπέδων κληρονομούν περισσότερες μεθόδους, γεγονός που αυξάνει την πολυπλοκότητα τους και περιορίζει τη δυνατότητα πρόβλεψης της συμπεριφοράς τους. Στο σχήμα 3.3 παρουσιάζουμε ένα ενδεικτικό δέντρο κληρονομικότητας ενός υποθετικού προγράμματος, όπου το βάθος του δέντρου κληρονομικότητας για την κλάση Γ είναι δυο και για την κλάση ΣΤ είναι τρία.



Σχήμα 3.3: Ενδεικτικό Δέντρο Κληρονομικότητας όπου Γίνεται Χρήση Πολλαπλής Κληρονομικότητας

Αριθμός Απογόνων (Number of Children - NOC): Ισούται με τον αριθμό των άμεσων απογόνων μιας κλάσης δηλαδή των "παιδιών" της. Όσο αυξάνει ο αριθμός των απογόνων αυξάνει ο βαθμός επαναχρησιμοποίησης, αλλά ταυτόχρονα αυξάνεται η προσπάθεια που απαιτείται για τον έλεγχο των μεθόδων της κλάσης αφού αυτή έχει μεγάλη σημασία για το σύστημα. Στο σχήμα 3.3 ο αριθμός απογόνων για την κλάση Α είναι τρία, για την κλάση Δ ένα και για την κλάση ΣΤ μηδέν.

Σύζευξη μεταξύ Κλάσεων (Coupling Between Objects - CBO): Ισούται με τον αριθμό των κλάσεων που παρέχουν σε μία κλάση τις απαραίτητες πληροφορίες (μπορεί να είναι δεδομένα, λειτουργίες κλπ) ώστε να ολοκληρώνονται οι μέθοδοί της. Σχετίζεται με την δυνατότητα επαναχρησιμοποίησης της κλάσης, τροποποίησης και ελέγχου της. Η απαίτηση να είναι η σύζευξη μεταξύ κλάσεων χαμηλή είναι παρόμοια με την απαίτηση για χαμηλή σύζευξη στα παραδοσιακά προγράμματα των διαδικαστικών γλωσσών προγραμματισμού.

Απόκριση για μια Κλάση (Responce for a Class - RFC): Ισούται με τον αριθμό των μεθόδων που μπορούν να εκτελεστούν ως αποτέλεσμα – απάντηση ενός μηνύματος, που λήφθηκε από κάποιο άλλο αντικείμενο της ίδιας κλάσης. Η τιμή της ισούται με το άθροισμα των τοπικών μεθόδων μιας κλάσης και τον αριθμό των μεθόδων που καλούνται από τις τοπικές μεθόδους:

$$RFC = \left| \{M\} \cup_{\forall i} \{R_i\} \right| \quad (3.6)$$

Όπου $\{M\}$ είναι το σύνολο όλων των μεθόδων της κλάσης και $\{R_i\}$ είναι το σύνολο των μεθόδων που καλούνται από την κλάση i. Η μετρική αυτή είναι ενδεικτική της πολυπλοκότητας της κλάσης και του βαθμού συσχέτισής της με άλλες κλάσεις. Από πειραματικά δεδομένα έχει προκύψει ότι όσο μεγαλύτερη είναι η τιμή της τόσο υψηλότερη είναι η πιθανότητα η εν λόγω κλάση να παρουσιάσει σφάλματα.. Επίσης, μεγάλη τιμή της συνεπάγεται ότι η κλάση γίνεται πιο πολύπλοκη και έτσι απαιτείται περισσότερος χρόνος για την κατανόηση ή τον έλεγχο της.

Έλλειψη Συνεκτικότητας των Μεθόδων (Lack of Cohesion Metric - LCOM): Ισούται με τον αριθμό των μεθόδων που πραγματοποιούν προσπελάσεις σε ένα ή περισσότερα κοινά μέλη δεδομένων. Δυο μέθοδοι λέμε ότι είναι συνεκτικές αν τα σύνολα των μελών δεδομένων που χρησιμοποιούν έχουν κοινά στοιχεία. Πιο τυπικά, έστω η κλάση C έχει $m_1, m_2, ..., m_n$ μεθόδους, τότε αν ορίσουμε ως $\{I_i\}$ το σύνολο όλων των μεταβλητών που χρησιμοποιούνται από την μέθοδο m_i υπάρχουν n τέτοια σύνολα $\{I_1\}, \{I_2\}, .., \{I_n\}$. Έστω $P = \left\{ \left(I_i, I_j \right) | \, I_i \cap I_j = \varnothing \right\}$ και $Q = \left\{ \left(I_i, I_j \right) | \, I_i \cap I_j \neq \varnothing \right\}$. Επιπλέον αν όλα τα n σύνολα $\{I_1\}, \{I_2\}, .., \{I_n\}$ είναι $\varnothing$ ορίζουμε ότι είναι $P = \varnothing$. Τότε η έλλειψη συνεκτικότητας ορίζεται ως εξής:

$$LCOM = \begin{cases} |P| - |Q|, \text{ αν } |P| > |Q| \\ 0, \text{ αλλιώς} \end{cases} \quad (3.7)$$

Η μετρική αυτή δείχνει την σχέση που υπάρχει ανάμεσα στις τοπικές μεθόδους μιας κλάσεις και στις εμφανίσεις των τοπικών ιδιοτήτων της κλάσης. Όσο περισσότερες είναι οι συνεκτικές μέθοδοι, τόσο μεγαλύτερη είναι η συνεκτικότητα της κλάσης και τόσο χαμηλότερη είναι η τιμή της μετρικής LCOM, γεγονός που ενδεχομένως να αυξάνει την πολυπλοκότητα της. Στην περίπτωση που η τιμή της LCOM είναι υψηλή, η συνεκτικότητα είναι χαμηλή και ενδεχομένως η κλάση να μπορεί να σχεδιαστεί καλύτερα με διάσπαση της σε δύο ή ακόμη και περισσότερες ανεξάρτητες νέες κλάσεις.

3.3.2.2 Συλλογή Μετρικών Martin (Martin Suite)

Αποτελείται από πέντε δημοφιλείς και πολύ γνωστές μετρικές επιπέδου πακέτου [137]:

Αριθμός Κλάσεων (Number of Classes - NC): Ορίζεται ως ο αριθμός των κανονικών και των αφηρημένων (abstract) κλάσεων του πακέτου. Είναι ένα μέτρο για το μέγεθος του πακέτου.

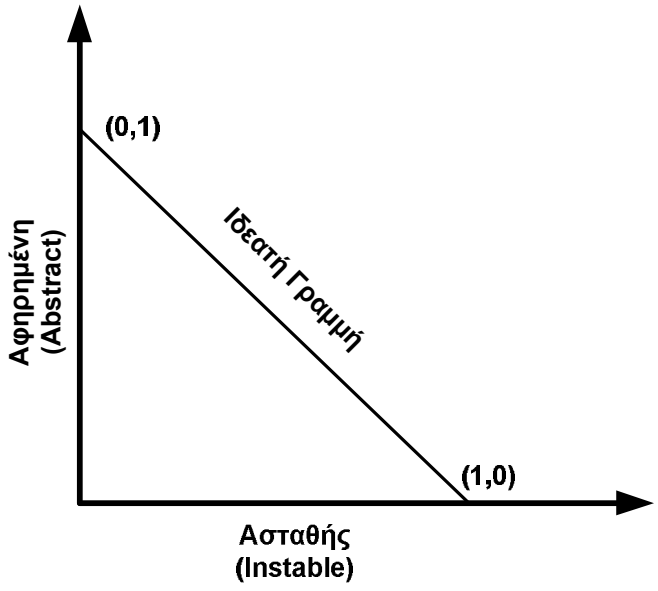
Κεντρομόλος Σύζευξη (Afferent Coupling- Ca): Ορίζεται ως ο αριθμός των άλλων πακέτων που εξαρτώνται από τις κλάσεις που βρίσκονται μέσα στο πακέτο. Μετράει τις εισερχόμενες εξαρτήσεις του πακέτου (Fan In).

Φυγόκεντρη Σύζευξη (Efferent Coupling - Ce): Ορίζεται ως ο αριθμός των άλλων πακέτων από τα οποία εξαρτώνται οι κλάσεις που βρίσκονται μέσα στο πακέτο. Μετράει τις εξερχόμενες εξαρτήσεις του πακέτου (Fan Out).

Αστάθεια (Instability - I): Ορίζεται ως η αναλογία της φυγόκεντρης σύζευξης (Ce) προς την συνολική σύζευξη (Ce + Ca) για το πακέτο, δηλαδή είναι:

$$I = \frac{Ce}{Ce + Ca} \quad (3.8)$$

Η μετρική αυτή είναι ένας δείκτης για την ανθεκτικότητα του πακέτου στις αλλαγές. Οι τιμές που παίρνει είναι από μηδέν δηλαδή τελείως σταθερό πακέτο έως ένα δηλαδή τελείως ασταθές.

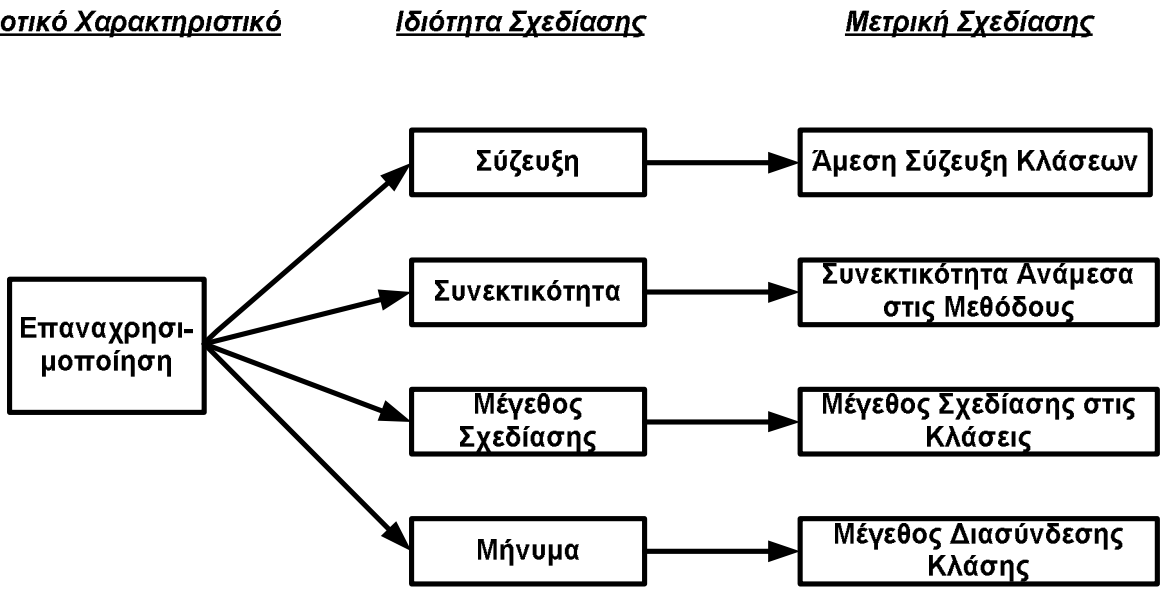


Σχήμα 3.4: Γραφική Αναπαράσταση της Μετρικής Απόσταση του Martin

Απόσταση (Distance - D): Ορίζεται ως η απόσταση του πακέτου από την ιδεατή γραμμή A+I=1. Αυτή η μετρική είναι ένας δείκτης της ισορροπίας του πακέτου ανάμεσα στο πόσο αφηρημένο είναι και στο πόσο σταθερό είναι. Ένα πακέτο που βρίσκεται στην ιδεατή γραμμή έχει την ιδανική αναλογία μεταξύ στο πόσο αφηρημένο και στο πόσο σταθερό είναι. Οι τιμές που παίρνει είναι από μηδέν όταν βρίσκεται πάνω στην ιδεατή γραμμή έως ένα όταν είναι όσο το δυνατόν πιο μακριά γίνεται.

3.3.2.3 Συλλογή Μετρικών QMOOD (Quality Model for Object Oriented Design)

Είναι ένα μοντέλο ποιότητας που ορίζει μια συγκεκριμένη σχέση μεταξύ των ποιοτικών χαρακτηριστικών της αντικειμενοτρεφής ανάλυσης και σχεδίασης (object oriented analysis and design) όπως είναι η κατανοησιμότητα και η επαναχρησιμοποίηση, και τα συσχετίζει με δομικά χαρακτηριστικά της αντικειμενοστρεφής ανάλυσης και σχεδίασης όπως είναι η ενθυλάκωση και η σύζευξη. Επίσης, παρουσιάζει και τις μετρικές που απαιτούνται για την ποσοτικοποίηση αυτών των χαρακτηριστικών. Αποτελείται από έξι εξισώσεις που ορίζουν την σχέση μεταξύ των έξι ποιοτικών χαρακτηριστικών και των έντεκα ιδιοτήτων σχεδίασης που χρησιμοποιεί το μοντέλο. Κάθε μια ιδιότητα συνδέεται με μια συγκεκριμένη μετρική που ορίζεται επίσης στο μοντέλο QMOOD. Μια αναλυτική παρουσίαση όλων των σχέσεων του μοντέλου ξεφεύγει από τα πλαίσια της διπλωματικής διατριβής και μπορεί να βρεθεί στο αρχικό άρθρο των Bansiya και Davis [035]. Στο σχήμα 3.5 παρουσιάζουμε τις σχέσεις που ορίζονται στο μοντέλο QMOOD για το ποιοτικό χαρακτηριστικό επαναχρησιμοποίηση ώστε τελικά να φτάσουμε στις αντίστοιχες μετρικές.



Σχήμα 3.5: Γραφική Αναπαράσταση των Σχέσεων για το Χαρακτηριστικό Επαναχρησιμοποίηση

Από όλες τις μετρικές που ορίζονται στο μοντέλο QMOOD θα εξετάσουμε μόνο αυτές που αναφέρονται σε επίπεδο κλάσης του πηγαίου κώδικα και άρα θα μπορούσαν να χρησιμοποιηθούν για την πρόβλεψη σφαλμάτων [035]:

Αριθμός Δημοσίων Μεθόδων (Number of Public Methods - NPM): Είναι ο αριθμός των μεθόδων σε μια κλάση που έχουν δηλωθεί ως δημόσιες. Αυτή η μετρική είναι γνωστή και ως Μέγεθος Διασύνδεσης Κλάσης (Class Interface Size - CIS).

Μέτρηση Πρόσβασης Δεδομένων (Data Access Metric - DAM): Είναι ο λόγος του αριθμού ιδιωτικών (private) και προστατευμένων(protected) μεταβλητών προς τον συνολικό αριθμό των μεταβλητών που έχουν δηλωθεί στην κλάση. Οι τιμές της κυμαίνονται από μηδέν έως ένα, όπου μια υψηλή τιμή είναι επιθυμητή.

Μέτρο της Συσσωμάτωσης (Measure of Aggregation - MOA): Αφορά τον βαθμό της συσχέτισης μεταξύ ενός μέρους με το όλο (part-whole relationship) που πραγματοποιείται με τη χρήση μεταβλητών. Η μετρική είναι ο αριθμός των μεταβλητών δεδομένων της κλάσης οι οποίες ορίζονται από τύπους που έχει δημιουργήσει ο χρήστης.

Μέτρο της Λειτουργικής Αφαιρετικότητας (Measure of Functional Abstraction - MFA): Η μετρική αυτή είναι ο λόγος των μεθόδων που κληρονομούνται από μία κλάση προς το συνολικό αριθμό μεθόδων που είναι προσβάσιμες από τις μεθόδους της κλάσης.

Συνεκτικότητα Μεταξύ των Μεθόδων της Κλάσης (Cohesion Among Methods of Class - CAM): Η μετρική αυτή υπολογίζει τη σχετικότητα μεταξύ μεθόδων μιας κλάσης και βασίζεται στη λίστα παραμέτρων των μεθόδων. Υπολογίζεται χρησιμοποιώντας το άθροισμα των αριθμών των διαφορετικών τύπων παραμέτρων μεθόδων σε κάθε μέθοδο, διαιρούμενο από το γινόμενο του αριθμού των διαφορετικών παραμετρικών τύπων των μεθόδων σε ολόκληρη την κλάση και τον αριθμό των μεθόδων. Η τιμή της είναι μεταξύ μηδέν και ένα, όπου τιμές κοντά στο ένα είναι προτιμητέες.

3.3.2.4 Συλλογή Μετρικών LK(Lorenz and Kidd Metrics)

Οι Lorenz και Kidd πρότειναν αρκετές μετρικές για να ποσοτικοποιήσουν την ποιότητα λογισμικού δίνοντας μια αιτιολόγηση για τις επιλογές τους [038]. Αυτές που θα μπορούσαν να χρησιμοποιηθούν για την πρόβλεψη σφαλμάτων είναι οι εξής:

Αριθμός Δημόσιων Μεθόδων (Number of Public Methods - NPM): Είναι το άθροισμα όλων των δημόσιων μεθόδων της κλάσης.

Αριθμός Δημόσιων Μεταβλητών (Number of Public Variables - NPV): Είναι το άθροισμα όλων των δημόσιων μεταβλητών της κλάσης.

Αριθμός Μεταβλητών Κλάσης (Number of Class Variables - NCV): Είναι το άθροισμα όλων των μεταβλητών της κλάσης.

Αριθμός Μεθόδων Κλάσης (Number of Class Methods - NCM): Είναι το άθροισμα όλων των μεθόδων της κλάσης.

Αριθμός Επικαλυπτόμενων Μεθόδων(Number of Operations Overridden - NOO): Ένας μεγάλος αριθμός μεθόδων που επικαλύπτονται δείχνει ένα πρόβλημα σχεδιασμού.

Αριθμός Προστιθέμενων Μεθόδων (Number of Operations Added - NOA): Είναι ο αριθμός των μεθόδων που προσθέτουν νέες λειτουργίες σε μια κλάση.

Δείκτης Εξειδίκευσης (Specialization Index - SI): Είναι μια εξίσωση που δείχνει την ποιότητα της κληρονομικότητας δηλαδή SI = (NOO * Επίπεδο Κληρονομικότητας) / Αρ. Μεθόδων Κλάσης

3.3.2.5 Συλλογή Μετρικών MOOD (Metrics for Object Oriented Design - MOOD)

Αποτελείται από έξι μετρικές που μπορούν να υπολογιστούν είτε σε επίπεδο συστήματος είτε σε επίπεδο πακέτου, ανάλογα το επιθυμητό επίπεδο που θέλουμε να κάνουμε τις μετρήσεις [138]:

Παράγοντας Απόκρυψης Μεθόδου (Method Hiding Factor - MHF): Ορίζεται σαν ένα κλάσμα όπου ο αριθμητής είναι το άθροισμα όλων των μη ορατών από το εξωτερικό της κλάσης μεθόδων, που ορίζονται σε όλες τις κλάσεις ενός συστήματος. Ο παρανομαστής του κλάσματος είναι ο συνολικός αριθμός των μεθόδων που ορίζονται στο σύστημα. Ορίζεται τυπικά ως εξής:

$$MHF = \frac{\sum_{i=1}^{TC} \sum_{m=1}^{M_d(C_i)} \left(1 - V\left(M_{mi}\right)\right)}{\sum_{i=1}^{TC} M_d\left(C_i\right)} \quad (3.9)$$

όπου είναι

$M_d(C_i)$ ο αριθμός των μεθόδων που ορίζονται στην κλάση i (3.10)

$$V(M_{mi}) = \frac{\sum_{j=1}^{TC} is\_visible(M_{mi}, C_j)}{TC - 1} \quad (3.11)$$

$$is\_visible(M_{mi}, C_j) = \begin{cases} 1, \text{ αν και μόνο αν } j \neq i \text{ και} \\ C_j \text{ μπορεί να καλέσει την } M_{mi} \\ 0, \text{ αλλιώς} \end{cases} \quad (3.12)$$

Μια υψηλή τιμή δείχνει καλή λειτουργικότητα αλλά και ένα σχεδιασμό που περιλαμβάνει ένα μεγάλο αριθμό μεθόδων οι οποίες δεν μπορούν να επαναχρησιμοποιηθούν και αυτό συνήθως δεν είναι ένα επιθυμητό χαρακτηριστικό.

Παράγοντας Απόκρυψης Μεταβλητών(Attribute Hiding Factor- AHF): Η μετρική AHF ορίζεται σαν ένα κλάσμα, όπου ο αριθμητής είναι το άθροισμα όλων των μη ορατών από το εξωτερικό της κλάσης χαρακτηριστικών, που ορίζονται σε όλες τις κλάσεις ενός συστήματος. Ο παρανομαστής του κλάσματος είναι ο συνολικός αριθμός των χαρακτηριστικών ορατών και μη ορατών του συστήματος. Είναι επιθυμητό να παίρνει μεγάλες τιμές δηλαδή μεγάλο ποσοστό των χαρακτηριστικών του συστήματος να μην είναι ορατά. Μια μικρή τιμή δείχνει πως έχουμε μια κακή σχεδίαση. Ο τυπικός της ορισμός είναι αντίστοιχος με αυτόν της MHF αν στη θέση της μεθόδου βάλουμε ένα χαρακτηριστικό και δεν θα τον επαναλάβουμε. εδώ.

Παράγοντας της Κληρονομικότητας Μεθόδου(Method Inheritance Factor - MIF): Ορίζεται σαν ένα κλάσμα όπου ο αριθμητής είναι ίσος με το συνολικό άθροισμα των μεθόδων που κληρονομούνται σε όλες τις κλάσεις του συστήματος και ο παρανομαστής ίσος με το συνολικό αριθμό όλων των διαθέσιμων μεθόδων στο σύνολο των κλάσεων. Ορίζεται τυπικά ως εξής:

$$MIF = \frac{\sum_{i=1}^{TC} M_i(C_i)}{\sum_{i=1}^{TC} M_a(C_i)} \quad (3.13)$$

Όπου

$$M_a(C_i) = M_d(C_i) + M_i(C_i)$$

$M_a(C_i)$ είναι ο αριθμός των μεθόδων που μπορούν να χρησιμοποιηθούν στην κλάση i

$M_d(C_i)$ είναι ο αριθμός των μεθόδων που ορίζονται στην κλάση i

$M_i(C_i)$ είναι ο αριθμός των μεθόδων που κληρονομούνται (χωρίς επικάλυψη) στην κλάση i

Προσθέτοντας ανεξάρτητες κλάσεις στο σύστημα, δηλαδή κλάσεις που δεν κληρονομούν ούτε έχουν υποκλάσεις ελαττώνουμε την τιμή της μετρικής MIF. Γενικά η τιμή της μετρικής MIF, πρέπει να είναι σε λογικά όρια για ένα σύστημα. Πολύ υψηλή τιμή δείχνει πλεονασματική κληρονομικότητα ή μεγάλη εμβέλεια μελών.

Παράγοντας της Κληρονομικότητας Μεταβλητών(Attribute Inheritance Factor - AIF): Ορίζεται σαν ένα κλάσμα όπου ο αριθμητής είναι ίσος με το συνολικό άθροισμα των χαρακτηριστικών (attributes) που κληρονομούνται σε όλες τις κλάσεις του συστήματος και ο παρανομαστής είναι ίσος με το συνολικό αριθμό όλων των διαθέσιμων χαρακτηριστικών των κλάσεων. Μία κλάση που κληρονομεί πολλά χαρακτηριστικά από τους προγόνους της συμβάλει στην αύξηση της τιμής της μετρικής AIF. Αν προσθέσουμε νέα ορατά πεδία στις θυγατρικές κλάσεις η τιμή της μετρικής AIF μειώνεται αφού αυξάνεται η τιμή του παρανομαστή. Πολύ υψηλή τιμή της μετρικής AIF δείχνει πλεονασματική κληρονομικότητα ή υπερβολική χρήση ορατών χαρακτηριστικών. Επίσης, μία χαμηλή τιμή μπορεί να σημαίνει έλλειψη κληρονομικότητας ή τη χρήση πολλών μη ορατών χαρακτηριστικών. Ο τυπικός της ορισμός είναι αντίστοιχος με αυτόν της MIF αν στη θέση της μεθόδου βάλουμε μια μεταβλητή και δεν θα τον επαναλάβουμε πάλι εδώ.

Παράγοντας Σύζευξης (Coupling Factor - CF): Μετράει το βαθμό σύζευξης μεταξύ των κλάσεων ενός συστήματος. Ορίζεται σαν ένα κλάσμα όπου ο αριθμητής είναι ο αριθμός όλων των ενεργών συζεύξεων μεταξύ των κλάσεων και ο παρανομαστής είναι ο μέγιστος αριθμός όλων των δυνατών συζεύξεων μεταξύ των κλάσεων σε ένα σύστημα. Ενεργή σύζευξη της κλάσης Α με την κλάση Β, σημαίνει ότι η κλάση Α καλεί μεθόδους ή έχει πρόσβαση στα χαρακτηριστικά της κλάσης Β. Εάν η Β καλεί μεθόδους ή έχει πρόσβαση στα χαρακτηριστικά της Α τότε και αυτή η

σύζευξη υπολογίζεται στις ενεργές συζεύξεις. Με άλλα λόγια, συμπεριλαμβάνονται οι συζεύξεις και των δύο κατευθύνσεων ανάμεσα σε δύο κλάσεις. Μέγιστος αριθμός δυνατών συζεύξεων, είναι ο αριθμός συζεύξεων μεταξύ όλων των κλάσεων του συστήματος, αν υποθέσουμε ότι όλες αυτές οι κλάσεις βρίσκονται σε σύζευξη μεταξύ τους. Να σημειώσουμε ότι δεν μετρά τις συζεύξεις μεταξύ κλάσεων προγόνων με απογόνων σε σχέσεις κληρονομικότητας. Τυπικά μπορεί να οριστεί ως εξής:

$$CF = \frac{\sum_{i=1}^{TC}\left[\sum_{j=1}^{TC} is\_client\left(C_i, C_j\right)\right]}{TC^2 - TC} \quad (3.14)$$

όπου

$$is\_client\left(C_i, C_j\right) = \begin{cases} 1, \text{ αν και μόνο αν } C_i \Rightarrow C_j \text{ και } C_i \neq C_j \\ 0, \text{ αλλιώς} \end{cases}$$

$C_i \Rightarrow C_j$ αναπαριστά την σχέση μεταξύ της κλάσης i που καλεί την j

Παράγοντας Πολυμορφισμού (Polymorphism Factor - PF): Ορίζεται σαν ένα κλάσμα όπου ο αριθμητής αναπαριστά τον πραγματικό αριθμό όλων των δυνατών πολυμορφικών καταστάσεων σε ένα σύστημα. Ο παρανομαστής αναπαριστά το μέγιστο αριθμό όλων των δυνατών διαφορετικών πολυμορφικών καταστάσεων για όλες τις κλάσεις που υποστηρίζουν την επικάλυψη (overriding) μεθόδων στο σύστημα. Υποθέτοντας ότι μία νέα μέθοδος προστίθεται σε μία κλάση της οποίας οι μέθοδοι επικαλύπτονται στις κλάσεις απογόνους ο αριθμός των φορών που θα επικαλύπτει σε όλες τις τάξεις απογόνους της είναι ο μέγιστος αριθμός όλων των δυνατών πολυμορφικών καταστάσεων της συγκεκριμένης κλάσης. Τυπικά ορίζεται ως εξής:

$$PF = \frac{\sum_{i=1}^{TC} \mathrm{M}_o\left(C_i\right)}{\sum_{i=1}^{TC}\left[M_n\left(C_i\right) \cdot DC\left(C_i\right)\right]} \quad (3.15)$$

όπου

$M_d(C_i) = M_n(C_i) + M_o(C_i)$

$M_d(C_i)$ είναι ο αριθμός των μεθόδων που ορίζονται στην κλάση i

$M_n(C_i)$ είναι ο αριθμός των νέων μεθόδων που ορίζονται στην κλάση i

$M_o(C_i)$ είναι ο αριθμός των μεθόδων στην κλάση i που επικαλύπτουν μεθόδους

$DC(C_i)$ είναι ο αριθμός των απογόνων της κλάσης i

Κεφάλαιο 4
Ερευνητική Μεθοδολογία και Επιλογή Εργαλείων

Ξεκινάμε με μια σύγκριση των κυριότερων συλλογών αντικειμενοστρεφών μετρικών και κάνουμε μια αρχική επιλογή μετρικών για τα μοντέλα μας. Συνεχίζουμε με μια μεγάλη έρευνα για να βρούμε τα εργαλεία που να υποστηρίζουν τις συγκεκριμένες μετρικές και να είναι ανοικτού κώδικα. Το πιο πλήρες εργαλείο για τους σκοπούς μας αναδείχτηκε το ckjm στην επεκταμένη του έκδοση [142]. Στη συνέχεια αναζητήσαμε ένα πρόγραμμα ανοικτού κώδικα που να είναι γραμμένο σε Java, να είναι μεσαίου μεγέθους και να υπάρχουν διαθέσιμα τα σφάλματα που διορθώθηκαν σε κάθε έκδοση του. Τελικά επιλέξαμε την έκδοση 3.2 του προγράμματος jEdit. Κατεβάσαμε τον πηγαίο κώδικα της έκδοσης 3.2 του jEdit και με την χρήση του ckjm extended πήραμε τα αποτελέσματα των μετρικών. Μετά έπρεπε να επιλέξουμε τεχνικές κατασκευής για τα μοντέλα μας. Έγινε εκτεταμένη αναζήτηση σε άρθρα, έρευνες και γενικότερα σχετική βιβλιογραφία, όπου καταλήξαμε σε δυο μεθόδους στατιστικής ανάλυσης και δυο μεθόδους μηχανικής μάθησης. Αναζητήθηκαν και τα αντίστοιχα εργαλεία που να υποστηρίζουν τις συγκεκριμένες μεθόδους με κριτήρια επιλογής να είναι ανοικτού κώδικα και να υπάρχει μεγάλη κοινότητα χρηστών στην περίπτωση που θα χρειαζόμασταν κάποιας μορφής υποστήριξη. Επιλέξαμε το πρόγραμμα R για τις στατιστικές μεθόδους και το WEKA για την μηχανική μάθηση.

4.1 Σύγκριση και Επιλογή Μετρικών

Οι μετρικές που παρουσιάστηκαν στην ενότητα 3.3 είναι ένα μικρό μέρος των μετρικών που έχουν προταθεί στην βιβλιογραφία αλλά σίγουρα είναι αυτές που έχουν τραβήξει περισσότερο το ενδιαφέρον τον ερευνητών, αφού υπάρχουν αρκετές έρευνες που τις χρησιμοποιούν είτε για να μετρήσουν κάποια χαρακτηριστικά της ποιότητας τους είτε για διερευνήσουν την τάση για σφάλματα (fault proneness) ή της προσπάθειας που απαιτείται για την συντήρηση του λογισμικού (maintenance effort). Ακόμη όμως και αυτές οι μετρικές της ενότητας 3.3 είναι αρκετά μεγάλος αριθμός και θα πρέπει με κάποιο τρόπο να τις περιορίσουμε. Έτσι, θα προσπαθήσουμε να αναγνωρίσουμε ποιες από αυτές είναι οι καταλληλότερες για την χρήση ως είσοδο στα μοντέλα που θα κατασκευάσουμε, ώστε στη συνέχεια να επικεντρώσουμε μετά την έρευνα μας σε εργαλεία μέτρησης μετρικών που να τις υποστηρίζουν. Η συλλογή μετρικών CK είναι μια εύκολη και ασφαλής επιλογή λόγω της μεγάλης προσοχής που έχει προσελκύσει από την ακαδημαϊκή κοινότατα με την παραγωγή εκατοντάδων άρθρων που την χρησιμοποιούν αλλά και της ευρείας αποδοχής που έχει γνωρίσει από την συντριπτική πλειοψηφία των εργαλείων για μετρικές λογισμικού. Πρόκειται λοιπόν για μια συλλογή μετρικών που έχει αποδείξει την άξια της και την επιλέγουμε ως σημείο αναφοράς και για τις υπόλοιπες μετρικές.

Δυστυχώς, η αναζήτηση για την δεύτερη συλλογή μετρικών που θέλουμε να συμπεριλάβουμε στο εμπειρικό κομμάτι της διπλωματικής διατριβής δεν ήταν καθόλου εύκολη υπόθεση. Για την διευκόλυνση μας μετά από έρευνα στη βιβλιογραφία καταλήξαμε σε μια σειρά από επιθυμητά κριτήρια που θα ήταν καλό να ικανοποιεί η επιλογή μας για την δεύτερη συλλογή μετρικών (πέρα της CK που είναι πρώτη μας επιλογή) που θα επιλέγουμε. να μετρήσουμε [138, 139, 140]:

- **Υψηλού Επιπέδου Χαρακτηριστικά;** Να βασίζονται κατά προτίμηση σε υψηλού επιπέδου χαρακτηριστικά ώστε να μπορούν να χρησιμοποιηθούν από τα πρώτα στάδια κατά την υλοποίηση του λογισμικού.

- **Σαφή Ποιοτικά Χαρακτηριστικά;** Οι στόχοι του μοντέλου σχετικά με τα χαρακτηριστικά της ποιότητας που μετρώνται θα πρέπει να αναφέρονται ρητά.

- **Σαφής Ορισμός Μετρικών;** Οι μετρικές που προτείνει θα πρέπει να ορίζονται με σαφή τρόπο και να μην αφήνουν περιθώρια για παρερμηνείες ή για διαφορετική υλοποίηση ανάλογα με την γλώσσα προγραμματισμού.

- **Τυπικές Σχέσεις;** Θα πρέπει οι σχέσεις ανάμεσα στα χαρακτηριστικά της ποιότητας και των μετρικών να εκφράζονται κατά προτίμηση με τυπικό και αυστηρό τρόπο.

- **Ερμηνεία Αποτελεσμάτων;** Θα πρέπει να υπάρχει ερμηνεία των αποτελεσμάτων γιατί απλά ένα νούμερο χωρίς τίποτα άλλο δεν προσθέτει κάτι για την κατανόηση μας.

- **Εμπειρική Επαλήθευση;** Η Αξία τους θα πρέπει να έχει αποτιμηθεί εμπειρικά γιατί για τις συλλογές που δεν έχουν επαληθευτεί στην πράξη θα έχουμε πάντα αμφιβολίες σχετικά με την ορθότητα τους.

	CK	Lorenz και Kidd	Martin	MOOD	QMOOD
Υψηλού Επιπέδου Χαρακτηριστικά;	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ
Σαφή Ποιοτικά Χαρακτηριστικά;	ΌΧΙ	ΌΧΙ	ΌΧΙ	ΝΑΙ	ΝΑΙ
Σαφής Ορισμός Μετρικών;	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ
Τυπικές Σχέσεις;	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ
Ερμηνεία Αποτελεσμάτων;	ΝΑΙ/ΟΧΙ	ΌΧΙ	ΌΧΙ	ΝΑΙ	ΝΑΙ
Εμπειρική Επαλήθευση;	ΝΑΙ	ΌΧΙ	ΝΑΙ	ΝΑΙ	ΝΑΙ

Πίνακας 4.1: Συλλογές Μετρικών και Ποιες Επιθυμητές Ιδιότητες Ικανοποιούν

Τα αποτελέσματα για όλα τα παραπάνω κριτήρια για τις συλλογές μετρικών που εξετάσαμε στην ενότητα 3.3 παρουσιάζονται στην πίνακα 4.1. Στην ερμηνεία αποτελεσμάτων για τη συλλογή CK έχουμε βάλει και "ΝΑΙ" και "ΟΧΙ" γιατί στην αρχική τους έκδοση δεν ανέφεραν καθόλου για ποιοτικά χαρακτηριστικά αλλά υπάρχουν κατοπινές μελέτες που δείχνουν ότι υπάρχει ερμηνεία για τα αποτελέσματα μετρικών της συλλογής. Είναι φανερό ότι οι συλλογές MOOD και QMOOD είναι οι καλύτερες επιλογές με βάση τις επιθυμητές ιδιότητες που ορίσαμε. Επειδή όμως οι μετρικές MOOD αναφέρονται σε επίπεδο συστήματος και εμείς χρειαζόμαστε μετρικές σε επίπεδο κλάσης θα επιλέξουμε την συλλογή QMOOD. Άρα το εργαλείο μετρικών που θα αναζητήσουμε στην συνέχεια θα πρέπει να υποστηρίζει τις συλλογές CK και QMOOD. Επιπλέον, από τις παραδοσιακές μετρικές οι γραμμές πηγαίου κώδικα και η κυκλωματική πολυπλοκότητα έχουν βρεθεί να έχουν στατιστικά θετική σχέση με την ύπαρξη σφάλματος και άρα καλό θα ήταν να παρέχονται από το εργαλείο.

4.2 Σύγκριση και Επιλογή Εργαλείων για Μετρικές

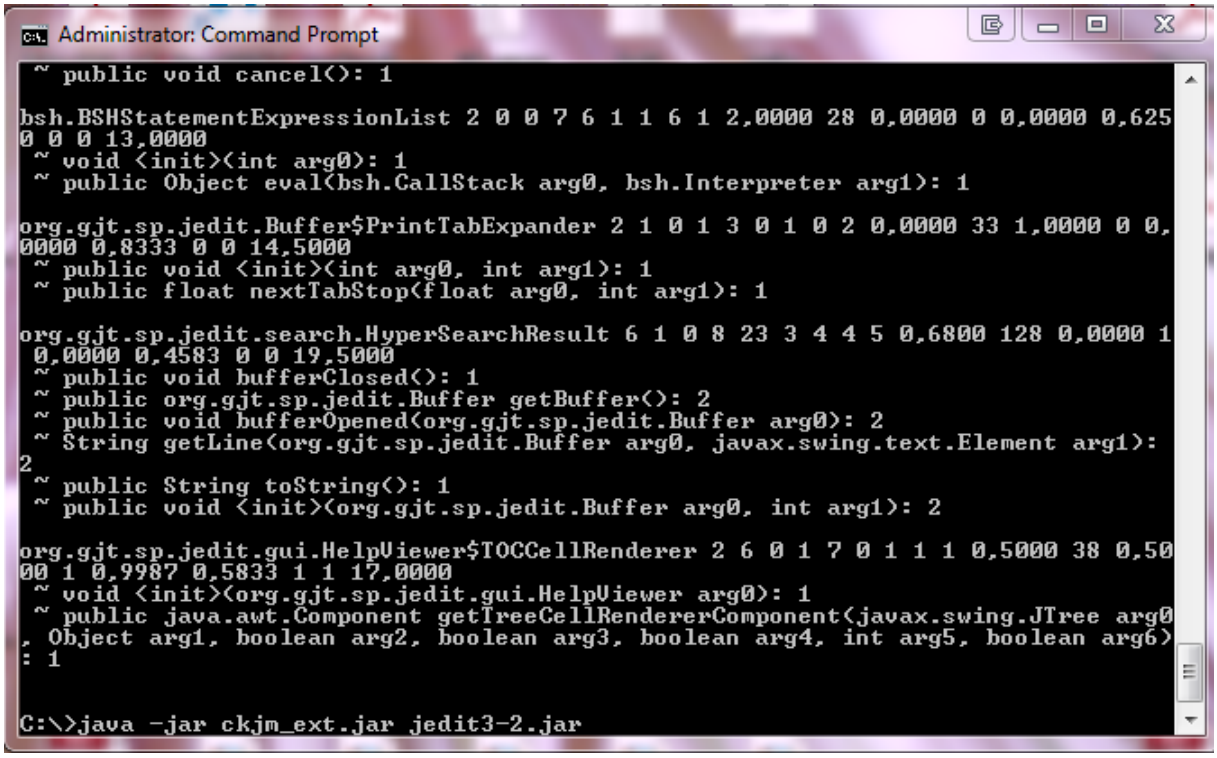
Για την εύρεση των κατάλληλων εργαλείων που θα μπορούσαν να εκτελέσουν τις μετρικές που επιλέξαμε στην προηγούμενη ενότητα, ξεκινήσαμε με μια έρευνα στο διαδίκτυο για να εντοπίσουμε όσο περισσότερα εργαλεία μπορούσαμε. Συλλέξαμε περισσότερα από πενήντα προγράμματα που ήταν είτε εργαλεία αποκλειστικά για μετρικές είτε υποστήριζαν έναν ικανοποιητικό αριθμό μετρικών. Στην συνέχεια αρχίσαμε να εφαρμόζουμε κάποια κριτήρια για να περιορίσουμε τον αριθμό τους και πιο συγκεκριμένα να μπορούν να μετρήσουν πηγαίο κώδικα σε Java και να διατίθενται είτε ελεύθερα (ελεύτερο λογισμικό, λογισμικό ανοικτού κώδικα ή freeware) είτε σε δοκιμαστική έκδοση λίγων ημερών που να παρέχει όλες τις λειτουργίες του προγράμματος. Μετά την εφαρμογή των παραπάνω περιορισμών τα εργαλεία μειώθηκαν στα δέκα πέντε και μια παρουσίαση τους γίνεται στο παράρτημα Ε. Κατά την διάρκεια της αξιολόγησης τους επειδή οι μετρήσεις περιλάμβαναν κάθε φορά τα ίδια αρχεία, ήταν λογικό να αναμένουμε ότι τα εργαλεία μετρικών θα έδιναν τις ίδιες ή πολύ κοντινές μετρήσεις. Δυστυχώς, όσο περισσότερο ερευνούσαμε το θέμα τόσο περισσότερο επιβεβαιώναμε την αρχική υποψία μας ότι υπάρχει έλλειψη αξιοπιστίας στα εργαλεία των μετρικών.

Αυτό είναι κάτι που σχεδόν όλες οι μελέτες που εξετάσαμε φαίνεται να παραβλέπουν και να μην λαμβάνουν υπ' όψιν τους. Οι διαφορές μπορεί να οφείλονται είτε στην ασάφεια του ορισμού μιας μετρικής με αποτέλεσμα το κάθε εργαλείο να κάνει διαφορετική υλοποίηση (π.χ. στο πως χειριζόμαστε τις αφηρημένες κλάσεις), είτε στο γεγονός ότι αρκετά εργαλεία χρησιμοποιούν δικό τους ορισμό που είναι παραλλαγή του επίσημου ορισμού είτε και σε σφάλματα υλοποίησης στα ίδια τα προγράμματα των μετρικών. Αυτό συνεπάγεται ότι κατά την μοντελοποίηση αν πάρουμε δεδομένα από διαφορετικά εργαλεία μετρικών για τον ίδιο πηγαίο κώδικα μπορεί να πάρουμε διαφορετικά αποτελέσματα. Οι Breuker et al [141] μελέτησαν αυτό το πρόβλημα και ανέφεραν ότι η διαφορά μπορεί κάποιες φορές να είναι μικρή δηλαδή μικρότερη του 1% αλλά κάποιες άλλες φορές οι διαφορές ήταν μεγαλύτερες του 10% και αυτό είναι κάτι που δεν μπορούμε να αγνοήσουμε. Στον πίνακα 4.2 παρουσιάζεται ανάγλυφα τα αποτελέσματα τους για την συλλογή μετρικών CK από τρία διαφορετικά εργαλεία μετρικών για τέσσερις ενδεικτικές κλάσεις. Καταλήγουν ότι υπάρχουν δυο κόσμοι, ο κόσμος των βιβλίων και των ερευνητικών εργασιών με τους τυπικούς ορισμούς και ο κόσμος των εργαλείων μετρικών. Μάλιστα ο κόσμος των εργαλείων μετρικών χωρίζεται σε δυο διαφορετικούς κόσμους, σε αυτό που λέει το εργαλείο ότι κάνει και σε αυτό που πραγματικά κάνει. Υπήρξαν περιπτώσεις που η υλοποίηση ενός εργαλείου είχε σφάλμα τα με αποτέλεσμα να μην παίρνουμε στην έξοδο το σωστό αποτέλεσμα.

	Κλάση 1 (LOC 16)			*Κλάση 2 (LOC 120)*			*Κλάση 3 (LOC 226)*			*Κλάση 4 (LOC 234)*		
	ckjm	**ES**	**RefIT**	**ckjm**	**ES**	**RefIT**	**ckjm**	**ES**	**RefIT**	**ckjm**	**ES**	**RefIT**
WMC	3	3	3	17	17	17	13	43	68	14	32	38
DIT	1	0	1	1	0	1	2	0	2	0	1	3
NOC	0	0	0	0	0	0	1	1	1	0	0	0
CBO	0	0	-	2	0	-	15	0	-	7	0	-
RFC	4	3	0	28	18	16	23	23	19	27	25	22
LCOM	1	0	0	74	57	0,825	46	42	0,933	27	10	0,811

Πίνακας 4.2: Διαφορές Αποτελεσμάτων Μετρικών της Συλλογής CK για Τέσσερις Ενδεικτικές Κλάσεις

Λαμβάνοντας υπόψιν τις συλλογές των μετρικών που υποστήριζε το καθένα από τα προγράμματα που αξιολογήσαμε και τα προβλήματα στα αποτελέσματα τους, καταλήξαμε τελικά στο εργαλείο ckjm extented [142]. Πρόκειται για μια αρκετά εξελιγμένη έκδοση του αρχικού ckjm [143] που υπολογίζει συνολικά 19 μετρικές. Από αυτές οι 17 είναι αντικειμενοστρεφής και οι 2 παραδοσιακές. Στην εικόνα 4.1 παρουσιάσουμε την εκτέλεση του και στον πίνακα 4.3 παρουσιάσουμε τις μετρικές που υπολογίζει. Η έξοδος του είναι με την ίδια σειρά των μετρικών του πίνακα 4.3, απλά μια κλάση έχει πολλές γραμμές γιατί αναφέρεται η κυκλωματική πολυπλοκότητα ανά μέθοδο. Αυτά είναι τα αποτελέσματα που θα χρησιμοποιήσουμε σαν είσοδο στα μοντέλα μας. Περισσότερες λεπτομέρειες για την επεξεργασία που πραγματοποιήθηκε στην έξοδο του προγράμματος στην επόμενη ενότητα.



Εικόνα 4.1: Το Πρόγραμμα ckjm extended σε Λειτουργία

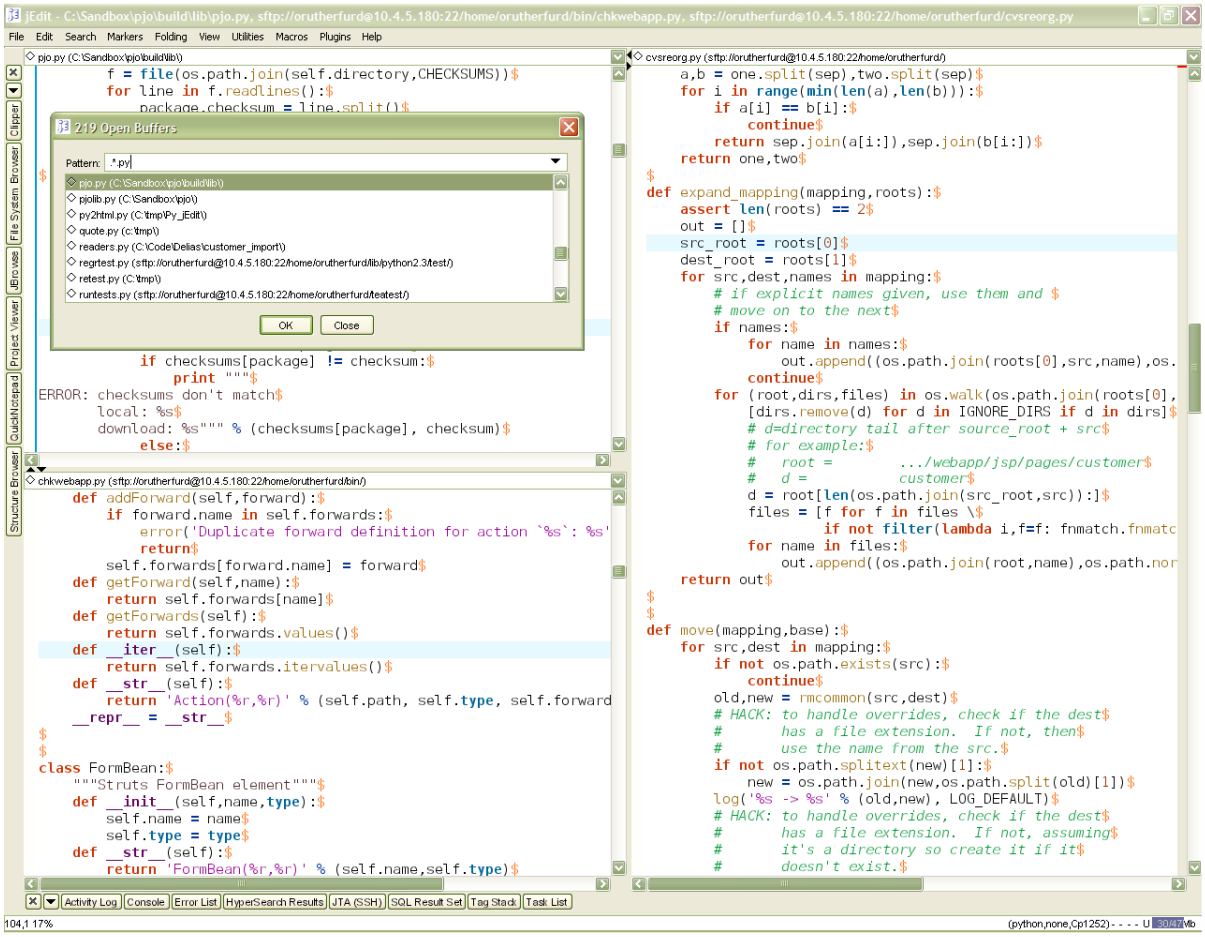
Α/Α	Ονομασία Μετρικής	Ορισμός	Πηγή
1	Σταθμισμένες Μέθοδοι ανά Κλάση (Weighted methods per class, WMC)	Η τιμή της WMC ισούται με τον αριθμό των μεθόδων στην κλάση (υποθέτοντας μοναδιαία βάρη σε όλες τις μεθόδους).	[039]
2	Βάθος του Δένδρου Κληρονομικότητας (Depth of Inheritance Tree, DIT)	Η μετρική DIT παρέχει για κάθε κλάση τον αριθμό των επιπέδων κληρονομικότητας από την κορυφή της ιεραρχίας του δέντρου κληρονομικότητας.	[039]
3	Αριθμός Απογόνων (Number of Children, NOC)	Η μετρική NOC μετράει τον αριθμό των άμεσων απογόνων της κλάσης.	[039]
4	Σύζευξη Μεταξύ Κλάσεων (Coupling between object classes, CBO)	Η μετρική CBO αναπαριστά τον αριθμό των κλάσεων που είναι συζευγμένες με την κλάση (φυγόκεντρος και κεντρομόλος σύζευξη). Αυτές οι συζεύξεις βρίσκονται σε κλήσεις μεθόδου, πρόσβαση πεδίων, κληρονομικότητα, ορίσματα μεθόδων, επιστροφή μεθόδου και εξαιρέσεις.	[039]
5	Απόκριση για μία κλάση (Response for a Class, RFC)	Η μετρική RFC μετράει τον αριθμό των διαφορετικών μεθόδων που μπορούν να εκτελεστούν όταν ένα αντικείμενο της κλάσης λαμβάνει ένα μήνυμα. Ιδανικά, επιθυμούμε να βρούμε τις μεθόδους που θα καλέσει η κλάση σε κάθε μέθοδο, και να το επαναλάβουμε για κάθε μέθοδο, υπολογίζοντας το μεταβατικό κλείσιμο του γράφου της μεθόδου κλήσης. Ωστόσο, αυτή η διαδικασία μπορεί να είναι δαπανηρή και αρκετά ανακριβής. Το Ckjm υπολογίζει μια πρόχειρη στρογγυλοποίηση της απόκρισης, απλά επιθεωρώντας τις κλήσεις μεθόδων μέσα στο σώμα των μεθόδων κλάσεων. Η τιμή της RFC είναι το άθροισμα του αριθμού των μεθδόων που καλείται μέσα στο σώμα της μεθόδου κλάσεων και του αριθμού των μεθόδων κλάσεων. Αυτή η απλοποίηση χρησιμοποιείται επίσης στους Chidamber & Kemerer [039].	[039]
6	Έλειψη Συνεκτικότητας των Μεθόδων (Lack of cohesion in methods, LCOM)	Η μετρική LCOM μετράει τις ομάδες των μεθόδων στε μια κλάση, οι οποίες δε σχετίζονται με κοινά πεδίων κλάσεων. Ο αρχικός ορισμός αυτής της μετρικής (αυτός που χρησιμοποιείται στο ckjm) θεωρεί όλα τα ζευγη μεθόδων της κλάσης. Σε κάποια απο αυτά τα ζευγη και οι δύο μέθοδοι προσπελαύνουν τουλάχιστον ένα κοινό πεδίο της κλάσης, ενώ σε άλλα ζεύγη οι δύο μέθοδοι δεν έχουν κανένα κοινό πεδίο προσπέλασης. Η έλλειψη συνεκτικότητας σε μεθόδους, υπολογίζεται αφαιρώντας απο τον αριθμό των ζευγών μεθόδων τα οποία δεν μοιράζονται ένα πεδίο πρόσβασης, τα ζεύγη εκείνα που μοιράζονται πεδίο πρόσβασης.	[039]
7	Έλειψη συνεκτικότητας στις μεθόδους (Lack of cohesion in methods, LCOM3)	$LCOM3 = \frac{\left(\frac{1}{a}\sum_{j=1}^{a}\mu(A_j)\right) - m}{1-m}$ m = ο αριθμός μεθόδων σε μια κλάση, α = αριθμός μεταβλητών μιας κλασης μ(Α) = ο αριθμός μεθόδων που προσπελαύνουν τη μεταβλητή Α.	[143]
8	Φυγόκεντρες Συζεύξεις (Afferent Couplings, Ca)	Η μετρική Ca αναπαριστά τον αριθμό των κλάσεων που εξαρτώνται από την κλάση η οποία μετράται.	[137]
9	Κεντρομόλες Συζεύξεις (Efferent Couplings Ce)	Η μετρική Ce αναπαριστά τον αριθμό των κλάσεων από τις οποίες εξαρτάται η κλάση που μετράται.	[137]
10	Αριθμός Δημοσίων Μεθόδων (Number of Public Methods, NPM)	Η μετρική NPM μετράει όλες τις μεθόδους σε μια κλάση που έχουν δηλωθεί ως δημόσιες. Αυτή η μετρικη είναι γνωστή και ως Class Interface Size (CIS).	[035]
11	Μέτρηση Πρόσβασης Δεδομένων (Data Access Metric, DAM)	Αυτή η μετρική είναι ο λόγος του αριθμού ιδιωτικών (protected) μεταβλητών προς τον συνολικό αριθμό των μεταβλητών που έχουν δηλωθεί στην κλάση.	[035]
12	Μέτρο της Συσσωμάτωσης (Measure of Aggregation, MOA)	Η μετρική MOA μετράει το βαθμό της σχέσης μέρους με όλο (part-whole) που πραγματοποιείται με τη χρήση μεταβλητών. Η μετρική είναι ο αριθμός των πεδίων στις κλάσεις που ορίζονται από το χρήστη.	[035]
13	Μέτρηση Λειτουργικής Αφαιρετικότητας (Measure of Functional Abstraction, MFA)	Η μετρική αυτή είναι ο λόγος των μεθόδων που κληρονομούνται από μία κλάση προς το συνολικό αριθμό προσβάσιμων μεθόδων από τις μεθόδους της κλάσης. Το java.lang.Object (ως γονιός) και οι κατασκευαστές (constructors) αγνοούνται.	[035]
14	Συνεκτικότητα Μεταξύ των Μεθόδων της Κλάσης (Cohesion Among Methods of Class, CAM)	Η μετρική αυτή υπολογίζει τη σχετικότητα μεταξύ μεθόδων μιας κλάσης και βασίζεται στη λίστα παραμέτρων των μεθόδων. Υπολογίζεται χρησιμοποιώντας το άθροισμα των αριθμών των διαφορετικών τύπων παραμέτρων μεθόδων σε κάθε μέθοδο, διαιρούμενο από το γινόμενο του αριθμού των διαφορετικών παραμέτρων μεθόδων σε ολόκληρη την κλάση και τον αριθμό των μεθόδων.	[035]
15	Σύζευξη Κληρονομικότητας (Inheritance Coupling, IC)	Αυτή η μετρική παρέχει τον αριθμό των κλάσεων γονέων οι οποίοι συνδέονται με μια συγκεκριμένη κλάση. Μια κλάση συνδέεται στην κλάση - γονέα της εάν μια από τις μεθόδους που κληρονομεί εξαρτώνται από τις νέες ή επανακαθορισμένες μεθόδους στην κλάση. Μια κλάση συνδέεται με την κλάση γονέα εάν ικανοποιείται μια απο τις παρακάτω συνθήκες: - Μια απο τις μεθοδους που κληρονομεί χρησιμοποιεί ένα γνώρισμα που ορίζεται σε μια νέα/επανακαθοριζόμενη μέθοδο. - Μια από τις μεθοδους που κληρονομεί καλεί μια επανακαθοριζόμενη μέθοδο. - Μια απο τις μεθοδους που κληρονομει καλειται από μια επανακαθοριζομενη μεθοδο και χρησιμοποιει μια παράμετρο που ορίζεται στην επανακαθοριζόμενη μέθοδο.	[005]
16	Σύζευξη Μεθόδων (Coupling Between Methods, CBM)	Αυτή η μετρική μετράει τον συνολικό αριθμό των νέων/επανακαθοριζόμενων μεθόδων για τις οποίες όλες οι κληρονομούμενες μέθοδοι συνδέονται. Υπάρχει σύνδεση/σύζευξη όταν τουλάχιστον μια από τις προϋποθέσεις που αναφέρονται στη μετρική IC ισχύει.	[005]
17	ΜέσηΠολυπλοκότητας Μεθόδων (Average Method Complexity, AMC)	Αυτή η μετρική μετράει το μέσο μέγεθος κάθε κλάσης. Το μέγεθος κάθε μεθόδου ισούται με τον αριθμό των bytecodes της συγκεκριμένης μεθόδου.	[005]
18	Κυκλωματική Πολυπλοκότητα McCabe (McCabe's cyclomatic complexity, CC)	Η CC ισούται με τον αριθμό των διαφορετικών μονοπατιών σε μια μέθοδο (method) συν ένα (+ 1). Η κυκλωματική πολυπλοκότητα ορίζεται ως: CC = E - N + P, E = αριθμός ακμών σε ένα γράφημα, N = αριθμός κόμβων σε ένα γράφημα, P = αριθμός συνδεδεμένων συνιστωσών. Η CC είναι η μοναδική μετρική μεγέθους μεθόδου. Τα κατασκευαζόμενα μοντέλα προβλέπουν το μέγεθος της κλάσης. Επομένως, η μετρική θα έπρεπε να μετατρέπει την τιμή αυτή σε μετρική μεγέθους κλάσης. Έτσι προέκυψαν δύο μετρικές: CC_MAX η μεγαλύτερη τιμή της CC μεταξύ των μεθόδων της εξεταζόμενης κλάσης και CC_AVG :Ο αριθμητικός μέσος της τιμής CC στην εξεταζόμενη κλάση.	[063]
19	Γραμμές Κώδικα (Lines of Code, LOC)	Η μετρική LOC βασίζεται στο πηγαίο κώδικα bytecode της Java. Πρόκειται για το άθροισμα των αριθμών πεδίων, μεθόδων και οδηγιών σε κάθε μέθοδο της εξεταζόμενης κλάσης.	[127]

Πίνακας 4.3: Ορισμός Μετρικών Εργαλείου ckjm extended και Σχετικές Αναφορές

4.3 Επιλογή Λογισμικού Μελέτης και Διαδικασία Συλλογής Δεδομένων

Από τις επιλογές που έχουμε κάνει σε προηγούμενες ενότητες έχουν προκύψει δυο βασικοί περιορισμοί για το πρόγραμμα που θα αντλήσουμε τα δεδομένα για την παρούσα μελέτη. Πρέπει να είναι λογισμικό ανοικτού κώδικα και να είναι υλοποιημένο στην αντικειμενοστρεφή γλώσσα προγραμματισμού Java. Υπάρχει μια μεγάλη πληθώρα προγραμμάτων που καλύπτουν τους παραπάνω δυο περιορισμούς. Για να μειώσουμε τους πιθανούς υποψηφίους αλλά και για να υπάρχουν αρκετές αναφορές σφαλμάτων επιλέξαμε να ασχοληθούμε με τις περιπτώσεις έργων ανοικτού κώδικα όπου υπάρχει μια μεγάλη ενεργή κοινότητα και καταγράφονται τα σφάλματα σε κάποιο σύστημα διαχείρισης σφαλμάτων (bug tracking system). Οι αρχικές μας επιλογές ήταν μεταξύ του OpenOffice [090] και του Eclipse [091], προγράμματα ανοικτού κώδικα με τεράστια βάση χρηστών που χρησιμοποιούν το ανοικτού κώδικα σύστημα διαχείρισης σφαλμάτων Bugzilla [092]. Η συλλογιστική μας ήταν να συνδυάσουμε την πληροφορία που υπήρχε στο σύστημα διαχείρισης πηγαίου κώδικα CVS που περιέχει τις αλλαγές στον πηγαίο κώδικα με σφάλματα που υπήρχαν στο Bugzilla. Ο συνδυασμός των στοιχείων που περιέχουν αυτά τα δυο συστήματα θα μπορούσε να οδηγήσει στην αυτοματοποιημένη αντιστοίχιση της διόρθωσης των σφαλμάτων με τις αντίστοιχες κλάσεις που μεταβλήθηκαν. Όμως, δεν υπάρχει καθόλου υποστήριξη από τα δυο συστήματα ώστε να μπορούν να συνδυαστούν οι πληροφορίες που υπάρχουν από αυτά τα δύο συστήματα με εύκολο τρόπο σε ενιαία πληροφορία [093].

Έτσι, επειδή τα δυο παραπάνω προγράμματα έχουν χιλιάδες κλάσεις θα ήταν αδύνατο να γίνει χειροκίνητα η αντιστοίχιση των σφαλμάτων με τις κλάσεις. Οπότε για να προχωρήσουμε την έρευνας μας, αποφασίσαμε να εγκαταλείψουμε την ιδέα να χρησιμοποιήσουμε ένα από τα δυο παραπάνω προγράμματα και επικεντρωθήκαμε στην αναζήτηση ενός λογισμικού ανοικτού κώδικα που να είναι μεσαίου μεγέθους, ώστε να μπορεί να γίνει χειροκίνητα η αντιστοίχηση. Από όλα τα υποψήφια λογισμικά καταλήξαμε σε λίστα που περιλάμβανε δέκα προγράμματα και τελικά επιλέξαμε το jEdit [094] που είναι ένας κειμενογράφος για προγραμματιστές γραμμένος σε Java και τρέχει σε οποιοδήποτε λειτουργικό σύστημα υπάρχει εικονική μηχανή Java. Περιλαμβάνει υποστήριξη για περισσότερα από διακόσια είδη αρχείων και είναι ιδιαίτερα παραμετροποιήσιμος αφού μπορούν να προστεθούν μακροεντολές οι οποίες είναι γραμμένες σε JavaScript. Επιπλέον, υποστηρίζει πρόσθετα (plug-ins) που επεκτείνουν ακόμη περισσότερο την λειτουργικότητα του όπως π.χ. για την επεξεργασία αρχείων που δεν υποστηρίζει εγγενώς. Στην εικόνα 4.2 δείχνουμε μια χαρακτηριστική εκτέλεση του προγράμματος jEdit.

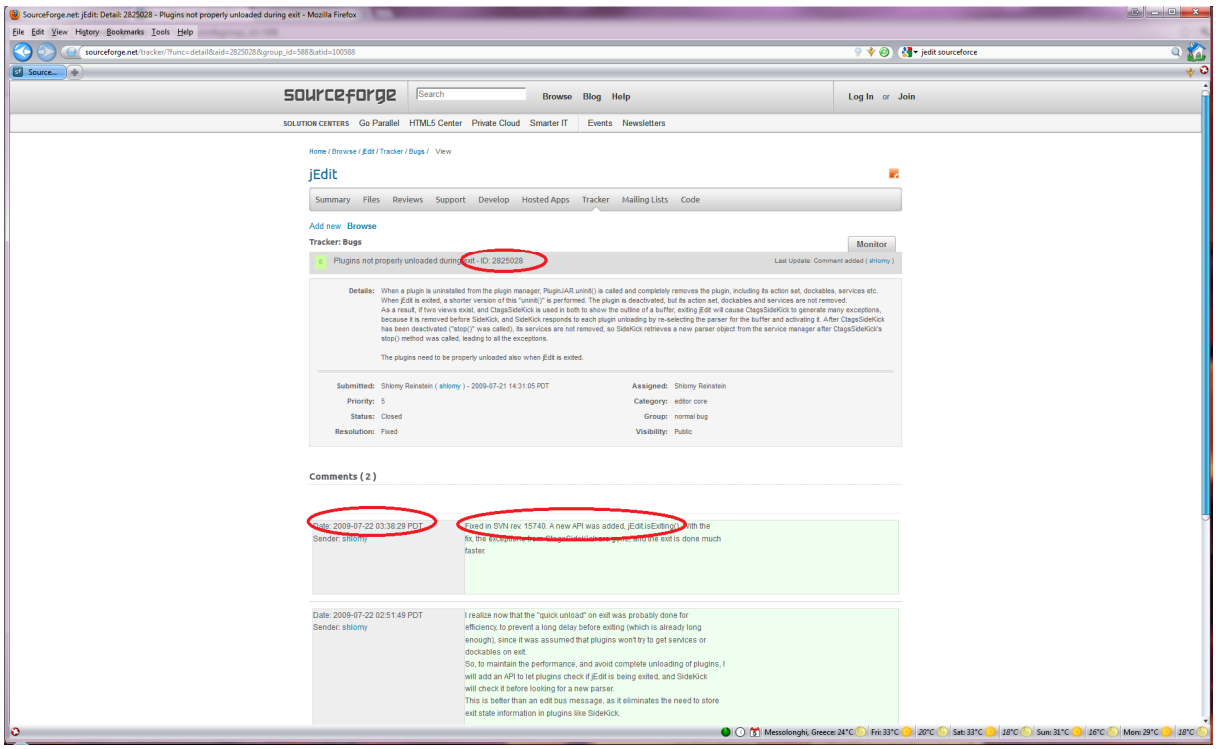


Εικόνα 4.2: Το Πρόγραμμα jEdit σε Λειτουργία

Για να υπάρχουν ιστορικά δεδομένα επιλέξαμε μια έκδοση του jEdit που να μην είναι σε χρήση. Η τρέχουσα έκδοση του jEdit είναι στην σειρά 4, οπότε επιλέξαμε την τελευταία έκδοση της προηγούμενης σειράς που ήταν η έκδοση 3.2, ώστε να υπάρχουν ολοκληρωμένα ιστορικά δεδομένα για τα αναφερθέντα σφάλματα και μην τροποποιείται πλέον ο πηγαίος κώδικας. Η αντιστοίχηση των σφαλμάτων με τις κλάσεις του πηγαίου κώδικα έγινε σε δυο διακριτά βήματα:

- **Από Αλλαγές Κώδικα σε Αναφορά Σφάλματος:** Από το ιστορικό αλλαγών στο σύστημα διαχείρισης κώδικα μελετήσαμε τα σχόλια. Ψάξαμε για μηνύματα που να περιέχουν σχετικές λέξεις όπως "bug" , "fixed" ή να υπάρχει κάποιος αριθμός (ώστε να μπορούμε να τον αντιπαραβάλουμε με το σύστημα διαχείρισης σφαλμάτων).

- **Από Αναφορά Σφάλματος σε Αλλαγή Κώδικα:** Δεν αναφέρονται όλα τα σφάλματα στο ιστορικό αλλαγών του πηγαίου κώδικα. Για να αντιστοιχίσουμε αυτά τα σφάλματα από το σύστημα αναφοράς σφαλμάτων στις κλάσεις του πηγαίου κώδικα ελέγξαμε τις αλλαγές που έγιναν σε κοντινό χρονικό διάστημα από το κλείσιμο του σφάλματος και που έγιναν από τον προγραμματιστή στον οποίο είχε ανατεθεί η επίλυση.



Εικόνα 4.3: Το Σύστημα Διαχείρισης Σφαλμάτων του Προγράμματος jEdit

Για το πρώτο βήμα χρησιμοποιήσαμε το πρόγραμμα ανοικτού κώδικα BugInfo [095] που συνδέεται με το σύστημα διαχείρισης κώδικα που χρησιμοποιεί το jEdit και ελέγχει τα μηνύματα με βάση τις προκαθορισμένες λέξεις που του έχουμε δώσει. Επειδή όμως δεν λειτούργησε χωρίς προβλήματα αναγκαστήκαμε να κάνουμε και χειροκίνητους ελέγχους ώστε να διορθώσουμε κάποιες λάθος αντιστοιχίσεις. Το δεύτερο βήμα έγινε εξ ολοκλήρου χειροκίνητα για όλους τους κωδικούς σφαλμάτων που δεν αντιστοιχίσαμε με το πρώτο βήμα. Στην εικόνα 4.3 δείχνουμε μια χαρακτηριστική εικόνα ενός σφάλματος που έχει καταχωρηθεί στο σύστημα και έχει διορθωθεί. Ο κωδικός του σφάλματος είναι ο 2825028 και διορθώθηκε από τον προγραμματιστή Shlomy Reinstein στις 22-07-2009 στην αναθεώρηση (revision) 15740. Ψάχνοντας στο σύστημα διαχείρισης SVN για την αναθεώρηση 15740 και τις κλάσεις που άλλαξε σε αυτή ο προγραμματιστής Shlomy Reinstein βρίσκουμε τις κλάσεις που είχαν σφάλμα. Όταν τελειώσαμε την αντιστοίχιση όλων των σφαλμάτων με τις κλάσεις της έκδοσης 3.2 του jEdit είχαμε ένα πίνακα με τρεις στήλες, το όνομα της κλάσης, το αριθμό τον σφαλμάτων και αν υπήρξε σφάλμα ή όχι. Στην συνέχεια χρησιμοποιήσαμε το εργαλείο ckjm extnted που επιλέξαμε στην προηγούμενη ενότητα και πήραμε σε ένα αρχείο κειμένου όλες τις μετρικές ανά κλάση. Αυτό το αρχείο χρειαζόταν επεξεργασία γιατί ανέφερε την κυκλωματική πολυπλοκότητα ανά συνάρτηση και όχι συνολικά. Έτσι για κάθε κλάση με βάση την έξοδο του ckjm υπολογίσαμε την μέγιστη και την μέση κυκλωματική πολυπλοκότητα της. Στο τέλος, συνδυάστηκαν όλα αυτά μαζί σε ένα αρχείο κειμένου σε δυο εκδόσεις, μια σε μορφή CSV για το R και μια σε μορφή ARFF για το WEKA.

4.4 Μοντέλα Στατιστικής Ανάλυσης

Αυτή η ενότητα έχει ως σκοπό να εκθέσει με συνοπτικό και επιστημονικό τρόπο στον αναγνώστη το θεωρητικό υπόβαθρο που είναι απαραίτητο για την κατανόηση των μοντέλων που θα κατασκευάσουμε με τις τεχνικές της στατιστικής ανάλυσης. Μια αναλυτική περιγραφή της γραμμικής και λογιστικής παλινδρόμησης μπορεί να βρεθεί σε συγγράμματα στατιστικής ή οικονομετρίας [076, 077, 078].

4.4.1 Γραμμική Παλινδρόμηση

Η απλή γραμμική παλινδρόμηση (univariate linear regression) χρησιμοποιείται όταν δυο μεταβλητές συσχετίζονται και συνδέονται με γραμμική σχέση. Έστω Χ και Υ είναι δυο μεταβλητές και (x_i, y_i) είναι τα δυνατά ζευγάρια που μπορούμε να δημιουργήσουμε. Συνήθως η Χ καλείται ανεξάρτητη ή επεξηγηματική (explanatory) μεταβλητή και η Υ καλείται συνήθως εξαρτημένη ή δεσμευμένη (response) μεταβλητή. Σκοπός της παλινδρόμησης είναι να διερευνηθεί η αλλαγή της ανεξάρτητης μεταβλητής σε σχέση με τις αλλαγές τιμών της εξαρτημένης μεταβλητής και πιο συγκεκριμένα να καθοριστούν οι συντελεστές αυτής της σχέσης. Το θεωρητικό γενικό μοντέλο για τον πληθυσμό (population regression line) είναι:

$$E(y|x) = \mu_{y|x} = \alpha + \beta x \quad (4.1)$$

Όμως πειραματικά οι παρατηρούμενες τιμές έχουν μια απόκλιση από την αναμενόμενη τιμή $E(y|x)$ την οποία ονομάζουμε ε_i και έτσι έχουμε ένα μοντέλο της μορφής:

$$y_i = \alpha + \beta x_i + \varepsilon_i \quad (4.2)$$

όπου y_i είναι η τιμή της εξαρτημένες μεταβλητής, x_i είναι η τιμή της ανεξάρτητης μεταβλητής, α είναι το σημείο τομής του άξονα της Υ από τη γραμμή παλινδρόμησης, β είναι η κλίση της γραμμής παλινδρόμησης και ε_i είναι το σφάλμα ή κατάλοιπο (residual). Το μοντέλο της εξίσωσης 4.2 στηρίζεται σε τέσσερις υποθέσεις:

1. Οι τιμές της μεταβλητής Υ δηλαδή τα y_i είναι ανεξάρτητα μεταξύ τους.

2. Για κάθε συγκεκριμένη τιμή της Χ δηλαδή για κάθε x_i αντιστοιχούν πολλές τιμές της Υ που ακολουθούν την κανονική κατανομή με $N\left(\mu_{y|x}, \sigma_{y|x}\right)$.

3. Ο μέσος της κάθε κανονικής κατανομής της Υ ισούται με $E\left(y\middle|x\right) = \mu_{y|x} = \alpha + \beta x$, δηλαδή όλοι οι μέσοι βρίσκονται σε μια ευθεία γραμμή που αποτελεί την γραμμή της παλινδρόμησης του πληθυσμού.

4. Ισχύει η ομοσκέδαση (homoscedacity) των καταλοίπων ε_i, δηλαδή η διακύμανση $\sigma_{y|x}$ παραμένει σταθερή για όλες τις τιμές της μεταβλητής Χ.

Την εκτίμηση των παραμέτρων α και β την συμβολίζουμε με ένα καπελάκι από πάνω δηλαδή $\hat{\alpha}$ και $\hat{\beta}$ αντίστοιχα. Μπορούν να εφαρμοστούν διάφορες αναλυτικές μέθοδοι για τον υπολογισμό τους όπως είναι των ελαχίστων τετραγώνων (least squares), της μέγιστης πιθανοφάνειας (maximum likehood) κλπ. Εμείς θα χρησιμοποιήσουμε την πρώτη μέθοδο που προσπαθεί να ελαχιστοποιήσει το άθροισμα των τετραγώνων από το σφάλμα ε_i, δηλαδή αν $\widehat{y_i}$ είναι η εκτίμηση της y_i τότε έχουμε:

$$\widehat{y_i} = \hat{\alpha} + \hat{\beta} x_i \quad (4.3)$$

Οπότε από την 4.3 προκύπτει ότι το εκτιμούμενο σφάλμα $\widehat{\varepsilon_i}$ είναι ίσο με:

$$\widehat{\varepsilon_i} = \left| y_i - \widehat{y_i} \right| \quad (4.4)$$

Άρα το άθροισμα των τετραγώνων όλων των σφαλμάτων $\widehat{\varepsilon_i}$ δίνεται από την συνάρτηση:

$$f\left(\hat{\alpha}, \hat{\beta}\right) = \sum \widehat{\varepsilon_i}^{\,2} = \sum \left| y_i - \widehat{y_i} \right|^2 = \sum \left| y_i - \left(\hat{\alpha} + \hat{\beta} x_i\right) \right|^2 \quad (4.5)$$

Για να ελαχιστοποιήσουμε την σχέση 4.5 αρκεί να πάρουμε τις μερικές παραγώγους της συνάρτησης $f\left(\hat{\alpha}, \hat{\beta}\right)$ για κάθε μια από τις εκτιμήσεις των συντελεστών, να θέσουμε τις υπολογισμένες παραγώγους ίσες με το μηδέν και να λύσουμε ένα γραμμικό σύστημα δυο εξισώσεων με δυο αγνώστους [078]:

$$\frac{\partial f\left(\hat{\alpha},\hat{\beta}\right)}{\partial \hat{\alpha}} = \frac{\partial\left(\sum\left(y_i-\left(\hat{\alpha}+\hat{\beta}x_i\right)\right)^2\right)}{\partial \hat{\alpha}} = 2\left(\sum\left(y_i-\hat{\alpha}-\hat{\beta}x_i\right)\right)\cdot\frac{\partial\left(-\hat{\alpha}\right)}{\partial \hat{\alpha}}$$

$$= 2\left(\sum\left(y_i-\hat{\alpha}-\hat{\beta}x_i\right)\right)\cdot(-1) = -2\sum y_i + 2n\hat{\alpha} + 2\hat{\beta}\sum x_i \quad (4.6)$$

$$\frac{\partial f\left(\hat{\alpha},\hat{\beta}\right)}{\partial \hat{\beta}} = \frac{\partial\left(\sum\left(y_i-\left(\hat{\alpha}+\hat{\beta}x_i\right)\right)^2\right)}{\partial \hat{\beta}} = 2\left(\sum\left(y_i-\hat{\alpha}-\hat{\beta}x_i\right)\right)\cdot\frac{\partial\left(-\hat{\beta}x_i\right)}{\partial \hat{\beta}}$$

$$= 2\left(\sum\left(y_i-\hat{\alpha}-\hat{\beta}x_i\right)\right)\cdot(-x_i) = -2\sum y_i x_i + 2\hat{\alpha}\sum x_i + 2\hat{\beta}\sum x_i^2 \quad (4.7)$$

Θέτουμε τις μερικές παραγώγους ίσες με το μηδέν, διαιρούμε με το -2 και έχουμε:

$$-2\sum y_i + 2n\hat{\alpha} + 2\hat{\beta}\sum x_i = 0 \Rightarrow \sum y_i - n\hat{\alpha} - \hat{\beta}\sum x_i = 0 \Rightarrow \sum y_i = n\hat{\alpha} + \hat{\beta}\sum x_i \quad (4.8)$$

$$-2\sum y_i x_i + 2\hat{\alpha}\sum x_i + 2\hat{\beta}\sum x_i^2 = 0 \Rightarrow$$

$$\sum y_i x_i - \hat{\alpha}\sum x_i - \hat{\beta}\sum x_i^2 = 0 \Rightarrow \sum y_i x_i = \hat{\alpha}\sum x_i + \hat{\beta}\sum x_i^2 \quad (4.9)$$

Οι εξισώσεις 4.8 και 4.9 είναι κανονικές εξισώσεις και λύνονται με την μέθοδο Cramer:

$$D_0 = \begin{vmatrix} \sum y_i & \sum x_i \\ \sum y_i x_i & \sum x_i^2 \end{vmatrix} = \sum y_i \sum x_i^2 - \sum y_i x_i \sum x_i \quad (4.10)$$

$$D_1 = \begin{vmatrix} n & \sum y_i \\ \sum x_i & \sum y_i x_i \end{vmatrix} = n\sum y_i x_i - \sum x_i \sum y_i \quad (4.11)$$

$$D = \begin{vmatrix} n & \sum x_i \\ \sum x_i & \sum x_i^2 \end{vmatrix} = n\sum x_i^2 - \left(\sum x_i\right)^2 \quad (4.12)$$

Άρα οι εκτιμήσεις των συντελεστών από τις 4.10, 4.11 και 4.12 προκύπτει ότι είναι:

$$\hat{\alpha} = \frac{D_0}{D} = \frac{\sum y_i \sum x_i^2 - \sum y_i x_i \sum x_i}{n\sum x_i^2 - \left(\sum x_i\right)^2} = \bar{y} - \hat{\beta}\bar{x} \quad (4.13)$$

$$\hat{\beta} = \frac{D_1}{D} = \frac{n\sum y_i x_i - \sum x_i \sum y_i}{n\sum x_i^2 - \left(\sum x_i\right)^2} \quad (4.14)$$

Οι εξισώσεις 4.13 και 4.14 δίνουν τις εκτιμήσεις των συντελεστών της αρχικής εξίσωσης 4.2, αλλά μας ενδιαφέρει να γνωρίσουμε και πόσο καλή πρόβλεψη μπορούμε να έχουμε από αυτές τις εκτιμήσεις. Αυτό μπορεί να απαντηθεί από τον συντελεστή προσδιορισμού που συμβολίζεται με R^2 και είναι ο λόγος της μεταβλητότητας της Υ που ερμηνεύεται από την παλινδρόμηση προς την συνολική μεταβλητότητα της Υ και δίνεται από την εξίσωση [077]:

$$R^2 = \frac{\sum\left(\hat{y}_i - \bar{y}\right)^2}{\sum\left(y_i - \bar{y}\right)^2} = 1 - \frac{\sum \varepsilon_i^2}{\sum\left(y_i - \bar{y}\right)^2} \quad (4.15)$$

Για να πραγματοποιήσουμε έλεγχο υποθέσεων και πάρουμε διαστήματα εμπιστοσύνης για τις εκτιμήσεις των παραμέτρων της γραμμικής παλινδρόμησης χρειαζόμαστε το τυπικό σφάλμα της κατανομής δειγματοληψίας του συντελεστή $\hat{\beta}$, όμως επειδή αυτό δεν είναι γνωστό χρησιμοποιούμε μια εκτίμηση από τα δεδομένα:

$$\sigma_{\hat{\beta}} = \frac{\sigma_\varepsilon}{\sqrt{\sum\left(x_i - \bar{x}\right)^2}} \quad (4.16)$$

Το $\sigma_{\hat{\beta}}^2$ ονομάζεται και τυπικό σφάλμα εκτίμησης του συντελεστή παλινδρόμησης. Για τον δίπλευρο έλεγχο έχουμε λοιπόν:

$$H_0 : \beta = \beta^*$$
$$H_1 : \beta \neq \beta^*$$

Όπου β^* είναι ο συντελεστής παλινδρόμησης του πληθυσμού. Ο έλεγχος γίνεται με το γνωστό κριτήριο t και $n-2$ βαθμούς ελευθερίας:

$$\left|t_{n-2}\right| = \frac{\hat{\beta} - \beta^*}{\sigma_{\hat{\beta}}} \quad (4.17)$$

Οπότε προκύπτει το διάστημα εμπιστοσύνης:

$$\hat{\beta} - \sigma_{\hat{\beta}} \cdot t_{n-2,a/2} < \beta < \hat{\beta} - \sigma_{\hat{\beta}} \cdot t_{n-2,a/2} \quad (4.18)$$

Όπου α είναι το επίπεδο σημαντικότητας και $n-2$ είναι οι βαθμοί ελευθερίας.

Η πολλαπλή γραμμική παλινδρόμηση (multivariate linear regression) μπορεί να θεωρηθεί ως επέκταση της απλή γραμμικής παλινδρόμησης που αναλύσαμε παραπάνω, όπου αντί για μία έχουμε δυο ή περισσότερες ανεξάρτητες μεταβλητές. Αν $k \geq 2$ τότε είναι:

$$E\left(y \middle| x_1, x_2, \ldots, x_k\right) = \mu_{y|x_1,x_2,\ldots,x_k} = \alpha + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_k x_k = a + \sum_{i=1}^{k} \beta_i x_i \quad (4.19)$$

Όμως πειραματικά οι παρατηρούμενες τιμές έχουν μια απόκλιση από την αναμενόμενη τιμή $E\left(y \middle| x_1, x_2, \ldots, x_k\right)$ την οποία ονομάζουμε ε_i και έτσι έχουμε ένα μοντέλο της μορφής:

$$y_i = \alpha + \beta_1 x_{i1} + \beta_2 x_{i2} + \ldots + \beta_k x_{ik} + \varepsilon_i = a + \sum_{m=1}^{k} \beta_m x_{im} + \varepsilon_i \quad (4.20)$$

Για την εξίσωση 4.20 ισχύουν ανάλογες υποθέσεις για αυτές που διατυπώσαμε για την εξίσωση της απλής παλινδρόμησης 4.2 και επιπλέον τη νέα υπόθεση ότι οι μεταβλητές $X_1, X_2, \ldots, X_k$ θα πρέπει να είναι ανεξάρτητες μεταξύ τους. Όπως και στην χρήση της απλής παλινδρόμησης μπορούμε να έχουμε ικανοποιητικές εκτιμήσεις των συντελεστών με την χρήση της μεθόδου των ελαχίστων τετραγώνων ή της μεθόδου της μέγιστης πιθανοφάνειας, μέσω της ελαχιστοποιήσεως του αθροίσματος των σφαλμάτων $\sum \hat{\varepsilon}_i^{\,2}$. Η διαδικασία εύρεσης διαστημάτων εμπιστοσύνης και ελέγχου υποθέσεων και έλεγχου υποθέσεων για τις παραμέτρους του πληθυσμού $\alpha, \beta_1, \beta_2, \ldots, \beta_k$ δεν διαφέρει από την αντίστοιχη διαδικασία που παρουσιάσαμε πιο πάνω για την απλή παλινδρόμηση. Για την επιλογή των πλέον κατάλληλων μεταβλητών χρησιμοποιούνται συνήθως τα στατιστικά μέτρα F και R^2 είτε με είσοδο μιας μεταβλητής κάθε φορά (stepwise forward) είτε με είσοδο όλων των μεταβλητών και αφαίρεση μιας μεταβλητής κάθε φορά (stepwise backward).

4.4.2 Λογιστική Παλινδρόμηση

Η μέθοδος της δυαδικής λογιστικής παλινδρόμησης χρησιμοποιείται ευρέως στην περίπτωση που η εξαρτημένη μεταβλητή μπορεί να κατηγοριοποιηθεί σε μια από δυο κατηγορίες. Ένα μεγάλο πλεονέκτημα της είναι ότι δεν απαιτεί τις υποθέσεις που αναφέραμε στην προηγούμενη ενότητα για την γραμμική παλινδρόμηση όπως είναι η γραμμική εξάρτηση των μεταβλητών, την ομοσκέδαση, την κανονική κατανομή των μεταβλητών κλπ Έστω λοιπόν η εξαρτημένη μεταβλητή Υ που έχει μόνο δυο πιθανές τιμές 0 (μπορεί να ερμηνευτεί σαν "όχι", αποτυχία κλπ) ή 1 (μπορεί να ερμηνευτεί σαν "ναι", επιτυχία κλπ) και οι ανεξάρτητες μεταβλητές $X_1, X_2, ..., X_k$. Τυπικά τα αποτελέσματα Y_i πρέπει να ακολουθούν την κατανομή Bernoulli όπου το κάθε αποτέλεσμα καθορίζεται από μια άγνωστη πιθανότητα p_i που είναι συγκεκριμένη για κάθε συγκεκριμένο αποτέλεσμα αλλά εξαρτάται από τις ανεξάρτητες μεταβλητές [076]:

$$E\left[y_i = 1 \middle| x_{i1}, x_{i2}, ..., x_{ik} \right] = p_i \quad (4.21)$$

Η βασική ιδέα της λογιστικής παλινδρόμησης είναι να χρησιμοποιήσουμε τους μηχανισμούς που αναπτύχθηκαν για την γραμμική παλινδρόμηση (και παρουσιάσαμε στην προηγούμενη ενότητα) για την μοντελοποίηση της πιθανότητας p_i χρησιμοποιώντας μια γραμμική συνάρτηση πρόβλεψης έστω λοιπόν ότι αυτή είναι η συνάρτηση $f(i)$, για την είσοδο i έχουμε:

$$f(i) = a + \beta_1 x_{i1} + \beta_2 x_{i2} + ... + \beta_k x_{ik} = a + \sum_{m=1}^{k} \beta_m x_{im} \quad (4.22)$$

όπου $\alpha, \beta_1, \beta_2, ..., \beta_k$ είναι οι συντελεστές της παλινδρόμησης. Τότε το μοντέλο της λογιστικής παλινδρόμησης είναι η πιθανότητα ένα συγκεκριμένο αποτέλεσμα να είναι συσχετισμένο με την γραμμική συνάρτηση πρόβλεψης 4.22, δηλαδή:

$$\text{logit}\left(E\left[y_i = 1 \middle| x_{i1}, x_{i2}, ..., x_{ik} \right] \right) = \text{logit}\left(p_i \right) = \ln\left(\frac{p_i}{1 - p_i} \right) = a + b_1 x_{i1} + b_2 x_{i2} + ... + b_k x_{ik} = a + \sum_{m=1}^{k} \beta_m x_{im} \quad (4.23)$$

Παίρνοντας τον αντιλογάριθμο της 4.23 καταλήγουμε στην γνωστή λογιστική συνάρτηση:

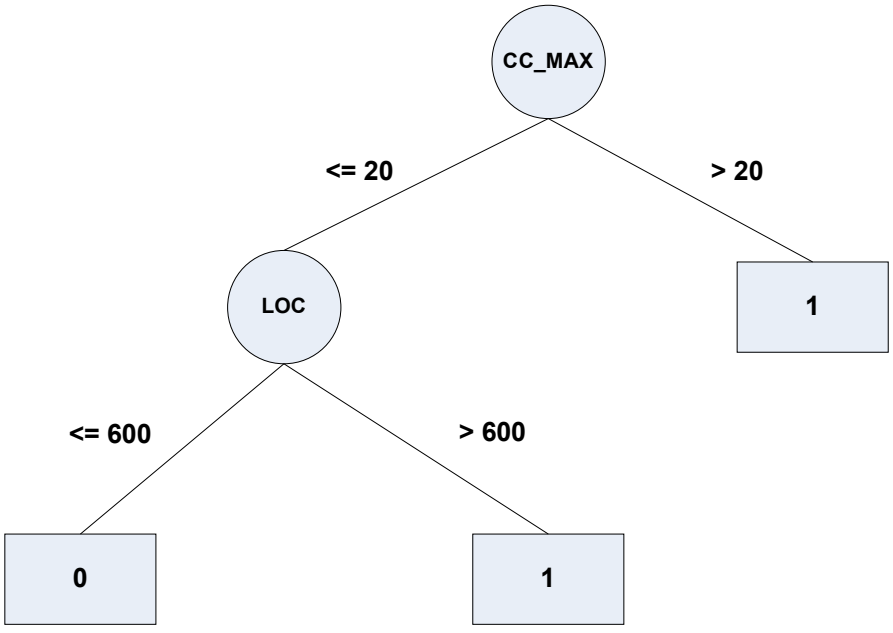
$$p\left(y_i = 1 \middle| x_{i1}, x_{i2}, ..., x_{ik} \right) = \frac{e^{a + b_1 x_{i1} + b_2 x_{i2} + ... + b_k x_{ik}}}{1 + e^{a + b_1 x_{i1} + b_2 x_{i2} + ... + b_k x_{ik}}} = \frac{e^{a + \sum_{m=1}^{k} \beta_m x_{im}}}{1 + e^{a + \sum_{m=1}^{k} \beta_m x_{im}}} \quad (4.24)$$

4.5 Μοντέλα Μηχανικής Μάθησης

Υπάρχει πλούσια βιβλιογραφία για την μηχανική μάθηση και τους σχετικούς αλγόριθμους [079, 080, 081, 082, 083]. Από όλους αυτούς το δέντρο απόφασης επιλέχτηκε γιατί μπορεί να δώσει μια αναπαράσταση από την οποία μπορούν να εξαχθούν κανόνες που να είναι εύκολα κατανοητοί από τον άνθρωπο. Το τεχνητό νευρωνικό δίκτυο έχει χρησιμοποιηθεί με επιτυχία σε πάρα πολλές εφαρμογές πρόβλεψης και είναι ένας από τους πιο χαρακτηριστικούς αλγόριθμους μηχανικής μάθησης. Σε αυτή την ενότητα γίνεται μια σύντομη παρουσίαση αυτών των τεχνικών, ώστε ο αναγνώστης να κατανοήσει την γενικότερη λειτουργία τους και να αναπτύξει το υπόβαθρο που απαιτείται για το επόμενο κεφάλαιο που παρουσιάζονται τα αποτελέσματα τους.

4.5.1 Δέντρο Απόφασης

Μεταξύ των αλγόριθμων μηχανικής μάθησης εξέχουσα θέση έχουν οι αλγόριθμοι για την επαγωγική κατασκευή δέντρων απόφασης με σημαντικότερους εκπρόσωπους της κατηγορίας αυτής τον ID3 [082], τον C4.5 [008] και τον C5.0. Ένα δέντρο απόφασης είναι ουσιαστικά ένα διάγραμμα ροής με δεντρική μορφή, όπου κάθε εσωτερικός κόμβος (nonleaf node) αποτελεί μια ερώτηση για ένα γνώρισμα, κάθε διακλάδωση (branch) αναπαριστά το αποτέλεσμα στην ερώτηση και κάθε φύλλο (terminal node) επιλέγει την κατηγορία. Η διαδικασία ξεκινάει από τον κόμβο που βρίσκεται πιο ψηλά και ονομάζεται ρίζα του δέντρου (root). Ένα χαρακτηριστικό αλλά απλό ταυτόχρονα παράδειγμα δέντρου απόφασης παρουσιάζεται στο σχήμα 4.1.



Σχήμα 4.1: Παράδειγμα Δέντρου Απόφασης, 0 η Κλάση δεν έχει Σφάλμα, 1 Υπάρχει Σφάλμα στην Κλάση

Για την εύρεση των κριτηρίων σε κάθε εσωτερικό κόμβο χρησιμοποιούνται διάφοροι ευρετικοί κανόνες που διαχωρίζουν τα δεδομένα εισόδου σε διαφορετικές κλάσεις επιλέγοντας το χαρακτηριστικό και το κριτήριο διαχωρισμού π..χ. για την ρίζα του δέντρου στο σχήμα 4 το χαρακτηριστικό είναι η μετρική της μέγιστης πολυπλοκότητας CC_MAX και το κριτήριο αν είναι μικρότερη ή μεγαλύτερη του 20. Μια μέθοδος για την επιλογή του χαρακτηριστικού σε κάθε εσωτερικό κόμβο είναι αυτό κέρδους της πληροφορίας (information gain). Πιο τυπικά, έστω D όλα τα δεδομένα εισόδου δηλαδή αποτελεί το σετ της εκπαίδευσης. Αν έχουμε m ξεχωριστές τιμές για την εξαρτημένη μεταβλητή, έχουμε m διαφορετικές κλάσεις C_i (για $i = 1, 2, \ldots, m$). Ορίζουμε ως $C_{i,D}$ τα δεδομένα από το D που ανήκουν στην κλάση C_i, $\left|C_{i,D}\right|$ τον αριθμό των δεδομένων που ανήκουν στην κλάση C_i και $\left|D\right|$ τον αριθμό των δεδομένων D. Τότε η αναμενόμενη πληροφορία (expected information) που χρειαζόμαστε δίνεται από τον τύπο [080]:

$$\Pi\lambda\eta\rho οφορί\alpha\left(D\right) = -\sum_{i=1}^{m} p_i \log_2\left(p_i\right) \quad (4.25)$$

όπου p_i είναι η πιθανότητα ένα τυχαίο δεδομένο εισόδου του D να ανήκει στην κλάση C_i και υπολογίζεται ως εξής:

$$p_i = \frac{\left|C_{i,D}\right|}{\left|D\right|} \quad (4.26)$$

Στην 4.25 χρησιμοποιούμε λογάριθμο με βάση το 2 γιατί η πληροφορία κωδικοποιείται σε δυαδικά ψηφία (bit) και εκφράζει την μέση πληροφορία που χρειαζόμαστε για να αναγνωρίσουμε την κλάση σε μία από τις εισόδους του D. Σε αυτό το σημείο η πληροφορία που έχουμε βασίζεται μόνο στην αναλογία κάθε κλάσης C_i στα δεδομένα εισόδου D και είναι γνωστή με την ονομασία ως εντροπία (entropy) του D. Ας υποθέσουμε τώρα ότι έχουμε ένα υποψήφιο γνώρισμα για τον διαχωρισμό, έστω T που έχει n διακριτές τιμές στο σετ εκπαίδευσης $\{t_1, t_2, \ldots, t_n\}$. Τότε μπορούμε να διαχωρίσουμε τα δεδομένα D σε n υποσύνολα $\{D_1, D_2, \ldots, D_n\}$, όπου D_i είναι όλα τα δεδομένα εισόδου του D που έχουν τιμή t_i στο χαρακτηριστικό T. Κάθε υποσύνολο D_i αντιστοιχεί και σε ένα κλαδί στο κόμβο N. Το ερώτημα που θα πρέπει να απαντηθεί τώρα είναι πόση πληροφορία χρειαζόμαστε αφού διαχωρίσουμε τα δεδομένα εισόδου με βάση το χαρακτηριστικό T. Όσο μικρότερη η πληροφορία που χρειαζόμαστε μετά τον διαχωρισμό τόσο και μεγαλύτερη είναι η καθαρότητα των υποσυνόλων δηλαδή τόσο

λιγότερα δεδομένα ανήκουν σε διαφορετικές κλάσεις έχουν. Μετά τον διαχωρισμό η απαιτούμενη πληροφορία δίνεται από τον παρακάτω τύπο:

$$\Pi\lambda\eta\rho οφορία_T\left(D\right)=-\sum_{i=1}^{n}\left(\frac{\left|D_i\right|}{\left|D\right|}\cdot\Pi\lambda\eta\rho οφορία\left(D_i\right)\right) \quad (4.27)$$

Το κέρδος της πληροφορίας ορίζεται ως η διαφορά μεταξύ της πληροφορίας που αρχικά χρειαζόμαστε για την κατηγοριοποίηση των κλάσεων (δηλαδή αυτή που στηριζόταν αποκλειστικά στην αναλογία των κλάσεων στα δεδομένα εισόδου) και αυτής που απαιτείται μετά τον διαχωρισμό (με βάση το χαρακτηριστικό Τ):

$$\text{Κέρδος Πληροφορίας}\left(\text{Τ}\right)=\Pi\lambda\eta\rho οφορία\left(D\right)-\Pi\lambda\eta\rho οφορία_T\left(D\right) \quad (4.28)$$

Με άλλα λόγια η σχέση 4.28 μας φανερώνει πόσο κερδίσαμε σε όρους απαιτούμενης πληροφορίας μετά τον διαχωρισμό με βάση το χαρακτηριστικό Τ. Θα πρέπει να επαναλάβουμε την διαδικασία για όλα τα χαρακτηριστικά των δεδομένων εισόδου D και το χαρακτηριστικό με το υψηλότερο κέρδος πληροφορίας είναι αυτό που τελικά θα επιλεγεί για τον κόμβο Ν. Η παραπάνω διαδικασία συνεχίζεται για κάθε επόμενο κόμβο του δέντρου μέχρι είτε όλα τα δεδομένα σε ένα κόμβο να ανήκουν στην ίδια κλάση είτε να μην υπάρχουν άλλα δεδομένα για εξέταση. Ένα πρόβλημα που υπάρχει με την χρησιμοποίηση του χαρακτηριστικού του κέρδος της πληροφορίας είναι ότι μεροληπτεί υπέρ των χαρακτηριστικών που έχουν πολλά πιθανά αποτελέσματα σε σχέση με αυτά που έχουν λιγότερα [081]. Ο αλγόριθμος C4.5 χρησιμοποιεί μια επέκταση του κέρδους της πληροφορίας που ονομάζεται αναλογία κέρδους πληροφορίας και έχει ως στόχο έχει να παρακάμψει το συγκεκριμένο πρόβλημα. Αρχικά ορίζεται η πληροφορία διαχωρισμού (split information) για το χαρακτηριστικό Τ:

$$\Pi\lambda\eta\rho οφορία\Delta\iota\alpha\chi\omega\rho ισμού_T\left(D\right)=-\sum_{i=1}^{n}\left(\frac{\left|D_i\right|}{\left|D\right|}\cdot\log_2\left(\frac{\left|D_i\right|}{\left|D\right|}\right)\right) \quad (4.29)$$

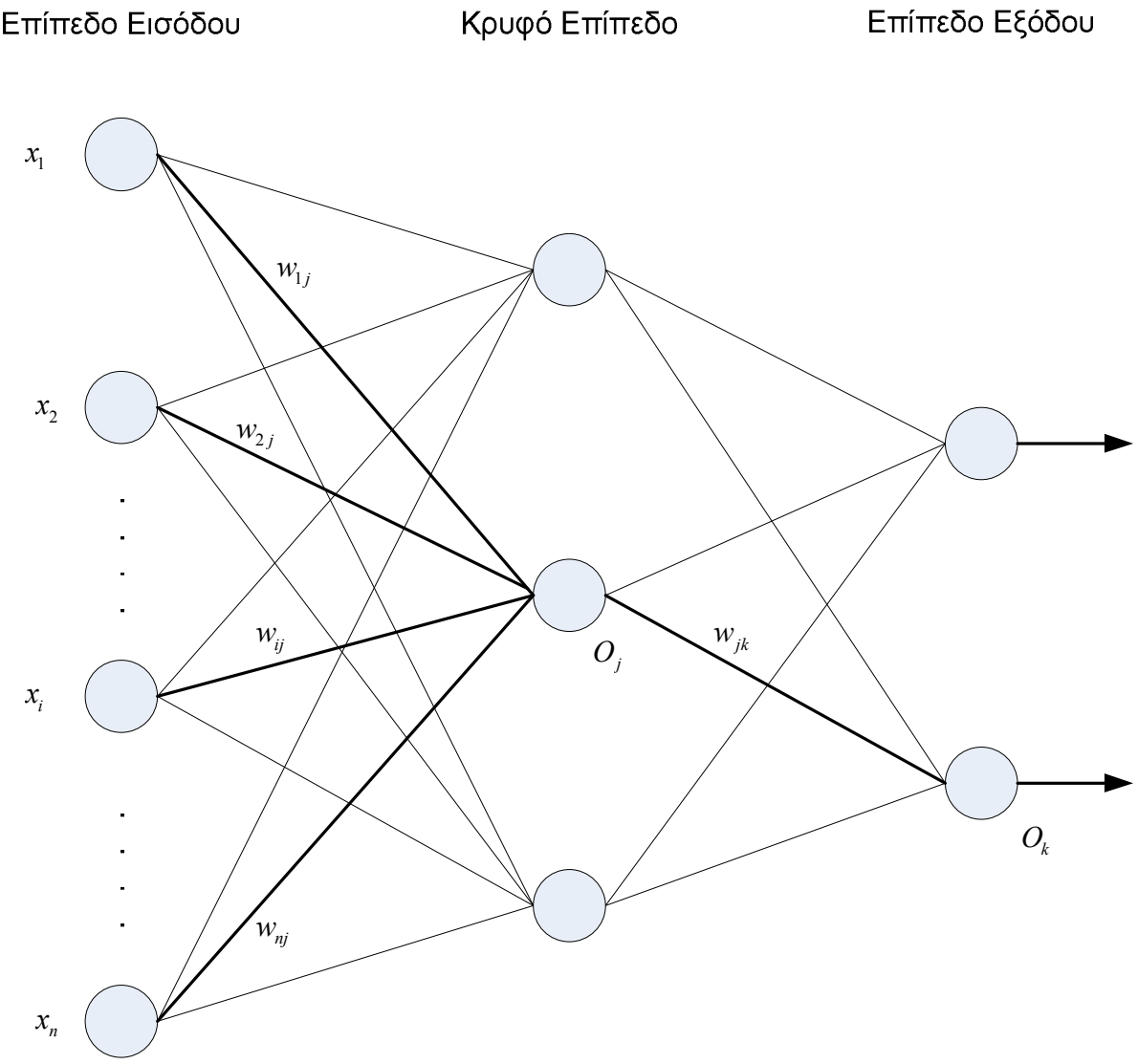
Θα επιλέξουμε το χαρακτηριστικό με την μεγαλύτερη αναλογία κέρδους πληροφορίας και χρησιμοποιώντας τον τύπο 4.29 μπορούμε να ορίσουμε αυτή την αναλογία ως εξής [080]:

$$\text{Αναλογία Κέρδους Πληροφορίας}\left(\text{Τ}\right)=\frac{\text{Κέρδος Πληροφορίας}\left(\text{Τ}\right)}{\Pi\lambda\eta\rho οφορία\Delta\iota\alpha\chi\omega\rho ισμού_T\left(D\right)} \quad (4.30)$$

4.5.2 Τεχνητό Νευρωνικό Δίκτυο

Το πεδίο των τεχνητών νευρωνικών δικτύων ξεκίνησε από τους ψυχολόγους και νευροβιολόγους που επιχείρησαν να αναπτύξουν και να ελέγξουν υπολογιστικά ανάλογα των νευρώνων [083]. Ένα νευρωνικό δίκτυο είναι ουσιαστικά ένα σύνολο από μονάδες εισόδου/εξόδου όπου σε κάθε σύνδεση έχει αντιστοιχιστεί ένα βάρος. Κατά την φάση της εκπαίδευσης, το δίκτυο προσαρμόζει αυτά τα βάρη, έτσι ώστε να μπορεί να προβλέπει καλύτερα την κατηγορία που ανήκει κάθε είσοδος που ελέγχει. Συνήθως χρειάζονται αρκετό χρόνο για να εκπαιδευτούν και απαιτούν αρκετές παραμέτρους που καθορίζονται εμπειρικά όπως είναι η τοπολογία του δικτύου. Μεταξύ των πλεονεκτημάτων τους συγκαταλέγονται η μεγάλη ανεκτικότητα που έχουν στα δεδομένα που περιέχουν θόρυβο και στο γεγονός ότι μπορούν να χρησιμοποιηθούν όταν δεν έχουμε μεγάλη γνώση για τις σχέσεις μεταξύ των χαρακτηριστικών και των κλάσεων που θέλουμε να κατηγοριοποιήσουμε [082]. Υπάρχουν αρκετά διαφορετικά είδη νευρωνικών δικτύων και αρκετοί διαφορετικοί αλγόριθμοι εκπαίδευσης τους. Στην παρούσα ενότητα θα ασχοληθούμε με μια σύντομη παρουσίαση των πολυεπίπεδων τεχνητών νευρωνικών δικτύων αισθητήρα (multilayer perceptron) που εκπαιδεύονται με τον αλγόριθμο της πίσω διάδοσης του λάθους (error back propagation algorithm).

Ένα δίκτυο με εμπρόσθια τροφοδότηση (feed forward) αποτελούμενο από ένα επίπεδο εισόδου, ένα κρυφό επίπεδο και ένα επίπεδο εξόδου παρουσιάζεται στο σχήμα 4.2. Ονομάζεται με εμπρόσθια τροφοδότηση γιατί κανένα από τα βάρη δεν γυρνάει πίσω σε μια μονάδα εισόδου ή σε μια μονάδα εξόδου προηγούμενου επιπέδου. Επίσης, είναι πλήρως συνδεδεμένο (fully connected) αφού κάθε μονάδα ενός επιπέδου παρέχει είσοδο σε κάθε μονάδα του επόμενου επιπέδου. Κάθε μονάδα εξόδου παίρνει είσοδο ένα σταθμισμένο άθροισμα των εξόδων από όλες τις μονάδες του προηγούμενου επιπέδου και εφαρμόζει μια μη γραμμική συνάρτηση ενεργοποίησης (activation function) στη σταθμισμένη είσοδο. Δηλαδή τα πολυεπίπεδα τεχνητά νευρωνικά δίκτυα με εμπρόσθια τροφοδότηση μοντελοποιούν την πρόβλεψη της κατηγορίας ως ένα μη γραμμικό συνδυασμό των εισόδων. Από στατιστική άποψη δηλαδή μπορούμε να πούμε ότι πραγματοποιούν μια μη γραμμική παλινδρόμηση [079]. Ο αλγόριθμος της πίσω διάδοσης του λάθους είναι μια αλληλεπιδραστική διαδικασία όπου επεξεργάζεται το σύνολο των δεδομένων εισόδου και συγκρίνει την πρόβλεψη του δικτύου με την παρατηρούμενη τιμή που αποτελεί και την τιμή στόχο. Για κάθε είσοδο εκπαίδευσης, τα βάρη αναπροσαρμόζονται ώστε να ελαχιστοποιηθεί το μέσο τετραγωνικό σφάλμα (mean squared error) μεταξύ της πρόβλεψης και του τιμής στόχου. Αυτές οι αναπροσαρμογές γίνονται από το επίπεδο εξόδου μέσω κάθε κρυμμένου επιπέδου μέχρι το πρώτο κρυμμένο επίπεδο.



Σχήμα 4.2: Παράδειγμα Τεχνητού Νευρωνικού Δικτύου Αισθητήρα με Εμπρόσθια Τροφοδότηση

Τα βήματα της πίσω διάδοσης του λάθους σκιαγραφούνται στην συνέχεια [079, 080, 082]:

Αρχικοποίηση Βαρών: Τα βάρη του δικτύου αρχικοποιούνται σε μικρούς τυχαίους αριθμούς συνήθως μέσα στο εύρος -1 έως 1. Κάθε μονάδα έχει ένα κατώφλι που το συμβολίζουμε με θ.

Εμπρόσθια Διάδοση Εισόδων: Οι είσοδοι περνάνε μέσω των μονάδων εισόδου χωρίς μεταβολές δηλαδή για την μονάδα εισόδου j η έξοδος O_j είναι ίση με την τιμή εισόδου I_j. Μετά η είσοδος και η έξοδος σε κάθε μονάδα των κρυφών επιπέδων ή του επιπέδου εξόδου υπολογίζεται. Η καθαρή είσοδος σε κάθε μονάδα των κρυφών επιπέδων ή του επιπέδου εξόδου υπολογίζεται σαν γραμμικός συνδυασμός των εισόδων τους:

$$I_j = \sum_i w_{ij} O_i + \theta_j \quad (4.31)$$

όπου j είναι η μονάδα του κρυφού επιπέδου ή του επιπέδου εξόδου, w_{ij} είναι το βάρος της σύνδεσης της μονάδας i από το προηγούμενο επίπεδο με την μονάδα j, O_i είναι η έξοδος της μονάδας i από το προηγούμενο επίπεδο και θ_j είναι το κατώφλι της μονάδας j. Στη συνέχεια κάθε μονάδα στα κρυφά επίπεδα ή στο επίπεδο εξόδου παίρνει την καθαρή είσοδο I_j και την εφαρμόζει στην συνάρτηση ενεργοποίησης για την παραγωγή της εξόδου O_j. Μια συνάρτηση που χρησιμοποιείται συχνά σαν συνάρτηση ενεργοποίησης είναι η λογιστική και στην περίπτωση που την υιοθετήσουμε η έξοδος της μονάδας j υπολογίζεται ως εξής:

$$O_j = \frac{1}{1+e^{-I_j}} \quad (4.32)$$

Στη συνέχεια υπολογίζονται οι έξοδοι για όλες τις μονάδες κάθε κρυφού επιπέδου μέχρι να φτάσουμε στο επίπεδο της εξόδου που δίνει και την πρόβλεψη του δικτύου.

Πίσω Διάδοση του Λάθους: Το λάθος μεταδίδεται προς τα πίσω με την ανανέωση των βαρών και των κατωφλιών σε κάθε μονάδα για να αντικατοπτρίζουν το σφάλμα στην πρόβλεψη του δικτύου. Για την μονάδα j στο επίπεδο εξόδου, το σφάλμα E_j υπολογίζεται ως εξής:

$$E_j = O_j\left(1-O_j\right)\left(T_j - O_j\right) \quad (4.33)$$

όπου το O_j είναι η πραγματική έξοδος της μονάδας j και το T_j είναι η έξοδος στόχος για την μονάδα j. Για να υπολογίσουμε το λάθος μιας μονάδας j που ανήκει σε κάποιο κρυφό επίπεδο του δικτύου έχουμε:

$$E_j = O_j\left(1-O_j\right)\sum_k E_k w_{jk} \quad (4.34)$$

όπου w_{jk} είναι το βάρος της σύνδεσης από την μονάδα j με την μονάδα k στο επόμενο υψηλότερο επίπεδο και E_k είναι το λάθος της μονάδας k. Στη συνέχεια τα βάρη και τα κατώφλια ανανεώνονται. Τα νέα βάρη υπολογίζονται από τις παρακάτω εξισώσεις:

$$\Delta w_{ij} = l \cdot E_j \cdot O_i \quad (4.35)$$

$$w_{ij}^{new} = w_{ij} + \Delta w_{ij} \quad (4.36)$$

όπου l είναι ο ρυθμός μάθησης (learning rate) που συνήθως παίρνει τιμές μεταξύ 0 και 1. Είναι μια σταθερά που χρησιμοποιείται για να μην σταματήσει ο αλγόριθμος σε κάποιο τοπικό ελάχιστο του χώρου των επιλογών και βοηθάει στην εύρεση του καθολικού ελάχιστου. Αν ο ρυθμός μάθησης l είναι πολύ μεγάλος τότε μπορεί να έχουμε λύση σε τοπικό ελάχιστο και αν είναι πολύ μικρός τότε η εκπαίδευση του νευρωνικού δικτύου θα γίνεται με πολύ μικρά βήματα. Τα κατώφλια ανανεώνονται με βάση τις παρακάτω εξισώσεις:

$$\Delta\theta_j = l \cdot E_j \quad (4.37)$$

$$\theta_j^{new} = \theta_j + \Delta\theta_j \quad (4.38)$$

Να σημειώσουμε ότι η ανανέωση των βαρών και των κατωφλιών μπορεί να γίνεται με την εισαγωγή κάθε διαφορετικής εισόδου στο δίκτυο ή εναλλακτικά οι διαφορές μπορούν να αποθηκευτούν σε προσωρινές μεταβλητές έτσι ώστε η ανανέωση στα βάρη και τα κατώφλια να γίνει συγκεντρωτικά αφού χρησιμοποιηθούν όλοι οι είσοδοι του σετ εκπαιδεύσεως. Ανεξάρτητα από τον τρόπο ανανέωσης των βαρών, όταν τελειώσουν όλοι οι είσοδοι του σετ εκπαιδεύσεως τελειώνει μια εποχή εκπαίδευσης (epoch training). Θεωρητικά από το μαθηματικό υπόβαθρο του αλγόριθμου της πίσω διάδοσης του λάθους η ανανέωση ανά εποχές είναι η σωστή στρατηγική, όμως πολλά πακέτα λογισμικού χρησιμοποιούν την ανανέωση ανά είσοδο γιατί στην πράξη τείνει να δίνει περισσότερο ακριβή αποτελέσματα [079].

Συνθήκες Τερματισμού Εκπαίδευσης: Ο αλγόριθμος της πίσω διάδοσης του λάθους τερματίζει όταν τουλάχιστον μια από τις παρακάτω συνθήκες ικανοποιηθεί:

- Όλες οι μεταβολές στα βάρη Δw_{ij} για την προηγούμενη εποχή εκπαίδευσης ήταν μικρότερες από μια προκαθορισμένη τιμή.

- Το ποσοστό των δεδομένων εισόδου του σετ εκπαίδευσης που κατηγοριοποιήθηκαν σωστά είναι μεγαλύτερη από κάποιο προκαθορισμένο ποσοστό.

- Το συνολικό σφάλμα για όλες τις εισόδους του σετ εκπαίδευσης να είναι μικρότερο από ένα προκαθορισμένο όριο.

- Ο προκαθορισμένος αριθμός εποχών εκπαίδευσης τελειώσει.

4.6 Επιλογή Εργαλείων για Μοντελοποίηση

Σε οποιοδήποτε είδος έρευνας εφαρμόζει μοντελοποίηση τα βοηθητικά εργαλεία που θα χρησιμοποιηθούν είναι ένας πολύ σημαντικός παράγοντας και απαιτείται ιδιαίτερη προσοχή κατά την επιλογή τους. Η αναζήτηση των κατάλληλων εργαλείων έγινε ανάμεσα στα προγράμματα που διατίθενται είτε ήταν δωρεάν είτε ήταν ανοικτού κώδικα. Η τελική επιλογή έγινε με κριτήριο κυρίως να υποστηρίζουν για την στατιστική ανάλυση τα μοντέλα που παρουσιάσαμε στην ενότητα 4.4 και για τις τεχνικές μηχανικής μάθησης αυτές που παρουσιάσαμε στην ενότητα 4.5. Για την στατιστική ανάλυση επιλέξαμε το πρόγραμμα R [084] και για την μηχανική μάθηση το WEKA [070].

Το R ξεκίνησε το 1992 από τους Ross Ihaka και Robert Gentleman στο πανεπιστήμιο του Auckland ως μια προσπάθεια να δημιουργηθεί μια γλώσσα που να υιοθετεί την σύνταξη της εμπορικής γλώσσας στατιστικής ανάλυσης S που αναπτύχθηκε στα εργαστήρια της Bell [085]. Το 1994 ολοκληρώθηκε η πρώτη έκδοση του και η διανομή του έγινε μέσω της GPL άδειας. Σύντομα εξελίχθηκε σε μια από τις πιο δημοφιλείς πλατφόρμες για ανάλυση δεδομένων. Είναι διαθέσιμο για όλα τα γνωστά λειτουργικά συστήματα συμπεριλαμβανομένων των Windows, Mac OS X και Linux. Μάλιστα, τα τελευταία πέντε χρόνια πολλά ερευνητικά ιδρύματα, μεγάλες εταιρίες και πανεπιστήμια έχουν επιλέξει το R ως κύριο εργαλείο στατιστικής ανάλυσης [086]. Βρίσκεται σε συνεχή ενημέρωση, με νέες λειτουργίες να προστίθενται καθημερινά. Το ενσωματωμένο σύστημα βοήθειας είναι αρκετά λεπτομερές, έχει αναφορές και παραδείγματα για κάθε συνάρτηση. Επιπλέον, υποστηρίζεται από μια μεγάλη κοινότητα που αποτελείται τόσο από ερευνητές δεδομένων όσο και προγραμματιστές που προσφέρουν βοήθεια και συμβουλές στους χρήστες. Είναι ελεύθερα διαθέσιμο από το δικτυακό τόπο Comprehensive R Archive Network [087]. Επιπλέον των αρχείων της βασικής εγκατάστασης του R, σε αυτό το δικτυακό τόπο υπάρχουν διαθέσιμα περισσότερα από 2.500 προαιρετικά πακέτα που επεκτείνουν κατά ένα πολύ μεγάλο βαθμό τις διαθέσιμες συναρτήσεις.

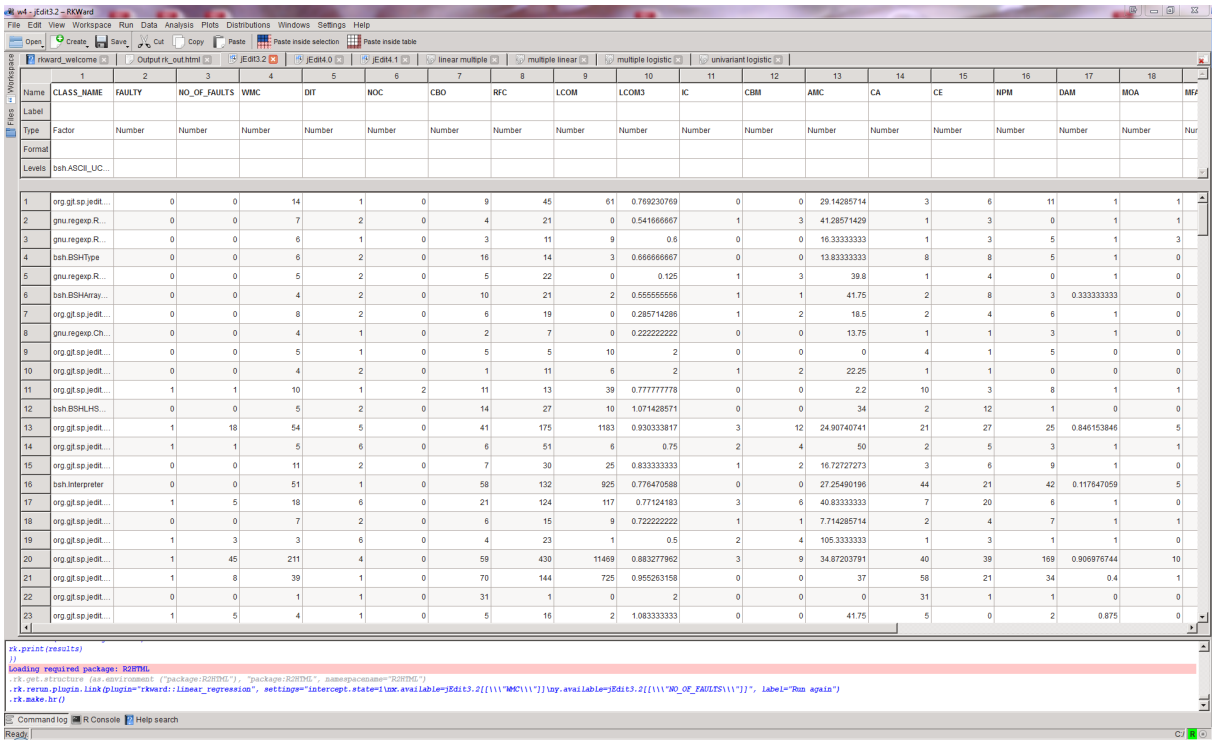
Μπορεί να σταθεί επάξια στα εμπορικά προγράμματα στατιστικής ανάλυσης και μάλιστα σε κάποια σημεία βρίσκεται και σε πλεονεκτική θέση, όπως:

- **Λειτουργίες:** Υπάρχουν χιλιάδες αλγόριθμοι δεδομένων και στατιστικής ανάλυσης. Κανένα εμπορικό πρόγραμμα δεν φτάνει καν κοντά την λειτουργικότητα που μπορεί να έχει κάποιος κατεβάζοντας τα προαιρετικά πακέτα του R[086].

- **Κοινότητα:** Υπάρχουν αρκετές χιλιάδες χρήστες του R παγκοσμίως και συνεχώς προσθέτονται νέοι. Έτσι, υπάρχει μια πολύ μεγάλη βάση χρηστών που είναι πρόθυμοι να βοηθήσουν και υπάρχει άφθονο βοηθητικό υλικό στο διαδίκτυο [085].

- **Απόδοση:** Έχει ελάχιστες υπολογιστικές απαιτήσεις και η απόδοση του είναι συγκρίσιμη ή και καλύτερη από τα περισσότερα εμπορικά πακέτα στατιστικής ανάλυσης [088].

Στα κυριότερα μειονεκτήματα του συγκαταλέγεται το γεγονός ότι για τους νέους χρήστες είναι μάλλον δύσκολο να εξοικειωθούν μαζί του και μπορεί να απογοητευτούν εύκολα στην αρχή. Από την εμπειρία μας με το R αυτό συμβαίνει κυρίως για δυο λόγους. Πρώτον, δεν παρέχει κάποια γραφική διεπαφή όπως συμβαίνει για την πλειοψηφία των σχετικών προγραμμάτων αλλά όλες οι λειτουργίες πραγματοποιούνται μόνο από τη γραμμή εντολών. Δεύτερον, δεν είναι απλά ένα πρόγραμμα για στατιστική ανάλυση αλλά μια ολοκληρωμένη αντικειμενοστρεφής γλώσσα προγραμματισμού για στατιστικούς υπολογισμούς και άρα απαιτεί ο χρήστης να γνωρίζει τουλάχιστον βασικές γνώσεις προγραμματισμού. Ευτυχώς, υπάρχουν διάφορα γραφικά περιβάλλοντα που μπορούν να εγκατασταθούν πάνω στο R και προσφέρουν διευκολύνσεις όπως το να εμφανίζουν την σύνταξη των εντολών κατά την πληκτρολόγηση τους ή να δείχνουν με γραφικό τρόπο όλα τα αντικείμενα που είναι ενεργά ή έχει δημιουργήσει ο χρήστης. Στην εικόνα 4.4 παρουσιάζουμε το ενδιάμεσο RKWard [147] που επιλέξαμε να χρησιμοποιήσουμε.



Εικόνα 4.4: Το Γραφικό Πρόσθετο RKWard για το Πρόγραμμα Στατιστικής Ανάλυσης R

Το WEKA [080] ξεκίνησε από το πανεπιστήμιο του Waikato το 1993 ως αποτέλεσμα μιας επιχορήγησης για την ανάπτυξη κουλτούρας σχετική με έρευνα πάνω σε θέματα μηχανικής μάθησης. Η αρχική του έκδοση ήταν γραμμένη σε C και ήταν απλά μια συλλογή λίγων αλγορίθμων μηχανικής μάθησης που είχαν μαζευτεί από διάφορες πηγές. Λόγω των πολλών εξαρτήσεων σε εξωτερικές βιβλιοθήκες κυρίως για την γραφική διεπαφή ήταν ιδιαίτερα δύσκολη η ανάπτυξη του και έτσι αποφασίστηκε το 1997 να υλοποιηθεί από την αρχή στη γλώσσα προγραμματισμού Java [089]. Μέχρι το 1999 δεν υπήρχε καθόλου γραφική διεπαφή στην έκδοση της Java και όλες οι λειτουργίες πραγματοποιούνταν από την γραμμή εντολών. Πλέον, υπάρχουν πολλοί τρόποι για να εκτελεστούν οι λειτουργίες του WEKA είτε μέσω γραφικής διεπαφής (π.χ. Explorer, Experimenter) είτε με τρίτες εφαρμογές (π.χ. με χρήση του Java API που παρέχει το WEKA). Τα βασικά του χαρακτηριστικά είναι [090]:

- **Επεξεργασία Δεδομένων:** Πέρα από τα αρχεία τύπου ARFF που αποτελεί τον εγγενή τύπο αρχείων του WEKA υποστηρίζονται πάρα πολλές άλλες μορφές όπως ASCII, CSV, Matlab κ.α. και σύνδεση με βάση δεδομένων μέσω JDBC. Τα δεδομένα μπορούν να φιλτραριστούν με την χρησιμοποίηση ενός μεγάλου αριθμού αλγορίθμων, από τους πιο απλούς όπως είναι η αφαίρεση συγκεκριμένων χαρακτηριστικών, έως τους πιο προχωρημένους όπως είναι η ανάλυση κυριών συνιστωσών.

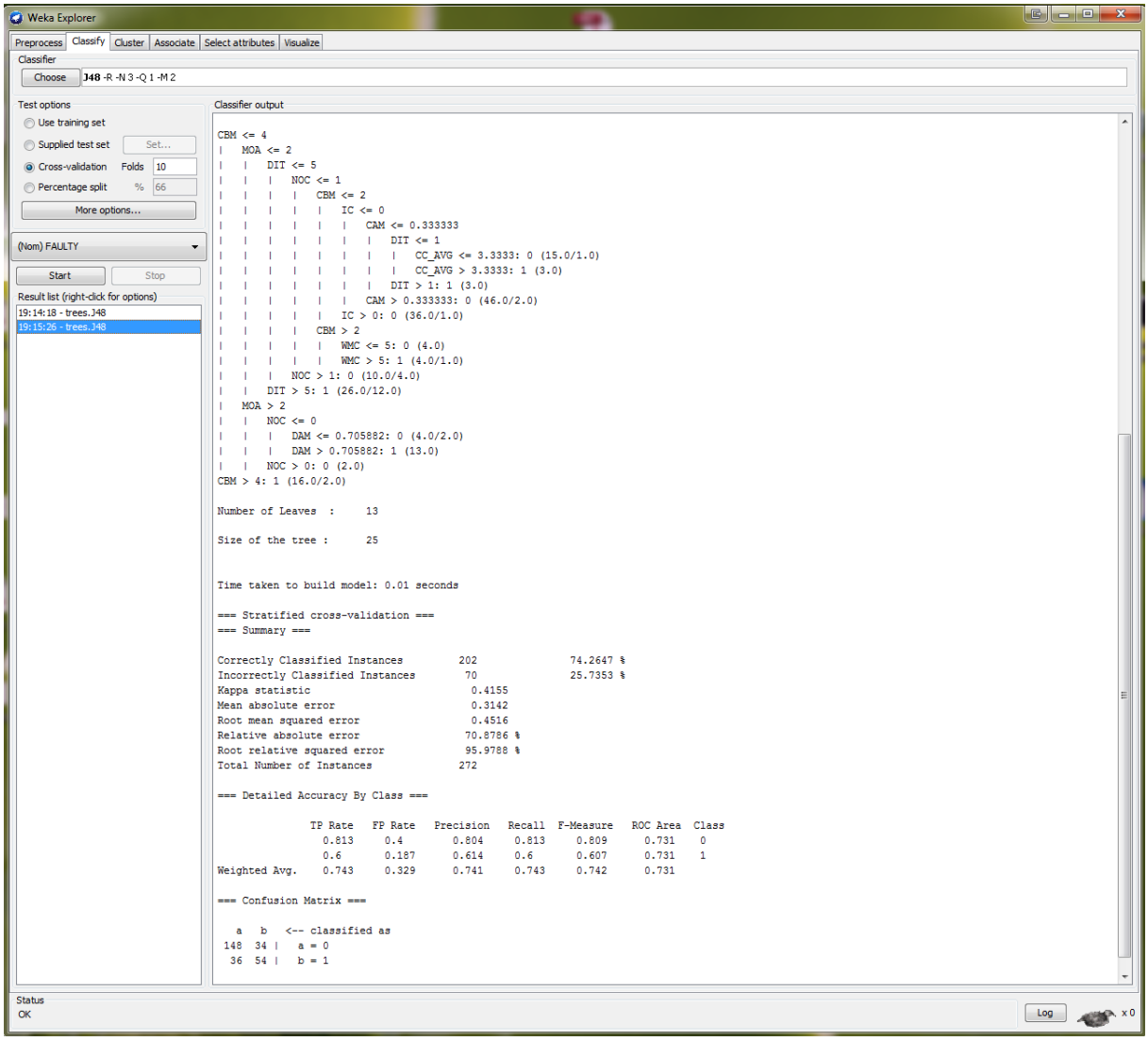
- **Κατηγοριοποίηση:** Υπάρχουν περισσότερες από εκατό μέθοδοι για την κατηγοριοποίηση των δεδομένων. Κατανέμονται στις μπεϊσιανές μεθόδους, τις οκνηρές (lazy) μεθόδους, τις βασιζόμενες σε κανόνες, τις δεντρικές, τις βασιζόμενες σε συναρτήσεις και τις διάφορες. Επιπλέον, το WEKA περιλαμβάνει μετά-ταξινομητές (meta-classifiers) όπως είναι οι bagging, boosting κλπ.

- **Ομαδοποίηση:** Η μη επιβλεπόμενη μάθηση υποστηρίζεται από πολλά διαφορετικά σχήματα όπως είναι τα EMbased μοντέλα, k-means και πολλούς διαφορετικούς ιεραρχικούς αλγορίθμους ομαδοποίησης. Παρά το γεγονός ότι δεν υπάρχουν τόσοι πολλοί αλγόριθμοι όσοι στην κατηγοριοποίηση, οι πιο κλασσικοί αλγόριθμοι υποστηρίζονται από το WEKA.

- **Επιλογή Χαρακτηριστικών:** Το σύνολο των χαρακτηριστικών που θα χρησιμοποιηθούν είναι ουσιώδες θέμα για την απόδοση της κατηγοριοποίησης. Πάρα πολλά κριτήρια επιλογής και μέθοδοι αναζήτησης είναι διαθέσιμοι.

- **Οπτικοποίηση Δεδομένων:** Μπορεί να γίνει εξέταση των δεδομένων με γραφική αναπαράσταση όπου με αντιπαράθεση των τιμών ενός χαρακτηριστικού με αυτά μιας κλάσης ή κάποιου άλλου χαρακτηριστικού μπορούν να βγουν χρήσιμα συμπεράσματα. Η έξοδος των μοντέλων μπορεί να συγκριθεί με τα δεδομένα εισόδου για να βρεθούν ασυνέπειες στα δεδομένα τους (outliers) ή να αναζητηθούν χρήσιμα χαρακτηριστικά τους. Για συγκεκριμένους αλγορίθμους υπάρχουν ειδικά εργαλεία για την οπτικοποίηση των αποτελεσμάτων π.χ. για τις μεθόδους που έχουν ως έξοδο κάποιο δέντρο υπάρχει σχετική δυνατότητα να αναπαρασταθεί η δομή του με γραφικό τρόπο.

Ο πιο δημοφιλής τρόπος χρησιμοποίησης του WEKA είναι μέσω της γραφικής διεπαφής που ονομάζεται Explorer και αυτός παρέχει υποστήριξη για όλες τις λειτουργίες που αναφέραμε πιο πάνω, με έναν απλό και διαδραστικό τρόπο. Στην εικόνα 4.5 παρουσιάζουμε την εκτέλεση του αλγορίθμου J48 για την δημιουργία δέντρου απόφασης στον Explorer.



Εικόνα 4.5: Το Γραφικό Περιβάλλον του Explorer Κατά την Εκτέλεση του Αλγόριθμου J48 στο WEKA

4.7 Κίνδυνοι για την Εγκυρότητα των Αποτελεσμάτων

Η παρούσα έρευνα έχει κάποιους περιορισμούς που μπορεί να είναι άλλωστε κοινοί στις περισσότερες εμπειρικές μελέτες αλλά θα πρέπει να αναφερθούν, ώστε να μπορεί να γίνει σωστή ερμηνεία των αποτελεσμάτων που ακολουθούν στο επόμενο κεφάλαιο.

Για την κατασκευή των μοντέλων πρόβλεψης σφαλμάτων χρησιμοποιήσαμε δεδομένα από ένα πρόγραμμα ανοικτού κώδικα μεσαίου μεγέθους που είναι υλοποιημένο στην αντικειμενοστρεφή γλώσσα προγραμματισμού Java. Για να μπορέσουμε να καταλήξουμε σε πιο γενικά συμπεράσματα θα πρέπει να άρουμε τους παραπάνω δύο περιορισμούς. Σχετικά με τον πρώτο περιορισμό, θα πρέπει να εξεταστούν περισσότερα προγράμματα ανοικτού κώδικα, διαφόρων μεγεθών από τα μικρότερα των λίγων κλάσεων μέχρι τα πιο μεγάλα συστήματα που περιλαμβάνουν χιλιάδες κλάσεις και εκατοντάδες πακέτα. Μάλιστα, χρειάζεται ιδιαίτερη προσοχή στην επιλογή τους ώστε αυτά που θα επιλεγούν τελικά να αποτελούν ένα αντιπροσωπευτικό δείγμα όλων των έργων ανοικτού κώδικα που είναι διαθέσιμα. Ιδανικά για την επιλογή τους θα πρέπει να δημιουργεί μια σχετική μεθοδολογία με συγκεκριμένα κριτήρια επιλογής. Σχετικά με τον δεύτερο περιορισμό, η επιλογή της γλώσσας προγραμματισμού Java έγινε γιατί είναι "καθαρή" αντικειμενοστρεφή γλώσσα προγραμματισμού και όχι "υβριδική" όπως είναι π.χ. η C++, την οποία εξετάζουν άλλωστε οι περισσότερες σχετικές έρευνες. Όμως, για να έχουμε γενικευμένα αποτελέσματα, θα πρέπει να εξεταστούν και οι υπόλοιπες αντικειμενοστρεφείς γλώσσες προγραμματισμού όπως είναι οι C#, Object Pascal, Ruby κλπ για τις οποίες δυστυχώς δεν υπάρχουν σχεδόν καθόλου στοιχεία.

Στην παρούσα διπλωματική διατριβή χρησιμοποιήσαμε τρεις παραδοσιακές μετρικές και δεκαεφτά αντικειμενοστρεφείς, αριθμός ικανοποιητικός μεν αλλά ίσως όχι πλήρης. Στην βιβλιογραφία έχουν προταθεί κατά καιρούς δεκάδες μετρικές για τις οποίες είτε δεν υπάρχουν καθόλου εμπειρικά αποτελέσματα είτε αυτά αν υπάρχουν είναι πολύ περιορισμένα. Κάποιες από αυτές τις μετρικές θα μπορούσαν να βοηθήσουν στην πρόβλεψη σφαλμάτων στις κλάσεις ενός προγράμματος περισσότερο από αυτές που χρησιμοποιήθηκαν στην έρευνα μας. Τα συμπεράσματα μας περιορίζονται λοιπόν στο σετ μετρικών που δίνει το εργαλείο ckjm extended και δεν μπορούμε σε καμία περίπτωση να γενικεύσουμε τους ισχυρισμούς μας για όλες τις αντικειμενοστρεφείς μετρικές. Άλλος ένας πιθανός κίνδυνος είναι η αξιοπιστία των δεδομένων που συλλεχτήκαν. Όπως είδαμε στην ενότητα 4.2, τα εργαλεία μετρικών υπολογίζουν με διαφορετικό τρόπο τις ίδιες μετρικές με αποτέλεσμα να είναι επισφαλής η σύγκριση των αποτελεσμάτων που έχουν προέρθει από διαφορετικά εργαλεία ή αφορούν διαφορετικές

γλώσσες προγραμματισμού. Επιπλέον, δεν υπάρχει αξιόπιστος τρόπος για την αυτόματη συλλογή και αντιστοίχηση των σφαλμάτων με τις κλάσεις του πηγαίου κώδικα. Το εργαλείο BugInfo που χρησιμοποιήθηκε για την συλλογή των σφαλμάτων δεν λειτούργησε με τρόπο ικανοποιητικό αλλά ούτε και αποδοτικό. Ακόμη και όταν το εργαλείο λειτούργησε σωστά, πολλές φορές είχαμε λάθος αντιστοίχιση λόγω της πλημμελούς τεκμηρίωσης από την πλευρά των προγραμματιστών σχετικά με τα σφάλματα που διορθώνονται. Οπότε επειδή η αντιστοίχηση των σφαλμάτων με τις κλάσεις έγινε με αυτόματο τρόπο αρχικά και μετά με χειροκίνητο για διορθώσεις, δεν μπορούμε να είμαστε απόλυτα βέβαιοι ότι είναι και απολύτως ορθά π.χ. είναι δύσκολο να εντοπιστεί μια κλάση στην οποία έγιναν αλλαγές και ταυτόχρονα άλλαξε όνομα ή και πακέτο.

Η μη κατηγοριοποίηση των σφαλμάτων ανάλογα με την σπουδαιότητα τους, επίσης είναι ένας σημαντικός περιορισμός που διέπει την έρευνα μας. Αν και όλα τα σφάλματα δεν είναι ίδια, στην ανάλυση μας δίνουμε την ίδια βαρύτητα σε όλα τα σφάλματα του λογισμικού που μελετάμε. Όμως, στην πραγματικότητα μας ενδιαφέρουν περισσότερο αυτά που κρίνονται ως σοβαρότερα και χρίζουν άμεσης διόρθωσης π.χ. αυτά που μπορούν να έχουν καταστροφικά αποτελέσματα για τον χειριστή τους όπως είναι ο απότομος τερματισμός της εφαρμογής χωρίς να αποθηκευτεί η εργασία που έχει κάνει μέχρι εκείνο το σημείο ο χρήστης. Μια πιο ολοκληρωμένη προσέγγιση οφείλει να λαμβάνει υπ' όψιν της την σοβαρότητα του κάθε σφάλματος και να δημιουργήσει μοντέλα που να μπορούν να κάνουν διάκριση μεταξύ της σοβαρότητας του κάθε σφάλματος, ώστε οι προσπάθειες εντοπισμού των σφαλμάτων να μπορούν να διεξάγονται έχοντας πρώτα ιεραρχήσει τα πιθανά σφάλματα με βάση την σοβαρότητα τους. Όμως, υπάρχει μεγάλη δυσκολία στην συλλογή αυτών των πρωτογενών στοιχείων που θα περιέχουν κατηγοριοποίηση των σφαλμάτων ανάλογα την σημαντικότητα τους, αφού στα περισσότερα έργα ανοικτού κώδικα είτε δεν υπάρχει αυτή η κατηγοριοποίηση είτε αν υπάρχει έχουν παρατηρηθεί πολλά προβλήματα που αφορούν την αξιοπιστία τους.

Υπάρχει μια τεράστια ποικιλία από μεθόδους στατιστικής ανάλυσης αλλά και μηχανικής μάθησης που μπορεί να χρησιμοποιήσει κάποιος για την κατασκευή μοντέλων πρόβλεψης σφαλμάτων στις κλάσεις ενός λογισμικού ανοικτού κώδικα. Στην παρούσα έρευνα επικεντρωθήκαμε αναφορικά με τις στατιστικές τεχνικές σε αυτές που έχουν αποδείξει την αξία τους στο διάβα του χρόνου όπως είναι η γραμμική και η λογιστική παλινδρόμηση. Όμως υπάρχουν και άλλες πάρα πολλές τεχνικές στατιστικής ανάλυσης που θα μπορούσε κάποιος να εφαρμόσει για την δημιουργία μοντέλων πρόβλεψης σφαλμάτων όπως είναι οι rubust regression, probit regression, poisson regression, principal component analysis, discriminant

function analysis κλπ. Περίπου τα ίδια ισχύουν και για την μηχανική μάθηση όπου η επιλογή των πιο αποδοτικών αλγορίθμων εκμάθησης για ένα πρόβλημα δεν είναι κάτι εύκολο και απαιτούνται πολλοί πειραματισμοί για την εξεύρεση της πιο αποδοτικής λύσης. Οι αλγόριθμοι μηχανικής μάθησης που εφαρμόσαμε είναι το δέντρο απόφασης και το πολυεπίπεδο τεχνητό νευρωνικό δίκτυο αισθητήρα που εκπαιδεύεται με την μέθοδο της πίσω διάδοσης του λάθους. Για πιο ασφαλή και γενικευμένα συμπεράσματα σχετικά με την ικανότητα των τεχνικών μηχανικής μάθησης στην πρόβλεψη σφαλμάτων σε λογισμικό σε σύγκριση με τις τεχνικές στατιστικής ανάλυσης, υπάρχουν πολλοί άλλοι αλγόριθμοι μηχανικής μάθησης που μπορούν να εξεταστούν για να επιτύχει κάποιος τον παραπάνω σκοπό όπως είναι π.χ. οι Bayesian Classifiers, Support Vector Machines, AdaBoost κλπ.

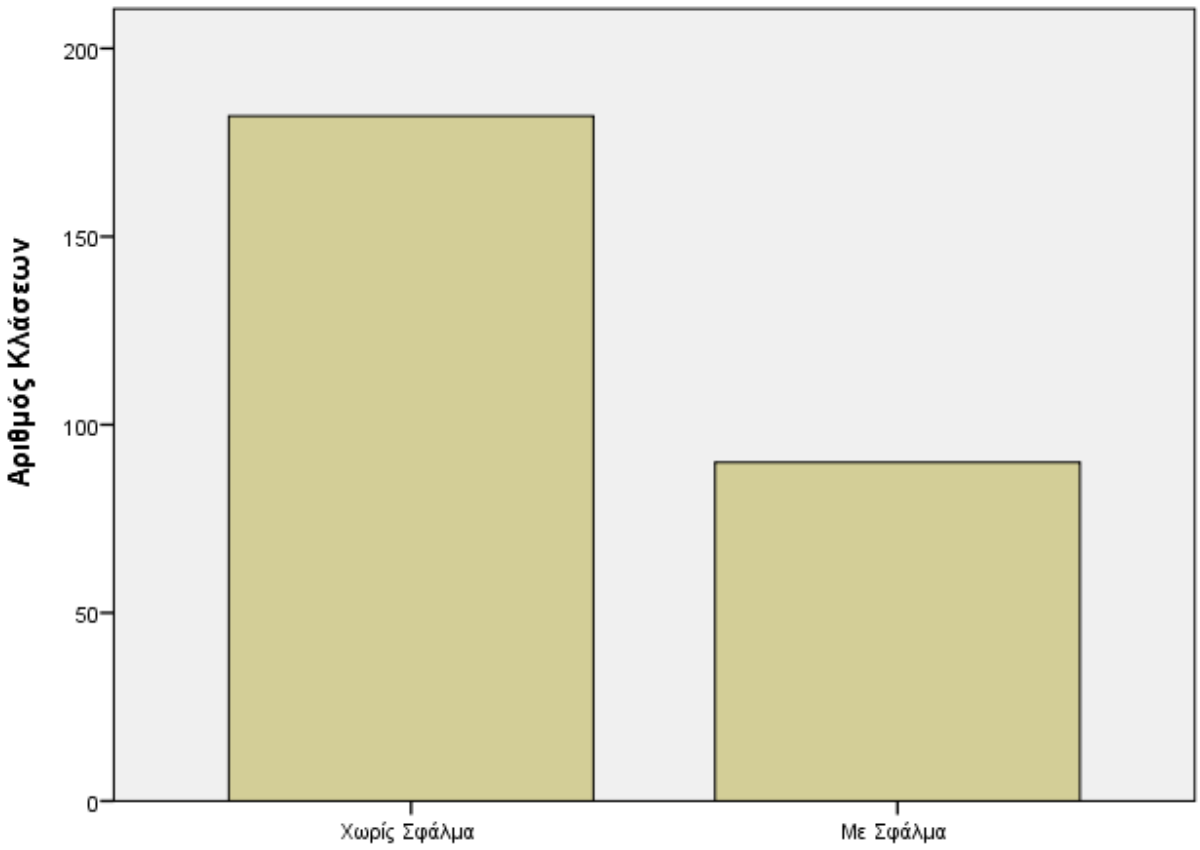
Κλείνοντας αυτή την ενότητα θα πρέπει να διευκρινιστεί ότι τα συμπεράσματα μας για την καταλληλότητα των αντικειμενοστρεφών μετρικών αφορούν αποκλειστικά την καταλληλότητα τους ως εισόδους σε μοντέλα για την πρόβλεψη σφαλμάτων στις κλάσεις ενός προγράμματος ανοικτού κώδικα. Οποιαδήποτε επέκταση της χρησιμότητας τους όταν η εξαρτημένη μεταβλητή είναι κάποια άλλη μεταβλητή που μπορεί να έχει κάποια σχέση με τα σφάλματα όπως είναι π.χ. η συντηρητικότητα του πηγαίου κώδικα ή η προσπάθεια που απαιτείται για την διόρθωση τους, είναι κάτι που δεν πραγματοποιείται αυτομάτως και χρειάζονται νέες εμπειρικές μελέτες για να επιβεβαιωθεί ή απορριφθεί μια εικασία αυτού του είδους.

Κεφάλαιο 5
Πειραματικά Αποτελέσματα και Ανάλυση

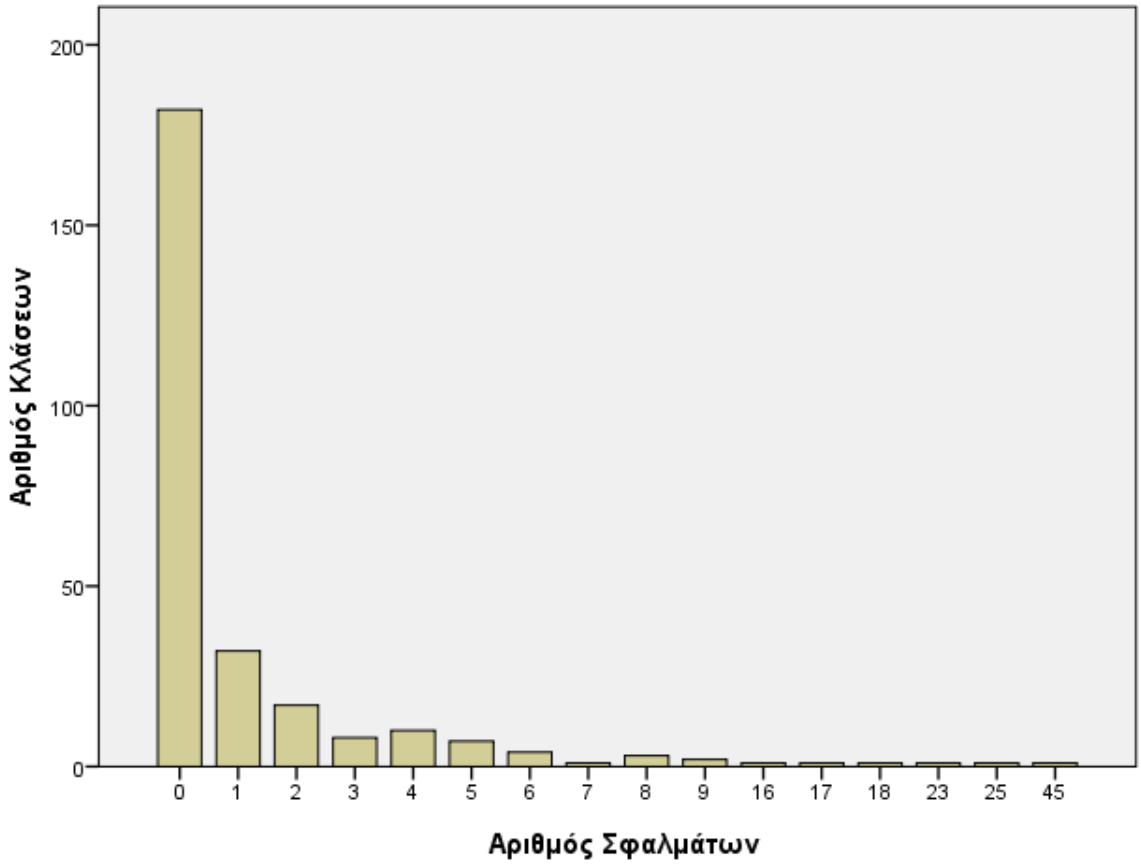
Ξεκινάμε τα αποτελέσματα μας με περιγραφική στατιστική για τις κλάσεις και τις μετρικές που αφορούν το πρόγραμμα jEdit έκδοση 3.2. Μετά εξετάζουμε τις πιθανές συσχετίσεις μεταξύ των μετρικών με τη χρήση του συντελεστή Spearman. Η πρώτη μέθοδος που χρησιμοποιούμε για την κατασκευή μοντέλων είναι στατιστική και είναι η απλή γραμμική παλινδρόμηση όπου διερευνούμε κατά πόσο κάθε μετρική ξεχωριστά μπορεί να μας βοηθήσει να προβλέψουμε τον αριθμό των λαθών που χρειάστηκαν διόρθωση σε μια κλάση. Μετά χρησιμοποιούμε βηματική πολλαπλή γραμμική παλινδρόμηση όπου η προσθήκη των ανεξάρτητων μεταβλητών γίνεται με την επιλογή προς τα μπρος (stepwise forward selection). Η δεύτερη μέθοδος μας είναι η δυαδική λογιστική παλινδρόμηση όπου τώρα η εξαρτημένη μεταβλητή είναι δυαδική με τιμές την ύπαρξη ή όχι προβλήματος σε μια κλάση. Με την έννοια πρόβλημα εννοούμε στα πλαίσια του πειράματος μας την ύπαρξη ενός ή περισσότερων σφαλμάτων στη συγκεκριμένη κλάση. Συνεχίζουμε με δύο μεθόδους που ανήκουν στη γενικότερη κατηγορία της μηχανικής μάθησης το δέντρο απόφασης και το τεχνητό νευρωνικό δίκτυο, όπου πάλι εξετάζουμε κατά πόσο προκύπτουν μοντέλα που προβλέπουν επιτυχώς την ύπαρξη προβλήματος σε μια κλάση. Τέλος, κάνουμε μια ανάλυση των αποτελεσμάτων μας και τα συγκρίνουμε με άλλες σχετικές μελέτες.

5.1 Περιγραφική Στατιστική

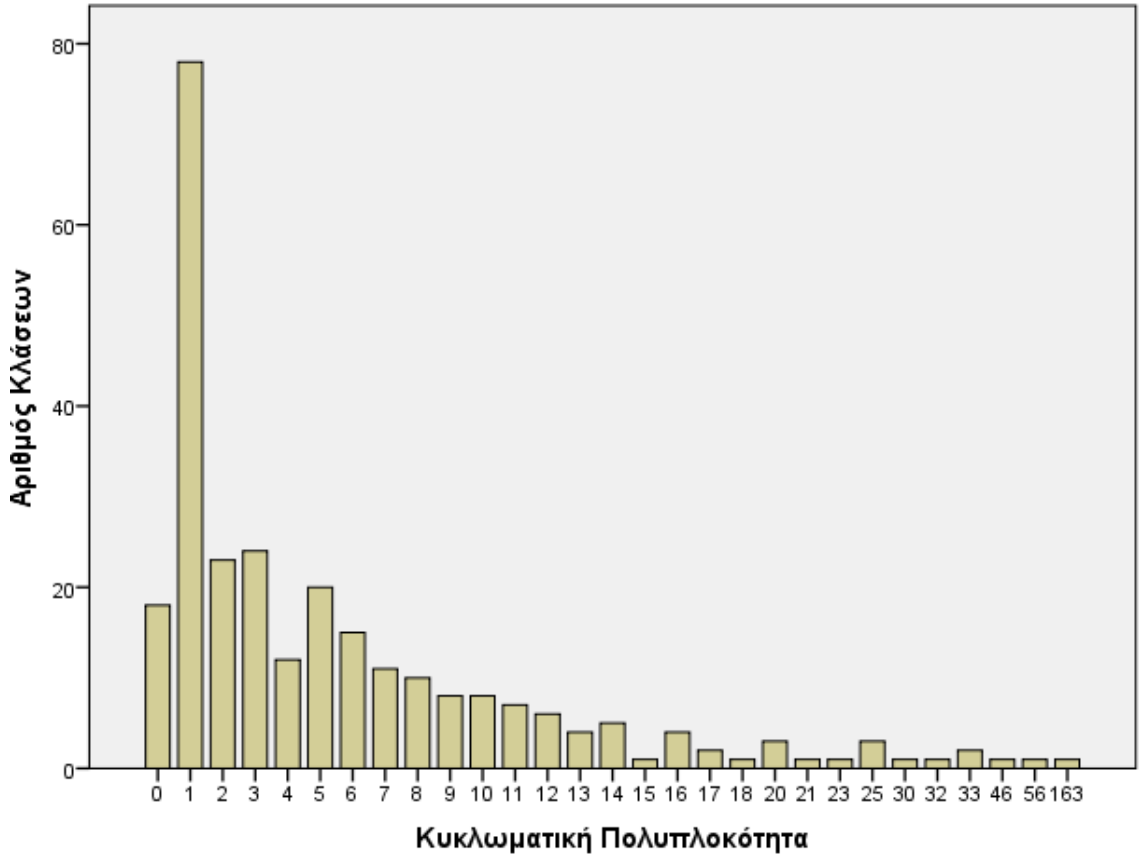
Η έκδοση 3.2 του προγράμματος jEdit αποτελείται συνολικά από 272 κλάσεις. Από αυτές όπως παρουσιάζεται στο σχήμα 5.1, δεν διορθώθηκε κανένα σφάλμα στις 182 και στις υπόλοιπες 90 διορθώθηκε τουλάχιστον ένα σφάλμα. Δηλαδή περίπου στα δύο τρίτα των κλάσεων δεν έγινε καμία αλλαγή διορθωτικού χαρακτήρα και μόνο περίπου στο ένα τρίτο των κλάσεων είχαμε τη διόρθωση κάποιου ή κάποιων σφαλμάτων. Επιπλέον, στις κλάσεις που παρουσίασαν κάποιο σφάλμα παρατηρούμε ότι αν χωρίσουμε τον αριθμό των σφαλμάτων σε τρεις κατηγορίες που η καθεμία να έχει το ένα τρίτο του ποσοστού αυτής της κατηγορίας των κλάσεων, τότε η μια κατηγορία έχει μόνο ένα σφάλμα, η επόμενη από δύο έως τέσσερα και η τελευταία περισσότερα από πέντε σφάλματα (βλέπε και σχήμα 5.2). Χρήσιμη είναι και η μελέτη της κατανομής κάθε μετρικής σε σχέση με τον αριθμό των κλάσεων που αυτή παρατηρείται, όμως για λόγους οικονομίας του χώρου έχουν μετακινηθεί στο παράρτημα Β. Ενδεικτικά στην ενότητα αυτή παρουσιάζουμε στο σχήμα 5.3 την κατανομή της μέγιστης κυκλωματικής πολυπλοκότητας των μεθόδων μιας κλάσης, όπου γίνεται σαφές ότι η συντριπτική πλειοψηφία των κλάσεων στο πρόγραμμα jEdit 3.2 έχει μέγιστη κυκλωματική πολυπλοκότητα μικρότερη του είκοσι και άρα δεν πρόκειται για ιδιαίτερα πολύπλοκο πηγαίο κώδικα [001].



Σχήμα 5.1: Ραβδόγραμμα με Διαχωρισμό των Κλάσεων σε Εσφαλμένες ή μη του Προγράμματος jEdit 3.2



Σχήμα 5.2: Ραβδόγραμμα με τη Συχνότητα των Σφαλμάτων στις Κλάσεις του Προγράμματος jEdit 3.2



Σχήμα 5.3: Ραβδόγραμμα με τη Συχνότητα της Κυκλωματικής Πολυπλοκότητας στις Κλάσεις του Προγράμματος jEdit 3.2

Μετρική	Ελάχιστο	Μέγιστο	Μέσος		Τυπική Απόκλιση	Ασυμμετρία		Κύρτωση	
				Τυπικό Σφάλμα			Τυπικό Σφάλμα		Τυπικό Σφάλμα
WMC	1	399	12,53	1,86	30,64	8,99	0,15	100,81	0,29
DIT	1	8	2,83	0,13	2,13	1,00	0,15	-0,43	0,29
NOC	0	35	0,43	0,16	2,69	9,67	0,15	109,35	0,29
CBO	1	162	12,04	1,00	16,42	4,92	0,15	33,28	0,29
RFC	1	487	37,74	3,35	55,21	4,69	0,15	29,27	0,29
LCOM	0	11469	171,26	57,96	955,97	8,79	0,15	87,30	0,29
LCOM3	0	2	1,05	0,04	0,61	0,61	0,15	-0,96	0,29
IC	0	4	0,65	0,06	0,93	1,63	0,15	2,48	0,29
CBM	0	18	1,46	0,17	2,84	3,23	0,15	12,09	0,29
AMC	0	496	32,65	2,35	38,71	6,81	0,15	75,63	0,29
CA	0	137	7,15	0,87	14,34	5,11	0,15	33,92	0,29
CE	0	48	6,23	0,43	7,08	2,90	0,15	11,55	0,29
NPM	0	169	7,32	0,92	15,15	6,48	0,15	55,33	0,29
DAM	0	1	0,53	0,03	0,47	-0,15	0,15	-1,88	0,29
MOA	0	16	0,90	0,11	1,81	3,81	0,15	21,42	0,29
MFA	0	1	0,56	0,03	0,44	-0,37	0,15	-1,70	0,29
CAM	0	1	0,47	0,02	0,25	0,69	0,15	-0,14	0,29
LOC	1	23350	473,83	98,28	1620,84	11,18	0,15	149,36	0,29
CC_AVG	0	9	1,72	0,09	1,45	2,11	0,15	6,31	0,29
CC_MAX	0	163	6,31	0,73	11,98	8,93	0,15	109,42	0,29

Πίνακας 5.1: Περιγραφικά Στατιστικά των Κλάσεων του Προγράμματος jEdit 3.2

Στον πίνακα 5.1 παρουσιάζουμε τα αναλυτικά περιγραφικά στατιστικά στοιχεία και για τις είκοσι μετρικές. Από αυτόν μπορούν να εξαχθούν πολύ χρήσιμα συμπεράσματα. Από τις τιμές της ασυμμετρίας (skewness) και της κύρτωσης (kurtosis) προκύπτει το γεγονός ότι σχεδόν καμία μετρική δεν ακολουθεί την κανονική κατανομή και έτσι οποιοδήποτε στατιστικό τεστ που έχει ως προαπαιτούμενο την κανονική κατανομή των μεταβλητών δεν θα πρέπει να χρησιμοποιηθεί γιατί δεν θα μας δώσει σωστά αποτελέσματα. Το μέσο μέγεθος μιας κλάσης είναι περίπου πεντακόσιες γραμμές με τυπικό σφάλμα τις εκατό γραμμές, πράγμα που σημαίνει ότι δεν είναι ιδιαίτερα μεγάλες και δεν υπάρχει γενικά μεγάλη απόκλιση από αυτό το μέγεθος δηλαδή δεν υπάρχουν πολλές κλάσεις με το μέγιστο μέγεθος που είναι είκοσι τρεις χιλιάδες γραμμές. Αν και εμπειρικές μελέτες [002, 003] έχουν δείξει ότι η πυκνότητα λαθών είναι μικρότερη όταν το τμήμα κάθε κώδικα είναι μεταξύ 200 και 750 γραμμών, το ιστορικό του προγράμματος jEdit δεν φαίνεται να τις επιβεβαιώνει. Από τις 272 κλάσεις της έκδοσης 3.2 οι ενενήντα είχαν κάποιο σφάλμα που διορθώθηκε και τα συνολικά σφάλματα που διορθώθηκαν ήταν 382, δηλαδή αντιστοιχεί κατά μέσο όρο σχεδόν ενάμιση σφάλμα σε κάθε κλάση του

προγράμματος. Η κληρονομικότητα βρίσκεται σε λογικά πλαίσια με τη μέση κλάση να είναι στο τρίτο επίπεδο που συνεπάγεται στην πραγματικότητα κατά μέσο όρο δύο επίπεδα προγράμματος , αφού όπως είναι γνωστό στη γλώσσα προγραμματισμού Java όλες οι κλάσεις που δημιουργούνται κληρονομούν αναγκαστικά από την κλάση Object [004].

Ο αριθμός των μετρικών που εξετάσαμε είναι σχετικά μεγάλος και είναι λογικό να υπάρχει κάποια μεταξύ τους αλληλοεπικάλυψη σχετικά με την πρόβλεψη σφαλμάτων σε μία κλάση. Για να διερευνήσουμε την ένταση που μπορεί να ισχύει κάτι τέτοιο, υπολογίσαμε τον συντελεστή συσχέτισης Spearman για όλα τα ζευγάρια μετρικών. Τα αποτελέσματα παρουσιάζονται στον πίνακα 5.2, όπου επιλέξαμε πέρα από το αυστηρό επίπεδο σημαντικότητας 1% να σημειώσουμε και τους συντελεστές συσχέτισης που είναι στατιστικά σημαντικοί και στο επίπεδο του 5%. Από τη μελέτη του προκύπτουν πολύ χρήσιμα συμπεράσματα κάποια από τα οποία θα περιμέναμε να ισχύουν ούτως ή άλλως, όπως ενδεικτικά:

- Η μετρική WMC είναι ισχυρά θετικά συσχετισμένη με την RFC, LCOM, NPM και CAM. Δηλαδή όσο μεγαλύτερος είναι ο αριθμός τον μεθόδων σε μια κλάση τόσο μεγαλύτερη είναι η έλλειψη της συνεκτικότητας μεταξύ τους (LCOM)., τόσες περισσότερες δημόσιες μεθόδους έχει η κλάση (NPM) κλπ

- Η μέση πολυπλοκότητα (CC_AVG) είναι πάρα πολύ θετικά συσχετισμένη (0,942) με τη μέγιστη πολυπλοκότητα σε μια κλάση (CC_MAX) που σημαίνει ότι όσο πιο μεγάλος είναι ο μέσος όρος σε μια κλάση τόσο υπάρχει η τάση να υπάρχουν και μέθοδοι με μεγαλύτερη κυκλωματική πολυπλοκότητα. Αντίθετα, η συσχέτιση της μέσης πολυπλοκότητας με την μετρικής LOC δεν είναι ιδιαίτερα υψηλή δηλαδή οι περισσότερες γραμμές κώδικα σε μια κλάση δεν συνεπάγονται και μεγαλύτερη μέση πολυπλοκότητα.

- Η τάση της σύζευξης μεταξύ των μεθόδων μιας κλάσης (CBM) είναι σχεδόν ταυτισμένη (0,976) με αυτής της κληρονομικής σύζευξης (IC), οπότε ουσιαστικά θα μπορούσαμε να χρησιμοποιήσουμε μόνο μια από αυτές τις δυο μετρικές που πρότεινε ο Tang [005] για την επέκταση των μετρικών CK.

- Όσο πιο βαθιά είναι μια κλάση στο δέντρο της κληρονομικότητας (DIT) τόσο περισσότερο αυξάνεται (0,928) ο λόγος των μεθόδων που κληρονομεί η κλάση ως προς το σύνολο των μεθόδων που είναι προσβάσιμες μέσω αυτής της κλάσης (MFA), οπότε έχουμε αλληλοεπικάλυψη μεταξύ των μετρικών CK και του μοντέλου QMOOD.

* Στατιστικά Σημαντικός στο Επίπεδο 5%
** Στατιστικά Σημαντικός στο Επίπεδο 1%

	WMC	DIT	NOC	CBO	RFC	LCOM	LCOM3	IC	CBM	AMC	CA	CE	NPM	DAM	MOA	MFA	CAM	LOC	CC_AVG	CC_MAX
WMC	1,00	-0,08	,160**	,456**	,813**	,776**	-,312**	0,05	0,07	,192**	,440**	,385**	,781**	,453**	,493**	-,290**	-,800**	,755**	,570**	,676**
DIT	-0,08	1,00	-,153*	-0,01	,274**	0,03	-0,03	,531**	,518**	,322**	-,190**	,322**	-,211**	,175**	0,06	,928**	0,08	,157**	-0,05	0,00
NOC	,160**	-,153*	1,00	,229**	0,01	,169**	-0,01	-,202**	-,203**	-,163**	,320**	-0,08	,193**	0,01	0,07	-,164**	-0,08	-0,04	-0,06	-0,06
CBO	,456**	-0,01	,229**	1,00	,483**	,447**	0,05	0,03	-0,01	,144*	,679**	,648**	,370**	0,03	,382**	-0,07	-,399**	,392**	0,10	,199**
RFC	,813**	,274**	0,01	,483**	1,00	,675**	-,285**	,187**	,204**	,584**	,271**	,640**	,567**	,456**	,489**	0,10	-,689**	,903**	,577**	,699**
LCOM	,776**	0,03	,169**	,447**	,675**	1,00	0,09	0,11	0,11	0,09	,382**	,417**	,652**	,188**	,345**	-,123*	-,640**	,555**	,359**	,477**
LCOM3	-,312**	-0,03	-0,01	0,05	-,285**	0,09	1,00	-0,11	-,139*	-,226**	0,06	-0,05	-,152*	-,701**	-,254**	0,08	,250**	-,312**	-,430**	-,400**
IC	0,05	,531**	-,202**	0,03	,187**	0,11	-0,11	1,00	,976**	,224**	-,215**	,310**	-0,09	0,09	0,06	,444**	0,00	,165**	,130*	,176**
CBM	0,07	,518**	-,203**	-0,01	,204**	0,11	-,139*	,976**	1,00	,237**	-,231**	,293**	-0,09	,136*	0,06	,431**	0,00	,188**	,172**	,215**
AMC	,192**	,322**	-,163**	,144*	,584**	0,09	-,226**	,224**	,237**	1,00	-,169**	,464**	0,02	,230**	,218**	,265**	-,266**	,744**	,467**	,501**
CA	,440**	-,190**	,320**	,679**	,271**	,382**	0,06	-,215**	-,231**	-,169**	1,00	0,09	,436**	0,06	,312**	-,257**	-,342**	,199**	0,03	0,11
CE	,385**	,322**	-0,08	,648**	,640**	,417**	-0,05	,310**	,293**	,464**	0,09	1,00	,192**	,152*	,404**	,258**	-,395**	,543**	,226**	,353**
NPM	,781**	-,211**	,193**	,370**	,567**	,652**	-,152*	-0,09	-0,09	0,02	,436**	,192**	1,00	,260**	,375**	-,365**	-,579**	,506**	,413**	,475**
DAM	,453**	,175**	0,01	0,03	,456**	,188**	-,701**	0,09	,136*	,230**	0,06	,152*	,260**	1,00	,363**	0,04	-,418**	,430**	,450**	,467**
MOA	,493**	0,06	0,07	,382**	,489**	,345**	-,254**	0,06	0,06	,218**	,312**	,404**	,375**	,363**	1,00	-0,06	-,472**	,477**	,268**	,372**
MFA	-,290**	,928**	-,164**	-0,07	0,10	-,123*	0,08	,444**	,431**	,265**	-,257**	,258**	-,365**	0,04	-0,06	1,00	,264**	-0,01	-,209**	-,158**
CAM	-,800**	0,08	-0,08	-,399**	-,689**	-,640**	,250**	0,00	0,00	-,266**	-,342**	-,395**	-,579**	-,418**	-,472**	,264**	1,00	-,703**	-,521**	-,625**
LOC	,755**	,157**	-0,04	,392**	,903**	,555**	-,312**	,165**	,188**	,744**	,199**	,543**	,506**	,430**	,477**	-0,01	-,703**	1,00	,657**	,765**
CC_AVG	,570**	-0,05	-0,06	0,10	,577**	,359**	-,430**	,130*	,172**	,467**	0,03	,226**	,413**	,450**	,268**	-,209**	-,521**	,657**	1,00	,942**
CC_MAX	,676**	0,00	-0,06	,199**	,699**	,477**	-,400**	,176**	,215**	,501**	0,11	,353**	,475**	,467**	,372**	-,158**	-,625**	,765**	,942**	1,00

Πίνακας 5.2: Συντελεστής Συσχέτισης Spearman για τις Μετρικές στο Πρόγραμμα jEdit 3.2

5.2 Γραμμική Παλινδρόμηση

Στην ενότητα αυτή θα περιγράψουμε τα αποτελέσματα που είχαμε για την πρόβλεψη του αριθμού των σφαλμάτων μιας κλάσης με τη χρήση αρχικά της απλής παλινδρόμησης για να απομονώσουμε την σημαντικότητα κάθε μιας μετρικής και στη συνέχεια με πολλαπλή παλινδρόμηση για να διερευνήσουμε τη συνδυασμένη επίδραση τους.

5.2.1 Απλή Γραμμική Παλινδρόμηση

Τα αποτελέσματα εφαρμογής της απλής γραμμικής παλινδρόμησης παρουσιάζονται στον πίνακα 5.3 όπου για κάθε μια από τις μετρικές υπάρχει ο σταθερός όρος της εξίσωσης, ο συντελεστής της μετρικής και το επίπεδο σημαντικότητας του συντελεστή. Έτσι, π.χ. ο συντελεστής της μετρικής WMC είναι στατιστικά σημαντικός στο επίπεδο σημαντικότητας 1% και η εξίσωση της πρόβλεψης του αριθμού των σφαλμάτων που προκύπτει είναι η εξής:

$$\text{Αριθμός Σφαλμάτων Κλάσης} = 0{,}624 + 0{,}062 * \text{WMC} \quad (5.1)$$

Επιπρόσθετα επειδή η κάθε μετρική έχει διαφορετικό εύρος τιμών και είναι δύσκολο να αποτιμήσει κανείς την σχετική τους σημαντικότητα όσο αφορά τον αριθμό των σφαλμάτων σε μια κλάση, έχουμε προσθέσει και τους τυποποιημένους συντελεστές. Αυτοί υπολογίζονται λαμβάνοντας υπόψη τη διακύμανση κάθε μετρικής και έτσι μας διευκολύνουν ώστε να έχουμε μια άποψη για την επίδραση τους στην μεταβολή του αριθμού των σφαλμάτων. Έτσι, π.χ. ο συντελεστής του αριθμού των γραμμών πηγαίου κώδικα LOC μπορεί να έχει συντελεστή 0,001 αλλά δεν σημαίνει ότι έχει μικρότερη επιρροή από τη μέση κυκλωματική πολυπλοκότητα CC_AVG που έχει συντελεστή 0,695. Αυτό συμβαίνει γιατί ο τυποποιημένος συντελεστής της LOC είναι 0,327 (που όπως είδαμε στην ενότητα έχει εύρος τιμών από 1 έως 23350) και της CC_AVG είναι 0,245 (που έχει εύρος τιμών από 0 έως 9).

Οι περισσότερες μετρικές έχουν στατιστικά σημαντικούς συντελεστές και μάλιστα με P-Value μηδέν (με την στρογγυλοποίηση στα τρία δεκαδικά). Εξαίρεση έχουμε για πέντε μετρικές και πιο συγκεκριμένα για τις NOC, LCOM3, AMC, MFA σε επίπεδο σημαντικότητας 5% και επιπλέον την DIT σε επίπεδο σημαντικότητας 1%. Αυτές τις μετρικές μπορούμε με ασφάλεια να μην τις χρησιμοποιήσουμε στην πολλαπλή γραμμική παλινδρόμηση και γενικότερα στην ανάλυση που ακολουθεί, αφού μας ενδιαφέρουν οι καλύτερες μεταβλητές για πρόβλεψη σφαλμάτων.

	Σταθερός Όρος (α)	Συντελεστής (β)	Τυποποιημένος Συντελεστής (Standard Coefficient)	Επίπεδο Σημαντικότητας (P-Value)	Συντελεστής Προσδιορισμού (R^2)
WMC	0,624	0,062	0,465	0,000	0,216
DIT	0,685	0,254	0,132	0,030	0,017
NOC	1,431	-0,063	-0,041	0,498	0,002
CBO	-0,195	0,133	0,531	0,000	0,282
RFC	-0,535	0,051	0,691	0,000	0,478
LCOM	0,868	0,003	0,730	0,000	0,532
LCOM3	2,059	-0,622	-0,092	0,129	0,009
IC	0,760	0,985	0,223	0,000	0,050
CBM	0,756	0,446	0,308	0,000	0,095
AMC	1,179	0,007	0,065	0,285	0,004
CA	0,478	0,129	0,452	0,000	0,204
CE	-0,791	0,352	0,607	0,000	0,369
NPM	-0,032	0,196	0,724	0,000	0,524
DAM	0,375	1,926	0,219	0,000	0,048
MOA	0,353	1,167	0,515	0,000	0,266
MFA	1,196	0,375	0,040	0,512	0,002
CAM	3,753	-5,014	-0,307	0,000	0,094
LOC	1,012	0,001	0,327	0,000	0,107
CC_AVG	0,212	0,695	0,245	0,000	0,060
CC_MAX	0,806	0,095	0,277	0,000	0,077

Πίνακας 5.3: Αποτελέσματα Απλής Γραμμικής Παλινδρόμησης για το Πρόγραμμα jEdit 3.2

Μια άλλη σημαντική παράμετρος είναι το ποσοστό μεταβλητότητας των δεδομένων που μπορεί να εξηγηθεί από το μοντέλο μας και ένας τρόπος για να γίνει αυτό στην γραμμική παλινδρόμηση είναι ο συντελεστής προσδιορισμού R^2. Έτσι με μια προσεκτική ματιά στον πίνακα 5.3 παρατηρούμε ότι οι μετρικές RFM, LCOM και NPM έχουν συντελεστή καλής προσαρμογής μεγαλύτερο του 0,5 και αποτελούν τις καλύτερες μετρικές για τον προσδιορισμό του αριθμού των σφαλμάτων σε μια κλάση. Αντίθετα, οι μετρικές DIT, NOC, LCOM3, AMC, MFA έχουν συντελεστή καλής προσαρμογής μικρότερο του 0,01 και άρα τα μοντέλα τους δεν προσεγγίζουν καθόλου καλά τον αριθμό των λαθών. Στο ενδιάμεσο βρίσκονται οι μετρικές WMC, CBO. RFC, CE και MOA με συντελεστή προσαρμογής ανάμεσα στο 0,25 και 0,5, δηλαδή μπορεί να μην προσεγγίζουν πάρα πολύ καλά τον αριθμό των λαθών αλλά έχουν να μας δώσουν πληροφορία.

5.2.2 Πολλαπλή Γραμμική Παλινδρόμηση

Όπως αναλύσαμε στην ενότητα 4.4 για την πολλαπλή γραμμική παλινδρόμηση σκοπός δεν είναι να χρησιμοποιήσουμε όλες τις μεταβλητές αλλά να βρούμε αυτές που εξηγούν καλύτερα την μεταβολή στον αριθμό των σφαλμάτων. Για την επίτευξη αυτού του στόχου χρησιμοποιήσαμε βηματική παλινδρόμηση με επιλογή μεταβλητών προς τα εμπρός, όπου σε κάθε βήμα μπαίνει στην εξίσωση της παλινδρόμησης η μεταβλητή με το μικρότερο επίπεδο σημαντικότητας της στατιστικής F (P-Value of F) που είναι κάτω από ένα καθορισμένο όριο ή βγαίνει από την εξίσωση οποιαδήποτε μεταβλητή ξεπεράσει ένα προκαθορισμένο επίπεδο σημαντικότητας της στατιστικής F. Εμείς χρησιμοποιήσαμε σαν όριο εισόδου επίπεδο σημαντικότητας 5% και σαν όριο εξόδου το 10% όπως προτείνετε από διάφορα εγχειρίδια στατιστικής [006, 007].

		Βήμα									
		1	2	3	4	5	6	7	8	9	10
	Κριτήριο Πληροφορίας AICC	564,533	511,758	486,667	477,310	464,183	433,562	416,887	416,724	404,847	401,571
Μετρικές	LCOM	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	CE		✓	✓	✓	✓	✓	✓			
	CBM			✓	✓	✓	✓	✓	✓	✓	✓
	IC				✓	✓	✓	✓	✓	✓	✓
	WMC					✓	✓	✓	✓	✓	✓
	RFC						✓	✓	✓	✓	✓
	NPM							✓	✓	✓	✓
	LOC									✓	✓
	MOA										✓

Πίνακας 5.4: Σταδιακή Προσθήκη Μετρικών στην Προς τα Εμπρός Βηματική Παλινδρόμηση

Η διαδικασία που ακολουθήθηκε κατά τη βηματική παλινδρόμηση προς τα εμπρός παρουσιάζεται παραστατικά στον πίνακα 5.4, όπου επιπρόσθετα αναφέρουμε και την εξέλιξη του κριτηρίου πληροφορίας του Akaike. Παρατηρούμε ότι, ενώ στην εισαγωγή των πρώτων έξι μεταβλητών αυτό μειώθηκε αισθητά, στις επόμενες μεταβλητές είχαμε μια οριακή μείωση του. Μετά το δέκατο βήμα δεν υπήρχε κάποια μεταβλητή για προσθήκη στην εξίσωση της παλινδρόμησης οπότε και η διαδικασία τελείωσε. Λεπτομέρειες για κάθε βήμα που ακολουθήσαμε μπορούν να βρεθούν στη σχετική ενότητα του παραρτήματος Β.

	Συντελεστής	Τυπικό Σφάλμα	Τυποποιημένος Συντελεστής (Standardized Coefficient)	Επίπεδο Σημαντικότητας (P-Value)
Σταθερός Όρος	-0,551	0,192	-	0,004
WMC	-0,189	0,027	-1,412	0,000
RFC	0,050	0,006	0,678	0,000
LCOM	0,001	0,000	0,320	0,000
IC	-0,652	0,260	-0,148	0,013
CBM	0,390	0,085	0,270	0,000
NPM	0,157	0,028	0,580	0,000
MOA	0,204	0,089	0,900	0,022
LOC	0,002	0,000	0,595	0,000

Πίνακας 5.5: Αποτελέσματα Πολλαπλής Γραμμικής Παλινδρόμησης με Επιλογή προς τα Εμπρός

Τα τελικά αποτελέσματα εκθέτονται στον πίνακα 5.5, όπου η εξίσωση που προκύπτει είναι:

$$\text{Αριθμός Σφαλμάτων Κλάσης} = -0{,}511 - 0{,}189 * \text{WMC} + 0{,}05 * \text{RCF} + 0{,}001 * \text{LCOM} - 0{,}652 * \text{IC} + 0{,}39 * \text{CMB} + 0{,}157 * \text{NPM} + 0{,}204 * \text{MOA} + 0{,}002 * \text{LOC} \quad (5.2)$$

Σε επίπεδο στατιστικής σημαντικότητας 5% όλοι οι συντελεστές της εξίσωσης 5.2 είναι στατιστικά σημαντικοί αλλά αν το περιορίσουμε στο πιο αυστηρό 1% τότε οι μετρικές IC και MOA δεν έχουν στατιστικά σημαντικό συντελεστή (δηλαδή θα μπορούσε να είναι μηδέν). Αν όμως τις βγάζαμε και επαναλαμβάναμε την βηματική παλινδρόμηση, θα παίρναμε σαν τελικό μοντέλο ένα πολύ χειρότερο μοντέλο αφού ο συντελεστής καλής προσαρμογής R^2 θα μειωνόταν στο 0,394 από το 0,757 που έχει το επιλεγμένο μοντέλο (περισσότερες αναλυτικές λεπτομέρειες υπάρχουν στο παράρτημα Β). Επιπλέον κάτι που αξίζει να παρατηρήσουμε είναι ότι σε σχέση με την απλή παλινδρόμηση όπου η κάθε μετρική ήταν μόνη της, ο συντελεστής των μετρικών WMC και IC έχει αλλάξει πρόσημο όπου από θετικό έχει γίνει αρνητικό.

Λαμβάνοντας υπόψη τους τυποποιημένους συντελεστές οι μετρικές με τη μεγαλύτερη επίδραση στον προσδιορισμό του αριθμού των σφαλμάτων είναι κατά φθίνουσα σειρά είναι οι WMC, MOA, RFC, LOC και NPM., ενώ αυτές με την μικρότερη είναι κατά αύξουσα σειρά οι IC, CMB και LCOM. Αν εξαιρέσουμε τις μετρικές της συλλογής του Martin, όλες οι άλλες συλλογές μετρικών έχουν και κάποιο εκπρόσωπο στο μοντέλο. Οι μετρικές CK έχουν τις WMC, RFC, LCOM, οι βελτιώσεις στις CK του Tang τις IC και CBM, το μοντέλο QMOOD τις NPM, MOA και από τις παραδοσιακές μετρικές έχουμε τον αριθμό γραμμών του πηγαίου κώδικα LOC.

5.3 Δυαδική Λογιστική Παλινδρόμηση

Συνεχίζουμε τώρα την μοντελοποίηση με εξαρτημένη μεταβλητή την ύπαρξη ή όχι ενός σφάλματος σε μια κλάση. Θεωρούμε ότι μια κλάση είναι προβληματική αν έχει γίνει τουλάχιστον μια διόρθωση κατά τη διάρκεια της έκδοσης 3.2 του προγράμματος jEdit. Ακολουθώντας την ίδια λογική με την προηγούμενη ενότητα αρχικά κάνουμε χρήση της απλής δυαδικής λογιστικής παλινδρόμησης για να απομονώσουμε τη σημαντικότητα κάθε μιας μετρικής ξεχωριστά. Μετά προχωράμε στην πολλαπλή δυαδική λογιστική παλινδρόμηση όπου καταλήγουμε στις μεταβλητές του μοντέλου με χρήση της τεχνικής επιλογή προς τα μπρος. Με τον τρόπο αυτό διερευνούμε και τη συνδυασμένη επίδραση των μετρικών για τη διάγνωση της ύπαρξης προβλήματος σε μια κλάση.

5.3.1 Απλή Δυαδική Λογιστική Παλινδρόμηση

Τα αποτελέσματα για όλες τις μετρικές παρουσιάζονται στον πίνακα 5.6, όπου για κάθε μια υπάρχει ο σταθερός όρος της εξίσωσης, ο συντελεστής της μετρικής και το επίπεδο σημαντικότητας του συντελεστή. Έτσι, π.χ. ο συντελεστής για την μετρική AMC δεν είναι στατιστικά σημαντικός ούτε καν στο επίπεδο σημαντικότητας 5% αφού έχει P-Value ίση με 0,084 και η εξίσωση που προκύπτει για την πρόβλεψη της πιθανότητας να είναι προβληματική μια κλάση είναι η εξής:

$$P(AMC) = \frac{e^{(-0,942+0,007*AMC)}}{1+e^{(-0,942+0,007*AMC)}} \quad (5.3)$$

Σε επίπεδο σημαντικότητας 5% οι μετρικές WMC, NOC, LCOM, AMC, και LOC δεν έχουν στατιστικά σημαντικούς συντελεστές και άρα δεν μας είναι ιδιαίτερα χρήσιμες για την πρόβλεψη σφάλματος σε μια κλάση. Αν περιορίσουμε το επίπεδο σημαντικότητας στο 1% τότε θα πρέπει να μην λάβουμε υπ' όψιν μας και τις μετρικές CA, NPM και MFA. Οπότε στην απλή δυαδική λογιστική παλινδρόμηση μας είναι χρήσιμες δώδεκα από τις είκοσι συνολικά μετρικές. Επιπλέον στον πίνακα 5.6 δίνουμε για κάθε μετρική το λόγο πιθανοτήτων (odds ratio) που προκύπτει, δηλαδή το λόγο του να είναι μια κλάση προβληματική προς το λόγο να μην είναι. Οπότε μπορούμε να ερμηνεύσουμε καλύτερα την επίδραση που έχει κάθε μετρική στην αύξηση της πιθανότητας σφάλματος π.χ. για την μετρική DIT ο λόγος πιθανοτήτων είναι 1,432 που σημαίνει ότι για κάθε μονάδα αύξησης του βάθους στο δέντρο της κληρονομικότητας η εξαρτημένη μεταβλητή δηλαδή η πιθανότητα σφάλματος στην κλάση αυξάνει κατά 1,432 φορές.

	Σταθερός Όρος (α)	Συντελεστής (β)	Τυπικό Σφάλμα (Standard Error)	Επίπεδο Συμαντικότητας (P-Value)	Λόγος Πιθανοτήτων (Odds Ratio)
WMC	-0,807	0,008	0,005	0,128	1,008
DIT	-1,789	0,359	0,064	0,000	1,432
NOC	-0,688	-0,044	0,065	0,499	0,957
CBO	-1,054	0,029	0,010	0,004	1,029
RFC	-1,485	0,021	0,004	0,000	1,022
LCOM	-0,75319	0,00029	0,00017	0,10000	1,00029
LCOM3	-0,051	-0,647	0,230	0,005	0,524
IC	-1,167	0,652	0,148	0,000	1,920
CBM	-1,171	0,324	0,071	0,000	1,382
AMC	-0,942	0,007	0,004	0,084	1,007
CA	-0,868	0,022	0,010	0,026	1,022
CE	-1,442	0,116	0,025	0,000	1,123
NPM	-0,919	0,029	0,012	0,014	1,030
DAM	-1,896	1,952	0,330	0,000	7,040
MOA	-1,092	0,421	0,098	0,000	1,523
MFA	-1,111	0,700	0,307	0,022	2,014
CAM	0,462	-2,660	0,627	0,000	0,070
LOC	-0,74862	0,00009	0,00009	0,29750	1,00009
CC_AVG	-1,505	0,449	0,104	0,000	1,567
CC_MAX	-1,000	0,048	0,018	0,009	1,049

Πίνακας 5.6: Αποτελέσματα Απλής Δυαδικής Λογιστικής Παλινδρόμησης για το Πρόγραμμα jEdit 3.2

	Συντελεστής	Τυπικό Σφάλμα	Επίπεδο Σημαντικότητας (P-Value)	Τυποποιημένος Συντελέστής (Stadardized Coefficient)	Λόγος Πιθανοτήτων (Odds Ratio)
Σταθερός Όρος	-3,508	0,419	0,000	-1,051	0,030
DIT	0,251	0,073	0,001	0,534	1,286
RFC	0,043	0,010	0,000	2,384	1,044
DAM	0,992	0,392	0,011	0,463	2,697
MOA	0,230	0,102	0,024	0,417	1,259
LOC	0,002	0,001	0,001	4,014	0,998
CC_MAX	0,088	0,027	0,001	1,049	1,091

Πίνακας 5.7: Αποτελέσματα Πολλαπλής Δυαδικής Λογιστικής Παλινδρόμησης με Επιλογή προς τα Εμπρός για το Πρόγραμμα jEdit 3.2

5.3.2 Πολλαπλή Δυαδική Λογιστική Παλινδρόμηση

Το μοντέλο που θα προκύψει από την εφαρμογή της πολλαπλής δυαδικής λογιστικής παλινδρόμησης επιδιώκουμε να έχει μόνο τις μετρικές που με στατιστικά σημαντικό τρόπο βελτιώνουν την πρόβλεψη του. Έτσι, όπως αναλύσαμε στην ενότητα 4.4 δεν θέλουμε να χρησιμοποιήσουμε όλες τις μεταβλητές αλλά να βρούμε αυτές που μπορούν να συμβάλουν στην ανίχνευση των σφαλμάτων σε μια κλάση. Για να το πετύχουμε αυτό εκτελέστηκε βηματική παλινδρόμηση με επιλογή μεταβλητών προς τα εμπρός, όπου σε κάθε βήμα προστίθεται στην εξίσωση της παλινδρόμησης η μεταβλητή που βελτιώνει περισσότερο την προβλεψιμότητα του μοντέλου ή βγαίνει από την εξίσωση οποιαδήποτε μεταβλητή δεν προσφέρει στατιστικά σημαντικά. Η διαδικασία ολοκληρώθηκε σε οκτώ βήματα όπου τα αναλυτικά στοιχεία για κάθε βήμα υπάρχουν στο παράρτημα Β. Τα τελικά αποτελέσματα έχουν συγκεντρωθεί στον πίνακα 5.7, όπου η εξίσωση που προκύπτει είναι:

$$P(DIT, RFC, DAM, MOA, LOC, CC\_MAX) = \frac{e^{(-3{,}508+0{,}251*DIT+0{,}043*RFC+0{,}992*DAM+0{,}23*MOA+0{,}002*LOC+0{,}088*CC\_MAX)}}{1+e^{(-3{,}508+0{,}251*DIT+0{,}043*RFC+0{,}992*DAM+0{,}23*MOA+0{,}002*LOC+0{,}088*CC\_MAX)}} \quad (5.4)$$

Όλοι οι συντελεστές της εξίσωσης 5.4 είναι στατιστικά σημαντικοί σε επίπεδο στατιστικής σημαντικότητας 5%. Όμως αν θέλουμε να είμαστε πιο αυστηροί και το περιορίσουμε στο 1% τότε οι μετρικές DAM (οριακά γιατί έχει P-Value 0,011) και MOA δεν έχουν στατιστικά σημαντικό συντελεστή (δηλαδή δεν μπορούμε να πούμε με 99% πιθανότητα ότι ο συντελεστής τους δεν είναι μηδέν). Αν υπολογίσουμε τη δυαδική λογιστική παλινδρόμηση χωρίς αυτές τις δύο μετρικές τότε το μοντέλο έχει λίγο χαμηλότερη προβλεψιμότητα και έτσι επιλέξαμε να τις αφήσουμε στο μοντέλο μας. Όμως αν κάποιος δεν έχει διαθέσιμες τις DAM και MOA αλλά έχει τις υπόλοιπες τέσσερις DIT, RFC, LOC και CC_MAX μπορεί να χρησιμοποιήσει το μοντέλο με αυτές τις τέσσερις μεταβλητές (λεπτομέρειες υπάρχουν στο παράρτημα Β.) Σε αντίθεση με τη γραμμική παλινδρόμηση όπου από την απλή στην πολλαπλή περίπτωση είχαμε μετρικές που άλλαξαν πρόσημο στη λογιστική παλινδρόμηση όλες οι μετρικές κράτησαν το ίδιο πρόσημο που είχαν και στην απλή λογιστική παλινδρόμηση. Από τον λόγο των πιθανοτήτων (odds ratio) καταλαβαίνουμε ότι οι μετρικές με τη μεγαλύτερη επίδραση στην πιθανότητα ύπαρξης σφαλμάτων είναι κατά φθίνουσα σειρά οι DAM, DIT και MOA, ενώ αυτές με την μικρότερη είναι κατά αύξουσα σειρά οι LOC, RFC και CC_MAX. Από τις διαφορετικές κατηγορίες μετρικών έχουμε στο μοντέλο από την συλλογή CK τις DIT και RFC, από το QMOOD τις DAM και MOA, και από τις παραδοσιακές μετρικές έχουμε τον αριθμό γραμμών του πηγαίου κώδικα LOC και τη μέγιστη κυκλωματική πολυπλοκότητα CC_MAX.

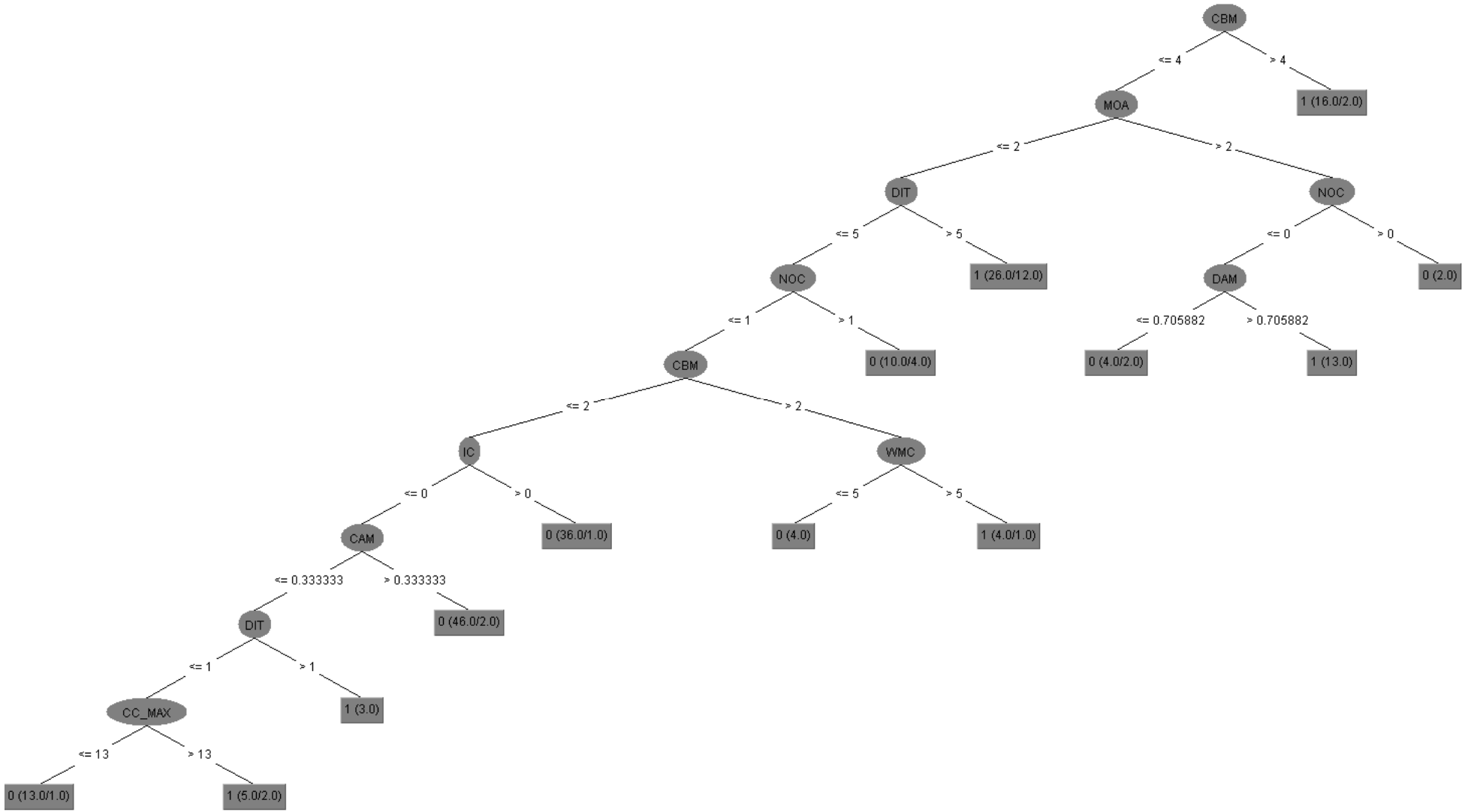
5.4 Δέντρο Απόφασης

Σε αυτή την ενότητα θα εξετάσουμε την αποτελεσματικότητα από τη χρήση των δέντρων απόφασης και πιο συγκεκριμένα της υλοποίησης του αλγορίθμου C4.5 [008] από το πρόγραμμα ανοικτού κώδικα WEKA. Σαν παράγοντα εμπιστοσύνης (confidence factor) για το κλάδεμα του δέντρου χρησιμοποιήσαμε το 0,25 και επιλέξαμε να γίνεται εξομάλυνση της εκτίμησης Laplace στα φύλλα του δέντρου. Επιπλέον, επειδή ο αρχικός αλγόριθμος C4.5 είναι αρκετά επιθετικός στο κλάδεμα του δέντρου επιλέξαμε να χρησιμοποιηθεί μια παραλλαγή του που κάνει μικρότερα λάθη κατά το κλάδεμα του δέντρου απόφασης (Το WEKA ονομάζει αυτή την επιλογή reducedErrorPruning). Για την αποτίμηση κάθε μοντέλου που προέκυψε επιλέξαμε να χρησιμοποιήσαμε τη διασταυρωμένη επικύρωση σε 10 μέρη (10-fold cross validation).

	Συνολικό Ποσοστό Σωστής Κατηγοριοποίησης (Total Model Accurancy)	Αναλογία Σωστών Θετικών Προβλέψεων (TP Rate)	Αναλογία Λανθασμένων Θετικών Προβλέψεων (FP Rate)	Ακρίβεια (Precision)	Ανάκληση (Recall)	Αρμονικός Διαιρέτης (F-Measure)	Περιοχή Κάτω από την Καμπύλη ROC (AUC)
WMC	68,38%	0,433	0,192	0,527	0,433	0,476	0,633
DIT	73,90%	0,511	0,148	0,630	0,511	0,564	0,651
NOC	66,91%	0,000	0,000	0,000	0,000	0,000	0,496
CBO	66,54%	0,000	0,005	0,000	0,000	0,000	0,491
RFC	70,96%	0,667	0,269	0,550	0,667	0,603	0,723
LCOM	73,53%	0,522	0,159	0,618	0,522	0,566	0,657
LCOM3	70,22%	0,711	0,302	0,538	0,711	0,612	0,675
IC	72,43%	0,178	0,005	0,941	0,178	0,299	0,560
CBM	75,00%	0,378	0,066	0,739	0,378	0,500	0,611
AMC	66,91%	0,000	0,000	0,000	0,000	0,000	0,496
CA	66,91%	0,000	0,000	0,000	0,000	0,000	0,496
CE	68,01%	0,078	0,022	0,636	0,078	0,139	0,519
NPM	64,34%	0,200	0,137	0,419	0,200	0,271	0,541
DAM	64,34%	0,156	0,115	0,400	0,156	0,224	0,532
MOA	67,65%	0,300	0,137	0,519	0,300	0,380	0,653
MFA	73,89%	0,322	0,055	0,744	0,322	0,450	0,630
CAM	70,59%	0,533	0,209	0,558	0,533	0,545	0,608
LOC	73,16%	0,400	0,104	0,655	0,400	0,497	0,622
CC_AVG	67,28%	0,144	0,066	0,520	0,144	0,226	0,564
CC_MAX	73,16%	0,533	0,170	0,608	0,533	0,568	0,648
Όλες Μαζί	74,27%	0,600	0,187	0,614	0,600	0,607	0,741

Πίνακας 5.8: Αποτελέσματα Δέντρου Απόφασης για το Πρόγραμμα Edit 3.2

Αρχικά υπολογίσαμε ένα δέντρο απόφασης για κάθε μια μετρική ξεχωριστά και στη συνέχεια ένα δέντρο απόφασης με όλες μαζί τις μετρικές. Στον πίνακα 5.8 συγκεντρώσαμε τις πιο σημαντικές παραμέτρους για κάθε δέντρο απόφασης που προέκυψε, δηλαδή το συνολικό ποσοστό σωστής κατηγοριοποίησης, την αναλογία σωστών θετικών προβλέψεων (TP rate), την αναλογία λανθασμένων θετικών προβλέψεων (FP rate), την ακρίβεια (precision), την ανάκληση (recall) , τον αρμονικό διαιρέτη (F-Measure) και την περιοχή κάτω από την καμπύλη ROC (AUC).



Σχήμα 5.4: Δέντρο Απόφασης με Είσοδο Όλες τις Μετρικές για το Πρόγραμμα jEdit 3.2

Σε γενικές γραμμές οι επιδόσεις όλων των δέντρων αποφάσεων που προέκυψαν δεν είναι ιδιαίτερα ενθαρρυντικές. Υπήρχαν μετρικές όπως η NOC, CBO, AMC και CA που έδωσαν μηδενική αναλογία θετικών προβλέψεων δηλαδή δεν έκαναν καμιά σωστή πρόβλεψη σε ότι αφορά την ύπαρξη σφάλματος σε μια κλάση. Αλλά και η πλειοψηφία από τις υπόλοιπες μετρικές στο TP rate δεν έδωσε καλά αποτελέσματα αφού αυτό είναι συνήθως μικρότερο του 0,5. Από τις ελάχιστες μετρικές που ξεχώρισαν είναι η RFC που στην αναλογία σωστών θετικών προβλέψεων έδωσε 0,667 δηλαδή από όλες τις κλάσεις με σφάλμα βρήκε το 66,7% και στην περιοχή κάτω από την καμπύλη (AUC) έχει 0,723. Αυτή η επίδοση μάλιστα είναι λίγο μικρότερη από το 0,741 δηλαδή αυτή του μοντέλου που είχε σαν είσοδο όλες τις μετρικές και που είχε την καλύτερη AUC. Αν και κάποιος θα μπορούσε να μπει στον πειρασμό να χρησιμοποιήσει μόνο το δέντρο που προκύπτει από την RFC, δεν θα το προτείναμε για τρεις λόγους. Πρώτον, έχει υψηλότερη αναλογία λανθασμένων θετικών προβλέψεων δηλαδή μας δίνει 50% περισσότερες κλάσεις λανθασμένα σαν προβληματικές σε σχέση με το δέντρο με όλες τις μετρικές (FP rate 0,269 έναντι 0,187). Δεύτερον, η ακρίβεια του είναι 0,55 ενώ του δέντρου με όλες τις μετρικές είναι υψηλότερη στο 0,614. Τρίτον, το συνολικό ποσοστό σωστής κατηγοριοποίησης του είναι 70,96% έναντι 74,27%. Οπότε θα προτείναμε τη χρήση του δέντρου με την RFC μόνο στην περίπτωση που δεν υπάρχουν διαθέσιμες οι άλλες μετρικές, ως μια καλή πρώτη προσέγγιση.

Η γραφική αναπαράσταση του δέντρου που προέκυψε με είσοδο όλες τις μετρικές παρουσιάζεται στο σχήμα 5.4. Ο ενδιαφερόμενος αναγνώστης μπορεί να μελετήσει λεπτομέρειες για όλα τα υπόλοιπα δέντρα που προέκυψαν στο παράρτημα Γ. Μια πολύ ενδιαφέρουσα δυνατότητα που μας δίνουν τα δέντρα απόφασης είναι ότι μπορούμε από την γραφική τους αναπαράσταση να δημιουργήσουμε κανόνες της μορφής "**ΑΝ** Χ **ΤΟΤΕ** Υ". Για να δημιουργήσουμε ένα τέτοιο κανόνα αρκεί να ξεκινήσουμε από τη ρίζα του δέντρου και να κατέβουμε προς τα κάτω μέχρι να φτάσουμε σε ένα φύλλο που δεν έχει απογόνους. Οπότε από το δέντρο του σχήματος 5.4 προκύπτουν συνολικά δεκατρείς κανόνες που θα μπορούσαν εύκολα να γίνουν κατανοητοί από έναν άνθρωπο ή να υλοποιηθούν σε κάποιο πρόγραμμα. Ενδεικτικά δείχνουμε δύο από τους δεκατρείς κανόνες του δέντρου, έναν για την περίπτωση που η πρόβλεψη είναι ότι δεν υπάρχει σφάλμα και έναν με την πρόβλεψη ότι υπάρχει σφάλμα:

- **ΑΝ** (CMB <= 4 **ΚΑΙ** MOA > 2 **ΚΑΙ** NOC > 0) **ΤΟΤΕ** η κλάση δεν είναι προβληματική

- **ΑΝ** (CMB > 4 **ΚΑΙ** MOA <= 2 **ΚΑΙ** DIT <= 5 **ΚΑΙ** NOC <= 1 **ΚΑΙ** CMB > 2 **ΚΑΙ** WMC>5) **ΤΟΤΕ** η κλάση είναι προβληματική

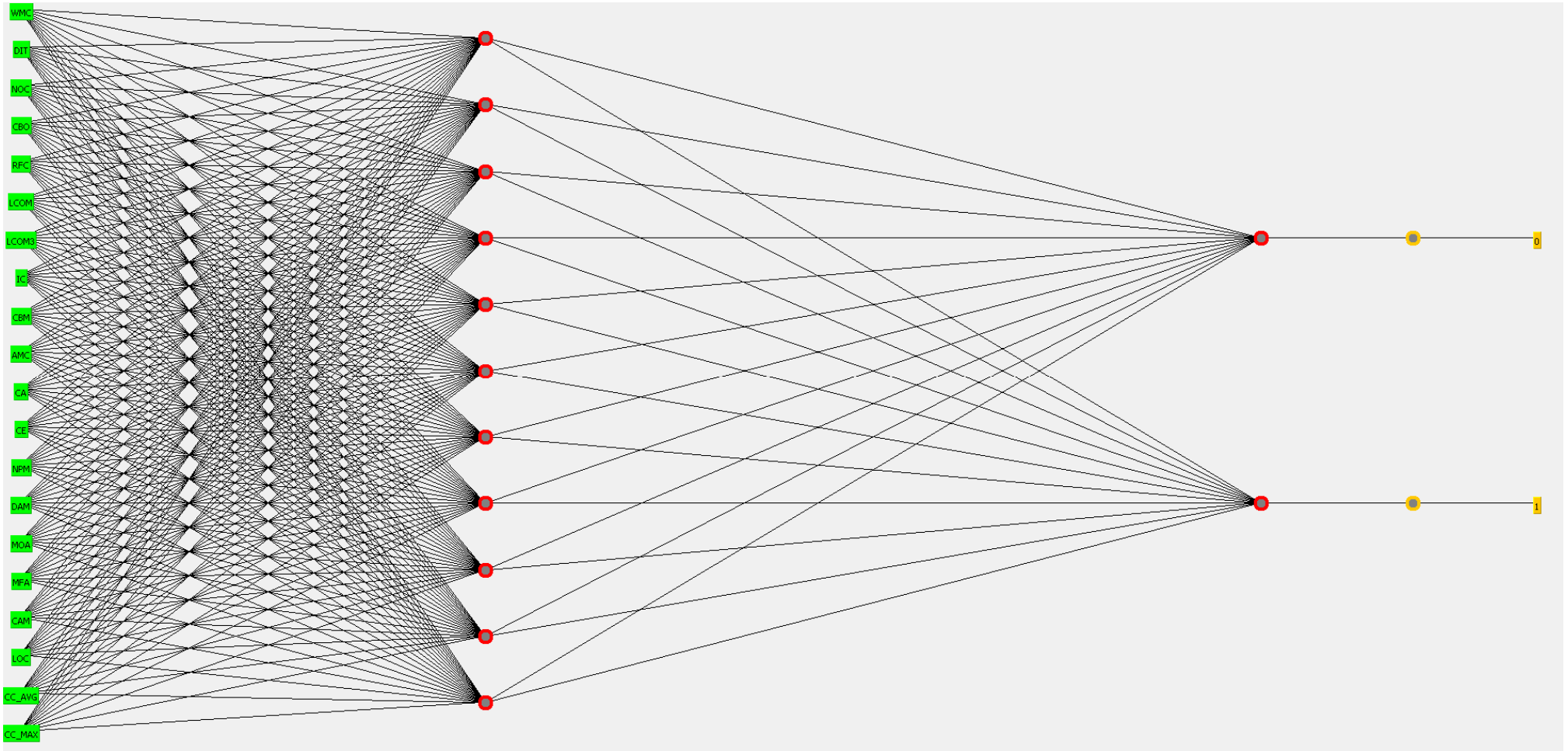
5.5 Τεχνητό Νευρωνικό Δίκτυο

Από την κατηγορία της μηχανικής μάθησης θα εξετάσουμε τώρα ένα μοντέλο που ανήκει στην ευρύτερη κατηγορία των τεχνητών νευρωνικών δικτύων [009] και πιο συγκεκριμένα πρόκειται για ένα πολυεπίπεδο τεχνητό νευρωνικό δίκτυο αισθητήρα (perceptron) που εκπαιδεύεται με την μέθοδο της πίσω διάδοσης του λάθους [010, 011]. Για να επιτύχουμε τα καλύτερα δυνατά αποτελέσματα, μετά από αρκετές δοκιμές και στηριζόμενοι στη μελέτη της σχετικής βιβλιογραφίας [012, 013] επιλέξαμε η εκπαίδευση να σταματάει στις πέντε χιλιάδες επαναλήψεις (από τις 500 που προτείνει το WEKA), σαν ρυθμό μάθησης (learning rate) επιλέξαμε το 0,3 και σαν σταθερά ορμής (momentum) το 0,2.

	Συνολικό Ποσοστό Σωστής Κατηγοριοποίησης (Total Model Accurancy)	Αναλογία Σωστών Θετικών Προβλέψεων (TP Rate)	Αναλογία Λανθασμένων Θετικών Προβλέψεων (FP Rate)	Ακρίβεια (Precision)	Ανάκλυση (Recall)	Αρμονικός Διαιρέτης (F-Measure)	Περιοχή Κάτω από την Καμπύλη ROC (AUC)
WMC	68,75%	0,189	0,066	0,586	0,189	0,286	0,705
DIT	73,89%	0,511	0,148	0,630	0,511	0,564	0,693
NOC	66,91%	0,000	0,000	0,000	0,000	0,000	0,497
CBO	65,80%	0,078	0,055	0,412	0,078	0,131	0,622
RFC	74,63%	0,578	0,170	0,627	0,578	0,601	0,812
LCOM	66,54%	0,011	0,001	0,330	0,011	0,022	0,529
LCOM3	73,16%	0,544	0,176	0,605	0,544	0,573	0,736
IC	71,69%	0,233	0,044	0,724	0,233	0,353	0,602
CBM	74,63%	0,322	0,044	0,784	0,322	0,457	0,680
AMC	64,70%	0,122	0,093	0,393	0,122	0,186	0,660
CA	67,28%	0,033	0,011	0,600	0,033	0,063	0,528
CE	66,91%	0,278	0,137	0,500	0,278	0,357	0,713
NPM	65,07%	0,078	0,066	0,368	0,078	0,128	0,619
DAM	66,54%	0,322	0,165	0,492	0,322	0,389	0,685
MOA	72,06%	0,389	0,115	0,625	0,389	0,479	0,674
MFA	66,18%	0,078	0,049	0,438	0,078	0,132	0,572
CAM	70,59%	0,400	0,143	0,581	0,400	0,474	0,662
LOC	67,65%	0,044	0,011	0,667	0,044	0,083	0,544
CC_AVG	67,27%	0,289	0,137	0,510	0,289	0,396	0,696
CC_MAX	72,06%	0,400	0,121	0,621	0,400	0,486	0,719
Όλες Μαζί	77,57%	0,633	0,154	0,671	0,633	0,651	0,793

Πίνακας 5.9: Αποτελέσματα Τεχνητού Νευρωνικού Δικτύου για το Πρόγραμμα Edit 3.2

Υπολογίσαμε ένα νευρωνικό δίκτυο για κάθε μια μετρική ξεχωριστά και παρουσιάζουμε στον πίνακα 5.9 συγκεντρωτικά τα πιο σημαντικά αποτελέσματα δηλαδή το συνολικό ποσοστό σωστής κατηγοριοποίησης, την αναλογία σωστών θετικών προβλέψεων, την αναλογία λανθασμένων θετικών προβλέψεων, την ακρίβεια, την ανάκληση, τον αρμονικό διαιρέτη και την περιοχή κάτω από την καμπύλη ROC (AUC). Ο ενδιαφερόμενος αναγνώστης μπορεί να μελετήσει και άλλα αποτελέσματα όπως είναι π.χ. ο πίνακας σύγχυσης(confusion matrix) στο παράρτημα Γ.



Εικόνα 5.1: Κατασκευή του Τεχνητού Νευρωνικού Δικτύου με Εισόδους Όλες τις Μετρικές στο WEKA

Επιπλέον, έγιναν αρκετές δοκιμές και για την περίπτωση που σαν είσοδο είχαμε όλες τις μετρικές μαζί. Τελικά καταλήξαμε σε δύο επίπεδα, όπου στο πρώτο επίπεδο έχουμε έντεκα αισθητήρες που είναι το άθροισμα των εισόδων και εξόδων δια δύο. Στο δεύτερο επίπεδο έχουμε δύο αισθητήρες που συνδέονται με όλους τους αισθητήρες του πρώτου επιπέδου και η έξοδος τους αποτελεί και την έξοδο του νευρωνικού δικτύου. Δοκιμάσαμε και πολύ πιο πολύπλοκα νευρωνικά δίκτυα που είχαν είτε πολύ περισσότερους αισθητήρες σε κάθε επίπεδο είτε περισσότερα κρυφά επίπεδα αλλά δεν μας έδωσαν καλύτερα αποτελέσματα. Το WEKA έχει επιλογή για τη γραφική αναπαράσταση της αρχιτεκτονικής του νευρωνικού δικτύου και ο χρήστης μπορεί να κάνει τις όποιες τροποποιήσεις επιθυμεί πριν την έναρξη της εκπαίδευσης του. Στην εικόνα 5.1 παρουσιάζουμε το νευρωνικό δίκτυο που καταλήξαμε για την περίπτωση που είχαμε ως είσοδο όλες τις μετρικές. Τα συγκεντρωτικά αποτελέσματα βρίσκονται στον πίνακα 5.9 και τα αναλυτικά υπάρχουν στο παράρτημα Γ. Ακολουθήσαμε την ίδια πρακτική με την αποτίμηση των μοντέλων του δέντρου απόφασης, δηλαδή για την αποτίμηση της καταλληλότητας των προβλέψεων κάθε νευρωνικού δικτύου χρησιμοποιήσαμε την διασταυρωμένη επικύρωση σε 10 μέρη.

Με μια προσεκτική μελέτη του πίνακα 5.9 σε σχέση με αυτά του δέντρου αποφάσεων (βλέπε πίνακα 5.8) προκύπτει το συμπέρασμα ότι το νευρωνικό δίκτυο έχει αρκετά καλύτερη προσαρμογή. Αν χρησιμοποιήσουμε σαν κριτήριο την περιοχή κάτω από την καμπύλη ROC (AUC) τότε έχουμε πέντε μετρικές με AUC μεγαλύτερη του 0,7 ενώ στο δέντρο απόφασης είχαμε μόνο μια τέτοια περίπτωση. Μάλιστα η μετρική RFC παραμένει πολύ καλή με ακριβώς την ίδια AUC όπως και στο δέντρο απόφασης (0,812) που είναι υψηλότερη από αυτή που πετυχαίνει το νευρωνικό δίκτυο όταν έχει σαν εισόδους όλες τις μετρικές. Όμως, το νευρωνικό δίκτυο που προκύπτει μόνο με χρήση της RFC στις υπόλοιπες παραμέτρους καλής προσαρμογής όπως είναι το συνολικό ποσοστό σωστής κατηγοριοποίησης, αναλογία σωστών θετικών προβλέψεων, αναλογία λανθασμένων θετικών προβλέψεων, ακρίβεια, ανάκληση και αρμονικός διαιρέτης, έχει χαμηλότερα νούμερα από αντίστοιχο που έχει εισόδους όλες τις μετρικές. Από την άλλη πλευρά υπάρχουν περιπτώσεις όπου είτε η AUC είναι κοντά στο 0,5 είτε η αναλογία σωστών θετικών προβλέψεων είναι πολύ χαμηλή και μας αποτρέπουν να χρησιμοποιήσουμε αυτές τις μετρικές. Πιο συγκεκριμένα είναι οι WMC, NOC, CBO, LCOM, AMC, CA, NPM, MFA και LOC. Δηλαδή περίπου τα μισά μοντέλα που προκύπτουν από την χρήση του πολυεπίπεδου τεχνητού νευρωνικού δικτύου αισθητήρα με μέθοδο εκπαίδευσης την πίσω διάδοση του λάθους δεν μπορούν να χρησιμοποιηθούν πρακτικά. Οπότε όπως και στην περίπτωση με το δέντρο απόφασης προτείνουμε την χρησιμοποίηση του μοντέλου που έχει όλες τις μετρικές σαν εισόδους.

5.6 Σχολιασμός και Σύγκριση με Άλλες Σχετικές Μελέτες

Σε όλες τις προηγούμενες ενότητες περιγράψαμε τα αποτελέσματα για κάθε ένα μοντέλο που αξιολογήσαμε, χωρίς να κάνουμε ιδιαίτερο σχολιασμό και σύγκριση με άλλες σχετικές μελέτες. Στην πρώτη υποενότητα θα σχολιάσουμε τα αποτελέσματα για κάθε διαφορετική μέθοδο μοντελοποίησης που χρησιμοποιήσαμε και θα τα συγκρίνουμε με παρόμοιες μελέτες. Στη δεύτερη υποενότητα θα κάνουμε ιδιαίτερη αναφορά στην συλλογή μετρικών CK και θα συγκρίνουμε τα δικά μας αποτελέσματα με αυτά των πιο σημαντικών ερευνητών στο πεδίο της πρόβλεψης σφαλμάτων με χρήση αντικειμενοστρεφών μετρικών.

5.6.1 Σύγκριση Αποτελεσμάτων Στατιστικής Ανάλυσης με Μηχανικής Μάθησης

Στον πίνακα 5.10 παρουσιάζουμε τα συγκεντρωτικά αποτελέσματα για το μοντέλο της πολλαπλής δυαδικής λογιστικής παλινδρόμησης, του δέντρου απόφασης με είσοδο όλες τις μετρικές και του τεχνητού νευρωνικού δικτύου με επίσης είσοδο όλες τις μετρικές. Αυτά είναι τα τρία καλύτερα μοντέλα που προέκυψαν στα πλαίσια εφαρμογής της μεθοδολογίας που περιγράφηκε αναλυτικά στο προηγούμενο κεφάλαιο στον πηγαίο κώδικα της έκδοσης 3.2 του προγράμματος jEdit.

	Συνολικό Ποσοστό Σωστής Κατηγοριοποίησης (Total Model Accurancy)	Αναλογία Σωστών Θετικών Προβλέψεων (TP Rate)	Αναλογία Λανθασμένων Θετικών Προβλέψεων (FP Rate)	Ακρίβεια (Precision)	Ανάκληση (Recall)	Αρμονικός Διαιρέτης (F-Measure)	Περιοχή Κάτω από την Καμπύλη ROC (AUC)
Λογιστική Παλινδρόμηση	80,88%	0,622	0,099	0,757	0,622	0,683	0,841
Δέντρο Απόφασης	74,27%	0,600	0,187	0,614	0,600	0,607	0,741
Τεχνητό Νευρωνικό Δίκτυο	77,57%	0,633	0,154	0,671	0,633	0,651	0,793

Πίνακας 5.10: Συγκριτικά Αποτελέσματα Στατιστικής Ανάλυσης με Μηχανικής Μάθησης

Μεταξύ αυτών των τριών μοντέλων, τις καλύτερες επιδόσεις έχει το μοντέλο της δυαδικής πολλαπλής λογιστικής παλινδρόμησης. Μπορεί να έχει λίγο χαμηλότερη ανάκληση και αναλογία σωστών θετικών προβλέψεων σε σχέση με το τεχνητό νευρωνικό δίκτυο (0,622 έναντι 0,633) αλλά σε όλα τα υπόλοιπα επέδειξε ανώτερα αποτελέσματα. Συγκεκριμένα, η αναλογία λανθασμένων θετικών προβλέψεων είναι 0,099 έναντι 0,154, κάτι που σημαίνει ότι μόνο μια στις δέκα κλάσεις όπου θα κάνει θετική πρόβλεψη ότι υπάρχει σφάλμα, τελικά δεν θα έχει σφάλμα (FP rate). Η ακρίβεια της παλινδρόμησης είναι 0,757 έναντι 0,671 του νευρωνικού δικτύου, ο αρμονικός διαιρέτης 0,683 έναντι 0,651 και η περιοχή κάτω από την καμπύλη ROC είναι 0,841 έναντι 0,793. Τα χειρότερα αποτελέσματα έστω και με όχι μεγάλη διαφορά από του νευρωνικού δικτύου έχει το δέντρο απόφασης π.χ. η περιοχή κάτω από την καμπύλη ROC είναι 0,741 έναντι 0,793 του νευρωνικού δικτύου δηλαδή μόλις 0,05 λιγότερο.

Με βάση τα παραπάνω προκύπτει το συμπέρασμα ότι οι στατιστικές τεχνικές που εφαρμόσαμε έδωσαν καλύτερα αποτελέσματα από αυτά της μηχανικής μάθησης. Βέβαια η διαφορά δεν είναι μεγάλη και κάποιος θα μπορούσε πιθανώς να πετύχει καλύτερα αποτελέσματα αν είχε χρησιμοποιήσει άλλες πιο προχωρημένες μεθόδους μηχανικής μάθησης όπως είναι οι μπεϋσιανοί ταξινομητές (Bayesian Classifiers) ή οι μηχανές διανυσμάτων υποστήριξης (support vector machines). Ο Zhou και Leung [023] χρησιμοποίησαν λογιστική παλινδρόμηση, τυχαία δάση (random forests) και μπεϋζιανούς ταξινομητές για την εύρεση σφαλμάτων που τα είχαν χωρίσει σε δυο κατηγορίες, μικρής και μεγάλης σπουδαιότητας. Η λογιστική παλινδρόμηση έδωσε καλύτερα αποτελέσματα από τις δυο τεχνικές μηχανικής μάθησης. Ο Pai [024] χρησιμοποίησε μπεϋσιανούς ταξινομητές για την εύρεση σφαλμάτων σε κλάσεις της C++ από τη δημόσια συλλογή δεδομένων KC1 της NASA, όπου η αναλογία σωστών θετικών προβλέψεων, η αναλογία λανθασμένων θετικών προβλέψεων και η ακρίβεια που αναφέρεται στο άρθρο είναι μικρότερη από αυτή που πέτυχε το δικό μας μοντέλο λογιστικής παλινδρόμησης.

Από την άλλη πλευρά οι Singh et al. [022] χρησιμοποιώντας τα ίδια δεδομένα με τον Pai αλλά έχοντας χωρίσει τα σφάλματα σε τρεις διαφορετικές κατηγορίες σοβαρότητας, καταλήγουν ότι οι μέθοδοι της μηχανικής μάθησης έχουν καλύτερα αποτελέσματα από τις στατιστικές μεθόδους. Μάλιστα κάποια μοντέλα από τα δέντρα απόφασης έχουν πολύ καλή προσαρμογή στα δεδομένα αφού η περιοχή κάτω από την καμπύλη ROC φτάνει το 0,888. Βέβαια αυτό δεν είναι μεγάλη διαφορά με το 0,841 που φτάνει η λογιστική παλινδρόμηση στην έρευνά μας αλλά εξακολουθεί να είναι καλύτερο αποτέλεσμα. Στο τεχνητό νευρωνικό δίκτυο έχουμε παρόμοια αποτελέσματα αφού το καλύτερο τους έχει AUC 0,809 και το δικό μας έχει 0,793, η ακρίβεια τους κυμαίνεται από 0,683 έως 0,718 με την έρευνα μας να έχει φτάσει μέχρι 0,671. Οι Malhotra et al [025] στην εργασία τους συνέκριναν τα αποτελέσματα της πολλαπλής λογιστικής παλινδρόμησης με πέντε μεθόδους τεχνικής μάθησης χρησιμοποιώντας σαν κριτήριο επιλογής του καλύτερου μοντέλου την περιοχή κάτω από την καμπύλη ROC. Η μέθοδος του τυχαίου δάσους, της προσαρμοστικής ώθησης (AdaBoost) και του σακουλιάσματος (bagging) έδωσαν πολύ καλά αποτελέσματα με AUC 0,875, 0,861 και 0,876 αντίστοιχα. Το τεχνητό νευρωνικό δίκτυο τους, είχε παρόμοιο αποτέλεσμα με το δικό μας αλλά η λογιστική τους παλινδρόμηση πέτυχε AUC 0,791 ενώ εμείς 0,841. Εν κατακλείδι, δεν φαίνεται να είναι ξεκάθαρο αν οι μέθοδοι της μηχανικής μάθησης δίνουν τελικά καλύτερα αποτελέσματα από τις στατιστικές μεθόδους. Εκεί που φαίνεται να συγκλίνουν οι περισσότερες εργασίες είναι ότι μέχρι τώρα δεν υπάρχει πολύ μεγάλη διαφορά στα αποτελέσματα των δύο μεθόδων αλλά υπάρχουν και άλλες μέθοδοι από την μηχανική μάθηση που θα μπορούσαν να διερευνηθούν για την πρόβλεψη σφαλμάτων στο λογισμικό.

5.6.2 Σχολιασμός Μετρικών CK και Σύγκριση Αποτελεσμάτων με Άλλες Μελέτες

Η συλλογή των μετρικών CK επιβεβαίωσε τη σημαντικότητα της στην έρευνα μας, αφού οι τέσσερις από τις έξι μετρικές βρέθηκαν να επηρεάζουν στατιστικά σημαντικά την πιθανότητα ύπαρξης σφάλματος σε μία κλάση. Μάλιστα υπάρχει πλήθος εμπειρικών μελετών που έχουν χρησιμοποιήσει τις συγκεκριμένες μετρικές. Στον πίνακα 5.11 για τη σύγκριση των αποτελεσμάτων προσθέσαμε και τον αριθμό των γραμμών του πηγαίου κώδικα LOC που εξετάζεται στις περισσότερες αντίστοιχες έρευνες και τα συγκρίνουμε με τα δικά μας. Ακολουθεί μια ανάλυση ξεχωριστά ανά μετρική:

- **WMC**: Είναι μία μετρική για τη μέτρηση της πολυπλοκότητας της κλάσης. Στη γραμμική παλινδρόμηση βρέθηκε να είναι στατιστικά πολύ σημαντική (P-Value 0,000) με τον αριθμό των λαθών σε μια κλάση αλλά αντίθετα στη λογιστική παλινδρόμηση δεν ήταν στατιστικά σημαντική (P-Value 0,128). Όπως και στις δυο μεθόδους μηχανικής μάθησης φάνηκε να μπορεί να εξηγήσει σε ικανοποιητικό βαθμό την ύπαρξη σφάλματος σε μία κλάση π.χ. στο νευρωνικό δίκτυο έχει AUE 0,705. Έτσι καταλήγουμε για την μελέτη μας ότι υπάρχει θετική συσχέτιση με την ύπαρξη λάθους σε μία κλάση, κάτι που έρχεται σε συμφωνία σχεδόν με όλες τις μελέτες. Οι Basili et al. [015] ανάλυσαν οκτώ έργα φοιτητών και βρήκαν θετική συσχέτιση του WMC με την ύπαρξη σφάλματος. Οι Emam et al. [017] βρήκαν παρόμοια αποτελέσματα αλλά η σημαντικότητα της μειώθηκε όταν προστέθηκε και το μέγεθος της κλάσης στο μοντέλο, κάτι που δεν επαληθεύεται από την έρευνα μας. Η μετρική WMC ήταν ο καλύτερος παράγοντας για την πρόβλεψη σφάλματος στο πρόγραμμα ανοικτού λογισμικού Mozilla 1.6 [019] και αρκετά ικανοποιητικός στο επίσης ανοικτού κώδικα Java Development Kit [018].

- **DIT**: Είναι το βάθος του δέντρου της κληρονομικότητας. Στη γραμμική παλινδρόμηση δεν είναι στατιστικά σημαντική σε επίπεδο 1% αλλά είναι στο 5% (P-Value 0,03) ενώ αντίθετα στη λογιστική παλινδρόμηση είναι στατιστικά πολύ σημαντική (P-Value 0,000) με θετικό πρόσημο. Ομοίως, στις μεθόδους μηχανικής μάθησης μπορούσε να εξηγήσει την ύπαρξη σφάλματος σε μία κλάση σε έναν απλά ικανοποιητικό βαθμό. Για τους παραπάνω λόγους, καταλήγουμε ότι στην έρευνα μας φαίνεται να υπάρχει μία θετική σχέση μεταξύ της DIT και της ύπαρξης σφάλματος αλλά γενικότερα η μετρική αυτή δεν μας βοηθάει ιδιαίτερα στο να κάνουμε καλές προβλέψεις. Οι Basili et al. [015] βρήκαν ότι υπάρχει ισχυρή σχέση μεταξύ του βάθους και της ύπαρξης σφάλματος, ενώ οι Briand et al. [016] βρήκαν ότι υπάρχει μεν θετική σχέση αλλά όχι ιδιαίτερα σημαντική. Αντίθετα, οι

Olague et al. [020] που εξέτασαν τέσσερις εκδόσεις του προγράμματος ανοικτού κώδικα Mozilla Rhino, δεν βρήκαν να υπάρχει στατιστικά σημαντική σχέση του DIT με την ύπαρξη σφαλμάτων. Στο ίδιο συμπέρασμα κατέληξαν και οι Malhotra et al. [025] όταν εξέτασαν το πρόγραμμα ανοικτού κώδικα Apache POI.

- **NOC**: Είναι ο αριθμός των άμεσων απογόνων μιας κλάσης. Στη στατιστική ανάλυση έδωσε τα λιγότερο ικανοποιητικά αποτελέσματα από οποιαδήποτε άλλη μετρική. Συγκεκριμένα στη γραμμική παλινδρόμηση είχε P-Value 0,498 και στη λογιστική παλινδρόμηση P-Value 0,499. Το ίδιο επαναλήφθηκε και στη μηχανική μάθηση όπου πάλι με διαφορά ήταν η λιγότερο ικανοποιητική μεταβλητή, αφού πέτυχε να έχει και στις δύο μεθόδους που εξετάσαμε αναλογία σωστών θετικών προβλέψεων ίση με μηδέν (με άλλα λόγια δεν έκανε καμιά σωστή πρόβλεψη σχετικά με την ύπαρξη σφάλματος σε μια κλάση). Εύκολα προκύπτει από την έρευνα μας το συμπέρασμα ότι η μετρική NOC δεν μπορεί να χρησιμοποιηθεί για την πρόβλεψη σφαλμάτων. Οι Olague et al. [020] εξέτασαν τέσσερις εκδόσεις του Mozilla Rhino και βρήκαν ότι η συνεισφορά της μετρικής NOC για την πρόβλεψη σφαλμάτων είναι μηδενική. Στο ίδιο συμπέρασμα κατέληξαν και οι Gyimothy et al. [019] που εξέτασαν την έκδοση 1.6 του Mozilla. Αντίθετα, οι Basili et al. [015] ανάλυσαν οκτώ έργα φοιτητών και βρήκαν θετική συσχέτιση της NOC με την ύπαρξη σφάλματος σε μια κλάση. Θετική συσχέτιση αλλά όχι ιδιαίτερα σημαντική βρήκαν οι English et al. [018] που εξέτασαν τον πηγαίο κώδικα του Java Development Kit.

- **RFC**: Είναι μία μετρική που αναφέρεται στην απόκριση μιας κλάσης και παρουσίασε την καλύτερη εικόνα από οποιαδήποτε άλλη μετρική στην έρευνα μας. Τόσο στη γραμμική όσο και στη λογιστική παλινδρόμηση το επίπεδο σημαντικότητάς της ήταν άριστο (P-Value 0,000) και συμπεριλαμβάνεται στα μοντέλα που προέκυψαν από τη βηματική επιλογή προς τα εμπρός. Στο δέντρο απόφασης έδωσε περιοχή κάτω από την καμπύλη ROC 0,723 που ήταν η καλύτερη μεταξύ των είκοσι μετρικών και το ίδιο ισχύει στο τεχνητό νευρωνικό δίκτυο που η AUC ήταν 0,812. Αν έπρεπε να επιλέξουμε μόνο μία μετρική για να προβλέψουμε την ύπαρξη σφάλματος σε μια κλάση, θα επιλέγαμε αυτή. Οι Emam et al. [017] βρήκαν παρόμοια αποτελέσματα αλλά η σημαντικότητα της μειώθηκε όταν προστέθηκε και το μέγεθος της κλάσης στο μοντέλο, κάτι που δεν επαληθεύεται από την έρευνα μας. Η RFC βρέθηκε να είναι στατιστικά πολύ σημαντική στην πρόβλεψη σφαλμάτων έξι εκδόσεων του Mozilla Rhino [020]. Γενικότερα, όλες οι μελέτες που αναφέρονται στον συγκεντρωτικό πίνακα 5.11, άλλες σε μικρότερο και άλλες σε μεγαλύτερο βαθμό βρίσκουν ότι η μετρική RFC βοηθάει στην πρόβλεψη σφαλμάτων.

	Μέθοδος	Τύπος Δεδομένων	Γλώσσα Προγραμματισμού	WMC	DIT	NOC	CBO	RFC	LCOM	LOC
Aggrarwal et al. [014]	Παλινδρόμηση	Λογισμικό από Φοιτητές	Java	▲	—	—	▲	▲	▲	▲
Basili et al. [015]	Παλινδρόμηση	Λογισμικό από Φοιτητές	C++	▲	▲	▲	▲	▲	—	Δεν Εξετ.
Briand et al. [016]	Παλινδρόμηση	Λογισμικό από Φοιτητές	C++	▲	▲	▲	▲	▲	—	▲
Emam et al. [017]	Παλινδρόμηση	Εμπορικό Λογισμικό	C++	▲	—	Δεν Εξετ.	▲	▲	Δεν Εξετ.	▲
English et al. [018]	Παλινδρόμηση	Λογισμικό Ανοικτού Κώδικα	Java	▲	▲	▲	▲	▲	Δεν Εξετ.	▲
Gyimonthy et al. [019]	Παλινδρόμηση, Μαχανική Μάθηση	Λογισμικό Ανοικτού Κώδικα	C++	▲	▲	—	▲	▲	▲	▲
Olague et al. [020]	Παλινδρόμηση	Λογισμικό Ανοικτού Κώδικα	Java	▲	—	—	▲	▲	▲	Δεν Εξετ.
Shatnawi et al. [021]	Παλινδρόμηση	Λογισμικό Ανοικτού Κώδικα	Java	▲	—	—	▲	▲	Δεν Εξετ.	Δεν Εξετ.
Singh et al. [022]	Παλινδρόμηση, Μαχανική Μάθηση	Συλλογή Δεδομένων NASA	C++	▲	—	—	▲	▲	▲	▲
Tang et al. [005]	Παλινδρόμηση	Εμπορικό Λογισμικό	C++	▲	—	—	—	▲	Δεν Εξετ.	Δεν Εξετ.
Τα Δικά μας Αποτελέσματα	Παλινδρόμηση, Μαχανική Μάθηση	Λογισμικό Ανοικτού Κώδικα	Java	▲	▲	—	▲*	▲	—	▲*

Πίνακας 5.11: Σύγκριση των Αποτελεσμάτων μας με Άλλες Σχετικές Μελέτες

- **CBO**: Αναφέρεται στο βαθμό σύζευξης μιας κλάσης. Στα μοντέλα που προέκυψαν από στατιστική ανάλυση η επιρροή της CBO στην πρόβλεψη σφαλμάτων είναι στατιστικά σημαντική. Συγκεκριμένα, στο μοντέλο της απλής γραμμικής παλινδρόμησης έχει P-Value ίση με 0,000 και στο δυαδικής λογιστικής παλινδρόμησης ίση με 0,004. Όμως, στις μεθόδους μηχανικής μάθησης δεν είχαμε καθόλου καλά αποτελέσματα αφού στο δέντρο απόφασης η αναλογία σωστών θετικών προβλέψεων ήταν μηδενική και στο νευρωνικό δίκτυο απλά λίγο καλύτερη με TP rate 0,078. Επειδή τα αποτελέσματα στη στατιστική ανάλυση είναι πολύ καλά για να τα αγνοήσουμε, έχουμε σημειώσει στον πίνακα 5.1 ότι η μετρική CBO μπορεί να μας βοηθήσει στην πρόβλεψη σφαλμάτων αλλά έχουμε βάλει ένα αστερίσκο με την ένδειξη ότι απαιτείται να εξεταστεί η χρησιμότητα της και με άλλες μεθόδους μηχανικής μάθησης προκειμένου να έχουμε μια πιο ολοκληρωμένη εικόνα. Οι Aggararwal et al. [014] εξέτασαν έργα λογισμικού που είχαν δημιουργηθεί από φοιτητές και βρήκαν ότι υπάρχει στατιστικά σημαντική σχέση μεταξύ της μετρικής CBO και της πιθανότητας να έχει σφάλμα μια κλάση. Παρόμοια αποτελέσματα με τα παραπάνω είχαν οι μελέτες των Basili et al. [015] και Briand et al. [016] που βασίζονται επίσης στην εξέταση λογισμικού κατασκευασμένου από φοιτητές. Τέλος, οι Emam et al. [017] και οι English et al. [018] βρήκαν όχι απλά στατιστικά σημαντική την CBO στην πρόβλεψη σφαλμάτων αλλά ήταν και η μετρική με τα καλύτερα αποτελέσματα στην πρόβλεψη σφαλμάτων. Από όλες τις μελέτες που εντοπίσαμε η μόνη που δεν κατέληξε θετικά σχετικά με την σημαντικότητα της CBO είναι η μελέτη των Tang et al. [005].

- **LCOM**: Αναφέρεται στη μέτρηση της έλλειψης συνεκτικότητας μίας κλάσης, δηλαδή όσο πιο υψηλή είναι η μετρική LCOM τόσο λιγότερη συνεκτική είναι μια κλάση. Στη γραμμική παλινδρόμηση έδωσε καλά αποτελέσματα με P-Value 0,000 τόσο στο απλό όσο και στο πολλαπλό μοντέλο. Η παραλλαγή της LCOM3 δεν έδωσε ικανοποιητικά αποτελέσματα στη γραμμική παλινδρόμηση αφού δεν βρέθηκε στατιστικά σημαντική (P-Value 0,129). Αντίθετα στη λογιστική παλινδρόμηση η LCOM δεν ικανοποιητικά αποτελέσματα (P-Value 0,1) σε αντίθεση με την παραλλαγή της LCOM3 που βρέθηκε στατιστικά σημαντική ακόμη σε επίπεδο σημαντικότητας 1% (P-Value 0,005). Στο δέντρο απόφασης η LCOM έδωσε μέτρια αποτελέσματα με AUC 0,657 αλλά και η παραλλαγή της LCOM3 δεν τα πήγε πολύ καλύτερα με AUC 0,675. Στο νευρωνικό δίκτυο η LCOM δεν έδωσε ικανοποιητικά αποτελέσματα αφού η αναλογία σωστών θετικών προβλέψεων ήταν μόλις 0,011 σε αντίθεση με την LCOM3 που έφτασε το 0,544 και είχε AUC 0,736 (δεύτερη καλύτερη επίδοση μετά την RFC που είχε 0,812). Με βάση τα παραπάνω καταλήγουμε

στο συμπέρασμα ότι ανάλογα με το μοντέλο που κατασκευάζουμε θα πρέπει να εξετάσουμε την περίπτωση της αντικατάστασης της LCOM με την παραλλαγή της LCOM3. Οι Briand et al. [016] ανέφεραν ότι η LCOM σε σχέση με παραλλαγές της που εξέτασαν, έδωσε τα λιγότερο καλύτερα αποτελέσματα σχετικά με την εύρεση σφαλμάτων σε μια κλάση. Παρόμοια, οι Basili et al. [015] που εξέτασαν λογισμικό που έγραψαν φοιτητές κατέληξαν στο ότι η LCOM δεν βοηθά την πρόβλεψη σφαλμάτων. Από την άλλη πλευρά, οι Olague et al. [020] που εξέτασαν τέσσερις εκδόσεις του προγράμματος Rhino κατέληξαν ότι η μετρική LCOM είναι καλός παράγοντας για την πρόβλεψη σφαλμάτων. Προς την ίδια κατεύθυνση κινήθηκαν οι Gyimothy et al. [019] οι οποίοι εξέτασαν την έκδοση 1.6 του προγράμματος ανοικτού κώδικα Mozilla και βρήκαν ότι υπάρχει σχέση της LCOM με την πρόβλεψη σφαλμάτων.

- **LOC**: Η εξέταση του αριθμού γραμμών πηγαίου κώδικα στην απλή γραμμική παλινδρόμηση έδωσε στατιστικά σημαντικά αποτελέσματα ακόμη σε επίπεδο σημαντικότητας 1% (P-Value 0,000). Στην απλή δυαδική λογιστική παλινδρόμηση δεν έδωσε ικανοποιητικά αποτελέσματα αφού η σχέση της με την ύπαρξη σφάλματος δεν ήταν στατιστικά σημαντική (P-Value 0,298). Όμως στην πολλαπλή λογιστική παλινδρόμηση με βηματική επιλογή προς τα εμπρός επιλέγεται και μάλιστα έχει χαμηλή P-Value 0,001. Στη μηχανική μάθηση δεν δίνει ικανοποιητικά αποτελέσματα, αφού στο δέντρο απόφασης δίνει TP rate 0,4 (δηλαδή ανακαλύπτει το 40% των σφαλμάτων που υπάρχουν στον πηγαίο κώδικα) ενώ στο νευρωνικό δίκτυο το TP rate πέφτει στο 0,044. Οπότε καταλήγουμε στο συμπέρασμα ότι ο αριθμός γραμμών πηγαίου κώδικα δεν μπορεί από μόνος του να μας δώσει καλές προβλέψεις για την ύπαρξη σφάλματος αλλά σε συνδυασμό με άλλες μετρικές μπορεί να βοηθήσει σημαντικά στην πρόβλεψη σφαλμάτων σε μία κλάση. Όλες οι μελέτες που αναφέρουμε στο πίνακα 5.11 και εξέτασαν την μετρική του αριθμού των γραμμών του πηγαίου κώδικα [014, 016, 017, 018, 019, 022] κατέληξαν στο συμπέρασμα ότι υπάρχει στατιστικά σημαντική θετική συσχέτιση ανάμεσα στη LOC και στην ύπαρξη σφάλματος σε μια κλάση.

Κεφάλαιο 6
Επίλογος

Η παρούσα μεταπτυχιακή διατριβή εντάσσεται στην ευρύτερη περιοχή της Τεχνολογίας Λογισμικού (Software Engineering) και πιο συγκεκριμένα ασχολείται με το περισπούδαστο θέμα της Ποιότητας Λογισμικού (Software Quality). Τα τελευταία χρόνια όλο και περισσότερο χρησιμοποιούμε έργα ανοικτού κώδικα και η σημασία τους στην καθημερινότητα μας γίνεται όλο και μεγαλύτερη. Όμως σχετικά μικρός αριθμός μελετών έχει γίνει για την αποτίμηση της ποιότητας τους αλλά και γενικότερα για την εύρεση τρόπων που θα βελτιώσουν την ποιότητα τους. Ένας παράγοντας που μπορεί να επηρεάσει αρνητικά την ποιότητα ενός λογισμικού είναι η ύπαρξη σφαλμάτων στο πρόγραμμα. Το κύριο ερώτημα που προσπαθήσαμε να απαντήσουμε στα πλαίσια της παρούσας έρευνας είναι αν οι αντιμενοστρεφείς μετρικές μπορούν να μας βοηθήσουν πραγματικά στην πρόβλεψη αυτών των σφαλμάτων. Για να το επιτύχουμε αυτό κάναμε μια βιβλιογραφική επισκόπηση και συγκεντρώσαμε μία μεγάλη συλλογή από εσωτερικές μετρικές. Κατόπιν θέσαμε συγκεκριμένα κριτήρια που πρέπει να έχει μία μετρική για να επιλεγεί ως ανεξάρτητη μεταβλητή στα μοντέλα μας και κάναμε μία σημαντική έρευνα για να βρούμε τα εργαλεία που να υποστηρίζουν τις συγκεκριμένες μετρικές. Οι μόνες προϋποθέσεις που θέσαμε για τα εργαλεία αυτά ήταν: α) να είναι δωρεάν διαθέσιμα, β) να μπορούν να υπολογίσουν τις μετρικές στην γλώσσα προγραμματισμού Java και γ) να αναφέρουν τα αποτελέσματα σε επίπεδο κλάσης. Επιλέξαμε το πρόγραμμα ανοικτού κώδικα ckjm στην

επεκταμένη του έκδοση που υποστηρίζει 19 διαφορετικές μετρικές. Μία αναλυτική περιγραφή όλων των εργαλείων που αξιολογήθηκαν δίνεται στο παράρτημα Ε. Στη συνέχεια αναζητήσαμε ένα πρόγραμμα ανοικτού κώδικα που να είναι γραμμένο στην Java, να είναι μεσαίου μεγέθους και να υπάρχουν διαθέσιμα τα σφάλματα που διορθώθηκαν σε κάθε έκδοση του. Τελικά επιλέξαμε την έκδοση 3.2 του προγράμματος jEdit που είναι ένα δημοφιλές πρόγραμμα διόρθωσης κειμένου (text editor) εξειδικευμένο για προγραμματιστές. Κατεβάσαμε τον πηγαίο κώδικα της έκδοσης 3.2 του jEdit και με την χρήση του ckjm extented πήραμε τα αποτελέσματα των μετρικών. Επειδή η έξοδος του ckjm δίνει την κυκλωματική πολυπλοκότητα σε επίπεδο συνάρτησης κάθε κλάσης, υπολογίσαμε τη μέγιστη και τη μέση κυκλωματική πολυπλοκότητα για κάθε κλάση. Εν συνέχεια χρησιμοποιήσαμε το πρόγραμμα ανοικτού κώδικα BugInfo για να συλλέξουμε όλα τα σφάλματα της έκδοσης 3.2 του jEdit από τις καταχωρήσεις του ιστορικού (log files) του συστήματος διαχείρισης κώδικα SVN. Επιπλέον, επειδή η αυτοματοποιημένη διαδικασία του BugInfo δεν εγγυάται την πληρότητα των συλλεγμένων σφαλμάτων, έγιναν και διορθώσεις που προήλθαν από την χειροκίνητη αντιστοίχηση της διόρθωσης λαθών με τα συγκεκριμένα αρχεία κλάσεων του πηγαίου κώδικα που διορθώθηκαν. Στο τέλος αυτής της διαδικασίας είχαμε για κάθε κλάση τα επεξεργασμένα αποτελέσματα των μετρικών από το ckjm extented και τον αριθμό λαθών που βρέθηκαν στην έκδοση 3.2 του προγράμματος jEdit από τη δημόσια διάθεση του, μέχρι να βγει η νέα έκδοση 4.0 του προγράμματος. Τα δεδομένα αυτά αποτέλεσαν την είσοδο στα μοντέλα μας και έχουν ενσωματωθεί στο παράρτημα Δ.

Το επόμενο μεγάλο βήμα ήταν η εύρεση των μοντέλων που θα εξετάζονταν στα πλαίσια της διπλωματικής διατριβής. Ένα μοντέλο ουσιαστικά είναι μία συνάρτηση της μορφής `Y = f(X)`, όπου `X` είναι το διάνυσμα των ανεξάρτητων μεταβλητών και `Y` το διάνυσμα των εξαρτημένων μεταβλητών. Στην περίπτωση μας η εξαρτημένη μεταβλητή `Y` ήταν ο αριθμός των σφαλμάτων ή η ύπαρξη σφάλματος σε μία κλάση και οι ανεξάρτητες μεταβλητές `X` οι μετρικές που συλλέξαμε για τη συγκεκριμένη κλάση. Έγινε εκτεταμένη αναζήτηση σε άρθρα, έρευνες και γενικότερα σχετική βιβλιογραφία, όπου καταλήξαμε να χρησιμοποιήσουμε δύο μεθόδους στατιστικής ανάλυσης και δύο μεθόδους μηχανικής μάθησης. Αναζητήθηκαν εργαλεία που να υποστηρίζουν τις συγκεκριμένες μεθόδους, τα οποία να είναι ανοικτού κώδικα και να υπάρχει μεγάλη βάση χρηστών ώστε να εξασφαλίζεται η υποστήριξη. Καταλήξαμε στο πρόγραμμα R για τις στατιστικές μεθόδους και στο WEKA για τη μηχανική μάθηση. Πιο συγκεκριμένα:

- **Στατιστική Ανάλυση**: Για την εύρεση του αριθμού των σφαλμάτων σε μία κλάση χρησιμοποιήσαμε απλή παλινδρόμηση προκειμένου να εξετάσουμε τη σχέση κάθε μετρικής ξεχωριστά με τον αριθμό των σφαλμάτων. Μετά εφαρμόσαμε πολλαπλή

βηματική παλινδρόμηση με επιλογή μεταβλητών προς τα εμπρός (forward stepwise selection) για να διερευνήσουμε τη συνδυασμένη επίδραση τους στην πρόβλεψη του αριθμού των σφαλμάτων. Η δεύτερη μέθοδος που εφαρμόσαμε είναι η δυαδική λογιστική παλινδρόμηση για την πρόβλεψη της ύπαρξης σφάλματος σε μία κλάση. Αρχικά με απλή λογιστική προσπαθήσαμε να απομονώσουμε την επίδραση κάθε μετρικής ξεχωριστά και μετά στην πολλαπλή βηματική λογιστική παλινδρόμηση με επιλογή μεταβλητών προς τα εμπρός εξακριβώσαμε τη συνδυαστική τους επίδραση για την πρόβλεψη σφαλμάτων. Για την εφαρμογή των στατιστικών μεθόδων χρησιμοποιήσαμε το πρόγραμμα ανοικτού κώδικα R και για τις γραφικές παραστάσεις το SPSS. Τα αναλυτικά script που δημιουργήσαμε δίνονται στο παράρτημα Α και τα αναλυτικά αποτελέσματα στο παράρτημα Β.

- **Μηχανική Μάθηση**: Τα δέντρα απόφασης (decision trees) είναι πολύ χρήσιμα γιατί το αποτέλεσμα τους μπορεί να αναπαρασταθεί γραφικά και να εξαχθούν κανόνες της μορφής "**AN** X **TOTE** Y". Τα τεχνητά νευρωνικά δίκτυα χρησιμοποιούνται πλέον σε ένα ευρύτατο φάσμα επιστημονικών περιοχών για πρόβλεψη. Μειονεκτήματά τους είναι ότι δε μας δίνουν κάποια ερμηνεία για το φαινόμενο που μελετάται και δεν υπάρχει κάποιος κανόνας για τη βέλτιστη αρχιτεκτονική τους. Χρειάζονται αρκετές δοκιμές και ποτέ δεν ξέρουμε αν κάποια άλλη αρχιτεκτονική θα έδινε καλύτερα αποτελέσματα. Για τα δέντρα απόφασης χρησιμοποιήσαμε στο WEKA τον αλγόριθμο J48 που είναι μια μικρή παραλλαγή του C4.5 και για τα νευρωνικά δίκτυα τον αλγόριθμο MultilayerPerceptron που είναι ένα πολυεπίπεδο τεχνητό νευρωνικό δίκτυο αισθητήρα (perceptron) που εκπαιδεύεται με την μέθοδο της οπίσθιας διάδοσης του λάθους (error back propagation). Όλα τα αναλυτικά αποτελέσματα από το WEKA συγκεντρώνονται στο παράρτημα Γ.

Όλα τα μοντέλα που προέκυψαν, η απόδοση και τα συγκεντρωτικά αποτελέσματα τους παρουσιάζονται αναλυτικά για κάθε μετρική και μέθοδο. Για την αποτίμηση κάθε μοντέλου κατηγοριοποίησης που καταλήξαμε, επιλέξαμε να χρησιμοποιηθεί η διασταυρωμένη επικύρωση σε 10 μέρη (10-fold cross validation). Σχολιάσαμε τα αποτελέσματα του καλύτερου μοντέλου για κάθε διαφορετική μέθοδο μοντελοποίησης που εφαρμόσαμε και έγινε σύγκριση της απόδοσης τους με τις πιο σημαντικές παρόμοιες μελέτες. Ιδιαίτερη ανάλυση έγινε για την συλλογή μετρικών CK που είναι το σημείο αναφοράς για αντικειμενοστρεφείς μετρικές. Τέλος, στη συνέχεια του παρόντος κεφαλαίου συγκεντρώσαμε τα κυριότερα συμπεράσματα που προέκυψαν από την αξιολόγηση των μοντέλων μας και προτείνουμε τρόπους με τους οποίους μπορεί η παρούσα έρευνα να επεκταθεί και να βελτιωθεί σε μελλοντικές εργασίες.

6.1 Συμπεράσματα

Η παρούσα μεταπτυχιακή διατριβή διερεύνησε εμπειρικά το βαθμό της καταλληλότητας χρήσης των αντικειμενοστρεφών μετρικών για τη βελτίωση της ποιότητας σε λογισμικό ανοικτού κώδικα με τη δημιουργία μοντέλων πρόβλεψης σφαλμάτων. Οι κυριότερες συνεισφορές της έρευνας μας μπορούν να συνοψιστούν στα παρακάτω:

1. Έγινε αντιστοίχιση των διορθώσεων στα σφάλματα ενός προγράμματος ανοικτού κώδικα με τις κλάσεις πηγαίου κώδικα που χρειάστηκαν αλλαγές. Λόγω των περιορισμών στα σύγχρονα συστήματα διαχείρισης κώδικα (CVS, SVN κλπ) δεν υπάρχει εύκολος ή αξιόπιστος αυτοματοποιημένος τρόπος για να τα συλλέξει κανείς. Συνεπώς συλλέχθηκαν με χρήση αυτόματων και κυρίως χειροκίνητων τρόπων και αναλύθηκαν σημαντικά δεδομένα για λογισμικό ανοικτού κώδικα. Υπάρχουν λίγες σχετικές εμπειρικές μελέτες και η διάθεση αντίστοιχων δεδομένων είναι αρκετά περιορισμένη.

2. Για την πρόβλεψη των σφαλμάτων στις κλάσεις του προγράμματος, πέρα από τις ευρύτατα χρησιμοποιούμενες στην βιβλιογραφία για τη δημιουργία μοντέλων τεχνικές στατιστικής ανάλυσης, διερευνήθηκε και η χρησιμότητα των μεθόδων μηχανικής μάθησης (δέντρο απόφασης και τεχνητό νευρωνικό δίκτυο). Έγινε αποτίμηση των επιδόσεων τους και σύγκριση των μοντέλων που προέκυψαν από τις δύο τεχνικές.

3. Από τις είκοσι μετρικές που χρησιμοποιήθηκαν ως ανεξάρτητες μεταβλητές στα μοντέλα μας, αναγνωρίστηκε ένα υποσύνολο που μπορεί να χρησιμοποιηθεί για την πρόβλεψη σφαλμάτων. Οι περισσότερες μελέτες είτε ασχολήθηκαν μόνο με μία συλλογή μετρικών είτε αποτίμησαν ξεχωριστά τις μετρικές κάθε συλλογής. Η έρευνά μας συνδύασε όλες τις μετρικές όλων των συλλογών ταυτόχρονα και αναγνώρισε τον ιδανικό συνδυασμό μετρικών χωρίς τον περιορισμό να ανήκουν όλες στην ίδια συλλογή.

4. Διαπιστώθηκε ότι υπάρχει μεγάλη έλλειψη αξιοπιστίας στα αποτελέσματα των εργαλείων για μετρικές, αφού δίνουν διαφορετικά αποτελέσματα για τον ίδιο πηγαίο κώδικα. Διαπίστωση που όλες οι μελέτες που εξετάσαμε φαίνεται να παραβλέπουν και να μη λαμβάνουν υπ' όψιν τους. Οι διαφορές μπορεί να οφείλονται είτε στην ασάφεια του ορισμού μιας μετρικής με αποτέλεσμα το κάθε εργαλείο να κάνει διαφορετική υλοποίηση(π.χ. στο πώς χειριζόμαστε τις εσωτερικές μεθόδους), είτε στο γεγονός ότι αρκετά εργαλεία χρησιμοποιούν δικό τους ορισμό που είναι παραλλαγή του επίσημου ορισμού είτε και σε σφάλματα υλοποίησης στα ίδια τα προγράμματα των μετρικών.

Τα πιο σημαντικά συμπεράσματα της έρευνας μας συγκεφαλαιώνονται στα εξής:

1. Οι αντικειμενοστρεφείς μετρικές δίνουν ικανοποιητικά αποτελέσματα σε μοντέλα για την έγκαιρη πρόγνωση σφαλμάτων σε προγράμματα ανοικτού κώδικα. Η απόδοση των μοντέλων μπορεί να χαρακτηριστεί πολύ καλή αλλά απέχει από την άριστη πρόβλεψη.

2. Οι παραδοσιακές μετρικές και ειδικότερα ο αριθμός των γραμμών του πηγαίου κώδικα LOC και η μέγιστη κυκλωματική πολυπλοκότητα μιας κλάσης CC_MAX συνδέονται θετικά και στατιστικά σημαντικά (σε επίπεδο σημαντικότητας 1%) με την ύπαρξη σφάλματος σε μια κλάση. Αντίθετα, τα μοντέλα μηχανικής μάθησης επέδειξαν πολύ χαμηλή ικανότητα πρόβλεψης ύπαρξης σφάλματος σε μια κλάση.

3. Οι στατιστικές τεχνικές έδωσαν μοντέλα με καλύτερη προσαρμοστικότητα από αυτές της μηχανικής μάθησης. Το συμπέρασμα αυτό μπορεί να αμφισβητηθεί στο μέλλον γιατί αφενός η διαφορά τους δεν ήταν μεγάλη (π.χ. η AUC στην πολλαπλή δυαδική λογιστική παλινδρόμηση ήταν 0,841 και στο τεχνητό νευρωνικό δίκτυο 0,793), αφετέρου δε χρησιμοποιήσαμε μόνο δύο από τις δεκάδες τεχνικές μηχανικής μάθησης που υπάρχουν (π.χ. Bayesian Classifiers, Support Vector Machines, AdaBoost, Bagging Predictors, Nearest Neighbors Classifiers, Random Forests, Genetic Programming κλπ).

4. Δεν υπάρχει μία συλλογή μετρικών που να δίνει από μόνη της ανώτερα αποτελέσματα. Τα καλύτερα μοντέλα είχαν ως είσοδο όλες τις μετρικές ή ένα υποσύνολο αυτών. Π.χ. το μοντέλο με τη μεγαλύτερη προσαρμοστικότητα που δημιουργήθηκε με τη μέθοδο της πολλαπλής δυαδικής λογιστικής παλινδρόμησης με επιλογή προς τα εμπρός, είχε ως είσοδο στο μοντέλο τις μετρικές DIT, RFC, DAM, MOA, LOC και CC_MAX. Δηλαδή είχαμε δύο μετρικές από την συλλογή CK (DIT, RFC), δύο από την συλλογή QMOOD (DAM, MOA) και δύο από τις παραδοσιακές μετρικές (LOC, CC_MAX).

5. Η συλλογή των μετρικών CK επιβεβαίωσε την σημαντικότητά της αφού οι τέσσερις (WMC, DIT, CBO και RFC) από τις έξι μετρικές της συλλογής, βρέθηκαν να επηρεάζουν θετικά και στατιστικά σημαντικά (σε επίπεδο σημαντικότητας 1%) την πιθανότητα ύπαρξης σφάλματος σε μία κλάση. Στις μεθόδους μηχανικής μάθησης δεν έδωσαν βέβαια τόσο καλά αποτελέσματα, αλλά είναι άξιο αναφοράς το γεγονός ότι η μετρική RFC μόνη της σαν είσοδο, τόσο στο δέντρο απόφασης όσο και στο νευρωνικό δίκτυο έδωσε αποτελέσματα λίγο χαμηλότερα από το καλύτερο μοντέλο με είσοδο όλες τις μετρικές.

Ολοκληρώνοντας την ενότητα αυτή, επιγραμματικά παρουσιάζουμε τις εντυπώσεις που αποκομίσαμε από τη χρησιμοποίηση εργαλείων ανοικτού κώδικα που ήταν αναγκαία για την ολοκλήρωση της έρευνας μας:

- Δεν υπάρχει εργαλείο μετρικών ανοικτού κώδικα που να προσφέρει μεγάλο αριθμό μετρικών και που να μπορεί να ανταγωνιστεί το εμπορικό λογισμικό σε επίπεδο προσφερομένων λειτουργιών. Το πιο ολοκληρωμένο είναι το ckjm extended που χρησιμοποιήσαμε το οποίο όμως εκτελείται από τη γραμμή εντολών και προσφέρει έξοδο μόνο σε μορφή κειμένου. Από τα υπόλοιπα εργαλεία που αξιολογήσαμε διαπιστώσαμε ότι είτε υλοποιούν ένα μικρό αριθμό μετρικών είτε δεν περιγράφουν αναλυτικά τον ορισμό που υλοποιούν. Μάλιστα, αρκετά έχουν αναπτυχθεί στα πλαίσια ακαδημαϊκών εργασιών και δεν έχουν εξελιχθεί έκτοτε, είτε βρίσκονται απλά στο στάδιο της διόρθωσης σφαλμάτων από εθελοντές.

- Δεν υπάρχει εργαλείο που να μπορεί να αυτοματοποιήσει την αντιστοίχιση της διόρθωσης των σφαλμάτων με αξιόπιστο τρόπο. Το πρόβλημα είναι ότι δεν υπάρχει αντιστοίχιση στη βάση δεδομένων (π.χ. Bugzilla) των σφαλμάτων που διορθώθηκαν με τις κλάσεις που χρειάστηκαν τροποποίηση. Υπάρχουν κάποιες αξιόλογες προσπάθειες που δουλεύουν με χρήση κανονικών εκφράσεων (regular expressions) στα σχόλια των προγραμματιστών αλλά υπάρχει ακόμη αρκετός δρόμος για να θεωρείται αυτή η εργασία αυτοματοποιημένη και αξιόπιστη. Η χειροκίνητη αντιστοίχιση που εφαρμόσαμε είναι ιδιαίτερα χρονοβόρα και όχι πολύ αποδοτική. Το πρόγραμμα ανοικτού κώδικα BugInfo που χρησιμοποιήθηκε στα πλαίσια της μεταπτυχιακής διατριβής βρίσκεται ακόμη στην έκδοση 0.1 από το 2010, οπότε ίσως και αυτό να έχει ήδη εγκαταλειφθεί.

- Η μοντελοποίηση με χρήση προγραμμάτων ανοικτού κώδικα είναι εφάμιλλη με αυτή των προγραμμάτων εμπορικού λογισμικού. Συγκεκριμένα, το πρόγραμμα στατιστικής ανάλυσης R προσφέρει μία μεγάλη ποικιλία έτοιμων στατιστικών πακέτων στην αρχική του εγκατάσταση και επιπλέον μπορούν εύκολα να προστεθούν πάρα πολλά νέα πακέτα βιβλιοθηκών. Το πρόγραμμα R δεν είναι απλά ένα πρόγραμμα στατιστικής ανάλυσης όπως π.χ. το SPSS, αλλά μια ολοκληρωμένη αντικειμενοστρεφής γλώσσα προγραμματισμού για στατιστική ανάλυση. Αντίστοιχα, το πρόγραμμα ανοικτού κώδικα WEKA προσφέρει μια τεράστια ποικιλότητα αλγορίθμων μηχανικής μάθησης όπου ο χρήστης μπορεί με εύκολο τρόπο τόσο να ελέγξει τις παραμέτρους όσο και να μελετήσει τα αποτελέσματα τους. Μάλιστα προσφέρει αναλυτικό Java API για την χρησιμοποίηση των αλγόριθμων αυτών από οποιοδήποτε εξωτερικό πρόγραμμα.

6.2 Προτάσεις για Περαιτέρω Έρευνα

Υπάρχουν πάρα πολλές κατευθύνσεις που θα μπορούσε κάποιος να επεκτείνει την έρευνα που διεξαγάγαμε στα πλαίσια της παρούσας διπλωματικής εργασίας. Στην ενότητα αυτή αναδεικνύουμε τις σημαντικότερες προτάσεις για περαιτέρω έρευνα:

- **Περισσότερα Προγράμματα**: Τα δεδομένα εισόδου για τη δημιουργία των μοντέλων πρόβλεψης σφαλμάτων προέρχονται από ένα πρόγραμμα ανοικτού κώδικα μεσαίου μεγέθους. Προκειμένου να υπάρχει δυνατότητα γενίκευσης των συμπερασμάτων θα πρέπει να εξεταστούν περισσότερα προγράμματα ανοικτού κώδικα και μάλιστα το ιδανικό θα ήταν να μην επιλεγούν τυχαία αλλά μέσω μιας σχετικής μεθοδολογίας που θα πρέπει να περιλαμβάνει συγκεκριμένα κριτήρια επιλογής και στόχους.

- **Διαφορετικές Γλώσσες Προγραμματισμού**: Ως το πιο αντιπροσωπευτικό αλλά και δημοφιλές παράδειγμα αντικειμενοστρεφούς γλώσσας προγραμματισμού επιλέχθηκε η Java. Οι περισσότερες αντίστοιχες ερευνητικές προσπάθειες έχουν γίνει για τη γλώσσα προγραμματισμού C++. Όμως για τις υπόλοιπες αντικειμενοστρεφείς γλώσσες προγραμματισμού όπως είναι οι C#, Object Pascal, Ruby κλπ δεν υπάρχουν σχεδόν καθόλου στοιχεία. Η διερεύνηση της καταλληλότητας των αντικειμενοστρεφών μετρικών για την πρόβλεψη σφαλμάτων και σε άλλες γλώσσες προγραμματισμού είναι μία σημαντική πρόκληση μιας και θα πρέπει να υλοποιηθούν και τα αντίστοιχα εργαλεία που θα πραγματοποιήσουν τις μετρήσεις.

- **Περισσότερες Μετρικές**: Υπολογίστηκαν δεκαεπτά αντικειμενοστρεφείς μετρικές με το εργαλείο ανοικτού κώδικα ckjm extended που περιλαμβάνουν μετρικές των συλλογών CK, QMOOD, Martin κλπ όπως και προτεινόμενες παραλλαγές τους. Όμως, υπάρχουν εκατοντάδες αντικειμενοστρεφείς μετρικές που έχουν προταθεί στη βιβλιογραφία και που πιθανώς να μπορούσαν να βελτιώσουν σημαντικά την απόδοση των μοντέλων πρόβλεψης σφαλμάτων. Μία ερευνητική κατεύθυνση θα ήταν η προσπάθεια υλοποίησης όλων αυτών των μετρικών και η σύγκριση της χρησιμότητας τους για τη βελτίωση της ποιότητας λογισμικού.

- **Μοντέλα Εκτίμησης Προσπάθειας**: Μια ενδιαφέρουσα επέκταση στην παρούσα έρευνα θα ήταν επιπρόσθετα της εκτίμησης για την ύπαρξη σφάλματος σε μια κλάση, να υπάρχει εκτίμηση και της προσπάθειας που θα απαιτηθεί για τη διόρθωσή του. Για να

πραγματοποιηθεί αυτό βέβαια θα χρειαστούν επιπλέον δεδομένα για κάθε κλάση του πηγαίου κώδικα, όπως είναι οι ικανότητες του αρχικού προγραμματιστή ή προγραμματιστών της κλάσης, η εμπειρία του προγραμματιστή ή προγραμματιστών που θα διενεργήσουν τις διορθώσεις, το ιστορικό με τις ώρες που χρειάστηκε η υλοποίηση της κλάσης, των διορθώσεων που ήδη έχουν γίνει κλπ.

- **Κατηγοριοποίηση Σφαλμάτων**: Ένας σημαντικός περιορισμός της παρούσας έρευνας αποτελεί το γεγονός ότι δίνει την ίδια βαρύτητα σε όλα τα σφάλματα του λογισμικού. Όμως, μεγαλύτερη αξία έχουν τα σφάλματα που κρίνονται σημαντικά π.χ. αυτά που μπορούν να οδηγήσουν στον απότομο τερματισμό της εφαρμογής χωρίς καμία προειδοποίηση. Μια πιο ολοκληρωμένη προσέγγιση οφείλει να λαμβάνει υπ' όψιν της τη σοβαρότητα του κάθε σφάλματος και να δημιουργήσει μοντέλα που να μπορούν να κάνουν διάκριση μεταξύ της σοβαρότητας του κάθε σφάλματος, ώστε οι προσπάθειες εντοπισμού των σφαλμάτων να μπορούν να διεξάγονται έχοντας ιεραρχήσει τα πιθανά σφάλματα στις κλάσεις. Βέβαια, μεγάλο πρόβλημα για την εφαρμογή αυτής της κατεύθυνσης είναι ότι θα πρέπει να υπάρχουν πρωτογενή στοιχεία με κατηγοριοποίηση των σφαλμάτων ανάλογα την σημαντικότητα τους. Κάτι τέτοιο όμως στα περισσότερα έργα ανοικτού κώδικα δεν υπάρχει.

- **Δημόσιο Αποθετήριο Δεδομένων**: Η συλλογή και επεξεργασία των δεδομένων σχετικά με μετρικές αλλά και γενικότερα σημαντικών χαρακτηριστικών του λογισμικού ανοικτού κώδικα είναι μια χρονοβόρα διαδικασία που αποτελεί εμπόδιο για τη διεξαγωγή έρευνας σχετικά με την πρόβλεψη σφαλμάτων. Η δημιουργία και η λειτουργία ενός δημόσιου αποθετηρίου δεδομένων ειδικά για μετρικές προγραμμάτων ανοικτού λογισμικού θα έδινε τεράστια ώθηση στις σχετικές ερευνητικές προσπάθειες. Ιδανικά πέρα από τα δημόσια δεδομένα θα μπορούσαν να είναι διαθέσιμα και τα διάφορα εργαλεία που υλοποιήθηκαν στα πλαίσια διαφόρων ερευνητικών προσπαθειών για να μην ξεκινάει κάθε νέα προσπάθεια από την αρχή.

- **Εργαλείο Συλλογής Στοιχείων Σφαλμάτων**: Όλα τα εργαλεία που αξιολογήθηκαν στο πλαίσιο της μεταπτυχιακής διατριβής σχετικά με την αυτόματη συλλογή και αντιστοίχηση των σφαλμάτων με τις κλάσεις του πηγαίου κώδικα, δεν λειτούργησαν με τρόπο ούτε ικανοποιητικό αλλά ούτε και αποδοτικό. Αυτό βέβαια μπορεί να μην οφείλεται αποκλειστικά στα εργαλεία μόνο αλλά και στην πλημμελή τεκμηρίωση από την πλευρά των προγραμματιστών, σχετικά με τα σφάλματα που διορθώνονται. Όμως,

πρέπει να βρεθούν τρόποι ώστε να αυτοματοποιηθεί η διαδικασία προκειμένου να μην απαιτείται επίπονη χειρωνακτική εργασία. Έτσι, για να διευκολυνθεί η προσπάθεια δημιουργίας μοντέλων για την πρόβλεψη σφαλμάτων είναι επιτακτική ανάγκη τόσο ο ορισμός αξιόπιστων διαδικασιών στην καταχώρηση των σφαλμάτων από τους προγραμματιστές όσο και η υλοποίηση νέων εργαλείων που θα αυτοματοποιήσουν τη συλλογή αυτής της πολύτιμης πληροφορίας από την ερευνητική κοινότητα.

- **Περισσότεροι Αλγόριθμοι Μηχανικής Μάθησης**: Η μηχανική μάθηση είναι ένας ραγδαία εξελισσόμενος τομέας της τεχνητής νοημοσύνης (artificial intelligence) που τα τελευταία χρόνια έχει αρχίσει να χρησιμοποιείται σε πολλούς επιστημονικούς κλάδους για τη δημιουργία μοντέλων πρόβλεψης. Η επιλογή των πιο αποδοτικών αλγορίθμων μηχανικής μάθησης για ένα πρόβλημα δεν είναι κάτι εύκολο και απαιτούνται πολλοί πειραματισμοί για την εξεύρεση της πιο αποδοτικής λύσης. Πέρα από τους δυο αλγόριθμους μηχανικής μάθησης που εφαρμόσαμε στην παρούσα έρευνα, υπάρχουν πολλοί άλλοι (π.χ. Bayesian Classifiers, Support Vector Machines, AdaBoost κλπ) που θα μπορούσε να εκτιμηθεί η χρησιμότητά τους στην πρόβλεψη σφαλμάτων σε λογισμικό ανοικτού κώδικα με χρήση αντικειμενοστρεφών μετρικών.

Βιβλιογραφία

[001] Δικτυακός τόπος σχετικά με μετρικές από έργα λογισμικού στη NASA, Απρίλιος 2012 http://mdp.ivv.nasa.gov/complexity_metrics.html

[002] L. Hatton, 'Software Failures: Follies and Fallacies', IEEE Review, 1997

[003] Y. Malayia, J. Denton, 'Module size distribution and defect density', 11th International Symposium on Software Reliability Engineering, 2000

[004] B. Eckel, 'Thinking in Java', Prentice Hall, 2006

[005] M. Tang, M. Kao, M Chen, 'An Empirical Study on Object - Oriented Metrics', Proceedings of the 6th International Symposium on Software Metrics, pp. 242 - 249, 1999

[006] L. Lapin, 'Quantitative Methods for Business Decisions', Duxbury Press, 1994

[007] R. Pindyck, D. Rubinfeld, 'Econometric Models & Economic Forecasts', McGraw Hill, 1991

[008] J. Quinlan, 'C4.5: Programs for Machine Learning', Morgan Kaufmann, 1993

[009] M. Minsky, S. Pappert, 'Perceptrons', MIT Press, 1969

[010] P. Werbos, 'Backpropagation through time: what it does and how to do it', Proceedings of the IEEE, 1990

[011] R. Hech-Nielsen, 'Theory of Backpropagation Neural Networks', Proceedings of the International Joint Conference on Neural Networks, 1989

[012] T. Mitchell, 'Machine Learning', McGraw-Hill, 1997

[013] R. Reed and R. Marks, 'Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks', MIT Press, 1999

[014] K. Aggrarwal, Y. Singh, A. Kaur, R. Malhotra, 'Investigating Effect of Design Metrics on Fault Proneness in Object-Oriented Systems', Journal of Object Technology, Vol. 6, pp. 127 - 141, 2007

[015] V. Basili, L. Briand, W. Melo, 'A Validation of Object-Oriented Design Metrics as Quality Indicator', IEEE Transactions on Software Engineering, Vol. 22, No. 10, pp. 751 - 761, 1996

[016] L. Briand, J Wuest, H. Lounis, 'Replicated Case Studies for Investigating Quality Factors in Object-Oriented Desings', Empirical Software Engineering International Journal (Toronto, Ont.), Vol. 6, No.1, pp. 11-58, 2001

[017] K. Emam, S. Benlarbi, N. Goel, S. Rai, 'The Confounding Effect of Class Size on the Validity of Object-Oriented Metrics', Empirical Software Engineering International Journal (Toronto, Ont.), Vol. 27, No. 7, pp. 630 - 650, 2001

[018] M. English, C. Exton, I. Rigon, B. Clearyp, 'Fault Detection and Prediction in an Open Source Software Project', Proceeding of the 5th International Conference on Predictor Models in Software Engineering, 2009

[019] T. Gyimothy, R. Ferenc, I. Siket, 'Empirical Validation of Object-Oriented Metrics on Open Source Software for Fault Prediction', IEEE Transactions on Software Engineering, Vol. 31, pp. 482 - 492, 2007

[020] H. Olague, L. Etzkorn, S. Gholston, S. Quattlebaum, 'Empirical Validation of Three Software Metrics Suites to Predict Fault-Proneness of Object-Oriented Classes Developed Using Highly Iterative or Agile Software Development Processes', IEEE Transactions on Software Engineering, Vol. 33, No. 8, pp. 402 - 419, 2007

[021] R. Shatnawi, W. Li, 'The Effectiveness of Software Metrics in Identifying Error-Prone Classes in Post Release Software Evolution Process', The Journal of Systems and Software, Vol. 81, pp. 1868 - 1882, 2008

[022] Y. Singh, A. Kaur, R. Malhotra, 'Empirical Validation of Object-Oriented Metrics for Predicting Fault Proneness', Journal of Software Quality Control, Vol. 18, pp. 3 - 35, 2010

[023] Y. Zou, H. Leung, 'Empirical Analysis of Object - Oriented Design Metrics for Predicting High Severity Faults', IEEE Transactions on Software Engineering, Vol. 32, No. 10, pp. 771 - 784, 2006

[024] G. Pai, 'Empirical Analysis of Software Fault Content and Fault Proneness Using Bayesian Methods', IEEE Transactions on Software Engineering, Vol. 33, No. 10, pp. 675 - 686, 2007

[025] R. Malhotra, A. Jain, 'Fault Prediction Using Statistical and Machine Learning Methods for Improving Software Quality', Journal of Information Processing Systems, Vol. 8, No. 2, pp. 241 - 262, 2012

[026] P. Vixie, 'Open Sources: Voices from the Open Source Revolution', Chapter Software Engineering, O' Reilly and Associates, 1999

[027] I. Stamelos, L. Angelis, A. Oikonomou, G. Bleris, 'Code Quality Analysis in Open Source Software Development', Information Systems Journal, Vol. 12, No. 1, pp. 43 - 60, 2002

[028] E. Raymond, 'The Cathedral and the Bazaar: Musings on Linux and Open Source Software Projects', 1o International Conference on Open Source Systems, 2005

[029] S. Williams, 'Free as in Freedom: Richard Stallman's Crusade for Free Software', O' Reilly and Associates, 2002

[030] A. Mubarak, S. Counsell, R. Hierons, 'Does an 80:20 Rule Apply to Java Coupling?', Proceeding of the International Conference on Evaluation and Assessment in Software Engineering, Keele, UK, 2009

[031] M. Gen, R. Cheng, 'Generic Algorithms and Engineering Optimization', Wiley Series in Engineering Design and Automation, 2000

[032] T. Zimmermann, N. Nagappan, A. Zeller, 'Predicting Bugs from History', In Software Evolution, Spinger, 2008

[033] F. Abreu, R. Carapua, 'Candidate Metric for OSS Within Taxonomy Framework', Journal of System and Software, Vol. 26, No. 1, 1994

[034] F. Abreu, 'The MOOD Metrics Set', Proceedings of ECOOP, Workshop in Metrics, 1995

[035] J. Bansiya, C. Davis, 'A Hierarchical Model for Object-Oriented Design Quality Assessment', IEEE Transactions on Software Engineering, Vol. 28, No. 1, 2002

[036] L. Briand, J. Daly, J. Wust, 'A Unified Framework for Coupling Measurement in Object-Oriented Systems', IEEE Transactions on Software Engineering, Vol. 25, No.1, pp. 91 - 121, 1999

[037] M. Bocco, M. Piattini, C. Calero, 'A Survey of Metrics for UML Class Diagrams', Journal of Object Technology, Vol. 4, pp. 59 - 92, 2005

[038] M. Lorenz, J. Kidd, 'Object-Oriented Software Metrics', Prentice Hall, 1994

[039] S. Chidamber, C. Kemerer, 'A Metrics Suite for Object Oriented Design', IEEE Transactions on Software Engineering, Vol. 20, No. 6, pp. 476 - 493, 1994

[040] W. Li, S. Henry, 'Object-Oriented Metrics that Predict Maintanability', Journal of Systems and Software, Vol. 23, No. 2, pp. 111 - 122, 1993

[041] W. Li, 'Another Metric Suite for Object for Object Oriented Programming', The Journal of Systems and Software, Vol. 44, No. 2, pp. 155 - 162, 1998

[042] F. Lanubile, A. Lonigro, G. Visaggio, 'Comparing Models for Identifying Fault-Prone Software Components', 7th International Conference on Software Engineering and Knowledge Engineering, pp. 312 - 319, 1995

[043] T. Khoshgoftaar, E. Allen, J. Hudepohl, S. Aud, 'Application of Neural Networks to Software Quality Modeling of a Very Large Telecommunications System', IEEE Transactions on Neural Networks, Vol. 8, No. 4, pp. 902 - 909, 1997

[044] M. Evett, T. Khoshgoftar, P. Chen, E. Allen, 'GP-based Software Quality Prediction', 3th Annual Conference on Genetic Programming, pp. 60 - 65, 1998

[045] G. Kaszycki, 'Using Process Metrics to Enhance Software Fault Prediction Models', 10th International Symposium on Software Reliability Engineering, Florida, 1999

[046] G. Denaro, 'Estimating Software Fault-Proneness for Tuning Testing Activities', 22th International Conference on Software Engineering, New York, pp., 704 - 706, 2000

[047] Z. Xu, T. Khoshgoftaar, E. Allen, 'Prediction of Software Faults Using Fuzzy Nonlinear Regression Modeling', 5th IEEE International Symposium on High Assurance System Engineering, New Mexico, pp. 281 - 290, 2000

[048] K. Emam, W. Melo, J. Machado, 'The Prediction of Faulty Classes Using Object-Oriented Design Metrics', Journa of Systems and Software, Vol. 56, No. 1, pp. 63 - 75, 2001

[049] T. Khoshgoftaar, N. Seliya, 'Software Quality Classification Modeling Using the SPRINT Decision Tree Algorithm', 4th IEEE International Conference on Tools with Artificial Intelligence, Washington, pp. 365 - 374, 2002

[050] G. Denaro, L. Lavazza, M. Pezze, 'An Empirical Evaluation of Object Oriented Metrics in Industrial Setting', 5th CaberNet Planary Workshop, Porto Santo, Portugal, 2003

[051] G. Denaro, M. Pezze, S. Morasca, 'Towards Industrially Relevant Fault-Proneness Models', International Journal of Software Engineering and Knowledge Engineering, Vol. 13, No. 3, pp. 395 - 417, 2003

[052] A. Mahaweerawat, P. Sophatsathit, C. Lursinsap, P. Musilek, 'Fault Prediction in Object-Oriented Software Using Neural Network Techiques', Proceedings of the InTech Conference, Houston, pp. 27 - 34, 2004

[053] A. Koru, H. Liu, 'An investigation of the Effect of Module Size on Defect Prefiction Using Static Measures', Workshop on Predictor Models in Software Engineering, Missouri, St. Louis, pp. 1 - 5, 2005

[054] G. Boetticher, 'Improving credibility of Machine Learner Models in Software Engineering', Advanced Machine Learning Applications in Software Engineering, Series on Software Engineering and Knowledge Engineering, Idea Group Publishing, 2006

[055] P. Tomaszewski, J. Hakansson, H. Grahn, L. Lundberg, 'Statistical Models vs Expert Estimation for Fault Prediction in Modified Code - An Industrial Case Study', Journal of Systems and Software, Vol. 80, No. 8, pp. 1227 - 1238, 2007

[056] S. Bibi, G. Tsoumakas, I. Stamelos, I. Vlahvas, 'Regression via Classification Applied on Software Defect Estimation', Expert Systems with Applications, Vol. 34, No. 3, pp. 2091 - 2101, 2008

[057] J. Riquelme, R. Ruiz, D. Rodriguez, J. Moreno, 'Finding Defective Modules from Highly Unbalanced Datasets', 8o Taller Sobre el Apoyo a la Decision en ingenieria del Software, pp. 67 - 74, 2008

[058] C. Chang, C. Chu, Y. Yeh, 'Integrating In-Process Software Defect Prediction with Association Mining to Discover Defect Pattern', Information and Software Technology, Vol. 51, No. 2, pp. 375 - 384, 2009

[059] B. Turhan, G. Kocak, A. Bener, 'Data Mining Source Code for Locating Software Bugs: A Case Study in Telecommunication Industry', Expert Systems and Application, Vol. 36, No. 6, pp. 9986 - 9990, 2009

[060] Δικτυακός Τόπος για το Πρόγραμμα Καταχώρησης Σφαλμάτων Bugzilla, Ιούλιος 2012, http://www.bugzilla.org/

[061] M. Halstead, 'Elements of Software Science', Elsevier, 1977

[062] E. Doren, 'Halstead Complexity Measures', Software Engineering Institute, Carnegie – Mellon University, 1997

[063] T. McCabe, 'A Complexity Measure', IEEE Transactions on Software Engineering, Vol. 2, pp. 308-320, 1976

[064] S. Henry, D. Kafura, 'Software Structure Metrics Based on Information Flow', IEEE Transactions on Software Engineering Journal, Vol. 7, No. 5, pp. 510 - 518, 1981

[065] Δικτυακός Τόπος της Εταιρίας Λογισμικού EMERALD Software, Ιούλιος 2012, http://www.emeraldsoftware.com/

[066] L. Briand, V. Basili, C. Hetmanski, 'Developing Interpretable Models with Optimized Set Reduction for Identifying High Risk Software Components', IEEE Transactions on Software Engineering, Vol. 19, No. 11, pp. 1028 - 1044, 1993

[067] J. Shafer, R. Agrawal, M. Mehta, 'SPRINT: A Scalable Parallel Classifier for Data Mining', 22th International Conference on Very Large Databases, pp. 544 - 555, 1996

[068] L. Breiman, J. Friedman, R. Olshen, C. Stone, 'Classification and Regression Trees', Belmont, CA: Wadsworth, 1984

[069] J. Cleay, L. Trig, 'K*: An Instance-Based Learner Using an Entropic Distance Measure', 12th International Conference on Machine Learning, pp. 108 - 114, 1995

[070] Δικτυακός Τόπος του Λογισμικού Μηχανικής Μάθησης WEKA, Ιούνιος 2012, http://www.cs.waikato.ac.nz/ml/weka/

[071] Δικτυακός Τόπος Ιδρύματος για Προγράμματα Ανοικτού Κώδικα Mozilla, Ιούνιος 2012, http://www.mozilla.org/en-US/

[072] Δικτυακός Τόπος Δημόσιου Αποθετηρίου Δεδομένων Μετρικών PROMISE, Ιούνιος 2012, http://promise.site.uottawa.ca/SERepository/datasets-page.html

[073] Δικτυακός Τόπος του Προγράμματος Ανοικτού Κώδικα Mozilla Rhino, Μάιος 2012, https://developer.mozilla.org/en-US/docs/Rhino

[074] A. Marcus, D. Poshyvanyk, R. Ferenc, 'Using the Conceptual Cohesion of Classes for Fault Prediction in Object-Oriented Systems', IEEE Transactions Software Engineering, Vol. 34, No. 2, pp. 287 - 300, 2008

[075] Δικτυακός Τόπος του Προγράμματος Ανοικτού Κώδικα WinMerge, Μάιος 2012, http://www.winmerge.org

[076] D. Hosmer, S. Lemeshow, 'Applied Logistic Regression', Wiley Interscience, 2000

[077] G. Maddala, 'Introduction to Econometrics', Prentice Hall International, 1992

[078] D. Gujarati, 'Basic Econometrics', McGraw Hill International, 1995

[079] I. Witten, E. Frank, 'Data Mining: Practical Machine Learning Tools and Techniques', The Morgan Kaumann Series in Data Management Systems, 2005

[080] J. Han, M. Kamber, ' Data Mining: Concepts and Techniques', The Morgan Kaumann Series in Data Management Systems, 2006

[081] M. Kantardzic, 'Data Mining: Concepts, Models, Methods, and Algorithms', John Wiley and Sons Publication, 2003

[082] D. Larose, 'Discovering Knowledge Data: An Introduction to Data Mining', John Wiley and Sons Publication, 2005

[083] T. Mitchell, 'Machine Learning', McGraw Hill, 1997

[084] Δικτυακός Τόπος του Προγράμματος Στατιστικής Ανάλυσης Project R, Μάιος 2012, http://www.r-project.org/

[085] M. Crawley, 'The R Book', John Wiley and Sons, 2007

[086] R. Muenchen, 'R for SAS and SPSS Users', Springer, 2009

[087] Δικτυακός Τόπος με όλα τα εκτελέσιμα αρχεία και όλα τα προαιρετικά πακέτα για το πρόγραμμα Project R και ονομάζεται Comprehensive R Archive Network, Μάιος 2012, http://cran.r-project.org/

[088] R. Kabacoff, 'R in Action: Data Analysis and Graphics with R', Manning, 2011

[089] R. Bouckaert, E. Frank, M. Hall, G. Holmes, B. Pfahringer, P. Reutemann, I. Witten, 'WEKA - Experiences with a Java Open Source Project', Journal of Machine Learning Research, Vol. 11, pp. 2533 - 2541, 2010

[090] Δικτυακός Τόπος του Προγράμματος Ανοικτού Κώδικα OpenOffice, Μάιος 2012, http://www.openoffice.org

[091] Δικτυακός Τόπος του Προγράμματος Ανοικτού Κώδικα Eclipse, Μάιος 2012, http://www.eclipse.org/

[092] Δικτυακός Τόπος του Προγράμματος Διαχείρισης Σφαλμάτων Bugzilla, Μάιος 2012, http://www.bugzilla.org/

[093] M. Fischer, M. Pinzger, H. Gall, 'Populating a Release History Database from Version Control and Bug Tracking Systems', Proceedings of the International Conference on Software Maintenance, pp. 23 - 32, 2003

[094] Δικτυακός Τόπος του Προγράμματος Ανοικτού Κώδικα jEdit, Ιούνιος 2012, http://www.jedit.org/

[095] Δικτυακός Τόπος του Προγράμματος Ανοικτού Κώδικα BugInfo, Ιούνιος 2012, http://kenai.com/projects/buginfo

[096] J. Feller, B. Fitzgerald, 'Understanding Open Source Software Development' Addison - Wesley, 2001

[097] P. Kaur, H. Singh, 'Measurement of Process in Open Source Software Development', Trends in Information Management, Vol. 7, No. 2, pp. 198 - 207, 2011

[098] I. Samoladas, I. Stamelos, L. Angelis, 'Survival Analysis on the Duration of Open Source Projects', Information and Software Technology, Vol. 52, No. 9, 2010

[099] G. Krogh, S. Spaeth, 'The Open Source Software Phenomenon: Characteristics that Promote Research', Journal of Strategic Information Systems, Vol. 16, pp. 236 - 253, 2007

[0100] J. Lerner, J. Tirole, 'Some Simple Economics of Open Source', Journal of Industrial Economics, Vol. 50, No. 2, pp. 197 - 234, 2002

[0101] G. Krogh, S. Spaeth, K. Kakhani, 'Community, Joining, and Specialization in Open Source Software Innovation: A Case Study', Research Policy, Vol. 32, No. 7, pp. 1217 - 1241, 2003

[0102] M. Mustonen, 'Copyleft - the Economics of Linux and Other Open Source Software', Information Economics and Policy, Vol. 15, No. 1, pp. 99 - 121, 2003

[0103] N. Economides, E. Katsamakas, 'Two-Sided Competition of Proprietaty vs Open Source Technology Platforms and Implications for the Software Industry', Management Science, Vol. 52, No. 7, pp. 1057 - 1071, 2006

[0104] L. Dahlander, M. Magnusson, 'Relationships Between Open Source Companies and Communities: Observations from Nordic Firms', Journal of Research Policy, Vol.34, pp. 481-493, 2005

[0105] J. Satzinger, R. Jackson, S. Burd, 'Systems Analysis and Design in a Changing World', Thomson Course Technology, 2004

[0106] N. Jorgensen, Putting it All in the truck: Incremental Software Development in the FreeBSD Open Source Project', Information Systems Journal, Vol. 11, No. 4, pp. 321 - 336, 2001

[0107] FLOSS Project Report, 'Floss Project Report: Free/Libre and Open Source Software (FLOSS): Survey and Study', Ανακτήθηκε τον Ιούνιο 2012 http://www.flossproject.org/report/index.htm

[0108] J. Wynn, 'Organizational Structure of Open Source Projects: A Life Cycle Approach', Proceeding of the 7th Annual Conference of the Southern Association for Information Systems, pp. 285 - 290, 2003

[0109] R. Roets, M. Minnaar, K. Wright, 'Open Source: Towards Successful Systems Development Projects in Developing Countries', Proceedings of the 9th International Conference on Social Implications of Computers in Developing Countries, 2007

[0110] K. Crowston, J. Howison, 'The Social Structure of Free and Open Source Software Development', First Monday, Vol. 10, No. 2, 2005

[0111] Δικτυακός Τόπος του Ιδρύματος Ελεύθερου Λογισμικού - Ορισμός, Ιούνιος 2012, http://www.gnu.org/philosophy/free-sw.html

[0112] Δικτυακός Τόπος της Πρωτοβουλίας Ανοικτού Κώδικα - Ορισμός, Ιούνιος 2012, http://opensource.org/docs/osd

[0113] D. Garvin, 'What Does Product Quality Mean?', Sloan Management Review, Vol. 26, No. 1, pp. 25 - 43, 1984

[0114] B. Kitchenham, S. Lawrence, 'Software Quality: The Elusive Target', IEEE Software, Vol. 13, No. 1, pp. 12 -21, 1996

[0115] M. Xenos et al., 'The Correlation Between Developer - Oriented and User - Oriented Software Quality Measurements (A Case Study)', 5th European Conference on Software Quality, pp. 267 - 275, 1996

[0116] Δικτυακός Τόπος του Ελληνικού Οργανισμού Τυποποίησης (ΕΛΟΤ), Ιούλιος 2012, http://www.elot.gr/

[0117] American Society for Quality Control, 'Standard A3', 1978

[0118] P. Crosby, 'Quality is Free', McGraw - Hill, 1979

[0119] J. Juran, F. Gryna, 'Quality Planning and Analysis', McGraw - Hill, 1980

[0120] J. McCall et al, 'Factors in Software Quality, Vols I, II, III', US Rome Air Development Canter Reports NTIS AD/A-049 014, NTIS AD/A-049 015, NTIS AD/A-049 016, 1977

[0121] B. Boehm et al., 'Characteristics of Software Quality', North Holland, 1978

[0122] ISO, 'Information technology ≠ Evolution of Software - Quality Characteristics and Guides for Their Use', International Standard, ISO/IEC 9126, 1991

[0123] B. Golden, 'Making Open Source Ready for the Enterprise, The Open Source Maturity Model' in Succeeding With Open Source, Addison-Wesley Professional, 2005

[0124] Δικτυακός Τόπος για το Open Business Readiness Rating Project (OBRRP), Μάιος 2012, http://www.openbrr.org/

[0125] Δικτυακός Τόπος για το Qualification and Selection of Open Source Software, Μάιος 2012, http://www.qsos.org/

[0126] R. Park, 'Software size measurement: a framework for counting source statements', Software Engineering Institute, Carnegie – Mellon University, 1992

[0127] N. Fenton, 'A critique of software defect prediction models', IEEE Transactions on Software Engineering 25(5), pp. 675 – 689, 1999

[0128] L. Laird, M. Brennan, 'Software Measurement and Estimation', Wiley – Interscience, 2006

[0129] Δικτυακός Τόπος για μετρικές από την εταιρεία Aivosto, Μάιος 2012, http://www.aivosto.com/project/help/pm-complexity.html

[0130] Δικτυακός τόπος που περιλαμβάνονται όλα τα πρότυπα του IEEE, Μάιος 2012, http://standards.ieee.org/catalog/olis/arch_se.html

[0131] B. Kitchenham, L. Pickard, S. Linkman, 'An evaluation of some design metrics', Software Engineering Journal, 1990

[0132] P. Oman, J. Hagemeister, D. Ash, 'A definition and Taxonomy for Software Maintainability', Technical Report, Software Engineering Test Laboratory, University of Idaho, 1991

[0133] D. Kurt, 'The Software Maintainability Index Revisited', Journal of Defense Software Engineering, August 2001

[0134] Δικτυακός Τόπος για τον δείκτη συντηρησιμότητας MI, Μάιος 2012, http://www.verifysoft.com/en_maintainability.html

[0135] D. Coleman, D. Asg, B. Lowther, P. Oman, 'Software maintainability metrics models in practice', Journal of Defense Software Engineering, 8(11), pp. 19 – 23, 1995

[0136] K. Welker, P. Oman, 'Software Maintainability Metrics Models in Practice', Journal of Defence Software Engineering, Vol. 8, pp. 19 -23, 1995

[0137] R. Martin, 'Agile Software Development: Principles, Patterns and Practices', Prentice Hall, 2003

[0138] R. Harrison, S. Cousell, R. Nithi, 'An Evaluation of the MOOD set of Object-Oriented Software Metrics', IEEE Transactions on Software Engineering, Vol. 24,pp. 491 - 496, 1998

[0139] M. El-Wakil, A. El-Bastawissi, M. Boshra, A. Fahmy, 'Object-Oriented Design Quality Models: A Survey and Comparison', 2nd International Conference on Informatics and Systems, 2004

[0140] F. Brito,, G. Miguel, R. Esteves, 'Toward the Design Quality Evaluation of Object-Oriented Software Systems', 5th International Conference on Software Quality, 1995

[0141] D. Breuker, J. Brunekreef, J. Derriks, A. Aicha, 'Reliability of Software Metrics Tools', International Conference on Software Process and Product Measurement, Amsterdam, Netherlands, pp. 10 - 22, 2009

[0142] Δικτυακός τόπος για το εργαλείο ανοικτού κώδικα μετρικών ckjm extended, Μάιος 2012, http://gromit.iiar.pwr.wroc.pl/p_inf/ckjm/

[0143] Δικτυακός τόπος για το εργαλείο ανοικτού κώδικα για μετρικές ckjm, Μάιος 2012, http://www.spinellis.gr/sw/ckjm/

[0144] M. Jureczko, 'Significance of Different Software Metrcis in Defect Prediction', Software Engineering: An International Journal, Vol. 1, No. 1, pp. 86 - 96, 2011

[0145] N. Fenton, S. Pfleeger, 'Software Metrics: A Rigorous and Practical Approach', PWS Publishing Company, 1997

[0146] R. Pressman, 'Software Engineering: A Practitioner's Approach', Mc-Graw Hill, 2000

[0147] Δικτυακός τόπος για το εργαλείο ανοικτού κώδικα RKWard, Μάιος 2012, http://rkward.sourceforge.net/

Παράρτημα Α
R Project Scripts

Δίνεται ο κώδικας που χρησιμοποιήσαμε στο πρόγραμμα στατιστικής ανάλυσης R.

Α.1 Φόρτωμα Δεδομένων

```
local({
## Prepare
## Compute
imported <- read.csv2 (file="c:/jEdit3-2.csv",  na.strings = "NA", nrows = -1,
skip = 0, check.names = TRUE, strip.white = FALSE, blank.lines.skip = TRUE)

# copy from the local environment to globalenv()
.GlobalEnv$jEdit3.2 <- imported

rk.edit (.GlobalEnv$jEdit3.2)
## Print result
rk.header("Import text / csv data", parameters=list("File", "c:/jEdit3-2.csv",
        "Import as", "jEdit3.2"))
})

local({
```

```
## Prepare
## Compute
imported <- read.csv2 (file="c:/jEdit4-0.csv",  na.strings = "NA", nrows = -1,
skip = 0, check.names = TRUE, strip.white = FALSE, blank.lines.skip = TRUE)

# copy from the local environment to globalenv()
.GlobalEnv$jEdit4.0 <- imported

rk.edit (.GlobalEnv$jEdit4.0)
## Print result
rk.header("Import text / csv data", parameters=list("File", "c:/jEdit4-0.csv",
        "Import as", "jEdit4.0"))
})

local({
## Prepare
## Compute
imported <- read.csv2 (file="c:/jEdit4-1.csv",  na.strings = "NA", nrows = -1,
skip = 0, check.names = TRUE, strip.white = FALSE, blank.lines.skip = TRUE)

# copy from the local environment to globalenv()
.GlobalEnv$jEdit4.1 <- imported

rk.edit (.GlobalEnv$jEdit4.1)
## Print result
rk.header("Import text / csv data", parameters=list("File", "c:/jEdit4-1.csv",
        "Import as", "jEdit4.1"))
})
```

A.2 Περιγραφική Στατιστική

A.2.1 Δημιουργία Ραβδογράμματος

```
local({
## Prepare
## Compute
## Print result
rk.header ("Histogram", list ("Variable", rk.get.description
(jEdit3.2[["NO_OF_FAULTS"]])  , "Break points", "Equally spaced vector of length
```

```
10", "Right closed", "TRUE", "Include in lowest cell", "TRUE", "Scale",
"Frequency"))

rk.graph.on ()
try ({
        hist (jEdit3.2[["NO_OF_FAULTS"]], breaks=(function(x) {y =
extendrange(x,f=0.1); seq(from=y[1], to=y[2],
length=10)})(jEdit3.2[["NO_OF_FAULTS"]]), lty="solid", density=-1)
})
rk.graph.off ()
})
```

A.2.2 Δημιουργία Πίνακα Περιγραφικών Στατιστικών

```
## Prepare
## Compute
vars <- rk.list (jEdit3.2[["CC_AVG"]], jEdit3.2[["CC_MAX"]], jEdit3.2[["CE"]],
jEdit3.2[["DAM"]], jEdit3.2[["DIT"]], jEdit3.2[["IC"]], jEdit3.2[["LCOM"]],
jEdit3.2[["LCOM3"]], jEdit3.2[["LOC"]], jEdit3.2[["MFA"]], jEdit3.2[["MOA"]],
jEdit3.2[["NOC"]], jEdit3.2[["NPM"]], jEdit3.2[["RFC"]], jEdit3.2[["WMC"]],
jEdit3.2[["CBO"]], jEdit3.2[["CBM"]], jEdit3.2[["CAM"]], jEdit3.2[["CA"]],
jEdit3.2[["AMC"]])
results <- data.frame ('Object'=I(names (vars)))
for (i in 1:length (vars)) {
        var <- vars[[i]]

        # we wrap each single call in a "try" statement to always continue on
errors.
        results[i, 'mean'] <- try (mean (var, trim = 0.00, na.rm=TRUE))
        results[i, 'median'] <- try (median (var, na.rm=TRUE))
        try ({
                range <- try (range (var, na.rm=TRUE))
                results[i, 'min'] <- range[1]
                results[i, 'max'] <- range[2]
        })
        results[i, 'standard deviation'] <- try (sd (var, na.rm=TRUE))
        results[i, 'length of sample'] <- length (var)
        results[i, 'number of NAs'] <- sum (is.na(var))
}
## Print result
rk.header ("Descriptive statistics", parameters=list (
```

```
                "Trim of mean", 0.00))

rk.results (results)
})
```

A.2.3 Δημιουργία Πίνακα Συσχετίσεων

```
local({
## Prepare
## Compute
# cor requires all objects to be inside the same data.frame.
# Here we construct such a temporary frame from the input variables
data    <-    as.data.frame    (rk.list    (jEdit3.2[["AMC"]],    jEdit3.2[["CA"]],
jEdit3.2[["CAM"]],  jEdit3.2[["CBM"]],  jEdit3.2[["CBO"]],  jEdit3.2[["CC_AVG"]],
jEdit3.2[["CC_MAX"]],   jEdit3.2[["CE"]],   jEdit3.2[["DAM"]],   jEdit3.2[["DIT"]],
jEdit3.2[["IC"]],   jEdit3.2[["LCOM"]],   jEdit3.2[["LCOM3"]],   jEdit3.2[["LOC"]],
jEdit3.2[["MFA"]],   jEdit3.2[["MOA"]],   jEdit3.2[["NOC"]],   jEdit3.2[["NPM"]],
jEdit3.2[["RFC"]], jEdit3.2[["WMC"]]), check.names=FALSE)

# calculate correlation matrix
result <- cor (data, use="complete.obs", method="spearman")
# calculate matrix of probabilities
result.p <- matrix (nrow = length (data), ncol = length (data), dimnames=list
(names (data), names (data)))
# as we need to do pairwise comparisons for technical reasons,
#  we  need  to  exclude  incomplete  cases  first  to  match  the  use="complete.obs"
parameter to cor()
data <- data[complete.cases (data),]
for (i in 1:length (data)) {
        for (j in i:length (data)) {
                if (i != j) {
                        t <- cor.test (data[[i]], data[[j]], method="spearman")
                        result.p[i, j] <- t$p.value
                        result.p[j,   i]   <-   sum   (complete.cases   (data[[i]],
data[[j]]))
                }
        }
}
## Print result
rk.header   ("Correlation   Matrix",   parameters=list   ("Method",   "spearman",
"Exclusion", "complete.obs"))
```

```
rk.results (data.frame (result, check.names=FALSE), titles=c ("Coefficient", names
(data)))
rk.results (data.frame (result.p, check.names=FALSE), titles=c ("n \\ p", names
(data)))
})
```

A.3 Γραμμική Παλινδρόμηση

A.3.1 Απλή Γραμμική Παλινδρόμηση

```
local({
## Prepare
## Compute
results <- lm (jEdit3.2[["NO_OF_FAULTS"]] ~ jEdit3.2[["WMC"]])
## Print result
rk.header ("Linear Regression")
rk.print(results)
})
```

A.3.2 Πολλαπλή Γραμμική Παλινδρόμηση

```
local({
## Prepare
## Compute
results <- step(lm (jEdit3.2[["NO_OF_FAULTS"]] ~ jEdit3.2[["AMC"]] +
jEdit3.2[["CA"]] + jEdit3.2[["CAM"]] + jEdit3.2[["CBM"]] + jEdit3.2[["CBO"]] +
jEdit3.2[["CC_AVG"]] + jEdit3.2[["CC_MAX"]] + jEdit3.2[["CE"]] + jEdit3.2[["DAM"]]
+ jEdit3.2[["DIT"]] + jEdit3.2[["IC"]] + jEdit3.2[["LCOM"]] + jEdit3.2[["LCOM3"]]
+ jEdit3.2[["LOC"]] + jEdit3.2[["MFA"]] + jEdit3.2[["MOA"]] + jEdit3.2[["NOC"]] +
jEdit3.2[["NPM"]] + jEdit3.2[["RFC"]] + jEdit3.2[["WMC"]]), direction="forward")
## Print result
rk.header ("Linear Regression")
rk.print(results)
})
```

A.4 Δυαδική Λογιστική Παλινδρόμηση

A.4.1 Απλή Δυαδική Λογιστική Παλινδρόμηση

```
local({
## Prepare
## Compute
results <- glm(jEdit3.2[["FAULTY"]] ~ jEdit3.2[["WMC"]])
## Print result
rk.header ("Logistic Regression")
rk.print(results)
})
```

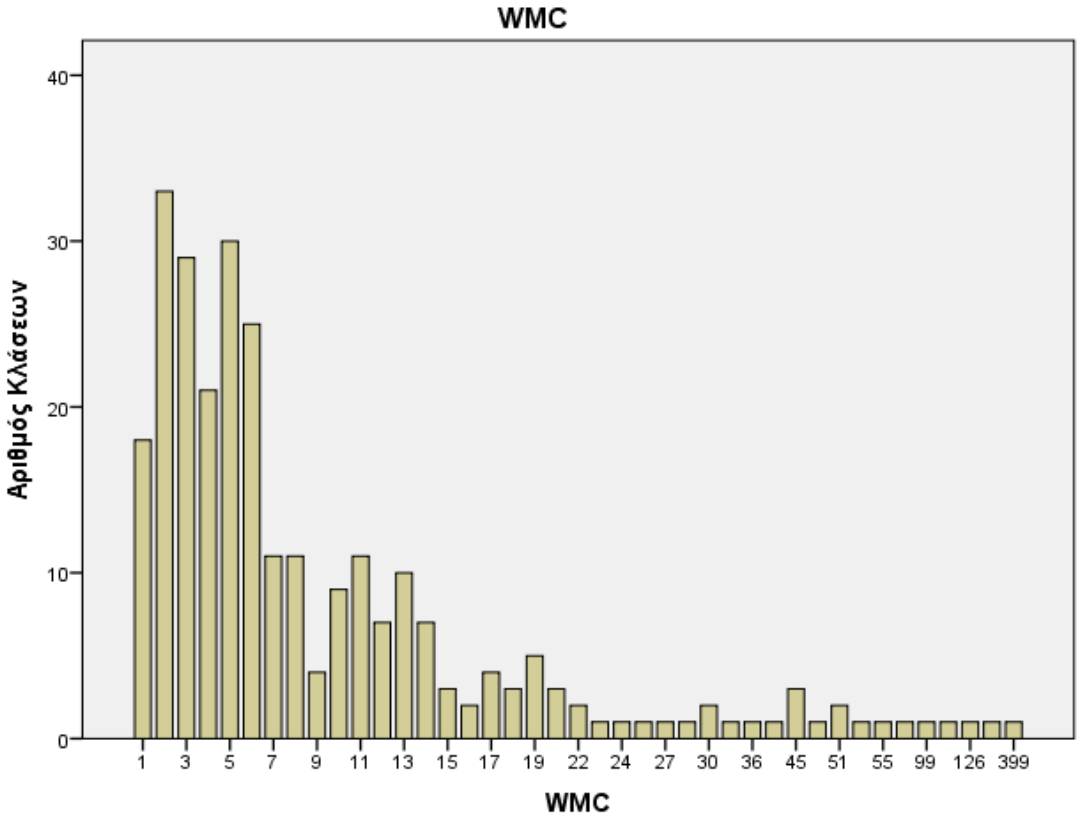
A.4.2 Πολλαπλή Δυαδική Λογιστική Παλινδρόμηση

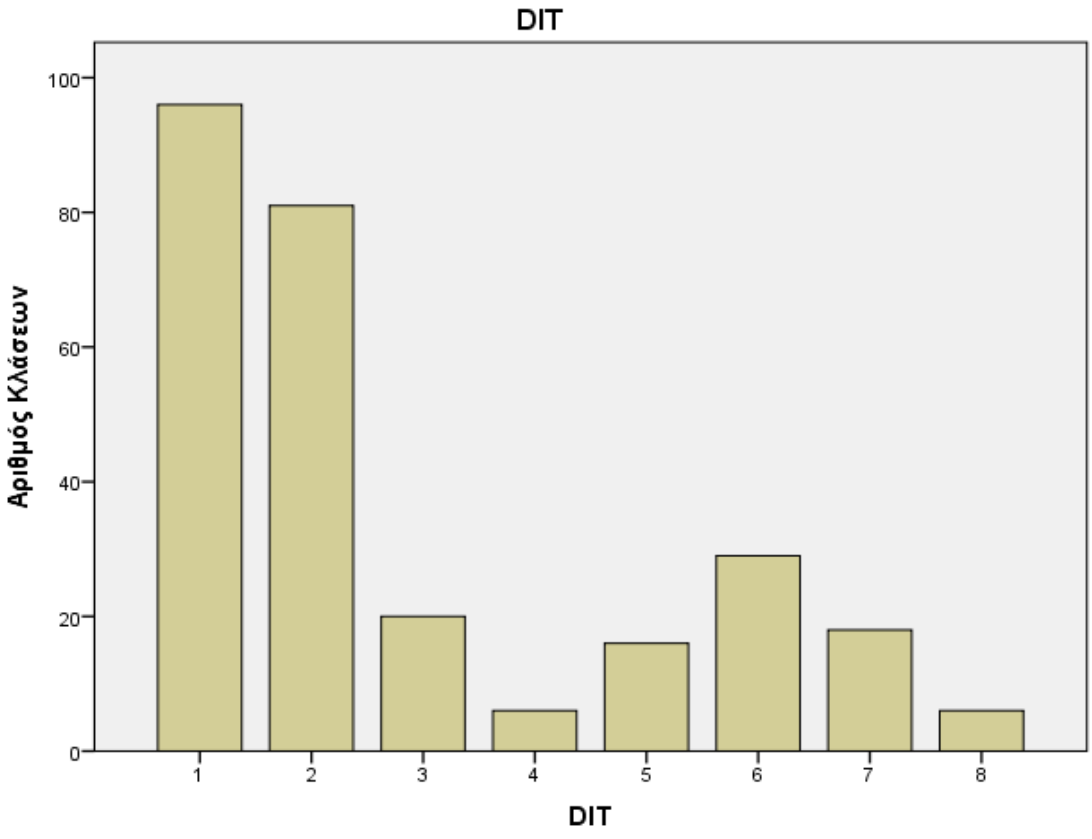
```
local({
## Prepare
## Compute
results <- step(glm(jEdit3.2[["FAULTY"]] ~ jEdit3.2[["AMC"]] + jEdit3.2[["CA"]] +
jEdit3.2[["CAM"]] + jEdit3.2[["CBM"]] + jEdit3.2[["CBO"]] + jEdit3.2[["CC_AVG"]] +
jEdit3.2[["CC_MAX"]] + jEdit3.2[["CE"]] + jEdit3.2[["DAM"]] + jEdit3.2[["DIT"]] +
jEdit3.2[["IC"]] + jEdit3.2[["LCOM"]] + jEdit3.2[["LCOM3"]] + jEdit3.2[["LOC"]] +
jEdit3.2[["MFA"]] + jEdit3.2[["MOA"]] + jEdit3.2[["NOC"]] + jEdit3.2[["NPM"]] +
jEdit3.2[["RFC"]] + jEdit3.2[["WMC"]]), direction="forward")
## Print result
rk.header ("Logistic Regression")
rk.print(results)
})
```

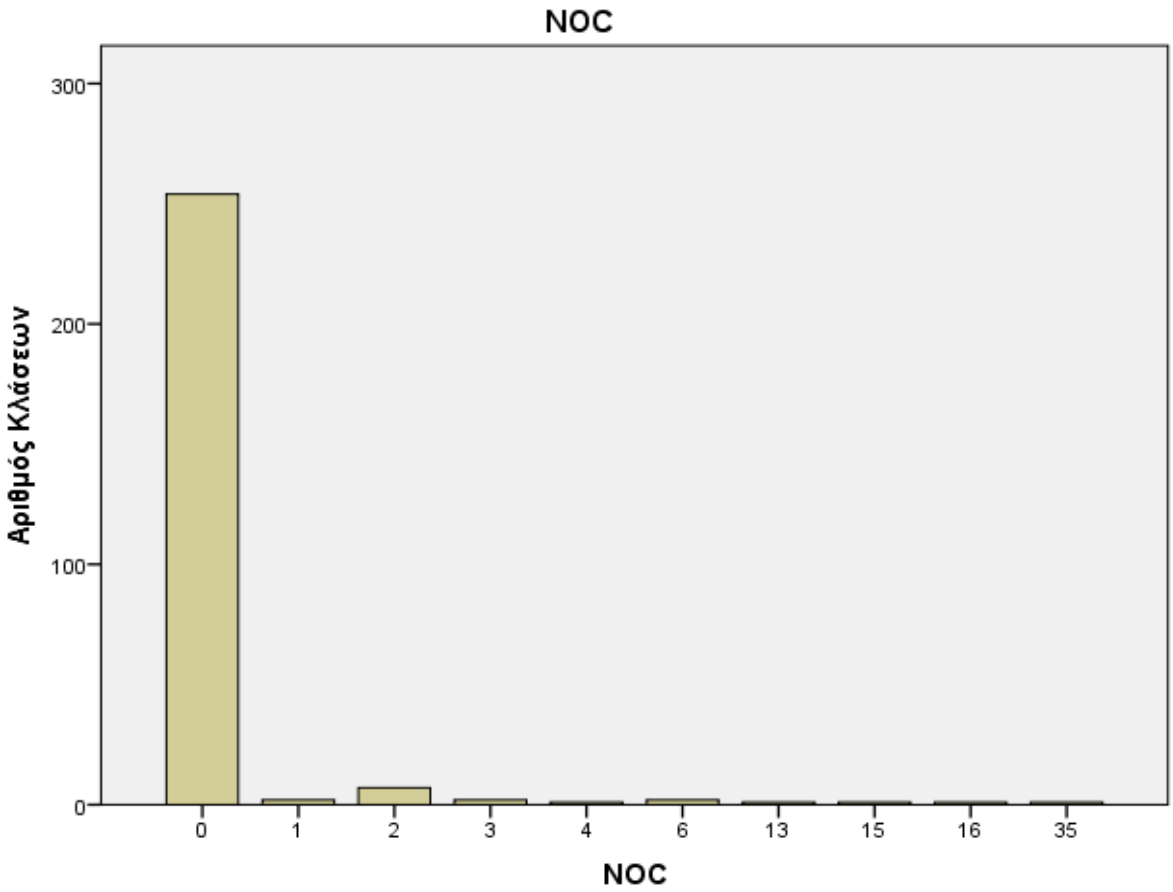
Παράρτημα Β
Λεπτομερή Στατιστικά Αποτελέσματα από R και SPSS

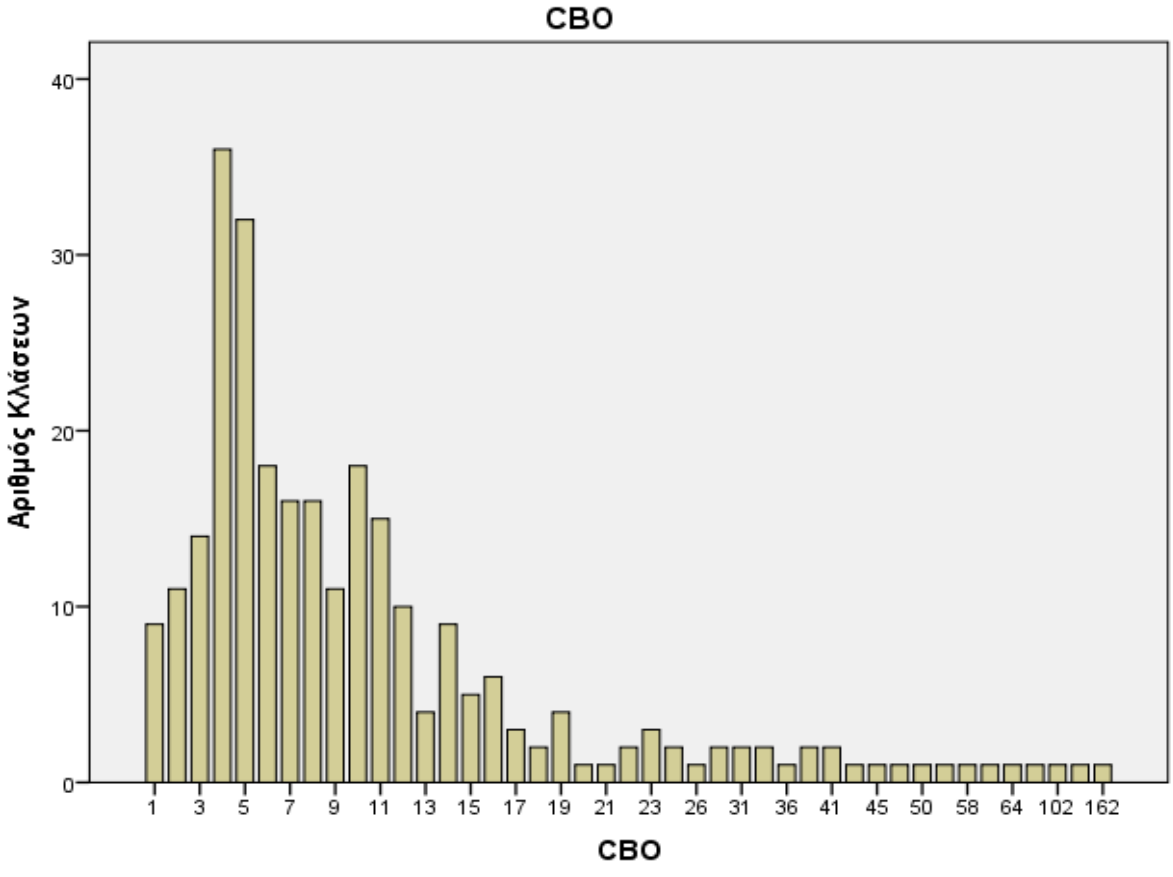
Στο παρόν παράρτημα δίνουμε αναλυτικά όλα τα αποτελέσματα που προέκυψαν από την ανάλυση στο πρόγραμμα στατιστικής ανάλυσης R και όλες τις γραφικές αναπαραστάσεις που δημιουργήθηκαν στο SPPS για την οπτική υποστήριξη των αποτελεσμάτων στα μοντέλα στατιστικής ανάλυσης. Στην πρώτη ενότητα της περιγραφικής στατιστικής παρουσιάζουμε την συχνότητα των τιμών από τις μετρικές που υπολογίσαμε με το πρόγραμμα ckjm extended. Στην απλή γραμμική παλινδρόμηση τα στοιχεία που περιλαμβάνονται σε κάθε μοντέλο είναι ο συντελεστής προσδιορισμού, η ανάλυση ANOVA, το επίπεδο σημαντικότητας των συντελεστών και διάφορες στατιστικές που αφορούν την κατανομή των σφαλμάτων. Στην πολλαπλή γραμμική παλινδρόμηση δείχνουμε με γραφικό τρόπο τα βήματα που ακολουθήθηκαν κατά την εφαρμογή της μεθόδους της επιλογής προς τα εμπρός, την κατανομή των σφαλμάτων, τον συντελεστή προσδιορισμού, τους συντελεστές του μοντέλου μαζί με το επίπεδο σημαντικότητας, τα άνω και κάτω όρια, καθώς και τις γραφικές παραστάσεις με την σημαντικότητα κάθε μετρικής και την συσχέτιση της με τα σφάλματα στις κλάσεις. Ακριβώς την ίδια τακτική έχουμε ακολουθήσει και για την περίπτωση της δυαδικής λογιστικής παλινδρόμησης, τόσο για την περίπτωση της απλής όσο και της πολλαπλής.

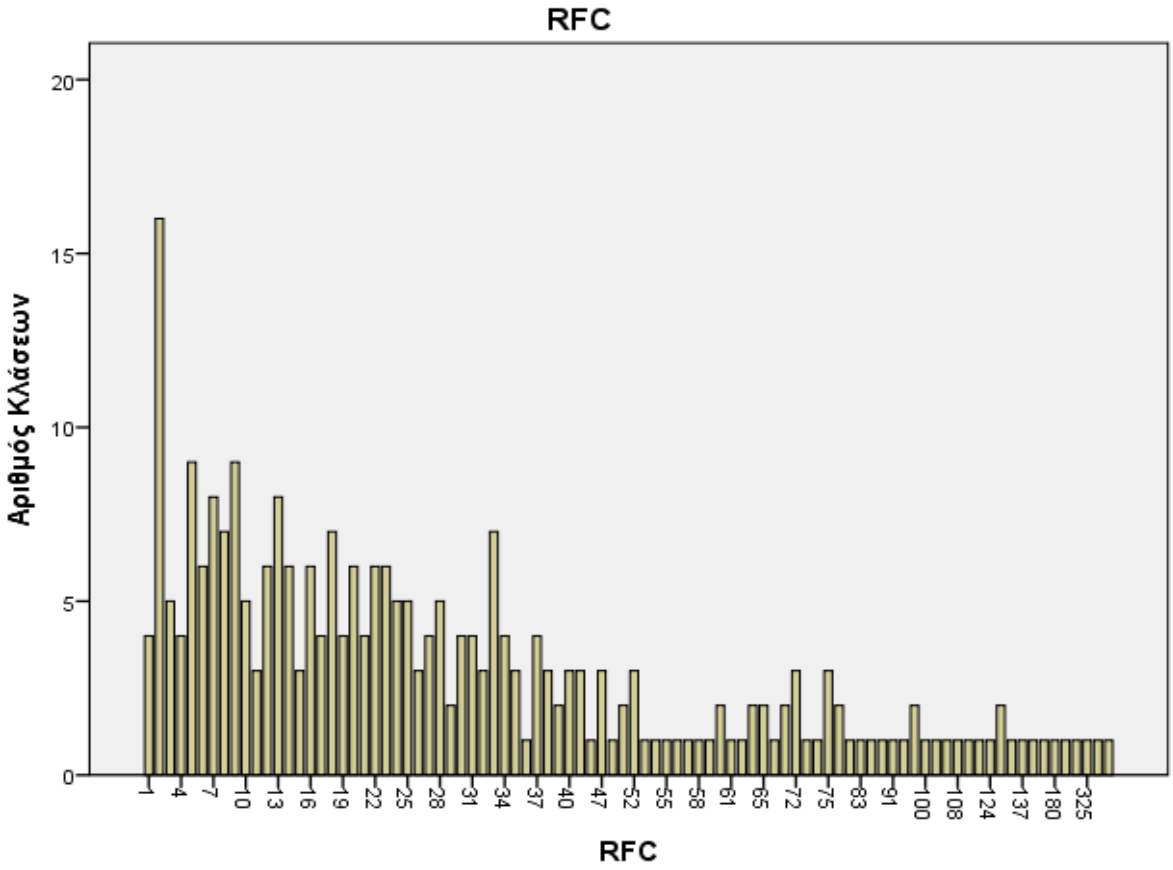
Β.1 Περιγραφική Στατιστική

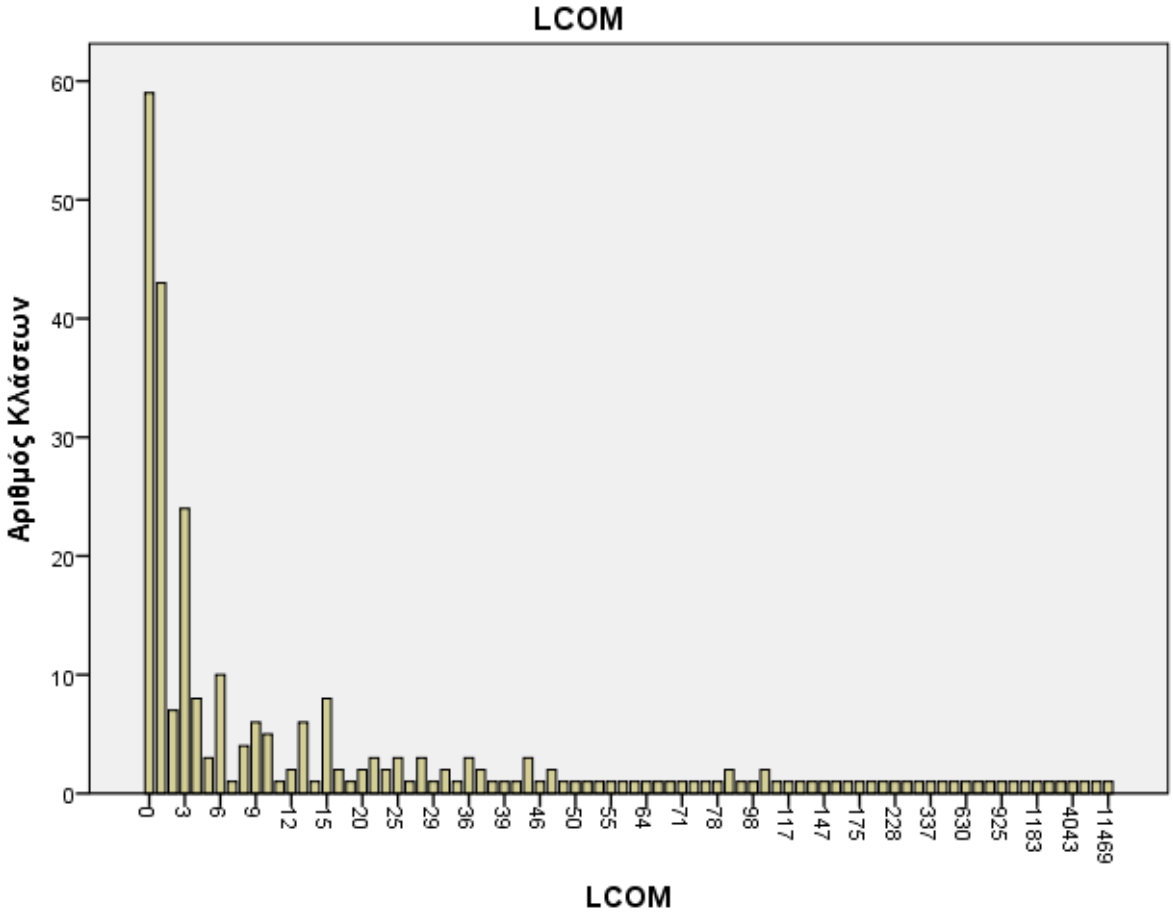


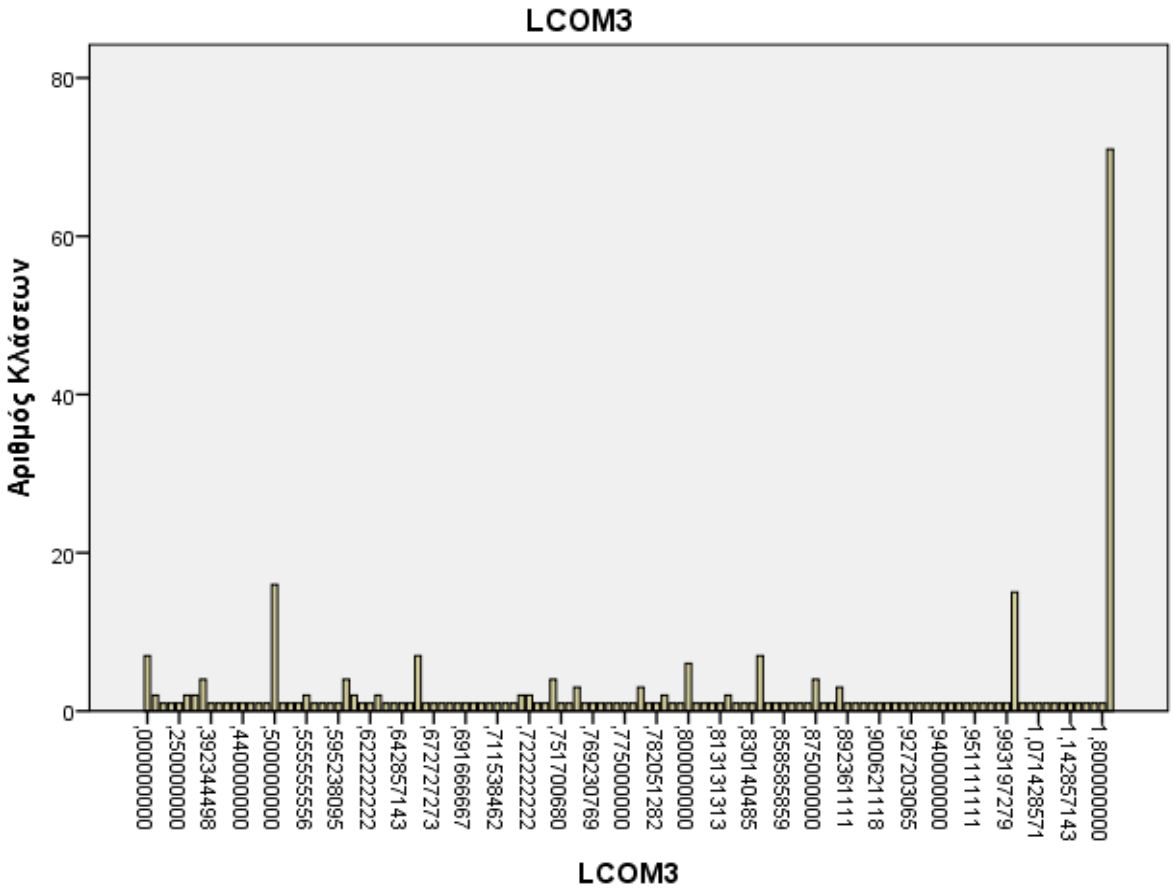


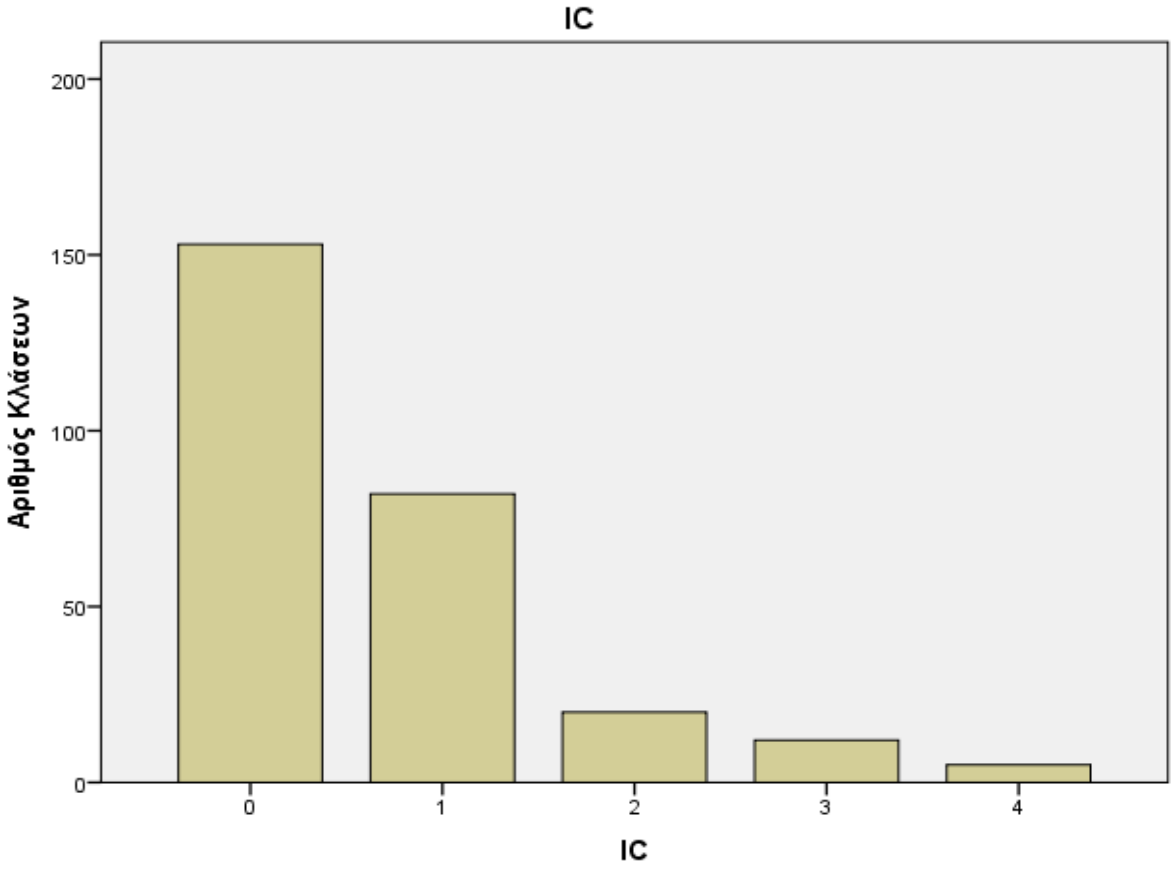

NOC
Αριθμός Κλάσεων
300
200
100
0
0
1
2
3
4
6
13
15
16
35
NOC

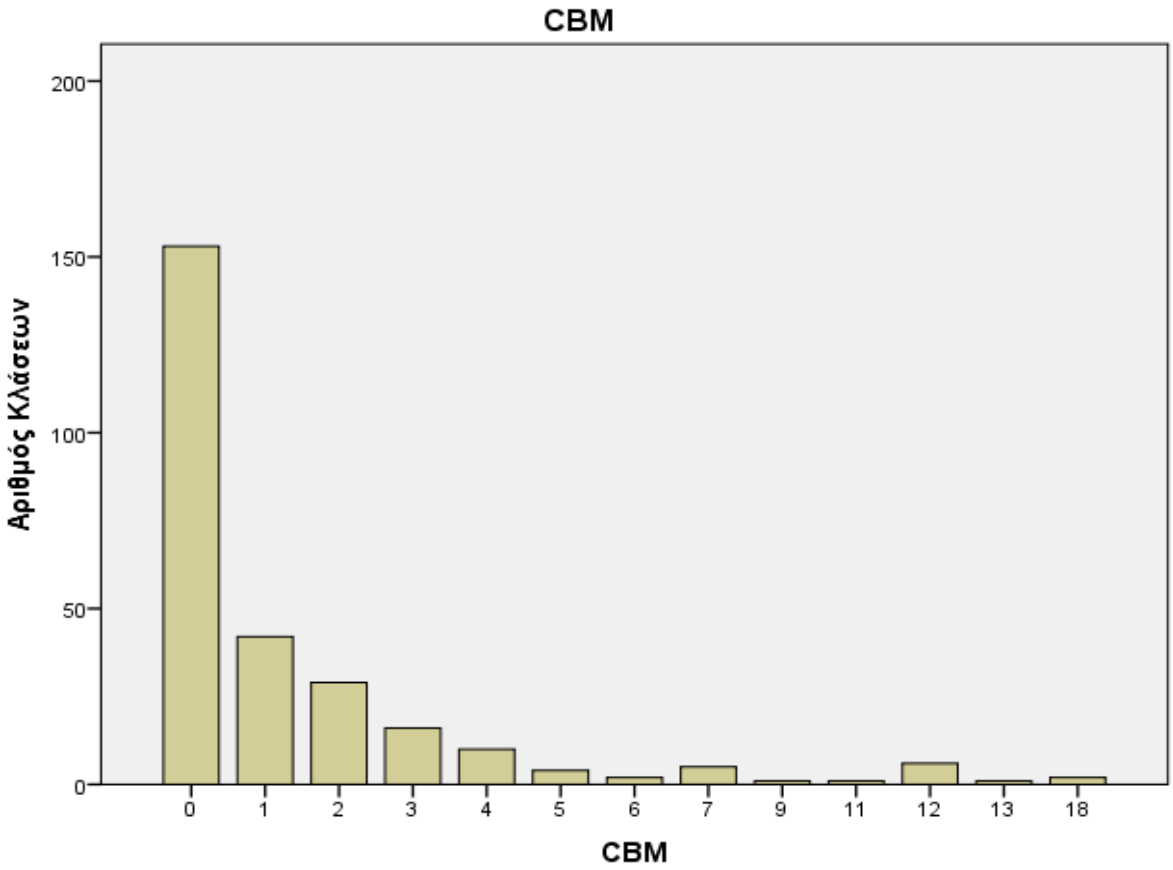

CBO
Αριθμός Κλάσεων
40
30
20
10
0
1
3
5
7
9
11
13
15
17
19
21
23
26
31
36
41
45
50
58
64
102
162
CBO

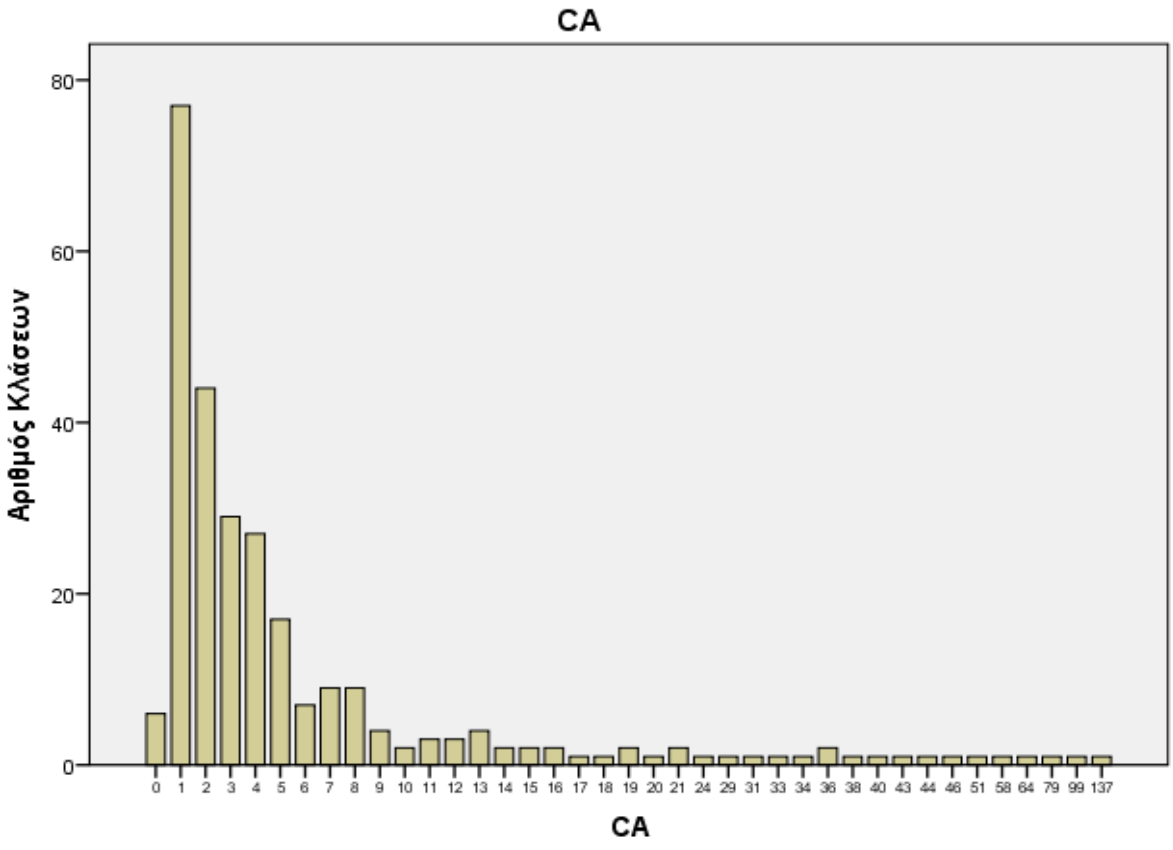

RFC
Αριθμός Κλάσεων
0
5
10
15
20
1
4
7
10
13
16
19
22
25
28
31
34
37
40
47
52
55
58
61
65
72
75
83
91
100
108
124
137
180
325
RFC

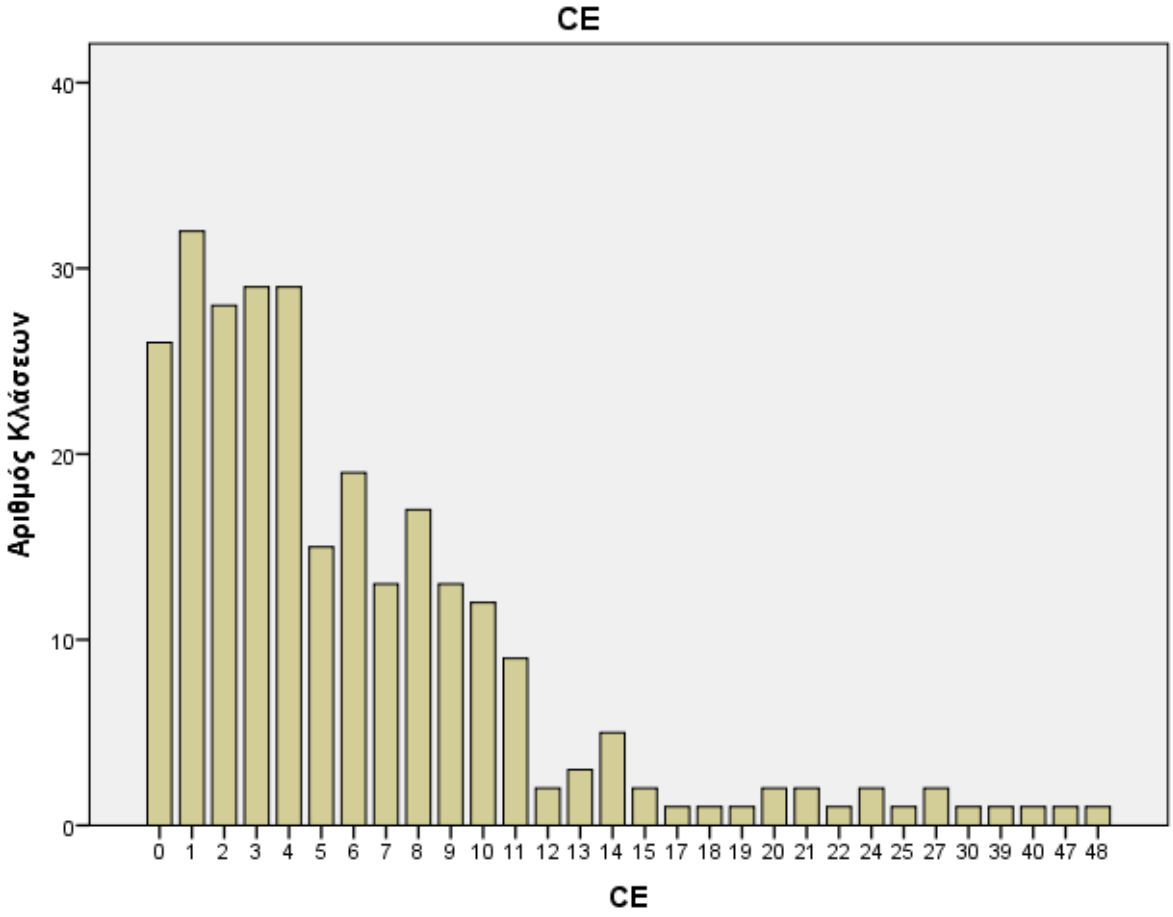

LCOM
Αριθμός Κλάσεων
0
10
20
30
40
50
60
0
3
6
9
12
15
20
25
29
36
39
46
50
55
64
71
78
98
117
147
175
228
337
630
925
1183
4043
11469
LCOM

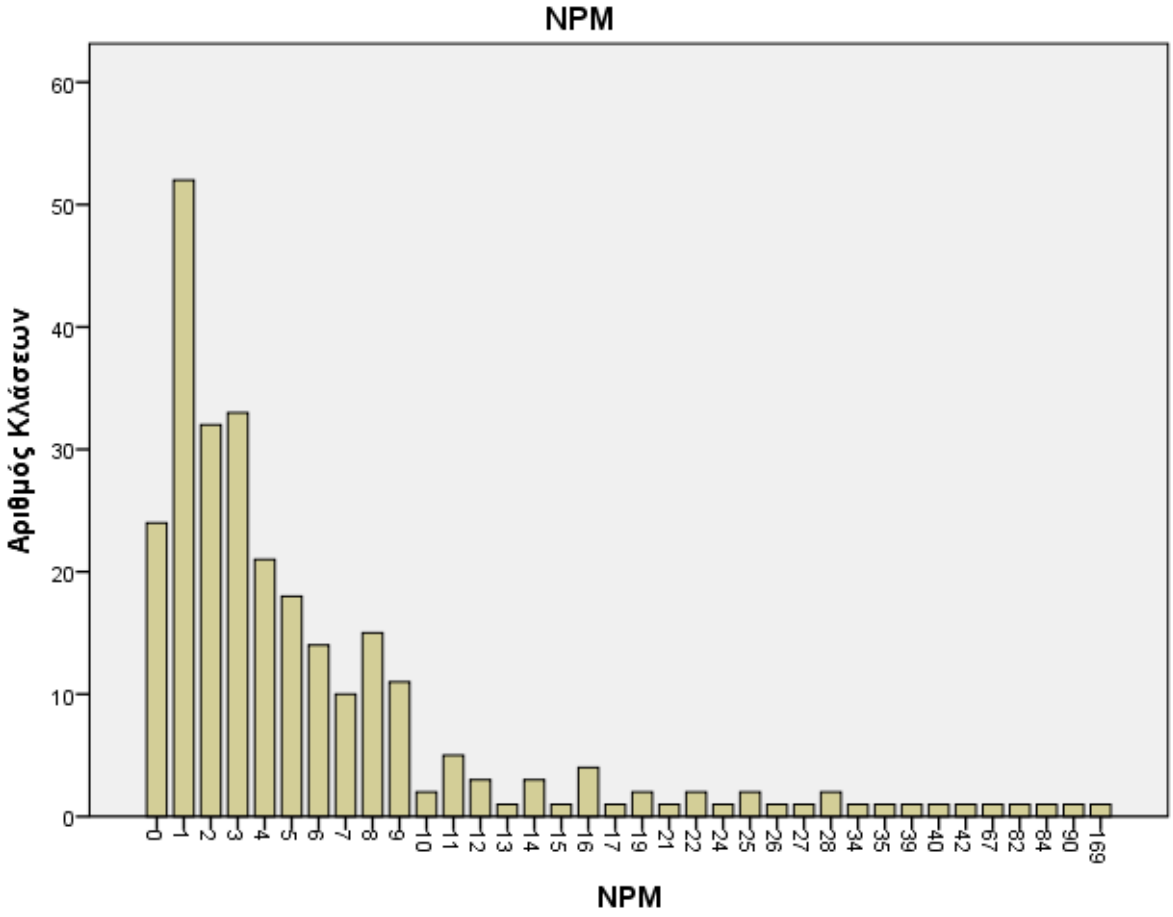

LCOM3
Αριθμός Κλάσεων
80
60
40
20
0
,000000000
,250000000
,392344498
,440000000
,500000000
,555555556
,595238095
,622222222
,642857143
,672727273
,691666667
,711538462
,722222222
,751700680
,769230769
,775000000
,782051282
,800000000
,813131313
,830140485
,858585859
,875000000
,892361111
,900621118
,927203065
,940000000
,951111111
,993197279
1,071428571
1,142857143
1,800000000
LCOM3

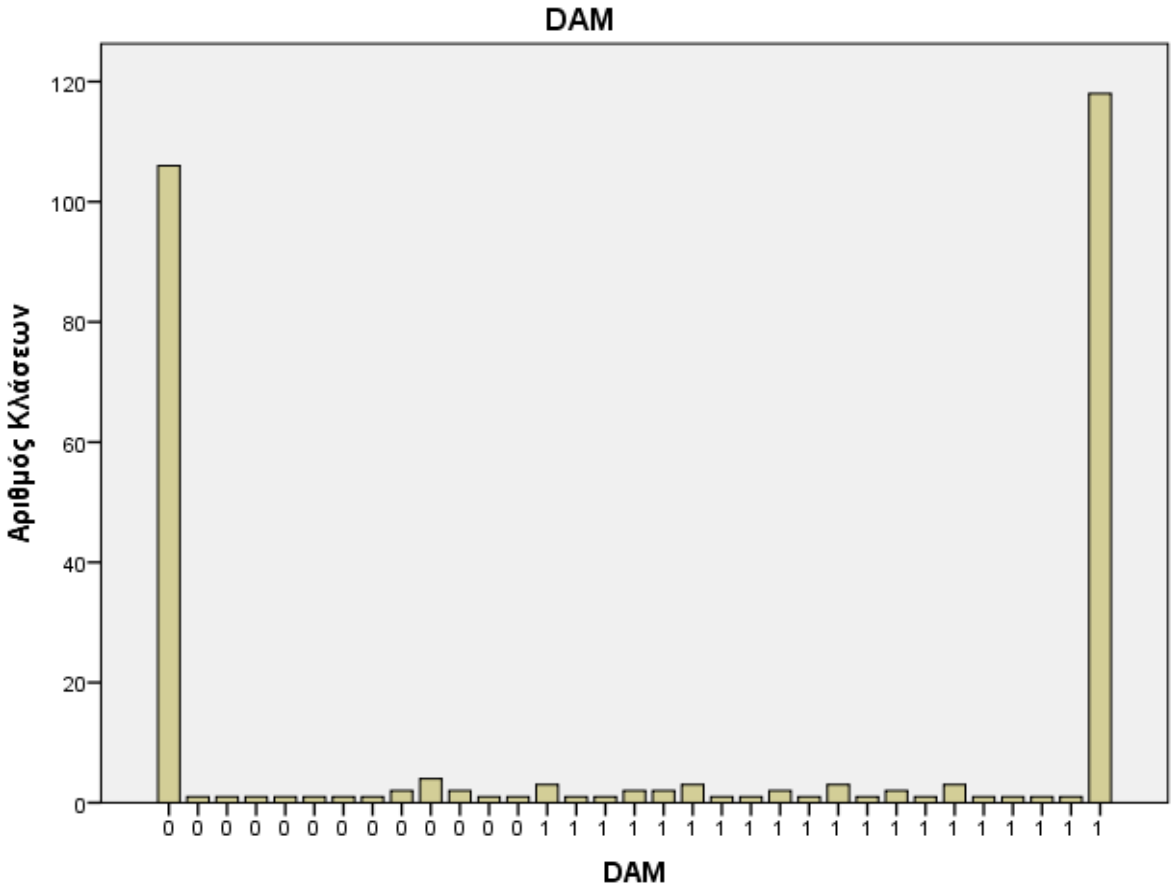

IC
Αριθμός Κλάσεων
200
150
100
50
0
0
1
2
3
4
IC

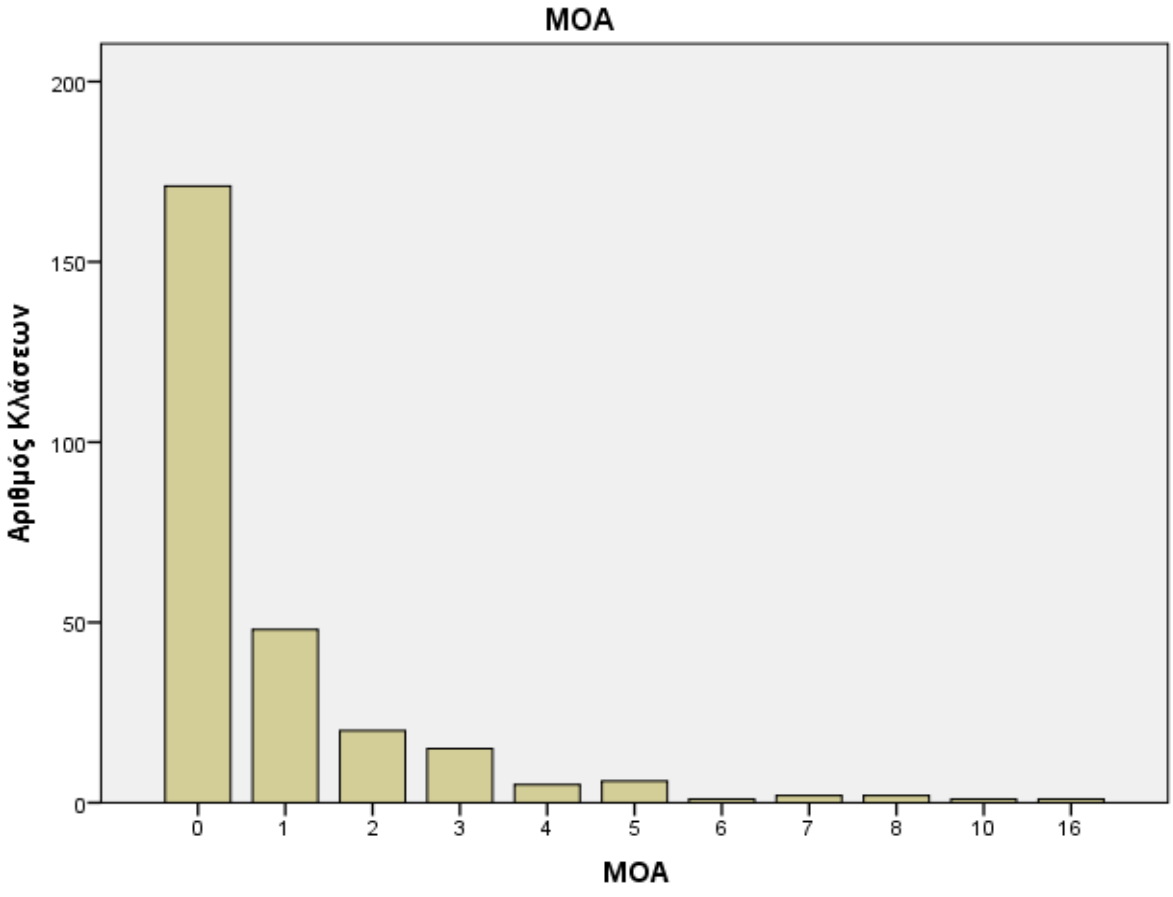

CBM
Αριθμός Κλάσεων
200
150
100
50
0
0
1
2
3
4
5
6
7
9
11
12
13
18
CBM

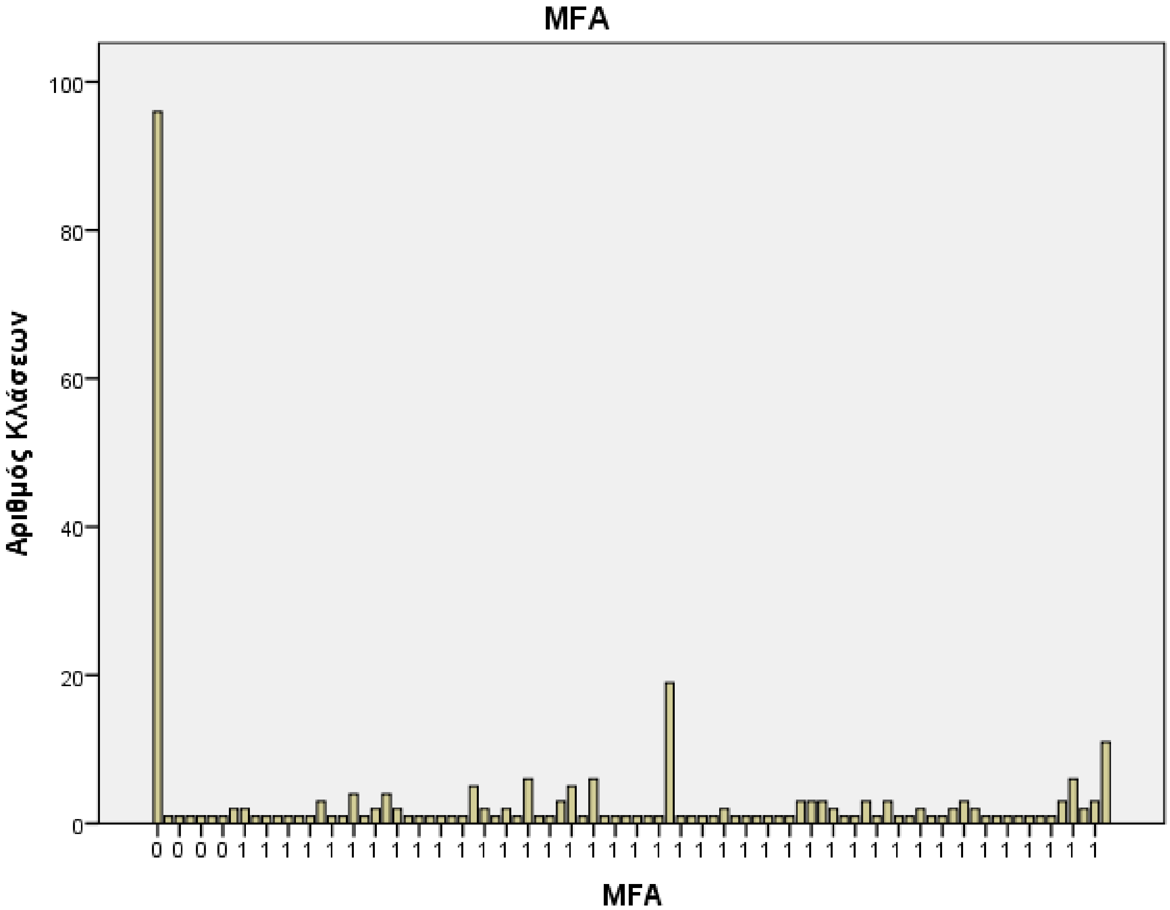

CA
Αριθμός Κλάσεων
80
60
40
20
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 24 29 31 33 34 36 38 40 43 44 46 51 58 64 79 99 137
CA

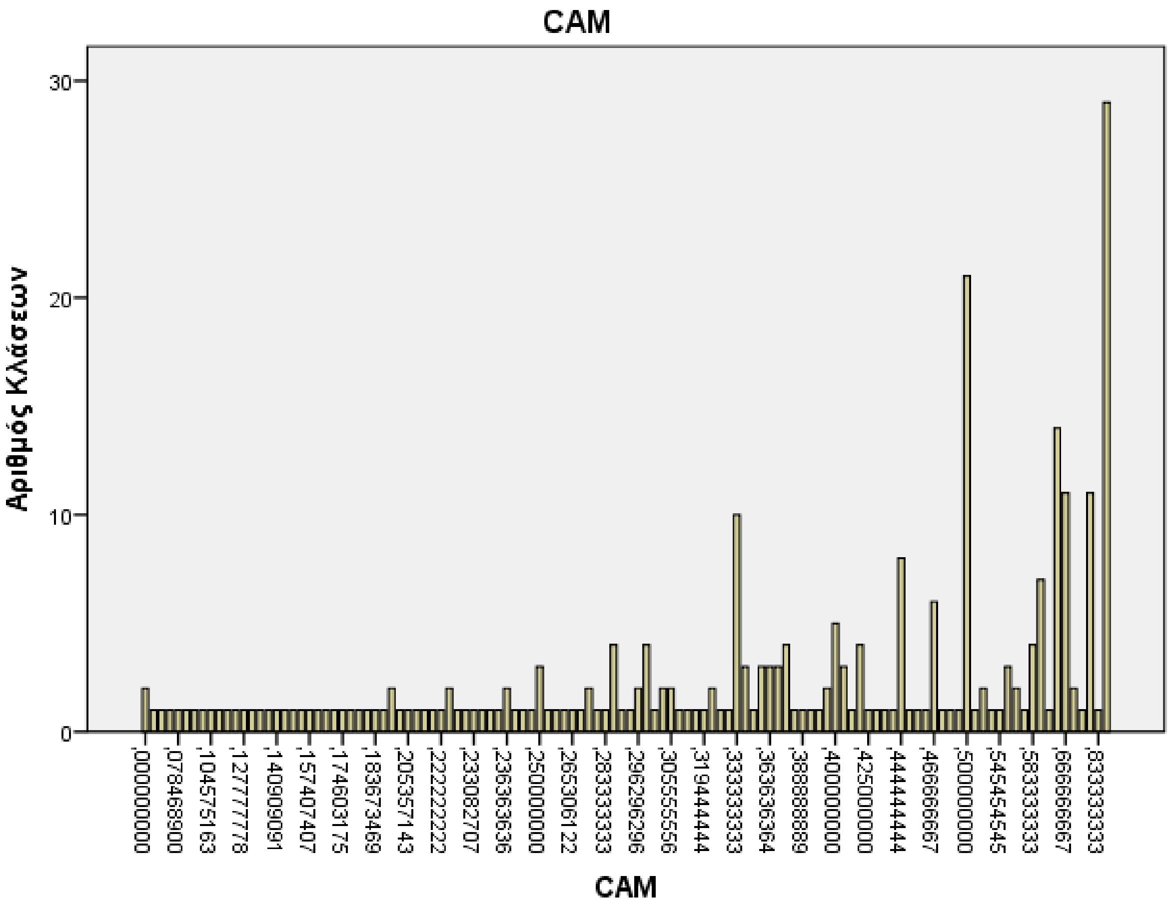

CE
Αριθμός Κλάσεων
40
30
20
10
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 17 18 19 20 21 22 24 25 27 30 39 40 47 48
CE

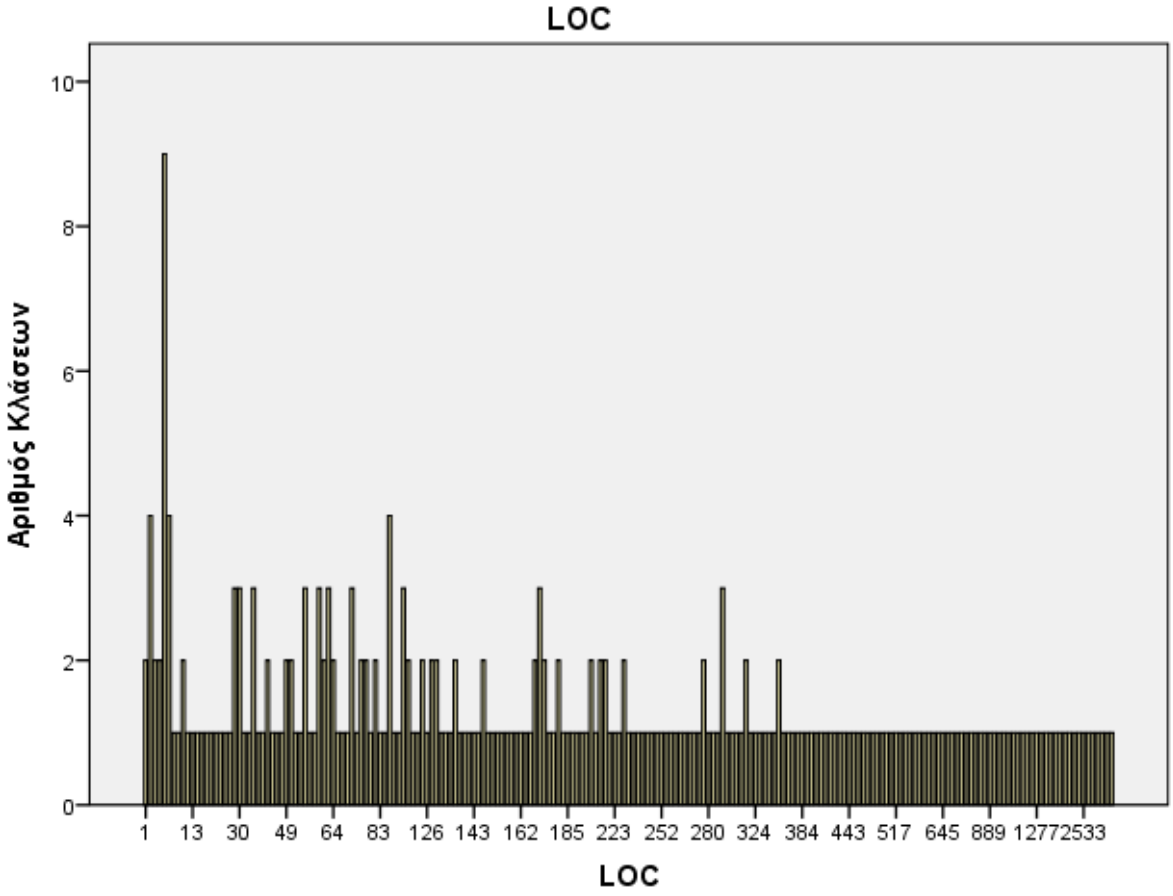

NPM
Αριθμός Κλάσεων
60
50
40
30
20
10
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 19 21 22 24 25 26 27 28 34 35 39 40 42 67 82 84 90 169
NPM

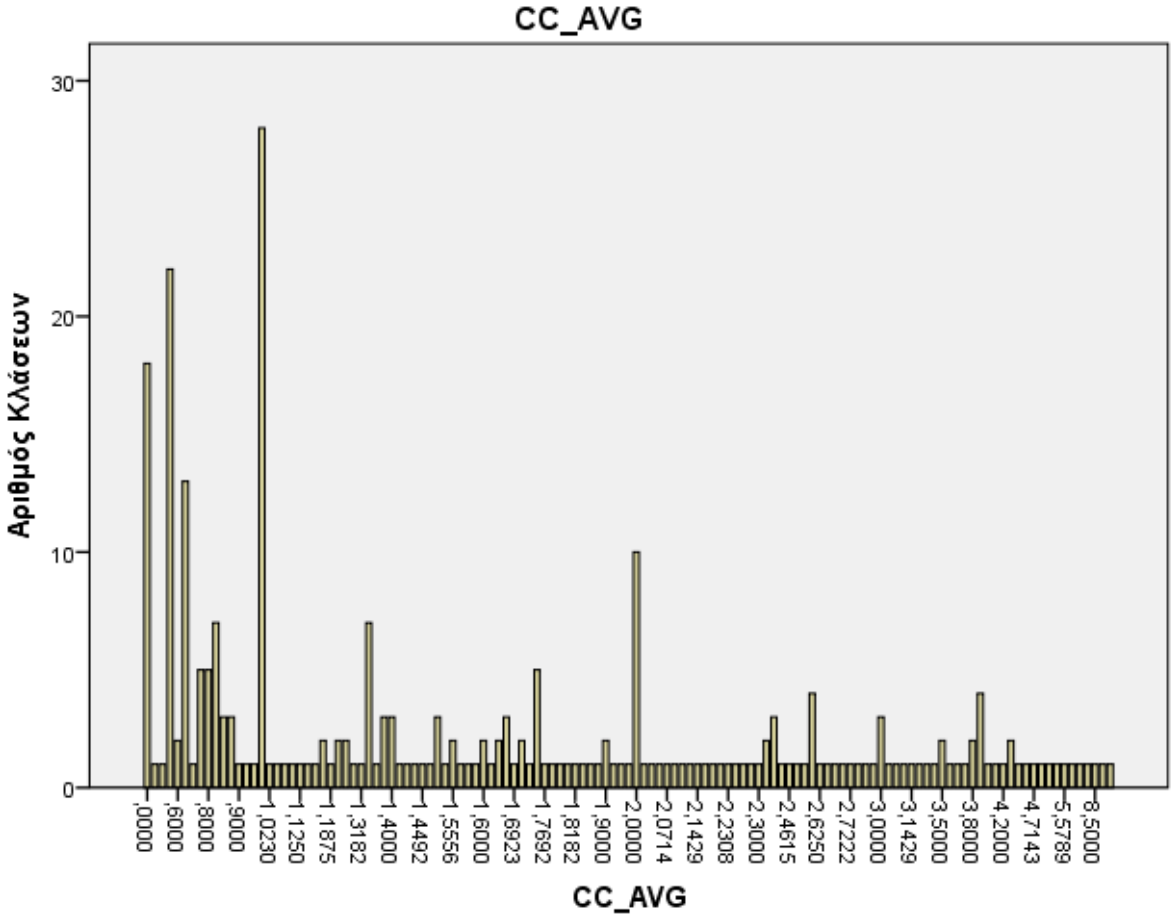

DAM
Αριθμός Κλάσεων
120
100
80
60
40
20
0
0 0 0 0 0 0 0 0 0 0 0 0 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
DAM

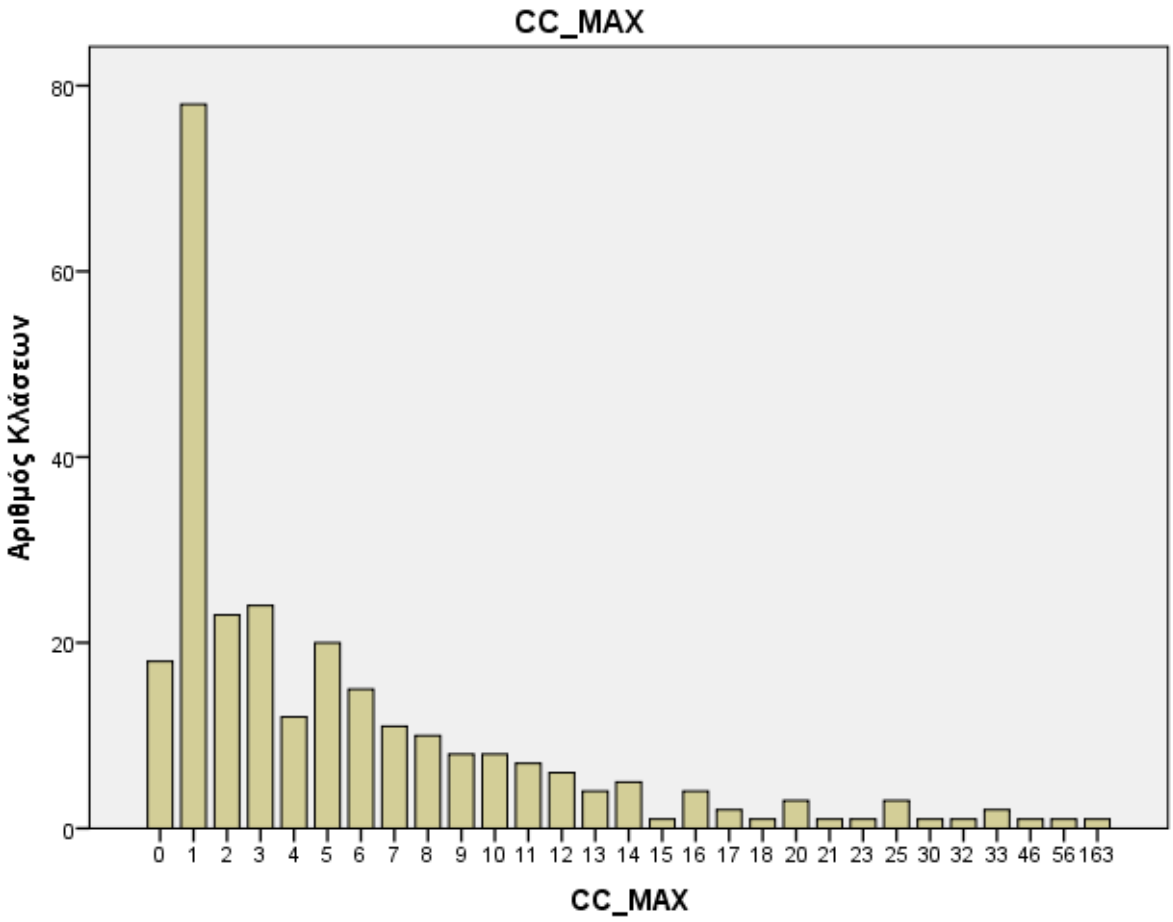

MOA
Αριθμός Κλάσεων
200
150
100
50
0
0 1 2 3 4 5 6 7 8 10 16
MOA

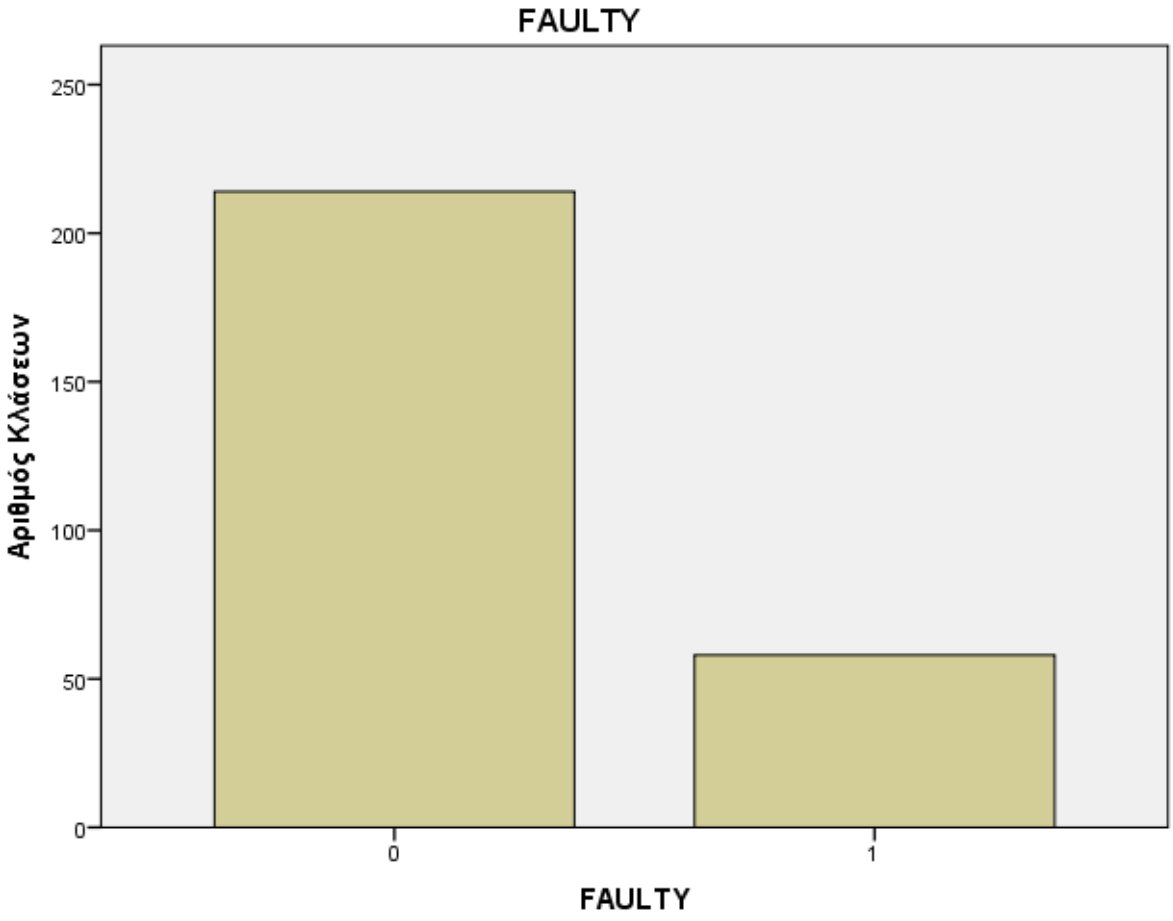

MFA
Αριθμός Κλάσεων
MFA


CAM
Αριθμός Κλάσεων
CAM


LOC
Αριθμός Κλάσεων
0
2
4
6
8
10
1 13 30 49 64 83 126 143 162 185 223 252 280 324 384 443 517 645 889 12772533
LOC


CC_AVG
Αριθμός Κλάσεων
0
10
20
30
,0000 ,6000 ,8000 ,9000 1,0230 1,1250 1,1875 1,3182 1,4000 1,4492 1,5556 1,6000 1,6923 1,7692 1,8182 1,9000 2,0000 2,0714 2,1429 2,2308 2,3000 2,4615 2,6250 2,7222 3,0000 3,1429 3,5000 3,8000 4,2000 4,7143 5,5789 8,5000
CC_AVG


CC_MAX
Αριθμός Κλάσεων
80
60
40
20
0
0 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 20 21 23 25 30 32 33 46 56 163
CC_MAX


FAULTY
Αριθμός Κλάσεων
250
200
150
100
50
0
0
1
FAULTY

Β.2 Απλή Γραμμική Παλινδρόμηση

Variables Entered/Removed[a]

Model	Variables Entered	Variables Removed	Method
1	WMC[b]	.	Enter

a. Dependent Variable: NO_OF_FAULTS

b. All requested variables entered.

Model Summary[b]

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	,465[a]	,216	,213	3,641

a. Predictors: (Constant), WMC

b. Dependent Variable: NO_OF_FAULTS

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	985,757	1	985,757	74,350	,000[b]
	Residual	3579,757	270	13,258		
	Total	4565,515	271			

a. Dependent Variable: NO_OF_FAULTS

b. Predictors: (Constant), WMC

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	,624	,239		2,616	,009
	WMC	,062	,007	,465	8,623	,000

a. Dependent Variable: NO_OF_FAULTS

Residuals Statistics[a]

	Minimum	Maximum	Mean	Std. Deviation	N
Predicted Value	,69	25,46	1,40	1,907	272
Std. Predicted Value	-,376	12,614	,000	1,000	272
Standard Error of Predicted Value	,221	2,799	,253	,183	272
Adjusted Predicted Value	,69	62,23	1,51	3,832	272
Residual	-25,462	31,241	,000	3,634	272
Std. Residual	-6,993	8,580	,000	,998	272
Stud. Residual	-10,932	9,353	-,011	1,148	272
Deleted Residual	-62,226	37,126	-,110	5,182	272
Stud. Deleted Residual	-14,615	11,355	-,015	1,357	272
Mahal. Distance	,000	159,114	,996	10,029	272
Cook's Distance	,000	86,273	,353	5,253	272
Centered Leverage Value	,000	,587	,004	,037	272

a. Dependent Variable: NO_OF_FAULTS

Variables Entered/Removed[a]

Model	Variables Entered	Variables Removed	Method
1	DIT[b]	.	Enter

a. Dependent Variable: NO_OF_FAULTS

b. All requested variables entered.

Model Summary[b]

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	,132[a]	,017	,014	4,076

a. Predictors: (Constant), DIT

b. Dependent Variable: NO_OF_FAULTS

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	79,183	1	79,183	4,765	,030[b]
	Residual	4486,331	270	16,616		
	Total	4565,515	271			

a. Dependent Variable: NO_OF_FAULTS

b. Predictors: (Constant), DIT

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	,685	,412		1,663	,097
	DIT	,254	,116	,132	2,183	,030

a. Dependent Variable: NO_OF_FAULTS

Residuals Statistics[a]

	Minimum	Maximum	Mean	Std. Deviation	N
Predicted Value	,94	2,72	1,40	,541	272
Std. Predicted Value	-,861	2,430	,000	1,000	272
Standard Error of Predicted Value	,248	,651	,337	,092	272
Adjusted Predicted Value	,80	2,79	1,41	,545	272
Residual	-2,718	43,298	,000	4,069	272
Std. Residual	-,667	10,622	,000	,998	272
Stud. Residual	-,675	10,648	,000	1,001	272
Deleted Residual	-2,789	43,507	-,002	4,092	272
Stud. Deleted Residual	-,675	13,954	,016	1,158	272
Mahal. Distance	,006	5,905	,996	1,245	272
Cook's Distance	,000	,273	,003	,019	272
Centered Leverage Value	,000	,022	,004	,005	272

a. Dependent Variable: NO_OF_FAULTS

Variables Entered/Removed[a]

Model	Variables Entered	Variables Removed	Method
1	NOC[b]	.	Enter

a. Dependent Variable: NO_OF_FAULTS

b. All requested variables entered.

Model Summary[b]

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	,041[a]	,002	-,002	4,109

a. Predictors: (Constant), NOC

b. Dependent Variable: NO_OF_FAULTS

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	7,756	1	7,756	,459	,498[b]
	Residual	4557,759	270	16,881		
	Total	4565,515	271			

a. Dependent Variable: NO_OF_FAULTS

b. Predictors: (Constant), NOC

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	1,431	,252		5,674	,000
	NOC	-,063	,093	-,041	-,678	,498

a. Dependent Variable: NO_OF_FAULTS

Residuals Statistics[a]

	Minimum	Maximum	Mean	Std. Deviation	N
Predicted Value	-,77	1,43	1,40	,169	272
Std. Predicted Value	-12,852	,160	,000	1,000	272
Standard Error of Predicted Value	,252	3,217	,280	,214	272
Adjusted Predicted Value	-1,99	1,44	1,40	,231	272
Residual	-1,431	43,569	,000	4,101	272
Std. Residual	-,348	10,604	,000	,998	272
Stud. Residual	-,349	10,624	,000	1,000	272
Deleted Residual	-1,437	43,733	,004	4,118	272
Stud. Deleted Residual	-,348	13,901	,016	1,155	272
Mahal. Distance	,026	165,181	,996	10,437	272
Cook's Distance	,000	,214	,002	,015	272
Centered Leverage Value	,000	,610	,004	,039	272

a. Dependent Variable: NO_OF_FAULTS

Variables Entered/Removed[a]

Model	Variables Entered	Variables Removed	Method
1	CBO[b]	.	Enter

a. Dependent Variable: NO_OF_FAULTS

b. All requested variables entered.

Model Summary[b]

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	,531[a]	,282	,280	3,483

a. Predictors: (Constant), CBO

b. Dependent Variable: NO_OF_FAULTS

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	1289,702	1	1289,702	106,300	,000[b]
	Residual	3275,813	270	12,133		
	Total	4565,515	271			

a. Dependent Variable: NO_OF_FAULTS

b. Predictors: (Constant), CBO

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	-,195	,262		-,742	,459
	CBO	,133	,013	,531	10,310	,000

a. Dependent Variable: NO_OF_FAULTS

Residuals Statistics[a]

	Minimum	Maximum	Mean	Std. Deviation	N
Predicted Value	-,06	21,32	1,40	2,182	272
Std. Predicted Value	-,672	9,131	,000	1,000	272
Standard Error of Predicted Value	,211	1,944	,258	,151	272
Adjusted Predicted Value	-,06	20,57	1,40	2,161	272
Residual	-9,285	37,357	,000	3,477	272
Std. Residual	-2,666	10,725	,000	,998	272
Stud. Residual	-2,862	10,911	,000	1,013	272
Deleted Residual	-10,700	38,666	,003	3,582	272
Stud. Deleted Residual	-2,901	14,566	,015	1,177	272
Mahal. Distance	,000	83,384	,996	5,875	272
Cook's Distance	,000	2,086	,016	,141	272
Centered Leverage Value	,000	,308	,004	,022	272

a. Dependent Variable: NO_OF_FAULTS

Variables Entered/Removed[a]

Model	Variables Entered	Variables Removed	Method
1	RFC[b]	.	Enter

a. Dependent Variable: NO_OF_FAULTS

b. All requested variables entered.

Model Summary[b]

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	,691[a]	,478	,476	2,971

a. Predictors: (Constant), RFC

b. Dependent Variable: NO_OF_FAULTS

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	2181,982	1	2181,982	247,169	,000[b]
	Residual	2383,533	270	8,828		
	Total	4565,515	271			

a. Dependent Variable: NO_OF_FAULTS

b. Predictors: (Constant), RFC

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	-,535	,218		-2,451	,015
	RFC	,051	,003	,691	15,722	,000

a. Dependent Variable: NO_OF_FAULTS

Residuals Statistics[a]

	Minimum	Maximum	Mean	Std. Deviation	N
Predicted Value	-,48	24,49	1,40	2,838	272
Std. Predicted Value	-,665	8,138	,000	1,000	272
Standard Error of Predicted Value	,180	1,480	,221	,127	272
Adjusted Predicted Value	-,49	32,57	1,41	2,966	272
Residual	-24,495	23,435	,000	2,966	272
Std. Residual	-8,244	7,887	,000	,998	272
Stud. Residual	-9,507	8,763	-,001	1,070	272
Deleted Residual	-32,574	28,930	-,004	3,430	272
Stud. Deleted Residual	-11,635	10,341	-,001	1,198	272
Mahal. Distance	,000	66,220	,996	5,532	272
Cook's Distance	,000	14,906	,094	1,055	272
Centered Leverage Value	,000	,244	,004	,020	272

a. Dependent Variable: NO_OF_FAULTS

Variables Entered/Removed[a]

Model	Variables Entered	Variables Removed	Method
1	LCOM[b]	.	Enter

a. Dependent Variable: NO_OF_FAULTS

b. All requested variables entered.

Model Summary[b]

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	,730[a]	,532	,531	2,812

a. Predictors: (Constant), LCOM

b. Dependent Variable: NO_OF_FAULTS

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	2430,086	1	2430,086	307,256	,000[b]
	Residual	2135,429	270	7,909		
	Total	4565,515	271			

a. Dependent Variable: NO_OF_FAULTS

b. Predictors: (Constant), LCOM

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	,868	,173		5,010	,000
	LCOM	,003	,000	,730	17,529	,000

a. Dependent Variable: NO_OF_FAULTS

Residuals Statistics[a]

	Minimum	Maximum	Mean	Std. Deviation	N
Predicted Value	,87	36,79	1,40	2,995	272
Std. Predicted Value	-,179	11,818	,000	1,000	272
Standard Error of Predicted Value	,171	2,026	,192	,146	272
Adjusted Predicted Value	,85	27,94	1,38	2,713	272
Residual	-19,898	15,910	,000	2,807	272
Std. Residual	-7,075	5,657	,000	,998	272
Stud. Residual	-7,649	5,668	,003	1,035	272
Deleted Residual	-23,256	17,062	,022	3,068	272
Stud. Deleted Residual	-8,627	6,027	,003	1,085	272
Mahal. Distance	,000	139,669	,996	9,347	272
Cook's Distance	,000	9,553	,059	,652	272
Centered Leverage Value	,000	,515	,004	,034	272

a. Dependent Variable: NO_OF_FAULTS

Variables Entered/Removed[a]

Model	Variables Entered	Variables Removed	Method
1	LCOM3[b]	.	Enter

a. Dependent Variable: NO_OF_FAULTS

b. All requested variables entered.

Model Summary[b]

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	,092[a]	,009	,005	4,095

a. Predictors: (Constant), LCOM3

b. Dependent Variable: NO_OF_FAULTS

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	38,886	1	38,886	2,319	,129[b]
	Residual	4526,629	270	16,765		
	Total	4565,515	271			

a. Dependent Variable: NO_OF_FAULTS

b. Predictors: (Constant), LCOM3

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	2,059	,497		4,147	,000
	LCOM3	-,622	,408	-,092	-1,523	,129

a. Dependent Variable: NO_OF_FAULTS

Residuals Statistics[a]

	Minimum	Maximum	Mean	Std. Deviation	N
Predicted Value	,82	2,06	1,40	,379	272
Std. Predicted Value	-1,554	1,729	,000	1,000	272
Standard Error of Predicted Value	,248	,497	,340	,086	272
Adjusted Predicted Value	,76	2,09	1,41	,379	272
Residual	-2,059	43,490	,000	4,087	272
Std. Residual	-,503	10,621	,000	,998	272
Stud. Residual	-,507	10,643	,000	1,000	272
Deleted Residual	-2,090	43,663	-,003	4,105	272
Stud. Deleted Residual	-,506	13,942	,016	1,156	272
Mahal. Distance	,000	2,990	,996	1,017	272
Cook's Distance	,000	,225	,002	,015	272
Centered Leverage Value	,000	,011	,004	,004	272

a. Dependent Variable: NO_OF_FAULTS

Variables Entered/Removed[a]

Model	Variables Entered	Variables Removed	Method
1	IC[b]	.	Enter

a. Dependent Variable: NO_OF_FAULTS

b. All requested variables entered.

Model Summary[b]

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	,223[a]	,050	,046	4,009

a. Predictors: (Constant), IC

b. Dependent Variable: NO_OF_FAULTS

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	226,567	1	226,567	14,099	,000[b]
	Residual	4338,948	270	16,070		
	Total	4565,515	271			

a. Dependent Variable: NO_OF_FAULTS

b. Predictors: (Constant), IC

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	,760	,298		2,553	,011
	IC	,985	,262	,223	3,755	,000

a. Dependent Variable: NO_OF_FAULTS

Residuals Statistics[a]

	Minimum	Maximum	Mean	Std. Deviation	N
Predicted Value	,76	4,70	1,40	,914	272
Std. Predicted Value	-,705	3,604	,000	1,000	272
Standard Error of Predicted Value	,259	,911	,323	,118	272
Adjusted Predicted Value	,64	4,96	1,40	,917	272
Residual	-4,700	41,285	,000	4,001	272
Std. Residual	-1,172	10,299	,000	,998	272
Stud. Residual	-1,204	10,442	,000	1,007	272
Deleted Residual	-4,956	42,441	,002	4,070	272
Stud. Deleted Residual	-1,205	13,499	,016	1,150	272
Mahal. Distance	,139	12,990	,996	2,096	272
Cook's Distance	,000	1,526	,009	,094	272
Centered Leverage Value	,001	,048	,004	,008	272

a. Dependent Variable: NO_OF_FAULTS

Variables Entered/Removed[a]

Model	Variables Entered	Variables Removed	Method
1	CBM[b]	.	Enter

a. Dependent Variable: NO_OF_FAULTS

b. All requested variables entered.

Model Summary[b]

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	,308[a]	,095	,092	3,912

a. Predictors: (Constant), CBM

b. Dependent Variable: NO_OF_FAULTS

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	433,952	1	433,952	28,359	,000[b]
	Residual	4131,563	270	15,302		
	Total	4565,515	271			

a. Dependent Variable: NO_OF_FAULTS

b. Predictors: (Constant), CBM

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	,756	,267		2,834	,005
	CBM	,446	,084	,308	5,325	,000

a. Dependent Variable: NO_OF_FAULTS

Residuals Statistics[a]

	Minimum	Maximum	Mean	Std. Deviation	N
Predicted Value	,76	8,78	1,40	1,265	272
Std. Predicted Value	-,513	5,826	,000	1,000	272
Standard Error of Predicted Value	,240	1,405	,299	,153	272
Adjusted Predicted Value	,65	8,89	1,40	1,257	272
Residual	-6,103	40,234	,000	3,905	272
Std. Residual	-1,560	10,285	,000	,998	272
Stud. Residual	-1,605	10,442	,000	1,008	272
Deleted Residual	-6,455	41,466	,003	3,986	272
Stud. Deleted Residual	-1,609	13,498	,016	1,153	272
Mahal. Distance	,026	33,940	,996	3,714	272
Cook's Distance	,000	1,670	,011	,104	272
Centered Leverage Value	,000	,125	,004	,014	272

a. Dependent Variable: NO_OF_FAULTS

Variables Entered/Removed[a]

Model	Variables Entered	Variables Removed	Method
1	AMC[b]	.	Enter

a. Dependent Variable: NO_OF_FAULTS

b. All requested variables entered.

Model Summary[b]

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	,065[a]	,004	,001	4,103

a. Predictors: (Constant), AMC

b. Dependent Variable: NO_OF_FAULTS

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	19,346	1	19,346	1,149	,285[b]
	Residual	4546,169	270	16,838		
	Total	4565,515	271			

a. Dependent Variable: NO_OF_FAULTS

b. Predictors: (Constant), AMC

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	1,179	,326		3,620	,000
	AMC	,007	,006	,065	1,072	,285

a. Dependent Variable: NO_OF_FAULTS

Residuals Statistics[a]

	Minimum	Maximum	Mean	Std. Deviation	N
Predicted Value	1,18	4,60	1,40	,267	272
Std. Predicted Value	-,843	11,969	,000	1,000	272
Standard Error of Predicted Value	,249	2,994	,303	,178	272
Adjusted Predicted Value	1,17	9,84	1,42	,544	272
Residual	-4,602	43,580	,000	4,096	272
Std. Residual	-1,122	10,621	,000	,998	272
Stud. Residual	-1,640	10,640	-,002	1,003	272
Deleted Residual	-9,840	43,742	-,020	4,147	272
Stud. Deleted Residual	-1,645	13,937	,014	1,158	272
Mahal. Distance	,000	143,254	,996	8,714	272
Cook's Distance	,000	1,531	,008	,094	272
Centered Leverage Value	,000	,529	,004	,032	272

a. Dependent Variable: NO_OF_FAULTS

Variables Entered/Removed[a]

Model	Variables Entered	Variables Removed	Method
1	CA[b]	.	Enter

a. Dependent Variable: NO_OF_FAULTS

b. All requested variables entered.

Model Summary[b]

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	,452[a]	,204	,201	3,668

a. Predictors: (Constant), CA

b. Dependent Variable: NO_OF_FAULTS

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	933,290	1	933,290	69,376	,000[b]
	Residual	3632,225	270	13,453		
	Total	4565,515	271			

a. Dependent Variable: NO_OF_FAULTS

b. Predictors: (Constant), CA

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	,478	,249		1,924	,055
	CA	,129	,016	,452	8,329	,000

a. Dependent Variable: NO_OF_FAULTS

Residuals Statistics[a]

	Minimum	Maximum	Mean	Std. Deviation	N
Predicted Value	,48	18,21	1,40	1,856	272
Std. Predicted Value	-,499	9,056	,000	1,000	272
Standard Error of Predicted Value	,222	2,030	,269	,163	272
Adjusted Predicted Value	,48	16,09	1,40	1,815	272
Residual	-8,762	39,345	,000	3,661	272
Std. Residual	-2,389	10,727	,000	,998	272
Stud. Residual	-2,466	10,853	,001	1,012	272
Deleted Residual	-9,813	40,272	,005	3,765	272
Stud. Deleted Residual	-2,490	14,427	,016	1,172	272
Mahal. Distance	,000	82,006	,996	5,929	272
Cook's Distance	,000	1,389	,015	,111	272
Centered Leverage Value	,000	,303	,004	,022	272

a. Dependent Variable: NO_OF_FAULTS

Variables Entered/Removed[a]

Model	Variables Entered	Variables Removed	Method
1	CE[b]	.	Enter

a. Dependent Variable: NO_OF_FAULTS

b. All requested variables entered.

Model Summary[b]

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	,607[a]	,369	,367	3,267

a. Predictors: (Constant), CE

b. Dependent Variable: NO_OF_FAULTS

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	1684,557	1	1684,557	157,875	,000[b]
	Residual	2880,957	270	10,670		
	Total	4565,515	271			

a. Dependent Variable: NO_OF_FAULTS

b. Predictors: (Constant), CE

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	-,791	,264		-2,996	,003
	CE	,352	,028	,607	12,565	,000

a. Dependent Variable: NO_OF_FAULTS

Residuals Statistics[a]

	Minimum	Maximum	Mean	Std. Deviation	N
Predicted Value	-,79	16,12	1,40	2,493	272
Std. Predicted Value	-,881	5,903	,000	1,000	272
Standard Error of Predicted Value	,198	1,188	,253	,121	272
Adjusted Predicted Value	-,83	18,05	1,40	2,469	272
Residual	-15,770	32,049	,000	3,260	272
Std. Residual	-4,828	9,811	,000	,998	272
Stud. Residual	-5,165	10,245	,001	1,026	272
Deleted Residual	-18,048	34,943	,008	3,449	272
Stud. Deleted Residual	-5,430	13,079	,012	1,150	272
Mahal. Distance	,001	34,850	,996	3,642	272
Cook's Distance	,000	4,739	,031	,313	272
Centered Leverage Value	,000	,129	,004	,013	272

a. Dependent Variable: NO_OF_FAULTS

Variables Entered/Removed[a]

Model	Variables Entered	Variables Removed	Method
1	NPM[b]	.	Enter

a. Dependent Variable: NO_OF_FAULTS

b. All requested variables entered.

Model Summary[b]

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	,724[a]	,524	,522	2,838

a. Predictors: (Constant), NPM

b. Dependent Variable: NO_OF_FAULTS

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	2390,814	1	2390,814	296,832	,000[b]
	Residual	2174,701	270	8,054		
	Total	4565,515	271			

a. Dependent Variable: NO_OF_FAULTS

b. Predictors: (Constant), NPM

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	-,032	,191		-,166	,869
	NPM	,196	,011	,724	17,229	,000

a. Dependent Variable: NO_OF_FAULTS

Residuals Statistics[a]

	Minimum	Maximum	Mean	Std. Deviation	N
Predicted Value	-,03	33,11	1,40	2,970	272
Std. Predicted Value	-,483	10,674	,000	1,000	272
Standard Error of Predicted Value	,172	1,848	,204	,133	272
Adjusted Predicted Value	-,04	24,35	1,38	2,702	272
Residual	-16,440	15,463	,000	2,833	272
Std. Residual	-5,793	5,448	,000	,998	272
Stud. Residual	-6,100	5,522	,004	1,039	272
Deleted Residual	-18,231	20,649	,025	3,113	272
Stud. Deleted Residual	-6,557	5,852	,004	1,078	272
Mahal. Distance	,000	113,928	,996	7,489	272
Cook's Distance	,000	11,225	,061	,700	272
Centered Leverage Value	,000	,420	,004	,028	272

a. Dependent Variable: NO_OF_FAULTS

Variables Entered/Removed[a]

Model	Variables Entered	Variables Removed	Method
1	DAM[b]	.	Enter

a. Dependent Variable: NO_OF_FAULTS

b. All requested variables entered.

Model Summary[b]

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	,219[a]	,048	,044	4,012

a. Predictors: (Constant), DAM

b. Dependent Variable: NO_OF_FAULTS

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	219,171	1	219,171	13,615	,000[b]
	Residual	4346,344	270	16,098		
	Total	4565,515	271			

a. Dependent Variable: NO_OF_FAULTS

b. Predictors: (Constant), DAM

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	,375	,370		1,014	,311
	DAM	1,926	,522	,219	3,690	,000

a. Dependent Variable: NO_OF_FAULTS

Residuals Statistics[a]

	Minimum	Maximum	Mean	Std. Deviation	N
Predicted Value	,38	2,30	1,40	,899	272
Std. Predicted Value	-1,144	,997	,000	1,000	272
Standard Error of Predicted Value	,244	,370	,342	,034	272
Adjusted Predicted Value	,34	2,32	1,41	,900	272
Residual	-2,301	42,878	,000	4,005	272
Std. Residual	-,574	10,687	,000	,998	272
Stud. Residual	-,576	10,719	,000	1,001	272
Deleted Residual	-2,318	43,138	-,001	4,030	272
Stud. Deleted Residual	-,575	14,117	,016	1,161	272
Mahal. Distance	,005	1,310	,996	,366	272
Cook's Distance	,000	,348	,003	,023	272
Centered Leverage Value	,000	,005	,004	,001	272

a. Dependent Variable: NO_OF_FAULTS

Variables Entered/Removed[a]

Model	Variables Entered	Variables Removed	Method
1	MOA[b]	.	Enter

a. Dependent Variable: NO_OF_FAULTS

b. All requested variables entered.

Model Summary[b]

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	,515[a]	,266	,263	3,524

a. Predictors: (Constant), MOA

b. Dependent Variable: NO_OF_FAULTS

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	1212,320	1	1212,320	97,616	,000[b]
	Residual	3353,194	270	12,419		
	Total	4565,515	271			

a. Dependent Variable: NO_OF_FAULTS

b. Predictors: (Constant), MOA

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	,353	,239		1,480	,140
	MOA	1,167	,118	,515	9,880	,000

a. Dependent Variable: NO_OF_FAULTS

Residuals Statistics[a]

	Minimum	Maximum	Mean	Std. Deviation	N
Predicted Value	,35	19,02	1,40	2,115	272
Std. Predicted Value	-,497	8,330	,000	1,000	272
Standard Error of Predicted Value	,214	1,796	,269	,139	272
Adjusted Predicted Value	,33	25,70	1,42	2,296	272
Residual	-19,024	32,978	,000	3,518	272
Std. Residual	-5,398	9,358	,000	,998	272
Stud. Residual	-6,274	9,846	-,001	1,040	272
Deleted Residual	-25,699	36,507	-,011	3,835	272
Stud. Deleted Residual	-6,776	12,275	,008	1,150	272
Mahal. Distance	,003	69,396	,996	4,787	272
Cook's Distance	,000	6,907	,051	,524	272
Centered Leverage Value	,000	,256	,004	,018	272

a. Dependent Variable: NO_OF_FAULTS

Variables Entered/Removed[a]

Model	Variables Entered	Variables Removed	Method
1	MFA[b]	.	Enter

a. Dependent Variable: NO_OF_FAULTS

b. All requested variables entered.

Model Summary[b]

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	,040[a]	,002	-,002	4,109

a. Predictors: (Constant), MFA

b. Dependent Variable: NO_OF_FAULTS

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	7,262	1	7,262	,430	,512[b]
	Residual	4558,253	270	16,882		
	Total	4565,515	271			

a. Dependent Variable: NO_OF_FAULTS

b. Predictors: (Constant), MFA

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	1,196	,404		2,959	,003
	MFA	,375	,571	,040	,656	,512

a. Dependent Variable: NO_OF_FAULTS

Residuals Statistics[a]

	Minimum	Maximum	Mean	Std. Deviation	N
Predicted Value	1,20	1,57	1,40	,164	272
Std. Predicted Value	-1,275	1,014	,000	1,000	272
Standard Error of Predicted Value	,249	,404	,349	,051	272
Adjusted Predicted Value	,98	1,58	1,40	,165	272
Residual	-1,570	43,521	,000	4,101	272
Std. Residual	-,382	10,592	,000	,998	272
Stud. Residual	-,384	10,616	,000	1,001	272
Deleted Residual	-1,582	43,715	,000	4,126	272
Stud. Deleted Residual	-,383	13,882	,016	1,155	272
Mahal. Distance	,000	1,626	,996	,551	272
Cook's Distance	,000	,251	,003	,019	272
Centered Leverage Value	,000	,006	,004	,002	272

a. Dependent Variable: NO_OF_FAULTS

Variables Entered/Removed[a]

Model	Variables Entered	Variables Removed	Method
1	CAM[b]	.	Enter

a. Dependent Variable: NO_OF_FAULTS

b. All requested variables entered.

Model Summary[b]

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	,307[a]	,094	,091	3,913

a. Predictors: (Constant), CAM

b. Dependent Variable: NO_OF_FAULTS

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	430,456	1	430,456	28,107	,000[b]
	Residual	4135,059	270	15,315		
	Total	4565,515	271			

a. Dependent Variable: NO_OF_FAULTS

b. Predictors: (Constant), CAM

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	3,753	,503		7,468	,000
	CAM	-5,014	,946	-,307	-5,302	,000

a. Dependent Variable: NO_OF_FAULTS

Residuals Statistics[a]

	Minimum	Maximum	Mean	Std. Deviation	N
Predicted Value	-1,26	3,75	1,40	1,260	272
Std. Predicted Value	-2,115	1,864	,000	1,000	272
Standard Error of Predicted Value	,237	,556	,321	,097	272
Adjusted Predicted Value	-1,35	3,82	1,40	1,263	272
Residual	-3,753	41,670	,000	3,906	272
Std. Residual	-,959	10,648	,000	,998	272
Stud. Residual	-,967	10,714	,001	1,004	272
Deleted Residual	-3,816	42,188	,005	3,950	272
Stud. Deleted Residual	-,967	14,105	,016	1,161	272
Mahal. Distance	,000	4,473	,996	1,355	272
Cook's Distance	,000	,714	,006	,046	272
Centered Leverage Value	,000	,017	,004	,005	272

a. Dependent Variable: NO_OF_FAULTS

Variables Entered/Removed[a]

Model	Variables Entered	Variables Removed	Method
1	LOC[b]	.	Enter

a. Dependent Variable: NO_OF_FAULTS

b. All requested variables entered.

Model Summary[b]

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	,327[a]	,107	,104	3,886

a. Predictors: (Constant), LOC

b. Dependent Variable: NO_OF_FAULTS

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	487,761	1	487,761	32,296	,000[b]
	Residual	4077,754	270	15,103		
	Total	4565,515	271			

a. Dependent Variable: NO_OF_FAULTS

b. Predictors: (Constant), LOC

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	1,012	,246		4,122	,000
	LOC	,001	,000	,327	5,683	,000

a. Dependent Variable: NO_OF_FAULTS

Residuals Statistics[a]

	Minimum	Maximum	Mean	Std. Deviation	N
Predicted Value	1,01	20,34	1,40	1,342	272
Std. Predicted Value	-,292	14,114	,000	1,000	272
Standard Error of Predicted Value	,236	3,340	,265	,203	272
Adjusted Predicted Value	1,01	77,85	1,60	4,679	272
Residual	-20,339	37,687	,000	3,879	272
Std. Residual	-5,234	9,698	,000	,998	272
Stud. Residual	-10,239	10,084	-,017	1,148	272
Deleted Residual	-77,848	40,754	-,198	6,080	272
Stud. Deleted Residual	-13,067	12,749	-,014	1,347	272
Mahal. Distance	,000	199,200	,996	12,168	272
Cook's Distance	,000	148,215	,564	8,989	272
Centered Leverage Value	,000	,735	,004	,045	272

a. Dependent Variable: NO_OF_FAULTS

Variables Entered/Removed[a]

Model	Variables Entered	Variables Removed	Method
1	CC_AVG[b]	.	Enter

a. Dependent Variable: NO_OF_FAULTS

b. All requested variables entered.

Model Summary[b]

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	,245[a]	,060	,057	3,986

a. Predictors: (Constant), CC_AVG

b. Dependent Variable: NO_OF_FAULTS

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	274,940	1	274,940	17,302	,000[b]
	Residual	4290,575	270	15,891		
	Total	4565,515	271			

a. Dependent Variable: NO_OF_FAULTS

b. Predictors: (Constant), CC_AVG

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	,212	,375		,564	,573
	CC_AVG	,695	,167	,245	4,160	,000

a. Dependent Variable: NO_OF_FAULTS

Residuals Statistics[a]

	Minimum	Maximum	Mean	Std. Deviation	N
Predicted Value	,21	6,29	1,40	1,007	272
Std. Predicted Value	-1,184	4,850	,000	1,000	272
Standard Error of Predicted Value	,242	1,199	,315	,133	272
Adjusted Predicted Value	,20	6,85	1,41	1,036	272
Residual	-6,241	42,668	,000	3,979	272
Std. Residual	-1,566	10,704	,000	,998	272
Stud. Residual	-1,640	10,740	-,001	1,003	272
Deleted Residual	-6,849	42,961	-,006	4,016	272
Stud. Deleted Residual	-1,645	14,165	,015	1,163	272
Mahal. Distance	,000	23,526	,996	2,854	272
Cook's Distance	,000	,395	,005	,029	272
Centered Leverage Value	,000	,087	,004	,011	272

a. Dependent Variable: NO_OF_FAULTS

Variables Entered/Removed[a]

Model	Variables Entered	Variables Removed	Method
1	CC_MAX[b]	.	Enter

a. Dependent Variable: NO_OF_FAULTS

b. All requested variables entered.

Model Summary[b]

Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	,277[a]	,077	,073	3,951

a. Predictors: (Constant), CC_MAX

b. Dependent Variable: NO_OF_FAULTS

ANOVA[a]

Model		Sum of Squares	df	Mean Square	F	Sig.
1	Regression	350,025	1	350,025	22,419	,000[b]
	Residual	4215,490	270	15,613		
	Total	4565,515	271			

a. Dependent Variable: NO_OF_FAULTS

b. Predictors: (Constant), CC_MAX

Coefficients[a]

Model		Unstandardized Coefficients		Standardized Coefficients	t	Sig.
		B	Std. Error	Beta		
1	(Constant)	,806	,271		2,973	,003
	CC_MAX	,095	,020	,277	4,735	,000

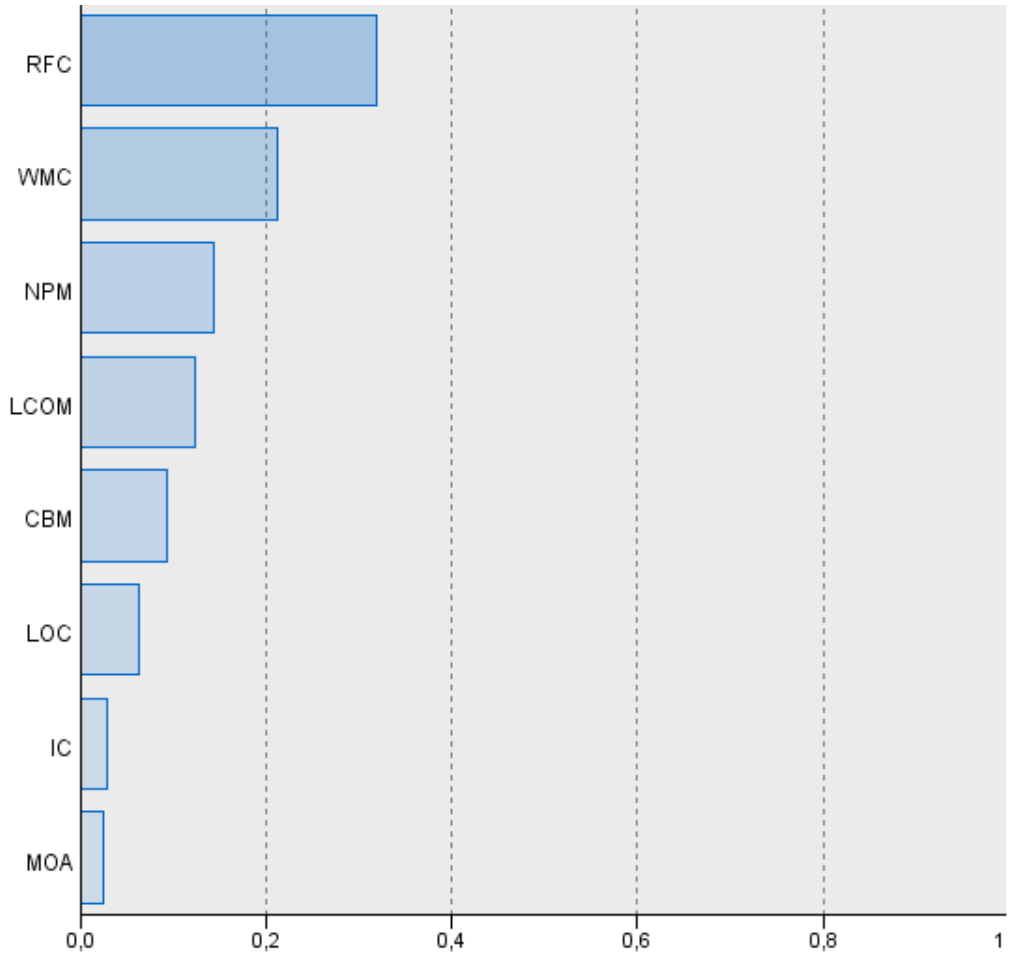
a. Dependent Variable: NO_OF_FAULTS

Residuals Statistics[a]

	Minimum	Maximum	Mean	Std. Deviation	N
Predicted Value	,81	16,27	1,40	1,136	272
Std. Predicted Value	-,527	13,079	,000	1,000	272
Standard Error of Predicted Value	,240	3,149	,281	,190	272
Adjusted Predicted Value	,80	44,56	1,50	2,702	272
Residual	-16,269	41,823	,000	3,944	272
Std. Residual	-4,117	10,585	,000	,998	272
Stud. Residual	-6,814	10,652	-,009	1,058	272
Deleted Residual	-44,564	42,359	-,099	4,725	272
Stud. Deleted Residual	-7,475	13,964	,004	1,218	272
Mahal. Distance	,001	171,070	,996	10,440	272
Cook's Distance	,000	40,381	,155	2,449	272
Centered Leverage Value	,000	,631	,004	,039	272

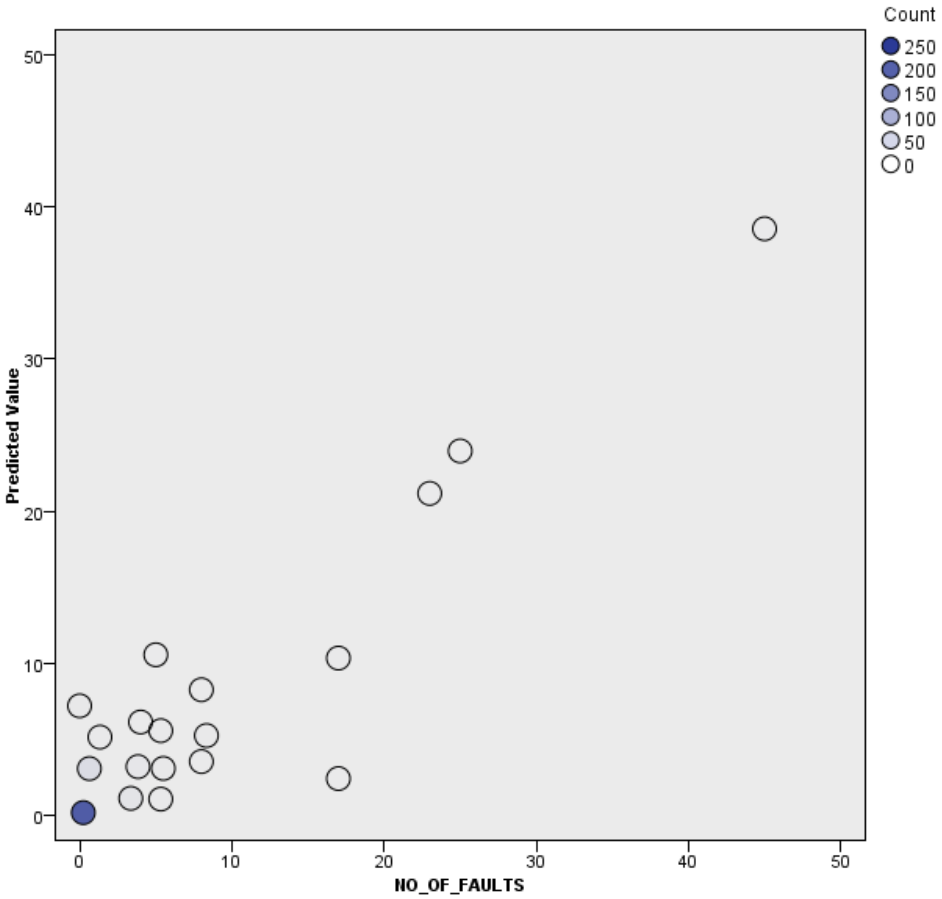
a. Dependent Variable: NO_OF_FAULTS

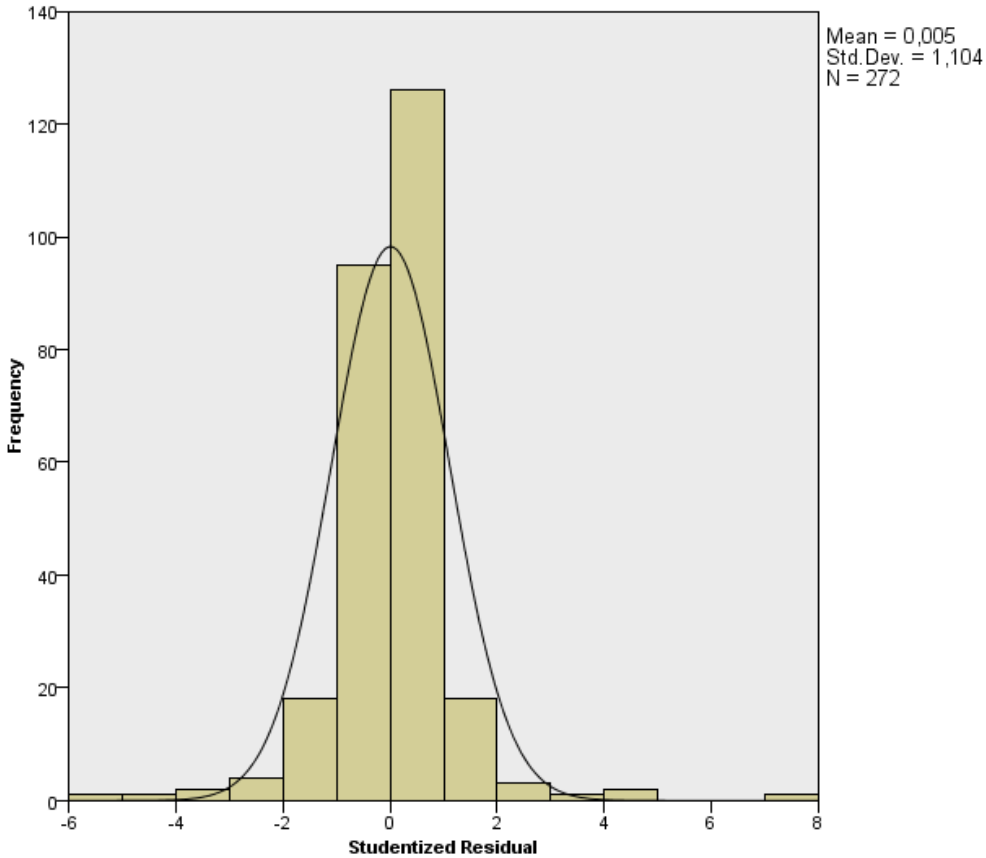
B.3 Πολλαπλή Γραμμική Παλινδρόμηση



		Step									
		1	2	3	4	5	6	7	8	9	10
Information Criterion		564,533	511,758	486,667	477,310	464,183	433,562	416,887	416,724	404,847	401,571
Effect	LCOM	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	CE		✓	✓	✓	✓	✓	✓			
	CBM			✓	✓	✓	✓	✓	✓	✓	✓
	IC				✓	✓	✓	✓	✓	✓	✓
	WMC					✓	✓	✓	✓	✓	✓
	RFC						✓	✓	✓	✓	✓
	NPM							✓	✓	✓	✓
	LOC									✓	✓
	MOA										✓

The model building method is Forward Stepwise using the Information Criterion.
A checkmark means the effect is in the model at this step.
The maximum number of effects 8 was reached by the Stepwise Selection method.





The histogram of Studentized residuals compares the distribution of the residuals to a normal distribution.The smooth line represents the normal distribution.The closer the frequencies of the residuals are to this line, the closer the distribution of the residuals is to the normal distribution.

Outliers

Target: NO_OF_FAULTS

Record ID	NO_OF_FAULTS	Cook's Distance
228	0	4,112
20	45	4,065
132	0	33,268
104	0	0,992
31	0	0,529
73	0	0,494
13	18	0,318
210	16	0,231
61	17	0,157
85	5	0,135
16	0	0,106
220	0	0,049
214	0	0,028
260	23	0,024
205	9	0,022
107	8	0,019
191	0	0,019
27	1	0,016

Records with large Cook's distance values are highly influential in the model computations. Such records may distort the model accuracy.

Model Summary[b]

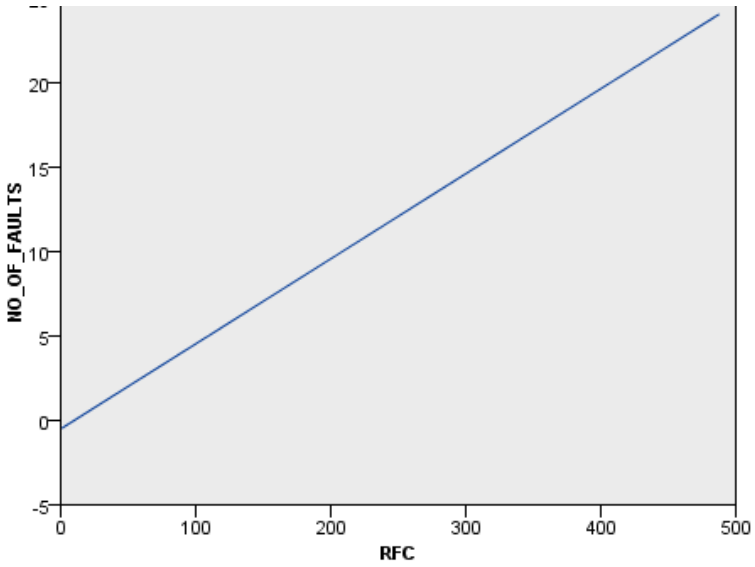
Model	R	R Square	Adjusted R Square	Std. Error of the Estimate
1	,870[a]	,757	,749	2,056

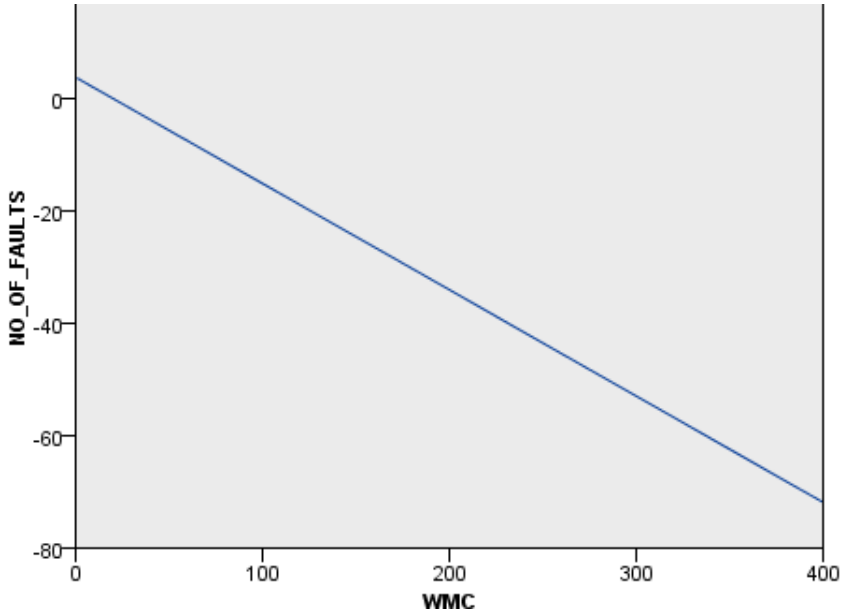
a. Predictors: (Constant), LOC, IC, LCOM, MOA, CBM, NPM, RFC, WMC

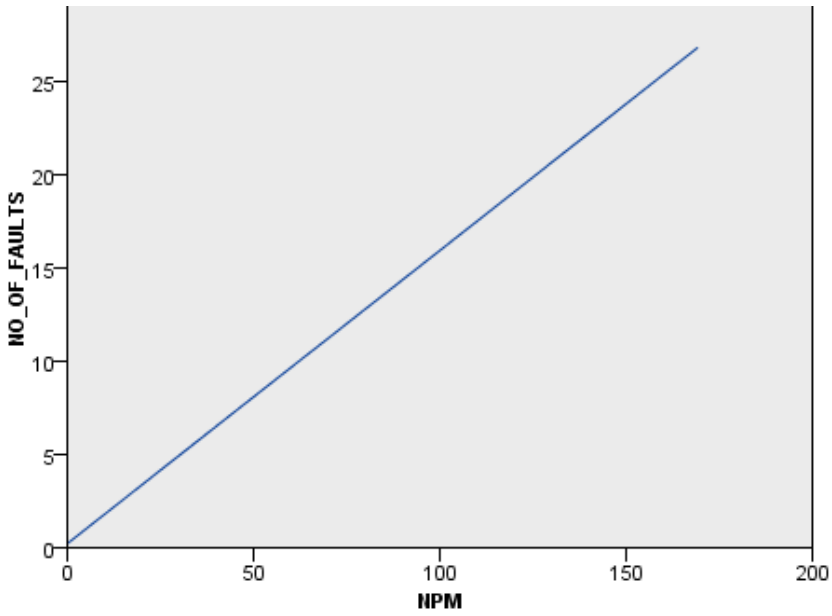
b. Dependent Variable: NO_OF_FAULTS

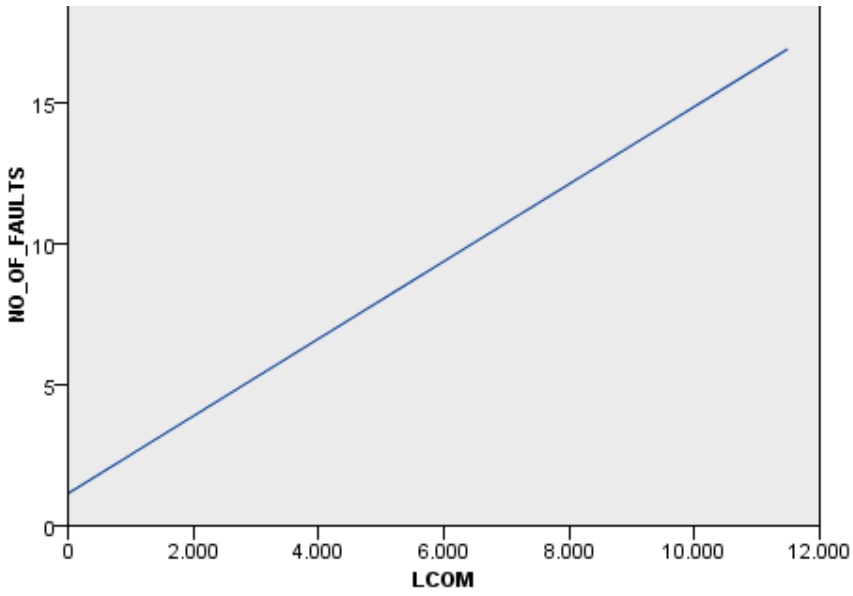
Model Term	Coefficient ▼	Std.Error	t	Sig.	95% Confidence Interval		Importance
					Lower	Upper	
Intercept	-0,551	0,192	-2,871	,004	-0,929	-0,173	
RFC	0,050	0,006	8,519	,000	0,039	0,062	0,318
WMC	-0,189	0,027	-6,944	,000	-0,243	-0,136	0,211
NPM	0,157	0,028	5,707	,000	0,103	0,211	0,143
LCOM	0,001	0,000	5,274	,000	0,001	0,002	0,122
CBM	0,390	0,085	4,579	,000	0,223	0,558	0,092
LOC	0,002	0,000	3,778	,000	0,001	0,002	0,063
IC	-0,652	0,260	-2,509	,013	-1,165	-0,140	0,028
MOA	0,204	0,089	2,300	,022	0,029	0,378	0,023

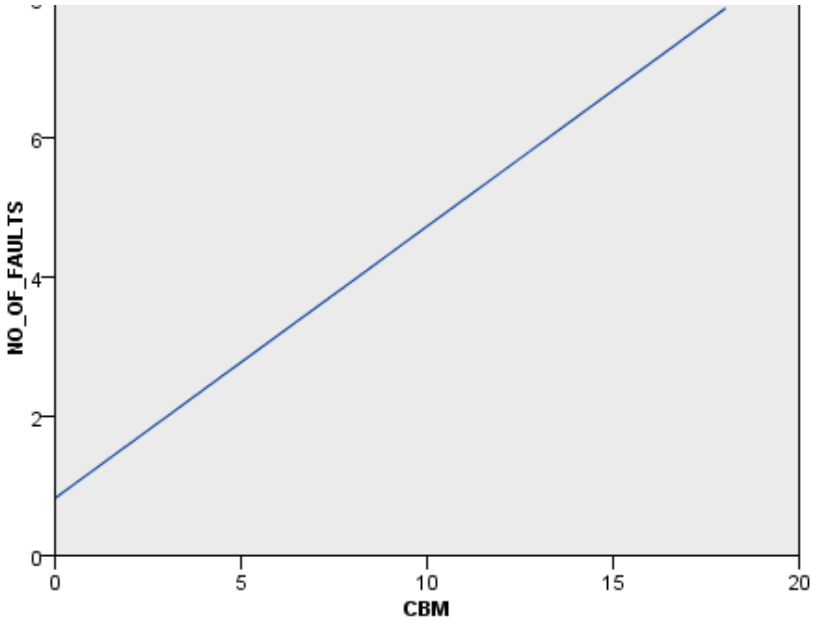
Source	Sum of Squares	df	Mean Square	F	Sig.	Importance
Corrected Model ▼	3.454,015	8	431,752	102,160	,000	
RFC	306,731	1	306,731	72,578	,000	0,318
WMC	203,772	1	203,772	48,216	,000	0,211
NPM	137,647	1	137,647	32,570	,000	0,143
LCOM	117,555	1	117,555	27,816	,000	0,122
CBM	88,611	1	88,611	20,967	,000	0,092
LOC	60,333	1	60,333	14,276	,000	0,063
IC	26,605	1	26,605	6,295	,013	0,028
MOA	22,348	1	22,348	5,288	,022	0,023
Residual	1.111,500	263	4,226			
Corrected Total	4.565,515	271				

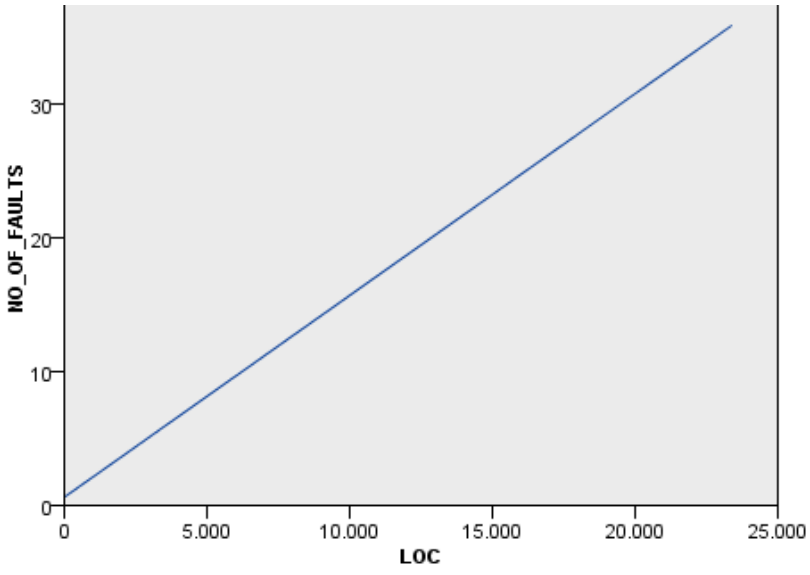

NO_OF_FAULTS
20
15
10
5
0
-5
0
100
200
300
400
500
RFC

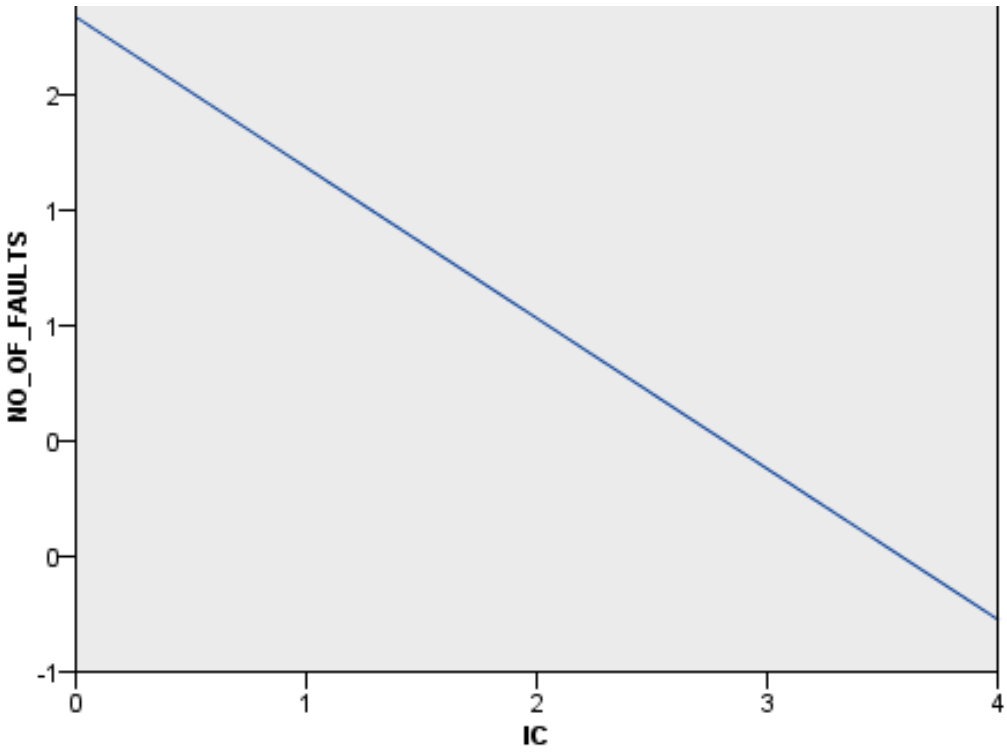

NO_OF_FAULTS
0
-20
-40
-60
-80
0
100
200
300
400
WMC

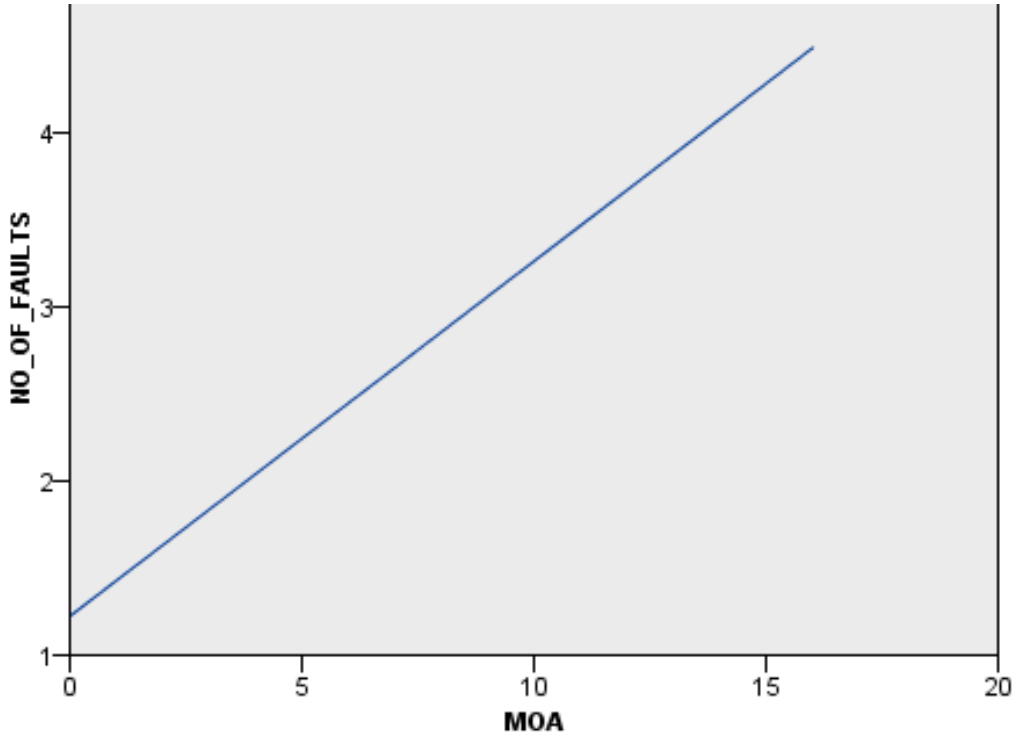

NO_OF_FAULTS
25
20
15
10
5
0
0
50
100
150
200
NPM


NO_OF_FAULTS
15
10
5
0
0
2.000
4.000
6.000
8.000
10.000
12.000
LCOM


NO_OF_FAULTS
6
4
2
0
0
5
10
15
20
CBM


NO_OF_FAULTS
30
20
10
0
0
5.000
10.000
15.000
20.000
25.000
LOC


NO_OF_FAULTS
2
1
1
0
0
-1
0
1
2
3
4
IC


NO_OF_FAULTS
4
3
2
1
0
5
10
15
20
MOA

Β.4 Απλή Δυαδική Λογιστική Παλινδρόμηση

Iteration History[a,b,c,d]

Iteration		-2 Log likelihood	Coefficients	
			Constant	WMC
Step 1	1	342,251	-,761	,007
	2	342,127	-,806	,008
	3	342,126	-,807	,008
	4	342,126	-,807	,008

a. Method: Enter

b. Constant is included in the model.

c. Initial -2 Log Likelihood: 345,332

d. Estimation terminated at iteration number 4 because parameter estimates changed by less than ,001.

Omnibus Tests of Model Coefficients

		Chi-square	df	Sig.
Step 1	Step	3,206	1	,073
	Block	3,206	1	,073
	Model	3,206	1	,073

Model Summary

Step	-2 Log likelihood	Cox & Snell R Square	Nagelkerke R Square
1	342,126[a]	,012	,016

a. Estimation terminated at iteration number 4 because parameter estimates changed by less than ,001.

Classification Table[a]

Observed			Predicted		
			FAULTY		Percentage Correct
			0	1	
Step 1	FAULTY	0	180	2	98,9
		1	87	3	3,3
	Overall Percentage				67,3

a. The cut value is ,500

Variables in the Equation

		B	S.E.	Wald	df	Sig.	Exp(B)
Step 1[a]	WMC	,008	,005	2,314	1	,128	1,008
	Constant	-,807	,145	31,124	1	,000	,446

a. Variable(s) entered on step 1: WMC.

Correlation Matrix

		Constant	WMC
Step 1	Constant	1,000	-,444
	WMC	-,444	1,000

Iteration History[a,b,c,d]

Iteration		-2 Log likelihood	Coefficients	
			Constant	DIT
Step 1	1	311,428	-1,583	,320
	2	310,665	-1,781	,358
	3	310,664	-1,789	,359
	4	310,664	-1,789	,359

a. Method: Enter

b. Constant is included in the model.

c. Initial -2 Log Likelihood: 345,332

d. Estimation terminated at iteration number 4 because parameter estimates changed by less than ,001.

Omnibus Tests of Model Coefficients

		Chi-square	df	Sig.
Step 1	Step	34,668	1	,000
	Block	34,668	1	,000
	Model	34,668	1	,000

Model Summary

Step	-2 Log likelihood	Cox & Snell R Square	Nagelkerke R Square
1	310,664[a]	,120	,166

a. Estimation terminated at iteration number 4 because parameter estimates changed by less than ,001.

Classification Table[a]

Observed		Predicted		
		FAULTY		Percentage Correct
		0	1	
Step 1	FAULTY 0	158	24	86,8
	1	45	45	50,0
	Overall Percentage			74,6

a. The cut value is ,500

Variables in the Equation

		B	S.E.	Wald	df	Sig.	Exp(B)
Step 1[a]	DIT	,359	,064	31,768	1	,000	1,432
	Constant	-1,789	,242	54,448	1	,000	,167

a. Variable(s) entered on step 1: DIT.

Correlation Matrix

		Constant	DIT
Step 1	Constant	1,000	-,822
	DIT	-,822	1,000

Iteration History[a,b,c,d]

Iteration		-2 Log likelihood	Coefficients	
			Constant	NOC
Step 1	1	344,851	-,664	-,030
	2	344,752	-,688	-,042
	3	344,751	-,688	-,044
	4	344,751	-,688	-,044

a. Method: Enter

b. Constant is included in the model.

c. Initial -2 Log Likelihood: 345,332

d. Estimation terminated at iteration number 4 because parameter estimates changed by less than ,001.

Omnibus Tests of Model Coefficients

		Chi-square	df	Sig.
Step 1	Step	,581	1	,446
	Block	,581	1	,446
	Model	,581	1	,446

Model Summary

Step	-2 Log likelihood	Cox & Snell R Square	Nagelkerke R Square
1	344,751[a]	,002	,003

a. Estimation terminated at iteration number 4 because parameter estimates changed by less than ,001.

Classification Table[a]

			Predicted		
			FAULTY		Percentage Correct
	Observed		0	1	
Step 1	FAULTY	0	182	0	100,0
		1	90	0	,0
	Overall Percentage				66,9

a. The cut value is ,500

Variables in the Equation

		B	S.E.	Wald	df	Sig.	Exp(B)
Step 1[a]	NOC	-,044	,065	,457	1	,499	,957
	Constant	-,688	,130	27,803	1	,000	,503

a. Variable(s) entered on step 1: NOC.

Correlation Matrix

		Constant	NOC
Step 1	Constant	1,000	-,152
	NOC	-,152	1,000

Iteration History[a,b,c,d]

Iteration		-2 Log likelihood	Coefficients	
			Constant	CBO
Step 1	1	334,900	-,952	,023
	2	334,495	-1,049	,028
	3	334,493	-1,054	,029
	4	334,493	-1,054	,029

a. Method: Enter

b. Constant is included in the model.

c. Initial -2 Log Likelihood: 345,332

d. Estimation terminated at iteration number 4 because parameter estimates changed by less than ,001.

Omnibus Tests of Model Coefficients

		Chi-square	df	Sig.
Step 1	Step	10,839	1	,001
	Block	10,839	1	,001
	Model	10,839	1	,001

Model Summary

Step	-2 Log likelihood	Cox & Snell R Square	Nagelkerke R Square
1	334,493[a]	,039	,054

a. Estimation terminated at iteration number 4 because parameter estimates changed by less than ,001.

Classification Table[a]

Observed		Predicted		
		FAULTY		Percentage Correct
		0	1	
Step 1	FAULTY 0	174	8	95,6
	1	82	8	8,9
	Overall Percentage			66,9

a. The cut value is ,500

Variables in the Equation

		B	S.E.	Wald	df	Sig.	Exp(B)
Step 1[a]	CBO	,029	,010	8,111	1	,004	1,029
	Constant	-1,054	,177	35,489	1	,000	,349

a. Variable(s) entered on step 1: CBO.

Correlation Matrix

		Constant	CBO
Step 1	Constant	1,000	-,668
	CBO	-,668	1,000

Iteration History[a,b,c,d]

Iteration		-2 Log likelihood	Coefficients	
			Constant	RFC
Step 1	1	314,736	-1,101	,011
	2	308,973	-1,404	,019
	3	308,660	-1,481	,021
	4	308,659	-1,485	,021
	5	308,659	-1,485	,021

a. Method: Enter

b. Constant is included in the model.

c. Initial -2 Log Likelihood: 345,332

d. Estimation terminated at iteration number 5 because parameter estimates changed by less than ,001.

Omnibus Tests of Model Coefficients

		Chi-square	df	Sig.
Step 1	Step	36,673	1	,000
	Block	36,673	1	,000
	Model	36,673	1	,000

Model Summary

Step	-2 Log likelihood	Cox & Snell R Square	Nagelkerke R Square
1	308,659[a]	,126	,175

a. Estimation terminated at iteration number 5 because parameter estimates changed by less than ,001.

Classification Table[a]

			Predicted		
			FAULTY		Percentage Correct
	Observed		0	1	
Step 1	FAULTY	0	170	12	93,4
		1	63	27	30,0
	Overall Percentage				72,4

a. The cut value is ,500

Variables in the Equation

		B	S.E.	Wald	df	Sig.	Exp(B)
Step 1[a]	RFC	,021	,004	23,028	1	,000	1,022
	Constant	-1,485	,206	51,871	1	,000	,226

a. Variable(s) entered on step 1: RFC.

Correlation Matrix

		Constant	RFC
Step 1	Constant	1,000	-,741
	RFC	-,741	1,000

Iteration History[a,b,c,d]

Iteration		-2 Log likelihood	Coefficients	
			Constant	LCOM
Step 1	1	341,619	-,717	,000
	2	341,488	-,753	,000
	3	341,487	-,753	,000

a. Method: Enter

b. Constant is included in the model.

c. Initial -2 Log Likelihood: 345,332

d. Estimation terminated at iteration number 3 because parameter estimates changed by less than ,001.

Omnibus Tests of Model Coefficients

		Chi-square	df	Sig.
Step 1	Step	3,845	1	,050
	Block	3,845	1	,050
	Model	3,845	1	,050

Model Summary

Step	-2 Log likelihood	Cox & Snell R Square	Nagelkerke R Square
1	341,487[a]	,014	,020

a. Estimation terminated at iteration number 3 because parameter estimates changed by less than ,001.

Classification Table[a]

Observed			Predicted		
			FAULTY		Percentage Correct
			0	1	
Step 1	FAULTY	0	180	2	98,9
		1	87	3	3,3
	Overall Percentage				67,3

a. The cut value is ,500

Variables in the Equation

		B	S.E.	Wald	df	Sig.	Exp(B)
Step 1[a]	LCOM	,000	,000	2,698	1	,100	1,000
	Constant	-,753	,132	32,493	1	,000	,471

a. Variable(s) entered on step 1: LCOM.

Correlation Matrix

		Constant	LCOM
Step 1	Constant	1,000	-,186
	LCOM	-,186	1,000

Iteration History[a,b,c,d]

Iteration		-2 Log likelihood	Coefficients	
			Constant	LCOM3
Step 1	1	337,228	-,112	-,536
	2	336,864	-,054	-,642
	3	336,863	-,051	-,646
	4	336,863	-,051	-,647

a. Method: Enter

b. Constant is included in the model.

c. Initial -2 Log Likelihood: 345,332

d. Estimation terminated at iteration number 4 because parameter estimates changed by less than ,001.

Omnibus Tests of Model Coefficients

		Chi-square	df	Sig.
Step 1	Step	8,469	1	,004
	Block	8,469	1	,004
	Model	8,469	1	,004

Model Summary

Step	-2 Log likelihood	Cox & Snell R Square	Nagelkerke R Square
1	336,863[a]	,031	,043

a. Estimation terminated at iteration number 4 because parameter estimates changed by less than ,001.

Classification Table[a]

Observed			Predicted		
			FAULTY		Percentage Correct
			0	1	
Step 1	FAULTY	0	182	0	100,0
		1	90	0	,0
	Overall Percentage				66,9

a. The cut value is ,500

Variables in the Equation

		B	S.E.	Wald	df	Sig.	Exp(B)
Step 1[a]	LCOM3	-,647	,230	7,899	1	,005	,524
	Constant	-,051	,258	,039	1	,844	,950

a. Variable(s) entered on step 1: LCOM3.

Correlation Matrix

		Constant	LCOM3
Step 1	Constant	1,000	-,862
	LCOM3	-,862	1,000

Iteration History[a,b,c,d]

Iteration		-2 Log likelihood	Coefficients	
			Constant	IC
Step 1	1	323,972	-1,059	,584
	2	323,563	-1,164	,650
	3	323,562	-1,167	,652
	4	323,562	-1,167	,652

a. Method: Enter

b. Constant is included in the model.

c. Initial -2 Log Likelihood: 345,332

d. Estimation terminated at iteration number 4 because parameter estimates changed by less than ,001.

Omnibus Tests of Model Coefficients

		Chi-square	df	Sig.
Step 1	Step	21,770	1	,000
	Block	21,770	1	,000
	Model	21,770	1	,000

Model Summary

Step	-2 Log likelihood	Cox & Snell R Square	Nagelkerke R Square
1	323,562[a]	,077	,107

a. Estimation terminated at iteration number 4 because parameter estimates changed by less than ,001.

Classification Table[a]

Observed			Predicted		
			FAULTY		Percentage Correct
			0	1	
Step 1	FAULTY	0	171	11	94,0
		1	64	26	28,9
	Overall Percentage				72,4

a. The cut value is ,500

Variables in the Equation

		B	S.E.	Wald	df	Sig.	Exp(B)
Step 1[a]	IC	,652	,148	19,458	1	,000	1,920
	Constant	-1,167	,172	46,045	1	,000	,311

a. Variable(s) entered on step 1: IC.

Correlation Matrix

		Constant	IC
Step 1	Constant	1,000	-,623
	IC	-,623	1,000

Iteration History[a,b,c,d]

Iteration		-2 Log likelihood	Coefficients	
			Constant	CBM
Step 1	1	313,217	-1,012	,231
	2	311,213	-1,154	,308
	3	311,161	-1,171	,323
	4	311,161	-1,171	,324

a. Method: Enter

b. Constant is included in the model.

c. Initial -2 Log Likelihood: 345,332

d. Estimation terminated at iteration number 4 because parameter estimates changed by less than ,001.

Omnibus Tests of Model Coefficients

		Chi-square	df	Sig.
Step 1	Step	34,171	1	,000
	Block	34,171	1	,000
	Model	34,171	1	,000

Model Summary

Step	-2 Log likelihood	Cox & Snell R Square	Nagelkerke R Square
1	311,161[a]	,118	,164

a. Estimation terminated at iteration number 4 because parameter estimates changed by less than ,001.

Classification Table[a]

			Predicted		
			FAULTY		Percentage Correct
	Observed		0	1	
Step 1	FAULTY	0	177	5	97,3
		1	63	27	30,0
	Overall Percentage				75,0

a. The cut value is ,500

Variables in the Equation

		B	S.E.	Wald	df	Sig.	Exp(B)
Step 1[a]	CBM	,324	,071	20,890	1	,000	1,382
	Constant	-1,171	,163	51,337	1	,000	,310

a. Variable(s) entered on step 1: CBM.

Correlation Matrix

		Constant	CBM
Step 1	Constant	1,000	-,537
	CBM	-,537	1,000

Iteration History[a,b,c,d]

Iteration		-2 Log likelihood	Coefficients	
			Constant	AMC
Step 1	1	341,612	-,867	,006
	2	341,446	-,939	,007
	3	341,445	-,942	,007
	4	341,445	-,942	,007

a. Method: Enter

b. Constant is included in the model.

c. Initial -2 Log Likelihood: 345,332

d. Estimation terminated at iteration number 4 because parameter estimates changed by less than ,001.

Omnibus Tests of Model Coefficients

		Chi-square	df	Sig.
Step 1	Step	3,887	1	,049
	Block	3,887	1	,049
	Model	3,887	1	,049

Model Summary

Step	-2 Log likelihood	Cox & Snell R Square	Nagelkerke R Square
1	341,445[a]	,014	,020

a. Estimation terminated at iteration number 4 because parameter estimates changed by less than ,001.

Classification Table[a]

Observed			Predicted		
			FAULTY		Percentage Correct
			0	1	
Step 1	FAULTY	0	180	2	98,9
		1	89	1	1,1
	Overall Percentage				66,5

a. The cut value is ,500

Variables in the Equation

		B	S.E.	Wald	df	Sig.	Exp(B)
Step 1[a]	AMC	,007	,004	2,979	1	,084	1,007
	Constant	-,942	,188	24,967	1	,000	,390

a. Variable(s) entered on step 1: AMC.

Correlation Matrix

		Constant	AMC
Step 1	Constant	1,000	-,725
	AMC	-,725	1,000

Iteration History[a,b,c,d]

Iteration		-2 Log likelihood	Coefficients	
			Constant	CA
Step 1	1	339,588	-,818	,020
	2	339,462	-,868	,022
	3	339,462	-,868	,022

a. Method: Enter

b. Constant is included in the model.

c. Initial -2 Log Likelihood: 345,332

d. Estimation terminated at iteration number 3 because parameter estimates changed by less than ,001.

Omnibus Tests of Model Coefficients

		Chi-square	df	Sig.
Step 1	Step	5,870	1	,015
	Block	5,870	1	,015
	Model	5,870	1	,015

Model Summary

Step	-2 Log likelihood	Cox & Snell R Square	Nagelkerke R Square
1	339,462[a]	,021	,030

a. Estimation terminated at iteration number 3 because parameter estimates changed by less than ,001.

Classification Table[a]

Observed			Predicted		
			FAULTY		Percentage Correct
			0	1	
Step 1	FAULTY	0	178	4	97,8
		1	84	6	6,7
	Overall Percentage				67,6

a. The cut value is ,500

Variables in the Equation

		B	S.E.	Wald	df	Sig.	Exp(B)
Step 1[a]	CA	,022	,010	4,930	1	,026	1,022
	Constant	-,868	,149	34,046	1	,000	,420

a. Variable(s) entered on step 1: CA.

Correlation Matrix

		Constant	CA
Step 1	Constant	1,000	-,482
	CA	-,482	1,000

Iteration History[a,b,c,d]

Iteration		-2 Log likelihood	Coefficients	
			Constant	CE
Step 1	1	317,097	-1,217	,087
	2	315,587	-1,421	,112
	3	315,570	-1,442	,116
	4	315,570	-1,442	,116

a. Method: Enter

b. Constant is included in the model.

c. Initial -2 Log Likelihood: 345,332

d. Estimation terminated at iteration number 4 because parameter estimates changed by less than ,001.

Omnibus Tests of Model Coefficients

		Chi-square	df	Sig.
Step 1	Step	29,762	1	,000
	Block	29,762	1	,000
	Model	29,762	1	,000

Model Summary

Step	-2 Log likelihood	Cox & Snell R Square	Nagelkerke R Square
1	315,570[a]	,104	,144

a. Estimation terminated at iteration number 4 because parameter estimates changed by less than ,001.

Classification Table[a]

Observed			Predicted		
			FAULTY		Percentage Correct
			0	1	
Step 1	FAULTY	0	173	9	95,1
		1	71	19	21,1
	Overall Percentage				70,6

a. The cut value is ,500

Variables in the Equation

		B	S.E.	Wald	df	Sig.	Exp(B)
Step 1[a]	CE	,116	,025	20,819	1	,000	1,123
	Constant	-1,442	,207	48,387	1	,000	,236

a. Variable(s) entered on step 1: CE.

Correlation Matrix

		Constant	CE
Step 1	Constant	1,000	-,752
	CE	-,752	1,000

Iteration History[a,b,c,d]

Iteration		-2 Log likelihood	Coefficients	
			Constant	NPM
Step 1	1	337,042	-,838	,022
	2	336,582	-,914	,029
	3	336,577	-,919	,029
	4	336,577	-,919	,029

a. Method: Enter

b. Constant is included in the model.

c. Initial -2 Log Likelihood: 345,332

d. Estimation terminated at iteration number 4 because parameter estimates changed by less than ,001.

Omnibus Tests of Model Coefficients

		Chi-square	df	Sig.
Step 1	Step	8,755	1	,003
	Block	8,755	1	,003
	Model	8,755	1	,003

Model Summary

Step	-2 Log likelihood	Cox & Snell R Square	Nagelkerke R Square
1	336,577[a]	,032	,044

a. Estimation terminated at iteration number 4 because parameter estimates changed by less than ,001.

Classification Table[a]

Observed		Predicted		
		FAULTY		Percentage Correct
		0	1	
Step 1	FAULTY 0	178	4	97,8
	1	84	6	6,7
	Overall Percentage			67,6

a. The cut value is ,500

Variables in the Equation

		B	S.E.	Wald	df	Sig.	Exp(B)
Step 1[a]	NPM	,029	,012	6,018	1	,014	1,030
	Constant	-,919	,154	35,488	1	,000	,399

a. Variable(s) entered on step 1: NPM.

Correlation Matrix

		Constant	NPM
Step 1	Constant	1,000	-,528
	NPM	-,528	1,000

Iteration History[a,b,c,d]

Iteration		-2 Log likelihood	Coefficients	
			Constant	DAM
Step 1	1	306,110	-1,496	1,533
	2	303,687	-1,850	1,903
	3	303,657	-1,896	1,951
	4	303,657	-1,896	1,952

a. Method: Enter

b. Constant is included in the model.

c. Initial -2 Log Likelihood: 345,332

d. Estimation terminated at iteration number 4 because parameter estimates changed by less than ,001.

Omnibus Tests of Model Coefficients

		Chi-square	df	Sig.
Step 1	Step	41,675	1	,000
	Block	41,675	1	,000
	Model	41,675	1	,000

Model Summary

Step	-2 Log likelihood	Cox & Snell R Square	Nagelkerke R Square
1	303,657[a]	,142	,198

a. Estimation terminated at iteration number 4 because parameter estimates changed by less than ,001.

Classification Table[a]

Observed			Predicted		
			FAULTY		Percentage Correct
			0	1	
Step 1	FAULTY	0	119	63	65,4
		1	35	55	61,1
	Overall Percentage				64,0

a. The cut value is ,500

Variables in the Equation

		B	S.E.	Wald	df	Sig.	Exp(B)
Step 1[a]	DAM	1,952	,330	35,011	1	,000	7,040
	Constant	-1,896	,267	50,257	1	,000	,150

a. Variable(s) entered on step 1: DAM.

Correlation Matrix

		Constant	DAM
Step 1	Constant	1,000	-,854
	DAM	-,854	1,000

Iteration History[a,b,c,d]

Iteration		-2 Log likelihood	Coefficients	
			Constant	MOA
Step 1	1	321,447	-,955	,310
	2	319,979	-1,083	,410
	3	319,965	-1,092	,421
	4	319,965	-1,092	,421

a. Method: Enter

b. Constant is included in the model.

c. Initial -2 Log Likelihood: 345,332

d. Estimation terminated at iteration number 4 because parameter estimates changed by less than ,001.

Omnibus Tests of Model Coefficients

		Chi-square	df	Sig.
Step 1	Step	25,367	1	,000
	Block	25,367	1	,000
	Model	25,367	1	,000

Model Summary

Step	-2 Log likelihood	Cox & Snell R Square	Nagelkerke R Square
1	319,965[a]	,089	,124

a. Estimation terminated at iteration number 4 because parameter estimates changed by less than ,001.

Classification Table[a]

Observed		Predicted		
		FAULTY		Percentage Correct
		0	1	
Step 1	FAULTY 0	173	9	95,1
	1	66	24	26,7
	Overall Percentage			72,4

a. The cut value is ,500

Variables in the Equation

		B	S.E.	Wald	df	Sig.	Exp(B)
Step 1[a]	MOA	,421	,098	18,530	1	,000	1,523
	Constant	-1,092	,159	47,061	1	,000	,335

a. Variable(s) entered on step 1: MOA.

Correlation Matrix

		Constant	MOA
Step 1	Constant	1,000	-,525
	MOA	-,525	1,000

Iteration History[a,b,c,d]

Iteration		-2 Log likelihood	Coefficients	
			Constant	MFA
Step 1	1	340,155	-1,011	,601
	2	339,956	-1,108	,698
	3	339,956	-1,111	,700
	4	339,956	-1,111	,700

a. Method: Enter

b. Constant is included in the model.

c. Initial -2 Log Likelihood: 345,332

d. Estimation terminated at iteration number 4 because parameter estimates changed by less than ,001.

Omnibus Tests of Model Coefficients

		Chi-square	df	Sig.
Step 1	Step	5,377	1	,020
	Block	5,377	1	,020
	Model	5,377	1	,020

Model Summary

Step	-2 Log likelihood	Cox & Snell R Square	Nagelkerke R Square
1	339,956[a]	,020	,027

a. Estimation terminated at iteration number 4 because parameter estimates changed by less than ,001.

Classification Table[a]

			Predicted		
			FAULTY		Percentage Correct
	Observed		0	1	
Step 1	FAULTY	0	182	0	100,0
		1	90	0	,0
	Overall Percentage				66,9

a. The cut value is ,500

Variables in the Equation

		B	S.E.	Wald	df	Sig.	Exp(B)
Step 1[a]	MFA	,700	,307	5,208	1	,022	2,014
	Constant	-1,111	,227	24,018	1	,000	,329

a. Variable(s) entered on step 1: MFA.

Correlation Matrix

		Constant	MFA
Step 1	Constant	1,000	-,819
	MFA	-,819	1,000

Iteration History[a,b,c,d]

Iteration		-2 Log likelihood	Coefficients	
			Constant	CAM
Step 1	1	325,369	,266	-2,013
	2	323,955	,444	-2,599
	3	323,943	,462	-2,659
	4	323,943	,462	-2,660

a. Method: Enter

b. Constant is included in the model.

c. Initial -2 Log Likelihood: 345,332

d. Estimation terminated at iteration number 4 because parameter estimates changed by less than ,001.

Omnibus Tests of Model Coefficients

		Chi-square	df	Sig.
Step 1	Step	21,389	1	,000
	Block	21,389	1	,000
	Model	21,389	1	,000

Model Summary

Step	-2 Log likelihood	Cox & Snell R Square	Nagelkerke R Square
1	323,943[a]	,076	,105

a. Estimation terminated at iteration number 4 because parameter estimates changed by less than ,001.

Classification Table[a]

Observed		Predicted		
		FAULTY		Percentage Correct
		0	1	
Step 1	FAULTY 0	170	12	93,4
	1	77	13	14,4
	Overall Percentage			67,3

a. The cut value is ,500

Variables in the Equation

		B	S.E.	Wald	df	Sig.	Exp(B)
Step 1[a]	CAM	-2,660	,627	18,009	1	,000	,070
	Constant	,462	,290	2,548	1	,110	1,588

a. Variable(s) entered on step 1: CAM.

Correlation Matrix

		Constant	CAM
Step 1	Constant	1,000	-,887
	CAM	-,887	1,000

Iteration History[a,b,c,d]

Iteration		-2 Log likelihood	Coefficients	
			Constant	LOC
Step 1	1	344,080	-,716	,000
	2	344,020	-,748	,000
	3	344,020	-,749	,000

a. Method: Enter

b. Constant is included in the model.

c. Initial -2 Log Likelihood: 345,332

d. Estimation terminated at iteration number 3 because parameter estimates changed by less than ,001.

Omnibus Tests of Model Coefficients

		Chi-square	df	Sig.
Step 1	Step	1,312	1	,252
	Block	1,312	1	,252
	Model	1,312	1	,252

Model Summary

Step	-2 Log likelihood	Cox & Snell R Square	Nagelkerke R Square
1	344,020[a]	,005	,007

a. Estimation terminated at iteration number 3 because parameter estimates changed by less than ,001.

Classification Table[a]

Observed			Predicted		
			FAULTY		Percentage Correct
			0	1	
Step 1	FAULTY	0	181	1	99,5
		1	90	0	,0
	Overall Percentage				66,5

a. The cut value is ,500

Variables in the Equation

		B	S.E.	Wald	df	Sig.	Exp(B)
Step 1[a]	LOC	,000	,000	1,085	1	,298	1,000
	Constant	-,749	,136	30,493	1	,000	,473

a. Variable(s) entered on step 1: LOC.

Correlation Matrix

		Constant	LOC
Step 1	Constant	1,000	-,303
	LOC	-,303	1,000

Iteration History[a,b,c,d]

Iteration		-2 Log likelihood	Coefficients	
			Constant	CC_AVG
Step 1	1	323,081	-1,328	,379
	2	322,460	-1,498	,445
	3	322,459	-1,505	,449
	4	322,459	-1,505	,449

a. Method: Enter

b. Constant is included in the model.

c. Initial -2 Log Likelihood: 345,332

d. Estimation terminated at iteration number 4 because parameter estimates changed by less than ,001.

Omnibus Tests of Model Coefficients

		Chi-square	df	Sig.
Step 1	Step	22,873	1	,000
	Block	22,873	1	,000
	Model	22,873	1	,000

Model Summary

Step	-2 Log likelihood	Cox & Snell R Square	Nagelkerke R Square
1	322,459[a]	,081	,112

a. Estimation terminated at iteration number 4 because parameter estimates changed by less than ,001.

Classification Table[a]

Observed		Predicted		
		FAULTY		Percentage Correct
		0	1	
Step 1	FAULTY 0	168	14	92,3
	1	75	15	16,7
	Overall Percentage			67,3

a. The cut value is ,500

Variables in the Equation

		B	S.E.	Wald	df	Sig.	Exp(B)
Step 1[a]	CC_AVG	,449	,104	18,570	1	,000	1,567
	Constant	-1,505	,229	43,055	1	,000	,222

a. Variable(s) entered on step 1: CC_AVG.

Correlation Matrix

		Constant	CC_AVG
Step 1	Constant	1,000	-,809
	CC_AVG	-,809	1,000

Iteration History[a,b,c,d]

Iteration		-2 Log likelihood	Coefficients	
			Constant	CC_MAX
Step 1	1	337,298	-,845	,027
	2	335,875	-,980	,044
	3	335,839	-1,000	,048
	4	335,839	-1,000	,048

a. Method: Enter

b. Constant is included in the model.

c. Initial -2 Log Likelihood: 345,332

d. Estimation terminated at iteration number 4 because parameter estimates changed by less than ,001.

Omnibus Tests of Model Coefficients

		Chi-square	df	Sig.
Step 1	Step	9,493	1	,002
	Block	9,493	1	,002
	Model	9,493	1	,002

Model Summary

Step	-2 Log likelihood	Cox & Snell R Square	Nagelkerke R Square
1	335,839[a]	,034	,048

a. Estimation terminated at iteration number 4 because parameter estimates changed by less than ,001.

Classification Table[a]

Observed			Predicted		
			FAULTY		Percentage Correct
			0	1	
Step 1	FAULTY	0	177	5	97,3
		1	83	7	7,8
	Overall Percentage				67,6

a. The cut value is ,500

Variables in the Equation

		B	S.E.	Wald	df	Sig.	Exp(B)
Step 1[a]	CC_MAX	,048	,018	6,755	1	,009	1,049
	Constant	-1,000	,172	33,954	1	,000	,368

a. Variable(s) entered on step 1: CC_MAX.

Correlation Matrix

		Constant	CC_MAX
Step 1	Constant	1,000	-,645
	CC_MAX	-,645	1,000

Β.5 Πολλαπλή Δυαδική Λογιστική Παλινδρόμηση

Classification Table[a,b]

Observed			Predicted		
			FAULTY		Percentage Correct
			0	1	
Step 0	FAULTY	0	182	0	100,0
		1	90	0	,0
	Overall Percentage				66,9

a. Constant is included in the model.

b. The cut value is ,500

Variables in the Equation

		B	S.E.	Wald	df	Sig.	Exp(B)
Step 0	Constant	-,704	,129	29,863	1	,000	,495

Variables not in the Equation

			Score	df	Sig.
Step 0	Variables	DIT	35,535	1	,000
		RFC	29,572	1	,000
		LOC	1,390	1	,238
		CC_MAX	7,843	1	,005
		MOA	24,107	1	,000
		WMC	3,283	1	,070
		NOC	,499	1	,480
		CBO	10,833	1	,001
		LCOM	3,969	1	,046
		LCOM3	8,142	1	,004
		IC	22,495	1	,000
		CBM	32,796	1	,000
		AMC	3,897	1	,048
		CA	6,135	1	,013
		CE	28,808	1	,000
		NPM	8,540	1	,003
		DAM	39,226	1	,000
		MFA	5,277	1	,022
		CAM	19,579	1	,000
		CC_AVG	23,145	1	,000
	Overall Statistics		117,311	20	,000

Omnibus Tests of Model Coefficients

		Chi-square	df	Sig.
Step 1	Step	41,675	1	,000
	Block	41,675	1	,000
	Model	41,675	1	,000
Step 2	Step	21,539	1	,000
	Block	63,214	2	,000
	Model	63,214	2	,000
Step 3	Step	18,227	1	,000
	Block	81,441	3	,000
	Model	81,441	3	,000
Step 4	Step	5,205	1	,023
	Block	86,646	4	,000
	Model	86,646	4	,000
Step 5	Step	6,226	1	,013
	Block	92,872	5	,000
	Model	92,872	5	,000
Step 6	Step	13,560	1	,000
	Block	106,432	6	,000
	Model	106,432	6	,000
Step 7[a]	Step	-,608	1	,435
	Block	105,823	5	,000
	Model	105,823	5	,000
Step 8	Step	4,891	1	,027
	Block	110,714	6	,000
	Model	110,714	6	,000

a. A negative Chi-squares value indicates that the Chi-squares value has decreased from the previous step.

Model Summary

Step	-2 Log likelihood	Cox & Snell R Square	Nagelkerke R Square
1	303,657[a]	,142	,198
2	282,118[b]	,207	,288
3	263,891[b]	,259	,360
4	258,686[b]	,273	,379
5	252,460[c]	,289	,402
6	238,901[c]	,324	,450
7	239,509[c]	,322	,448
8	234,618[c]	,334	,465

a. Estimation terminated at iteration number 4 because parameter estimates changed by less than ,001.

b. Estimation terminated at iteration number 5 because parameter estimates changed by less than ,001.

c. Estimation terminated at iteration number 6 because parameter estimates changed by less than ,001.

Variables in the Equation

		B	S.E.	Wald	df	Sig.	
Step 1[a]	DAM	1,952	,330	35,011	1	,000	7,040
	Constant	-1,896	,267	50,257	1	,000	,150
Step 2[b]	DIT	,299	,066	20,812	1	,000	1,349
	DAM	1,722	,341	25,502	1	,000	5,596
	Constant	-2,660	,334	63,566	1	,000	,070
Step 3[c]	DIT	,281	,068	17,000	1	,000	1,324
	CE	,098	,027	12,841	1	,000	1,103
	DAM	1,638	,354	21,397	1	,000	5,144
	Constant	-3,191	,383	69,273	1	,000	,041
Step 4[d]	DIT	,264	,069	14,607	1	,000	1,303
	CE	,139	,034	16,964	1	,000	1,149
	DAM	1,716	,361	22,650	1	,000	5,565
	LOC	,000	,000	4,293	1	,038	1,000
	Constant	-3,343	,403	68,749	1	,000	,035
Step 5[e]	DIT	,281	,070	15,965	1	,000	1,324
	CE	,146	,037	15,521	1	,000	1,157
	DAM	1,701	,369	21,244	1	,000	5,479
	LOC	,000	,000	2,103	1	,147	1,000
	CC_MAX	,039	,018	4,757	1	,029	1,040
	Constant	-3,588	,433	68,717	1	,000	,028
Step 6[f]	DIT	,235	,073	10,411	1	,001	1,265
	RFC	,039	,011	11,887	1	,001	1,040
	CE	,039	,050	,605	1	,437	1,040
	DAM	1,157	,395	8,581	1	,003	3,182
	LOC	,002	,001	9,328	1	,002	,998
	CC_MAX	,078	,025	9,389	1	,002	1,081
	Constant	-3,501	,435	64,801	1	,000	,030
Step 7[f]	DIT	,235	,073	10,438	1	,001	1,264
	RFC	,044	,009	22,678	1	,000	1,045
	DAM	1,091	,383	8,103	1	,004	2,977
	LOC	,002	,001	9,562	1	,002	,998
	CC_MAX	,078	,026	9,112	1	,003	1,082
	Constant	-3,393	,407	69,647	1	,000	,034
Step 8[g]	DIT	,251	,073	11,688	1	,001	1,286
	RFC	,043	,010	20,246	1	,000	1,044
	DAM	,992	,392	6,418	1	,011	2,697
	MOA	,230	,102	5,078	1	,024	1,259
	LOC	,002	,001	10,810	1	,001	,998
	CC_MAX	,088	,027	10,316	1	,001	1,091
	Constant	-3,508	,419	70,251	1	,000	,030

a. Variable(s) entered on step 1: DAM.

b. Variable(s) entered on step 2: DIT.

c. Variable(s) entered on step 3: CE.

d. Variable(s) entered on step 4: LOC.

e. Variable(s) entered on step 5: CC_MAX.

f. Variable(s) entered on step 6: RFC.

g. Variable(s) entered on step 8: MOA.

Model if Term Removed

Variable		Model Log Likelihood	Change in -2 Log Likelihood	df	Sig. of the Change
Step 1	DAM	-172,666	41,675	1	,000
Step 2	DIT	-151,828	21,539	1	,000
	DAM	-155,332	28,546	1	,000
Step 3	DIT	-140,637	17,382	1	,000
	CE	-141,059	18,227	1	,000
	DAM	-143,769	23,647	1	,000
Step 4	DIT	-136,780	14,874	1	,000
	CE	-140,647	22,608	1	,000
	DAM	-141,993	25,300	1	,000
	LOC	-131,945	5,205	1	,023
Step 5	DIT	-134,430	16,400	1	,000
	CE	-136,998	21,536	1	,000
	DAM	-138,064	23,667	1	,000
	LOC	-130,908	9,355	1	,002
	CC_MAX	-129,343	6,226	1	,013
Step 6	DIT	-124,665	10,429	1	,001
	RFC	-126,230	13,560	1	,000
	CE	-119,754	,608	1	,435
	DAM	-123,917	8,934	1	,003
	LOC	-130,897	22,894	1	,000
	CC_MAX	-123,773	8,645	1	,003
Step 7	DIT	-124,971	10,434	1	,001
	RFC	-136,998	34,488	1	,000
	DAM	-123,932	8,355	1	,004
	LOC	-133,077	26,645	1	,000
	CC_MAX	-124,130	8,750	1	,003
Step 8	DIT	-123,164	11,711	1	,001
	RFC	-130,448	26,279	1	,000
	DAM	-120,590	6,562	1	,010
	MOA	-119,754	4,891	1	,027
	LOC	-130,845	27,073	1	,000
	CC_MAX	-121,993	9,369	1	,002

Classification Table[a]

Observed			Predicted		
			FAULTY		Percentage Correct
			0	1	
Step 1	FAULTY	0	119	63	65,4
		1	35	55	61,1
	Overall Percentage				64,0
Step 2	FAULTY	0	165	17	90,7
		1	48	42	46,7
	Overall Percentage				76,1
Step 3	FAULTY	0	162	20	89,0
		1	41	49	54,4
	Overall Percentage				77,6
Step 4	FAULTY	0	163	19	89,6
		1	39	51	56,7
	Overall Percentage				78,7
Step 5	FAULTY	0	161	21	88,5
		1	36	54	60,0
	Overall Percentage				79,0
Step 6	FAULTY	0	165	17	90,7
		1	35	55	61,1
	Overall Percentage				80,9
Step 7	FAULTY	0	162	20	89,0
		1	34	56	62,2
	Overall Percentage				80,1
Step 8	FAULTY	0	164	18	90,1
		1	34	56	62,2
	Overall Percentage				80,9

a. The cut value is ,500

Παράρτημα Γ
Αναλυτικά Αποτελέσματα Μηχανικής Μάθησης από WEKA

Γ.1 Δέντρο Απόφασης

WMC

```
Correctly Classified Instances         186               68.3824 %
Incorrectly Classified Instances        86               31.6176 %
Kappa statistic                          0.2524
Mean absolute error                      0.4011
Root mean squared error                  0.4555
Relative absolute error                 90.4902 %
Root relative squared error             96.8011 %
Total Number of Instances              272
```

	TP Rate	FP Rate	Precision	Recall	F-Measure	ROC Area	Class
	0.808	0.567	0.742	0.808	0.774	0.633	0
	0.433	0.192	0.527	0.433	0.476	0.633	1
Weighted Avg.	0.684	0.443	0.671	0.684	0.675	0.633	

```
   a   b   <-- classified as
 147  35 |   a = 0
  51  39 |   b = 1
```

DIT

```
Correctly Classified Instances         201               73.8971 %
Incorrectly Classified Instances        71               26.1029 %
Kappa statistic                          0.3809
Mean absolute error                      0.3745
Root mean squared error                  0.436
Relative absolute error                 84.4796 %
Root relative squared error             92.6547 %
Total Number of Instances              272

=== Detailed Accuracy By Class ===
               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.852     0.489      0.779     0.852     0.814      0.651    0
                 0.511     0.148      0.63      0.511     0.564      0.651    1
Weighted Avg.    0.739     0.376      0.73      0.739     0.731      0.651

=== Confusion Matrix ===
   a   b   <-- classified as
 155  27 |   a = 0
  44  46 |   b = 1
```

NOC

```
Correctly Classified Instances         182               66.9118 %
Incorrectly Classified Instances        90               33.0882 %
Kappa statistic                          0
Mean absolute error                      0.4428
Root mean squared error                  0.4705
Relative absolute error                 99.8954 %
Root relative squared error             99.9996 %
Total Number of Instances              272

=== Detailed Accuracy By Class ===
               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         1          0.669     1         0.802      0.496    0
                 0         0          0         0         0          0.496    1
Weighted Avg.    0.669     0.669      0.448     0.669     0.536      0.496

=== Confusion Matrix ===
   a   b   <-- classified as
 182   0 |   a = 0
  90   0 |   b = 1
```

CBO

```
Correctly Classified Instances         181               66.5441 %
Incorrectly Classified Instances        91               33.4559 %
Kappa statistic                         -0.0073
Mean absolute error                      0.4444
Root mean squared error                  0.4751
Relative absolute error                100.2492 %
Root relative squared error            100.9739 %
Total Number of Instances              272

=== Detailed Accuracy By Class ===
               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.995     1          0.668     0.995     0.799      0.491    0
                 0          0.005      0         0         0          0.491    1
Weighted Avg.    0.665     0.671      0.447     0.665     0.535      0.491

=== Confusion Matrix ===
   a   b   <-- classified as
 181   1 |   a = 0
  90   0 |   b = 1
```

RFC

```
Correctly Classified Instances         193               70.9559 %
Incorrectly Classified Instances        79               29.0441 %
Kappa statistic                          0.3773
Mean absolute error                      0.3483
Root mean squared error                  0.4349
Relative absolute error                 78.5778 %
Root relative squared error             92.4265 %
Total Number of Instances              272

=== Detailed Accuracy By Class ===
               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.731     0.333      0.816     0.731     0.771      0.723    0
                 0.667     0.269      0.55      0.667     0.603      0.723    1
Weighted Avg.    0.71      0.312      0.728     0.71      0.715      0.723

=== Confusion Matrix ===
   a   b   <-- classified as
 133  49 |   a = 0
  30  60 |   b = 1
```

LCOM

```
Correctly Classified Instances         200               73.5294 %
Incorrectly Classified Instances        72               26.4706 %
Kappa statistic                          0.3777
Mean absolute error                      0.3735
Root mean squared error                  0.437
Relative absolute error                 84.2687 %
Root relative squared error             92.8798 %
Total Number of Instances              272

=== Detailed Accuracy By Class ===
               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.841     0.478      0.781     0.841     0.81       0.657    0
                 0.522     0.159      0.618     0.522     0.566      0.657    1
Weighted Avg.    0.735     0.372      0.727     0.735     0.729      0.657

=== Confusion Matrix ===
   a   b   <-- classified as
 153  29 |   a = 0
  43  47 |   b = 1
```

LCOM3

```
Correctly Classified Instances         191               70.2206 %
Incorrectly Classified Instances        81               29.7794 %
Kappa statistic                          0.3781
Mean absolute error                      0.3622
Root mean squared error                  0.4346
Relative absolute error                 81.7065 %
Root relative squared error             92.3578 %
Total Number of Instances              272

=== Detailed Accuracy By Class ===
               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.698     0.289      0.83      0.698     0.758      0.675    0
                 0.711     0.302      0.538     0.711     0.612      0.675    1
Weighted Avg.    0.702     0.293      0.733     0.702     0.71       0.675

=== Confusion Matrix ===
   a   b   <-- classified as
 127  55 |   a = 0
  26  64 |   b = 1
```

IC

```
Correctly Classified Instances         197               72.4265 %
Incorrectly Classified Instances        75               27.5735 %
Kappa statistic                          0.2167
Mean absolute error                      0.3931
Root mean squared error                  0.444
Relative absolute error                 88.6908 %
Root relative squared error             94.3561 %
Total Number of Instances              272

=== Detailed Accuracy By Class ===
               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.995     0.822      0.71      0.995     0.828      0.56     0
                 0.178     0.005      0.941     0.178     0.299      0.56     1
Weighted Avg.    0.724     0.552      0.786     0.724     0.653      0.56

=== Confusion Matrix ===
   a   b   <-- classified as
 181   1 |   a = 0
  74  16 |   b = 1
```

CBM

```
Correctly Classified Instances         204               75      %
Incorrectly Classified Instances        68               25      %
Kappa statistic                          0.3558
Mean absolute error                      0.3712
Root mean squared error                  0.4352
Relative absolute error                 83.7491 %
Root relative squared error             92.4886 %
Total Number of Instances              272

=== Detailed Accuracy By Class ===
               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.934     0.622      0.752     0.934     0.833      0.611    0
                 0.378     0.066      0.739     0.378     0.5        0.611    1
Weighted Avg.    0.75      0.438      0.748     0.75      0.723      0.611

=== Confusion Matrix ===
   a   b   <-- classified as
 170  12 |   a = 0
  56  34 |   b = 1
```

AMC

```
Correctly Classified Instances         182               66.9118 %
Incorrectly Classified Instances        90               33.0882 %
Kappa statistic                          0
Mean absolute error                      0.4428
Root mean squared error                  0.4705
Relative absolute error                 99.8954 %
Root relative squared error             99.9996 %
Total Number of Instances              272

=== Detailed Accuracy By Class ===
               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         1          0.669     1         0.802      0.496    0
                 0         0          0         0         0          0.496    1
Weighted Avg.    0.669     0.669      0.448     0.669     0.536      0.496

=== Confusion Matrix ===
   a   b   <-- classified as
 182   0 |   a = 0
  90   0 |   b = 1
```

CA

```
Correctly Classified Instances         182               66.9118 %
Incorrectly Classified Instances        90               33.0882 %
Kappa statistic                          0
Mean absolute error                      0.4428
Root mean squared error                  0.4705
Relative absolute error                 99.8954 %
Root relative squared error             99.9996 %
Total Number of Instances              272

=== Detailed Accuracy By Class ===
               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         1          0.669     1         0.802      0.496    0
                 0         0          0         0         0          0.496    1
Weighted Avg.    0.669     0.669      0.448     0.669     0.536      0.496

=== Confusion Matrix ===
   a   b   <-- classified as
 182   0 |   a = 0
  90   0 |   b = 1
```

CE

```
Correctly Classified Instances         185               68.0147 %
Incorrectly Classified Instances        87               31.9853 %
Kappa statistic                          0.0717
Mean absolute error                      0.429
Root mean squared error                  0.4687
Relative absolute error                 96.7771 %
Root relative squared error             99.611  %
Total Number of Instances              272

=== Detailed Accuracy By Class ===
               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.978     0.922      0.682     0.978     0.804      0.519    0
                 0.078     0.022      0.636     0.078     0.139      0.519    1
Weighted Avg.    0.68      0.624      0.667     0.68      0.584      0.519

=== Confusion Matrix ===
   a   b   <-- classified as
 178   4 |   a = 0
  83   7 |   b = 1
```

NPM

```
Correctly Classified Instances         175               64.3382 %
Incorrectly Classified Instances        97               35.6618 %
Kappa statistic                          0.0722
Mean absolute error                      0.435
Root mean squared error                  0.472
Relative absolute error                 98.1331 %
Root relative squared error            100.3098 %
Total Number of Instances              272

=== Detailed Accuracy By Class ===
               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.863     0.8        0.686     0.863     0.764      0.541    0
                 0.2       0.137      0.419     0.2       0.271      0.541    1
Weighted Avg.    0.643     0.581      0.597     0.643     0.601      0.541

=== Confusion Matrix ===
   a   b   <-- classified as
 157  25 |   a = 0
  72  18 |   b = 1
```

DAM

```
Correctly Classified Instances         175               64.3382 %
Incorrectly Classified Instances        97               35.6618 %
Kappa statistic                          0.0475
Mean absolute error                      0.4385
Root mean squared error                  0.4712
Relative absolute error                 98.918  %
Root relative squared error            100.1413 %
Total Number of Instances              272

=== Detailed Accuracy By Class ===
               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.885     0.844      0.679     0.885     0.768      0.532    0
                 0.156     0.115      0.4       0.156     0.224      0.532    1
Weighted Avg.    0.643     0.603      0.587     0.643     0.588      0.532

=== Confusion Matrix ===
   a   b   <-- classified as
 161  21 |   a = 0
  76  14 |   b = 1
```

MOA

```
Correctly Classified Instances         184               67.6471 %
Incorrectly Classified Instances        88               32.3529 %
Kappa statistic                          0.1821
Mean absolute error                      0.3803
Root mean squared error                  0.4419
Relative absolute error                 85.7882 %
Root relative squared error             93.9048 %
Total Number of Instances              272

=== Detailed Accuracy By Class ===
               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.863     0.7        0.714     0.863     0.781      0.653    0
                 0.3       0.137      0.519     0.3       0.38       0.653    1
Weighted Avg.    0.676     0.514      0.649     0.676     0.648      0.653

=== Confusion Matrix ===
   a   b   <-- classified as
 157  25 |   a = 0
  63  27 |   b = 1
```

MFA

```
Correctly Classified Instances         201               73.8971 %
Incorrectly Classified Instances        71               26.1029 %
Kappa statistic                          0.312
Mean absolute error                      0.3715
Root mean squared error                  0.4371
Relative absolute error                 83.7987 %
Root relative squared error             92.8978 %
Total Number of Instances              272

=== Detailed Accuracy By Class ===
               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.945     0.678      0.738     0.945     0.829      0.63     0
                 0.322     0.055      0.744     0.322     0.45       0.63     1
Weighted Avg.    0.739     0.472      0.74      0.739     0.703      0.63

=== Confusion Matrix ===
   a   b   <-- classified as
 172  10 |   a = 0
  61  29 |   b = 1
```

CAM

```
Correctly Classified Instances         192               70.5882 %
Incorrectly Classified Instances        80               29.4118 %
Kappa statistic                          0.3282
Mean absolute error                      0.393
Root mean squared error                  0.4464
Relative absolute error                 88.6553 %
Root relative squared error             94.8722 %
Total Number of Instances              272

=== Detailed Accuracy By Class ===
               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.791     0.467      0.774     0.791     0.783      0.608    0
                 0.533     0.209      0.558     0.533     0.545      0.608    1
Weighted Avg.    0.706     0.381      0.703     0.706     0.704      0.608

=== Confusion Matrix ===
   a   b   <-- classified as
 144  38 |   a = 0
  42  48 |   b = 1
```

LOC

```
Correctly Classified Instances         199               73.1618 %
Incorrectly Classified Instances        73               26.8382 %
Kappa statistic                          0.3278
Mean absolute error                      0.3813
Root mean squared error                  0.4425
Relative absolute error                 86.0305 %
Root relative squared error             94.0454 %
Total Number of Instances              272

=== Detailed Accuracy By Class ===
               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.896     0.6        0.751     0.896     0.817      0.622    0
                 0.4       0.104      0.655     0.4       0.497      0.622    1
Weighted Avg.    0.732     0.436      0.719     0.732     0.711      0.622

=== Confusion Matrix ===
   a   b   <-- classified as
 163  19 |   a = 0
  54  36 |   b = 1
```

CC_AVG

```
Correctly Classified Instances         183               67.2794 %
Incorrectly Classified Instances        89               32.7206 %
Kappa statistic                          0.096
Mean absolute error                      0.429
Root mean squared error                  0.4661
Relative absolute error                 96.7852 %
Root relative squared error             99.063  %
Total Number of Instances              272

=== Detailed Accuracy By Class ===
               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.934     0.856      0.688     0.934     0.793      0.564    0
                 0.144     0.066      0.52      0.144     0.226      0.564    1
Weighted Avg.    0.673     0.594      0.633     0.673     0.605      0.564

=== Confusion Matrix ===
   a   b   <-- classified as
 170  12 |   a = 0
  77  13 |   b = 1
```

CC_MAX

```
Correctly Classified Instances         199               73.1618 %
Incorrectly Classified Instances        73               26.8382 %
Kappa statistic                          0.3746
Mean absolute error                      0.3783
Root mean squared error                  0.4393
Relative absolute error                 85.3488 %
Root relative squared error             93.357  %
Total Number of Instances              272

=== Detailed Accuracy By Class ===
               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.83      0.467      0.782     0.83      0.805      0.648    0
                 0.533     0.17       0.608     0.533     0.568      0.648    1
Weighted Avg.    0.732     0.369      0.725     0.732     0.727      0.648

=== Confusion Matrix ===
   a   b   <-- classified as
 151  31 |   a = 0
  42  48 |   b = 1
```

Όλες οι Μετρικές

```
Correctly Classified Instances         202               74.2647 %
Incorrectly Classified Instances        70               25.7353 %
Kappa statistic                          0.4155
Mean absolute error                      0.3224
Root mean squared error                  0.4376
Relative absolute error                 72.7281 %
Root relative squared error             93.0009 %
Total Number of Instances              272

=== Detailed Accuracy By Class ===
               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.813     0.4        0.804     0.813     0.809      0.741    0
                 0.6       0.187      0.614     0.6       0.607      0.741    1
Weighted Avg.    0.743     0.329      0.741     0.743     0.742      0.741

=== Confusion Matrix ===
   a   b   <-- classified as
 148  34 |   a = 0
  36  54 |   b = 1
```

Γ.2 Τεχνητό Νευρωνικό Δίκτυο

WMC

```
Correctly Classified Instances         187               68.75   %
Incorrectly Classified Instances        85               31.25   %
Kappa statistic                          0.1484
Mean absolute error                      0.402
Root mean squared error                  0.4513
Relative absolute error                 90.6824 %
Root relative squared error             95.9032 %
Total Number of Instances              272

=== Detailed Accuracy By Class ===
               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.934     0.811      0.7       0.934     0.8        0.705    0
                 0.189     0.066      0.586     0.189     0.286      0.705    1
Weighted Avg.    0.688     0.565      0.662     0.688     0.63       0.705

=== Confusion Matrix ===
   a   b   <-- classified as
 170  12 |   a = 0
  73  17 |   b = 1
```

DIT

```
Correctly Classified Instances         201               73.8971 %
Incorrectly Classified Instances        71               26.1029 %
Kappa statistic                          0.3809
Mean absolute error                      0.3719
Root mean squared error                  0.4351
Relative absolute error                 83.9054 %
Root relative squared error             92.4641 %
Total Number of Instances              272
=== Detailed Accuracy By Class ===

               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.852     0.489      0.779     0.852     0.814      0.693    0
                 0.511     0.148      0.63      0.511     0.564      0.693    1
Weighted Avg.    0.739     0.376      0.73      0.739     0.731      0.693

=== Confusion Matrix ===
   a   b   <-- classified as
 155  27 |   a = 0
  44  46 |   b = 1
```

NOC

```
Correctly Classified Instances         182               66.9118 %
Incorrectly Classified Instances        90               33.0882 %
Kappa statistic                          0
Mean absolute error                      0.4429
Root mean squared error                  0.4724
Relative absolute error                 99.9062 %
Root relative squared error            100.3853 %
Total Number of Instances              272

=== Detailed Accuracy By Class ===
               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 1         1          0.669     1         0.802      0.497    0
                 0         0          0         0         0          0.497    1
Weighted Avg.    0.669     0.669      0.448     0.669     0.536      0.497

=== Confusion Matrix ===
   a   b   <-- classified as
 182   0 |   a = 0
  90   0 |   b = 1
```

CBO

```
Correctly Classified Instances         179               65.8088 %
Incorrectly Classified Instances        93               34.1912 %
Kappa statistic                          0.0287
Mean absolute error                      0.4203
Root mean squared error                  0.4608
Relative absolute error                 94.8098 %
Root relative squared error             97.9228 %
Total Number of Instances              272

=== Detailed Accuracy By Class ===
               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.945     0.922      0.675     0.945     0.787      0.622    0
                 0.078     0.055      0.412     0.078     0.131      0.622    1
Weighted Avg.    0.658     0.635      0.588     0.658     0.57       0.622

=== Confusion Matrix ===
   a   b   <-- classified as
 172  10 |   a = 0
  83   7 |   b = 1
```

RFC

```
Correctly Classified Instances         203               74.6324 %
Incorrectly Classified Instances        69               25.3676 %
Kappa statistic                          0.4156
Mean absolute error                      0.324
Root mean squared error                  0.4037
Relative absolute error                 73.1031 %
Root relative squared error             85.8001 %
Total Number of Instances              272

=== Detailed Accuracy By Class ===
               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.83      0.422      0.799     0.83      0.814      0.812    0
                 0.578     0.17       0.627     0.578     0.601      0.812    1
Weighted Avg.    0.746     0.339      0.742     0.746     0.744      0.812

=== Confusion Matrix ===
   a   b   <-- classified as
 151  31 |   a = 0
  38  52 |   b = 1
```

LCOM

```
Correctly Classified Instances         181               66.5441 %
Incorrectly Classified Instances        91               33.4559 %
Kappa statistic                          0.0002
Mean absolute error                      0.4407
Root mean squared error                  0.4721
Relative absolute error                 99.4137 %
Root relative squared error            100.3243 %
Total Number of Instances              272

=== Detailed Accuracy By Class ===
               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.989     0.989      0.669     0.989     0.798      0.529    0
                 0.011     0.011      0.333     0.011     0.022      0.529    1
Weighted Avg.    0.665     0.665      0.558     0.665     0.541      0.529

=== Confusion Matrix ===
   a   b   <-- classified as
 180   2 |   a = 0
  89   1 |   b = 1
```

LCOM3

```
Correctly Classified Instances         199               73.1618 %
Incorrectly Classified Instances        73               26.8382 %
Kappa statistic                          0.3782
Mean absolute error                      0.3584
Root mean squared error                  0.4283
Relative absolute error                 80.8437 %
Root relative squared error             91.0219 %
Total Number of Instances              272

=== Detailed Accuracy By Class ===
               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.824     0.456      0.785     0.824     0.804      0.736    0
                 0.544     0.176      0.605     0.544     0.573      0.736    1
Weighted Avg.    0.732     0.363      0.726     0.732     0.728      0.736

=== Confusion Matrix ===
   a   b   <-- classified as
 150  32 |   a = 0
  41  49 |   b = 1
```

IC

```
Correctly Classified Instances         195               71.6912 %
Incorrectly Classified Instances        77               28.3088 %
Kappa statistic                          0.2285
Mean absolute error                      0.3872
Root mean squared error                  0.4436
Relative absolute error                 87.3611 %
Root relative squared error             94.2738 %
Total Number of Instances              272

=== Detailed Accuracy By Class ===
               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.956     0.767      0.716     0.956     0.819      0.602    0
                 0.233     0.044      0.724     0.233     0.353      0.602    1
Weighted Avg.    0.717     0.528      0.719     0.717     0.665      0.602

=== Confusion Matrix ===
   a   b   <-- classified as
 174   8 |   a = 0
  69  21 |   b = 1
```

CBM

```
Correctly Classified Instances         203               74.6324 %
Incorrectly Classified Instances        69               25.3676 %
Kappa statistic                          0.3269
Mean absolute error                      0.3607
Root mean squared error                  0.43
Relative absolute error                 81.3715 %
Root relative squared error             91.3855 %
Total Number of Instances              272

=== Detailed Accuracy By Class ===
               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.956     0.678      0.74      0.956     0.835      0.68     0
                 0.322     0.044      0.784     0.322     0.457      0.68     1
Weighted Avg.    0.746     0.468      0.755     0.746     0.71       0.68

=== Confusion Matrix ===
   a   b   <-- classified as
 174   8 |   a = 0
  61  29 |   b = 1
```

AMC

```
Correctly Classified Instances         176               64.7059 %
Incorrectly Classified Instances        96               35.2941 %
Kappa statistic                          0.0349
Mean absolute error                      0.4153
Root mean squared error                  0.4539
Relative absolute error                 93.6861 %
Root relative squared error             96.4585 %
Total Number of Instances              272

=== Detailed Accuracy By Class ===
               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.907     0.878      0.676     0.907     0.775      0.66     0
                 0.122     0.093      0.393     0.122     0.186      0.66     1
Weighted Avg.    0.647     0.618      0.582     0.647     0.58       0.66

=== Confusion Matrix ===
   a   b   <-- classified as
 165  17 |   a = 0
  79  11 |   b = 1
```

CA

```
Correctly Classified Instances         183               67.2794 %
Incorrectly Classified Instances        89               32.7206 %
Kappa statistic                          0.0294
Mean absolute error                      0.4367
Root mean squared error                  0.4692
Relative absolute error                 98.5169 %
Root relative squared error             99.7153 %
Total Number of Instances              272

=== Detailed Accuracy By Class ===
               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.989     0.967      0.674     0.989     0.802      0.528    0
                 0.033     0.011      0.6       0.033     0.063      0.528    1
Weighted Avg.    0.673     0.65       0.65      0.673     0.557      0.528

=== Confusion Matrix ===
   a   b   <-- classified as
 180   2 |   a = 0
  87   3 |   b = 1
```

CE

```
Correctly Classified Instances         182               66.9118 %
Incorrectly Classified Instances        90               33.0882 %
Kappa statistic                          0.1582
Mean absolute error                      0.3847
Root mean squared error                  0.441
Relative absolute error                 86.7847 %
Root relative squared error             93.7179 %
Total Number of Instances              272

=== Detailed Accuracy By Class ===
               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.863     0.722      0.707     0.863     0.777      0.713    0
                 0.278     0.137      0.5       0.278     0.357      0.713    1
Weighted Avg.    0.669     0.529      0.639     0.669     0.638      0.713

=== Confusion Matrix ===
   a   b   <-- classified as
 157  25 |   a = 0
  65  25 |   b = 1
```

NPM

```
Correctly Classified Instances         177               65.0735 %
Incorrectly Classified Instances        95               34.9265 %
Kappa statistic                          0.0148
Mean absolute error                      0.4252
Root mean squared error                  0.4645
Relative absolute error                 95.9207 %
Root relative squared error             98.7131 %
Total Number of Instances              272

=== Detailed Accuracy By Class ===
               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.934     0.922      0.672     0.934     0.782      0.619    0
                 0.078     0.066      0.368     0.078     0.128      0.619    1
Weighted Avg.    0.651     0.639      0.572     0.651     0.565      0.619

=== Confusion Matrix ===
   a   b   <-- classified as
 170  12 |   a = 0
  83   7 |   b = 1
```

DAM

```
Correctly Classified Instances         181               66.5441 %
Incorrectly Classified Instances        91               33.4559 %
Kappa statistic                          0.1724
Mean absolute error                      0.3732
Root mean squared error                  0.4366
Relative absolute error                 84.1881 %
Root relative squared error             92.786  %
Total Number of Instances              272

=== Detailed Accuracy By Class ===
               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.835     0.678      0.714     0.835     0.77       0.685    0
                 0.322     0.165      0.492     0.322     0.389      0.685    1
Weighted Avg.    0.665     0.508      0.64      0.665     0.644      0.685

=== Confusion Matrix ===
   a   b   <-- classified as
 152  30 |   a = 0
  61  29 |   b = 1
```

MOA

```
Correctly Classified Instances         196               72.0588 %
Incorrectly Classified Instances        76               27.9412 %
Kappa statistic                          0.3024
Mean absolute error                      0.3794
Root mean squared error                  0.4426
Relative absolute error                 85.5834 %
Root relative squared error             94.0604 %
Total Number of Instances              272

=== Detailed Accuracy By Class ===
               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.885     0.611      0.745     0.885     0.809      0.674    0
                 0.389     0.115      0.625     0.389     0.479      0.674    1
Weighted Avg.    0.721     0.447      0.706     0.721     0.7        0.674

=== Confusion Matrix ===
   a   b   <-- classified as
 161  21 |   a = 0
  55  35 |   b = 1
```

MFA

```
Correctly Classified Instances         180               66.1765 %
Incorrectly Classified Instances        92               33.8235 %
Kappa statistic                          0.0358
Mean absolute error                      0.4317
Root mean squared error                  0.4688
Relative absolute error                 97.4004 %
Root relative squared error             99.6334 %
Total Number of Instances              272

=== Detailed Accuracy By Class ===
               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.951     0.922      0.676     0.951     0.79       0.572    0
                 0.078     0.049      0.438     0.078     0.132      0.572    1
Weighted Avg.    0.662     0.633      0.597     0.662     0.572      0.572

=== Confusion Matrix ===
   a   b   <-- classified as
 173   9 |   a = 0
  83   7 |   b = 1
```

CAM

```
Correctly Classified Instances         192               70.5882 %
Incorrectly Classified Instances        80               29.4118 %
Kappa statistic                          0.2791
Mean absolute error                      0.3951
Root mean squared error                  0.4501
Relative absolute error                 89.1425 %
Root relative squared error             95.6628 %
Total Number of Instances              272

=== Detailed Accuracy By Class ===
               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.857     0.6        0.743     0.857     0.796      0.662    0
                 0.4       0.143      0.581     0.4       0.474      0.662    1
Weighted Avg.    0.706     0.449      0.689     0.706     0.689      0.662

=== Confusion Matrix ===
   a   b   <-- classified as
 156  26 |   a = 0
  54  36 |   b = 1
```

LOC

```
Correctly Classified Instances         184               67.6471 %
Incorrectly Classified Instances        88               32.3529 %
Kappa statistic                          0.0438
Mean absolute error                      0.435
Root mean squared error                  0.4673
Relative absolute error                 98.1243 %
Root relative squared error             99.3071 %
Total Number of Instances              272

=== Detailed Accuracy By Class ===
               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.989     0.956      0.677     0.989     0.804      0.544    0
                 0.044     0.011      0.667     0.044     0.083      0.544    1
Weighted Avg.    0.676     0.643      0.673     0.676     0.565      0.544

=== Confusion Matrix ===
   a   b   <-- classified as
 180   2 |   a = 0
  86   4 |   b = 1
```

CC_AVG

```
Correctly Classified Instances         183               67.2794 %
Incorrectly Classified Instances        89               32.7206 %
Kappa statistic                          0.1702
Mean absolute error                      0.3923
Root mean squared error                  0.4474
Relative absolute error                 88.4928 %
Root relative squared error             95.0861 %
Total Number of Instances              272

=== Detailed Accuracy By Class ===
               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.863     0.711      0.71      0.863     0.779      0.696    0
                 0.289     0.137      0.51      0.289     0.369      0.696    1
Weighted Avg.    0.673     0.521      0.644     0.673     0.643      0.696

=== Confusion Matrix ===
   a   b   <-- classified as
 157  25 |   a = 0
  64  26 |   b = 1
```

CC_MAX

```
Correctly Classified Instances         196               72.0588 %
Incorrectly Classified Instances        76               27.9412 %
Kappa statistic                          0.3067
Mean absolute error                      0.3679
Root mean squared error                  0.4363
Relative absolute error                 82.9897 %
Root relative squared error             92.7297 %
Total Number of Instances              272

=== Detailed Accuracy By Class ===
               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.879     0.6        0.748     0.879     0.808      0.719    0
                 0.4       0.121      0.621     0.4       0.486      0.719    1
Weighted Avg.    0.721     0.441      0.706     0.721     0.702      0.719

=== Confusion Matrix ===
   a   b   <-- classified as
 160  22 |   a = 0
  54  36 |   b = 1
```

Όλες οι Μετρικές Μαζί

```
Correctly Classified Instances         211               77.5735 %
Incorrectly Classified Instances        61               22.4265 %
Kappa statistic                          0.4863
Mean absolute error                      0.2567
Root mean squared error                  0.4236
Relative absolute error                 57.9115 %
Root relative squared error             90.0293 %
Total Number of Instances              272

=== Detailed Accuracy By Class ===
               TP Rate   FP Rate   Precision   Recall  F-Measure   ROC Area  Class
                 0.846     0.367      0.824     0.846     0.835      0.793    0
                 0.633     0.154      0.671     0.633     0.651      0.793    1
Weighted Avg.    0.776     0.296      0.773     0.776     0.774      0.793

=== Confusion Matrix ===
   a   b   <-- classified as
 154  28 |   a = 0
  33  57 |   b = 1
```

Παράρτημα Δ

Αρχικά Δεδομένα Μοντέλων

```
@RELATION jEdit3-21

@ATTRIBUTE CLASS_NAME STRING
@ATTRIBUTE FAULTY {0,1}
@ATTRIBUTE NO_OF_FAULTS NUMERIC
@ATTRIBUTE WMC NUMERIC
@ATTRIBUTE DIT NUMERIC
@ATTRIBUTE NOC NUMERIC
@ATTRIBUTE CBO NUMERIC
@ATTRIBUTE RFC NUMERIC
@ATTRIBUTE LCOM NUMERIC
@ATTRIBUTE LCOM3 NUMERIC
@ATTRIBUTE IC NUMERIC
@ATTRIBUTE CBM NUMERIC
@ATTRIBUTE AMC NUMERIC
@ATTRIBUTE CA NUMERIC
@ATTRIBUTE CE NUMERIC
@ATTRIBUTE NPM NUMERIC
@ATTRIBUTE DAM NUMERIC
@ATTRIBUTE MOA NUMERIC
@ATTRIBUTE MFA NUMERIC
@ATTRIBUTE CAM NUMERIC
@ATTRIBUTE LOC NUMERIC
@ATTRIBUTE CC_AVG NUMERIC
@ATTRIBUTE CC_MAX NUMERIC
```

@DATA
org.gjt.sp.jedit.Registers,0,0,14,1,0,9,45,61,0.769230769,0,0,29.14285714,3,6,11,1,1,0,0.294871795,425,2.4286,8
gnu.regexp.RETokenRepeated,0,0,7,2,0,4,21,0,0.541666667,1,3,41.28571429,1,3,0,1,1,0.538461538,0.333333333,300,4.7143,16
gnu.regexp.REMatchEnumeration,0,0,6,1,0,3,11,9,0.6,0,0,16.33333333,1,3,5,1,3,0,0.333333333,113,1.5,5
bsh.BSHType,0,0,6,2,0,16,14,3,0.666666667,0,0,13.83333333,8,8,5,1,0,0.782608696,0.444444444,92,0.8333,1
gnu.regexp.RETokenOneOf,0,0,5,2,0,5,22,0,0.125,1,3,39.8,1,4,0,1,0,0.7,0.35,206,4,11
bsh.BSHArrayDimensions,0,0,4,2,0,10,21,2,0.555555556,1,1,41.75,2,8,3,0.333333333,0,0.857142857,0.5,174,0.75,1
org.gjt.sp.jedit.options.StyleTableModel,0,0,8,2,0,6,19,0,0.285714286,1,2,18.5,2,4,6,1,0,0.695652174,0.40625,157,1.375,2
gnu.regexp.CharIndexedString,0,0,4,1,0,2,7,0,0.222222222,0,0,13.75,1,1,3,1,0,0,0.666666667,62,1.75,3
org.gjt.sp.jedit.gui.DockableWindowContainer,0,0,5,1,0,5,5,10,2,0,0,0,4,1,5,0,0,0,1,5,1,1
org.gjt.sp.jedit.textarea.TextRendererAWT,0,0,4,2,0,1,11,6,2,1,2,22.25,1,1,0,0,0,0.785714286,0.571428571,93,1.75,5
org.gjt.sp.jedit.EditPlugin,1,1,10,1,2,11,13,39,0.777777778,0,0,2.2,10,3,8,1,1,0,0.25,33,0.9,1
bsh.BSHLHSPrimarySuffix,0,0,5,2,0,14,27,10,1.071428571,0,0,34,2,12,1,0,0,0.818181818,0.72,182,0.8,1
org.gjt.sp.jedit.browser.VFSBrowser,1,18,54,5,0,41,175,1183,0.930333817,3,12,24.90740741,21,27,25,0.846153846,5,0.926241135,0.122895623,1425,2.1296,11
org.gjt.sp.jedit.options.DockingOptionPane,1,1,5,6,0,6,51,6,0.75,2,4,50,2,5,3,1,1,0.993993994,1,260,1.4,3
org.gjt.sp.jedit.options.WindowTableModel,0,0,11,2,0,7,30,25,0.833333333,1,2,16.72727273,3,6,9,1,0,0.615384615,0.386363636,198,1.8182,4
bsh.Interpreter,0,0,51,1,0,58,132,925,0.776470588,0,0,27.25490196,44,21,42,0.117647059,5,0,0.144,1458,1.6667,33
org.gjt.sp.jedit.gui.HelpViewer,1,5,18,6,0,21,124,117,0.77124183,3,6,40.83333333,7,20,6,1,0,0.975794251,0.1875,762,1.5556,7
org.gjt.sp.jedit.msg.CreateDockableWindow,0,0,7,2,0,6,15,9,0.722222222,1,1,7.714285714,2,4,7,1,1,0.454545455,0.357142857,64,0.8571,1
org.gjt.sp.jedit.options.EditingOptionPane,1,3,3,6,0,4,23,1,0.5,2,4,105.3333333,1,3,1,1,0,0.996987952,1,330,1.3333,3
org.gjt.sp.jedit.textarea.JEditTextArea,1,45,211,4,0,59,430,11469,0.883277962,3,9,34.87203791,40,39,169,0.906976744,10,0.755555556,0.084391534,7612,3.0521,25
org.gjt.sp.jedit.GUIUtilities,1,8,39,1,0,70,144,725,0.955263158,0,0,37,58,21,34,0.4,1,0,0.135964912,1492,2.6923,16
org.gjt.sp.jedit.EBComponent,0,0,1,1,0,31,1,0,2,0,0,0,31,1,1,0,0,0,1,1,1,1
org.gjt.sp.jedit.gui.KeyEventWorkaround,1,5,4,1,0,5,16,2,1.083333333,0,0,41.75,5,0,2,0.875,0,0,0.333333333,179,8.75,20
org.gjt.sp.jedit.gui.PastePrevious,1,1,12,7,0,7,57,52,0.709090909,0,0,21.83333333,1,7,10,1,2,0.983433735,0.277777778,279,1.3333,3
org.gjt.sp.jedit.EditPane,1,5,19,5,0,31,132,57,0.595238095,1,1,42.21052632,16,22,12,1,6,0.973174367,0.210526316,828,3.2105,21
org.gjt.sp.jedit.EBMessage,0,0,6,1,3,29,14,7,0.6,0,0,6.833333333,29,1,6,1,1,0,0.583333333,49,0.8333,1

org.gjt.sp.jedit.options.AbbrevsOptionPane,1,1,7,6,0,9,58,5,0.733333333,4,12,41.14285714,4,8,1,1,1,0.991017964,0.5,300,1.7143,5
org.gjt.sp.jedit.gui.CloseDialog,1,3,14,7,0,7,54,65,0.846153846,0,0,20.85714286,3,7,4,1,1,0.98048048,0.285714286,313,1.0714,3
org.gjt.sp.jedit.search.SearchDialog,1,7,30,7,0,22,137,315,0.830140485,2,2,51.53333333,7,20,4,0.888888889,5,0.958883994,0.233333333,1603,2.0333,11
gnu.regexp.RETokenLookAhead,0,0,2,2,0,6,8,0,0,1,1,32.5,1,6,0,0,1,0.875,0.571428571,69,3.5,7
gnu.regexp.RESyntax,0,0,9,1,0,4,18,0,1.015957447,0,0,33.88888889,4,1,8,0.106382979,16,0,0.40625,361,1,2
org.gjt.sp.jedit.search.AllBufferSet,0,0,5,2,0,7,22,4,0.625,0,0,17.2,1,6,3,1,0,0.636363636,0.6,93,1.4,4
bsh.ReturnControl,0,0,1,1,0,10,2,0,2,0,0,9,9,1,1,0,0,0,1,12,0,0
org.gjt.sp.jedit.gui.EditAbbrevDialog,0,0,5,6,0,6,39,0,0.8,0,0,40,3,5,2,1,1,0.99389313,0.32,210,1,2
org.gjt.sp.jedit.gui.SelectLineRange,0,0,5,7,0,8,52,2,0.5,0,0,70.8,1,8,4,1,1,0.99391172,0.466666667,364,1.6,3
bsh.BSHSwitchStatement,0,0,2,2,0,10,9,1,2,2,2,52,1,9,2,0,0,0.947368421,0.625,106,0.5,1
org.gjt.sp.jedit.io.FavoritesVFS,1,2,10,2,0,5,28,13,0.722222222,1,3,21.6,3,4,9,0.75,1,0.724137931,0.472222222,230,1.2,3
org.gjt.sp.jedit.options.GeneralOptionPane,0,0,3,6,0,4,34,1,0.5,2,4,118,1,3,1,1,0,0.996987952,1,370,2.6667,5
org.gjt.sp.jedit.gui.CurrentDirectoryMenu,1,3,2,8,0,10,30,1,2,2,3,86.5,2,9,2,0,0,0.998835856,0.75,175,7.5,15
org.gjt.sp.jedit.gui.JCheckBoxList,0,0,8,5,0,4,33,28,2,1,1,16.875,2,2,7,0,0,0.992974239,0.5,143,1.125,3
bsh.Node,0,0,7,1,0,36,7,21,2,0,0,0,36,0,7,0,0,0,0.523809524,7,1,1
org.gjt.sp.jedit.options.ColorOptionPane,1,2,8,6,0,5,31,20,0.761904762,4,12,14,2,4,1,0.666666667,1,0.991017964,0.5,123,0.875,2
org.gjt.sp.jedit.textarea.Gutter,1,8,30,4,0,10,73,337,0.927203065,3,18,19.33333333,7,7,25,1,3,0.957037037,0.127777778,628,1.9,20
bsh.BSHBinaryExpression,0,0,4,2,0,12,21,6,1,1,1,68.5,1,11,1,0,0,0.857142857,0.45,279,2.25,4
bsh.This,0,0,11,1,2,14,30,15,0.566666667,0,0,15.27272727,9,8,7,0,3,0,0.236363636,182,1.0909,3
bsh.BSHTryStatement,0,0,2,2,0,12,22,1,2,1,2,91.5,1,11,1,0,0,0.947368421,0.625,185,0.5,1
bsh.BSHWhileStatement,0,0,2,2,0,10,7,1,1,1,1,38,1,9,1,0,0,0.947368421,0.625,79,0.5,1
org.gjt.sp.jedit.proto.jeditresource.Handler,0,0,2,2,0,1,5,1,2,0,0,6,0,1,2,0,0,0.923076923,0.75,14,0.5,1
org.gjt.sp.jedit.options.BrowserOptionPane,1,2,3,6,0,3,17,1,0.5,2,4,84,1,2,3,1,0,0.996987952,1,263,2,5
org.gjt.sp.jedit.textarea.TextRenderer,0,0,13,1,2,6,32,76,0.972222222,0,0,32.53846154,5,2,8,0.333333333,0,0,0.318181818,439,2.4615,10
org.gjt.sp.jedit.gui.EnhancedMenuItem,1,2,5,6,0,5,33,2,0.458333333,2,11,36,2,3,3,1,1,0.996240602,0.4375,191,2.6,8
org.gjt.sp.jedit.browser.BrowserListener,0,0,2,1,0,4,2,1,2,0,0,0,3,2,2,0,0,0,1,2,1,1
bsh.Primitive,0,0,45,1,0,41,77,898,0.795454545,0,0,31.62222222,36,5,28,0.25,2,0,0.083333333,1472,2.0222,12
bsh.Reflect,0,0,36,1,0,23,109,630,2,0,0,42.25,12,14,14,0,0,0,0.155982906,1557,3.1111,30
gnu.regexp.CharUnit,0,0,1,1,0,1,2,0,2,0,0,3,1,0,0,0,0,0,1,6,0,0
org.gjt.sp.util.WorkThreadPool,1,1,22,1,0,12,51,129,0.873015873,0,0,25.27272727,11,6,9,0.8,5,0,0.140909091,593,2.6818,11
bsh.Capabilities,0,0,7,1,0,5,12,13,0.5,0,0,7.857142857,5,0,6,1,0,0,0.166666667,64,1.5714,4

org.gjt.sp.jedit.gui.FontSelectorDialog,0,0,17,7,0,5,80,98,0.875,1,3,37.29411765,3,4,4,1,0,0.97754491,0.1875,662,1.1765,5
bsh.ClassPathException,0,0,1,4,0,3,2,0,2,0,0,4,2,1,1,0,0,1,1,5,0,0
org.gjt.sp.jedit.gui.EnhancedDialog,0,0,3,6,16,19,13,3,1,0,0,18.33333333,19,3,3,1,1,0.996937213,0.5,59,0.6667,1
org.gjt.sp.jedit.browser.BrowserView,1,17,19,5,0,13,77,71,0.813131313,0,0,24.63157895,8,10,8,1,3,0.973174367,0.198830409,498,2.0526,9
org.gjt.sp.jedit.options.PrintOptionPane,1,1,3,6,0,4,24,1,0.5,1,3,80,1,3,1,1,1,0.996987952,1,253,0.6667,1
com.microstar.xml.XmlException,0,0,5,3,0,5,6,2,0.75,1,1,5.4,5,0,5,1,0,0.80952381,0.466666667,36,0.8,1
org.gjt.sp.jedit.msg.PropertiesChanged,0,0,1,3,0,8,2,0,2,0,0,4,6,2,1,0,0,1,1,5,0,0
org.gjt.sp.jedit.AbstractOptionPane,1,1,10,5,15,17,27,29,0.805555556,0,0,24.4,15,2,5,1,0,0.986404834,0.5,258,1.1,2
gnu.regexp.REToken,0,0,8,1,13,15,9,12,0.761904762,0,0,6.875,13,2,0,1,2,0,0.354166667,66,1.25,3
org.gjt.sp.jedit.syntax.XModeHandler,1,2,19,2,0,11,75,123,0.888888889,1,1,55.89473684,1,11,8,1,5,0.419354839,0.233918129,1106,5.5789,33
org.gjt.sp.jedit.Macros,1,6,24,1,0,29,89,228,0.900621118,0,0,20.95833333,11,24,16,1,0,0,0.155279503,534,1.875,6
org.gjt.sp.jedit.syntax.TokenMarker,1,1,10,1,0,10,46,9,0.951111111,0,0,140.6,4,8,7,0.48,1,0,0.314285714,1441,8.5,32
org.gjt.sp.jedit.options.ColorTableModel,0,0,8,2,0,5,20,0,0.285714286,1,2,20.625,2,3,6,1,0,0.695652174,0.40625,174,1.375,2
bsh.ParserTreeConstants,0,0,1,1,0,2,1,0,2,0,0,148,2,0,0,0,0,0,0,186,0,0
org.gjt.sp.jedit.search.SearchAndReplace,1,9,31,1,0,34,108,197,0.64,0,0,38.29032258,9,27,28,1,2,0,0.112903226,1228,3.129,23
bsh.ParserTokenManager,0,0,50,1,0,5,66,3,0.75170068,0,0,123.7,1,4,6,0.166666667,1,0,0.5,6259,8.68,163
gnu.regexp.CharIndexedStringBuffer,0,0,4,1,0,2,7,0,0,0,0,13.25,1,1,3,1,0,0,0.666666667,59,1.75,3
org.gjt.sp.jedit.EditBus,0,0,11,1,0,22,29,23,0.72,0,0,20.72727273,19,3,8,1,1,0,0.16,244,1.7273,5
org.gjt.sp.jedit.gui.VariableGridLayout,0,0,17,1,0,1,37,100,0.7,0,0,39.05882353,1,0,16,0.8,0,0,0.323529412,691,2.1176,13
org.gjt.sp.jedit.options.GutterOptionPane,0,0,3,6,0,4,27,1,0.5,2,4,76,1,3,3,1,1,0.996987952,1,238,1.6667,3
org.gjt.sp.jedit.options.StyleOptionPane,1,1,8,6,0,6,31,20,0.761904762,4,12,14,4,4,1,0.666666667,1,0.991017964,0.5,123,0.875,2
bsh.BSHArrayInitializer,0,0,4,2,0,10,23,6,2,1,1,42.75,3,7,2,0,0,0.857142857,0.583333333,175,0.75,1
org.gjt.sp.jedit.pluginmgr.PluginListHandler,1,1,12,2,0,7,38,38,0.863636364,1,3,36.5,1,7,8,1,4,0.541666667,0.319444444,470,3.3333,12
bsh.ASCII_UCodeESC_CharStream,0,0,27,1,0,3,40,25,0.529411765,0,0,45.66666667,3,0,21,0.705882353,0,0,0.296296296,1277,1.5926,7
org.gjt.sp.jedit.BeanShellAction,1,1,7,2,0,5,16,3,0.80952381,1,1,14.42857143,1,4,7,1,0,0.7,0.392857143,115,1.4286,3
org.gjt.sp.jedit.gui.BufferSwitcher,0,0,4,5,0,6,16,0,0.5,0,0,14,4,5,2,1,1,0.995856354,0.416666667,62,1,2
gnu.regexp.RETokenEnd,0,0,3,2,0,4,9,1,0.5,1,2,22.33333333,1,3,0,1,0,0.777777778,0.444444444,71,2.3333,6
org.gjt.sp.jedit.View,1,5,55,6,0,109,200,1197,0.892361111,1,1,24.83636364,99,30,40,0.875,8,0.922746781,0.0784689,1437,2.5091,13

org.gjt.sp.jedit.io.VFS,1,4,22,1,4,23,36,175,0.993197279,0,0,9.545454545,20,6,22,0.142857143,0,0,0
.392045455,239,1.3182,4
bsh.LHS,0,0,8,1,0,15,31,0,0.766233766,0,0,42.125,8,9,3,0,1,0,0.375,356,0.375,1
gnu.regexp.RETokenEndSub,0,0,3,2,0,4,5,3,2,1,2,5.666666667,1,3,0,0,0,0.777777778,0.466666667,20,0.
6667,1
gnu.regexp.RETokenRange,0,0,4,2,0,4,9,0,0.444444444,1,2,18.25,1,3,0,1,0,0.7,0.357142857,80,1.75,5
org.gjt.sp.jedit.options.ContextAddDialog,0,0,10,7,0,7,56,21,0.833333333,0,0,36.6,2,6,4,1,0,0.9864
04834,0.3,384,1.3,5
org.gjt.sp.jedit.gui.RecentFilesMenu,1,1,2,8,0,11,23,1,2,1,2,49,3,11,2,0,0,0.998835856,0.75,100,3.
5,7
bsh.commands.dir,0,0,6,1,0,2,40,13,0.6,0,0,61.16666667,0,2,5,0,0,0,0.28,374,3,11
gnu.regexp.REException,0,0,4,3,0,4,13,0,1.222222222,1,1,13,4,1,3,0.133333333,0,0.85,0.5,71,0.75,1
bsh.BSHLHSPrimaryExpression,0,0,2,2,0,11,9,1,2,0,0,31.5,3,8,1,0,0,0.947368421,0.625,65,0.5,1
bsh.BshClassManager,0,0,25,1,0,10,37,290,0.888888889,0,0,5.2,7,3,22,1,1,0,0.157407407,161,1.16,3
bsh.Name,0,0,19,1,0,19,71,91,0.708333333,0,0,65.05263158,6,15,9,0.625,1,0,0.274853801,1263,1.6316,
5
org.gjt.sp.jedit.gui.OpenWithEncodingMenu,1,1,1,8,0,4,15,0,2,0,0,82,2,4,1,0,0,1,1,83,0,0
gnu.regexp.REMatch,0,0,15,1,0,19,28,0,0.669642857,0,0,24.53333333,18,1,11,0.125,1,0,0.333333333,39
1,2.2667,6
gnu.regexp.REFilterReader,0,0,5,3,0,4,18,8,0.375,2,3,32.4,0,4,5,1,2,0.818181818,0.366666667,173,2.
6,7
org.gjt.sp.jedit.gui.CheckBoxListModel,0,0,11,2,0,2,23,25,0.833333333,2,2,13.90909091,1,1,7,0.6666
66667,0,0.64,0.303030303,167,1.5455,4
org.gjt.sp.jedit.search.SearchMatcher,1,2,2,1,0,4,2,1,2,0,0,0,4,0,2,0,0,0,0.666666667,2,1,1
bsh.ParseException,0,0,6,3,0,3,19,3,0.44,1,1,62.66666667,2,1,5,0.4,1,0.85,0.333333333,387,4.3333,1
4
bsh.Token,0,0,3,1,0,5,4,3,1.4375,0,0,4,5,0,2,0,2,0,0.5,23,1,2
bsh.JThis,0,0,87,2,0,8,100,3741,2,1,2,7.172413793,0,8,84,0,0,0.104166667,0.054675365,711,1.023,3
org.gjt.sp.jedit.msg.EditorExitRequested,0,0,2,3,0,5,4,1,2,0,0,4,1,4,2,0,0,0.857142857,0.75,10,0.5
,1
org.gjt.sp.jedit.browser.BrowserIORequest,1,5,7,2,0,7,27,0,0.616666667,0,0,69,1,7,3,0.6,2,0.454545
455,0.285714286,500,3.1429,5
org.gjt.sp.jedit.gui.DockableWindowManager,1,8,18,5,0,16,62,31,0.794117647,0,0,28.94444444,5,14,16
,0.583333333,5,0.976083707,0.317647059,551,1.9444,7
bsh.BSHArguments,0,0,2,2,0,10,6,1,2,0,0,14,5,5,1,0,0,0.947368421,0.625,30,0.5,1
org.gjt.sp.jedit.textarea.ScrollListener,0,0,2,1,0,3,2,1,2,0,0,0,3,1,2,0,0,0,1,2,1,1
org.gjt.sp.jedit.gui.AbbrevEditor,0,0,5,5,0,4,26,0,0.25,0,0,66.4,3,1,5,1,0,0.99391172,0.6,339,4.2,
12
org.gjt.sp.jedit.gui.EnhancedButton,0,0,3,6,0,3,13,3,2,2,2,14,1,2,3,0,0,0.997382199,0.5,45,0.6667,
1
org.gjt.sp.jedit.options.ToolBarOptionPane,1,1,13,6,0,14,75,44,0.814814815,3,5,38.07692308,4,13,1,
1,0,0.982195846,0.5,517,2.3077,12
bsh.ReflectError,0,0,2,3,0,7,4,1,2,0,0,3.5,7,0,2,0,0,1,0.75,9,0,0
org.gjt.sp.jedit.ActionListHandler,1,2,11,2,0,5,31,31,0.8,1,3,19.27272727,1,5,7,1,0,0.565217391,0.
4,231,2.2727,8
org.gjt.sp.jedit.gui.MarkersMenu,1,2,2,8,0,9,25,1,2,3,4,62.5,2,8,2,0,0,0.998835856,0.75,127,4.5,9
org.gjt.sp.jedit.browser.VFSFileChooserDialog,1,4,10,7,0,15,74,17,0.814814815,4,7,46.1,4,15,5,1,1,
0.986404834,0.25,477,2.3,9
org.gjt.sp.jedit.search.HyperSearchResult,1,1,6,1,0,8,23,3,0.68,0,0,19.5,4,4,5,0,1,0,0.458333333,1
28,1.3333,2

org.gjt.sp.jedit.gui.HistoryTextField,1,2,21,6,0,17,83,64,0.691666667,3,7,26.76190476,13,7,9,1,1,0
.978986403,0.224489796,589,2.7619,13
bsh.BSHImportDeclaration,0,0,2,2,0,10,9,1,1,1,1,16,1,9,1,0,0,0.947368421,0.625,36,0.5,1
bsh.BSHAssignment,0,0,3,2,0,11,18,3,1,1,1,89.66666667,1,10,1,0,0,0.9,0.533333333,273,0.6667,1
org.gjt.sp.jedit.gui.BeanShellErrorDialog,1,1,3,7,0,5,33,3,2,0,0,53,2,4,3,0,0,0.996946565,0.555555
556,162,0.6667,1
org.gjt.sp.jedit.options.LoadSaveOptionPane,1,1,5,6,0,4,35,0,0.711538462,3,5,73.6,2,3,3,1,0,0.9939
93994,0.5,386,2,6
bsh.SimpleNode,0,0,19,1,35,50,30,147,0.888888889,0,0,11.31578947,46,7,19,0.5,4,0,0.233082707,240,1
.6316,6
bsh.BSHForStatement,0,0,2,2,0,12,10,1,1.125,1,1,63.5,1,11,1,0.625,4,0.947368421,0.625,137,0.5,1
bsh.BSHVariableDeclarator,0,0,2,2,0,10,9,1,2,1,2,18.5,2,8,1,0,0,0.947368421,0.6,40,0.5,1
org.gjt.sp.util.WorkThreadProgressListener,0,0,1,1,0,5,1,0,2,0,0,0,5,1,1,0,0,0,1,1,1,1
org.gjt.sp.jedit.syntax.ParserRuleSet,1,1,17,1,0,8,22,100,0.91875,0,0,10.11764706,5,3,17,1,4,0,0.1
83823529,199,1.4118,4
org.gjt.sp.jedit.gui.IOProgressMonitor,1,1,5,6,0,10,33,4,0.666666667,1,1,28.8,4,8,2,1,2,0.99389313
,0.4,152,0.8,1
org.gjt.sp.jedit.pluginmgr.PluginListDownloadProgress,1,1,7,6,0,7,33,11,0.777777778,0,0,19.2857142
9,4,6,0,1,3,0.99086758,0.285714286,145,0.8571,1
org.gjt.sp.jedit.msg.BufferUpdate,0,0,6,3,0,14,14,5,0.927272727,1,1,10.66666667,11,5,5,0.181818182
,1,0.6,0.4,81,0.6667,1
org.gjt.sp.jedit.search.SearchFileSet,0,0,5,1,0,9,5,10,2,0,0,0,7,2,5,0,0,0,0.6,5,1,1
bsh.Parser,0,0,399,1,0,47,487,0,0.71839196,0,0,57.45864662,1,47,82,0.8,8,0,0.183375104,23350,5.543
9,46
org.gjt.sp.jedit.syntax.ParserRuleFactory,0,0,11,1,0,3,16,55,2,0,0,40.45454545,2,1,9,0,0,0,0.36363
6364,456,3.5455,6
bsh.TokenMgrError,0,0,6,3,0,2,19,15,1.12,1,1,28.83333333,2,0,4,0,0,0.85,0.5,184,2.8333,14
org.gjt.sp.jedit.BufferHistory,1,1,13,1,0,11,61,44,0.833333333,0,0,36.15384615,4,10,6,1,0,0,0.1805
55556,488,2.5385,6
bsh.BSHThrowStatement,0,0,2,2,0,7,7,1,2,1,1,14,1,6,1,0,0,0.947368421,0.625,30,0.5,1
org.gjt.sp.jedit.options.ToolBarAddDialog,0,0,13,7,0,10,72,44,0.782051282,0,0,46.53846154,2,9,4,1,
0,0.981954887,0.217948718,631,1.7692,7
bsh.BSHTernaryExpression,0,0,2,2,0,7,6,1,2,1,1,17,1,6,1,0,0,0.947368421,0.625,36,0.5,1
org.gjt.sp.jedit.search.DirectoryListSet,0,0,8,2,0,8,34,12,0.714285714,0,0,26.375,2,6,5,1,0,0.5,0.
291666667,223,2,8
org.gjt.sp.jedit.gui.StatusBar,1,4,14,5,0,20,72,37,0.857142857,2,5,36.07142857,17,11,8,0.928571429
,3,0.98048048,0.257142857,533,1.7857,6
bsh.XThis,0,0,5,2,0,4,18,8,0.75,1,2,20,1,4,2,0,0,0.714285714,0.333333333,106,1.2,3
org.gjt.sp.jedit.Mode,1,1,12,1,0,19,32,36,0.745454545,0,0,18.25,15,6,12,1,3,0,0.416666667,236,2,5
bsh.BSHLiteral,0,0,5,2,0,7,20,4,0.5,0,0,29.4,1,6,2,0,0,0.818181818,0.366666667,153,4,10
org.gjt.sp.jedit.browser.FileCellRenderer,1,4,4,5,0,5,32,0,0.575757576,0,0,74.5,1,4,2,0.909090909,
0,0.995601173,0.375,313,4,11
org.gjt.sp.jedit.options.AbbrevsModel,0,0,10,2,0,8,29,3,0.333333333,1,2,17.8,4,5,8,0,0,0.666666667
,0.425,189,1.9,4
org.gjt.sp.jedit.TextUtilities,1,5,9,1,0,8,39,36,2,0,0,99.44444444,5,4,9,0,0,0,0.301587302,904,6,2
0
org.gjt.sp.jedit.options.TextAreaOptionPane,0,0,3,6,0,4,24,1,0.5,2,4,102,1,3,3,1,1,0.996987952,1,3
21,1.6667,3
gnu.regexp.CharIndexedReader,0,0,5,1,0,3,12,0,0.46875,0,0,51.2,2,1,3,1,0,0,0.6,269,3.8,10

org.gjt.sp.util.Log,0,0,13,1,0,64,55,36,0.886904762,0,0,31.23076923,64,1,7,0.571428571,0,0,0.17857
1429,433,2.3077,7
bsh.EvalError,0,0,11,3,2,51,20,0,0.1,1,1,11.36363636,51,1,9,0,1,0.653846154,0.575757576,137,1.1818
,2
org.gjt.sp.jedit.syntax.KeywordMap,0,0,8,1,0,5,15,0,0.428571429,0,0,16,4,2,6,1,1,0,0.395833333,139
,1.375,5
org.gjt.sp.jedit.browser.BrowserPopupMenu,0,0,5,5,0,14,45,6,0.833333333,0,0,90,4,11,1,1,3,0.994436
718,0.32,458,1.4,3
org.gjt.sp.jedit.JARClassLoader,1,4,14,2,0,11,88,35,0.673076923,1,7,62.21428571,6,7,8,1,1,0.842105
263,0.619047619,889,2.0714,10
org.gjt.sp.jedit.gui.DefaultInputHandler,1,5,12,3,0,6,52,8,0.672727273,1,4,40.33333333,2,5,10,1,0,
0.666666667,0.303030303,501,4.3333,17
org.gjt.sp.jedit.gui.GrabKeyDialog,1,1,21,6,0,7,71,170,0.9,1,1,22.71428571,4,6,5,1,2,0.970193741,0
.169312169,508,1.8571,8
bsh.ConsoleInterface,0,0,6,1,0,1,6,15,2,0,0,0,1,0,6,0,0,0,0.75,6,1,1
bsh.BSHBlock,0,0,3,2,0,14,10,3,2,1,1,20,5,9,2,0,0,0.9,0.6,63,0.6667,1
bsh.BSHFormalParameters,0,0,2,2,0,9,6,1,1,1,2,24.5,3,6,1,0,0,0.947368421,0.666666667,54,0.5,1
bsh.ReflectManager,0,0,4,1,0,3,12,6,0.666666667,0,0,8,1,2,4,1,1,0,0.5,37,1,2
org.gjt.sp.jedit.pluginmgr.PluginManager,1,3,11,6,0,11,64,27,0.775,0,0,52.36363636,6,9,1,1,0,0.984
871407,0.363636364,595,1.9091,12
bsh.BSHPrimaryExpression,0,0,2,2,0,8,8,1,2,1,1,24,1,7,1,0,0,0.947368421,0.625,50,0.5,1
org.gjt.sp.jedit.search.CurrentBufferSet,0,0,6,1,0,6,8,15,2,0,0,3,3,3,6,0,0,0,0.555555556,24,1,2
org.gjt.sp.jedit.search.SearchBar,1,3,13,5,0,14,65,14,0.633333333,2,5,29.23076923,4,14,4,1,2,0.981
954887,0.323076923,398,1.6154,8
org.gjt.sp.jedit.textarea.MarkerHighlight,0,0,8,1,0,8,28,6,0.678571429,0,0,18.5,1,7,8,1,2,0,0.2343
75,160,2,6
org.gjt.sp.jedit.gui.AboutDialog,1,2,3,7,0,4,33,3,1,0,0,57.33333333,1,4,3,1,0,0.996946565,0.666666
667,176,0.6667,1
org.gjt.sp.jedit.BeanShell,1,4,15,1,0,26,75,73,0.785714286,0,0,47.66666667,9,19,12,1,3,0,0.2476190
48,736,4.4,13
org.gjt.sp.jedit.msg.EditPaneUpdate,0,0,5,3,0,5,13,4,0.875,1,1,8.2,2,4,4,0.25,0,0.666666667,0.5,50
,0.6,1
bsh.BSHTypedVariableDeclaration,0,0,3,2,0,11,13,3,1,1,1,32.33333333,1,10,1,0,0,0.9,0.444444444,101
,2.3333,6
org.gjt.sp.jedit.EditAction,0,0,16,1,2,23,25,106,0.8,0,0,6.875,21,2,15,1,0,0,0.205357143,128,1.187
5,3
org.gjt.sp.jedit.EditServer,1,4,6,2,0,6,50,9,0.8,0,0,49.16666667,3,5,3,1,0,0.924242424,0.305555556
,305,2,8
org.gjt.sp.jedit.gui.MacrosMenu,0,0,6,8,0,10,24,15,2,1,7,23,1,10,4,0,0,0.994206257,0.333333333,144
,2.1667,5
org.gjt.sp.jedit.options.ModeOptionPane,1,3,12,6,0,6,38,28,0.392344498,1,3,51.16666667,3,5,1,1,2,0
.983655275,0.5,645,1.4167,5
org.gjt.sp.jedit.gui.OptionsDialog,1,2,14,7,0,24,106,69,0.869230769,0,0,48.85714286,3,24,8,1,2,0.9
8048048,0.174603175,708,2.2143,9
org.gjt.sp.jedit.ModeCatalogHandler,1,1,6,2,0,5,28,3,0.771428571,1,3,36.16666667,1,5,4,1,0,0.72222
2222,0.722222222,230,3,7
org.gjt.sp.jedit.textarea.TextAreaHighlight,0,0,3,1,0,5,3,3,2,0,0,0,4,1,3,0,0,0,0.444444444,3,1,1
org.gjt.sp.jedit.EBPlugin,0,0,2,2,0,4,3,1,2,0,0,2,1,3,1,0,0,0.9,0.75,6,0.5,1
org.gjt.sp.jedit.MiscUtilities,1,4,28,1,0,45,60,378,2,0,0,32.92857143,43,2,24,0,0,0,0.183673469,95
0,3.7143,14

gnu.regexp.RETokenStart,0,0,3,2,0,4,9,1,0.5,1,2,34,1,3,0,1,0,0.777777778,0.444444444,106,4,11
org.gjt.sp.jedit.textarea.TextRenderer2D,0,0,6,2,0,1,23,1,0.6,1,2,22.16666667,0,1,2,1,0,0.6875,0.476190476,141,1.3333,3
org.gjt.sp.jedit.search.BoyerMooreSearchMatcher,1,3,7,1,0,5,16,0,0.666666667,0,0,57.14285714,1,4,4,1,1,0,0.265306122,416,4.1429,14
org.gjt.sp.jedit.options.StyleEditor,0,0,5,7,0,5,52,2,0.642857143,0,0,73.6,1,4,4,1,0,0.99391172,0.4,380,2.6,6
org.gjt.sp.jedit.proto.jeditresource.PluginResURLConnection,1,1,5,2,0,5,28,2,0.5625,1,7,38,1,4,4,1,0,0.935483871,0.466666667,199,2.2,8
org.gjt.sp.jedit.Buffer,1,25,126,3,0,102,316,6941,0.95255814,1,1,38.71428571,79,40,90,0.790697674,7,0.324324324,0.068783069,5047,3.5317,25
org.gjt.sp.jedit.gui.BufferOptions,1,6,15,7,0,12,91,47,0.808035714,1,1,77,2,11,3,1,3,0.979010495,0.283333333,1186,1.3333,7
gnu.regexp.CharIndexed,0,0,3,1,0,24,3,3,1.5,0,0,0,24,0,3,0,0,0,0.833333333,4,1,1
bsh.InterpreterError,0,0,1,4,0,16,2,0,2,0,0,4,16,0,1,0,0,1,1,5,0,0
org.gjt.sp.jedit.gui.HistoryModel,0,0,10,1,0,10,47,17,0.622222222,0,0,28.4,8,3,8,1,0,0,0.35,299,2.8,10
com.microstar.xml.XmlHandler,0,0,13,1,0,6,13,78,2,0,0,0,6,0,13,0,0,0,0.430769231,13,1,1
org.gjt.sp.jedit.syntax.ParserRule,0,0,1,1,0,6,2,0,2,0,0,15,6,0,0,0,1,0,1,21,0,0
org.gjt.sp.jedit.gui.SplashScreen,0,0,3,5,0,4,37,1,0.5,0,0,44.66666667,2,3,2,1,0,0.996710526,0.5,138,1,2
org.gjt.sp.jedit.options.ShortcutsOptionPane,0,0,13,6,0,11,60,50,0.816666667,4,12,26.38461538,5,10,1,1,1,0.982195846,0.243589744,361,1.9231,5
org.gjt.sp.jedit.gui.ViewRegisters,0,0,6,7,0,8,47,9,0.8,0,0,50.33333333,3,8,3,1,0,0.992401216,0.388888889,311,0.8333,1
org.gjt.sp.jedit.gui.EnhancedMenu,0,0,1,7,6,8,21,0,2,0,0,92,7,2,1,0,0,1,1,93,0,0
org.gjt.sp.jedit.gui.DockableWindow,0,0,2,1,0,12,2,1,2,0,0,0,12,0,2,0,0,0,1,3,1,1
org.gjt.sp.jedit.Autosave,0,0,4,1,0,2,11,4,0.666666667,0,0,13.25,1,2,3,1,0,0,0.333333333,58,2,4
org.gjt.sp.jedit.gui.CompleteWord,1,1,6,5,0,5,34,0,0.4,0,0,20,3,5,2,1,1,0.991816694,0.277777778,129,0.8333,1
bsh.BSHMethodDeclaration,0,0,3,2,0,11,14,1,0.875,1,3,42.33333333,2,10,2,0,2,0.9,0.555555556,134,0.6667,1
org.gjt.sp.jedit.options.ContextOptionPane,1,1,13,6,0,13,65,46,0.785714286,2,4,32.15384615,7,9,1,1,0,0.982195846,0.333333333,438,2.2308,9
bsh.BSHSwitchLabel,0,0,2,2,0,7,5,1,1,1,1,9.5,2,5,2,0,0,0.947368421,0.625,22,0.5,1
org.gjt.sp.jedit.io.UrlVFS,1,1,6,2,0,4,19,15,2,1,3,14.5,1,4,6,0,0,0.807692308,0.5,93,1,2
org.gjt.sp.util.WorkThread,1,1,14,2,0,8,34,37,0.760683761,0,0,21.5,5,4,11,1,1,0.824324324,0.224489796,324,1.7143,8
bsh.BSHPrimarySuffix,0,0,5,2,0,16,35,10,1.142857143,0,0,48.8,2,14,1,0,0,0.818181818,0.72,256,0.8,1
org.gjt.sp.jedit.Abbrevs,0,0,18,1,0,12,96,47,0.773109244,0,0,39.16666667,4,10,9,1,0,0,0.104575163,730,2.7222,16
bsh.BSHPrimitiveType,0,0,2,2,0,4,3,1,1,0,0,3.5,3,1,1,0,0,0.947368421,0.75,10,0.5,1
org.gjt.sp.jedit.io.BufferIORequest,1,9,11,2,0,10,96,21,0.828571429,0,0,113.3636364,2,10,3,0.5,3,0.333333333,0.236363636,1272,4.9091,17
org.gjt.sp.jedit.search.BufferListSet,0,0,9,1,2,9,24,6,0.375,0,0,18.22222222,4,7,8,1,0,0,0.296296296,174,1.8889,6
org.gjt.sp.jedit.gui.FontSelector,0,0,3,6,0,5,18,3,2,0,0,22.66666667,4,2,1,0,0,0.997382199,0.444444444,71,1,2
org.gjt.sp.jedit.textarea.Selection,1,1,10,1,2,15,17,23,0.777777778,0,0,6.6,14,1,7,0,0,0,0.4,80,0.7,1

org.gjt.sp.jedit.syntax.Token,1,1,2,1,0,4,7,0,1.8,0,0,13,4,0,2,0,2,0,0.666666667,48,0.5,1
org.gjt.sp.jedit.textarea.TextAreaPainter,1,16,45,4,0,15,122,462,0.899521531,3,18,23.97777778,8,8,39,1,3,0.936231884,0.136752137,1143,2.0889,12
bsh.BSHIfStatement,0,0,3,2,0,11,10,3,2,1,1,25.33333333,4,7,2,0,0,0.9,0.533333333,79,0.6667,1
org.gjt.sp.jedit.syntax.SyntaxStyle,0,0,4,1,0,10,5,0,0.666666667,0,0,5.25,10,0,4,1,0,0,0.5,28,0.75,1
org.gjt.sp.jedit.Marker,1,2,6,1,0,8,10,0,0.65,0,0,7.833333333,7,2,3,1,1,0,0.416666667,57,1,2
gnu.regexp.RE,0,0,45,2,0,34,107,928,0.93006993,1,2,55,13,25,26,0.461538462,2,0.152173913,0.231060606,2533,1.7778,9
org.gjt.sp.jedit.msg.EditorExiting,0,0,1,3,0,3,2,0,2,0,0,4,1,2,1,0,0,1,1,5,0,0
gnu.regexp.RETokenPOSIX,0,0,6,2,0,4,22,3,1.0625,1,2,44.33333333,1,3,0,0,0,0.636363636,0.285714286,288,5.1667,25
gnu.regexp.REFilterInputStream,0,0,5,3,0,4,18,8,0.375,2,3,32.4,0,4,5,1,2,0.818181818,0.366666667,173,2.6,7
org.gjt.sp.jedit.msg.SearchSettingsChanged,0,0,1,3,0,4,2,0,2,0,0,4,2,2,1,0,0,1,1,5,0,0
gnu.regexp.RETokenAny,0,0,4,2,0,4,8,4,0.666666667,1,2,12.25,1,3,0,1,0,0.7,0.375,55,2,6
bsh.NameSpace,0,0,51,1,1,44,135,1139,0.949090909,0,0,30.80392157,34,18,35,0.818181818,4,0,0.111764706,1644,1.5882,6
bsh.ParserConstants,0,0,1,1,0,12,1,0,2,0,0,496,12,0,0,0,0,0,0,617,0,0
org.gjt.sp.jedit.msg.MacrosChanged,0,0,1,3,0,4,2,0,2,0,0,4,2,2,1,0,0,1,1,5,0,0
org.gjt.sp.jedit.search.CharIndexedSegment,0,0,4,1,0,2,5,0,0,0,0,14.25,1,1,3,1,0,0,0.666666667,63,1.5,2
org.gjt.sp.jedit.pluginmgr.PluginManagerProgress,1,1,17,6,0,14,59,94,0.8984375,3,6,18.17647059,13,9,8,1,1,0.976011994,0.213235294,334,1.3529,5
gnu.regexp.RETokenWordBoundary,0,0,3,2,0,4,8,0,0.75,1,2,48,1,3,0,0.5,0,0.777777778,0.444444444,151,7.3333,18
org.gjt.sp.jedit.gui.AddAbbrevDialog,0,0,6,6,0,7,35,5,0.833333333,0,0,36.16666667,3,6,1,1,2,0.992378049,0.333333333,229,0.8333,1
bsh.BSHFormalParameter,0,0,3,2,0,9,7,1,1,1,2,8,3,6,1,0,0,0.947368421,0.666666667,30,0.3333,1
com.microstar.xml.XmlParser,0,0,118,1,0,6,180,6075,0.921125975,0,0,42.48305085,5,1,27,0.717391304,1,0,0.138067061,5223,1.4492,9
org.gjt.sp.jedit.io.VFSManager,1,2,23,1,0,39,64,159,0.858585859,0,0,11.65217391,33,13,16,1,3,0,0.103896104,300,1.4783,9
org.gjt.sp.jedit.OptionGroup,0,0,9,1,0,5,20,0,0.5,0,0,11.11111111,3,2,9,1,0,0,0.259259259,111,1.5556,3
bsh.JJTParserState,0,0,12,1,0,3,25,0,0.436363636,0,0,16.75,2,1,0,1,0,0,0.395833333,218,1.3333,3
bsh.StringUtil,0,0,5,1,0,1,24,10,2,0,0,40.2,1,0,5,0,0,0,0.3,206,3,6
org.gjt.sp.jedit.options.Abbrev,0,0,2,1,0,2,3,1,1,0,0,6,2,0,0,0,0,0,0.75,16,0,0
org.gjt.sp.jedit.OptionPane,0,0,4,1,0,5,4,6,2,0,0,0,5,0,4,0,0,0,1,4,1,1
gnu.regexp.RETokenChar,0,0,5,2,0,5,12,0,0.375,1,2,24,2,3,0,1,0,0.636363636,0.3,127,1.6,4
org.gjt.sp.util.WorkRequest,0,0,6,1,3,4,12,15,2,0,0,7.166666667,3,1,6,0,0,0,0.416666667,49,1.5,2
org.gjt.sp.jedit.pluginmgr.InstallPluginsDialog,1,4,12,7,0,9,72,40,0.943181818,1,3,54.91666667,3,8,2,0.875,1,0.983433735,0.222222222,687,1.25,5
bsh.BSHCastExpression,0,0,6,2,0,16,45,15,2,1,1,45.66666667,3,13,6,0,0,0.782608696,0.229166667,280,0.8333,1
com.microstar.xml.HandlerBase,0,0,14,1,6,8,16,91,2,0,0,1.714285714,6,2,14,0,0,0,0.414285714,38,0.9286,1
org.gjt.sp.jedit.msg.ViewUpdate,0,0,5,3,0,6,13,4,0.833333333,1,1,7.8,2,4,4,0.333333333,0,0.666666667,0.5,47,0.6,1

org.gjt.sp.jedit.gui.TipOfTheDay,1,2,6,7,0,6,18,13,0.8,0,0,8,3,4,3,1,0,0.992401216,0.25,55,0.8333,1
org.gjt.sp.jedit.gui.PluginsMenu,0,0,1,8,0,9,22,0,2,0,0,110,1,9,1,0,0,1,1,111,0,0
bsh.BSHUnaryExpression,0,0,5,2,0,11,33,6,0.625,1,1,39.4,1,10,1,0,0,0.818181818,0.433333333,204,0.8,1
bsh.BSHReturnStatement,0,0,2,2,0,9,7,1,1,1,1,12.5,1,8,1,0,0,0.947368421,0.625,28,0.5,1
gnu.regexp.RETokenBackRef,0,0,3,2,0,4,8,0,0.5,1,2,24,1,3,0,1,0,0.777777778,0.444444444,77,2,5
org.gjt.sp.jedit.pluginmgr.PluginList,1,2,5,1,0,11,26,0,0.5,0,0,29.4,4,8,0,0,0,0,0.466666667,155,1.8,5
gnu.regexp.CharIndexedCharArray,0,0,4,1,0,2,5,0,0,0,0,13.25,1,1,3,1,0,0,0.666666667,59,1.75,3
bsh.TargetError,0,0,11,4,0,18,26,13,0.1,2,2,14.45454545,14,5,11,0,0,0.764705882,0.545454545,171,0.9091,2
gnu.regexp.CharIndexedInputStream,0,0,5,1,0,3,12,0,0.53125,0,0,47.6,2,1,3,1,0,0,0.6,251,3.8,10
bsh.BSHAllocationExpression,0,0,8,2,0,18,37,28,2,1,1,32.875,1,17,1,0,0,0.72,0.363636364,271,0.875,1
bsh.BSHMethodInvocation,0,0,2,2,0,12,17,1,2,1,1,30.5,1,11,1,0,0,0.947368421,0.625,63,0.5,1
org.gjt.sp.jedit.msg.VFSUpdate,0,0,3,2,0,6,9,0,0,1,1,10,4,2,3,1,0,0.714285714,0.666666667,34,0.6667,1
bsh.BSHReturnType,0,0,2,2,0,8,5,1,1,0,0,8,2,6,1,0,0,0.947368421,0.666666667,19,0.5,1
bsh.BSHStatementExpressionList,0,0,2,2,0,7,6,1,2,1,1,13,1,6,1,0,0,0.947368421,0.625,28,0.5,1
bsh.BshMethod,0,0,4,1,0,13,20,0,0.555555556,0,0,73.5,5,10,4,0.333333333,2,0,0.357142857,301,1,2
org.gjt.sp.jedit.gui.LogViewer,0,0,3,5,0,5,14,3,2,0,0,17.33333333,1,4,3,0,0,0.996946565,1,55,0.6667,1
gnu.regexp.IntPair,0,0,1,1,0,1,2,0,2,0,0,3,1,0,0,0,0,0,1,6,0,0
org.gjt.sp.jedit.io.FileVFS,1,6,21,2,0,12,94,198,0.94,2,4,45.47619048,5,9,19,0.6,0,0.525,0.375,981,3.1905,16
org.gjt.sp.jedit.gui.EnhancedCheckBoxMenuItem,1,2,7,7,0,5,38,9,0.611111111,3,13,28.14285714,2,4,4,1,1,0.993811881,0.3,210,2.1429,8
org.gjt.sp.jedit.jEdit,1,23,99,1,0,162,325,4043,0.915102041,0,0,42.85858586,137,48,67,1,7,0,0.059417706,4367,4.0707,56
org.gjt.sp.jedit.msg.EditorStarted,0,0,1,3,0,3,2,0,2,0,0,4,1,2,1,0,0,1,1,5,0,0
bsh.BSHAmbiguousName,0,0,7,2,0,16,17,19,0.333333333,1,1,6.428571429,8,8,6,0,0,0.75,0.457142857,53,0.8571,1
org.gjt.sp.jedit.io.FileRootsVFS,0,0,6,2,0,4,20,13,0.95,1,3,19.33333333,1,3,5,0.75,0,0.807692308,0.541666667,126,1.3333,4
org.gjt.sp.jedit.search.RESearchMatcher,1,4,4,1,0,11,25,0,0.611111111,0,0,33.75,2,9,3,0.833333333,3,0,0.583333333,145,1,3
org.gjt.sp.jedit.search.HyperSearchRequest,1,3,6,2,0,14,40,3,0.72,1,1,40.16666667,3,14,2,1,3,0.5,0.305555556,252,2.3333,10
bsh.BlockNameSpace,0,0,3,2,0,4,10,3,2,0,0,12.33333333,2,2,2,0,0,0.96,0.583333333,40,1,2
org.gjt.sp.jedit.gui.InputHandler,1,6,16,2,1,13,47,48,0.780952381,0,0,26.25,8,10,14,1,2,0.166666667,0.241071429,443,2.625,14
bsh.NameSource,0,0,2,1,0,2,2,1,2,0,0,0,2,1,2,0,0,0,0.75,2,1,1
org.gjt.sp.jedit.pluginmgr.Roster,0,0,5,1,0,9,14,0,0,0,0,15.2,8,2,0,1,0,0,0.466666667,82,2,4
bsh.CommandLineReader,0,0,4,3,0,1,8,4,1,2,2,18.5,1,0,4,0,0,0.857142857,0.35,82,0.75,1
bsh.CallStack,0,0,11,1,0,39,25,0,0,0,0,11.18181818,38,2,11,1,0,0,0.484848485,135,1.1818,2
org.gjt.sp.jedit.search.HyperSearchResults,1,2,13,5,0,17,53,54,0.875,3,12,21.84615385,8,12,9,0.833333333,1,0.981954887,0.261538462,303,1.6923,10

Παράρτημα Ε
Προγράμματα Μετρικών που Αξιολογήθηκαν

Στο παρών παράρτημα δίνουμε μια σύντομη περιγραφή των εργαλείων για μετρικές λογισμικού που αξιολογήθηκαν στα πλαίσια διεξαγωγής της έρευνας μας:

- **JArchitect**: Εϊναι freeware λογισμικό που μπορεί να χρησιμοποιηθεί για κάποιες μέρες χωρίς κανένα περιορισμό αλλά μετά την λήξη της δοκιμαστικής περιόδου θα πρέπει να αγοραστεί. Περιλαμβάνει αρκετές μετρικές και οι περισσότερες από αυτές αφορούν αντικειμενοστρεφή χαρακτηριστικά της γλώσσας Java.

- **Understand**: Εμπορικό λογισμικό που ως στόχο έχει την μέτρηση, την ανάλυση και την συντήρηση του πηγαίου κώδικα. Δέχεται ως είσοδο οποιαδήποτε σύγχρονη γλώσσα προγραμματισμού αλλά και παλιότερες όπως Ada, Pascal, Fortran, PL/M κ.α. Διαθέτει περισσότερες από τριάντα μετρικές και αυτοματοποιημένες αναφορές προβλημάτων. Μεγάλη δύναμη του δίνουν οι γραφικές παραστάσεις που μπορεί να δημιουργήσει και τα διάφορα πρόσθετα (plugins) που επεκτείνουν τις δυνατότητες του ώστε να προσαρμόζεται στις ιδιαίτερες ανάγκες του κάθε χρήστη.

- **RSM**: Εϊναι freeware λογισμικό που μπορεί να χρησιμοποιηθεί για ένα περιορισμένο αριθμό αρχείων χωρίς να αγοραστεί. Υποστηρίζει όλες τις σύγχρονες γλώσσες όπως C, C++, C#, και Java αλλά σε ότι αφορά τις μετρικές που μας ενδιέφεραν για την έρευνα μας είναι σχετικά φτωχό. Πολύ μεγάλο του προσόν είναι τα πενήντα έξι είδη ειδοποιήσεων που προσφέρει σχετικά με τις προγραμματιστικές τακτικές και κανόνες που θα πρέπει να τηρούνται στον πηγαίο κώδικα. Δυστυχώς, ο περιορισμός των αρχείων στην ελεύθερη του έκδοση δεν μας επέτρεψε να το χρησιμοποιήσουμε για να διαπιστώσουμε πόσο καλή δουλεία κάνουν αυτές οι προειδοποιήσεις της ανάλυσης ποιότητας του εργαλείου.

- **EZ-Metrix:** Είναι ένα freeware εργαλείο που όμως πέρα από τις βασικές μετρικές δεν περιλαμβάνει κάτι παραπάνω. Το γεγονός ότι δουλεύει μόνο μέσω διαδικτύου μάλλον τελικά μας δυσκόλεψε, γιατί είχαμε ένα μεγάλο αριθμό αρχείων που έπρεπε να εξεταστούν στα πλαίσια της έρευνας μας.

- **Code Counter Pro:** Εμπορικό λογισμικό και μάλιστα σε πάρα πολύ καλή τιμή. Δυστυχώς, η τρέχουσα έκδοση δεν μπορούσε να τρέξει σε 64bit λειτουργικό σύστημα και έτσι δεν μπορέσαμε να το τρέξουμε ώστε να το αξιολογήσουμε.

- **Insight:** Πρόκειται για ένα εμπορικό λογισμικό από την γνωστή εταιρία Klockwork που διαθέτει μια πλούσια σειρά εργαλείων σχετικά με την παραγωγή και την διαχείριση του πηγαίου κώδικα. Πέρα από όλες τις γνωστές μετρικές που διαθέτει, μπορεί να κάνει ανάλυση του κώδικα για προβλήματα ασφαλείας, αναδόμηση του, ιστορικό συντήρησης και συσχέτισης του με τα διάφορα προβλήματα, plugin για όλα τα γνωστά IDEs.

- **CTM Java**: Είναι ένα εμπορικό εργαλείο της εταιρίας Testwell που διανέμεται μέσω της Verisoft. Αν και η γραφική διεπαφή του μοιάζει αρκετά ξεπερασμένη σε σχέση με τα υπόλοιπα εμπορικά εργαλεία που εξετάσαμε, διακρίνεται για την πολύ μεγάλη ταχύτητα επεξεργασίας του πηγαίου κώδικα και για την πληρότητα των μετρικών που παρέχει σχετικά με την πολυπλοκότητα του κώδικα όπως την κυκλωματική πολυπλοκότητα, τις μετρικές Halstead, τον δείκτη συντηρησιμότητας κ.α.

- **McCabe IQ**: Εμπορικό λογισμικό που αναλύει τον πηγαίο κώδικα για πιθανά σφάλματα και πολυπλοκότητα. Μεγάλες εταιρίες χρησιμοποιούν αυτό το λογισμικό που μάλιστα έρχεται σε τρεις διαφορετικές εκδόσεις ώστε να καλύψει όλες τις πιθανές ανάγκες από την ομάδα των προγραμματιστών, των μηχανικών λογισμικού ή και των δυο μαζί. Πέρα

από το πολύ κατατοπιστικό βίντεο που υπάρχει στο δικτυακό τους τόπο, δεν μπορέσαμε να αξιολογήσουμε το λογισμικό αφού δεν απάντησαν στο email που στείλαμε.

- **Logiscope**: Πρόκειται για ένα από τα πιο γνωστά εμπορικά εργαλεία σχετικά με λειτουργίες για την διασφάλιση ποιότητας. Περιλαμβάνει μια μεγάλη γκάμα από στατικές και δυναμικές μετρικές σχετικά με την πολυπλοκότητα και την συντηρησιμότητα του πηγαίου κώδικα. Μπορούν να οριστούν προσαρμοσμένες μετρικές και είναι συμβατό με τις γλώσσες προγραμματισμού C, C++ και Java. Δυστυχώς, παρά την προσπάθεια μας να μας αποσταλεί δοκιμαστική έκδοση αυτό δεν έγινε και έτσι δεν μπορέσαμε να το δούμε αναλυτικά.

- **Application Intelligence Platform**: Είναι ένα εμπορικό λογισμικό που μπορεί να αναλύσει σχεδόν οποιαδήποτε σύγχρονη γλώσσα προγραμματισμού περιλαμβανομένων των scripting και interface γλωσσών, 3GLS και 4GLS. Η δυνατότητες του δεν περιορίζονται μόνο στην ανάλυση κώδικα αλλά περιλαμβάνει ολόκληρη διαχείριση σχετικά με την ποιότητα λογισμικού και μπορεί να παρακολουθήσει μέχρι και οικονομικά στοιχεία π.χ. κόστος συντήρησης ενός προγράμματος.

- **Sonar**: Είναι ένα εργαλείο ανοικτού κώδικα για την διαχείριση της ποιότητας κώδικα και είναι πάρα πολύ καλό, αφού συμπεριλαμβάνει δυνατότητες ορισμού κανόνων κωδικοποίησης, ποικιλία μετρικών, δοκιμές μονάδων και άλλα πολλά. Δεν χρησιμοποιήθηκε στα πλαίσια της διπλωματικής εργασίας γιατί ενώ προσφέρει ένα μεγάλο αριθμό μετρικών, δεν υπάρχουν ολοκληρωμένες όλες οι συλλογές μετρικών που επιλέξαμε στην ενότητα 4.1. Κατά την άποψη μας, είναι ένα εργαλείο που στο μέλλον έχει όλα τα φόντα για να πρωταγωνιστήσει στην διαχείριση της ποιότητας λογισμικού και να γίνει ισάξιο με τα αντίστοιχα εμπορικά λογισμικά αρκεί να προστεθούν κάποιες επιπλέον λειτουργίες στην πλατφόρμα που έχει ήδη αναπτυχτεί.

- **SourceMonitor**: Πρόκειται για ένα freeware εργαλείο της εταιρίας Campwood Software για τις γλώσσες προγραμματισμού C, C++, C#, Java, VB.NET, Visual Basic, Delphi και HTML. Η έμφαση δίνεται στην στατική ανάλυση κώδικα και στην αναθεώρηση του μέσω της χρήσης σημείων ελέγχου (checkpoints) κατά την διάρκεια της παραγωγής του πηγαίου κώδικα. Το μεγαλύτερο του μειονέκτημα είναι ο σχετικά μικρός αριθμός μετρικών που υποστηρίζει.

- **CCCC**: Είναι ένα εργαλείο για στατικές μετρικές κώδικα που δημιουργήθηκε από τον Tim Littlefair ως μέρος της διδακτορικής διατριβής του στο κομμάτι που αφορούσε την εύρεση των ποιοτικών χαρακτηριστικών του πηγαίου κώδικα. Από τότε βρίσκεται σε συνεχή εξέλιξη και είναι διαθέσιμο μέσω του SourceForge. Υποστηρίζει τις γλώσσες προγραμματισμού C, C++ και Java. Μεταξύ των μετρικών που υποστηρίζει είναι διάφορες παραλλαγές του LOC, της κυκλωματικής πολυπλοκότητας, των μετρικών που έχουν προταθεί από τους Chidamber και Kemerer, Henry και Kafura κλπ.

- **Code Analyzer**: Είναι ένα εργαλείο ανοικτού κώδικα αλλά δεν προσφέρει πολλές μετρικές πέρα από αυτές που αφορούν τον αριθμό των γραμμών, των κενών γραμμών και των σχολίων. Είναι όμως ιδανικό για μια γρήγορη εικόνα του πηγαίου κώδικα και πιθανώς στο μέλλον να προστεθούν και άλλες λειτουργίες .

Παράρτημα ΣΤ
Συνοδευτικά Αρχεία

Στο παρόν παράρτημα αναφέρουμε όλα τα αρχεία που είτε χρησιμοποιήθηκαν κατά τα πλαίσια της έρευνας μας είτε ήταν το αποτέλεσμα αυτής:

- **BugInfo.jar**: Είναι το εργαλείο που χρησιμοποιήσαμε για να αντιστοιχίσουμε τα σφάλματα με τις κλάσεις του πηγαίου κώδικα. Η λογική του είναι να διαβάζει τα σχόλια των προγραμματιστών κατά την υποβολή κώδικα και με βάση κάποια κριτήρια να αποφασίζει ποιές υποβολές αφορούν σφάλματα. Περισσότερα στην ενότητα 4.3.

- **ckjm_ext**: Είναι μια επεκταμένη έκδοση του προγράμματος ckjm που διαβάζει τον πηγαίο κώδικα που είναι γραμμένος στην γλώσσα προγραμματισμού και αναφέρει δέκα εφτά αντικειμενοστρεφείς μετρικές και δυο παραδοσιακές (Κυκλωματική Πολυπλοκότητα και Γραμμές Πηγαίου Κώδικα). Περισσότερα στην ενότητα 4.2.

- **jEdit-3.2.ARFF**: Το αρχείο με όλα τα δεδομένα που αποτέλεσαν την είσοδο για την δημιουργία των μοντέλων στο WEKA. Περιλαμβάνει είκοσι μετρικές ανά κλάση για τον πηγαίο κώδικα του προγράμματος ανοικτού λογισμικού jEdit 3.2, το όνομα της κλάσης, τον αριθμό των σφαλμάτων που βρέθηκαν και δυαδική μεταβλητή για την ύπαρξη ή όχι σφάλματος. Η διαδικασία της συλλογής τους περιγράφεται αναλυτικά στην ενότητα 4.3

- **jEdit-3.2.cvs**: Το ίδιο με το παραπάνω αρχείο αλλά σε μορφή κειμένου ώστε να μπορεί να χρησιμοποιηθεί από το πρόγραμμα στατιστικής ανάλυσης R.

- **jEdit-3.2.xls**: Το ίδιο με το παραπάνω αρχείο αλλά σε μορφή Microsoft Excel 2003.

- **jEdit-3.2.jar**: Ο πηγαίος κώδικας του προγράμματος jEdit 3.2 σε μορφή bytecode που χρησιμοποιήθηκε για τις μέτρηση των μετρικών με το εργαλείο ckjm extended.

- **jEdit-3.2.txt:** Είναι η αρχική έξοδος του εργαλείου ckjm extended όταν μετρήθηκε το αρχείο jEdit3-2.jar. Αυτή η έξοδος χρειάζεται επεξεργασία γιατί π.χ. δεν αναφέρει την μέγιστη ή την μέση κυκλωματική πολυπλοκότητα αλλά την κυκλωματική πολυπλοκότητα ανά μέθοδο κάθε κλάσης. Περισσότερες λεπτομέρειες δίνονται στην ενότητα 4.3.

Παράρτημα Z
Απόδοση Όρων στα Αγγλικά

Στο παρόν παράρτημα δίνουμε την απόδοση των όρων από τα Ελληνικά στα Αγγλικά όπως αυτή αποδόθηκε στο κείμενο. Την πρώτη φορά που ένας Αγγλικός όρος εμφανίζεται στο κείμενο της μεταπτυχιακής διατριβής εμφανίζεται σε παρένθεση για την διευκόλυνση του αναγνώστη αλλά δεν επαναλαμβάνεται στη συνέχεια. Έτσι οι δυο παρακάτω ενότητες μπορούν να φανούν ιδιαίτερα χρήσιμες στην περίπτωση που υπάρχει αμφιβολία για την απόδοση ενός όρου ή χρειαζόμαστε την αγγλική του απόδοση για περεταίρω έρευνα πάνω σε ένα συγκεκριμένο θέμα.

Z.1 Από Ελληνικά σε Αγγλικά

Ελληνικά	Αγγλικά
Ακρίβεια	Precision
Ακτινική Βάση Συνάρτησης	Radial Basis Function
Αναδόμηση	Refactoring
Αναθεώρηση	Review
Αναθεώρηση (Πηγαίου Κώδικα)	Revision
Αναθεώρηση Ομότιμων	Peer Review

Ελληνικά	Αγγλικά
Ανάκληση	Recall
Ανάλυση Κυρίων Συνιστωσών	Principal Component Analysis
Αναμενόμενη Πληροφορία	Expected Information
Ανεξάρτητη Μεταβλητή	Explanatory Variable
Αντικειμενοτρεφή ανάλυση και σχεδίαση	Object Oriented Analysis and Design - OOAD
Αξιολόγηση και Επιλογή Λογισμικού Ανοικτού Κώδικα	Qualification and Selection of Open Source Software
Απλή Γραμμική Παλινδρόμηση	Univariate Linear Regression
Απόκριση για μια Κλάση	Responce for a Class - RFC
Απόσταση	Distance - D
Αριθμός Απογόνων	Number of Children - NOC
Αριθμός Γραμμών Κώδικα	Lines of Code - LOC
Αριθμός Δημοσίων Μεθόδων	Number of Public Methods - NPM
Αριθμός Δημόσιων Μεθόδων	Number of Public Methods - NPM
Αριθμός Δημόσιων Μεταβλητών	Number of Public Variables - NPV
Αριθμός Επικαλυπτόμενων Μεθόδων	Number of Operations Overridden - NOO
Αριθμός Κλάσεων	Number of Classes - NC
Αριθμός Μεθόδων Κλάσης	Number of Class Methods - NCM
Αριθμός Μεταβλητών Κλάσης	Number of Class Variables - NCV
Αριθμός Παραδομένων Λαθών	Number of Delivered Bugs
Αριθμός Προστιθέμενων Μεθόδων	Number of Operations Added - NOA
Αρμονικός Διαιρέτης	F-measure
Αρχείο Ιστορικού	Log File
Αστάθεια	Instability - I
Ασυμμετρία	Skewness
Βάθος Δέντρου Κληρονομικότητας	Depth of Inheritance Tree - DIT
Γενετικός προγραμματισμός	Genetic Programming
Δεδομένα Εκπαίδευσης	Training Set
Δεδομένα Ελέγχου	Testing Set
Δείκτης Εξειδίκευσης	Specialization Index - SI

Ελληνικά	Αγγλικά
Δείκτης Συντηρησιμότητας	Maintainability Index
Δέντρο Απόφασης	Decision Tree
Δεσμευμένη Πιθανότητας	Conditional Probability
Διακλάδωση	Branch
Διακρίνουσα Ανάλυση	Discriminant Analysis
Διασταυρωμένη επικύρωση	Cross Validation
Διάστημα Εμπιστοσύνης	Confidence Interval
Διατάξιμη Αποτίμηση	Ordinal Evaluation
Δοκιμή πριν την Υποβολή του Κώδικα	Pre-Commit Test
Δυαδικό Ψηφίο	Bit
Έκδοση Προγραμματιστή	Development Release
Εκτελέσιμες Εντολές	Executable Statements - ES
Ελαχίστων Τετραγώνων	Least Squares
Ελεύθερου Επιβάτη	Free Rider
Ελεύθερου Λογισμικού	Free Software
Έλλειψη Συνεκτικότητας των Μεθόδων	Lack of Cohesion Metric - LCOM
Εμπρόσθια Τροφοδότηση	Feed Forward
Ενδιάμεσες Κατασκευές	Intermediate Constructs
Εντροπία	Entropy
Εξαρτημένη Μεταβλητή	Response Variable
Εξυπηρετητής Παγκόσμιου Ιστού	Web Server
Επικάλυψη	Overriding
Επιλογή Μεταβλητών Προς τα Εμπρός	Forward Stepwise Selection
Επίπεδο Δυσκολίας	Difficulty Level
Επίπεδο Προγράμματος	Program Level
Εποχή Εκπαίδευσης	Epoch Training
Εσωτερικός Κόμβος	Nonleaf Node
Εύκαμπτη Μεθοδολογία	Agile Methology
Ιδρύμα Ελεύθερου Λογισμικού	Free Software Foundation

Ελληνικά	Αγγλικά
Κανόνες Συσχέτισης	Association Rules
Κανονικές Εκφράσεις	Regular Expressions
Κατάλοιπο	Residual
Κατάταξη Ετοιμότητας Ανοικτής Εργασίας	Open Business Readiness Rating
Κεντρομόλος Σύζευξη	Afferent Coupling- Ca
Κέρδος της Πληροφορίας	Information Gain
Κοντινότερου Γείτονα	Nearest Neighbor
Κύκλος Ζωής Ανάπτυξης Πληροφοριακών Συστημάτων	System Development Life Cycle
Κυκλωματική Πολυπλοκότητα	Cyclomatic Complexity
Κύρτωση	Kurtosis
Λογιστική Παλινδρόμηση	Logistic Regression
Λόγος Πιθανοτήτων	Odds Ratio
Μέγεθος Λεξιλογίου	Vocabulary Size
Μέγιστη πιθανοφάνεια	Maximum Likehood
Μέθοδος της Πίσω Διάδοσης του Λάθους	Error Back Propagation
Μέσο Απόλυτο Σφάλμα	Mean Absolute Error
Μέτρηση Πρόσβασης Δεδομένων	Data Access Metric - DAM
Μετρικές Ροής Πληροφορίας	Information Flow Metrics
Μέτρο της Λειτουργικής Αφαιρετικότητας	Measure of Functional Abstraction - MFA
Μέτρο της Συσσωμάτωσης	Measure of Aggregation - MOA
Μη Σχολιασμένος Αριθμός Γραμμών	Non Commented Lines of Code - NCLOC
Μήκος Προγράμματος	Program Length
Μηχανές Διανυσμάτων Υποστήριξης	Support Vector Machines
Μοντέλο Ωριμότητας Ανοικτού Κώδικα	Open Source Maturity Model
Μπεϋσιανοί Ταξινομητές	Bayesian Classifiers
Όγκος Προγράμματος	Program Volume
Οκνηρές Μέθοδοι	Lazy Methods
Ολοκληρωμένο Περιβάλλον Ανάπτυξης	Integrated Development Environment
Ομοσκέδαση	Homoscedacity

Ελληνικά	Αγγλικά
Ορθότητα	Correctness
Παλινδρόμηση μέσω ταξινόμησης	Regression via Classification
Παράγοντας Απόκρυψης Μεθόδου	Method Hiding Factor - MHF
Παράγοντας Απόκρυψης Μεταβλητών	Attribute Hiding Factor- AHF
Παράγοντας Εμπιστοσύνης	Confidence Factor
Παράγοντας Πολυμορφισμού	Polymorphism Factor - PF
Παράγοντας Σύζευξης	Coupling Factor - CF
Παράγοντας της Κληρονομικότητας Μεθόδου	Method Inheritance Factor - MIF
Παράγοντας της Κληρονομικότητας Μεταβλητών	Attribute Inheritance Factor - AIF
Παράγοντες Ποιότητας	Quality Factors
Παραγωγική Έκδοση (Προγράμματος)	Production Release
Παραδοτέος Αριθμός Εντολών	Delivered Source Instructions - DSI
Πίνακας Σύγχυσης	Confusion Matrix
Πληρότητα	Completeness
Πληροφορία Διαχωρισμού	Split Information
Πλήρως Συνδεδεμένο	Fully Connected
Ποιότητας Λογισμικού	Software Quality
Πολλαπλή Γραμμική Παλινδρόμηση	Multivariate Linear Regression
Πολυεπίπεδα Τεχνητά Νευρωνικά Δίκτυα	Multilayer Neural Networks
Πολυπλοκότητα Ροής Πληροφορίας	Information Flow Complexity - IFC
Ποσοστό Σωστών Θετικών Προβλέψεων	True Positive Rate - TP Rate
Πρόγραμμα Διόρθωσης Κειμένου	Text Editor
Προσαρμοστική Ώθηση	AdaBoost
Πρόσθετα	Plug-ins
Προσπάθεια Υλοποίησης	Effort to Implement
Πρωταρχικές Χρήσεις	Primary Uses
Πρωτοβουλία Ανοικτού Κώδικα	Open Source Initiative
Πρωτογενείς Κατασκευές	Primitive Constructs
Ρίζα του Δέντρου	Tree Root

Ελληνικά	Αγγλικά
Ρυθμός Μάθησης	Learning Rate
Σακουλιάσμα	Bagging
Σταθερά Ορμής	Momentum
Σταθμισμένες Μέθοδοι ανά Κλάση	Weighted Methods per Class - WMC
Σύζευξη Μεταξύ Κλάσεων	Coupling Between Objects - CBO
Συλλογή Μετρικών MOOD	Metrics for Object Oriented Design - MOOD
Συλλογή Μετρικών QMOOD	Quality Model for Object Oriented Design
Συλλογή Μετρικών CK	Chidamber and Kemerer Suite
Συνάρτηση Ενεργοποίησης	Activation Function
Συνεκτικότητα	Cohesion
Συνεκτικότητα Μεταξύ των Μεθόδων της Κλάσης	Cohesion Among Methods of Class - CAM)
Συντελεστής Προσδιορισμού	Goodness of Fit Coefficient
Συστατικά (Προγράμματος)	Components
Σύστημα Ανίσχευσης Σφαλμάτων	Bug Tracking System
Σύστημα Διαχείρισης Πηγαίου Κώδικα	Version Control Systems
Σχολιασμένος Αριθμός Γραμμών	Commented Lines of Code - CLOC
Σωστές αρνητικές προβλέψεις	True Negatives Rate - TN rate
Τάση για Σφάλματα	Fault Proneness
Τεχνητή Νοημοσύνη	Artificial Intelligence
Τεχνικές Εξισορρόπησης	Balancing Techniques
Τεχνολογία Λογισμικού	Software Engineering
Τυχαία Δάση	Random Forests
Φυγόκεντρη Σύζευξη	Efferent Coupling - Ce
Φύλλο (Δέντρου)	Terminal Node
Χρόνος Υλοποίησης	Time to Implement

Ζ.2 Από Αγγλικά σε Ελληνικά

Αγγλικά	Ελληνικά
Activation Function	Συνάρτηση Ενεργοποίησης
AdaBoost	Προσαρμοστική Ώθηση
Afferent Coupling- Ca	Κεντρομόλος Σύζευξη
Agile Methology	Εύκαμπτη Μεθοδολογία
Artificial Intelligence	Τεχνητή Νοημοσύνη
Association Rules	Κανόνες Συσχέτισης
Attribute Hiding Factor- AHF	Παράγοντας Απόκρυψης Μεταβλητών
Attribute Inheritance Factor - AIF	Παράγοντας της Κληρονομικότητας Μεταβλητών
Bagging	Σακουλιάσμα
Balancing Techniques	Τεχνικές Εξισορρόπησης
Bayesian Classifiers	Μπεϋσιανοί Ταξινομητές
Bit	Δυαδικό Ψηφίο
Branch	Διακλάδωση
Bug Tracking System	Σύστημα Ανίσχευσης Σφαλμάτων
Chidamber and Kemerer Suite	Συλλογή Μετρικών CK
Cohesion	Συνεκτικότητα
Cohesion Among Methods of Class - CAM)	Συνεκτικότητα Μεταξύ των Μεθόδων της Κλάσης
Commented Lines of Code - CLOC	Σχολιασμένος Αριθμός Γραμμών
Completeness	Πληρότητα
Components	Συστατικά (Προγράμματος)
Conditional Probability	Δεσμευμένη Πιθανότητας
Confidence Factor	Παράγοντας Εμπιστοσύνης
Confidence Interval	Διάστημα Εμπιστοσύνης
Confusion Matrix	Πίνακας Σύγχυσης
Correctness	Ορθότητα
Coupling Between Objects - CBO	Σύζευξη Μεταξύ Κλάσεων
Coupling Factor - CF	Παράγοντας Σύζευξης
Cross Validation	Διασταυρωμένη επικύρωση

Αγγλικά	Ελληνικά
Cyclomatic Complexity	Κυκλωματική Πολυπλοκότητα
Data Access Metric - DAM	Μέτρηση Πρόσβασης Δεδομένων
Decision Tree	Δέντρο Απόφασης
Delivered Source Instructions - DSI	Παραδοτέος Αριθμός Εντολών
Depth of Inheritance Tree - DIT	Βάθος Δέντρου Κληρονομικότητας
Development Release	Έκδοση Προγραμματιστή
Difficulty Level	Επίπεδο Δυσκολίας
Discriminant Analysis	Διακρίνουσα Ανάλυση
Distance - D	Απόσταση
Efferent Coupling - Ce	Φυγόκεντρη Σύζευξη
Effort to Implement	Προσπάθεια Υλοποίησης
Entropy	Εντροπία
Epoch Training	Εποχή Εκπαίδευσης
Error Back Propagation	Μέθοδος της Πίσω Διάδοσης του Λάθους
Executable Statements - ES	Εκτελέσιμες Εντολές
Expected Information	Αναμενόμενη Πληροφορία
Explanatory Variable	Ανεξάρτητη Μεταβλητή
Fault Proneness	Τάση για Σφάλματα
Feed Forward	Εμπρόσθια Τροφοδότηση
F-measure	Αρμονικός Διαιρέτης
Forward Stepwise Selection	Επιλογή Μεταβλητών Προς τα Εμπρός
Free Rider	Ελεύθερου Επιβάτη
Free Software	Ελεύθερου Λογισμικού
Free Software Foundation	Ιδρύμα Ελεύθερου Λογισμικού
Fully Connected	Πλήρως Συνδεδεμένο
Genetic Programming	Γενετικός προγραμματισμός
Goodness of Fit Coefficient	Συντελεστής Προσδιορισμού
Homoscedacity	Ομοσκέδαση
Information Flow Complexity - IFC	Πολυπλοκότητα Ροής Πληροφορίας
Information Flow Metrics	Μετρικές Ροής Πληροφορίας

Αγγλικά	Ελληνικά
Information Gain	Κέρδος της Πληροφορίας
Instability - I	Αστάθεια
Integrated Development Environment	Ολοκληρωμένο Περιβάλλον Ανάπτυξης
Intermediate Constructs	Ενδιάμεσες Κατασκευές
Kurtosis	Κύρτωση
Lack of Cohesion Metric - LCOM	Έλλειψη Συνεκτικότητας των Μεθόδων
Lazy Methods	Οκνηρές Μέθοδοι
Learning Rate	Ρυθμός Μάθησης
Least Squares	Ελαχίστων Τετραγώνων
Lines of Code - LOC	Αριθμός Γραμμών Κώδικα
Log File	Αρχείο Ιστορικού
Logistic Regression	Λογιστική Παλινδρόμηση
Maintainability Index	Δείκτης Συντηρησιμότητας
Maximum Likehood	Μέγιστη πιθανοφάνεια
Mean Absolute Error	Μέσο Απόλυτο Σφάλμα
Measure of Aggregation - MOA	Μέτρο της Συσσωμάτωσης
Measure of Functional Abstraction - MFA	Μέτρο της Λειτουργικής Αφαιρετικότητας
Method Hiding Factor - MHF	Παράγοντας Απόκρυψης Μεθόδου
Method Inheritance Factor - MIF	Παράγοντας της Κληρονομικότητας Μεθόδου
Metrics for Object Oriented Design - MOOD	Συλλογή Μετρικών MOOD
Momentum	Σταθερά Ορμής
Multilayer Neural Networks	Πολυεπίπεδα Τεχνητά Νευρωνικά Δίκτυα
Multivariate Linear Regression	Πολλαπλή Γραμμική Παλινδρόμηση
Nearest Neighbor	Κοντινότερου Γείτονα
Non Commented Lines of Code - NCLOC	Μη Σχολιασμένος Αριθμός Γραμμών
Nonleaf Node	Εσωτερικός Κόμβος
Number of Children - NOC	Αριθμός Απογόνων
Number of Class Methods - NCM	Αριθμός Μεθόδων Κλάσης
Number of Class Variables - NCV	Αριθμός Μεταβλητών Κλάσης
Number of Classes - NC	Αριθμός Κλάσεων

Αγγλικά	Ελληνικά
Number of Delivered Bugs	Αριθμός Παραδομένων Λαθών
Number of Operations Added - NOA	Αριθμός Προστιθέμενων Μεθόδων
Number of Operations Overridden - NOO	Αριθμός Επικαλυπτόμενων Μεθόδων
Number of Public Methods - NPM	Αριθμός Δημοσίων Μεθόδων
Number of Public Methods - NPM	Αριθμός Δημόσιων Μεθόδων
Number of Public Variables - NPV	Αριθμός Δημόσιων Μεταβλητών
Object Oriented Analysis and Design - OOAD	Αντικειμενοτρεφή ανάλυση και σχεδίαση
Odds Ratio	Λόγος Πιθανοτήτων
Open Business Readiness Rating	Κατάταξη Ετοιμότητας Ανοικτής Εργασίας
Open Source Initiative	Πρωτοβουλία Ανοικτού Κώδικα
Open Source Maturity Model	Μοντέλο Ωριμότητας Ανοικτού Κώδικα
Ordinal Evaluation	Διατάξιμη Αποτίμηση
Overriding	Επικάλυψη
Peer Review	Αναθεώρηση Ομότιμων
Plug-ins	Πρόσθετα
Polymorphism Factor - PF	Παράγοντας Πολυμορφισμού
Precision	Ακρίβεια
Pre-Commit Test	Δοκιμή πριν την Υποβολή του Κώδικα
Primary Uses	Πρωταρχικές Χρήσεις
Primitive Constructs	Πρωτογενείς Κατασκευές
Principal Component Analysis	Ανάλυση Κυρίων Συνιστωσών
Production Release	Παραγωγική Έκδοση (Προγράμματος)
Program Length	Μήκος Προγράμματος
Program Level	Επίπεδο Προγράμματος
Program Volume	Όγκος Προγράμματος
Qualification and Selection of Open Source Software	Αξιολόγηση και Επιλογή Λογισμικού Ανοικτού Κώδικα
Quality Factors	Παράγοντες Ποιότητας
Quality Model for Object Oriented Design	Συλλογή Μετρικών QMOOD
Radial Basis Function	Ακτινική Βάση Συνάρτησης
Random Forests	Τυχαία Δάση

Αγγλικά	Ελληνικά
Recall	Ανάκληση
Refactoring	Αναδόμηση
Regression via Classification	Παλινδρόμηση μέσω ταξινόμησης
Regular Expressions	Κανονικές Εκφράσεις
Residual	Κατάλοιπο
Responce for a Class - RFC	Απόκριση για μια Κλάση
Response Variable	Εξαρτημένη Μεταβλητή
Review	Αναθεώρηση
Revision	Αναθεώρηση (Πηγαίου Κώδικα)
Skewness	Ασυμμετρία
Software Engineering	Τεχνολογία Λογισμικού
Software Quality	Ποιότητας Λογισμικού
Specialization Index - SI	Δείκτης Εξειδίκευσης
Split Information	Πληροφορία Διαχωρισμού
Support Vector Machines	Μηχανές Διανυσμάτων Υποστήριξης
System Development Life Cycle	Κύκλος Ζωής Ανάπτυξης Πληροφοριακών Συστημάτων
Terminal Node	Φύλλο (Δέντρου)
Testing Set	Δεδομένα Ελέγχου
Text Editor	Πρόγραμμα Διόρθωσης Κειμένου
Time to Implement	Χρόνος Υλοποίησης
Training Set	Δεδομένα Εκπαίδευσης
Tree Root	Ρίζα του Δέντρου
True Negatives Rate - TN rate	Σωστές αρνητικές προβλέψεις
True Positive Rate - TP Rate	Ποσοστό Σωστών Θετικών Προβλέψεων
Univariate Linear Regression	Απλή Γραμμική Παλινδρόμηση
Version Control Systems	Σύστημα Διαχείρισης Πηγαίου Κώδικα
Vocabulary Size	Μέγεθος Λεξιλογίου
Web Server	Εξυπηρετητής Παγκόσμιου Ιστού
Weighted Methods per Class - WMC	Σταθμισμένες Μέθοδοι ανά Κλάση