Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Μεταπτυχιακό Πρόγραμμα Σπουδών

Ασφάλεια Υπολογιστών και Δικτύων

Μεταπτυχιακή Διατριβή

Εντοπισμός Εσωτερικών Απειλών με Χρήση Τεχνητής Νοημοσύνης σε Δεδομένα Κοινωνικής Δικτύωσης

Παναγιώτης Κυριάκου

Επιβλέπων Καθηγητής
Σταύρος Σιαηλής

Δεκέμβριος 2019

Ανοικτό Πανεπιστήμιο Κύπρου

Σχολή Θετικών και Εφαρμοσμένων Επιστημών

Μεταπτυχιακό Πρόγραμμα Σπουδών

Ασφάλεια Υπολογιστών και Δικτύων

Μεταπτυχιακή Διατριβή

Εντοπισμός Εσωτερικών Απειλών με Χρήση Τεχνητής Νοημοσύνης σε Δεδομένα Κοινωνικής Δικτύωσης

Παναγιώτης Κυριάκου

Επιβλέπων Καθηγητής
Σταύρος Σιαηλής

Η παρούσα μεταπτυχιακή διατριβή υποβλήθηκε προς μερική εκπλήρωση των απαιτήσεων για απόκτηση μεταπτυχιακού τίτλου σπουδών στην Ασφάλεια Υπολογιστών και Δικτύων από τη Σχολή Θετικών και Εφαρμοσμένων Επιστημών του Ανοικτού Πανεπιστημίου Κύπρου.

Δεκέμβριος 2019

ΛΕΥΚΗ ΣΕΛΙΔΑ

Περίληψη

Ο κίνδυνος της εσωτερικής απειλής είναι μια συνεχής κρυφή απειλή που επισκιάζει την ασφάλεια οργανισμών και Κρατών, οι οποίοι καλούνται να προστατεύσουν τους εαυτούς τους από κάτι που δεν μπορούν να δουν ή να υπολογίσουν. Αν και με την ανάπτυξη της τεχνολογίας έχουν δημιουργηθεί πολλές μέθοδοι αναγνώρισης επιθέσεων εκ του έσω, αυτοί οι μηχανισμοί δεν παύουν να είναι αντιδραστικοί και όχι προληπτικοί. Αυτό έχει ως συνέπεια τέτοιου είδους επιθέσεις να γίνονται αντιληπτές μόνο μετά την υλοποίηση της απειλής, όπου και η ζημιά έχει ήδη συμβεί. Έρευνες έχουν δείξει ότι υπάρχουν πολλές προσεγγίσεις για αντιμετώπιση του περίπλοκου αυτού προβλήματος, και μια εξ' αυτών είναι η μελέτη του συναισθήματος το οποίο εκφράζουμε όλοι μας με κάθε μας συναναστροφή στις ιστοσελίδες κοινωνικής δικτύωσης. Αυτή η μεταπτυχιακή διατριβή αναλαμβάνει να εξερευνήσει το συγκεκριμένο πεδίο, δημιουργώντας ταξινομητές μηχανικής μάθησης οι οποίοι μπορούν να εντοπίσουν το συναίσθημα το οποίο εκφράζεται από μηνύματα του κοινωνικού δικτύου Twitter. Για την εκτέλεση αυτού του έργου, έγινε συγκέντρωση συλλογών δεδομένων με ετικέτες συναισθημάτων, για τη δημιουργία μιας ενιαίας συλλογής δεδομένων η οποία χρησιμοποιήθηκε κατά την εκπαίδευση των μοντέλων μηχανικής μάθησης. Εκτός από την δημιουργία των ταξινομητών συναισθημάτων, αυτή η έρευνα θέτει μια θεμέλια λίθο στη προώθηση δημιουργίας ενός ολοκληρωμένου συστήματος εντόπισης εσωτερικής απειλής, δημιουργώντας μια βάση δεδομένων η οποία μπορεί να χρησιμοποιηθεί ως βάση ανάπτυξης και μελέτης.

Summary

The risk of insider threat is a hidden trap that is chipping away at the security of nations and organizations who have no means to detect it. Even though technology has provided us with great strides in functionality and newer flashy systems, these systems tend to face the insider threat in a reactive rather than a proactive manner. As a direct effect of this, insider attacks are more often than not discovered after the damage has been done. Studies have shown that there are numerous ways to approach insider threat detection. One of those is affect or emotion analysis, which focuses on the emotions that are emitted during our interactions on the internet and specifically social media sites. This thesis undertakes the goal of studying this field and developing document emotion classifiers based on machine learning and social media messages exchanged on the social media site Twitter.com. Focus is provided to gathering and creating an aggregated dataset with tagged emotional tweets that will be used for training the machine learning classifiers. In addition to developing the machine learning classifiers, this this also provides a relational database schema that can be used for the creation of a complete system for insider threat detection.

Ευχαριστίες

Θα ήθελα να ευχαριστήσω τον Καθηγητή μου, Δρ. Σταύρο Σιαηλή, που μου έδωσε την δυνατότητα να εργαστώ στο ενδιαφέρον αυτό θέμα, όπως και όλους όσους στάθηκαν δίπλα μου σε αυτό το ταξίδι της ζωής μου.

Πάνω από όλα θα ήθελα να ευχαριστήσω την αγάπη της ζωής μου, τη γυναίκα μου, Γιώτα, που στάθηκε δίπλα μου με αγάπη, υπομονή και επιμονή και μου έδωσε δύναμη όταν εγώ δεν είχα, ώστε να φέρω εις πέρας αυτό το δύσκολο έργο.

Σας ευχαριστώ όλους από καρδιάς!

Με εκτίμηση,
Παναγιώτης Κυριάκου

Περιεχόμενα

Κεφάλαιο 1
Εισαγωγή

Η άνθιση της πληροφορικής και του διαδικτύου έχει ωθήσει στην ψηφιακή αναβάθμιση του τρόπου λειτουργίας πολλών οργανισμών. Δυστυχώς, με αυτή την ανάπτυξη εμφανίζονται κενά ασφαλείας τα οποία κακόβουλα άτομα μπορούν να εκμεταλλευτούν για να προκαλέσουν ζημιά. Ως αποτέλεσμα, πολλοί οργανισμοί, αλλά και κράτη έχουν πέσει θύμα τέτοιων επιθέσεων, με αυξητικές τάσεις κάθε χρόνο.

Αυτές οι επιθέσεις δύναται να προέρχονται τόσο από το εξωτερικό όσο και το εσωτερικό περιβάλλον κάθε οργανισμού. Το National Infrastructure Advisory Council χαρακτηρίζει την εσωτερική απειλή ως την «απειλή η οποία προέρχεται από ένα ή περισσότερα άτομα τα οποία έχουν εσωτερική γνώση της λειτουργίας ενός οργανισμού, δίδοντας τους τη δυνατότητα εκμετάλλευσης αδυναμιών σε συστήματα, πολιτικές, διαδικασίες και χώρους αυτού, με απώτερο σκοπό την πρόκληση ζημιάς ή την επίτευξη ανταγωνιστικού πλεονεκτήματος» (Noonan, Archuleta, 2008).

Όπως αναφέρουν οι Brdiczka et al. (2012), υπάρχουν αρκετά παραδείγματα υποδειγματικών υπαλλήλων οι οποίοι χωρίς καμία ουσιαστική ή αντιληπτή προειδοποίηση στράφηκαν εναντίων του οργανισμού που τους εργοδοτούσε δίνοντας εμπιστευτικές πληροφορίες σε αντίπαλους οργανισμούς. Για παράδειγμα, σύμφωνα με τα αποτελέσματα μιας έρευνας, το 28% των ερωτηθέντων είχε αναφέρει ότι θα αξιοποιούσε ευαίσθητα εταιρικά δεδομένα για τη διαπραγμάτευση ενός καλύτερου συμβολαίου με μια δεύτερη εταιρεία, στην περίπτωση της αποδέσμευσής τους απ' την πρώτη (Cyber Ark, 2012). Αντίστοιχο ποσοστό ανταπόκρισης ανέφερε ότι είχαν την πρόθεση να χρησιμοποιήσουν τα συγκεκριμένα δεδομένα, ως εργαλείο στη νέα τους δουλειά. Εντυπωσιακή είναι και η διάθεση των ερωτηθέντων για λήψη τέτοιων ευαίσθητων δεδομένων απλά και μόνο γιατί μπορεί να τους είναι χρήσιμα στο μέλλον, μιας και το 56% των ερωτηθέντων απάντησε θετικά στη συγκεκριμένη ερώτηση.

Παρόλα αυτά αρκετές εταιρείες αδυνατούν ακόμη να αντιληφθούν τη σημαντικότητα της απειλής εκ των έσω. Σύμφωνα με την έκθεση αναφοράς των Cybersecurity Insiders και των Crowd Research Partners (2017) το 90% των υπό μελέτη οργανισμών είχαν εκφράσει αδυναμία έναντι εσωτερικών επιθέσεων, όμως μόνο το 56% εξ αυτών είχε απαντήσει ότι οι υπάλληλοι τους ενδεχομένως να αποτελούν τη μεγαλύτερη απειλή εσωτερικών επιθέσεων. Σύμφωνα με έρευνα της Verizon, για το έτος 2019, το 34% των επιθέσεων οφειλόταν σε δράστες εντός του οργανισμού, ενώ το 2% οφειλόταν σε συνεργάτες του οργανισμού (Verizon, 2019).

Αξίζει επίσης να αναφερθεί ότι πολλοί οργανισμοί δίνουν συχνά μεγαλύτερη έμφαση στην ενίσχυση του εσωτερικού συστήματος ασφαλείας τους, μέσω του καθορισμού αυστηρών πολιτικών και διαδικασιών ασφαλείας, θεωρώντας πως η θέσπιση αυτών θα αποτρέψει ουσιαστικά την όποια επίθεση (Jiang et al., 2018). Παρόλα αυτά, όπως τονίζει κι ο Colwill (2009) η ασφάλεια πληροφοριών δεν πρέπει να στηρίζεται μόνο σε τεχνολογικές λύσεις, μιας κι αυτές δεν αποτελούν πανάκεια. Ο ίδιος, έπειτα απ' τη διεξαγωγή της έρευνάς του, βρήκε ότι το 99% των υπό μελέτη Βρετανικών εταιρειών κάνουν λήψη αντιγράφων ασφαλείας των κρίσιμων συστημάτων και δεδομένων τους, το 98% έχουν λογισμικά ανίχνευσης spyware, το 97% προστατεύουν τις ιστοσελίδες τους με firewall και το 94% κάνουν κρυπτογράφηση της επικοινωνίας τους μέσω του ασύρματου δικτύου. Όπως αναφέρει επίσης ο ίδιος, παρότι το 70% των επιθέσεων που είχαν οι εταιρείες αυτές ανήκαν σε εσωτερικές απειλές, οι μηχανισμοί ασφαλείας και παρακολούθησης των εταιρειών στόχευαν στον εντοπισμό εξωτερικών απειλών. Σύμφωνα με τους Jiang et al. (2018) οι οργανισμοί οι οποίοι αγνοούν τα συγκεκριμένα κενά ασφαλείας είναι καταδικασμένοι να αποτύχουν.

Σε αντίθεση με τις εξωτερικές απειλές, μια επίθεση εκ των έσω είναι πολύ πιο δύσκολο να ανιχνευθεί. Ο Colwill (2009) τονίζει τη σημασία ουσιαστικής εκπαίδευσης των υπαλλήλων μιας εταιρείας και τη θέσπιση διαφόρων πολιτικών που θα δρουν ως μηχανισμοί μετριασμού αυτών των ρίσκων. Κάποιες απ' τις πιο εναλλακτικές προσεγγίσεις του, αφορούν για παράδειγμα τη σημασία θέσπισης ενός συστήματος εμπιστοσύνης όπου οι υπάλληλοι θα μπορούν να αναφέρουν με ασφάλεια μη-κανονικές συμπεριφορές συναδέλφων. Επίσης, προτείνει μεταξύ άλλων τη δημιουργία ενός μηχανισμού εκτόνωσης συναισθημάτων θρήνου, και την εκπαίδευση του διευθυντικού

προσωπικού για τον εντοπισμό ατόμων με τάσεις αρνητικής λεκτικής έκφρασης έναντι στην εργασία τους και στην εταιρεία. Με τις συγκεκριμένες εισηγήσεις του δηλαδή, δίνει ιδιαίτερη έμφαση στην παρακολούθηση της συναισθηματικής κατάστασης των υπαλλήλων μιας εταιρείας, μιας και κατά τον ίδιο η συγκεκριμένη πρακτική μπορεί να αποτρέψει πολλές ενδεχόμενες εσωτερικές επιθέσεις.

Παρόλα αυτά, ακόμα και αν ο οργανισμός έχει εκτελέσει εκπαίδευση των υπαλλήλων και συνεργατών του, έχει καθορίσει πολιτικές και διαδικασίες ασφαλείας, είναι πολύ εύκολο να υλοποιηθεί κάποια εσωτερική απειλή, είτε λόγω λάθους, άγνοιας, αδιαφορίας, απειλής/εκβιασμού από τρίτους, ή δολιότητας ενός υπαλλήλου ή συνεργάτη. Σε αντίθεση με δράστες του εξωτερικού περιβάλλοντος, τα άτομα αυτά έχουν εσωτερική γνώση του τρόπου λειτουργίας του οργανισμού και απευθείας πρόσβαση στα συστήματα του, κάτι που ενδεχομένως να διευρύνει και το εύρος απώλειας/ζημιάς που μια τέτοιας επίθεσης μπορεί να προκαλέσει.

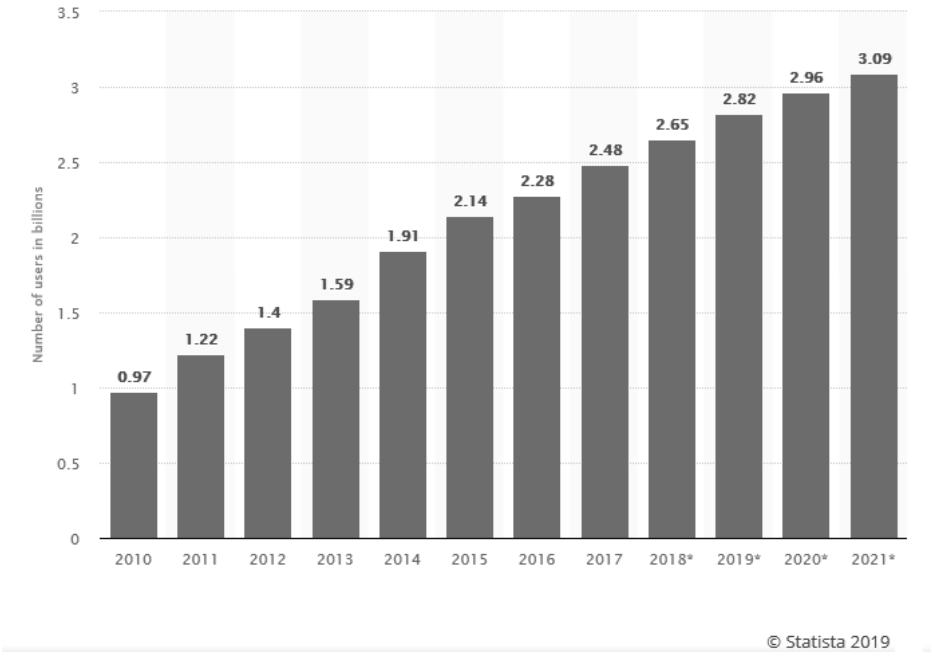
Υπάρχουν, βέβαια, συστήματα ασφαλείας τα οποία μπορούν να παρακολουθούν τη δραστηριότητα του χρήστη, να εντοπίζουν ιδιομορφίες και να ενημερώνουν ή να παίρνουν μέτρα αποτροπής. Ένα τέτοιο παράδειγμα είναι τα συστήματα αποτροπής απώλειας δεδομένων (Data Loss Prevention), συστήματα διαχείρισης πληροφοριών και συμβάντων ασφαλείας (Security Information and Event Monitoring) και συστήματα αναγνώρισης/πρόληψης εισβολής (Intrusion Detection System/Intrusion Prevention System).

Όλα αυτά τα μέτρα μετριασμού της εσωτερικής απειλής, αναγνωρίζουν ή/και αποτρέπουν την εκτέλεση κάποιας επίθεσης ενόσω αυτή εκτελείται, χωρίς να μετριάζουν το ρίσκο πρόκλησης πραγματικής ζημιάς, ακόμη και όταν η επίθεση έχει αποτραπεί. Επιπλέον, στην περίπτωση όπου η εσωτερική απειλή είναι δόλια και προέρχεται από άτομο-γνώστη των πιο πάνω συστημάτων (π.χ. διαχειριστής συστημάτων, άτομο με σπουδές/εμπειρία σχετική με την επιστήμη της Πληροφορικής), υπάρχει η πιθανότητα ο επιτιθέμενος να μεταβάλει την λειτουργία αυτών των συστημάτων για να αποκρύψει τις ενέργειες του, με αποτέλεσμα η επίθεση να μην γίνει αντιληπτή.

Για παράδειγμα, όπως αναφέρουν οι Tan et al. (2019), οι υπάλληλοι μιας εταιρείας έχουν ήδη το πάνω χέρι σε θέματα ασφαλείας, μιας κι έχουν ήδη νόμιμη ή/και προνομιούχα

πρόσβαση στα δεδομένα μιας εταιρείας, καθιστώντας έτσι πιο εύκολο τον εντοπισμό των αδυναμιών της, χωρίς να πρέπει να περάσουν απ' τους ελέγχους που την προστατεύουν από εξωτερικές επιθέσεις. Για το λόγο αυτό οι εσωτερικοί αντίπαλοι (adversarial insiders) μιας εταιρείας έχουν τη δυνατότητα να επιφέρουν και τη μεγαλύτερη ζημιά σε αυτήν, λόγω της πλεονεκτικότητας της θέσης τους. Αν λάβει κανείς υπόψη και τη δυνατότητα αυτών να καλύπτουν τα ίχνη τους έπειτα από τέτοιου είδους επιθέσεις, μπορεί κανείς να αντιληφθεί τη σημαντικότητα τέτοιων απειλών.

Μια σύγχρονη τάση που επικρατεί στο συγκεκριμένο τομέα είναι η αξιοποίηση των μέσων κοινωνικής δικτύωσης ως μέσο ανίχνευσης κακόβουλων τάσεων, για την έγκαιρη αποτροπή τέτοιων επιθέσεων. Η συνεχής ανάγκη του ανθρώπου για επικοινωνία μέσα απ' τα διάφορα μέσα κοινωνικής δικτύωσης καλλιεργεί ένα πρόσφορο έδαφος για τη φιλελεύθερη έκφραση των σκέψεων, συναισθημάτων, ακόμη και των προθέσεων ατόμων με τέτοιου είδους τάσεις. Σύμφωνα με την ιστοσελίδα Statista (2019) φέτος έχουν καταγραφεί περισσότεροι από 2.82 δισεκατομμύρια χρήστες κοινωνικών δικτύων, ανά το παγκόσμιο, με τον αριθμό αυτό να υπολογίζεται ότι θα αγγίξει τα 3.1 δισεκατομμύρια μέχρι το 2021. Συνεπώς, γίνεται άκρως αντιληπτή η άμεση αξιοποίηση αυτού του τεράστιου όγκου πληροφοριών για τη γενικότερη αποτροπή εσωτερικών επιθέσεων, είτε αυτά αφορούν οργανισμούς, ή/και ολόκληρα κράτη.



Διάγραμμα 1. Γραφική παράσταση η οποία δείχνει τον αριθμό χρηστών μέσων κοινωνικής δικτύωσης την τελευταία δεκαετία, καθώς και την εκτίμηση του αριθμού αυτών το 2020-2021. Όπως γίνεται αντιληπτό ο αριθμός αυτός αυξάνεται συνεχώς.

Για παράδειγμα, όπως αναφέρουν οι Glasser και Lindauer (2013), πριν από δέκα περίπου χρόνια ο ταγματάρχης Nidal Hasan άνοιξε πυρά και σκότωσε δεκατρία άτομα στο Κέντρο Ετοιμότητας Στρατιωτών. Έπειτα από έρευνα του διαδικτυακού προφίλ του, διαφάνηκε ότι είχε αναπτύξει στενές συνδέσεις με ακραία στελέχη του Ισλαμικού Κράτους, ενώ είχε αρκετές αναζητήσεις που αφορούσαν αυτοκτονίες με αυτοσχέδιες βόμβες. Τότε είχε δημιουργηθεί έντονη συζήτηση για την παρεμπόδιση τέτοιου είδους επιθέσεων μέσα απ' τον έγκαιρο εντοπισμό των όσων σχεδίαζε μέσα απ' το διαδικτυακό προφίλ του.

Αντίστοιχα, όπως αναφέρθηκε σε μια αναφορά του Ομοσπονδιακού Γραφείου Ερευνών (FBI) η ανάπτυξη ενός κλίματος τοξικότητας στο εργασιακό περιβάλλον μπορεί να γεννήσει αντίστοιχα ζημιογόνα αποτελέσματα οικονομικού (και όχι μόνο) χαρακτήρα για μια εταιρεία (Keeney et al., 2005). Η έκφραση συναισθημάτων δυσαρέσκειας, έως και τάσης εκδικητικότητας υπαλλήλων της εταιρείας στα μέσα κοινωνικής δικτύωσης μπορεί να σημάνει πρόωρα τον κώδωνα του κινδύνου πριν την υλοποίηση εσωτερικής απειλής.

1.2 Στόχος Μεταπτυχιακής Διατριβής

Έχοντας εντοπίσει ένα κενό στην βιβλιογραφία, που αφορά την δημιουργία μοντέλων μηχανικής μάθησης αναγνώρισης συναισθήματος για τον εντοπισμό εσωτερικής απειλής, ασχοληθήκαμε με την εξερεύνηση αυτών. Στόχος μας ήταν η δημιουργία μοντέλων που να επεκτείνουν τον αριθμό συναισθημάτων που μπορούν να εντοπιστούν στην επικοινωνίας χρηστών κοινωνικής δικτύωσης, χρησιμοποιώντας μηνύματα από το Twitter. Ακόμη, προετοιμάσαμε το έδαφος για μελλοντική ανάλυση δημιουργώντας μια βάση δεδομένων βασισμένη στη συλλογή δεδομένων Sentiment 140.

Γενικά, η μεταπτυχιακή αυτή διατριβή έχει ως στόχο της τον καθορισμό μιας θεμέλιας λίθου, που μέσω μελλοντικής εξέλιξης θα οδηγήσει στην δημιουργία ενός ολοκληρωμένου συστήματος αναγνώρισης εσωτερικής απειλής χρησιμοποιώντας πληροφορίες από μέσα κοινωνικής δικτύωσης και τεχνολογίες μηχανικής μάθησης.

1.3 Δομή Μεταπτυχιακής Διατριβής

Η μεταπτυχιακή αυτή διατριβή αποτελείται από 8 μέρη. Στο κεφάλαιο 2, εκτελούμε βιβλιογραφική ανασκόπηση και παρουσιάζουμε προκλήσεις και παραδείγματα που διέπουν το πεδίο που μελετάμε. Στο κεφάλαιο 3 αναφέρουμε την προσέγγιση και τις τεχνολογίες που χρησιμοποιήσαμε. Στη συνέχεια, το κεφάλαιο 4 αναφέρεται στις συλλογές δεδομένων που χρησιμοποιήσαμε και το κεφάλαιο 5 εξηγεί πως τις προετοιμάσαμε για τροφοδότηση σε μοντέλα μηχανικής μάθησης. Η ανάλυση των μοντέλων μηχανικής μάθησης που χρησιμοποιήσαμε γίνεται στο κεφάλαιο 6. Το κεφάλαιο 7 περιέχει τα αποτελέσματα της εκπαίδευσης των μοντέλων μηχανικής μάθησης και στο κεφάλαιο 8 αναφέρουμε τα συμπεράσματα της έρευνας μας και παραθέτουμε μελλοντικό έργο το οποίο θα μπορούσε να βασιστεί στο έργο μας.

Το παράρτημα Α περιέχει κώδικα Python που αφορά την ανάπτυξη που κάναμε ως μέρος αυτής της μεταπτυχιακής διατριβής. Αντιστοίχως, το παράρτημα Β περιγράφει τη δομή της βάσης δεδομένων που δημιουργήσαμε.

Κεφάλαιο 2
Υπάρχουσα Έρευνα

2.1 Η σημασία των μέσων κοινωνικής δικτύωσης

Όπως αναφέρθηκε και στην εισαγωγή, με την άνθηση του διαδικτύου έχουν εμφανιστεί και εδραιωθεί στη ζωή μας τα μέσα κοινωνικής δικτύωσης. Αυτές οι ιστοσελίδες επιτρέπουν στους χρήστες τους να δημιουργήσουν εξατομικευμένα προφίλ, να έρθουν σε επικοινωνία με φίλους, γνωστούς και ξένους, φυσικά πρόσωπα και οργανισμούς και να ανταλλάξουν γραπτά μηνύματα και οπτικοακουστικό περιεχόμενο. Ο άνθρωπος, όντας κοινωνικό ζώο, έχει επιτρέψει στη ζωή του να κατακυριευθεί με αυτά τα μέσα, μοιράζοντας σε αυτά τη ζωή του, τα νιώθω του και τα πιστεύω του.

Λόγω της έμμεσης κοινωνικοποίησης και την ευκολία που προσδίδουν τα συγκεκριμένα μέσα δικτύωσης ως προς την αλλαγή της πραγματικής ταυτότητας του χρήστη, πίσω από ένα ψευδώνυμο ή ακόμη κι ένα ψεύτικο προφίλ, οι χρήστες αφήνονται να εκφραστούν ελεύθερα και δημοκρατικά. Αυτή η πληθώρα δεδομένων που δημοσιεύεται, συχνά χωρίς περιορισμούς ως προς το ποιος γίνεται δέκτης αυτών, δίνει τη δυνατότητα συλλογής πολύτιμων πληροφοριών από ανθρώπους που έχουν χαρακτηριστεί ψυχολογικά ασταθείς. Όπως αναφέρουν οι Park et al. (2018), μια τέτοια ανάλυση αποτελεί πηγή χρυσού για την έρευνα που μελετά θέματα εσωτερικής απειλής, καθώς μας επιτρέπει να παρακολουθήσουμε απευθείας την ακατέργαστη συμπεριφορά του κάθε χρήστη σε μη-επιτηρούμενο περιβάλλον και να εξάγουμε πολύτιμα συμπεράσματα.

Η σύγχρονη τάση που επικρατεί σήμερα όσον αφορά το συγκεκριμένο πεδίο έρευνας, επικεντρώνεται στο συνδυασμό χαρακτηριστικών ψυχολογίας με άλλα υποκειμενικά κριτήρια για τον εντοπισμό κακόβουλων εσωτερικών επιθέσεων (Jiang et al., 2018). Οι συγκεκριμένες προσεγγίσεις στηρίζονται στην εξαγωγή και ανάλυση δεδομένων συγκεκριμένων δεικτών που σχετίζονται με τη ψυχολογία και το συναίσθημα, όπως για παράδειγμα, η ικανοποίηση, η πίεση και τα γενικότερα συναισθήματα των χρηστών.

2.2 Προκλήσεις

Η εξαγωγή κι ανάλυση δεδομένων με βάση τη ψυχολογία και το συναίσθημα ως προσέγγιση στο πρόβλημα δεν είναι και η ευκολότερη στην υλοποίηση, μιας και καλείται να αντιμετωπίσει μια σειρά προκλήσεων. Για παράδειγμα, όπως αναφέρουν οι Jiang et al. (2018), η απόκτηση δεδομένων που αφορούν τα συναισθήματα και τη γενικότερη ψυχολογική κατάσταση ενός χρήστη αποτελεί από μόνη της μια πρόκληση. Αυτό, σε συνδυασμό με το γεγονός ότι τα συγκεκριμένα δεδομένα χαρακτηρίζονται από υποκειμενικότητα, και ότι είναι εύκολο να νοθευτούν ή/και να πλαστογραφηθούν, δυσκολεύει ακόμη περισσότερο την αντιμετώπιση των συγκεκριμένων προκλήσεων.

Για το λόγο αυτό δεν είναι τυχαίο ότι αρκετά προγράμματα εντοπισμού εσωτερικών απειλών στηρίζονται στην ανάλυση των δεδομένων του ιστορικού πλοήγησης διαδικτύου του χρήστη και στο ηλεκτρονικό ταχυδρομείο αυτού (Jiang et al., 2018). Μέσα απ' αυτή την ανάλυση προκύπτει η δημιουργία του ψυχολογικού και συμπεριφορικού προφίλ του χρήστη, με αποτέλεσμα οποιαδήποτε απόκλιση απ' τη συγκεκριμένη σκιαγράφηση να χαρακτηρίζεται ως ενδεχόμενος κίνδυνος. Αξίζει να αναφερθεί όμως ότι η συγκεκριμένη προσέγγιση δε λαμβάνει υπόψη το περιεχόμενο των ιστοσελίδων που επισκέφθηκε ο χρήστης, αλλά και του ηλεκτρονικού ταχυδρομείου του, μιας και στηρίζεται στον εντοπισμό ανωμαλιών στα επικοινωνιακά μοτίβα αυτών.

Μια εκ των μεθόδων επίλυσης του προαναφερόμενου ζητήματος αποτελεί η χρήση του μοντέλου OCEAN των Alahmadi et al. (2015). Το συγκεκριμένο εργαλείο λαμβάνει υπόψη το συσχετισμό του περιεχομένου πλοήγησης με το ψυχολογικό προφίλ του χρήστη, καθώς επίσης και το πως οι πλοηγήσεις στο διαδίκτυο δύναται να αλλοιωθούν με την πάροδο του χρόνου. Με αυτόν τον τρόπο καθίσταται εφικτός ο εντοπισμός εσωτερικών απειλών πριν υλοποιηθούν. Παρόλα αυτά, όπως αναφέρουν οι Jiang et al. (2018), αυτού του είδους τα μοντέλα δε λαμβάνουν υπόψη ακραία περιστατικά έντονης ψυχολογικής μεταβλητότητας, τα οποία δύναται να επιφέρουν ενδεχομένως και το μεγαλύτερο κίνδυνο. Μια άλλη πρόκληση που αξίζει να αναφερθεί είναι αυτή που τονίζουν στην έρευνά τους οι Brdiczka et al. (2012), και αφορά την ανάλυση ενός τεράστιου όγκου δεδομένων από χρήστες μέσα σε σύντομο χρονικό διάστημα.

2.3 Παραδείγματα μέσα απ' την υφιστάμενη βιβλιογραφία

Η έρευνα γύρω απ' το συγκεκριμένο ζήτημα περιλαμβάνει μια πληθώρα διαφορετικών προσεγγίσεων. Κάποιες προσεγγίσεις είναι πιο τεχνολογικές, ενώ άλλες στηρίζονται στην ανάλυση του ψυχολογικού προφίλ και των συναισθημάτων των χρηστών. Κάποια παραδείγματα της πρώτης κατηγορίας αφορούν για παράδειγμα τη συλλογή δεδομένων από το ποντίκι και το πληκτρολόγιο των χρηστών, όπως ανέλυσαν μεταξύ άλλων οι Harilal et. al (2017). Αντίστοιχα, οι Legg et al. (2015) δημιούργησαν ένα εργαλείο για τον εντοπισμό ενδεχόμενης εσωτερικής απειλής μέσα απ' την ανάλυση δεδομένων που αφορούσαν τις ενέργειες του χρήστη, όπως για παράδειγμα τις συνδέσεις στα διάφορα συστήματα και τις συνδέσεις εξωτερικών δίσκων.

Αξίζει να αναφερθεί ότι πολλά εννοιολογικά μοντέλα εντοπισμού εσωτερικών απειλών προσπαθούν να προσδιορίσουν την ικανότητα (Capability) του χρήστη, το κίνητρο (Motivation) αυτού και την ευκαιρία (Opportunity) που ενδεχομένως μπορεί να του δοθεί. Όπως αναφέρει ο Schultz (2002), τα συγκεκριμένα μοντέλα προσπαθούν να εντοπίσουν τους ενδεχόμενους κινδύνους λαμβάνοντας υπόψη:
α) Τις ικανότητες (γνώσεις και εμπειρίες) ενός χρήστη που του επιτρέπουν να προβεί σε τέτοιου είδους ενέργειες.
β) Το κίνητρο πίσω απ' τις ενέργειες του το οποίο μπορεί να έχει τόσο εσωτερικά όσο και εξωτερικά ερεθίσματα.
γ) Τις ευκαιρίες που του δίνονται, ανάλογα με το πόσο εύκολο του είναι να υλοποιήσει μια τέτοια απειλή. Ένα άτομο δηλαδή που έχει πρόσβαση σε πληθώρα ευαίσθητων δεδομένων είναι λογικό να έχει και τις καλύτερες ευκαιρίες να τα αξιοποιήσει με κακόβουλο τρόπο.

Αυτή τη στιγμή υπάρχουν στη βιβλιογραφία πολλά μοντέλα που επικεντρώνονται στις ευκαιρίες και τις ικανότητες των χρηστών. Λίγα μοντέλα ασχολούνται με το κίνητρο αυτών, λόγω του ότι το κίνητρο καθορίζεται συχνά απ' τα στοιχεία της προσωπικότητας του χρήστη, και τη ψυχοσύνθεση αυτού, δεδομένα τα οποία για να συλλεχθούν απαιτούν ψυχολογική ανάλυση. Μιας και η άμεση πραγματοποίηση ψυχολογικών αναλύσεων είναι αθέμιτη σε ένα εργασιακό περιβάλλον, αν λάβει κανείς τους νομικούς και ηθικούς

περιορισμούς πίσω από μια τέτοια πρακτική γίνεται αντιληπτή η πρόκληση που καλούνται να απαντήσουν τα μοντέλα ανάλυσης κινήτρου.

Λαμβάνοντας υπόψη τους πιο πάνω περιορισμούς, ορισμένοι ερευνητές χρησιμοποιούν περιβάλλοντα προσομοίωσης. Ένα καλό παράδειγμα είναι η έρευνα των Ho et. al (2016), οι οποίοι προχώρησαν στην ανάπτυξη ενός διαδικτυακού παιχνιδιού, με το όνομα "Collabo" στην πλατφόρμα της Google+ Hangout. Οι παίκτες που συμμετείχαν στο συγκεκριμένο παιχνίδι είχαν χωριστεί με τυχαίο τρόπο σε εικονικές ομάδες. Κάθε εικονική ομάδα έφερε ένα ρόλο, ανάλογα με το επίπεδο επηρεασμού τους απ' τους ερευνητές. Οι παίκτες δηλαδή είτε θα ανήκαν στις ομάδες ελέγχου (controls), είτε στις ομάδες με κακόβουλες τάσεις (bait). Η ταυτότητα των παικτών ήταν καλυμμένη με τη χρήση ψευδωνύμων. Στόχος των παικτών ήταν η γρήγορη επίλυση επτά προβλημάτων λογικής μέσα σε 42 λεπτά. Κάθε ομάδα έφερε το δικό της αρχηγό, ο οποίος είχε την ευθύνη να επικοινωνεί στον ερευνητή που συμμετείχε ως παρατηρητής (game master) την πρόοδο και τα προβλήματα που αντιμετώπιζε η ομάδα του. Η νικητήρια ομάδα με τους καλύτερους χρόνους σε μια περίοδο 5 ημερών συνεχούς παιχνιδιού κέρδιζε ψηφιακά νομίσματα τα οποία εξαργυρώνονταν με κάρτες της Amazon.

Οι ερευνητές προσπάθησαν να δελεάσουν ορισμένους αρχηγούς των ομάδων με την προσφορά 200 επιπλέον νομισμάτων για να κρατήσουν την ταυτότητα τους κρυφή. Το συγκεκριμένο βραβείο οι αρχηγοί ήταν υποχρεωμένοι να το κατανείμουν ίσα στις ομάδες τους στην περίπτωση που θα κέρδιζαν, ενώ μπορούσαν να το κρατήσουν για τον εαυτό τους αν έχαναν. Οι αρχηγοί που επέλεγαν να αλλοιώσουν το αποτέλεσμα κοστίζοντας την ήττα στην ομάδα τους για να επωφεληθούν του βραβείου είχαν αρκετές αλλαγές και στη γενικότερη συμπεριφορά τους. Μέσα απ' τη γλωσσική ανάλυση της επικοινωνίας των δύο κατηγοριών ομάδων, οι ερευνητές κατάφεραν να εντοπίσουν αρκετά δείγματα αρνητικότητας, χρήσης λέξεων οι οποίες συνδέονταν με συναισθήματα, καθώς και φράσεων που συνδέονταν με τις γνωστικές διαδικασίες και με την αποκάλυψη παραπλανητικών πρακτικών.

Άλλες έρευνες, στις οποίες θα δοθεί και μεγαλύτερη έμφαση, χρησιμοποιούν συλλογές δεδομένων τις οποίες κι αναλύουν με διαφορετικές προσεγγίσεις. Μια εκ των πιο πολυσυζητημένων ερευνών είναι αυτή των Brdiczka et al (2012) οι οποίοι προσέγγισαν το συγκεκριμένο πρόβλημα μέσω της ανάπτυξης μιας μεθόδου η οποία συνδύαζε τη

δημιουργία του ψυχολογικού προφίλ του κάθε χρήστη και την ανάλυση της δραστηριότητας του σε κοινωνικά δίκτυα για εξαγωγή ανωμαλιών. Προτάθηκε η χρήση ανάλυση δομής γράφου για εξαγωγή της κανονικής δραστηριότητας των χρηστών, η οποία ακολούθως χρησιμοποιήθηκε ως βάση σύγκρισης. Επιπλέον, εισηγήθηκαν την δημιουργία ψυχολογικού προφίλ μέσω της ανάλυσης της δραστηριότητας του κάθε χρήστη καθώς και την ανάλυση θετικού/αρνητικού συναισθήματος στα μηνύματα που ανταλλάσσει στα μέσα κοινωνικής δικτύωσης.

Η δυναμική της έρευνας τους (Brdiczka et al., 2012) αναδεικνύεται κι απ' την εφαρμογή της μεθόδου τους σε ένα τεράστιο όγκο δεδομένων του διαδικτυακού παιχνιδιού World of Warcraft, τα οποία αφορούσαν δείγματα συμπεριφοράς από πάνω από 350 000 χαρακτήρες, σε μια περίοδο πέραν των 6 μηνών. Το μοντέλο τους στόχευε στον έγκαιρο εντοπισμό των παικτών οι οποίοι θα αποχωρούσαν απ' την ομάδα τους (guilds) και θα στρέφονταν εναντίον αυτής κατά τη διάρκεια του παιχνιδιού. Για τη δημιουργία του ψυχολογικού προφίλ των χρηστών αξιοποίησαν και ανέλυσαν ένα μεγάλο αριθμό πληροφοριών. Αυτά περιλάμβαναν τα δεδομένα απογραφής των παικτών στον κόσμο του World of Warcraft, τα χαρακτηριστικά προσωπικότητας που αναδεικνύονταν απ' τα ονόματα των παικτών και των ομάδων τους, καθώς επίσης και το κοινωνικό δίκτυο που ανέπτυσσε ο κάθε χρήστης μέσα στο παιχνίδι.

Μια εξίσου σημαντική έρευνα είναι αυτή των Kandias et al. (2013), οι οποίοι επικεντρώθηκαν στην ανάλυση του περιεχομένου που δημοσίευαν ανοικτά χρήστες της ιστοσελίδας YouTube, με στόχο τον εντοπισμό του κινήτρου και των προθέσεών τους. Στόχος τους ήταν να εντοπίσουν άτομα τα οποία είχαν αντιπάθεια για τις αρχές, και ιδιαίτερα την αστυνομία, μέσα απ' την ανάλυση των σχολίων των διαφόρων χρηστών (Kandias et al., 2013). Αυτό έγινε εφικτό με τη σύγκριση της εκτέλεσης ταξινόμησης αρνητικού/θετικού συναισθήματος με μοντέλα ταξινομητών επιτηρούμενης και μη-επιτηρούμενης μηχανικής μάθησης, όπως:

- Multinomial Naïve Bayes
- State Vector Machines
- Logistic Regression
- Λεξικό με λέξεις που υποδηλώνουν αντιπάθεια προς τις αρχές ή την αστυνομία

Με αυτόν τον τρόπο οι Kandias et al. (2013) απέδειξαν ότι τα χαρακτηριστικά του ψυχολογικού προφίλ ενός χρήστη, τα οποία καταδεικνύουν αρνητική συμπεριφορά, είναι αλληλένδετα με την επιθετική συμπεριφορά αυτού στο ενδεχόμενο υλοποίησης εσωτερικής απειλής. Ανάμεσα στις μεθοδολογίες που είχαν συγκρίνει, η πιο επιτυχημένη ήταν αυτή της εκπαίδευσης μηχανικής μάθησης, μιας και με αυτόν τον τρόπο ήταν εφικτή η εκμάθηση πολλών διαφορετικών λέξεων απ' το μοντέλο μάθησης, κάτι που έδωσε και τη μεγαλύτερη ακρίβεια.

Επεκτείνοντας την έννοια των μέσων κοινωνικής δικτύωσης στην πιο παλιά τους μορφή, το email, οι Jiang et al (2018), υπέδειξαν στο άρθρο τους τη μεθοδολογία αναγνώρισης εσωτερικής απειλής λαμβάνοντας υπόψη το ιστορικό πλοήγησης του χρήστη και το συναίσθημα των email που ανταλλάζει. Για τους σκοπούς ταξινόμησης χρησιμοποιήθηκε ένα μη-επιτηρούμενο μοντέλο ταξινόμησης αρνητικού/θετικού συναισθήματος και το συνελικτικό νευρωνικό δίκτυο (convolutional neural network) για την ταξινόμηση κακόβουλων URL στα οποία πλοηγήθηκε ο χρήστης (Jiang et al., 2018). Το συγκεκριμένο έργο ήταν αρκετά πρωτοπόρο καθώς προτάθηκε για πρώτη φορά, σύμφωνα με τους συγγραφείς, αλγόριθμος δημιουργίας προφίλ χρηστών (user profiling) ώστε να υπολογίζεται η ημερήσια και εβδομαδιαία απειλή που προέκυπτε από την συμπεριφορά τους.

Οι Park et. al (2018) προσέγγισαν το θέμα με διαφορετικό τρόπο. Έχοντας ως στόχο τον εντοπισμό υπαλλήλων που μπορούσαν να αποτελέσουν ενδεχόμενο κίνδυνο για μια επιχείρηση προχώρησαν στην ανάλυση πέραν του ενός εκατομμυρίου tweets, με τη χρήση της ταξινόμησης θετικού/αρνητικού συναισθήματος (sentiment analysis). Τα δεδομένα που χρησιμοποίησαν είχαν συλλεχθεί απ' τη διαδικτυακή συλλογή δεδομένων "Sentiment140". Με το πέρας της συγκεκριμένης ανάλυσης προχώρησαν στην ταξινόμηση των χρηστών με βάση το βαθμό επικινδυνότητάς τους.

Ακολούθως οι χρήστες που είχαν ταξινομηθεί ως επικίνδυνοι είχαν επαληθευθεί με το κατά πόσον ακολουθούσαν τις διαδικασίες συμμόρφωσης με τις πολιτικές της ασφάλειας πληροφοριών. Για τον εντοπισμό των πιθανών επικίνδυνων χρηστών είχαν χρησιμοποιήσει και αυτοί μοντέλα μηχανικής μάθησης. Αξίζει να αναφερθεί ότι ένα εκ των μοντέλων μηχανικής μάθησης που είχαν χρησιμοποιήσει, το Decision Tree είχε δώσει τη μεγαλύτερη ακρίβεια μεταξύ των μοντέλων επιτηρούμενης μάθησης (περίπου 90.7%),

ενώ το μοντέλο K-Means είχε τη μεγαλύτερη ακρίβεια στο μοντέλα μη-επιτηρούμενης μάθησης.

Αξίζει να αναφερθεί και η έρευνα των Tan et al. (2019) οι οποίοι είχαν ως στόχο την ολιστική προσέγγιση του θέματος, μέσω του συνδυασμού διαφόρων μεθόδων ανάλυσης δεδομένων για τον εντοπισμό όλων των πιθανών απειλών υλοποίησης εσωτερικής απειλής. Συγκεκριμένα οι ίδιοι είχαν προτείνει της ανάπτυξη ενός ψυχογλωσσικού πλαισίου (psycholinguistic framework) το οποίο συνδύαζε μεθόδους ανάλυσης πολλαπλού κειμένου (multiple text) όπως για παράδειγμα την ανάλυση θετικού/αρνητικού συναισθήματος, την ανάλυση συναισθήματος, καθώς επίσης και μεθόδους μοντελοποίησης διακριτικής ψυχολογικής εκτίμησης.

Μια ακόμη καινοτομία στην έρευνα των Tan et al. (2019) αποτελούσε και η χρήση διαφορετικών ειδών δεδομένων, μιας και είχαν προχωρήσει στην πολλαπλή ανάλυση τεσσάρων συλλογών δεδομένων. Συγκεκριμένα είχαν τα πιο κάτω:

1) Μια συλλογή εταιρικών emails από 150 χρήστες απ' το "CMU Enron email dataset" (Cohen, 2015), με όγκο πάνω από 500 000 μηνύματα.
2) Μια συλλογή γραμμάτων αγάπης, μηνυμάτων ηλεκτρονικού ταχυδρομείου που εξέφραζαν μίσος, καθώς και σημειωμάτων αυτοκτονίας απ' το "LHS dataset" (Mohammad, Saif M., Tony & Yang, 2013).
3) Τη συλλογή δεδομένων "UC Berkeley Enron email dataset" , η οποία περιλάμβανε 1700 μηνύματα ηλεκτρονικού ταχυδρομείου τα οποία είχαν σημανθεί (labelled) με βάση το συναίσθημα, μέσα από ένα όγκο 4,5 εκατομμυρίων μηνυμάτων.
4) Το "Multi-domain review dataset" (Blitzer, Dredze & Pereira, 2007) το οποίο περιλάμβανε θετικές και αρνητικές κριτικές σε προϊόντα που είχαν αγοραστεί απ' τη σελίδα της Amazon.com για μια πληθώρα αντικειμένων, για πάνω από 25 διαφορετικές κατηγορίες προϊόντων. Για να αντιληφθεί κανείς τον όγκο των δεδομένων της συγκεκριμένης συλλογής αξίζει να αναφερθεί ότι αντικείμενα όπως τα βιβλία περιείχαν εκατοντάδες χιλιάδες κριτικές, και τα μουσικά όργανα (μια λιγότερο προτιμώμενη κατηγορία) περιείχε κάποιες εκατοντάδες κριτικές.

Μέσα απ' αυτή την προσέγγιση, οι Tan et al. (2019) κατάφεραν να περιορίσουν τις περιπτώσεις μη εντοπισμού τέτοιων κακόβουλων προθέσεων, αλλά και τις περιπτώσεις όπου ένας χρήστης είχε λανθασμένα κατηγοριοποιηθεί ως χρήστης υψηλού κινδύνου. Το

κλειδί που καθιστά το πλαίσιο τους ως ένα εκ των καλύτερων είναι η δυνατότητα εντοπισμού των κενών μιας αναλυτικής μεθόδου από μια άλλη, περιορίζοντας σημαντικά το ρίσκο παράβλεψης πιθανών απειλών.

Λαμβάνοντας υπόψη όλα τα πιο πάνω (Sang-Sang Tan, Jin-Cheon Na & Duraisamy, 2019, Park, You & Lee, 2018), γίνεται αντιληπτό ότι η χρήση ανάλυσης συναισθήματος σε μηνύματα που οι χρήστες αποστέλλουν σε μέσα κοινωνικής δικτύωσης, είναι πολύ σημαντική για την αναγνώριση εσωτερικής απειλής. Συγκεκριμένα, παρατηρούμε πως η ανάλυση θετικού/αρνητικού συναισθήματος χρησιμοποιείται συχνά (Brdiczka et al., 2012, Park, You & Lee, 2018, Jiang et al., 2018, Kandias et al., 2013), σε αντίθεση με την ανάλυση βάση του κύκλου συναισθημάτων του Plutchik (1982), όπως προτείνουν οι Sang-sang et al. (2019). Εδώ παρατηρούμε πως εμφανίζεται ένα κενό στην μελέτη της χρήσης ανάλυσης συναισθημάτων για τον εντοπισμό εσωτερικής απειλής, ιδίως όταν λάβουμε υπόψη την κρυμμένη γνώση η οποία μπορεί να εξαχθεί με την εφαρμογή τέτοιων μεθόδων.

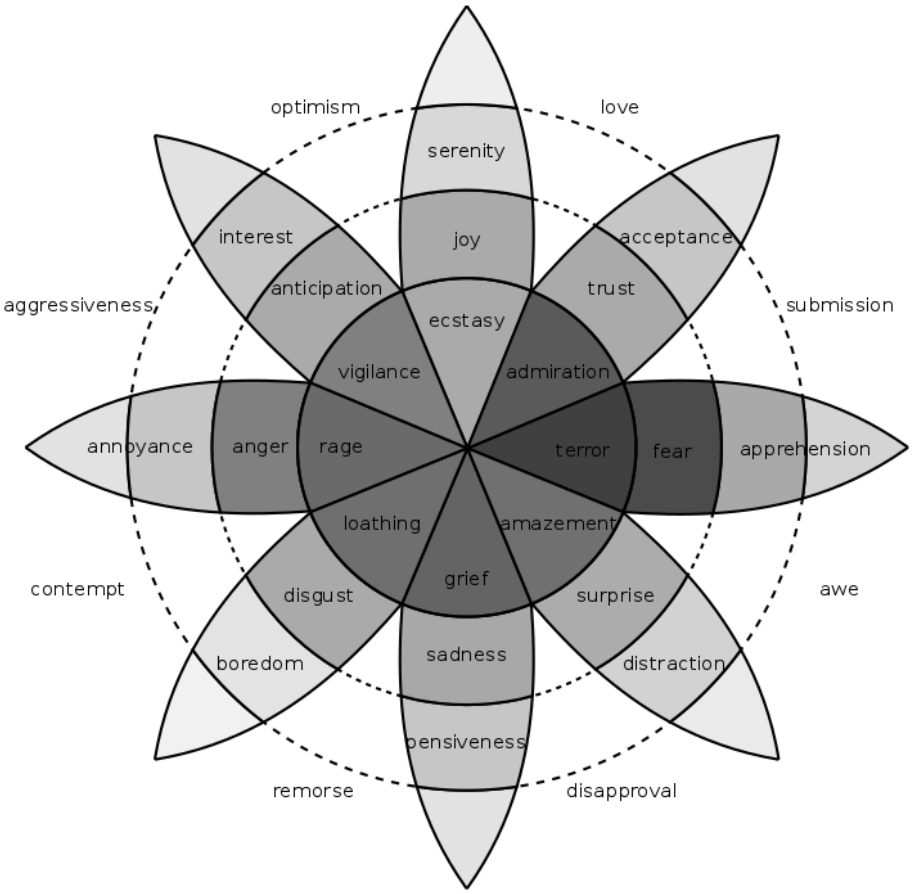
Κεφάλαιο 3
Προσέγγιση

3.1 Μεθοδολογία

Όπως παρατηρούμε από την μελέτη της υπάρχουσας έρευνας, η χρήση ανάλυσης θετικού/αρνητικού συναισθήματος για τον εντοπισμό εσωτερικής απειλής είναι πάντοτε στο προσκήνιο. Εδώ εντοπίζουμε ένα κενό, καθώς όπως διαφαίνεται στην εικόνα 1, ο αριθμός συναισθημάτων που μπορούν να μελετηθούν είναι πολύ μεγαλύτερος (Plutchik, 1982). Από τις έρευνες που έχουμε μελετήσει, μόνο μια (Sang-Sang Tan, Jin-Cheon Na & Duraisamy, 2019) έχει επιλέξει να εστιάσει στην ταξινόμηση συναισθημάτων, όπως ο θυμός και η χαρά. Για αυτό το λόγο, σε αντίθεση με άλλες παρόμοιες έρευνες, εστιάσαμε σε ταξινόμηση 5 διαφορετικών συναισθημάτων με σκοπό να έχουμε περισσότερη ευκρίνεια για τα συμπεράσματα μας:

- Θυμός
- Χαρά
- Λύπη
- Φόβος
- Ουδέτερο

Στη μεταπτυχιακή αυτή διατριβή, δώσαμε έμφαση στην ταξινόμηση μηνυμάτων που ανταλλάσσονται στο κοινωνικό δίκτυο Twitter. Ο κύριος λόγος που έχουμε επιλέξει το κοινωνικό δίκτυο Twitter είναι γιατί ο τρόπος λειτουργίας του Twitter ωθεί τους χρήστες του να μοιραστούν μικρά, ανεξάρτητα μηνύματα τα οποία είναι περιεκτικά σε περιεχόμενο και συναίσθημα. Επίσης, η χρήση hashtags και mentions εντός των μηνυμάτων επιτρέπει την εξαγωγή περαιτέρω μεταδεδομένων που μπορούν να μας βοηθήσουν στην κατάληξη συμπερασμάτων.



Εικόνα 1. Ο κύκλος συναισθημάτων του Plutchik (Wikimedia Commons, 2018). Αντίθετα συναισθήματα τοποθετούνται το ένα απέναντι από το άλλο, και παρόμοια συναισθήματα το ένα δίπλα από το άλλο. Όσο απομακρυνόμαστε από το κέντρο του κύκλου, η ένταση των συναισθημάτων μειώνεται. Τα συναισθήματα που εμφανίζονται στον άσπρο χώρο μεταξύ των πέταλων δυο συναισθημάτων αποτελούν συνδυασμό αυτών.

3.2 Γιατί το συναίσθημα

Όπως αναφέρει και ο Καναδός Νευρολόγος Donald Brian Calne “Η ουσιαστική διαφορά μεταξύ του συναισθήματος και της λογικής είναι ότι το συναίσθημα οδηγεί σε πράξεις ενώ η λογική οδηγεί σε συμπεράσματα”. Η ιστορία έχει δείξει ότι όντως σε πολλές περιπτώσεις το συναίσθημα ωθεί άτομα σε απερίσκεπτες πράξεις οι οποίες υλοποιούν την εσωτερική απειλή.

Αυτό μπορεί να εμφανιστεί με πολλούς τρόπους, όπως για παράδειγμα:

- **Θυμός:** Απόλυση υπαλλήλου τμήματος Πληροφορικής, ο οποίος ως εκδίκηση πριν την απομάκρυνση του διαγράφει μεγάλο αριθμό αρχείων, υποσκάπτοντας τα

μέτρα ασφαλείας τα οποία έχουν καθοριστεί από τον οργανισμό ώστε να μην διαφαίνεται η δόλια πράξη του.

- **Φόβος:** Άτομα εκτός του οργανισμού εκβιάζουν συνεργάτη αυτού με διαρροή ψεύτικων αλλά πειστικών ενοχοποιητικών στοιχείων προς τις αρχές, εκτός εάν ο συνεργάτης τους δώσει πρόσβαση στα συστήματα του οργανισμού. Ο συνεργάτης υποκύπτει στον εκβιασμό και δίνει πρόσβαση στα συστήματα, επιτρέποντας στους κακόβουλους πράκτορες να εγκαταστήσουν λογισμικό καταγραφής πληκτρολόγησης (keylogger) στον υπολογιστή του Διευθύνων Συμβούλου.
- **Λύπη:** Μετά από απώλεια συγγενικού του προσώπου, υπάλληλος δεν τηρεί σωστά τις διαδικασίες, οδηγώντας σε λήψη λανθασμένων αποφάσεων που επιφέρουν οικονομική ζημιά στον οργανισμό.
- **Χαρά:** Ένας εποχιακός υπάλληλος αντιλαμβάνεται πως μπορεί να μεταπωλήσει εμπιστευτικά αρχεία του οργανισμού στον οποίο δουλεύει για μεγάλο οικονομικό όφελος. Εκθαμβωμένος από τα πιθανά κέρδη που αυτή του η κίνηση μπορεί να φέρει, ο υπάλληλος προχωρεί στην διαρροή των αρχείων.

Όπως διαφαίνεται και από τα πιο πάνω παραδείγματα, ακόμη και αν είναι τεχνητά, δεν βρίσκονται εκτός της πραγματικότητας. Ακόμη και εσείς, ως αναγνώστης αυτού του κειμένου, με τη χρήση της εμπάθειας μπαίνετε στα παπούτσια του πράκτορα που υλοποιεί την εσωτερική απειλή και μπορείτε να νιώσετε το συναίσθημα που οδήγησε στην πράξη του. Αυτό υποδεικνύει στο μέγιστο γιατί το συναίσθημα είναι ένας από τους μεγαλύτερους δείκτες επικείμενης εσωτερικής απειλής.

Βέβαια, καθώς οι άνθρωποι είμαστε περίπλοκα όντα, η παρακολούθηση αυτών των συναισθημάτων αποτελεί ένα μικρό μόνο κομμάτι ενός μεγαλύτερου συνόλου παραγόντων που απαρτίζουν την ψυχοσύνθεση μας. Άλλοι τέτοιοι παράγοντες και χαρακτηριστικά, σύμφωνα με την συγγραφέα Δρ. Julie Mehan (Mehan, 2016), είναι:

1. Ανωριμότητα/Παρορμητικότητα
2. Ναρκισσισμός
3. Αντικοινωνική Συμπεριφορά
4. Αδυναμία Δημιουργίας Σχέσεων
5. Εκδικητικότητα
6. Παράνοια
7. Υπεραπόδοση

8. Κίνητρο: απληστία, ιδεολογία, εγωισμός, εκδίκηση και ευκαιρία.

Θεωρούμε πως οι ανωτέρω παράγοντες θα μπορούσαν να αποτελέσουν μελλοντικό έργο έρευνας για το θέμα αυτής της μεταπτυχιακής διατριβής.

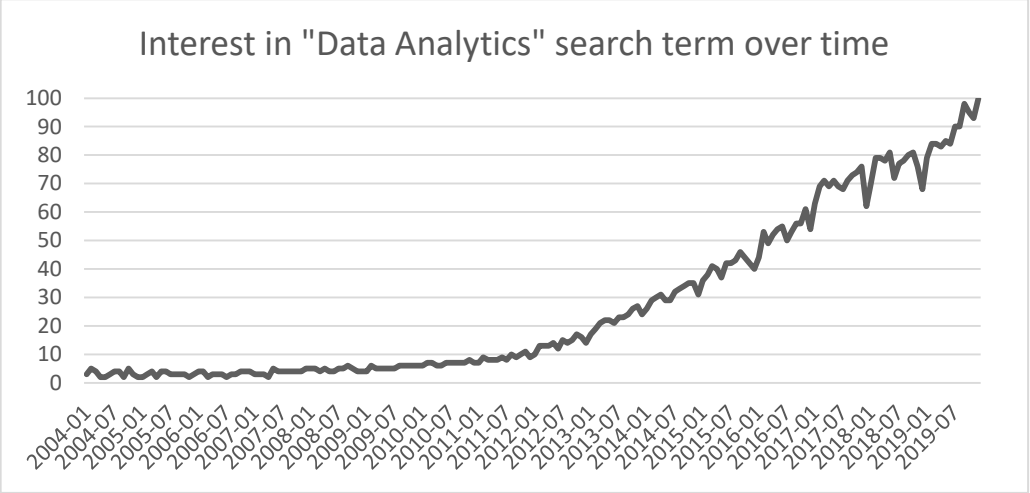
3.3 Γιατί μέσα κοινωνικής δικτύωσης

Με την άνθηση του διαδικτύου έχουν εμφανιστεί και εδραιωθεί στη ζωή μας τα μέσα κοινωνικής δικτύωσης. Αυτές οι ιστοσελίδες επιτρέπουν στους χρήστες τους να δημιουργήσουν εξατομικευμένα προφίλ, να έρθουν σε επικοινωνία με φίλους, ξένους και οργανισμούς και να ανταλλάξουν γραπτά μηνύματα και οπτικοακουστικό περιεχόμενο. Ο άνθρωπος, όντας κοινωνικό ζώο, έχει επιτρέψει στη ζωή του να κατακυριευθεί με αυτά τα μέσα, μοιράζοντας σε αυτά τη ζωή του, τα νιώθω του και τα πιστεύω του.

Έτσι, όπως εύκολα μπορεί κανείς να συμπεράνει, τα δεδομένα τα οποία διακινούνται στα μέσα κοινωνικής δικτύωσης αποτελούν πηγή χρυσού για την έρευνα και την μελέτη εσωτερικής απειλής, καθώς μας επιτρέπουν να παρακολουθήσουμε απευθείας την ακατέργαστη συμπεριφορά του κάθε χρήστη σε μη-επιτηρούμενο περιβάλλον και να εξάγουμε συμπεράσματα.

3.4 Τεχνολογίες που χρησιμοποιήθηκαν

Τα τελευταία χρόνια έχει γίνει τεράστια άνθηση του τομέα ανάλυσης δεδομένων. Σύμφωνα με το Google Trends, ο όρος data analytics για το μήνα Δεκέμβριο 2019, βρίσκεται στο μέγιστο ποσοστό ενδιαφέροντος στα τελευταία 15 χρόνια, με αυξητικές τάσεις. Αυτό έχει επιφέρει την ανάπτυξη αρκετών τεχνολογιών ανάλυσης όπως για παράδειγμα οι ευρέως διαδεδομένες γλώσσες προγραμματισμού Python και R.



Διάγραμμα 2. Το ποσοστό ενδιαφέροντος για τον όρο αναζήτησης "data analytics" στο πέρας του χρόνου, από τον Ιανουάριο 2004 μέχρι και Δεκέμβριο 2019. (Google Trends, 2019)

Για την ανάπτυξη της συστήματος που προδιαγράφεται σε αυτή τη μεταπτυχιακή διατριβή, χρησιμοποιήσαμε την γλώσσα προγραμματισμού Python με αριθμό έκδοσης 3.7.5. Εκτός από την ευκολία χρήσης της για ανάπτυξη λογισμικού, η Python είναι εμπλουτισμένη με τεράστιο πλήθος βιβλιοθηκών που επιτρέπουν τη γρήγορη εφαρμογή μεθόδων μηχανικής μάθησης και ανάλυσης φυσικής γλώσσας. Συγκεκριμένα, η υλοποίηση των ταξινομητών μας βασίστηκε στις πιο κάτω βιβλιοθήκες:

- NLTK (**N**atural **L**anguage **T**ool **K**it)
- Keras
- Scikit-Learn
- Imblearn

Η αποθήκευση των δεδομένων που χρησιμοποιήσαμε για τη λειτουργία και εκπαίδευση του συστήματος έγινε σε σχεσιακή βάση δεδομένων MariaDB. Αυτό επιτρέπει την γρήγορη ανάκληση δεδομένων και την εύκολη μετατροπή τους σε όποια μορφή χρειαστούμε. Περισσότερες λεπτομέρειες για την υλοποίηση της βάσης δεδομένων αναλύονται στο παράρτημα Α.

3.4.1 NLTK

Η βιβλιοθήκη NLTK ή Natural Language Tool Kit παρέχει μεγάλη γκάμα έτοιμων αλγορίθμων επεξεργασίας κειμένου της Αγγλικής γλώσσας. Ακόμη, προσφέρει μεγάλο

αριθμό συνόλων δεδομένων από αρκετές πηγές όπως και έτοιμα λεξικά για χρήση σε περιβάλλοντα μηχανικής μάθησης.

Για τη μεταπτυχιακή αυτή διατριβή, η κύρια χρήση της βιβλιοθήκης NLTK έγινε είναι για τον καθαρισμό του κειμένου το οποίο χρησιμοποιήσαμε για την εκπαίδευση των μοντέλων μηχανικής μάθησης. Χρησιμοποιήθηκαν οι λειτουργίες:

- Αφαίρεση κοινών λέξεων
- Διάσπαση κειμένου σε προτάσεις και λέξεις που το απαρτίζουν
- Λημματοποίηση λέξεων
- Εύρεση ρίζας λέξης
- Ενσωμάτωση σημαδιών άρνησης

3.4.2 Keras

Η βιβλιοθήκη Keras παρέχει ένα απλό application programming interface (API) για καθορισμό, δημιουργία εκπαίδευση και χρήση νευρωνικών δικτύων. Πρέπει να σημειωθεί πως το Keras δεν παρέχει από μόνο του αλγορίθμους νευρωνικών δικτύων. Στο υπόβαθρο, το Keras μπορεί να διασυνδεθεί με άλλες βιβλιοθήκες Python που παρέχουν τις απαραίτητους αλγορίθμους δημιουργίας και λειτουργίας των νευρωνικών δικτύων ώστε να παρέχει την λειτουργία που ζητείται. Στη δική μας περίπτωση, χρησιμοποιήσαμε το Keras με την ευρέως διαδεδομένη τεχνολογία TensorFlow από την ομάδα Google Brain της Google, η οποία παρέχει τη δυνατότητα δημιουργίας, εκπαίδευσης και χρήσης νευρωνικών δικτύων. Ένα πλεονέκτημα του TensorFlow είναι η δυνατότητα εκφόρτωσης εργασιών στην μονάδας επεξεργασίας γραφικών (GPU) έναντί της κεντρικής μονάδας επεξεργασίας (CPU), επιτρέποντας έτσι πολύ μεγαλύτερο παραλληλισμό και πολύ πιο γρήγορη εκπαίδευση των νευρωνικών δικτύων.

3.4.3 Scikit-Learn

Η βιβλιοθήκη Scikit-Learn παρέχει μεγάλη γκάμα έτοιμων αλγορίθμων μηχανικής μάθησης, όπως μεταξύ άλλων:

- Naïve Bayes
- Multinomial Naïve Bayes
- Gaussian Naïve Bayes
- Logistic Regression
- State Vector Machines

Μεταξύ άλλων, παρέχεται και λειτουργικότητα διαχωρισμού του συνόλου δεδομένων σε κομμάτι που χρησιμοποιήθηκε για την εκπαίδευση και κομμάτι που χρησιμοποιήθηκε για τον έλεγχο ποιότητας του τελικού μοντέλου μηχανικής μάθησης.

3.4.4 Imblearn

Η βιβλιοθήκη imblearn (imbalanced learn) παρέχει λειτουργίες για εξισορρόπηση συνόλου δεδομένων εκπαίδευσης μοντέλου για μηχανική μάθηση ταξινόμησης, το οποίο έχει προβλήματα ισορροπίας δεδομένων. Γίνεται χρήση αυτής της βιβλιοθήκης όταν για παράδειγμα δεν υπάρχουν αρκετά δεδομένα για μια από τις κλάσεις δεδομένων που θέλουμε να χρησιμοποιήσουμε, ή μια εκ των κλάσεων έχει πολύ περισσότερα δεδομένα σε σχέση με τις υπόλοιπες.

Οι λειτουργίες που παρέχονται αφορούν όχι μόνο την μείωση δεδομένων σε κλάσεις που έχουν περισσότερα δεδομένα (under-sampling), αλλά και τη δημιουργία νέων δεδομένων για κλάσεις που έχουν λιγότερα δεδομένα βάσει των υφιστάμενων δεδομένων (over-sampling).

Κεφάλαιο 4
Συλλογές Δεδομένων

4.1 Δεδομένα Χρηστών

Ένα ολοκληρωμένο σύστημα όπως αυτό που προτείνεται στην συγκεκριμένη μεταπτυχιακή διατριβή, πρέπει να έχει τη δυνατότητα συλλογής δεδομένων από ιστοσελίδες κοινωνικής δικτύωσης ώστε να μπορεί να αντλεί τα απαραίτητα δεδομένα για τα άτομα που θα μελετά.

Στην περίπτωση του Twitter, το οποίο έχουμε επιλέξει για μελέτη, αυτό μπορεί να γίνει με μεγάλη ευκολία χρησιμοποιώντας τη βιβλιοθήκη Python, Tweepy, η οποία επιτρέπει την απευθείας λήψη δεδομένων χρησιμοποιώντας το API του Twitter. Ωστόσο, η διαδικασία λήψης συγκατάθεσης χρηστών για τη συλλογή των δεδομένων, βάση του κανονισμού General Data Protection Regulation (EU) 2016/679 καθώς και το χρονικό διάστημα που θα χρειαζόταν για την συλλογή αυτών των δεδομένων, μας οδήγησαν στην απόφαση να μην προχωρήσουμε σε συλλογή δεδομένων χρηστών. Αντιθέτως, επιλέξαμε να χρησιμοποιήσουμε τη συλλογή δεδομένων Sentiment140 (Sentiment140, 2019).

Η συλλογή δεδομένων Sentiment140 περιέχει 1.6 εκατομμύρια tweets από τη χρονική περίοδο 6 Απριλίου 2009 μέχρι 25 Ιουνίου 2009. Το Sentiment140 ξεκίνησε ως ερευνητικό έργο υπολογισμού θετικού/αρνητικού συναισθήματος (Go, Bhayani & Huang, 2009). Για τη δημιουργία της συλλογής δεδομένων που χρησιμοποιήθηκε για εκπαίδευση των μοντέλων μηχανικής μάθησης, επέλεξαν να μαζέψουν tweets που περιείχαν emoticons. Στην εκπαιδευτική συλλογή δεδομένων, τα emoticons χρησιμοποιήθηκαν για να καθορίσουν αν μια πρόταση ήταν θετική ή αρνητική. Βάση αυτής της συλλογής, εκπαιδεύτηκαν μοντέλα μηχανικής μάθησης τύπου λεξικού, Naïve Bayes, Maximum Entropy και Support Vector Machines. Αξίζει να σημειωθεί ότι κατά την εκπαίδευση, τα emoticons αφαιρούνταν καθώς είχαν αρνητικό αντίκτυπο για κάποια από τα μοντέλα. Ο τελικός έλεγχος των μοντέλων έγινε με την χρήση tweets διαφόρων θεμάτων τα οποία

συλλέχθηκαν και κατηγοριοποιήθηκαν με το χέρι. Ως αποτέλεσμα της έρευνας τους, πέτυχαν ακρίβεια 83% και έδωσαν στο κοινό τη συλλογή δεδομένων Sentiment140.

Η δομή των δεδομένων που παρέχεται από το Sentiment140 έχει ως ακολούθως:

Sentiment	Message ID	Posted On	Query	User	Text
0	2003974780	Tue Jun 02 07:27:43 PDT 2009	NO_QUERY	allieloves	@Lo_R argh P7 makes me want to cry, I'm so so so so bad at it
0	2003974976	Tue Jun 02 07:27:44 PDT 2009	NO_QUERY	MayraJane	I no wanna go to work today
0	2003975192	Tue Jun 02 07:27:45 PDT 2009	NO_QUERY	abe11825	@laurenzettler i don't even have $5k to move out of my own house... so I can't donate it to anyone else! sorry
0	2003975240	Tue Jun 02 07:27:45 PDT 2009	NO_QUERY	Mangowe	Spectacular guy; @stephenfry. #starspotterrhymes Dreadfully sorry, no more double act with Laurie #starspotterrhymes
0	2003975313	Tue Jun 02 07:27:46 PDT 2009	NO_QUERY	Lydiajohn13	This is fun, relaxing at Starbuck, but will have to leave soon, CRAPS

Πίνακας 1. Δομή δεδομένων που εμπεριέχονται στη συλλογή δεδομένων Sentiment140.

Η στήλη sentiment ενδέχεται να περιέχει τις τιμές:

0 – Αρνητικό Μήνυμα

2 – Ουδέτερο Μήνυμα

4 – Θετικό Μήνυμα

Πρέπει να σημειωθεί ότι, για τους σκοπούς της μεταπτυχιακής διατριβής μας, επιλέξαμε να αγνοήσουμε το sentiment score που παρέχεται από τη συλλογή δεδομένων και να υπολογίσουμε δικό μας χρησιμοποιώντας την λειτουργία VADER που παρέχεται από τη βιβλιοθήκη NLTK.

Για την φόρτωση του συνόλου δεδομένων, δημιουργήσαμε πρόγραμμα Python το οποίο προδιαγράφεται στην ενότητα Α.4 του Παραρτήματος Α, το οποίο ανέλαβε τη μεταφορά των δεδομένων στην βάση δεδομένων που προδιαγράφεται στην ενότητα Β.1 του Παραρτήματος Β.

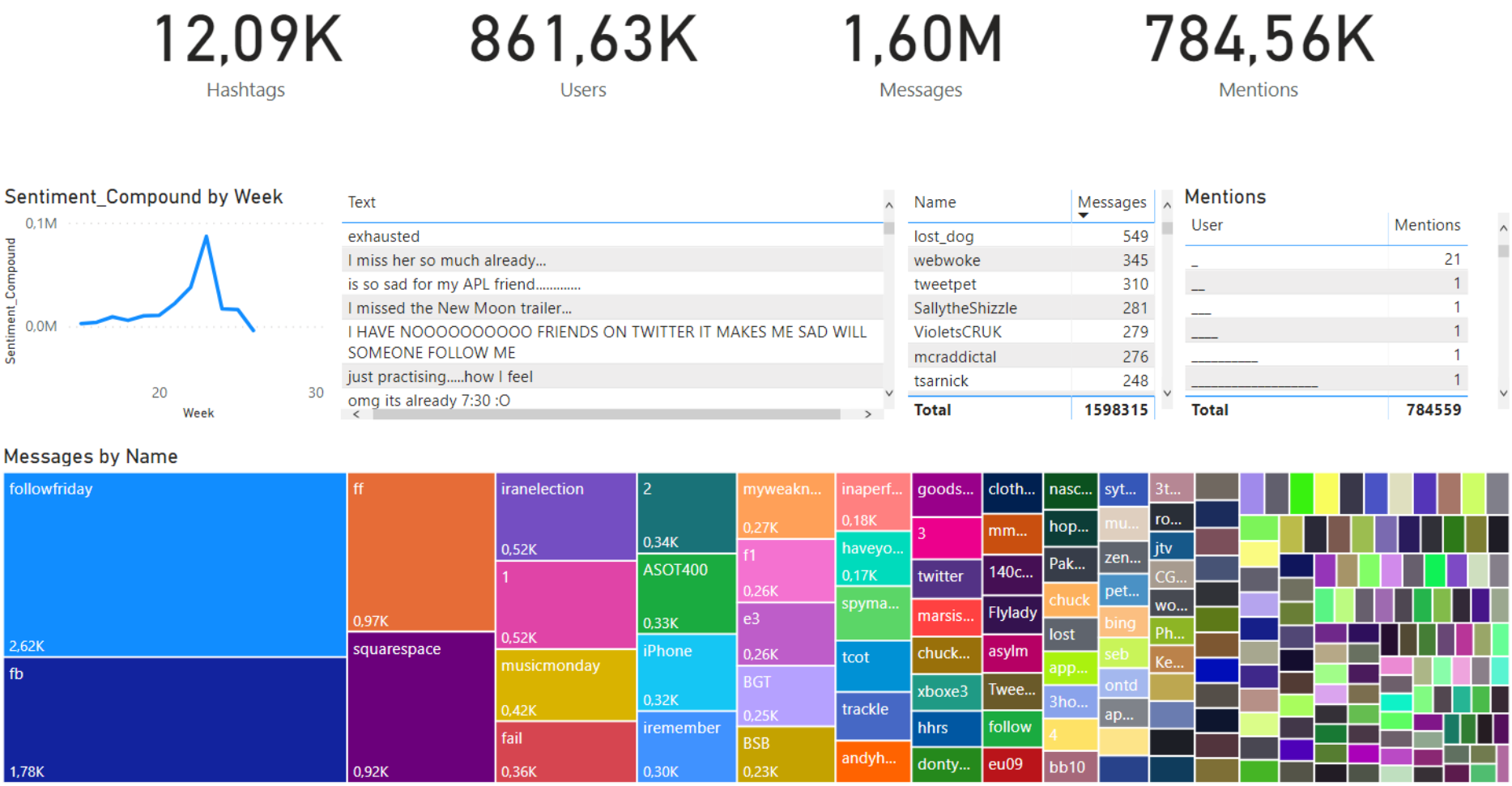
Κατά την φόρτωση των δεδομένων, το πρόγραμμα μας εξήγαγε από κάθε tweet τα ακόλουθα χαρακτηριστικά:

- Όνομα χρήστη
- Αναφορές σε άλλους χρήστες
- Hashtags
- VADER sentiment score

Στόχος μας ήταν να μαζέψουμε μεταδεδομένα τα οποία να μπορούμε να χρησιμοποιήσουμε για απεικόνιση των δεδομένων αλλά και για μελλοντική έρευνα. Ο πίνακας 2 αναφέρει τα μεταδεδομένα που συλλέχθηκαν. Στη συνέχεια, δημιουργήσαμε ένα PowerBI dashboard που επιτρέπει την περαιτέρω ανάλυση και καλύτερη απεικόνιση των δεδομένων.

Χαρακτηριστικό	Σύνολο
Μηνύματα	1598315
Αναφορές σε χρήστες	784559
Χρήστες	861632
Hashtags	12085
Χρήση Hashtag	41193

Πίνακας 2. Μεταδεδομένα που συλλέχθηκαν κατά την φόρτωση της συλλογής δεδομένων Sentiment140. Ο λόγος που ο αριθμός μηνυμάτων είναι λιγότερος από 1.6 εκατομμύρια είναι γιατί βρέθηκαν



Εικόνα 2. Το dashboard που δημιουργήθηκε σε PowerBI για μελέτη του συνόλου δεδομένων Sentiment140.

Για την προσομοίωση των δεδομένων που θα λαμβάναμε από τους χρήστες μιας επιχείρησης, επιλέξαμε χρήστες για τους οποίους είχαμε δεδομένα για διάστημα άνω των 4 εβδομάδων, με μέσο όρο άνω των 4 μηνυμάτων ανά εβδομάδα. Αυτό έριξε τον αριθμό των χρηστών μας από 861632 σε 2982.

Χρήστης	Σύνολο Μηνυμάτων	Μέσος Αριθμός Μηνυμάτων ανά Εβδομάδα	Μέσο Sentiment	Αριθμός Εβδομάδων
mo3ath	87	7,25	0,628341667	12
MyAppleStuff	109	9,0833	2,347433333	12
paulpuddifoot	69	5,75	0,004225	12
sebby_peek	102	8,5	-0,235975	12
StDAY	202	16,8333	0,575925	12
teejay0109	63	5,25	0,217458333	12
thepetshopboy	93	7,75	0,765766667	12

Πίνακας 3. Δείγμα πληροφοριών για τους χρήστες που έχουμε επιλέξει.

4.2 Δεδομένα εκπαίδευσης μοντέλων μηχανικής μάθησης

Για την εκπαίδευση των μοντέλων μηχανικής μάθησης για ταξινόμηση συναισθημάτων, χρησιμοποιήσαμε μηνύματα τα οποία είχαν ήδη ταξινομηθεί και τους είχε δοθεί ετικέτα με το αντίστοιχο συναίσθημα το οποίο εμφανίζεται στο μήνυμα. Το dataset το οποίο επιλέξαμε για την εκπαίδευση των ταξινομητών μας αποτελείται από την συσσωμάτωση δυο ξεχωριστών datasets:

1. Hashtag Emotion Corpus (Mohammad, Saif M., 2012)
2. SemEval-2018 Task 1: Affect in Tweets Data (Mohammad, Saif M. et al., 2018, Mohammad, Saif, Kiritchenko, 2018)

4.2.1 Hashtag Emotion Corpus

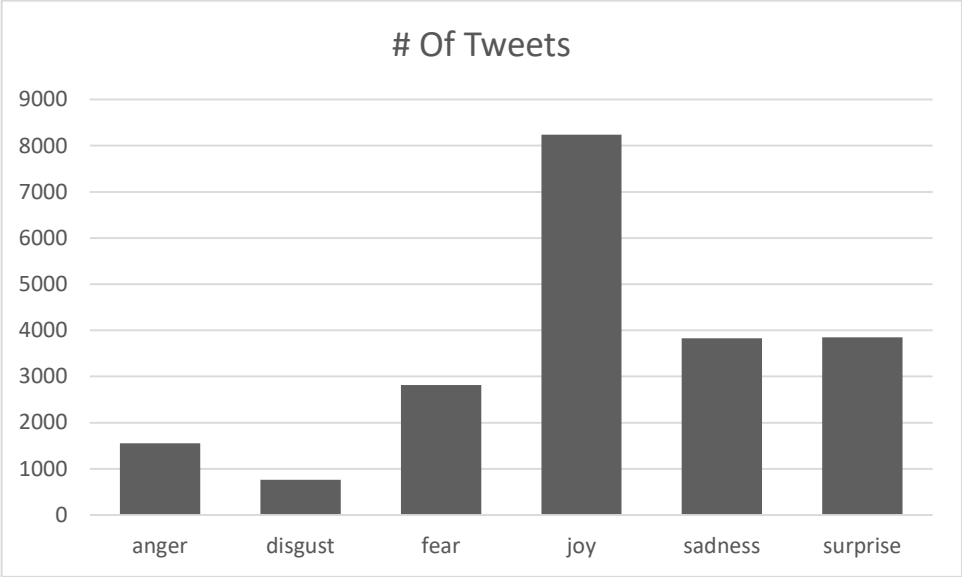
Το hashtag emotion corpus (HEC) δημιουργήθηκε από τον Saif M. Mohammad το 2012. Η μεθοδολογία που ακολουθήθηκε ήταν η αναζήτηση και ταξινόμηση tweets βάση 6 hashtag που υποδηλώνουν συναίσθημα Ekman. Τα hashtag τα οποία λήφθηκαν υπόψη ήταν:

- #anger
- #disgust
- #fear

- #happy
- #sadness
- #surprise

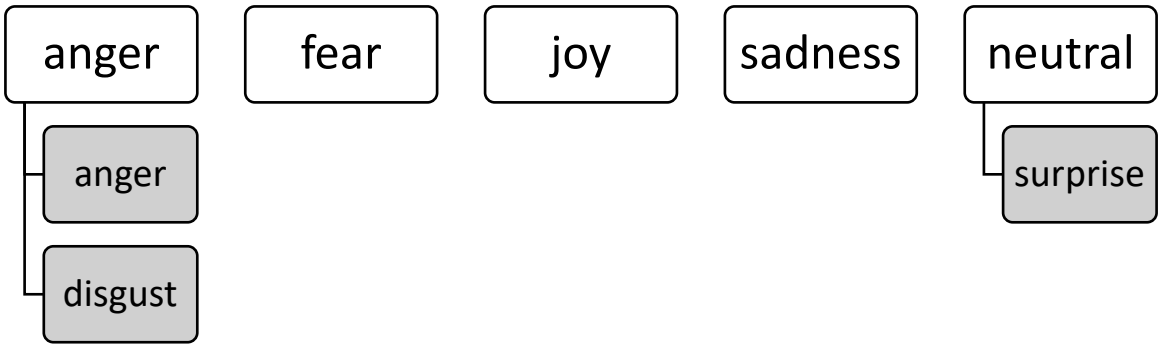
Emotion	# Of Tweets
anger	1555
disgust	761
fear	2816
joy	8240
sadness	3830
surprise	3849
Total	**21051**

Πίνακας 4. Αριθμός tweets ανά συναίσθημα στο Hashtag Emotion Corpus.



Διάγραμμα 3. Κατανομή tweets ανά συναίσθημα.

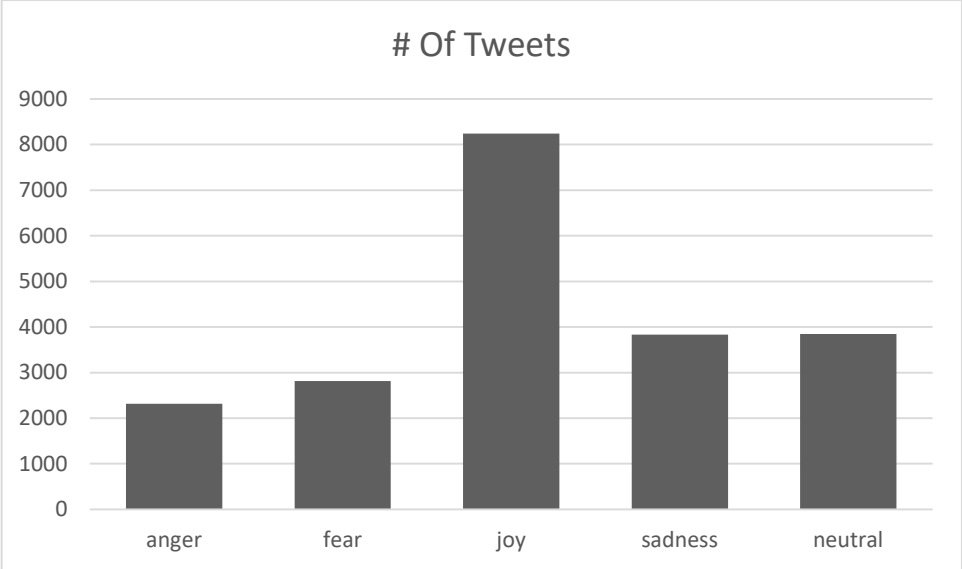
Όπως αναφέραμε στην αρχή της ενότητας 3.1, εστιάζουμε σε 5 συγκεκριμένα συναισθήματα. Για να το πετύχουμε αυτό, προχωρήσαμε σε αλλαγές βάση του διαγράμματος 3.



Διάγραμμα 4. Συμπεριλαμβάνουμε το συναίσθημα disgust με το συναίσθημα anger και μετονομάζουμε το συναίσθημα surprise σε neutral.

Emotion	# Of Tweets
anger	2316
fear	2816
joy	8240
sadness	3830
neutral	3849
Total	**21051**

Πίνακας 5. Ο αριθμός από tweets ανά συναίσθημα μετά τις αλλαγές.



Διάγραμμα 5. Κατανομή tweets ανά συναίσθημα μετά τις αλλαγές.

Παρατηρούμε πως το dataset μας δεν είναι ισορροπημένο καθώς έχουμε πολύ περισσότερα tweets του συναισθήματος joy σε σχέση με τα υπόλοιπα συναισθήματα. Η

χρήση του για εκπαίδευση μοντέλου μηχανικής μάθησης θα οδηγήσει σε ταξινομητή ο οποίος θα μεροληπτεί προς το συναίσθημα joy. Για την επίλυση αυτού του προβλήματος μελετήσαμε τεχνικές over-sampling και under-sampling.

4.2.2 SemEval-2018 Task 1: Affect in Tweets Data

Το συγκεκριμένο dataset χρησιμοποιήθηκε στο πρώτο μέρος του διαγωνισμού SemEval 2018. Στόχος αυτού του μέρους του διαγωνισμού ήταν η δημιουργία μοντέλων ψηφιακής μάθησης που να εκτελούν:

1. **Αναγνώριση έντασης συναισθήματος (EI-reg):** Υπολογισμός ποσοστού έντασης συναισθήματος (0 μέχρι 1) βάση δοθέν tweet και συναισθήματος. Τα συναισθήματα τα οποία μελετώνται είναι (anger, fear, joy, sadness).
2. **Ταξινόμηση έντασης συναισθήματος και σε κλάσεις (EI-oc):** Υπολογισμός κλάσης έντασης συναισθήματος (none, low, medium, high) βάση δοθέν tweet και συναισθήματος.
3. **Αναγνώριση θετικού/αρνητικού σθένους συναισθήματος (V-reg):** Υπολογισμός θετικότητας ή αρνητικότητας tweet (0: πολύ αρνητικό μέχρι 1: πολύ θετικό)
4. **Ταξινόμηση θετικού/αρνητικού σθένους συναισθήματος σε κλάσεις (V-oc):** Υπολογισμός κλάσης θετικότητας ή αρνητικότητας tweet (very negative, moderately negative, slightly negative, neutral, slightly positive, moderately positive, very positive).
5. **Ταξινόμηση συναισθήματος (E-c):** Αναγνώριση των συναισθημάτων που εμφανίζονται σε tweet και ταξινόμηση του σε ένα ή περισσότερα συναισθήματα (anger, anticipation, disgust, fear, joy, love, optimism, pessimism, sadness, surprise, trust, neutral).

Το dataset παρέχει διαφορετικά datasets για κάθε μια από τις πιο πάνω ασκήσεις σε τρεις διαφορετικές γλώσσες: Αραβικά, Αγγλικά και Ισπανικά. Για τους σκοπούς της μεταπτυχιακής διατριβής μας, χρησιμοποιήσαμε τα dataset για τις ασκήσεις EI-reg και E-c στην Αγγλική γλώσσα. Ο λόγος που επιλέξαμε τα dataset EI-reg και E-c είναι γιατί βρήκαμε ότι προσεγγίζουν περισσότερο αυτό που θέλουμε να υπολογίσουμε, σε σύγκριση με τα υπόλοιπα datasets.

Dataset EI-reg

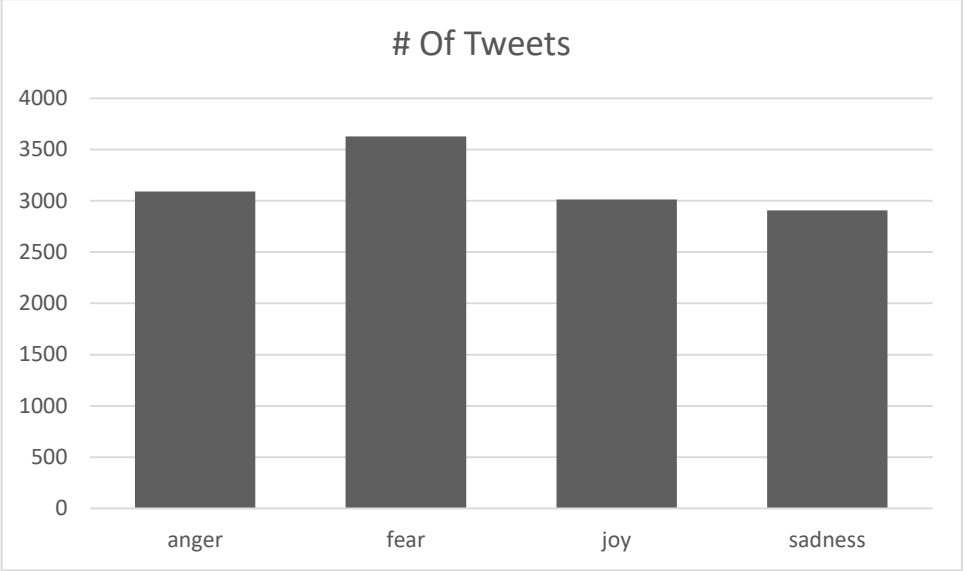
Το dataset EI-reg χωρίζεται σε 3 διαφορετικά μέρη: ανάπτυξη, εκπαίδευση, έλεγχος. Για τους σκοπούς μας, αποφασίσαμε να συνδέσουμε τα διάφορα μέρη του dataset ώστε να έχουμε ένα ολοκληρωμένο αρχείο. Πρέπει να σημειώσουμε ότι υπάρχει αλληλοεπικάλυψη συναισθημάτων για κάθε tweet, δηλαδή ένα tweet μπορεί να έχει πολλαπλά συναισθήματα τα οποία του αντιστοιχούν, με διαφορετική ένταση.

ID	Tweet	Affect Dimension	Intensity Score
2018-En-02533	Don't fucking tag me in pictures as 'family first' when you cut me out 5 years ago. You're no one to me.	anger	0.953
2018-En-01552	sick of this shit. #mad #angry. Rowan Atkinson Is Not Dead. Just A Bloody Online Hoax😠😠😠😠🤬🤬🤬🤬	anger	0.938
2018-En-04002	and after i got home in such a horrible mood my mom pissed me off the moment i stepped my feet in the house so i really almost go off on her	anger	0.931
2018-En-02083	Seriously about to smack someone in the face 😳 #arsehole	anger	0.922
2018-En-00135	And I'm really pissed the fuck off because I do a good job of keeping my kid well because I don't like to see her sick and sad.	anger	0.922
2018-En-01659	I'm soooo annoyed. Wait to start my morning.	anger	0.907

Πίνακας 6. Παράδειγμα δεδομένων του dataset EI-reg.

Emotion	# Of Tweets
anger	3091
fear	3627
joy	3011
sadness	2905
Total	**12634**

Πίνακας 7. Ο αριθμός από tweets ανά συναίσθημα στο dataset EI-reg του διαγωνισμού SemEval 2018 Task 1: Affect in Tweets Data.



Διάγραμμα 6. Κατανομή tweets ανά συναίσθημα στο dataset EI-reg του διαγωνισμού SemEval 2018 Task 1: Affect in Tweets Data.

Dataset E-c

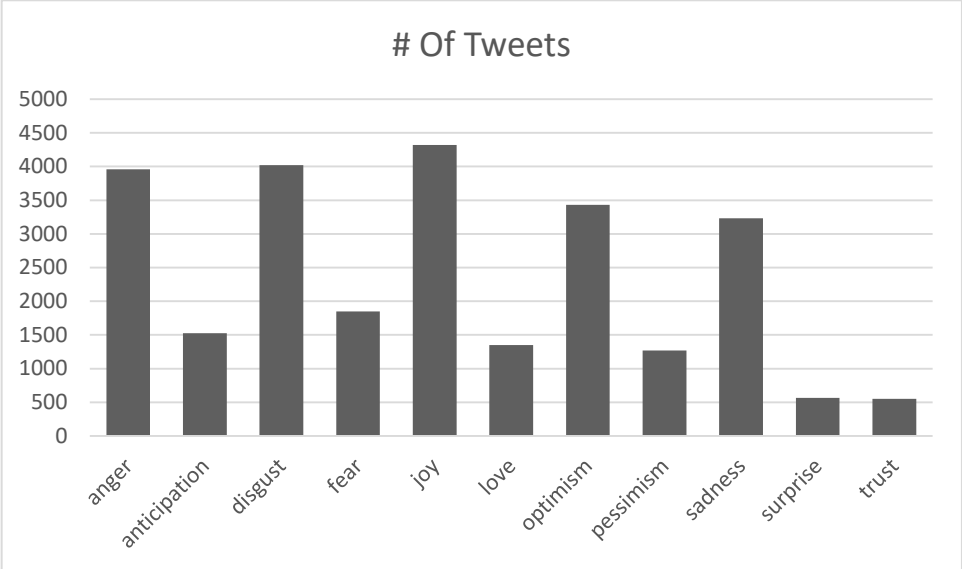
Το dataset E-c χωρίζεται σε 3 διαφορετικά μέρη: ανάπτυξη, εκπαίδευση, έλεγχος. Για τους σκοπούς μας, επιλέξαμε να συνδέσουμε τα διάφορα μέρη του dataset ώστε να έχουμε ένα ολοκληρωμένο αρχείο. Πρέπει να σημειώσουμε ότι υπάρχει αλληλοεπικάλυψη συναισθημάτων για κάθε tweet, δηλαδή ένα tweet μπορεί να έχει από κανένα μέχρι και πολλαπλά συναισθήματα τα οποία του αντιστοιχούν.

ID	Tweet	anger	…	sadness	surprise	trust
2017-En-31267	- blood and mucus and he chokes and has to swallow, mirth cut too short. 'Wrong answer.' It takes some awkward movements - @PersuasiveFuck	1	…	0	0	0
2017-En-10690	-- haired man strides close and watches as the Major flinches away from him, the reaction draws a growl from his throat. -- (@DocHQuinzel)	0	…	0	0	0
2017-En-10682	I can't guess if you holding a grudge against the best'	1	…	1	0	0
2017-En-11049	— Self-hatred gives rise to fury, fury to the desire for self-change.	1	…	1	0	0
2017-En-22239	Look for #contrasts....#behaviour may be a #camouflage....garments a #shell....a #bully may be a #baby inside.....' - Prof Caroline Taylor	0	…	0	0	0

Πίνακας 8. Παράδειγμα δεδομένων dataset E-c.

Emotion	# Of Tweets
anger	3960
anticipation	1527
disgust	4020
fear	1848
joy	4319
love	1348
optimism	3434
pessimism	1270
sadness	3233
surprise	566
trust	553
Total	**26078**
Total (Unique)	**10983**

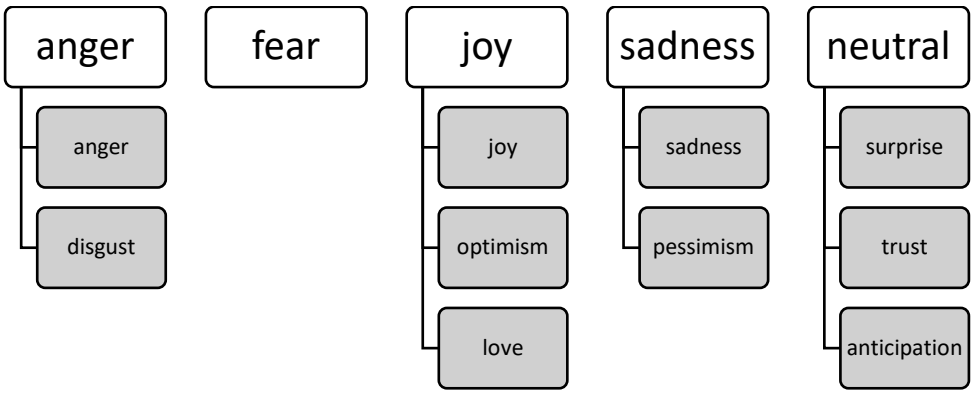
Πίνακας 9. Αριθμός από tweets ανά κατηγορία συναισθήματος. Πρέπει να σημειωθεί ότι υπάρχει αλληλοεπικάλυψη συναισθημάτων ανά tweet, δηλαδή κάποια tweet συμπεριλαμβάνονται σε παραπάνω από μια κατηγορίες συναισθήματος.



Διάγραμμα 7. Κατανομή tweets ανά συναίσθημα στο dataset E-c του διαγωνισμού SemEval 2018 Task 1: Affect in Tweets Data.

Όπως αναφέραμε στην αρχή της ενότητας 3.1, για τους σκοπούς αυτής της μεταπτυχιακής διατριβής, αποφασίσαμε να εστιάσουμε σε 5 συγκεκριμένα

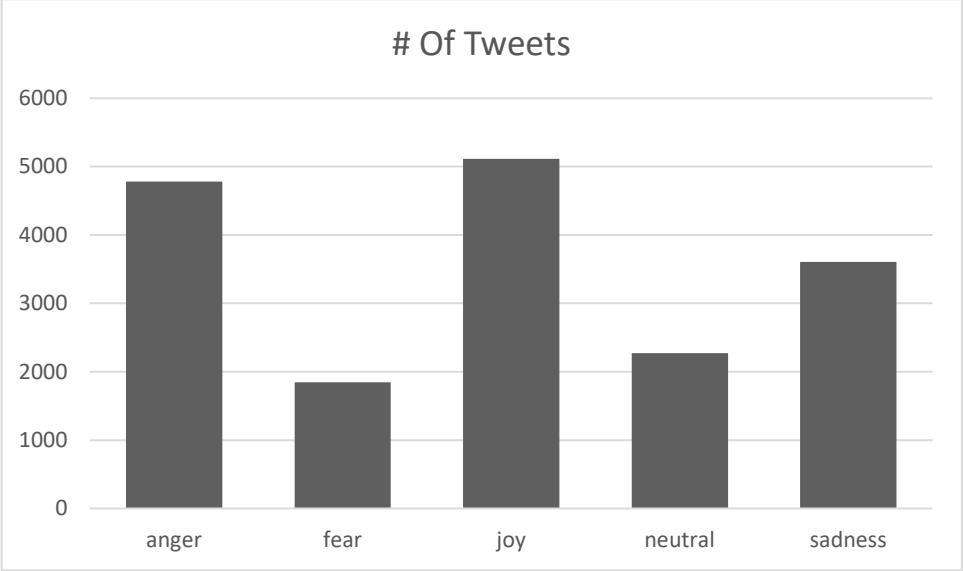
συναισθήματα. Οπότε, προχωρήσαμε σε συσσωμάτωση των 11 κατηγοριών συναισθημάτων βάση του διαγράμματος 7, ώστε να καταλήξουμε στις 5 κατηγορίες που μελετούμε.



Διάγραμμα 8. Η συσσωμάτωση κατηγοριών που εκτελέσαμε για το dataset E-c.

Emotion	# Of Tweets
anger	4780
fear	1848
joy	5114
neutral	2273
sadness	3607
Total	**17622**

Πίνακας 10. Αριθμός από tweets ανά κατηγορία συναισθήματος μετά την συσσωμάτωση των κατηγοριών. Πρέπει να σημειωθεί ότι υπάρχει αλληλοεπικάλυψη συναισθημάτων ανά tweet, δηλαδή κάποια tweet συμπεριλαμβάνονται σε παραπάνω από μια κατηγορίες συναισθήματος.



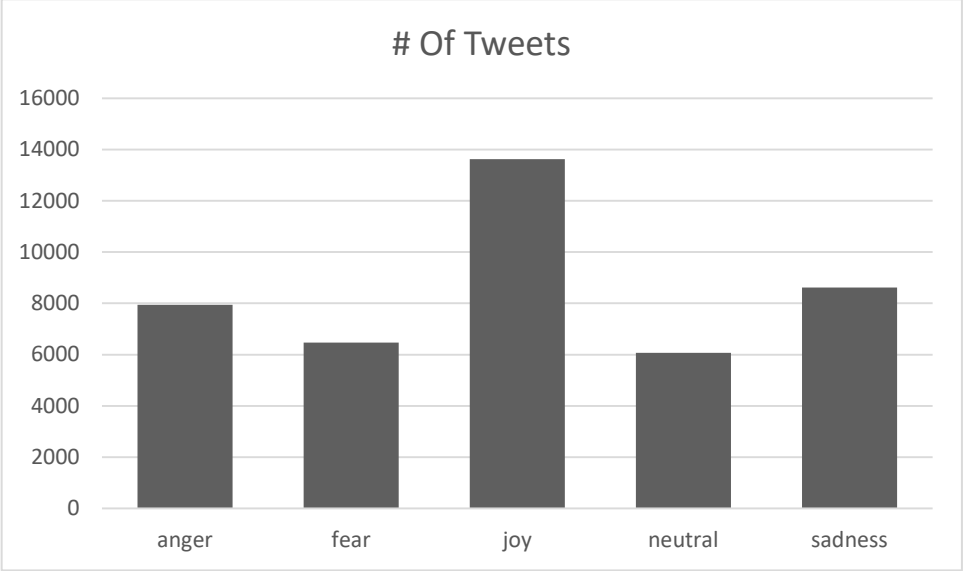
Διάγραμμα 9. Κατανομή από tweets ανά κατηγορία συναισθήματος μετά την συσσωμάτωση των κατηγοριών.

4.3 Τελικό Σύνολο Δεδομένων Εκπαίδευσης

Ενσωματώνοντας τα τρία dataset που επιλέξαμε στην ενότητα 4.2 σε ένα, έχουμε δημιουργήσει ένα dataset με 42733 tweets.

Emotion	# Of Tweets
anger	7947
fear	6471
joy	13622
neutral	6075
sadness	8618
Total	**42733**

Πίνακας 11. Αριθμός από tweets ανά κατηγορία συναισθήματος στο ολοκληρωμένο dataset.



Διάγραμμα 10. Η κατανομή των tweets ανά συναίσθημα.

Στο διάγραμμα 11, παρατηρούμε ότι το τελικό σύνολο δεδομένων μας παρουσιάζει αστάθεια στον αριθμό από tweets ανά κατηγορία. Αυτό πιθανό να προκαλέσει τα μοντέλα μηχανικής μάθησης μας να είναι προκατειλημμένα προς την κατηγορία συναισθήματος με τα περισσότερα tweets.

4.4 Δεδομένα μη-επιτηρούμενων μοντέλων μηχανικής μάθησης

Ως μέρος αυτής της μεταπτυχιακής διατριβής, έχουμε επιλέξει να δημιουργήσουμε και ένα μη-επιτηρούμενο μοντέλο μηχανικής μάθησης το οποίο χρησιμοποιεί λεξικό για την ταξινόμηση προτάσεων στα συναισθήματα τα οποία τις αφορούν.

Το λεξικό το οποίο επιλέξαμε για χρήση στο μη-επιτηρούμενο μοντέλο μηχανικής μάθησης μας αποτελεί το λεξικό το οποίο έχει δημιουργηθεί από το National Research Council Canada (NRC) με τίτλο NRC Emotion Lexicon (Mohammad, Saif, Turney, 2010, Mohammad, Saif M., Turney, 2013). Το λεξικό αυτό δημιουργήθηκε μέσω προσπάθειας crowd funding χρησιμοποιώντας την πλατφόρμα Amazon Mechanical Turk. Η κάθε λέξη στο λεξικό συσχετίζεται με 10 διαφορετικά συναισθήματα:

1. Anger
2. Anticipation
3. Disgust

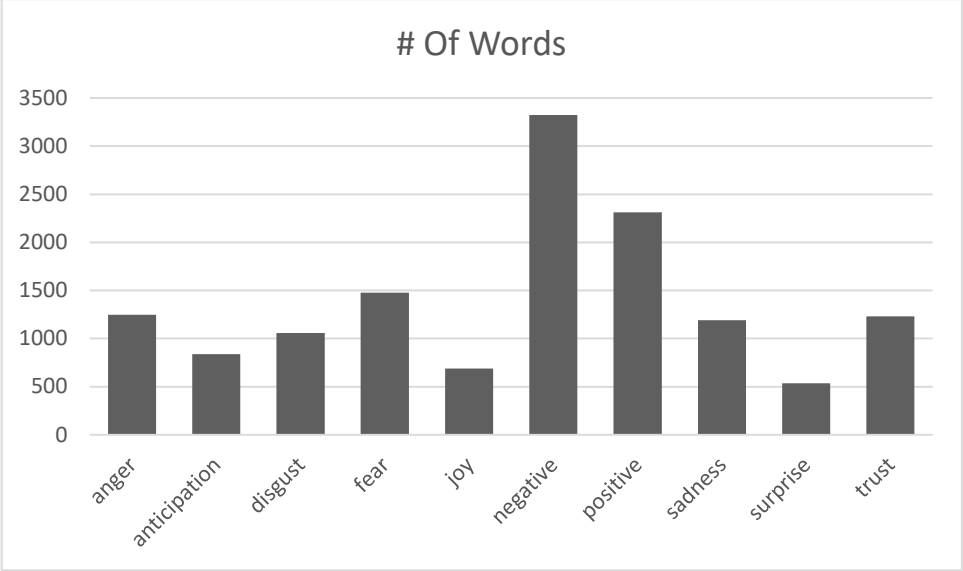
4. Fear
5. Joy
6. Negative
7. Positive
8. Sadness
9. Surprise
10. Trust

Term	Affect Category	Association Flag
abandon	anger	0
abandon	anticipation	0
abandon	disgust	0
abandon	fear	1
abandon	joy	0
abandon	negative	1
abandon	positive	0
abandon	sadness	1
abandon	surprise	0
abandon	trust	0

Πίνακας 12. Δείγμα δεδομένων από το λεξικό. Κάθε γραμμή καθορίζει τη συσχέτιση της λέξης με ένα συναίσθημα. Για κάθε λέξη έχουμε 10 γραμμές, μια για κάθε συναίσθημα. Το λεξικό έχει σύνολο 14182 λέξεις.

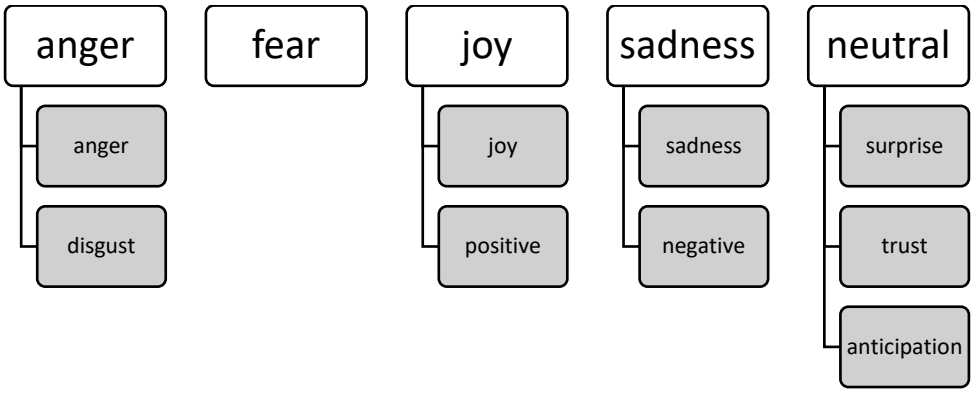
Emotion	# Of Words
anger	1247
anticipation	839
disgust	1058
fear	1476
joy	689
negative	3324
positive	2312
sadness	1191
surprise	534
trust	1231

Πίνακας 13. Αριθμός συσχετισμένων λέξεων ανά συναίσθημα στο λεξικό.



Διάγραμμα 11. Κατανομή λέξεων ανά συναίσθημα.

Καθώς έχουμε εκτελέσει συσσωμάτωση των συναισθημάτων που προδιαγράφονται στα dataset μας, πρέπει να προχωρήσουμε και στην συσσωμάτωση των συναισθημάτων που προδιαγράφονται στο λεξικό. Η συσσωμάτωση που εκτελέστηκε καθορίζεται στο διάγραμμα 9.



Διάγραμμα 12. Η συσσωμάτωση κατηγοριών που εκτελέσαμε για το λεξικό.

Κεφάλαιο 5
Επεξεργασία Δεδομένων

5.1 Προ-Επεξεργασία Δεδομένων

Πριν προωθήσουμε τα δεδομένα στους ταξινομητές για εκπαίδευση, πρέπει πρώτα να αφαιρέσουμε τυχόν θόρυβο που δύναται να υπάρχει μέσα τους ώστε να μειώσουμε την πολυπλοκότητα των δεδομένων μας και να αυξήσουμε την ακρίβεια των ταξινομητών μας. Αυτή η διαδικασία γίνεται σχεδόν αμέσως μετά την φόρτωση των δεδομένων, και πριν τον διαχωρισμό των δεδομένων μας σε δεδομένα εκπαίδευσης και ελέγχου.

Τα βήματα που εκτελούνται έχουν ως ακολούθως. Μπορείτε να βρείτε τον ολοκληρωμένο αλγόριθμο στο παράρτημα Α, ενότητα Α.2:

- Μετατροπή όλων των χαρακτήρων σε μικρά γράμματα. Αυτό μηδενίζει τη πιθανότητα να έχουμε λέξεις όπως “Word”, “word”, “WORD” οι οποίες θα θεωρούνται ως διαφορετικές λέξεις από τους ταξινομητές.
- Αφαιρέσαμε τους τονισμούς από τα γράμματα. Αυτό μηδενίζει την πιθανότητα να έχουμε λέξεις όπως “pēncïl” που να διαφέρουν από τη κανονική λέξη “pencil”
- Μετατροπή αντικειμένων τύπου HTML σε κανονικούς χαρακτήρες π.χ. μετατροπή " σε “, & σε &, < σε <, > σε >, κλπ.
- Αφαίρεση όλων των URL από το κείμενο και αντικατάσταση τους με τη λέξη “urlintext”. Ο λόγος που το κάνουμε αυτό είναι γιατί τα URL συνήθως διαφέρουν το ένα με το άλλο οπότε δεν είναι στατιστικά σημαντικά.
- Αφαίρεση όλων των αναφορών σε χρήστη από το κείμενο και αντικατάσταση τους με τη λέξη “mentionintext”. Ο λόγος που το κάνουμε αυτό είναι γιατί οι αναφορές σε χρήστες διαφέρουν από μήνυμα σε μήνυμα και έτσι δεν είναι στατιστικά σημαντικό.
- Αφαιρέθηκαν τα νούμερα καθώς δεν έχουν κάποια σημασία για το σκοπό μας.
- Αφαιρέθηκαν τα σημεία στίξης.

- Αφαιρέθηκαν επαναλαμβανόμενα γράμματα μέσα σε λέξεις, εμφανίζονται πλέον μόνο 2 φορές. Πχ. Από “heeeeellllooooooooo” σε “heelloo”. Αυτό κανονικοποιεί λέξεις που μπορεί να διαφέρουν λόγω του αριθμού που εμφανίζονται τα γράμματα. Π.χ. “heeeellllooo” και “heeelloooo” θα μετατραπούν και τα δυο σε “heelloo”.
- Εκτελέστηκε λημματοποίηση, δείτε υποενότητα 5.1.1.
- Εκτελέστηκε stemming, δείτε υποενότητα 5.1.2.
- Εκτελέστηκε σηματοδότηση άρνησης. Δηλαδή, λέξεις που έρχονται μετά από λέξη που δηλώνει άρνηση, π.χ. “not” σηματοδοτούνται με το σημάδι “_NEG” ώστε να ξεχωρίζει στους ταξινομητές ότι η συγκεκριμένη λέξη έχει διαφορετική σημασία απ’ότι σημαίνει κανονικά. Π.χ. “this is not so funny” μετατρέπεται σε “this is not so_NEG funny_NEG”. Πρέπει να σημειωθεί ότι εκτελούμε αυτή τη σηματοδότηση σε επίπεδο πρότασης ώστε η σηματοδότηση της αρνητικότητας να μην προχωρά μεταξύ προτάσεων. Π.χ. “This is not so good. It’s bad!” μετατρέπεται σε “This is not so_NEG good_NEG. It’s bad!”.
- Εκτελέστηκε αφαίρεση λέξεων κοινής χρήσης (stopwords). Γι’ αυτό το σκοπό χρησιμοποιήθηκε έτοιμο λεξικό με stopwords από τη βιβλιοθήκη NLTK. Μερικά παραδείγματα λέξεων που αφαιρούνται διαφαίνονται στον πίνακα 14.

i	they	a	in	own	doesn
me	them	an	out	same	doesn't
my	their	the	on	so	hadn
myself	theirs	and	off	than	hadn't
we	themselves	but	over	too	hasn
our	what	if	under	very	hasn't
ours	which	or	again	s	haven
ourselves	who	because	further	t	haven't
you	whom	as	then	can	isn
you're	this	until	once	will	isn't
you've	that	while	here	just	ma
you'll	that'll	of	there	don	mightn
you'd	these	at	when	don't	mightn't
your	those	by	where	should	mustn
yours	am	for	why	should've	mustn't
yourself	is	with	how	now	needn
yourselves	are	about	all	d	needn't
he	was	against	any	ll	shan
him	were	between	both	m	shan't
his	be	into	each	o	shouldn
himself	been	through	few	re	shouldn't
she	being	during	more	ve	wasn
she's	have	before	most	y	wasn't
her	has	after	other	ain	weren
hers	had	above	some	aren	weren't
herself	having	below	such	aren't	won
it	do	to	no	couldn	won't
it's	does	from	nor	couldn't	wouldn
its	did	up	not	didn	wouldn't
itself	doing	down	only	didn't	

Πίνακας 14. Λίστα από stopwords από την βιβλιοθήκη NLTK

5.1.1 Λημματοποίηση

Λημματοποίηση είναι η διαδικασία όπου μετατρέπουμε λέξεις στην λέξη-ρίζα τους. Στόχος είναι να μειώσουμε τον αριθμό των λέξεων-θορύβου που εμφανίζονται στα δεδομένα μας ώστε τα μοντέλα μηχανικής μάθησης να μπορούν να εστιάσουν καλύτερα στην επίλυση του προβλήματος.

Ως αποτέλεσμα εκτέλεσης αυτού του αλγορίθμου, έχουμε για παράδειγμα, «καλύτερο» μετατρέπεται σε «καλό», «γάτες» μετατρέπονται σε «γάτα», «ομορφιά» σε «όμορφο», «σπιτάκι» σε «σπίτι» κ.ο.κ.

Στην δική μας περίπτωση χρησιμοποιήσαμε την μοντέλο WordNetLemmatizer , το οποίο παρέχεται από την βιβλιοθήκη NLTK. Το μοντέλο αυτό έχει προ-εκπαιδευτεί σε δεδομένα

αγγλικής γλώσσας και έχει την δυνατότητα να επιλέγει την κατάλληλη λύση βάση του μέρους του λόγου (ρήμα, ουσιαστικό, επίθετο, επίρρημα) που αποτελεί κάθε λέξη.

Πρέπει να σημειωθεί ότι η λημματοποίηση θεωρείται καλύτερη από το Stemming, καθώς λόγω του τρόπου λειτουργίας του, το Stemming δύναται να επιστρέψει λέξεις οι οποίες δεν είναι σωστές, δυσχεραίνοντας το έργο μας και μειώνοντας την ακρίβεια μοντέλων που λειτουργούν με τη χρήση λεξικού όπως το μοντέλο ELC το οποίο περιγράφεται στο κεφάλαιο 6.

5.1.2 Stemming

Η διαδικασία stemming έχει τον ίδιο στόχο με τη διαδικασία λημματοποίησης, δηλαδή την μετατροπή πολλαπλών λέξεων που έχουν κοινή ρίζα σε μια λέξη, ώστε να μειώσουν το μέγεθος των παραμέτρων που τα μοντέλα μηχανικής μάθησης μας καλούνται να μελετήσουν.

Σε αντίθεση με την λημματοποίηση ωστόσο, η μετατροπή των λέξεων γίνεται μέσω συγκεκριμένων κανόνων, οι οποίοι αποκόπτουν και αντικαθιστούν μέρη της λέξης (εξ ου και ο όρος stemming) και μπορεί να οδηγήσουν σε λέξεις οι οποίες είναι άκυρες. Για παράδειγμά, χρησιμοποιώντας τον SnowballStemmer που παρέχεται από την βιβλιοθήκη NLTK, η λέξη “happy” μετατρέπεται σε “happi”, κάτι το οποίο σίγουρα θα επηρεάσει την απόδοση μοντέλων μηχανικής μάθησης όπως το ELC.

5.2 Τροφοδοσία Δεδομένων σε Μοντέλα Μηχανικής Μάθησης

Έχοντας μαζέψει και καθαρίσει τα δεδομένα μας, είμαστε πλέον έτοιμοι να τα τροφοδοτήσουμε στα μοντέλα μηχανικής μάθησης μας για να ξεκινήσουν τη διαδικασία της εκπαίδευσης. Πριν ξεκινήσουμε όμως, πρέπει να μετατρέψουμε τα δεδομένα μας σε μορφή την οποία τα μοντέλα μας θα μπορούν να επεξεργαστούν.

Πιο κάτω προδιαγράφονται τρόποι κωδικοποίησης των δεδομένων κειμένου σε μορφές που μπορούν να καταναλωθούν από τα μοντέλα μηχανικής μάθησης μας.

5.2.1 Bag of Words

Η μέθοδος Bag of Words επιτρέπει την δημιουργία ενός ενιαίου πίνακα ο οποίος περιέχει όλα τα χαρακτηριστικά που εμφανίζονται στη συλλογή δεδομένων μας και καθορίζει τη σχέση κάθε καταχώρησης με τα χαρακτηριστικά στα οποία της αντιστοιχούν. Παραδείγματος χάρη, έχοντας τις προτάσεις “this is such a nice thing to say”, “you don’t say”, βάση των χαρακτηριστικών των οποίων κτίζεται κωδικοποιητής Bag of Words. Επίσης, 2 προτάσεις κωδικοποιούνται από τον κωδικοποιητή.

such	this	is	a	thing	to	say	nice	You	don’t
1	1	1	1	1	1	1	1	0	0
0	0	0	0	0	0	1	0	1	1

Πίνακας 15. Παράδειγμα One-Hot encoding

Οπότε, αν προσθέσουμε την λέξη “I don’t like this very much”, έχουμε το ακόλουθο:

such	this	is	a	thing	to	say	nice	You	don’t
1	1	1	1	1	1	1	1	0	0
0	0	0	0	0	0	1	0	1	1
0	1	0	0	0	0	0	0	0	1

Πίνακας 16. Παράδειγμα One-Hot encoding

Όπως παρατηρούμε, το bag of words δεν διατηρεί τη σειρά τον λέξεων μέσα στην δομή της. Για κάθε δοθέν καταχώρηση που θέλουμε να κωδικοποιήσουμε, αναθέτει τον αριθμό εμφανίσεων της κάθε λέξης βάση του λεξικού στο οποίο έχει κτιστεί. Οπότε, υπάρχει η περίπτωση πολύ διαφορετικές προτάσεις να καταγραφούν με τον ίδιο τρόπο (π.χ. “this is fun” vs “is this fun”), το οποίο δημιουργεί προβλήματα κατά την εκπαίδευση των μοντέλων.

Ένα κύριο πρόβλημα με αυτή τη κωδικοποίηση δεδομένων είναι ανάγκη δημιουργίας λιστών μεγέθους όσο είναι το λεξικό του κωδικοποιητή, για κάθε λέξη. Προφανώς αυτό ενέχει προβλήματα καθώς η ποσότητα μνήμης που χρειαζόμαστε για την υλοποίηση μεγάλων λεξικών, είναι τεράστια.

Ακόμη ένα πρόβλημα που προκύπτει είναι η καταγραφή κοινών λέξεων, οι οποίες λόγω της αυξημένης συχνότητας χρήσης τους, έχουν μεγάλο αριθμό εμφανίσεων. Αυτό οδηγεί τα μοντέλα μηχανικής μάθησης μας στο να δώσουν παραπάνω σημασία σε αυτές απ'όσο χρειάζεται. Αυτό το πρόβλημα διορθώνεται με τη χρήση της κωδικοποίησης TF-IDF που αναλύεται στην επόμενη υποενότητα.

5.2.2 TF-IDF Encoding

Αυτός ο τρόπος κωδικοποίησης ονομάζεται Term Frequency – Inverse Document Frequency και έρχεται να επιλύσει το πρόβλημα που εμφανίζεται στην κωδικοποίηση Bag Of Words, χρησιμοποιώντας στατιστικές μεθόδους.

Συγκεκριμένα, λαμβάνεται υπόψη η συχνότητα εμφάνισης μιας λέξης σε ένα κείμενο και η συχνότητα εμφάνιση της λέξης σε όλα τα κείμενα που έχουν κωδικοποιηθεί από τον κωδικοποιητή. Στην περίπτωση που η λέξη έχει μεγάλη συχνότητα εμφάνισης σε πολλαπλά κείμενα, τότε η κανονικοποιημένη συχνότητα της τιμωρείται. Όσο πιο μεγάλη χρήση έχει μια λέξη μεταξύ κειμένων, τόσο πιο πολύ τιμωρείται.

5.2.3 Tokenizer (Keras)

Ο κωδικοποιητής Tokenizer παρέχεται από την βιβλιοθήκη Keras για τη τροφοδοσία νευρωνικών δικτύων. Η λειτουργία του περιγράφεται στην υποενότητα 6.1.1

5.2.4 n-grams

Τα n-grams αποτελούν ομάδες από n λέξεις, όπου το n καθορίζεται κατά την κωδικοποίηση. Στις περιπτώσεις που έχουμε μελετήσει μέχρι στιγμής, έχουμε μιλήσει για 1-grams ή unigrams, όπου κάθε λέξη είναι και ένα χαρακτηριστικό του κειμένου μας. Η χρήση των n-grams επιτρέπει την ομαδοποίηση λέξεων με την ελπίδα ότι αυτή η συσσωμάτωση μπορεί να μας δώσει περισσότερη πληροφορία απ' ότι αν είχαμε κάθε λέξη ξεχωριστά.

(2) *Παράδειγμα: "writing this paper has been fun"*

(3) *Σε unigrams: "Writing", "this", "paper", "has", "been", "fun"*

(4) *Σε 2-grams: "writing this", "this paper", "paper has", "has been", "been fun"*

(5) *Σε 3-grams: "writing this paper", "this paper has", "paper has been", "has been fun"*

5.2.5 Word2Vec

Το μοντέλο Word2Vec αποτελεί νευρωνικό δίκτυο το οποίο έχει δημιουργηθεί με στόχο να εξαγάγει σημαντικές συντακτικές πληροφορίες από το κείμενο που του δίδεται. Αυτές οι πληροφορίες ακολούθως παρέχονται σαν σειρά από βάρη τα οποία μπορούν να χρησιμοποιηθούν στο στρώμα embedding άλλων νευρωνικών δικτύων για να βελτιώσουν την απόδοση τους.

5.3 Ισορροπία Κλάσεων Συλλογής Δεδομένων

Όπως έχουμε δείξει στο κεφάλαιο 4, ο αριθμός δεδομένων που έχουμε για κάθε κλάση στις συλλογές δεδομένων μας δεν είναι πάντοτε ο ίδιος. Δυστυχώς αυτό δημιουργεί το προβλήμα όπου τα μοντέλα μηχανικής μάθησης μας μαθαίνουν καλύτερα τις κλάσεις με μεγαλύτερο αριθμό δεδομένων και αγνοούν άλλες κλάσεις.

Για την επίλυση αυτού του προβλήματος έχουμε 2 λύσεις:
1. Μείωση του αριθμού δεδομένων στις κλάσεις που έχουν μεγαλύτερο αριθμό δεδομένων (majority) σε σχέση με τις υπόλοιπες κλάσεις (minority).
2. Αύξηση του αριθμού δεδομένων στις κλάσεις που έχουν χαμηλότερο αριθμό δεδομένων

Για τους σκοπούς της μεταπτυχιακής αυτής διατριβής, λύνουμε το πρόβλημα χρησιμοποιώντας τη βιβλιοθήκη imblearn η οποία παρέχει μεθόδους επίλυσης αυτού του προβλήματος. Αυτοί είναι οι ακόλουθοι:

5.3.1 RandomUnderSampler

Επιτρέπει την αφαίρεση δεδομένων από τη κλάση majority ώστε να έχει την ίδια ποσότητα δεδομένων με τις υπόλοιπες. Η αφαίρεση των δεδομένων γίνεται τυχαία και ενέχει το πρόβλημα αφαίρεσης δεδομένων τα οποία μπορεί να είναι σημαντικά για το έργο μας.

5.3.2 RandomOverSampler

Επιτρέπει την ενίσχυση δεδομένων από τις κλάσεις minority ώστε να έχουν την ίδια ποσότητα δεδομένων με τη κλάση majority. Η πρόσθεση των δεδομένων γίνεται με τυχαία αντιγραφή υπάρχουσων δεδομένων της κλάσης. Πρέπει να σημειωθεί ότι αυτό ενέχει το πρόβλημα όπου το μοντέλο μας μπορεί να μην μπορεί να γενικέψει τις γνώσεις

του όταν πάμε να το χρησιμοποιήσουμε σε άγνωστα δεδομένα. Με λίγα λόγια, ενισχύει την πιθανότητα να κάνουμε overfit στα δεδομένα εκπαίδευσης μας.

5.3.3 SMOTE

Ο αλγόριθμος SMOTE (Chawla et al., 2002) παρέχει τη δυνατότητα *δημιουργίας* νέων δεδομένων βάση των ήδη υπάρχων δεδομένων της κλάσης minority. Χρησιμοποιώντας χαρακτηριστικά από τα υπάρχων δεδομένα, ο αλγόριθμος SMOTE δημιουργεί νέα δεδομένα ώστε να ισορροπήσει τις κλάσεις της συλλογής δεδομένων.

Η επιλογή και δημιουργία των συνθετικών δεδομένων γίνεται μέσω της επιλογής k κοντινότερων γειτόνων βάση της ευκλείδειας απόστασης των χαρακτηριστικών τους.

Κεφάλαιο 6
Μοντέλα Μηχανικής Μάθησης

6.1 Επιβλεπόμενα Μοντέλα Μηχανική Μάθησης

Για την ταξινόμηση μηνυμάτων σε κατηγορίες συναισθημάτων, προχωρήσαμε στην χρήση μοντέλων μηχανικής μάθησης, χρησιμοποιώντας το συσσωματωμένο dataset που δημιουργήσαμε στην ενότητα 4.2.

Ακολουθούν πιο κάτω τα μοντέλα μηχανικής μάθησης που χρησιμοποιήθηκαν.

6.1.1 LSTM

Το μοντέλο μηχανικής μάθησης Long Short-Term Memory, ή LSTM αποτελεί ένα επαναλαμβανόμενο νευρωνικό δίκτυο βαθιάς μάθησης το οποίο προτάθηκε από τους Sepp Hochreiter και Jürgen Schmidhuber το 1997 (Hochreiter, Schmidhuber, 1997).

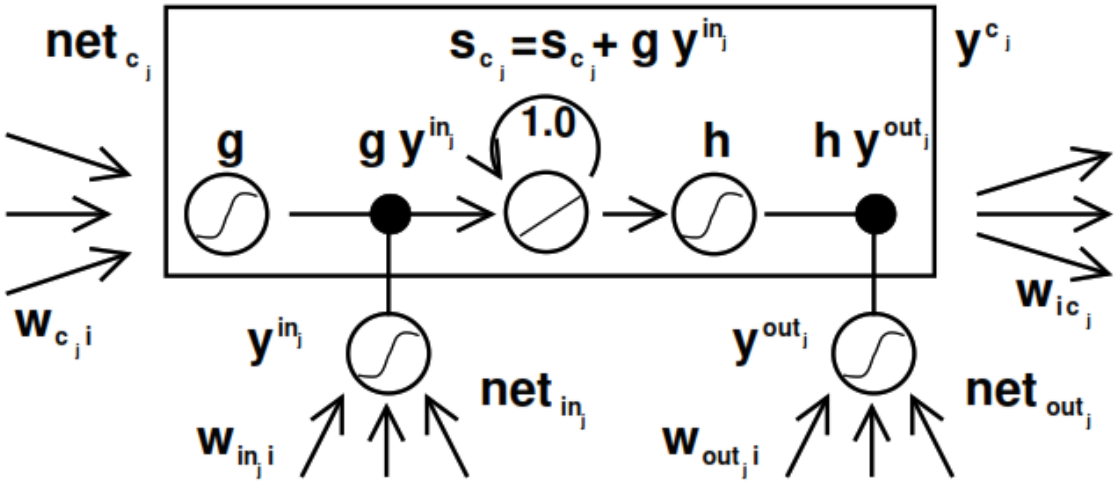
Σύμφωνα με τους συγγραφείς, ένα βασικό πρόβλημα το οποίο εμφανίζεται σε επαναλαμβανόμενα νευρωνικά δίκτυα είναι η αστάθεια που προκαλείται κατά τη διαδικασία ανατροφοδότησης του σφάλματος. Η ανατροφοδότηση του σφάλματος χρησιμοποιείται για την ενημέρωση των βαρών σε κάθε νευρώνα του δικτύου και αποτελεί το κύριο τρόπο διατήρησης γνώσης και εκπαίδευσης για αυτού του είδους τα νευρωνικά δίκτυα. Σε περιπτώσεις όπου το ανατροφοδοτούμενο σφάλμα είναι πολύ μικρό ή πολύ μεγάλο, τότε εμφανίζεται το πρόβλημα όπου τα βάρη είτε δεν αλλάζονται καθόλου, είτε αλλάζονται σε μεγάλο βαθμό. Στην πρώτη περίπτωση, παρατηρείται σημαντική καθυστέρηση στην διαδικασία εκπαίδευσης καθώς οι αλλαγές που συμβαίνουν στα βάρη των νευρώνων είναι πολύ μικρές, έως και πλήρης αποτυχία εκπαίδευσης καθώς οι αλλαγές είναι τόσο μικρές που το νευρωνικό δίκτυο σταματά να αλλάζει. Στην δεύτερη περίπτωση, παρουσιάζεται ταλάντωση των βαρών του δικτύου,

αποτρέποντας το από την σύγκλιση σε λύση. Αυτά τα δύο προβλήματα ονομάζονται στην επιστημονική κοινότητα ως vanishing gradient problem και exploding gradient problem.

(1) $$\boldsymbol{w\_t = w\_(t-1) - gradient * learningRate}$$

Εξίσωση που δείχνει τον τρόπο ενημέρωσης των βαρών ενός νευρωνικού δικτύου. Παρατηρούμε πως για πολύ μικρό gradient, το βάρος δεν αλλάζει, ενώ για μεγάλο gradient, έχουμε μεγάλη αλλαγή του βάρους πιθανό να προκαλέσει αστάθεια.

Κύριος στόχος του μοντέλου LSTM είναι η επίλυση των πιο πάνω προβλημάτων με τη χρήση λεγόμενων νευρώνων μνήμης. Ένας νευρώνας μνήμης LSTM αποτελείται από ένα γραμμικό νευρώνα ο οποίος ανατροφοδοτεί εσωτερικά τον εαυτό του με σταθερό βάρος 1. Αυτό ονομάζεται Constant Error Carousel (CEC) και αποτελεί τη βάση των νευρωνικών δικτύων LSTM. Η είσοδος και η έξοδος πληροφοριών από τον νευρώνα για σκοπούς εκπαίδευσης του CEC τυγχάνει ελέγχου από δυο πολλαπλασιαστικές πύλες, μια πύλη εισόδου και μια πύλη εξόδου, οι οποίες έχουν σιγμοειδή λειτουργία ενεργοποίησης. Ο σκοπός αυτών των δυο πυλών είναι να αποτρέψουν τον νευρώνα από το να μάθει ή να διαδώσει αχρείαστες πληροφορίες από και προς άλλους νευρώνες μνήμης.



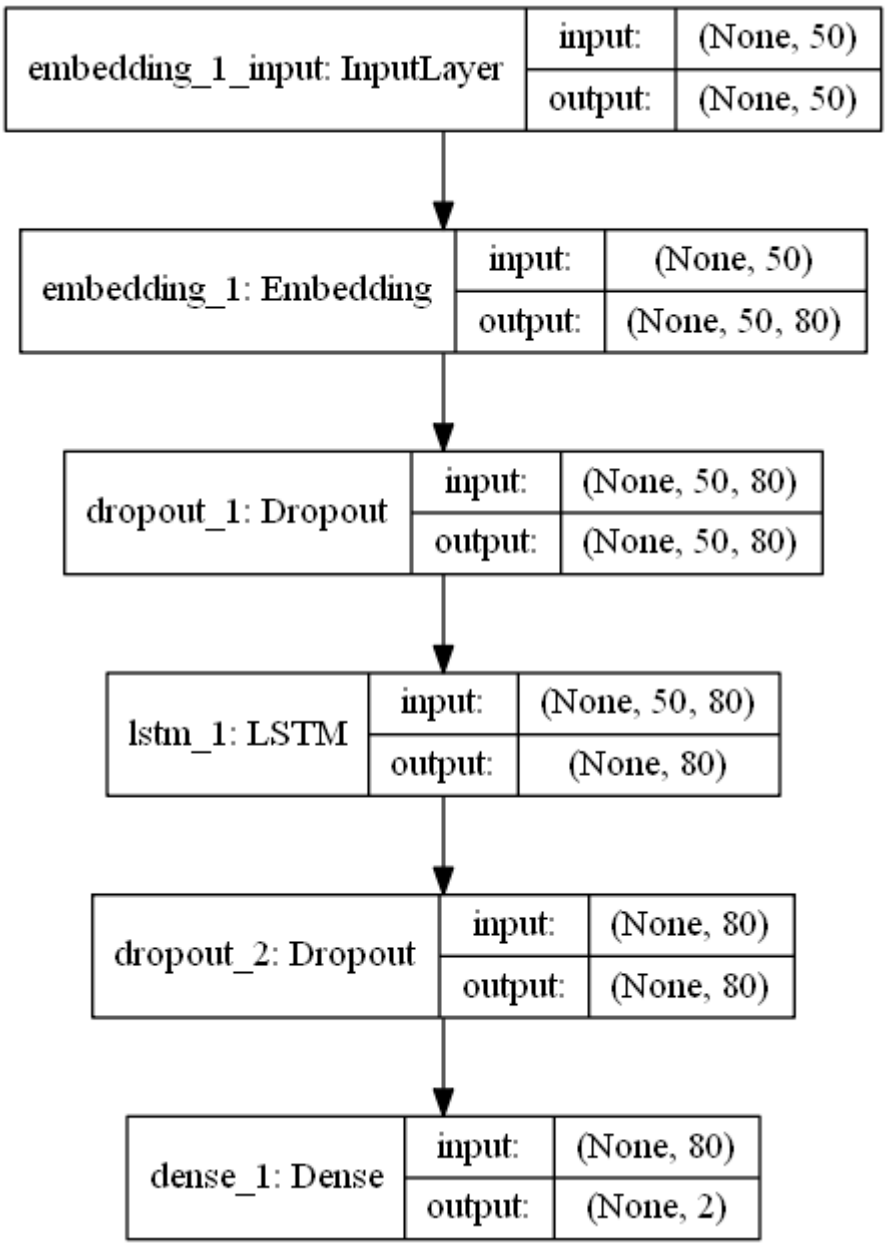
Εικόνα 3. Παράδειγμα ενός νευρώνα μνήμης LSTM, από το ερευνητικό κείμενο (Hochreiter, Schmidhuber, 1997)

Με το πέρας των χρόνων, τα νευρωνικά δίκτυα LSTM έχουν αποδείξει τις δυνατότητες τους για χρήση σε μεγάλο αριθμό πεδίων εφαρμογής. Ένα από αυτά είναι η επεξεργασία

φυσικής γλώσσας (NLP), με πολλές εφαρμογές τις οποίες βλέπουμε και χρησιμοποιούμε καθημερινά, όπως Google Translate (Wu et al., 2016), Siri (Levy, 2019) και Amazon Alexa (Naik et al., 2018).

Η δημοτικότητα των LSTM και η αντοχή τους σε προβλήματα vanishing gradient problem και exploding gradient problem αποτελεί τον κύριο λόγο που προχωρήσαμε στην χρήση τους για την ταξινόμηση μηνυμάτων κοινωνικών δικτύων βάση του συναισθήματος που προβάλλει ο χρήστης.

Για την υλοποίηση του μοντέλου ταξινόμησης LSTM, χρησιμοποιήσαμε την βιβλιοθήκη keras. Η υλοποίηση του νευρωνικού μας δικτύου είναι ως ακολούθως:



Διάγραμμα 13. Η δομή του νευρωνικού δικτύου με LSTM η οποία χρησιμοποιήθηκε για

Το νευρωνικό δίκτυο αποτελείται από 6 στρώματα:

1. Το στρώμα εισόδου δέχεται διανύσματα 50 θέσεων με ακέραιους αριθμούς τα οποία προωθούνται ως έχει στο επόμενο επίπεδο.
2. Το στρώμα embedding δέχεται διανύσματα 50 θέσεων με ακέραιους αριθμούς από το στρώμα εισόδου και τα μετατρέπει σε διανύσματα μεγέθους 80 θέσεων με δεκαδικούς αριθμούς τα οποία προωθούνται στο στρώμα Dropout.
3. Το στρώμα Dropout μηδενίζει τυχαία το 20% των δεδομένων και τα προωθεί στο δίκτυο LSTM.
4. Το στρώμα LSTM παίρνει διανύσματα 80 θέσεων, τα επεξεργάζεται και τα μεταφέρει στο δεύτερο στρώμα Dropout.
5. Το δεύτερο στρώμα Dropout μηδενίζει τυχαία το 20% των δεδομένων που έχουν υπολογιστεί και τα προωθεί σε πλήρες συνδεδεμένο στρώμα νευρώνων.
6. Το πλήρως συνδεδεμένο στρώμα νευρώνων αποτελείται από l νευρώνες, όπου l είναι ο συνολικός αριθμός των κλάσεων που περιλαμβάνονται στο dataset μας. Για τον υπολογισμό της εξόδου, εκτελείται η συνάρτηση ενεργοποίησης softmax.

$$S(y_i) = \frac{e^{y_i}}{\sum_j e^{y_j}} \quad (2)$$

Η εξίσωση υπολογισμού softmax.

Η χρήση των στρωμάτων dropout είναι αναγκαία καθώς ένα από τα προβλήματα που προκύπτουν κατά την χρήση του δικτύου LSTM είναι το γεγονός ότι το δίκτυο έχει την τάση να υπερ-προσαρμόζεται πολύ γρήγορα στα δεδομένα τα οποία εκπαιδεύεται. Αυτό έχει ως αποτέλεσμα την μειωμένη δυνατότητα γενίκευσης του σε άγνωστα δεδομένα, κάτι το οποίο είναι ύψιστης σημασίας για το σκοπό χρήσης μας. Για να αντιμετωπίσουμε αυτό το πρόβλημα, χρησιμοποιούμε το στρώμα Dropout το οποίο αφαιρεί μέρος των πληροφοριών που έχουν υπολογιστεί/μαθευτεί, με στόχο να δυσκολέψει το LSTM από το να προσαρμοστεί στα δεδομένα.

Για την τροφοδότηση του δικτύου με δεδομένα, χρησιμοποιήθηκε ο μηχανισμός Tokenizer() που παρέχεται από το Keras. Ο μηχανισμός αυτός παίρνει μια λίστα από

προτάσεις, τις σπάει στα συστατικά τους (δηλ. λέξεις που τις αποτελούν), καθορίζει σε κάθε λέξη ένα μοναδικό ακέραιο αριθμό και κτίζει ένα λεξικό με όλες τις λέξεις που εμφανίζονται. Ακολούθως, επιστρέφει μια λίστα από λίστες με τους μοναδικούς ακέραιους αριθμός που απαρτίζουν την κάθε πρόταση που δόθηκε. Επειδή αυτό το λεξικό δύναται να είναι πολύ μεγάλο, υπάρχει η δυνατότητα περιορισμού των προτάσεων που περιέχονται σε αυτό στις *X* πιο συχνά χρησιμοποιούμενες λέξεις. Για τους σκοπούς της μεταπτυχιακής αυτής διατριβής, έχουμε επιλέξει μέγιστο μέγεθος λεξικού στις 20000 λέξεις. Μετά την επιστροφή λίστας αριθμών για κάθε πρόταση, μεγαλώνουμε όλες τις λίστες αριθμών σε μήκος 50 κελιών ώστε να έχουμε ίσο μήκος σε όλες τις λίστες. Τα νέα κελιά που δημιουργούνται στην λίστα έχουν τιμή 0. Το πιο κάτω παράδειγμα παρουσιάζει το τρόπο λειτουργίας αυτού του αλγορίθμου.

Παράδειγμα τροφοδότησης νευρωνικού δικτύου LSTM:

Βήμα 1: Έστω ότι έχουμε τις προτάσεις

(3) *This sentence is to test the keras tokenizer*

(4) *The keras tokenizer is what we use to feed the LSTM RNN*

Βήμα 2: Οι προτάσεις μετατρέπονται σε μικρά γράμματα και σπάνε στις λέξεις που τις απαρτίζουν και ακολούθως δημιουργείται το λεξικό

this	sentence	is	to	test	the	keras	tokenizer
the	keras	tokenizer	is	what	we	use	to	feed	the	lstm	rnn

Word	Index
this	1
sentence	2
is	3
to	4
test	5
the	6
keras	7
tokenizer	8
what	9
we	10

use	11
feed	12
lstm	13
rnn	14

Βήμα 4: Επιστρέφονται οι προτάσεις μετετρεμμένες σε λίστα αριθμών

1	2	3	4	5	6	7	8				
6	7	8	3	9	10	11	4	12	6	13	14

Βήμα 5: Αλλάζουμε το μήκος της λίστας αριθμών που αντιστοιχεί σε κάθε πρόταση ώστε όλες οι προτάσεις να έχουν το ίδιο μήκος, π.χ. 12 για το παράδειγμα αυτό

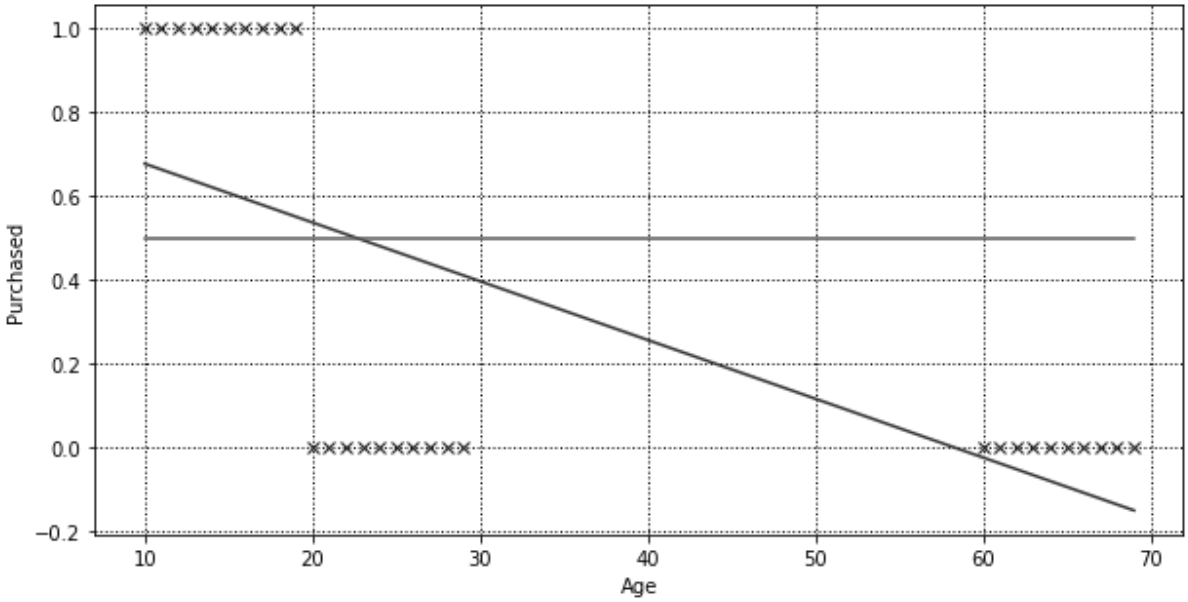
0	0	0	0	1	2	3	4	5	6	7	8
6	7	8	3	9	10	11	4	12	6	13	14

6.1.2 Λογιστική Παλινδρόμηση

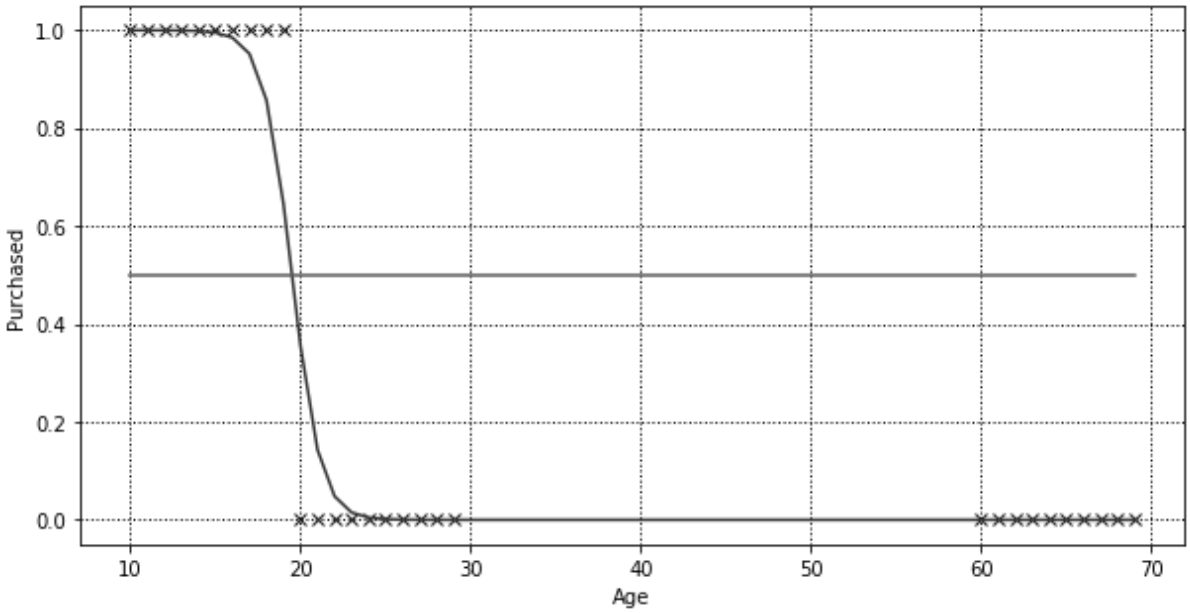
Η στατιστική μέθοδος λογιστικής παλινδρόμησης αποτελεί ειδική μορφή της γραμμικής παλινδρόμησης. Σε αντίθεση με την γραμμική παλινδρόμηση, η λογιστική παλινδρόμηση χρησιμοποιεί τη σιγμοειδή συνάρτηση, δίνοντας τις τιμές από 0 μέχρι και 1. Αυτό επιτρέπει στην λογιστική παλινδρόμηση να χρησιμοποιηθεί για σκοπούς ταξινόμησης δύο κλάσεων ή υπολογισμού πιθανότητας εκτέλεσης ενός συμβάντος.

(5) $\boldsymbol{y = ax + b}$

(6) $\boldsymbol{y = 1/(1 + e\text{^}(-x))}$



Εικόνα 4. Εφαρμογή γραμμικής παλινδρόμησης σε δεδομένα αγορών ανά ηλικία πελάτη. (Jing, 2019)



Εικόνα 5. Εφαρμογή λογιστικής παλινδρόμησης σε δεδομένα αγορών ανά ηλικία πελάτη. (Jing, 2019)

Στην εξίσωση 2 παρουσιάζεται η συνάρτηση για τη γραμμική παλινδρόμηση και στην εξίσωση 3 η συνάρτηση της λογιστικής παλινδρόμησης. Στην εικόνα 3 φαίνεται η εφαρμογή της γραμμικής παλινδρόμησης σε μη-ισορροπημένα δεδομένα πελατών που εκτέλεσαν ή όχι (1 ή 0) αγορές. Στην εικόνα 4 φαίνεται η εφαρμογή της λογιστικής παλινδρόμησης στο ίδιο σύνολο δεδομένων. Παρατηρούμε πως η λογιστική

παλινδρόμηση έχει καλύτερη εφαρμογή και δεν επηρεάζεται από το μη ισορροπημένο σύνολο δεδομένων.

Πρέπει να σημειωθεί πως υπάρχουν και πολυωνυμικές υλοποιήσεις της λογιστικής παλινδρόμησης, οι οποίες επιτρέπουν την ταξινόμηση πολλαπλών κλάσεων. Για τους σκοπούς αυτής της μεταπτυχιακής διατριβή, χρησιμοποιήσαμε την υλοποίηση της λογιστικής παλινδρόμησης από τη βιβλιοθήκη Scikit-learn.

6.1.3 Naïve Bayes

Το μοντέλο μηχανικής μάθησης Naïve Bayes είναι βασισμένο στο θεώρημα του Bayes. Ο λόγος που ονομάζεται Naïve ή Αφελής είναι γιατί θεωρεί ότι τα διάφορα χαρακτηριστικά που του δίνονται για να χαρακτηρίσουν κάποια κλάση δεδομένων, είναι ανεξάρτητα μεταξύ τους. Ακόμη και με αυτή την απλοποίηση, ο ταξινομητής Naïve Bayes έχει αποδείξει την αξία του σε πολλές εφαρμογές, όπως για παράδειγμα αναγνώριση spam.

Ένα από τα πλεονεκτήματα του είναι ότι είναι γρήγορος και απλός στην υλοποίηση.

Το θεώρημα του Bayes χρησιμοποιείται για να υπολογίσει την πιθανότητα να ισχύει το A δεδομένου ότι ισχύει το B.

$$(7) \qquad P(A|B) = \frac{P(A)P(B|A)}{P(B)} = \frac{P(A \cap B)}{P(B)}$$

Επεκτείνοντας το στο πρόβλημα που θέλουμε να επιλύσουμε, θέλουμε να υπολογίσουμε την πιθανότητα για ένα μήνυμα να ισχύει στο συναίσθημα y δεδομένου κάποιων εισόδων $x_1, \cdots, x_n$ που προέρχονται από το αποτέλεσμα εκτέλεσης αλγορίθμου One-Hot ή TF-IDF στο μήνυμα.

$$(8) \qquad P(y|x_1, \cdots, x_n) = \frac{P(y)P(x_1, \cdots, x_n|y)}{P(x_1, \cdots, x_n)}$$

Η αφελής υπόθεση του Bayes θεωρεί ότι τα x είναι ανεξάρτητα μεταξύ τους, οπότε υπολογίζουμε:

$$(9) \qquad P(x_i|y, x_1, \cdots, x_{i-1}, x_{i+1}, \cdots, x_n) = P(x_i|y)$$

Αντικαθιστώντας στην εξίσωση μας καταλήγουμε:

$$P(y|x_1, \cdots, x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i|y)}{P(x_1, \cdots, x_n)}$$

Επίσης, καθώς το $P(x_1, \cdots, x_n)$ είναι σταθερά μπορούμε να υπολογίσουμε ότι:

$$P(y|x_1, \cdots, x_n) \propto P(y)\prod_{i=1}^{n} P(x_i|y)$$

Για να μετατρέψουμε την εξίσωση σε κανόνα ταξινόμησης χρησιμοποιούμε τον κανόνα maximum a posteriori για τον υπολογισμό των $P(y)$ και $P(x_i|y)$.

$$\hat{y} = \underset{y}{\operatorname{argmax}}\, P(y)\prod_{i=1}^{n} P(x_i|y)$$

Σε αυτή την μεταπτυχιακή διατριβή, χρησιμοποιήσαμε 2 είδη ταξινομητών τύπου Naïve Bayes από τη βιβλιοθήκη Scikit-learn. Συγκεκριμένα, χρησιμοποιήσαμε τους ταξινομητής: Multinomial Naïve Bayes και Gaussian Naïve Bayes. Η διαφορά μεταξύ των δυο ταξινομητών είναι στο πως χαρακτηρίζουν την τιμή του $P(x_i|y)$

Multinomial Naïve Bayes

Ο ταξινομητής Multinomial Naïve Bayes χρησιμοποιείται κυρίως για ταξινόμηση κειμένων.

Complement Naïve Bayes

Ο ταξινομητής Complement Naïve Bayes αποτελεί βελτιωμένη έκδοση του Multinomial Naïve Bayes, και χρησιμοποιείται κυρίως για ταξινόμηση κειμένων όπου η συλλογή δεδομένων που χρησιμοποιείται δεν είναι ισορροπημένη. Οι (Rennie et al., 2003) αναφέρουν ότι οι βελτιώσεις που προσφέρει ο Complement Naïve Bayes, φέρνουν τον αλγόριθμο Naïve Bayes πιο κοντά στην πραγματικότητα της καθημερινότητας και του επιτρέπουν να έχει καλύτερη απόδοση.

Gaussian Naïve Bayes

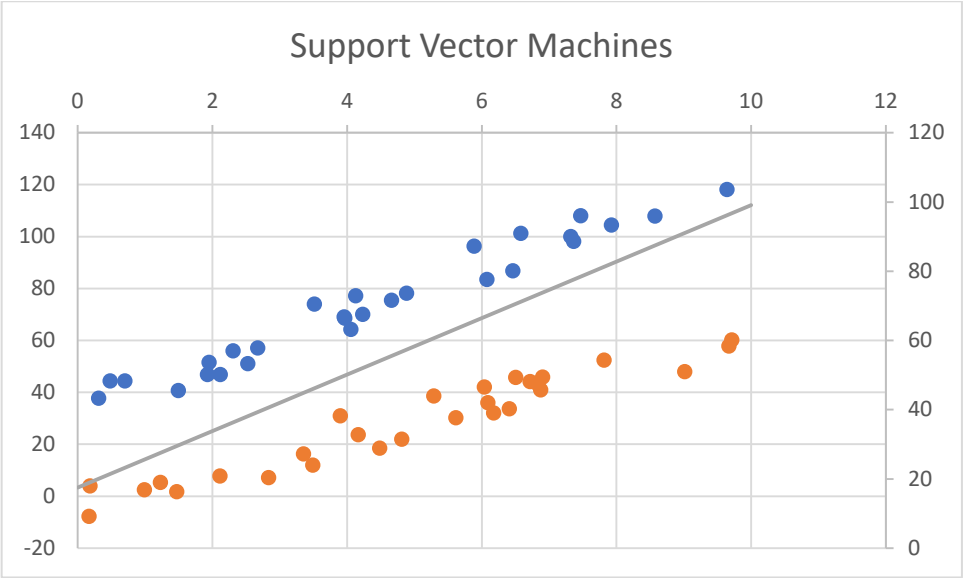
Ο ταξινομητής Gaussian Naïve Bayes θεωρεί ότι τα δεδομένα τα οποία του δίδονται κατανέμονται βάση της κανονικής κατανομής. Η τιμή του $P(x_i|y)$ καθορίζεται ως

$$P(x_i|y) = \frac{1}{\sqrt{2\pi\sigma_y^2}} e^{\left(-\frac{(x_i-\mu_y)^2}{\sigma_y^2}\right)}$$

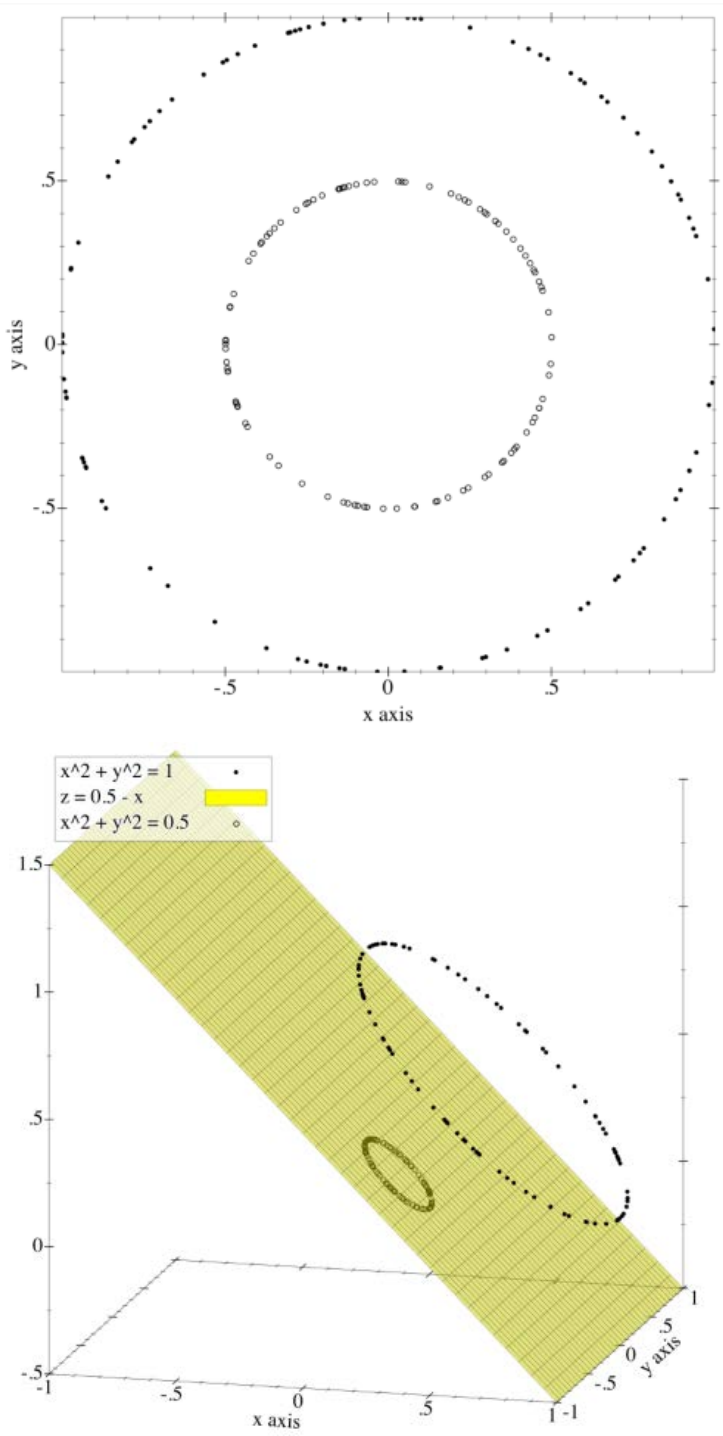
Όπου οι παράμετροί σ_y(διακύμανση των τιμών x της κλάσης y) και μ_y (μέσος των τιμών x της κλάσης y) υπολογίζονται χρησιμοποιώντας εκτίμηση μέγιστης πιθανότητας.

6.1.4 Support Vector Machines

Τα Support Vector Machines αποτελούν μια μέθοδο μηχανικής μάθησης η οποία διαχωρίζει δεδομένα μέσα στο πολυδιάστατο χώρο χρησιμοποιώντας επίπεδα ή γραμμές. Τα SVM έχουν ως στόχο να εντοπίσουν το μέγιστο όριο διαχωρισμού μεταξύ των κλάσεων, καταλήγοντας στη βέλτιστη λύση. Σε αντίθεση με άλλες μεθόδους μηχανικής μάθησης (όπως π.χ. Κ-Κοντινότεροι Γείτονες), τα SVM έχουν το πλεονέκτημα ότι μπορούν να διαχωρίσουν δεδομένα ακόμη και όταν ο αριθμός των διαστάσεων του προβλήματος (δηλ. ο αριθμός των στοιχείων που χαρακτηρίζουν το κάθε σημείο που θέλουμε να διαχωρίσουμε) είναι τεράστιος. Αυτό συμβαδίζει με τις απαιτήσεις μας, καθώς για το πρόβλημα που επιλύουμε, τα δεδομένα μας μπορεί να έχουν έως και 50 διαστάσεις (ο μέγιστος αριθμός λέξεων που έχουμε καθορίσει ανά tweet).



Διάγραμμα 14. Παράδειγμα διαχωρισμού δυο κατηγοριών δεδομένων χρησιμοποιώντας SVM.



Διάγραμμα 15. Στην περίπτωση που το πρόβλημα μας δεν μπορεί να επιλυθεί σε δυσδιάστατο χώρο, τότε τα SVM εφαρμόζουν μεθόδους "kernel trick" ώστε να διαχωρίσουν τα δεδομένα σε υψηλότερη διάσταση. Όπως βλέπουμε, ο διαχωρισμός στην πιο πάνω περίπτωση έγινε σε τρισδιάστατο χώρο χρησιμοποιώντας επίπεδο, ενώ τα δεδομένα μας ήταν δυσδιάστατα. Οι πιο πάνω εικόνες πάρθηκαν από το βιβλίο Thoughtful Machine Learning: A Test-Driven Approach (Kirk, 2014)

6.2 Μη-Επιβλεπόμενα Μοντέλα Μηχανική Μάθησης

6.2.1 Emotion Lexicon Classifier

Βασισμένοι πάνω στη συλλογή δεδομένων λεξικού NRC Emotion Lexicon (Mohammad, Saif, Turney, 2010, Mohammad, Saif M., Turney, 2013) που περιγράφεται στην ενότητα 4.4, δημιουργήσαμε ένα μη-επιβλεπόμενο ταξινομητή συναισθήματος Emotion Lexicon Classifier (ELC).

Ο ELC δέχεται μια λίστα μηνυμάτων τα οποία έχουν περάσει από την μέθοδο προ-επεξεργασίας που περιγράφεται στο κεφάλαιο 5 και ακολούθως χρησιμοποιεί τη μέθοδο word_tokenizer που παρέχεται από τη βιβλιοθήκη NLTK για να τα διασπάσει στις λέξεις που τα απαρτίζουν. Μετά, αρχικοποιεί ένα score card για κάθε μήνυμα και εξετάζει μια-μια κάθε λέξη του μηνύματος. Εάν η λέξη υπάρχει μέσα στο λεξικό, τότε για κάθε συναίσθημα που εφαρμόζεται στην λέξη, προστίθεται ένας βαθμός στο score card του μηνύματος. Στην περίπτωση που η πρόταση έχει δείκτη αρνητικότητας "_NEG", τότε αφαιρείται ένας βαθμός από το score card. Στο τέλος της διαδικασίας, ελέγχεται το score card και το συναίσθημα με το υψηλότερο score είναι το συναίσθημα το οποίο χαρακτηρίζει το μήνυμα. Στην περίπτωση που υπάρχει ισοβαθμία, τότε το συναίσθημα που επιστρέφεται είναι αυτό με την μεγαλύτερη δημοτικότητα στο λεξικό.

Παράδειγμα:

(10) *"I love candy! It's yummy!"*

Η πρόταση πριν την προ-επεξεργασία.

(11) *love candy yummy*

Η πρόταση μετά την προ-επεξεργασία.

(12) *["love", "candy", "yummy"]*

Η πρόταση ως λίστα λέξεων μετά το word_tokenize εντός του ELC.

Συναισθήματα για λέξη “candy”	Συσχέτιση
anger	0
fear	0
joy	0
neutral	0
sadness	0

Συναισθήματα για λέξη “love”	Συσχέτιση
anger	0
fear	0
joy	1
neutral	0
sadness	0

Πίνακας 17. Η συσχέτιση των συναισθημάτων για τις λέξεις “candy” και “love”. Η λέξη “yummy” δεν συμπεριλαμβάνεται στο λεξικό.

Συναίσθημα	Σκορ
anger	0
fear	0
joy	1
neutral	0
sadness	0

Πίνακας 18. Το score card της πρότασης μας. Καθώς το συναίσθημα joy έχει το ψηλότερο σκορ, τότε το ELC επιστρέφει ότι η πρόταση εμπεριέχει συναίσθημα joy.

Συναίσθημα	Σκορ
anger	0
fear	0
joy	-1
neutral	0
sadness	0

Πίνακας 19. Στην περίπτωση που η λέξη “candy” είχε άρνηση, δηλαδή ήταν “candy_NEG”, τότε θα αφαιρούσαμε 1 από το score card αντί να προσθέσουμε. Στην περίπτωση αυτή, το συναίσθημα sadness, το οποίο έχει την πλειοψηφία στον λεξικό θα συσχετιζόταν με την πρόταση.

Κεφάλαιο 7
Αποτελέσματα

7.1 Ανάλυση Αποτελεσμάτων

Για τον έλεγχο των μοντέλων μηχανικής μάθησης που έχουμε , θα τρέξουμε 5 σενάρια πλήρους εκπαίδευσης και ελέγχου.

Σενάριο 1: Εκπαίδευση Χωρίς Balancing

Χαρακτηριστικά συνόλου δεδομένων εκπαίδευσης:

Emotion	Presence	Absence
Anger	4768	20871
Sadness	5171	20468
Fear	3882	21757
Joy	8173	17466

Οι Βέλτιστοι ταξινομητές ανά συναίσθημα βάση του F1-Score για την minority κλάση είναι (Δείτε υποενότητα 7.1.2 για πλήρη ανάλυση):

Emotion	Classifier	F1-Score
Anger	LSTM	0.77
Fear	LSTM	0.81
Joy	LSTM	0.75
Sadness	GNB	0.51

Παρατηρήσαμε πλήρη αδυναμία ταξινόμησης για τα μοντέλα MNB και SVM.

Σενάριο 2: Balancing κλάσεων εκπαίδευσης με over sampling του minority class χρησιμοποιώντας τον αλγόριθμο SMOTE

Χαρακτηριστικά συνόλου δεδομένων εκπαίδευσης πριν και μετά το SMOTE:

Emotion	Initial Dataset		After SMOTE oversampling on minority class	
	Presence	Absence	Presence	Absence
Anger	4768	20871	20871	20871
Sadness	5171	20468	20468	20468
Fear	3882	21757	21757	21757
Joy	8173	17466	17466	17466

Βέλτιστοι ταξινομητές ανά συναίσθημα βάση του F1-Score για την minority κλάση (Δείτε υποενότητα 7.1.3 για πλήρη ανάλυση):

Emotion	Classifier	F1-Score
Anger	CNB/MNB	0.60
Fear	CNB/MNB	0.61
Joy	CNB/MNB	0.65
Sadness	CNB/MNB	0.67

Σενάριο 3: Balancing κλάσεων εκπαίδευσης με χρήση undersampling του majority class

Χαρακτηριστικά συνόλου δεδομένων εκπαίδευσης πριν και μετά το under sampling:

Emotion	Initial Dataset		After random under sampling on majority class	
	Presence	Absence	Presence	Absence
Anger	4768	20871	4768	4768
Sadness	5171	20468	5171	5171
Fear	3882	21757	3882	3882
Joy	8173	17466	8173	8173

Βέλτιστοι ταξινομητές ανά συναίσθημα βάση του F1-Score για την minority κλάση (Δείτε υποενότητα 7.1.4 για πλήρη ανάλυση):

Emotion	Classifier	F1-Score
Anger	CNB/MNB	0.68
Fear	CNB/MNB	0.77
Joy	Logistic Regression	0.72
Sadness	CNB/MNB	0.62

Σενάριο 4: Χρήση μεθόδου Word2Vec για αρχικοποίηση των αρχικών βαρών του LSTM

Παρατηρήσαμε πλήρης αποτυχία από το LSTM για ταξινόμηση. Δείτε υποενότητα 7.1.5 για πλήρη ανάλυση.

Σενάριο 5: Χρήση n-grams στους αλγορίθμους του scikit-learn

Χαρακτηριστικά συνόλου δεδομένων εκπαίδευσης (Δείτε υποενότητα 7.1.6 για πλήρη ανάλυση):

Emotion	Presence	Absence
Anger	4768	20871
Sadness	5171	20468
Fear	3882	21757
Joy	8173	17466

Βέλτιστοι ταξινομητές ανά συναίσθημα βάση του F1-Score για την minority κλάση:

Emotion	Classifier	F1-Score
Anger	Emotion Lexicon Classifier	0.28
Fear	Emotion Lexicon Classifier	0.31
Joy	Complement Naïve Bayes	0.56
Sadness	Emotion Lexicon Classifier	0.68

Γενικά Σχόλια

Σε μερικά σενάρια παρατηρήσαμε χαμηλά F1-Score, ασχέτως του μεγέθους ή του balance του συνόλου δεδομένων. Αυτό υποδηλώνει ότι ίσως να χρειάζεται περισσότερο tuning των μοντέλων μηχανικής μάθησης. Για να επιλέξουμε μεταξύ καλού true positive rate,

εστιάσαμε στην μεγιστοποίηση το F1-score για την minority class (anger, fear, joy, sadness).

Πρέπει να αναφερθεί ότι τα F1-score που παρατηρήσαμε για τα minority classes πλησιάζουν στα F1-score που παρατηρήθηκαν από τους Tan et al. (2019), οι οποίοι υποστήριξαν ότι η μείωση του F1-score οφείλεται στην ύπαρξη πολλών false positives, κάτι το οποίο είναι αναμενόμενο χαρακτηριστικό για ένα σύστημα τέτοιου είδους.

7.1.1 Μετρήσεις

Για να επιλέξουμε και να αξιολογήσουμε τον καλύτερο ταξινομητή πρέπει πρώτα να ορίσουμε τις μετρήσεις που θα μελετήσουμε για να καταλήξουμε στο συμπέρασμα μας. Συνολικά, έχουμε 4 είδη μετρήσεων, το Accuracy, το Precision, το Recall και το F1-score.

$$(13) \qquad accuracy = \frac{True\ Positive + True\ Negative}{True\ Positive + False\ Positive + True\ Negative + False\ Negative}$$

$$(14) \qquad precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

Όταν έχουμε false positive το precision πέφτει.

$$(15) \qquad recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Όταν έχουμε false negative το recall πέφτει.

$$(16) \qquad f1 = \frac{2 * recall * precision}{recall + precision}$$

Το f1-score συνδυάζει χαρακτηριστικά των recall, precision

Επειδή θέλουμε να μειώσουμε την πιθανότητα να έχουμε False Negative και False Positive, πρέπει να εστιάσουμε στην μέτρηση του score f1.

7.1.2 Εκπαίδευση με ανισόρροπο αριθμών δεδομένων ανά κλάση

Ταξινομητής	Κλάση	Ταξινομήθηκε ως		Ακρίβεια	Ανάκληση	F1-Score
		anger	not_anger			
Complement Naive Bayes	anger	617	2562	0.33	0.19	0.25
	not_anger	1240	12675	0.83	0.91	0.87
	Ζυγισμένος Μέσος	-	-	0.74	0.78	0.75
Emotion Lexicon Classifier	anger	750	2429	0.31	0.24	0.27
	not_anger	1670	12245	0.83	0.88	0.86
	Ζυγισμένος Μέσος	-	-	0.74	0.76	0.75
Gaussian Naive Bayes	anger	1589	1590	0.18	0.50	0.26
	not_anger	7450	6465	0.80	0.46	0.59
	Ζυγισμένος Μέσος	-	-	0.69	0.47	0.53
Logistic Regression	anger	282	2897	0.53	0.09	0.15
	not_anger	250	13665	0.83	0.98	0.90
	Ζυγισμένος Μέσος	-	-	0.77	0.82	0.76
LSTM	anger	812	2367	0.39	0.26	0.31
	not_anger	1297	12618	0.84	0.91	0.87
	Ζυγισμένος Μέσος	-	-	0.76	0.79	0.77
Multinomial Naive Bayes	anger	15	3164	0.22	0.00	0.01
	not_anger	54	13861	0.81	1.00	0.90
	Ζυγισμένος Μέσος	-	-	0.70	0.81	0.73
Support Vector Machine	anger	3	3176	0.43	0.00	0.00
	not_anger	4	13911	0.81	1.00	0.90
	Ζυγισμένος Μέσος	-	-	0.74	0.81	0.73

Πίνακας 20. Αποτελέσματα εκπαίδευσης ταξινομητών για το συναίσθημα θυμού χωρίς ισορρόπηση του αριθμού κλάσεων.

Ταξινομητής	Κλάση	Ταξινομήθηκε ως		Ακρίβεια	Ανάκληση	F1-Score
		fear	not_fear			
Complement Naive Bayes	fear	494	2095	0.32	0.19	0.24
	not_fear	1028	13477	0.87	0.93	0.90
	Ζυγισμένος Μέσος	-	-	0.78	0.82	0.80
Emotion Lexicon Classifier	fear	619	1970	0.25	0.24	0.24
	not_fear	1905	12600	0.86	0.87	0.87
	Ζυγισμένος Μέσος	-	-	0.77	0.77	0.77
Gaussian Naive Bayes	fear	1294	1295	0.16	0.50	0.24
	not_fear	6765	7740	0.86	0.53	0.66
	Ζυγισμένος Μέσος	-	-	0.75	0.53	0.59
Logistic Regression	fear	225	2364	0.59	0.09	0.15
	not_fear	156	14349	0.86	0.99	0.92
	Ζυγισμένος Μέσος	-	-	0.82	0.85	0.80
LSTM	fear	684	1905	0.40	0.26	0.32
	not_fear	1037	13468	0.88	0.93	0.90
	Ζυγισμένος Μέσος	-	-	0.80	0.83	0.81
Multinomial Naive Bayes	fear	63	2526	0.62	0.02	0.05
	not_fear	38	14467	0.85	1.00	0.92
	Ζυγισμένος Μέσος	-	-	0.82	0.85	0.79
Support Vector Machine	fear	44	2545	0.83	0.02	0.03
	not_fear	9	14496	0.85	1.00	0.92
	Ζυγισμένος Μέσος	-	-	0.85	0.85	0.78

Πίνακας 21. Αποτελέσματα εκπαίδευσης ταξινομητών για το συναίσθημα φόβου χωρίς ισορρόπηση του αριθμού κλάσεων.

Ταξινομητής	Κλάση	Ταξινομήθηκε ως		Ακρίβεια	Ανάκληση	F1-Score
		joy	not_joy			
Complement Naive Bayes	joy	2758	2691	0.63	0.51	0.56
	not_joy	1626	10019	0.79	0.86	0.82
	Ζυγισμένος Μέσος	-	-	0.74	0.75	0.74
Emotion Lexicon Classifier	joy	2727	2722	0.45	0.50	0.48
	not_joy	3267	8378	0.75	0.72	0.74
	Ζυγισμένος Μέσος	-	-	0.66	0.65	0.65
Gaussian Naive Bayes	joy	3873	1576	0.37	0.71	0.48
	not_joy	6665	4980	0.76	0.43	0.55
	Ζυγισμένος Μέσος	-	-	0.63	0.52	0.53
Logistic Regression	joy	2299	3150	0.73	0.42	0.53
	not_joy	847	10798	0.77	0.93	0.84
	Ζυγισμένος Μέσος	-	-	0.76	0.77	0.75
LSTM	joy	2941	2508	0.63	0.54	0.58
	not_joy	1758	9887	0.80	0.85	0.82
	Ζυγισμένος Μέσος	-	-	0.74	0.75	0.75
Multinomial Naive Bayes	joy	1113	4336	0.77	0.20	0.32
	not_joy	341	11304	0.72	0.97	0.83
	Ζυγισμένος Μέσος	-	-	0.74	0.73	0.67
Support Vector Machine	joy	223	5226	0.94	0.04	0.08
	not_joy	15	11630	0.69	1.00	0.82
	Ζυγισμένος Μέσος	-	-	0.77	0.69	0.58

Πίνακας 22. Αποτελέσματα εκπαίδευσης ταξινομητών για το συναίσθημα χαράς χωρίς ισορρόπηση του αριθμού κλάσεων.

Ταξινομητής	Κλάση	Ταξινομήθηκε ως		Ακρίβεια	Ανάκληση	F1-Score
		not_sadness	sadness			
Complement Naive Bayes	not_sadness	12375	1272	0.81	0.91	0.85
	sadness	2995	452	0.26	0.13	0.17
	Ζυγισμένος Μέσος	-	-	0.70	0.75	0.72
Emotion Lexicon Classifier	not_sadness	10083	3564	0.81	0.74	0.77
	sadness	2297	1150	0.24	0.33	0.28
	Ζυγισμένος Μέσος	-	-	0.70	0.66	0.68
Gaussian Naive Bayes	not_sadness	6087	7560	0.79	0.45	0.57
	sadness	1599	1848	0.20	0.54	0.29
	Ζυγισμένος Μέσος	-	-	0.67	0.46	0.51
Logistic Regression	not_sadness	13381	266	0.81	0.98	0.89
	sadness	3162	285	0.52	0.08	0.14
	Ζυγισμένος Μέσος	-	-	0.75	0.80	0.74
LSTM	not_sadness	12804	843	0.81	0.94	0.87
	sadness	2928	519	0.38	0.15	0.22
	Ζυγισμένος Μέσος	-	-	0.73	0.78	0.74
Multinomial Naive Bayes	not_sadness	13587	60	0.80	1.00	0.89
	sadness	3439	8	0.12	0.00	0.00
	Ζυγισμένος Μέσος	-	-	0.66	0.80	0.71
Support Vector Machine	not_sadness	13643	4	0.80	1.00	0.89
	sadness	3446	1	0.20	0.00	0.00
	Ζυγισμένος Μέσος	-	-	0.68	0.80	0.71

Πίνακας 23. Αποτελέσματα εκπαίδευσης ταξινομητών για το συναίσθημα λύπης χωρίς ισορρόπηση του αριθμού κλάσεων.

7.1.3 Ισορρόπηση δεδομένων μέσω SMOTE over sampling στη κλάση μειονότητας

Ταξινομητής	Κλάση	Ταξινομήθηκε ως		Ακρίβεια	Ανάκληση	F1-Score
		anger	not_anger			
Complement Naive Bayes	anger	2249	930	0.25	0.71	0.37
	not_anger	6762	7153	0.88	0.51	0.65
	Ζυγισμένος Μέσος	-	-	0.77	0.55	0.60
Emotion Lexicon Classifier	anger	761	2418	0.32	0.24	0.27
	not_anger	1630	12285	0.84	0.88	0.86
	Ζυγισμένος Μέσος	-	-	0.74	0.76	0.75
Gaussian Naive Bayes	anger	2664	515	0.18	0.84	0.30
	not_anger	11738	2177	0.81	0.16	0.26
	Ζυγισμένος Μέσος	-	-	0.69	0.28	0.27
Logistic Regression	anger	1736	1443	0.27	0.55	0.36
	not_anger	4752	9163	0.86	0.66	0.75
	Ζυγισμένος Μέσος	-	-	0.75	0.64	0.68
LSTM	anger	3179	0	0.19	1.00	0.31
	not_anger	13915	0	0.00	0.00	0.00
	Ζυγισμένος Μέσος	-	-	0.03	0.19	0.06
Multinomial Naive Bayes	anger	2249	930	0.25	0.71	0.37
	not_anger	6762	7153	0.88	0.51	0.65
	Ζυγισμένος Μέσος	-	-	0.77	0.55	0.60
Support Vector Machine	anger	1804	1375	0.20	0.57	0.29
	not_anger	7300	6615	0.83	0.48	0.60
	Ζυγισμένος Μέσος	-	-	0.71	0.49	0.55

Πίνακας 24. Αποτελέσματα εκπαίδευσης για το συναίσθημα θυμού μετά από ισορρόπηση του αριθμού κλάσεων.

Ταξινομητής	Κλάση	Ταξινομήθηκε ως		Ακρίβεια	Ανάκληση	F1-Score
		fear	not_fear			
Complement Naive Bayes	fear	1751	838	0.20	0.68	0.31
	not_fear	6952	7553	0.90	0.52	0.66
	Ζυγισμένος Μέσος	-	-	0.79	0.54	0.61
Emotion Lexicon Classifier	fear	630	1959	0.25	0.24	0.25
	not_fear	1916	12589	0.87	0.87	0.87
	Ζυγισμένος Μέσος	-	-	0.77	0.77	0.77
Gaussian Naive Bayes	fear	2140	449	0.15	0.83	0.25
	not_fear	12317	2188	0.83	0.15	0.26
	Ζυγισμένος Μέσος	-	-	0.73	0.25	0.25
Logistic Regression	fear	1332	1257	0.22	0.51	0.31
	not_fear	4650	9855	0.89	0.68	0.77
	Ζυγισμένος Μέσος	-	-	0.79	0.65	0.70
LSTM	fear	2589	0	0.15	1.00	0.26
	not_fear	14505	0	0.00	0.00	0.00
	Ζυγισμένος Μέσος	-	-	0.02	0.15	0.04
Multinomial Naive Bayes	fear	1751	838	0.20	0.68	0.31
	not_fear	6952	7553	0.90	0.52	0.66
	Ζυγισμένος Μέσος	-	-	0.79	0.54	0.61
Support Vector Machine	fear	1293	1296	0.15	0.50	0.23
	not_fear	7231	7274	0.85	0.50	0.63
	Ζυγισμένος Μέσος	-	-	0.74	0.50	0.57

Πίνακας 25. Αποτελέσματα εκπαίδευσης για το συναίσθημα φόβου μετά από ισορρόπηση του αριθμού κλάσεων.

Ταξινομητής	Κλάση	Ταξινομήθηκε ως		Ακρίβεια	Ανάκληση	F1-Score
		joy	not_joy			
Complement Naive Bayes	joy	4051	1398	0.46	0.74	0.57
	not_joy	4722	6923	0.83	0.59	0.69
	Ζυγισμένος Μέσος	-	-	0.71	0.64	0.65
Emotion Lexicon Classifier	joy	2762	2687	0.45	0.51	0.48
	not_joy	3354	8291	0.76	0.71	0.73
	Ζυγισμένος Μέσος	-	-	0.66	0.65	0.65
Gaussian Naive Bayes	joy	4527	922	0.33	0.83	0.48
	not_joy	9052	2593	0.74	0.22	0.34
	Ζυγισμένος Μέσος	-	-	0.61	0.42	0.38
Logistic Regression	joy	3858	1591	0.48	0.71	0.57
	not_joy	4257	7388	0.82	0.63	0.72
	Ζυγισμένος Μέσος	-	-	0.71	0.66	0.67
LSTM	joy	5449	0	0.32	1.00	0.48
	not_joy	11645	0	0.00	0.00	0.00
	Ζυγισμένος Μέσος	-	-	0.10	0.32	0.15
Multinomial Naive Bayes	joy	4051	1398	0.46	0.74	0.57
	not_joy	4722	6923	0.83	0.59	0.69
	Ζυγισμένος Μέσος	-	-	0.71	0.64	0.65
Support Vector Machine	joy	3243	2206	0.41	0.60	0.49
	not_joy	4602	7043	0.76	0.60	0.67
	Ζυγισμένος Μέσος	-	-	0.65	0.60	0.61

Πίνακας 26. Αποτελέσματα εκπαίδευσης για το συναίσθημα χαράς μετά από ισορρόπηση του αριθμού κλάσεων.

Ταξινομητής	Κλάση	Ταξινομήθηκε ως		Ακρίβεια	Ανάκληση	F1-Score
		not_sadness	sadness			
Complement Naive Bayes	not_sadness	6715	6932	0.85	0.49	0.62
	sadness	1213	2234	0.24	0.65	0.35
	Ζυγισμένος Μέσος	-	-	0.73	0.52	0.57
Emotion Lexicon Classifier	not_sadness	10031	3616	0.81	0.74	0.77
	sadness	2308	1139	0.24	0.33	0.28
	Ζυγισμένος Μέσος	-	-	0.70	0.65	0.67
Gaussian Naive Bayes	not_sadness	2463	11184	0.79	0.18	0.29
	sadness	653	2794	0.20	0.81	0.32
	Ζυγισμένος Μέσος	-	-	0.67	0.31	0.30
Logistic Regression	not_sadness	9119	4528	0.84	0.67	0.74
	sadness	1786	1661	0.27	0.48	0.34
	Ζυγισμένος Μέσος	-	-	0.72	0.63	0.66
LSTM	not_sadness	0	13647	0.00	0.00	0.00
	sadness	0	3447	0.20	1.00	0.34
	Ζυγισμένος Μέσος	-	-	0.04	0.20	0.07
Multinomial Naive Bayes	not_sadness	6715	6932	0.85	0.49	0.62
	sadness	1213	2234	0.24	0.65	0.35
	Ζυγισμένος Μέσος	-	-	0.73	0.52	0.57
Support Vector Machine	not_sadness	6535	7112	0.81	0.48	0.60
	sadness	1490	1957	0.22	0.57	0.31
	Ζυγισμένος Μέσος	-	-	0.69	0.50	0.54

Πίνακας 27. Αποτελέσματα εκπαίδευσης για το συναίσθημα λύπης μετά από ισορρόπηση του αριθμού κλάσεων.

7.1.4 Ισορρόπηση Δεδομένων μέσω Random Under Sampling της Κλάσης Πλειονότητας

Ταξινομητής	Κλάση	Ταξινομήθηκε ως		Ακρίβεια	Ανάκληση	F1-Score
		anger	not_anger			
Complement Naive Bayes	anger	2206	973	0.30	0.69	0.42
	not_anger	5241	8674	0.90	0.62	0.74
	Ζυγισμένος Μέσος	-	-	0.79	0.64	0.68
Emotion Lexicon Classifier	anger	756	2423	0.32	0.24	0.27
	not_anger	1632	12283	0.84	0.88	0.86
	Ζυγισμένος Μέσος	-	-	0.74	0.76	0.75
Gaussian Naive Bayes	anger	2465	714	0.19	0.78	0.31
	not_anger	10265	3650	0.84	0.26	0.40
	Ζυγισμένος Μέσος	-	-	0.72	0.36	0.38
Logistic Regression	anger	2023	1156	0.31	0.64	0.42
	not_anger	4439	9476	0.89	0.68	0.77
	Ζυγισμένος Μέσος	-	-	0.78	0.67	0.71
LSTM	anger	0	3179	0.00	0.00	0.00
	not_anger	0	13915	0.81	1.00	0.90
	Ζυγισμένος Μέσος	-	-	0.66	0.81	0.73
Multinomial Naive Bayes	anger	2206	973	0.30	0.69	0.42
	not_anger	5241	8674	0.90	0.62	0.74
	Ζυγισμένος Μέσος	-	-	0.79	0.64	0.68
Support Vector Machine	anger	1635	1544	0.25	0.51	0.33
	not_anger	5024	8891	0.85	0.64	0.73
	Ζυγισμένος Μέσος	-	-	0.74	0.62	0.66

Πίνακας 28. Αποτελέσματα εκπαίδευσης για το συναίσθημα θυμού μετά από ισορρόπηση του αριθμού κλάσεων.

Ταξινομητής	Κλάση	Ταξινομήθηκε ως		Ακρίβεια	Ανάκληση	F1-Score
		fear	not_fear			
Complement Naive Bayes	fear	1783	806	0.26	0.69	0.38
	not_fear	4946	9559	0.92	0.66	0.77
	Ζυγισμένος Μέσος	-	-	0.82	0.66	0.71
Emotion Lexicon Classifier	fear	588	2001	0.24	0.23	0.23
	not_fear	1867	12638	0.86	0.87	0.87
	Ζυγισμένος Μέσος	-	-	0.77	0.77	0.77
Gaussian Naive Bayes	fear	1319	1270	0.21	0.51	0.30
	not_fear	4937	9568	0.88	0.66	0.76
	Ζυγισμένος Μέσος	-	-	0.78	0.64	0.69
Logistic Regression	fear	1632	957	0.30	0.63	0.41
	not_fear	3835	10670	0.92	0.74	0.82
	Ζυγισμένος Μέσος	-	-	0.82	0.72	0.75
LSTM	fear	0	2589	0.00	0.00	0.00
	not_fear	0	14505	0.85	1.00	0.92
	Ζυγισμένος Μέσος	-	-	0.72	0.85	0.78
Multinomial Naive Bayes	fear	1783	806	0.26	0.69	0.38
	not_fear	4946	9559	0.92	0.66	0.77
	Ζυγισμένος Μέσος	-	-	0.82	0.66	0.71
Support Vector Machine	fear	1622	967	0.20	0.63	0.31
	not_fear	6351	8154	0.89	0.56	0.69
	Ζυγισμένος Μέσος	-	-	0.79	0.57	0.63

Πίνακας 29. Αποτελέσματα εκπαίδευσης για το συναίσθημα φόβου μετά από ισορρόπηση του αριθμού κλάσεων.

Ταξινομητής	Κλάση	Ταξινομήθηκε ως		Ακρίβεια	Ανάκληση	F1-Score
		joy	not_joy			
Complement Naive Bayes	joy	3845	1604	0.52	0.71	0.60
	not_joy	3617	8028	0.83	0.69	0.75
	Ζυγισμένος Μέσος	-	-	0.73	0.69	0.70
Emotion Lexicon Classifier	joy	2727	2722	0.45	0.50	0.48
	not_joy	3267	8378	0.75	0.72	0.74
	Ζυγισμένος Μέσος	-	-	0.66	0.65	0.65
Gaussian Naive Bayes	joy	4377	1072	0.35	0.80	0.49
	not_joy	8063	3582	0.77	0.31	0.44
	Ζυγισμένος Μέσος	-	-	0.64	0.47	0.46
Logistic Regression	joy	3822	1627	0.54	0.70	0.61
	not_joy	3258	8387	0.84	0.72	0.77
	Ζυγισμένος Μέσος	-	-	0.74	0.71	0.72
LSTM	joy	0	5449	0.00	0.00	0.00
	not_joy	0	11645	0.68	1.00	0.81
	Ζυγισμένος Μέσος	-	-	0.46	0.68	0.55
Multinomial Naive Bayes	joy	3845	1604	0.52	0.71	0.60
	not_joy	3617	8028	0.83	0.69	0.75
	Ζυγισμένος Μέσος	-	-	0.73	0.69	0.70
Support Vector Machine	joy	3010	2439	0.48	0.55	0.51
	not_joy	3241	8404	0.78	0.72	0.75
	Ζυγισμένος Μέσος	-	-	0.68	0.67	0.67

Πίνακας 30. Αποτελέσματα εκπαίδευσης για το συναίσθημα χαράς μετά από ισορρόπηση του αριθμού κλάσεων.

Ταξινομητής	Κλάση	Ταξινομήθηκε ως		Ακρίβεια	Ανάκληση	F1-Score
		not_sadness	sadness			
Complement Naive Bayes	not_sadness	7680	5967	0.87	0.56	0.68
	sadness	1195	2252	0.27	0.65	0.39
	Ζυγισμένος Μέσος	-	-	0.75	0.58	0.62
Emotion Lexicon Classifier	not_sadness	10137	3510	0.81	0.74	0.78
	sadness	2314	1133	0.24	0.33	0.28
	Ζυγισμένος Μέσος	-	-	0.70	0.66	0.68
Gaussian Naive Bayes	not_sadness	3599	10048	0.83	0.26	0.40
	sadness	752	2695	0.21	0.78	0.33
	Ζυγισμένος Μέσος	-	-	0.70	0.37	0.39
Logistic Regression	not_sadness	9077	4570	0.86	0.67	0.75
	sadness	1475	1972	0.30	0.57	0.39
	Ζυγισμένος Μέσος	-	-	0.75	0.65	0.68
LSTM	not_sadness	0	13647	0.00	0.00	0.00
	sadness	0	3447	0.20	1.00	0.34
	Ζυγισμένος Μέσος	-	-	0.04	0.20	0.07
Multinomial Naive Bayes	not_sadness	7680	5967	0.87	0.56	0.68
	sadness	1195	2252	0.27	0.65	0.39
	Ζυγισμένος Μέσος	-	-	0.75	0.58	0.62
Support Vector Machine	not_sadness	5939	7708	0.83	0.44	0.57
	sadness	1196	2251	0.23	0.65	0.34
	Ζυγισμένος Μέσος	-	-	0.71	0.48	0.52

Πίνακας 31. Αποτελέσματα εκπαίδευσης για το συναίσθημα λύπης μετά από ισορρόπηση του αριθμού κλάσεων.

7.1.5 Χρήση αλγορίθμου Word2Vec σε LSTM

Συναίσθημα	Κλάση	Ταξινομήθηκε ως		Ακρίβεια	Ανάκληση	F1-Score
		Απουσία	Ύπαρξη			
Anger	Απουσία	13915	0	0.81	1.00	0.90
	Ύπαρξη	3179	0	0.00	0.00	0.00
	Ζυγισμένος Μέσος	-	-	0.66	0.81	0.73
Fear	Απουσία	14505	0	0.85	1.00	0.92
	Ύπαρξη	2589	0	0.00	0.00	0.00
	Ζυγισμένος Μέσος	-	-	0.72	0.85	0.78
Joy	Απουσία	11403	242	0.70	0.98	0.82
	Ύπαρξη	4854	595	0.71	0.11	0.19
	Ζυγισμένος Μέσος	-	-	0.70	0.70	0.62
Sadness	Απουσία	13647	0	0.00	0.00	0.00
	Ύπαρξη	3447	0	0.80	1.00	0.89
	Ζυγισμένος Μέσος	-	-	0.64	0.80	0.71

Πίνακας 32. Αποτελέσματα εκπαίδευσης LSTM χρησιμοποιώντας τον αλγόριθμο Word2Vec.

7.1.6 Χρήση 2-Grams

Ταξινομητής	Κλάση	Ταξινομήθηκε ως		Ακρίβεια	Ανάκληση	F1-Score
		anger	not_anger			
Complement Naive Bayes	anger	629	2550	0.33	0.20	0.25
	not_anger	1259	12656	0.83	0.91	0.87
	Ζυγισμένος Μέσος	-	-	0.74	0.78	0.75
Emotion Lexicon Classifier	anger	783	2396	0.33	0.25	0.28
	not_anger	1588	12327	0.84	0.89	0.86
	Ζυγισμένος Μέσος	-	-	0.74	0.77	0.75
Gaussian Naive Bayes	anger	1656	1523	0.18	0.52	0.27
	not_anger	7616	6299	0.81	0.45	0.58
	Ζυγισμένος Μέσος	-	-	0.69	0.47	0.52
Logistic Regression	anger	306	2873	0.56	0.10	0.16
	not_anger	244	13671	0.83	0.98	0.90
	Ζυγισμένος Μέσος	-	-	0.78	0.82	0.76
Multinomial Naive Bayes	anger	15	3164	0.17	0.00	0.01
	not_anger	73	13842	0.81	0.99	0.90
	Ζυγισμένος Μέσος	-	-	0.69	0.81	0.73
Support Vector Machine	anger	1	3178	0.14	0.00	0.00
	not_anger	6	13909	0.81	1.00	0.90
	Ζυγισμένος Μέσος	-	-	0.69	0.81	0.73

Πίνακας 33. Αποτελέσματα εκπαίδευσης για το συναίσθημα θυμού με χρήση 2-grams.

Ταξινομητής	Κλάση	Ταξινομήθηκε ως		Ακρίβεια	Ανάκληση	F1-Score
		fear	not_fear			
Complement Naive Bayes	fear	519	2070	0.33	0.20	0.25
	not_fear	1035	13470	0.87	0.93	0.90
	Ζυγισμένος Μέσος	-	-	0.79	0.82	0.80
Emotion Lexicon Classifier	fear	616	1973	0.25	0.24	0.24
	not_fear	1873	12632	0.86	0.87	0.87
	Ζυγισμένος Μέσος	-	-	0.77	0.78	0.77
Gaussian Naive Bayes	fear	1298	1291	0.16	0.50	0.24
	not_fear	6718	7787	0.86	0.54	0.66
	Ζυγισμένος Μέσος	-	-	0.75	0.53	0.60
Logistic Regression	fear	254	2335	0.61	0.10	0.17
	not_fear	165	14340	0.86	0.99	0.92
	Ζυγισμένος Μέσος	-	-	0.82	0.85	0.81
Multinomial Naive Bayes	fear	56	2533	0.60	0.02	0.04
	not_fear	37	14468	0.85	1.00	0.92
	Ζυγισμένος Μέσος	-	-	0.81	0.85	0.79
Support Vector Machine	fear	34	2555	0.83	0.01	0.03
	not_fear	7	14498	0.85	1.00	0.92
	Ζυγισμένος Μέσος	-	-	0.85	0.85	0.78

Πίνακας 34. Αποτελέσματα εκπαίδευσης για το συναίσθημα φόβου με χρήση 2-grams.

Ταξινομητής	Κλάση	Ταξινομήθηκε ως		Ακρίβεια	Ανάκληση	F1-Score
		joy	not_joy			
Complement Naive Bayes	joy	2775	2674	0.62	0.51	0.56
	not_joy	1713	9932	0.79	0.85	0.82
	Ζυγισμένος Μέσος	-	-	0.73	0.74	0.74
Emotion Lexicon Classifier	joy	2752	2697	0.46	0.51	0.48
	not_joy	3257	8388	0.76	0.72	0.74
	Ζυγισμένος Μέσος	-	-	0.66	0.65	0.66
Gaussian Naive Bayes	joy	3770	1679	0.36	0.69	0.48
	not_joy	6623	5022	0.75	0.43	0.55
	Ζυγισμένος Μέσος	-	-	0.63	0.51	0.52
Logistic Regression	joy	2296	3153	0.72	0.42	0.53
	not_joy	886	10759	0.77	0.92	0.84
	Ζυγισμένος Μέσος	-	-	0.76	0.76	0.74
Multinomial Naive Bayes	joy	1071	4378	0.76	0.20	0.31
	not_joy	336	11309	0.72	0.97	0.83
	Ζυγισμένος Μέσος	-	-	0.73	0.72	0.66
Support Vector Machine	joy	197	5252	0.88	0.04	0.07
	not_joy	28	11617	0.69	1.00	0.81
	Ζυγισμένος Μέσος	-	-	0.75	0.69	0.58

Πίνακας 35. Αποτελέσματα εκπαίδευσης για το συναίσθημα χαράς με χρήση 2-grams.

Ταξινομητής	Κλάση	Ταξινομήθηκε ως		Ακρίβεια	Ανάκληση	F1-Score
		not_sadness	sadness			
Complement Naive Bayes	not_sadness	12479	1168	0.81	0.91	0.86
	sadness	2991	456	0.28	0.13	0.18
	Ζυγισμένος Μέσος	-	-	0.70	0.76	0.72
Emotion Lexicon Classifier	not_sadness	10092	3555	0.81	0.74	0.77
	sadness	2309	1138	0.24	0.33	0.28
	Ζυγισμένος Μέσος	-	-	0.70	0.66	0.68
Gaussian Naive Bayes	not_sadness	6150	7497	0.79	0.45	0.58
	sadness	1592	1855	0.20	0.54	0.29
	Ζυγισμένος Μέσος	-	-	0.67	0.47	0.52
Logistic Regression	not_sadness	13426	221	0.81	0.98	0.89
	sadness	3171	276	0.56	0.08	0.14
	Ζυγισμένος Μέσος	-	-	0.76	0.80	0.74
Multinomial Naive Bayes	not_sadness	13602	45	0.80	1.00	0.89
	sadness	3433	14	0.24	0.00	0.01
	Ζυγισμένος Μέσος	-	-	0.69	0.80	0.71
Support Vector Machine	not_sadness	13645	2	0.80	1.00	0.89
	sadness	3447	0	0.00	0.00	0.00
	Ζυγισμένος Μέσος	-	-	0.64	0.80	0.71

Πίνακας 36. Αποτελέσματα εκπαίδευσης για το συναίσθημα λύπης με χρήση 2-grams.

Κεφάλαιο 8
Συμπεράσματα και Μελλοντικό Έργο

8.1 Συμπεράσματα

Κατά τη διάρκεια αυτής της μεταπτυχιακής διατριβής, εντοπίσαμε κενό στην επιστημονική έρευνα που μελετά το συναίσθημα το οποίο μεταδίδεται από την επικοινωνία χρηστών μέσω κοινωνικών δικτύων. Όπως υποδείξαμε και στην βιβλιογραφική μας αναφορά, ο εντοπισμός συναισθήματος που εκδηλώνεται από χρήστες σε μέσα κοινωνικής δικτύωσης αποτελεί μέθοδο για εντοπισμό επικείμενης εσωτερικής απειλής. Συγκεκριμένα, εντοπίσαμε πως υπήρχε έλλειψη ευκρίνειας καθώς η ανάλυση συνήθως βασιζόταν σε θετικό/αρνητικό συναίσθημα ή απλοποιημένη λίστα συναισθημάτων όπως η χαρά και η λύπη.

Έτσι λοιπόν, σε αντίθεση με άλλες παρόμοιες έρευνες, όπως για παράδειγμα (Sang-Sang Tan, Jin-Cheon Na & Duraisamy, 2019), έχουμε δημιουργήσει μοντέλα μηχανικής μάθησης τα οποία μπορούν να ανιχνεύσουν κείμενα για εντοπισμό συναισθηματικά φορτισμένης συνομιλίας με μεγαλύτερη ευκρίνεια, όπως ο θυμός, η λύπη, ο φόβος και η χαρά. Περαιτέρω, έχουμε βρει πως η ποιότητα των μοντέλων μηχανικής μάθησης που έχουμε κτίσει είναι ισάξια με αυτή των Tan et al. (2019).

Ακόμη, ως μέρος αυτής της μεταπτυχιακής διατριβής, προετοιμάσαμε το έδαφος για μελλοντικό έργο. Αυτό το πετύχαμε με την δημιουργία βάσης δεδομένων η οποία περιέχει δεδομένα από τη συλλογή δεδομένων Sentiment 140 (2019), που μπορούν να χρησιμοποιηθούν μαζί με τους ταξινομητές συναισθήματος που έχουμε κτίσει για περαιτέρω ανάπτυξη.

8.2 Μελλοντικό Έργο

Ως μελλοντικό έργο μπορεί να συνεχιστεί η μελέτη και ανάπτυξη των ταξινομητών συναισθήματος. Συγκεκριμένα μπορεί να γίνει περαιτέρω ρύθμιση των υπερπαραμέτρων ή/και χρήση βιβλιοθηκών όπως η GloVe για καλύτερο αποτέλεσμα.

Ακόμη, η βάση δεδομένων που έχει κτιστεί επιτρέπει την ανάλυση των δεδομένων του συνόλου δεδομένων Sentiment 140, το οποίο μπορεί να μελετηθεί περαιτέρω και να χρησιμοποιηθεί για προσομοίωση επιθέσεων εσωτερικής απειλής, χρησιμοποιώντας του ταξινομητές που έχουμε κτίσει καθώς και άλλες heuristic μεθόδους για εντοπισμό τους, όπως η αναγνώριση κακόβουλων URL ή η ανίχνευση αρνητικών hashtag.

Βιβλιογραφία

Alahmadi, B., Legg, P. & Nurse, J. 2015, "Using Internet Activity Profiling for Insider-Threat Detection", , 01.

Berkley , *UC Berkeley Enron Email Analysis*. Available: http://bailando.sims.berkeley.edu/enron_email.html [2019, 12/11/2019].

Blitzer, J., Dredze, M. & Pereira, F. 2007, "Biographies, Bollywood, Boom-boxes and Blenders: Domain Adaptation for Sentiment Classification", *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*Association for Computational Linguistics, Prague, Czech Republic, jun, pp. 440.

Brdiczka, O., Liu, J., Price, B., Shen, J., Patil, A., Chow, R., Bart, E. & Ducheneaut, N. 2012, "Proactive Insider Threat Detection through Graph Learning and Psychological Context", *2012 IEEE Symposium on Security and Privacy Workshops, Security and Privacy Workshops (SPW), 2012 IEEE Symposium on,* , pp. 142-149.

Chawla, N., Bowyer, K., Hall, L. & Kegelmeyer, W. 2002, "SMOTE: Synthetic Minority Over-sampling Technique", *J.Artif.Intell.Res.(JAIR),* vol. 16, pp. 321-357.

Cohen, W. 2015, 05/08/2015-last update*, CMU Enron Email Dataset*. Available: https://www.cs.cmu.edu/~enron/ [2019, 12/11/2019].

Colwill, C. 2009, *Human factors in information security: The insider threat – Who can you trust these days?*.

Crowd Research Partners 2017, , *Insider Threat Report 2018*. Available: https://crowdresearchpartners.com/wp-content/uploads/2017/07/Insider-Threat-Report-2018.pdf [2019, 12/11/2019].

Cyber Ark 2012, , *Survey Finds That 85 Percent of Workers Know It's Illegal to Steal Corporate Data, Yet Many Are Willing to Risk the Consequences | CyberArk*. Available: https://www.cyberark.com/press/survey-finds-85-percent-workers-know-illegal-steal-corporate-data-yet-many-willing-risk-consequences/ [2019, 12/11/2019].

Glasser, J. & Lindauer, B. 2013, "Bridging the Gap: A Pragmatic Approach to Generating Insider Threat Data", , pp. 98.

Go, A., Bhayani, R. & Huang, L. 2009, "Twitter sentiment classification using distant supervision", *Processing,* vol. 150.

Google Trends 2019, 12/8/2019-last update*, data analytics - Explore - Google Trends*. Available: https://trends.google.com/trends/explore?date=all&geo=CY&q=data analytics [2019, 12/8/2019].

Harilal, A., Toffalini, F., Castellanos, J., Guarnizo, J., Homoliak, I. & Ochoa, M. 2017, "TWOS: A Dataset of Malicious Insider Threat Behavior Based on a Gamified Competition", *Proceedings of the 2017 International Workshop on Managing Insider Security Threats*ACM, New York, NY, USA, pp. 45.

Hochreiter, S. & Schmidhuber, J. 1997, "Long short-term memory", *Neural computation,* vol. 9, no. 8, pp. 1735-1780.

Jiang, J., Chen, J., Choo, K.R., Liu, K., Liu, C., Yu, M. & Mohapatra, P. 2018, "Prediction and Detection of Malicious Insiders' Motivation Based on Sentiment Profile on Webpages and Emails", *MILCOM 2018 - 2018 IEEE Military Communications Conference (MILCOM), Military Communications Conference (MILCOM), MILCOM 2018 - 2018 IEEE,* , pp. 1-6.

Jing, H. 2019, 05/07/2019-last update*, Why Linear Regression is not suitable for Classification* [Homepage of Towards Data Science], [Online]. Available: https://towardsdatascience.com/why-linear-regression-is-not-suitable-for-binary-classification-c64457be8e28 [2019, 12/9/2019].

Kandias, M., Stavrou, V., Bozovic, N. & Gritzalis, D. 2013, *Proactive insider threat detection through social media: The YouTube case,* .

Keeney, M., Kowalski, E., Cappelli, D.M., Moore, A.P., Shimeall, T.J. & Rogers, S. 2005, "Insider Threat Study: Computer System Sabotage in Critical Infrastructure Sectors", .

Kirk, M. 2014, *Thoughtful Machine Learning : A Test-Driven Approach,* O'Reilly Media, Sebastopol, CA.

Legg, P.A., Buckley, O., Goldsmith, M. & Creese, S. 2015, "Caught in the act of an insider attack: detection and assessment of insider threat", *2015 IEEE International Symposium on Technologies for Homeland Security (HST)*, April, pp. 1.

Levy, S. 2019, 06/24/2016-last update*, An Exclusive Look at How AI and Machine Learning Work at Apple | WIRED* [Homepage of Wired], [Online]. Available: https://www.wired.com/2016/08/an-exclusive-look-at-how-ai-and-machine-learning-work-at-apple/#.7re109jcn [2019, 12/9/2019].

Mehan, J.E. 2016, *Insider Threat : A Guide to Understanding, Detecting, and Defending Against the Enemy From Within,* IT Governance Publishing, Ely, Cambridgeshire, United Kingdom.

Mohammad, S.M. 2012, "# Emotional tweets", *Proceedings of the First Joint Conference on Lexical and Computational Semantics-Volume 1: Proceedings of the main conference and the shared task, and Volume 2: Proceedings of the Sixth International Workshop on Semantic Evaluation*Association for Computational Linguistics, , pp. 246.

Mohammad, S.M., Bravo-Marquez, F., Salameh, M. & Kiritchenko, S. 2018, "SemEval-2018 Task 1: Affect in Tweets", *Proceedings of International Workshop on Semantic Evaluation (SemEval-2018)*New Orleans, LA, USA.

Mohammad, S.M., Tony & Yang 2013, *Tracking Sentiment in Mail: How Genders Differ on Emotional Axes*.

Mohammad, S.M. & Turney, P.D. 2013, "Crowdsourcing a Word-Emotion Association Lexicon", vol. 29, no. 3, pp. 436-465.

Mohammad, S. & Kiritchenko, S. 2018, "Understanding Emotions: A Dataset of Tweets to Study Interactions between Affect Categories", *Proceedings of the Eleventh International Conference on Language Resources and Evaluation ({LREC} 2018)*European Language Resources Association, Miyazaki, Japan; ELRA, may.

Mohammad, S. & Turney, P. 2010, "Emotions Evoked by Common Words and Phrases: Using Mechanical Turk to Create an Emotion Lexicon", *Proceedings of the {NAACL} {HLT} 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*Association for Computational Linguistics, Los Angeles, CA, jun, pp. 26.

Naik, C., Gupta, A., Ge, H., Mathias, L. & Sarikaya, R. 2018, *Contextual Slot Carryover for Disparate Schemas*.

Noonan, T. & Archuleta, E. 2008, *The National Infrastructure Advisory Council's final report and recommendations on the insider threat to critical infrastructures [electronic resource] / Thomas Noonan, Edmund Archuleta,* Washington, D.C.] : DHS/NIAC, 2008.

Park, W., You, Y. & Lee, K. 2018, *Detecting Potential Insider Threat: Analyzing Insiders' Sentiment Exposed in Social Media*.

Plutchik, R. 1982, "A psychoevolutionary theory of emotions", *Social Science Information,* vol. 21, no. 4-5, pp. 529-553.

Rennie, J.D.M., Shih, L., Teevan, J. & Karger, D.R. 2003, "Tackling the Poor Assumptions of Naive Bayes Text Classifiers", *Proceedings of the Twentieth International Conference on International Conference on Machine Learning*AAAI Press, , pp. 616.

S. M. Ho, J. T. Hancock, C. Booth, M. Burmester, X. Liu & S. S. Timmarajus 2016, *Demystifying Insider Threat: Language-Action Cues in Group Dynamics*.

Sang-Sang Tan, Jin-Cheon Na & Duraisamy, S. 2019, "Unified Psycholinguistic Framework: An Unobtrusive Psychological Analysis Approach Towards Insider Threat Prevention and Detection", *Journal of Information Science Theory & Practice (JIStaP),* vol. 7, no. 1, pp. 52-71.

Schultz, E.E. 2002, *A framework for understanding and predicting insider attacks*.

Sentiment140 2019, , *Sentiment140 - A Twitter Sentiment Analysis Tool*. Available: http://help.sentiment140.com/for-students [2019, 12/10/2019].

Statista 2019, , • *Number of social media users worldwide 2010-2021 | Statista* [Homepage of Statista], [Online]. Available:

https://www.statista.com/statistics/278414/number-of-worldwide-social-network-users/ [2019, 12/8/2019].

Verizon 2019, *Verizon: 2019 Data Breach Investigations Report*, Elsevier Science B.V., Amsterdam, Netherlands.

Wikimedia Commons 2018, , *File:Plutchik-wheel.svg - Wikimedia Commons*. Available: https://commons.wikimedia.org/wiki/File:Plutchik-wheel.svg [2019, 12/6/2019].

Wu, Y., Schuster, M., Chen, Z., Le, Q.V., Norouzi, M., Macherey, W., Krikun, M., Cao, Y., Gao, Q., Macherey, K., Klingner, J., Shah, A., Johnson, M., Liu, X., Kaiser, L., Gouws, S., Kato, Y., Kudo, T., Kazawa, H., Stevens, K., Kurian, G., Patil, N., Wang, W., Young, C., Smith, J., Riesa, J., Rudnick, A., Vinyals, O., Corrado, G., Hughes, M. & Dean, J. 2016, *Google's Neural Machine Translation System: Bridging the Gap between Human and Machine Translation*.

Παράρτημα Α
Κώδικας Python

Α.1 Εκπαίδευση Μοντέλων Μηχανικής Μάθησης για Ταξινόμηση Συναισθήματος

Το παρακάτω πρόγραμμα σε γλώσσα προγραμματισμού Python, γράφτηκε και χρησιμοποιήθηκε για την εκπαίδευση των μοντέλων μηχανικής μάθησης που περιγράφονται σε αυτή τη μεταπτυχιακή διατριβή. Το πρόγραμμα παρέχει ρυθμίσεις κάτω από το πλαίσιο CONSTANTS που μπορούν να μεταβάλουν τη λειτουργία του συστήματος. Σε γενικές γραμμές το πρόγραμμα εκτελεί τα ακόλουθα βήματα:

- Φόρτωση συνόλου δεδομένων που θα χρησιμοποιηθούν για εκπαίδευση από τη βάση δεδομένων. Αυτό περιλαμβάνει τα tweets και την αντίστοιχη ταξινόμηση (Emotion ή label) του κάθε tweet. Η δομή δεδομένων έχει μορφή ({“TEXT”: …, “Emotion”: …}, … , {“TEXT”: …, “Emotion”: …}).
- Φορτώνεται το λεξικό που θα χρησιμοποιηθεί για τον ταξινομητή EmotionLexiconClassifier, το οποίο περιλαμβάνει τη λέξη, το συναίσθημα και μια σημαία που καθορίζει αν το συγκεκριμένο συναίσθημα συσχετίζεται με την λέξη. Η δομή δεδομένων έχει μορφή ({“Word”: …, “Emotion”: …, “Association”: …}, … , ({“Word”: …, “Emotion”: …, “Association”: …}).
- Γίνεται ανακάτωμα τις σειράς των δεδομένων ώστε να έχουμε τυχαία σειρά εμφάνισης δεδομένων. Χρησιμοποιείται συγκεκριμένο randomization seed ώστε να έχουμε πάντα την ίδια σειρά μετά το ανακάτωμα.
- Επεξεργασία κειμένων που έχουν φορτωθεί από χρησιμοποιώντας τη βιβλιοθήκη textPreprocessor που αναγράφεται στην ενότητα Α.2.
- Διάσπαση των δεδομένων σε υποσύνολα δεδομένων εκπαίδευσης και ελέγχου. Το ποσοστό διάσπασης των δεδομένων μπορεί να αλλαχθεί ρυθμίζοντας την τιμή της μεταβλητής testSize.

- Εάν έχει ενεργοποιηθεί η μεταβλητή useBalancing, γίνεται ισορρόπηση του συνόλου δεδομένων εκπαίδευσης χρησιμοποιώντας τους μηχανισμούς που καθορίζονται στη λίστα balancers.
- Για κάθε συναίσθημα εκτελούμε ξεχωριστή εκπαίδευση one-vs-rest. Αυτό γίνεται αλλάζοντας το σύνολο δεδομένων εκπαίδευσης και ελέγχου ώστε να περιλαμβάνει μόνο το συναίσθημα που θέλουμε να ελέγξουμε και το αντίθετο του, π.χ. αν μελετάμε το συναίσθημα anger, όλα τα δεδομένα που δεν είναι anger μετατρέπονται σε not_anger. Αυτό οδηγεί στην εκτέλεση binomial ταξινόμησης σε αντίθεση με multinomial, και μας επιτρέπει την δημιουργία ταξινομητών για κάθε κλάση. Α
- Πριν την εκτέλεση του μοντέλου LSTM, γίνεται μορφοποίηση των δεδομένων βάση του αντικειμένου Tokenizer που παρέχεται από τη βιβλιοθήκη Keras. Οι ρυθμίσεις του μοντέλου LSTM μπορούν να καθοριστούν από τον χρήστη του προγράμματος.
- Πριν την εκτέλεση των μοντέλων της βιβλιοθήκης Scikit-learn, γίνεται μορφοποίηση των δεδομένων βάση του αντικειμένου Vectorizer που καθορίζεται από τον χρήστη. Το αντικείμενο Vectorizer μπορεί να αντιστοιχεί σε αντικείμενα TfidfVectorizer ή CountVectorizer από τη βιβλιοθήκη Keras.
- Στο τέλος της εκπαίδευσης κάθε συναισθήματος, γίνεται αποθήκευση των μοντέλων και των συναφή στοιχείων τους χρησιμοποιώντας τη βιβλιοθήκη pickle.

```
#####################################
# IMPORTS
#####################################

# Import custom modules.
from textPreprocessor import preProcessText
from EmotionLexiconClassifier import EmotionLexiconClassifier

# Helper Modules
from nltk.tokenize import word_tokenize
import numpy as np

# DB connector import
import mysql.connector as db
```

```
# LSTM Imports
from keras.preprocessing.text import Tokenizer
from keras.preprocessing.sequence import pad_sequences
from keras.engine.sequential import Sequential
from keras.layers.embeddings import Embedding
from keras.layers.recurrent import GRU, LSTM
from keras.layers.core import Activation, Dense, Dropout
from keras.layers.wrappers import Bidirectional, TimeDistributed
from keras.utils.np_utils import to_categorical
from keras.layers.normalization import BatchNormalization
from keras.callbacks import EarlyStopping, ModelCheckpoint

# Sci-Kit Learn Imports
from sklearn.model_selection import StratifiedKFold, train_test_split
from sklearn.metrics import accuracy_score, classification_report, f1_score, recall_score
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.linear_model import LogisticRegression
from sklearn.naive_bayes import ComplementNB, GaussianNB, MultinomialNB
from sklearn.svm import SVC
from sklearn.preprocessing import StandardScaler

# Under/Over Balancing algorithms
from imblearn.over_sampling import SMOTE, RandomOverSampler
from imblearn.under_sampling import RandomUnderSampler

# Utility for saving/loading models
import pickle

# Plotting libraries
import matplotlib.pyplot as plt
from scikitplot.metrics import plot_confusion_matrix
```

```
def displayAccuracyStatistics(Y_true, Y_pred, labels, title = None, text_file = None, plot_file = None):
    if(text_file != None):
        file = open(text_file, "w")
        file.write(title+"\n")
        file.write("Accuracy: {}\n".format(accuracy_score(Y_true, Y_pred)))
        file.write("F1 Score: {}\n".format(f1_score(Y_true, Y_pred, labels = labels, average = "macro")))
        file.write("Classification Report\n")
        file.write(classification_report(Y_true, Y_pred, labels))
        file.close()
    print("Accuracy: {}".format(accuracy_score(Y_true, Y_pred)))
    print("F1 Score: {}".format(f1_score(Y_true, Y_pred, labels = labels, average = "macro")))
    print("Classification Report")
    print(classification_report(Y_true, Y_pred, labels))
    plot_confusion_matrix(Y_true, Y_pred, title = title)
    if(plot_file != None):
        plt.savefig(plot_file)
        plt.close()
    else:
        plt.show()


#####################################
# CONSTANTS
#####################################

scenario = "2grams"

# MySQL Settings
sqlHost = ""
sqlDB = ""
sqlUser = ""
```

```
sqlPassword = ""

# Dataset Settings
randomSeed = 28390
testSize = 0.4
useBalancing = False
balancers = [
   #RandomUnderSampler(sampling_strategy=0.6),
   #SMOTE(sampling_strategy=0.8)
   RandomUnderSampler(sampling_strategy="majority")

   ]

# Tokenization Settings
maxWords = 20000
maxSentenceLength = 30
useNgrams = False
SKVectorizer = TfidfVectorizer
ngramSize = (1, 2)

# LSTM Settings
useLSTM = False
useLSTMWord2Vec = False
lstmW2vMinCount = 4
lstmSize = 80
lstmDropout = 0.40
lstmEpochs = 10
lstmEarlyStopEnabled = True
lstmEarlyStopMinDelta = 0.1
lstmEarlyStopPatience = 3

# SKLearn Settings
useLogisticRegression = True
useMultinomialNB = True
```

```
useComplementNB = True
useGaussianNB = True
useLexiconClassifier = True
useSVM = True

# Model saving switch
saveModels = True

#####################################
# GET DATA FROM DB
#####################################

# Connect to DB
dbConnection = db.connect(host = sqlHost, database = sqlDB, user = sqlUser, password =
sqlPassword)
# Set dictionary flag so that we get a dictionary for each row.
dbCursor = dbConnection.cursor(dictionary = True)

# Get data (messages and labels)
dbCursor.execute("SELECT Text, Emotion FROM v_combined_emotion_data")
data = dbCursor.fetchall()

# Get emotion lexicon data (word, emotion, association)
dbCursor.execute("SELECT * FROM v_emotion_lexicon_aggregated")
lexiconData = dbCursor.fetchall()

#####################################
# PREPARE DATA
#####################################

# Randomize data in a reproducable way.
np.random.seed(randomSeed)
np.random.shuffle(data)
```

```
# Pre-process text
messages = np.array([preProcessText(row["Text"]) for row in data])

# Get labels
labels = np.array([row["Emotion"] for row in data])

## Create labels (emotions) dictionary {"emotion": index/unique id}
#labelIDMap = {value: index for index, value in enumerate(set(labels))}
labelNamesMaster = [label for label in set(labels)]

for label in labelNamesMaster:
  if(label == "neutral"):
    continue

  print("##########################################")
  print("Begin training for label: " + label)
  print("##########################################")
  labelsCurrent = [label if x == label else "not_" +label for x in labels]
  labelIDMap = {value: index for index, value in enumerate(set(labelsCurrent))}
  labelNames = [key for key in labelIDMap.keys()]
  lexiconDataCurrent = [row if row["Emotion"] == label else {"Word": row["Word"],
"Emotion": "not_" + label, "Association": row["Association"]} for row in lexiconData]

  print("Current Dataset:")
  print(np.asarray(np.unique(labelsCurrent, return_counts = True)).T)

  import os
  path = "./models/" + scenario + "/"
  try:
    if(os.path.exists(path) == False):
      os.mkdir(path)
  except OSError:
    print ("Creation of the directory %s failed" % path)
  else:
```

```
    print ("Successfully created the directory %s " % path)

  ####################################
  # SPLIT DATA INTO TRAINING / TESTING
  ####################################

  # Split dataset into training and testing.
  trainMessages, testMessages, trainLabels, testLabels = train_test_split(messages, labelsCurrent, test_size = testSize, random_state = None, stratify = labelsCurrent)

  print("Please verify training dataset balance and restart if required:")
  print(np.asarray(np.unique(trainLabels, return_counts = True)).T)

  # Balance training dataset by over-sampling using Balancing mechanism. Overwrites trainMessages and trainLabels with balanced sets.
  if(useBalancing):
    print("Balancing dataset!")
    # Convert trainMessages to padded sequences, balance them using Balancing Mechanism and then convert back to text. Then overwrite trainMessages and trainLabels
    tokenizer = Tokenizer()
    tokenizer.fit_on_texts(trainMessages)
    balancedMessages = tokenizer.texts_to_sequences(trainMessages)
    balancedMessages = pad_sequences(balancedMessages, maxSentenceLength)
    balancedLabels = trainLabels.copy() # Temporary, required to seed labels in for loop
    for balancer in balancers:
      balancedMessages, balancedLabels = balancer.fit_sample(balancedMessages, balancedLabels)
    balancedMessages = tokenizer.sequences_to_texts(balancedMessages)
    trainMessages = balancedMessages
    trainLabels = balancedLabels
    print("Dataset after balancing:")
    print(np.asarray(np.unique(trainLabels, return_counts = True)).T)

  # skf = StratifiedKFold(n_splits = 5, shuffle = True, random_state = None)
```

```
# trainMessages = []
# trainLabels = []
# testMessages = []
# testLabels = []
# for trainIndex, testIndex in skf.split(messages, labels):
#     trainMessages = messages[trainIndex]
#     trainLabels = labels[trainIndex]
#     testMessages = messages[testIndex]
#     testLabels = labels[testIndex]

#####################################
# LSTM CLASSIFIER
#####################################

# Tokenize messages and pad sequences to maxSentenceLength
tokenizer = Tokenizer(num_words = maxWords)
tokenizer.fit_on_texts(trainMessages)
trainSequences = tokenizer.texts_to_sequences(trainMessages)
testSequences = tokenizer.texts_to_sequences(testMessages)

X_train = pad_sequences(trainSequences, maxSentenceLength)
X_test = pad_sequences(testSequences, maxSentenceLength)

# Convert labels to unique IDs
trainLabelsAsIDs = [labelIDMap[label] for label in trainLabels]
testLabelsAsIDs = [labelIDMap[label] for label in testLabels]

# Convert labels to categories for LSTM consumption.
Y_train = to_categorical(trainLabelsAsIDs, num_classes = len(labelIDMap))
Y_test = to_categorical(testLabelsAsIDs, num_classes = len(labelIDMap))



if(useLSTM):
```

```
print("Starting LSTM")
# Create the model
model = Sequential()

if(useLSTMWord2Vec):
  from gensim.models import Word2Vec
  w2v = Word2Vec([word_tokenize(msg) for msg in trainMessages], size = lstmSize, min_count = lstmW2vMinCount)
  embedding_matrix = np.zeros((len(tokenizer.word_index) + 1, lstmSize))
  for word, i in tokenizer.word_index.items():
    if word in w2v.wv.vocab:
      embedding_matrix[i] = w2v.wv[word]
  model.add(Embedding(len(tokenizer.word_index) + 1, lstmSize, input_length = maxSentenceLength, weights = [embedding_matrix], trainable = False))
else:
  model.add(Embedding(len(tokenizer.word_index) + 1, lstmSize, input_length = maxSentenceLength))
model.add(Dropout(lstmDropout))
model.add(LSTM(lstmSize, dropout = lstmDropout, recurrent_dropout = lstmDropout))
model.add(Dropout(lstmDropout))
model.add(Dense(len(labelIDMap), activation="softmax"))
model.compile(loss="binary_crossentropy", optimizer="adam", metrics=["accuracy"])

# Print model summary
print()
model.summary()
print()

# from keras.utils.vis_utils import plot_model
# plot_model(model, to_file="model_plot.png", show_shapes=True, show_layer_names=True)
# exit()
```

```
    # Prepare callbacks for LSTM training
    lstmCallbacks = list()
    lstmCallbacks.append(ModelCheckpoint("emotionLSTM", monitor="val_loss", verbose=1, save_best_only=True, mode="auto"))
    if(lstmEarlyStopEnabled):
      lstmCallbacks.append(EarlyStopping(monitor="val_loss", min_delta=lstmEarlyStopMinDelta, patience= lstmEarlyStopPatience, verbose = 1, mode="auto"))

    # Fit the model
    result = model.fit(X_train, Y_train, validation_data=(X_test, Y_test), epochs=lstmEpochs, shuffle = False, callbacks = lstmCallbacks)

    # Validate model
    Y_pred_lstm = model.predict_classes(X_test)
    Y_pred_lstm = [labelNames[i] for i in Y_pred_lstm]
    displayAccuracyStatistics(testLabels, Y_pred_lstm, labelNames, "LSTM", "./models/" + scenario + "/" + label + "_lstm.txt", "./models/" + scenario + "/" + label + "_lstm.png")

  ####################################
  # OTHER CLASSIFIERS
  ####################################

  if(useNgrams):
    vectorizer = SKVectorizer(analyzer = "word", ngram_range = ngramSize, tokenizer = word_tokenize, max_features = maxWords)
  else:
    vectorizer = SKVectorizer()

  X_train = vectorizer.fit_transform(trainMessages)
  X_test = vectorizer.transform(testMessages)

  Y_train = trainLabels
```

```
Y_test = testLabels

if(useLogisticRegression):
    print("Starting Logistic Regression")
    lr = LogisticRegression(solver = "newton-cg")
    print(lr.fit(X_train, Y_train))
    Y_pred_lr = lr.predict(X_test)
    displayAccuracyStatistics(Y_test, Y_pred_lr, labelNames, "Logistic Regression", "./models/" + scenario + "/" + label + "_lr.txt", "./models/" + scenario + "/" + label + "_lr.png")


if(useMultinomialNB):
    print("Starting Multinomial Naive Bayes")
    mnb = MultinomialNB()
    print(mnb.fit(X_train.todense(), Y_train))
    Y_pred_mnb = mnb.predict(X_test)
    displayAccuracyStatistics(Y_test, Y_pred_mnb, labelNames, "Multinomial Naive Bayes", "./models/" + scenario + "/" + label + "_mnb.txt", "./models/" + scenario + "/" + label + "_mnb.png")

if(useComplementNB):
    print("Starting Complement Naive Bayes")
    cnb = ComplementNB()
    print(cnb.fit(X_train.todense(), Y_train))
    Y_pred_cnb = cnb.predict(X_test)
    displayAccuracyStatistics(Y_test, Y_pred_cnb, labelNames, "Complement Naive Bayes", "./models/" + scenario + "/" + label + "_cnb.txt", "./models/" + scenario + "/" + label + "_cnb.png")

if(useGaussianNB):
    print("Starting Gaussian Naive Bayes")
    gnb = GaussianNB()
    print(gnb.fit(X_train.todense(), Y_train))
```

```
    Y_pred_gnb = gnb.predict(X_test.todense())
    displayAccuracyStatistics(Y_test, Y_pred_gnb, labelNames, "Gaussian Naive Bayes", "./models/" + scenario + "/" + label + "_gnb.txt", "./models/" + scenario + "/" + label + "_gnb.png")

  if(useLexiconClassifier):
    print("Starting Emotion Lexicon Classifier (Unsupervised)")
    elc = EmotionLexiconClassifier(lexiconDataCurrent)
    Y_pred_elc = elc.predict(testMessages)
    displayAccuracyStatistics(Y_test, Y_pred_elc, labelNames, "Emotion Lexicon Classifier", "./models/" + scenario + "/" + label + "_elc.txt", "./models/" + scenario + "/" + label + "_elc.png")

  if(useSVM):
    print("Starting Support Vector Machine")

    # Using scaler to speed up results
    ss = StandardScaler(with_mean=False)
    X_train = ss.fit_transform(X_train)
    X_test = ss.transform(X_test)

    #svc = SVC(kernel = "poly", C = 0.1, gamma = "scale", degree = len(labelIDMap))
    svc = SVC(kernel = "rbf", C = 0.5, decision_function_shape = "ovr", gamma = "scale")
    print(svc.fit(X_train, Y_train))
    Y_pred_svc = svc.predict(X_test)
    displayAccuracyStatistics(Y_test, Y_pred_svc, labelNames, "Support Vector Machine", "./models/" + scenario + "/" + label + "_svm.txt", "./models/" + scenario + "/" + label + "_svm.png")


  ####################################
  # Save models, vectorizers and data
  ####################################
```

```
  if(saveModels):
    # Save label ID map for later use in LSTM predictions.
    pickle.dump(labelIDMap, open("./models/" + scenario + "/" + label + "_labelIDMap.sav", "wb"))


  if(saveModels and useLSTM):
    # Save LSTM tokenizer for conversion of data.
    pickle.dump(tokenizer, open("./models/" + scenario + "/" + label + "_lstmTokenizer.sav", "wb"))
    pickle.dump(model, open("./models/" + scenario + "/" + label + "_lstm.sav", "wb"))

  if(saveModels and (useGaussianNB or useLexiconClassifier or useLogisticRegression or useMultinomialNB or useSVM)):
    pickle.dump(vectorizer, open("./models/" + scenario + "/" + label + "_skVectorizer.sav", "wb"))

  if(saveModels and useLogisticRegression):
    pickle.dump(lr, open("./models/" + scenario + "/" + label + "_lr.sav", "wb"))

  if(saveModels and useMultinomialNB):
    pickle.dump(mnb, open("./models/" + scenario + "/" + label + "_mnb.sav", "wb"))

  if(saveModels and useComplementNB):
    pickle.dump(cnb, open("./models/" + scenario + "/" + label + "_cnb.sav", "wb"))

  if(saveModels and useGaussianNB):
    pickle.dump(gnb, open("./models/" + scenario + "/" + label + "_gnb.sav", "wb"))

  if(saveModels and useLexiconClassifier):
    pickle.dump(elc, open("./models/" + scenario + "/" + label + "_elc.sav", "wb"))

  if(saveModels and useSVM):
    pickle.dump(svc, open("./models/" + scenario + "/" + label + "_svc.sav", "wb"))
```

```
pickle.dump(ss, open("./models/" + scenario + "/" + label + "_svc_ss.sav", "wb"))
```

A.2 Προ-επεξεργασία κειμένου

Το πιο κάτω πρόγραμμα είναι γραμμένο σε γλώσσα Python και χρησιμοποιείται για την προ-επεξεργασία κειμένου των tweets πριν την εκπαίδευση των μοντέλων μηχανικής μάθησης. Εκτελεί τις ακόλουθες διαδικασίες:

- Μετατροπή σε μικρά γράμματα
- Κανονικοποίηση γραμμάτων με τόνους ώστε να μην έχουμε λέξεις που να διαφέρουν μεταξύ τους λόγω τονισμού
- Αφαίρεση οντοτήτων τύπου HTML που μπορεί να υπάρχουν στο κείμενο.
- Αφαίρεση URL από το κείμενο, μετατροπή τους σε “urlintext”
- Αφαίρεση χρηστών από το κείμενο, μετατροπή τους σε “mentionintext”
- Αφαίρεση αριθμών
- Αφαίρεση σημείων στίξης ή άλλων ειδικών χαρακτήρων
- Λημματοποίηση λέξεων, όπως περιγράφεται στο κεφάλαιο 5
- Εκτέλεση stemming
- Καθορισμός σημαιών άρνησης, π.χ. this is not good -> this is not good_NEG
- Αφαίρεση κοινών λέξεων (stop words)

```
# Text pre-processing imports
import re
import string
from nltk.corpus import stopwords, wordnet
from nltk.stem import SnowballStemmer
from nltk.stem import WordNetLemmatizer
from nltk.tokenize import TweetTokenizer, word_tokenize
from nltk.sentiment.util import mark_negation
from nltk.tokenize import sent_tokenize
from nltk import pos_tag
import html
import unicodedata

# Regexes
hashtag_re = re.compile("\#(\w+)")
mention_re = re.compile("\@([a-zA-Z_0-9]+)")
number_re = re.compile("[0-9]+")
```

```
uri_re = re.compile("https?://[^\s]+")
repeatingChar_re = re.compile(r"(.)\1{2,}")
punctuation_re = re.compile("[{0}]".format(re.escape(string.punctuation)))

# Stemmer
stemmer = SnowballStemmer("english")

# Lemmatizer
lemmatizer = WordNetLemmatizer()

# Stop words list
stopWords = stopwords.words("english")

def get_wordnet_pos(treebank_tag):
    """
    return WORDNET POS compliance to WORDENT lemmatization (a,n,r,v)
    """
    if treebank_tag.startswith('J'):
        return wordnet.ADJ
    elif treebank_tag.startswith('V'):
        return wordnet.VERB
    elif treebank_tag.startswith('N'):
        return wordnet.NOUN
    elif treebank_tag.startswith('R'):
        return wordnet.ADV
    else:
        # As default pos in lemmatization is Noun
        return wordnet.NOUN

def preProcessText(text):
    # Make text lower case
    text = text.lower()

    # Remove accented text
```

```
    text = unicodedata.normalize("NFKD", text).encode("ascii", "ignore").decode("utf-8", "ignore")

    # Replace escaped HTML tags found in text
    text = html.unescape(text)

    # Remove URIs
    text = uri_re.sub("urlintext", text)

    # Remove mentions
    text = mention_re.sub("mentionintext", text)

    # Remove numbers
    text = number_re.sub("", text)

    # Split text into sentences and then into words and execute lemmatization+stemming (at word-level), negation (at sentence-level) and remove stop words, then join everything back.
    textSentences = sent_tokenize(text)
    processedSentences = list()
    for sent in textSentences :
        # Remove punctuation from sentence.
        sent = punctuation_re.sub("", sent)
        # Tokenize sentence into words.
        words = word_tokenize(sent)
        # Remove repeating characters to only 2, e.g. Heeeeelllloooo -> Heelloo
        words = [repeatingChar_re.sub(r"\1\1", word) for word in words]
        # Execute lemmatization on words.
        words = [lemmatizer.lemmatize(word[0], get_wordnet_pos(word[1])) for word in pos_tag(words)]
        ## Execute stemming on words.
        #words = [stemmer.stem(word) for word in words]
        # Mark negative words in sentence.
        words = mark_negation(words)
```

```
    # Remove stop words, even if negative.
    words = [word for word in words if not word.replace("_NEG", "") in stopWords]
    # Join words into a sentence
    processedSentences.append(" ".join(words))

# Join processed sentences into a single string
text = " ".join(processedSentences)

return text
```

Α.3 Μη-Επιτηρούμενος Ταξινομητής Συναισθημάτων με χρήση Λεξικού

Η λειτουργία του μη-επιτηρούμενου ταξινομητή συναισθημάτων με χρήση λεξικού εξηγείται στο κεφάλαιο 6.

```
from nltk import word_tokenize

class EmotionLexiconClassifier:
    def __init__(self, lexiconData = None):
        if(lexiconData != None):
            self.loadLexicon(lexiconData)
        else:
            self.lexicon = dict()
            self.lexiconLabels = dict()

    def loadLexicon(self, lexiconData):
        self.lexicon = dict()
        self.lexiconLabels = dict() # Contains label: sum of associations
        for row in lexiconData:
            word = row['Word']
            emotion = row['Emotion']
            association = row['Association']

            if word not in self.lexicon:
                self.lexicon[word] = dict()

            if emotion not in self.lexicon[word]:
                self.lexicon[word][emotion] = association

            if emotion not in self.lexiconLabels:
                self.lexiconLabels[emotion] = 0

            self.lexiconLabels[emotion] += association
```

```
    return self

def predict(self, texts):
    scores = list()
    for text in texts:
        scoreCard = {emotion: 0 for emotion in self.lexiconLabels.keys()}
        words = word_tokenize(text)
        for word in words:
            # Handle mark_negation()
            neg = "_NEG" in word
            word = word.replace("_NEG", "")
            if word in self.lexicon:
                for emotion in self.lexicon[word]:
                    if not neg:
                        scoreCard[emotion] += self.lexicon[word][emotion]
                    else:
                        scoreCard[emotion] -= self.lexicon[word][emotion]
                        # if self.lexicon[word][emotion] == 0:
                        #     scoreCard[emotion] += 1
                        # else:
                        #     scoreCard[emotion] += 0
        emotionsFound = list()
        maxCount = 0
        for key in scoreCard.keys():
            if scoreCard[key] > 0 and maxCount <= scoreCard[key]:
                if maxCount == scoreCard[key]:
                    emotionsFound.append(key)
                else:
                    maxCount = scoreCard[key]
                    emotionsFound = [key]

        if(len(emotionsFound) == 1):
            scores.append(emotionsFound[0])
```

```
        # If no emotions were found, find the most popular lexicon label.
        elif(len(emotionsFound) == 0):
            maxCount = 0
            maxCountEmotion = ""
            for emotion in self.lexiconLabels.keys():
                if(maxCount < self.lexiconLabels[emotion]):
                    maxCount = self.lexiconLabels[emotion]
                    maxCountEmotion = emotion
            scores.append(maxCountEmotion)

        # If more than one emotion was found, find the most popular emotion from the lexicon labels dict.
        elif(len(emotionsFound) > 1):
            maxCount = 0
            maxCountEmotion = ""
            for emotion in emotionsFound:
                if(maxCount < self.lexiconLabels[emotion]):
                    maxCount = self.lexiconLabels[emotion]
                    maxCountEmotion = emotion
            scores.append(maxCountEmotion)
        return scores

    def score(self, X, Y):
        scores = self.predict(X)
        correct = 0
        for i in range(len(scores)):
            emotionFound = scores[i].split("/")
            # If multiple emotions were found but the correct one is also included, do not count as fully correct.
            # e.g. if found emotions A, B, C and correct one is C then the score is 0.33 (1/3)
            if Y[i] in emotionFound:
                correct += 1
                #correct += (1/len(emotionFound))
        return correct/len(scores)
```

A.4 Φόρτωση συνόλου δεδομένων Sentiment140

Το πιο κάτω πρόγραμμα Python χρησιμοποιήθηκε για την μεταφορά του συνόλου δεδομένων Sentiment140 από το .csv αρχείο στο οποίο βρισκόταν, στη βάση δεδομένων του συστήματος. Κατά την επεξεργασία κάθε μηνύματος, εξάγονται πληροφορίες όπως αναφορές σε άλλους χρήστες, hashtags που χρησιμοποιηθήκαν, και γίνεται ταξινόμηση θετικού/αρνητικού συναισθήματος χρησιμοποιώντας τον ταξινομητή VADER ο οποίος παρέχεται από τη βιβλιοθήκη NLTK.

```
# Imports
import csv
import re
import nltk
from nltk.sentiment.vader import SentimentIntensityAnalyzer
from dateutil import parser
import mysql.connector

# Classes
class Message:
	def __init__(self, id, postedOn, user, text, sentiment_score):
		self.id = id
		self.postedOn = postedOn
		self.user = user
		self.text = text
		self.score = sentiment_score

	def getSentiment(self):
		compound = self.score["compound"]
		return "Neutral" if compound == 0 else "Negative" if compound > 0 else "Positive"

class User:
	def __init__(self, name):
		self.name = name
		self.messages = list()
```

```
class Hashtag:
	def __init__(self, name):
		self.name = name
		self.messages = list()

# Global Variable Definition
users = dict()
sentimentAnalyzer = SentimentIntensityAnalyzer()

# Regexes for hashtag and mention extraction
hashtag_re = re.compile("\#(\w+)")
mention_re = re.compile("\@([a-zA-Z_0-9]+)")

try:
	connection = mysql.connector.connect(host='', database='', user='', password='')
	cursor = connection.cursor()

	# Load Data
	"""
		Data file format has 6 fields:
		0 - the polarity of the tweet (0 = negative, 2 = neutral, 4 = positive) -- ignored, we'll calculate our own
		1 - the id of the tweet (2087) -- stored in message item
		2 - the date of the tweet (Sat May 16 23:58:44 UTC 2009) -- stored in message item
		3 - the query (lyx). If there is no query, then this value is NO_QUERY. -- ignored
		4 - the user that tweeted (robotickilldozr) -- stored in user item
		5 - the text of the tweet (Lyx is cool) -- stored in message item
	"""
	with open('./all_data.csv') as dataset:
		fileReader = csv.reader(dataset)
		counter = 0
```

```
            canProceed = False
            for row in fileReader:
                if canProceed == True or row[1] == "2061058951":
                    canProceed = True
                    counter+=1
                    print("Writting message {0}, {1} in total".format(row[1], counter))
                else:
                    counter+=1
                    print("Skipped message {0}, {1} in total".format(row[1], counter))
                    continue

                userInsertQuery = """INSERT INTO User(Name) VALUES (%s) ON DUPLICATE KEY UPDATE Name = Name"""
                userData = (row[4],)
                cursor.execute(userInsertQuery, userData)
                connection.commit()

                # Parse date formatted as e.g. Sat May 16 23:58:44 UTC 2009
                postedOn = parser.parse(row[2], tzinfos={"PDT": -7 * 3600})
                # Perform sentiment analysis using VADER.
                sentiment_score = sentimentAnalyzer.polarity_scores(row[5])
                messageInsertQuery = """INSERT INTO Message(ID, User_Name, Posted_On, Text, Sentiment_Positive, Sentiment_Neutral, Sentiment_Negative, Sentiment_Compound) VALUES (%s, %s, %s, %s, %s, %s, %s, %s) ON DUPLICATE KEY UPDATE ID = ID"""
                messageData = (row[1], row[4], postedOn, row[5], sentiment_score["pos"], sentiment_score["neu"], sentiment_score["neg"], sentiment_score["compound"])
                cursor.execute(messageInsertQuery, messageData)
                connection.commit()

                hashtags = hashtag_re.findall(row[5])
```

```
                mentions = mention_re.findall(row[5])

                if len(hashtags) > 0:
                        hashtagInsertQuery = """INSERT INTO Hashtag(Name) VALUES (%s) ON DUPLICATE KEY UPDATE Name = Name"""
                        cursor.executemany(hashtagInsertQuery, [(val,) for val in hashtags])
                        connection.commit()

                        hashtagMessageInsertQuery = """INSERT INTO Message_Hashtag(Message_ID, Hashtag_Name) VALUES (%s, %s) ON DUPLICATE KEY UPDATE Message_ID = Message_ID"""
                        cursor.executemany(hashtagMessageInsertQuery, [(row[1], val) for val in hashtags])
                        connection.commit()

                if len(mentions) > 0:
                        cursor.executemany(userInsertQuery, [(val,) for val in mentions])
                        connection.commit()

                        mentionsInsertQuery = """INSERT INTO Mention(Message_ID, User_Name) VALUES (%s, %s) ON DUPLICATE KEY UPDATE Message_ID = Message_ID"""
                        cursor.executemany(mentionsInsertQuery, [(row[1], val) for val in mentions])
                        connection.commit()



                # # Add user to user dictionary, if not already there.
                # if row[4] not in users:
                #       user = User(row[4])
                #       users[row[4]] = user
```

```
                    # # Parse date formatted as e.g. Sat May 16 23:58:44 UTC 2009
                    # postedOn = parser.parse(row[2], tzinfos={"PDT": -7 * 3600})
                    # #postedOn = datetime.datetime.strptime(row[2], '%a %b %d
%H:%M:%S %Z %Y')
                    # # Perform sentiment analysis using VADER.
                    # sentiment_score = sentimentAnalyzer.polarity_scores(row[5])

                    # # Create message object.
                    # message = Message(row[1], postedOn, users[row[4]], row[5],
sentiment_score)
                    # # Insert message into user object.
                    # users[row[4]].messages.append(message)

                    # hashtags = hashtag_re.findall(message.text)
                    # mentions = mention_re.findall(message.text)
                    # counter +=1

except mysql.connector.Error as error:
        print("Failed to insert into MySQL table {}".format(error))

finally:
        if(connection.is_connected()):
                connection.close()
```

Παράρτημα Β
Βάση Δεδομένων

Β.1 Βάση δεδομένων για Sentiment140 και Emotion Classification

Η βάση δεδομένων αποτελείται από τα tables:

- Message: Περιέχει όλα τα μηνύματα που μαζεύει το σύστημα
- User: Περιέχει όλους τους χρήστες που έχουν τύχει επεξεργασίας από το σύστημα
- Mention: Συσχετίζει χρήστη με το μήνυμα στο οποίο υπήρξε αναφορά προς αυτόν
- Hashtag: Περιέχει όλα τα hashtag που έχουν τύχει επεξεργασίας από το σύστημα
- Message_Hashtag: Συσχετίζει τα hashtags με messages
- Emotion: Λίστα με συναισθήματα
- Emotion_Lexicon: Περιέχει τα δεδομένα του NRC Emotion Lexicon
- semeval_ei-reg-en-data: Περιέχει τα δεδομένα του dataset SemEval 2018 Task: 1 ei-reg
- semeval_e-c-en-data: Περιέχει τα δεδομένα του dataset SemEval 2018 Task: 1 e-c
- hec-data: Περιέχει τα δεδομένα του dataset Hashtag Emotion Corpus

Επίσης περιέχονται 2 views:

- v_combined_emotion_data: Συσσωματώνει τα 3 datasets που αναφέρουμε στο κεφάλαιο 4.
- v_emotion_lexicon_aggregated: Συσσωματώνει το emotion_lexicon ώστε να αφαιρέσει ή μεταβάλει κάποια συναισθήματα όπως καθορίζονται στο κεφάλαιο 4.

```
-- --------------------------------------------------------
-- Host:                    127.0.0.1
-- Server version:          10.4.8-MariaDB - mariadb.org binary distribution
```

```
-- Server OS:                    Win64
-- HeidiSQL Version:             10.2.0.5599
-- --------------------------------------------------------

/*!40101 SET @OLD_CHARACTER_SET_CLIENT=@@CHARACTER_SET_CLIENT */;
/*!40101 SET NAMES utf8 */;
/*!50503 SET NAMES utf8mb4 */;
/*!40014 SET @OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS, FOREIGN_KEY_CHECKS=0 */;
/*!40101 SET @OLD_SQL_MODE=@@SQL_MODE, SQL_MODE='NO_AUTO_VALUE_ON_ZERO' */;


-- Dumping database structure for insider_threat_db
CREATE DATABASE IF NOT EXISTS `insider_threat_db` /*!40100 DEFAULT CHARACTER SET utf8mb4 COLLATE utf8mb4_unicode_ci */;
USE `insider_threat_db`;

-- Dumping structure for table insider_threat_db.emotion
CREATE TABLE IF NOT EXISTS `emotion` (
  `Name` varchar(12) COLLATE utf8mb4_unicode_ci NOT NULL,
  KEY `Name` (`Name`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

-- Data exporting was unselected.

-- Dumping structure for table insider_threat_db.emotion_lexicon
CREATE TABLE IF NOT EXISTS `emotion_lexicon` (
  `Word` varchar(17) COLLATE utf8mb4_unicode_ci NOT NULL,
  `Emotion` varchar(12) COLLATE utf8mb4_unicode_ci NOT NULL,
  `Association` int(1) NOT NULL,
  PRIMARY KEY (`Word`,`Emotion`),
  KEY `FK_Lexicon_Emotion_Emotion` (`Emotion`),
```

```
  CONSTRAINT `FK_Lexicon_Emotion_Emotion` FOREIGN KEY (`Emotion`) REFERENCES `emotion` (`Name`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

-- Data exporting was unselected.

-- Dumping structure for table insider_threat_db.hashtag
CREATE TABLE IF NOT EXISTS `hashtag` (
  `Name` varchar(280) COLLATE utf8mb4_unicode_ci NOT NULL,
  PRIMARY KEY (`Name`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

-- Data exporting was unselected.

-- Dumping structure for table insider_threat_db.hec-data
CREATE TABLE IF NOT EXISTS `hec-data` (
  `ID` bigint(20) NOT NULL DEFAULT 0,
  `Text` varchar(800) COLLATE utf8mb4_unicode_ci NOT NULL,
  `Emotion` char(10) COLLATE utf8mb4_unicode_ci NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

-- Data exporting was unselected.

-- Dumping structure for table insider_threat_db.mention
CREATE TABLE IF NOT EXISTS `mention` (
  `Message_ID` bigint(20) NOT NULL,
  `User_Name` varchar(32) COLLATE utf8mb4_unicode_ci NOT NULL,
  PRIMARY KEY (`Message_ID`,`User_Name`),
  KEY `FK_Mention_User` (`User_Name`),
  CONSTRAINT `FK_Mention_Message` FOREIGN KEY (`Message_ID`) REFERENCES `message` (`ID`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `FK_Mention_User` FOREIGN KEY (`User_Name`) REFERENCES `user` (`Name`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;
```

```
-- Data exporting was unselected.

-- Dumping structure for table insider_threat_db.message
CREATE TABLE IF NOT EXISTS `message` (
  `ID` bigint(20) NOT NULL,
  `User_Name` varchar(32) COLLATE utf8mb4_unicode_ci NOT NULL,
  `Posted_On` datetime NOT NULL,
  `Text` varchar(560) COLLATE utf8mb4_unicode_ci NOT NULL,
  `Sentiment_Positive` double NOT NULL DEFAULT 0,
  `Sentiment_Neutral` double NOT NULL DEFAULT 0,
  `Sentiment_Negative` double NOT NULL DEFAULT 0,
  `Sentiment_Compound` double NOT NULL DEFAULT 0,
  `SHA1_Hash` char(40) COLLATE utf8mb4_unicode_ci NOT NULL DEFAULT '',
  `Added_On` datetime NOT NULL DEFAULT current_timestamp(),
  PRIMARY KEY (`ID`),
  KEY `FK_User` (`User_Name`),
  KEY `SHA1_Hash` (`SHA1_Hash`),
  CONSTRAINT `FK_User` FOREIGN KEY (`User_Name`) REFERENCES `user` (`Name`) ON
DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

-- Data exporting was unselected.

-- Dumping structure for table insider_threat_db.message_hashtag
CREATE TABLE IF NOT EXISTS `message_hashtag` (
  `Message_ID` bigint(20) NOT NULL,
  `Hashtag_Name` varchar(280) COLLATE utf8mb4_unicode_ci NOT NULL,
  PRIMARY KEY (`Message_ID`,`Hashtag_Name`),
  KEY `FK_Message_Hashtag_Hashtag` (`Hashtag_Name`),
  CONSTRAINT `FK_Message_Hashtag` FOREIGN KEY (`Message_ID`) REFERENCES
`message` (`ID`) ON DELETE CASCADE ON UPDATE CASCADE,
  CONSTRAINT `FK_Message_Hashtag_Hashtag` FOREIGN KEY (`Hashtag_Name`)
REFERENCES `hashtag` (`Name`) ON DELETE CASCADE ON UPDATE CASCADE
```

```
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

-- Data exporting was unselected.

-- Dumping structure for table insider_threat_db.selected_users
CREATE TABLE IF NOT EXISTS `selected_users` (
  `User_Name` varchar(32) COLLATE utf8mb4_unicode_ci NOT NULL,
  PRIMARY KEY (`User_Name`),
  CONSTRAINT `FK_selected_user_user` FOREIGN KEY (`User_Name`) REFERENCES `user`
(`Name`) ON DELETE CASCADE ON UPDATE CASCADE
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

-- Data exporting was unselected.

-- Dumping structure for table insider_threat_db.semeval_e-c-en-data
CREATE TABLE IF NOT EXISTS `semeval_e-c-en-data` (
  `ID` char(13) COLLATE utf8mb4_unicode_ci NOT NULL,
  `Text` varchar(160) COLLATE utf8mb4_unicode_ci NOT NULL,
  `anger` int(11) NOT NULL,
  `anticipation` int(11) NOT NULL,
  `disgust` int(11) NOT NULL,
  `fear` int(11) NOT NULL,
  `joy` int(11) NOT NULL,
  `love` int(11) NOT NULL,
  `optimism` int(11) NOT NULL,
  `pessimism` int(11) NOT NULL,
  `sadness` int(11) NOT NULL,
  `surprise` int(11) NOT NULL,
  `trust` int(11) NOT NULL
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

-- Data exporting was unselected.

-- Dumping structure for table insider_threat_db.semeval_ei-reg-en-data
```

```
CREATE TABLE IF NOT EXISTS `semeval_ei-reg-en-data` (
  `ID` char(14) COLLATE utf8mb4_unicode_ci NOT NULL,
  `Text` varchar(300) COLLATE utf8mb4_unicode_ci NOT NULL,
  `Emotion` char(10) COLLATE utf8mb4_unicode_ci NOT NULL,
  `Intensity` float NOT NULL DEFAULT 0
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

-- Data exporting was unselected.

-- Dumping structure for table insider_threat_db.user
CREATE TABLE IF NOT EXISTS `user` (
  `Name` varchar(32) COLLATE utf8mb4_unicode_ci NOT NULL,
  `Added_On` datetime DEFAULT current_timestamp(),
  PRIMARY KEY (`Name`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8mb4 COLLATE=utf8mb4_unicode_ci;

-- Data exporting was unselected.

-- Dumping structure for view insider_threat_db.v_combined_emotion_data
-- Creating temporary table to overcome VIEW dependency errors
CREATE TABLE `v_combined_emotion_data` (
	`Text` VARCHAR(800) NOT NULL COLLATE 'utf8mb4_unicode_ci',
	`emotion` VARCHAR(10) NOT NULL COLLATE 'utf8mb4_unicode_ci'
) ENGINE=MyISAM;

-- Dumping structure for view insider_threat_db.v_emotion_lexicon_aggregated
-- Creating temporary table to overcome VIEW dependency errors
CREATE TABLE `v_emotion_lexicon_aggregated` (
	`Word` VARCHAR(17) NOT NULL COLLATE 'utf8mb4_unicode_ci',
	`Emotion` VARCHAR(12) NOT NULL COLLATE 'utf8mb4_unicode_ci',
	`Association` INT(1) NULL
) ENGINE=MyISAM;

-- Dumping structure for view insider_threat_db.v_combined_emotion_data
```

```
-- Removing temporary table and create final VIEW structure
DROP TABLE IF EXISTS `v_combined_emotion_data`;
CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW `v_combined_emotion_data` AS
SELECT DISTINCT Text,
CASE(emotion)
	WHEN 'anger' THEN 'anger'
	WHEN 'disgust' THEN 'anger'
	WHEN 'fear' THEN 'fear'
	WHEN 'joy' THEN 'joy'
	WHEN 'sadness' THEN 'sadness'
	WHEN 'surprise' THEN 'neutral'
	ELSE emotion
END AS emotion
FROM `hec-data`
UNION
SELECT DISTINCT TEXT, emotion
FROM `semeval_ei-reg-en-data`
UNION
SELECT DISTINCT TEXT, emotion FROM
(
SELECT TEXT, 'anger' Emotion FROM `semeval_e-c-en-data` WHERE anger + disgust > 0
UNION
SELECT TEXT, 'fear' Emotion FROM `semeval_e-c-en-data` WHERE fear = 1
UNION
SELECT TEXT, 'joy' Emotion FROM `semeval_e-c-en-data` WHERE joy + optimism + love > 0
UNION
SELECT TEXT, 'sadness' Emotion FROM `semeval_e-c-en-data` WHERE sadness + pessimism > 0
UNION
SELECT TEXT, 'neutral' Emotion FROM `semeval_e-c-en-data` WHERE surprise + trust + anticipation > 0
) AS `semeval_e-c-en-data-aggregated` ;
```

```
-- Dumping structure for view insider_threat_db.v_emotion_lexicon_aggregated
-- Removing temporary table and create final VIEW structure
DROP TABLE IF EXISTS `v_emotion_lexicon_aggregated`;
CREATE ALGORITHM=UNDEFINED DEFINER=`root`@`localhost` SQL SECURITY DEFINER VIEW `v_emotion_lexicon_aggregated` AS SELECT l.Word, l.Emotion, MAX(l.Association) Association FROM
(SELECT DISTINCT
el.word,
CASE(el.Emotion)
	WHEN 'anticipation' THEN 'neutral'
	WHEN 'surprise' THEN 'neutral'
	WHEN 'trust' THEN 'neutral'
	WHEN 'disgust' THEN 'anger'
	WHEN 'positive' THEN 'joy'
	WHEN 'negative' THEN 'sadness'
	ELSE el.Emotion
END Emotion,
el.Association
FROM emotion_lexicon el
#WHERE el.Emotion NOT IN ('negative', 'positive')
) l
GROUP BY l.word, l.Emotion ;

/*!40101 SET SQL_MODE=IFNULL(@OLD_SQL_MODE, '') */;
/*!40014 SET FOREIGN_KEY_CHECKS=IF(@OLD_FOREIGN_KEY_CHECKS IS NULL, 1, @OLD_FOREIGN_KEY_CHECKS) */;
/*!40101 SET CHARACTER_SET_CLIENT=@OLD_CHARACTER_SET_CLIENT */;
```