

# **Ανοικτό Πανεπιστήμιο Κύπρου**

**Σχολή Θετικών και Εφαρμοσμένων Επιστημών**

## **Μεταπτυχιακή Διατριβή** **Στην Ασφάλεια Υπολογιστών και Δικτύων**



**FaceWallGrap: Χρήση Machine Learning για Παρακολούθηση και Συλλογή Πληροφοριών από το Facebook**

**Αμιλία Παναγιώτου**

**Επιβλέπων Καθηγητής**  
**Δρ. Σταύρος Σιαηλής**

**Δεκέμβριος 2017**

# **Ανοικτό Πανεπιστήμιο Κύπρου**

## **Σχολή Θετικών και Εφαρμοσμένων Επιστημών**

### **FaceWallGrap: Χρήση Machine Learning για Παρακολούθηση και Συλλογή Πληροφοριών από το Facebook**

**Αμιλία Παναγιώτου**

**Επιβλέπων Καθηγητής**  
**Δρ. 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 2017**

ΛΕΥΚΗ ΣΕΛΙΔΑ

## Περίληψη

Η επικοινωνία μεταξύ των ανθρώπων έχει γίνει πολύ πιο εύκολη με την χρήση των κοινωνικών μέσων δικτύωσης. Σε ένα πλανήτη με 2 δισεκατομμύρια ανθρώπους όπου ένα μεγάλο ποσοστό είναι ενεργοί χρήστες στα μέσα κοινωνικής δικτύωσης και κυρίως στο Facebook. Τα μέσα κοινωνικής δικτύωσης αποτελούν βασικό ρόλο στον τρόπο της επικοινωνίας μεταξύ των χρηστών αλλά και την έκφραση συναισθημάτων μέσα από τα μέσα κοινωνικής δικτύωσης. Η άμεση επικοινωνία που έχουν οι χρήστες καθημερινά αποτελεί ένα τεράστιο όγκο πληροφοριών που ανταλλάζονται με τις πλατφόρμες κοινωνικής δικτύωσης. Αναπόφευκτα η χρήση των μέσων κοινωνικής δικτύωσης οδηγεί τους χρήστες στην αποκάλυψη των προσωπικών τους δεδομένων αφού είναι απαραίτητα για την δημιουργία λογαριασμού στην οποιαδήποτε πλατφόρμα. Επιπλέον για τις εταιρίες που ασχολούνται με National Security οι πληροφορίες από τα profiles των χρηστών, όπου μπορούν να χρησιμοποιηθούν για να αποτρέψουν insider threats. Με αφορμή του μεγάλου όγκου πληροφοριών που προσφέρονται από τους χρήστες του Facebook παρέχοντας τα προσωπικά τους στοιχεία αλλά και το πώς εκφράζονται μέσα από τα μέσα κοινωνικής δικτύωσης, θέτουμε το ερευνητικό ερώτημα της άντλησης των πληροφοριών με αυτοματοποιημένο και μη ανιχνεύσιμο τρόπο όπου θα συμβάλει στην βοήθεια εταιριών που ασχολούνται με National Security να μαζέψουν τις πληροφορίες που χρειάζονται.

Στην απάντηση του παραπάνω ερευνητικού ερωτήματος, η παρούσα διατριβή πραγματοποιεί την άντληση πληροφοριών με ένα web crawler για OSINT, από το δημοφιλή μέσο κοινωνικής δικτύωσης, Facebook. Επιπλέον δημιουργώντας μια βάση δεδομένων με τα στοιχεία που θα συλλεχθούν από την πλατφόρμα κοινωνικής δικτύωσης που επιλέξαμε αναπτύσσεται μια μέθοδος με την οποία δημιουργείται μια learning machine για να μπορούν κάποιες λέξεις να φιλτράρονται. Οι λέξεις που φιλτράρονται χρησιμοποιούνται για ένα και μόνο σκοπό, να καταλάβουμε την ψυχολογία του χρήστη. Επίσης, τα αποτελέσματα δείχνουν ότι ο τρόπος χρήσης του Facebook μπορεί να αποτελέσει αντικείμενο εκμετάλλευσης αφού παρέχονται ευάλωτα προσωπικά δεδομένα των χρηστών (φωτογραφίες, τοποθεσίες και συναισθήματα χρηστών). Τέλος, αξίζει ότι οι εταιρίες National Security θα ήθελαν να γνωρίζουν την ψυχολογία των χρηστών. Τα αποτελέσματα για τις εναλλαγές της ψυχοσύνθεσης τους σε διάφορες χρονικές περιόδους παρουσιάζονται σε Ngrams γραφήματα και μπορούν να χρησιμοποιηθούν για πρόληψη από Insider Threats.

# Summary

Communication between people has become much easier with the use of social media tools. On a planet with 2 billion people where a large percentage of active users are in social media, especially Facebook. Social media have a key role to play in communicating between users and expressing emotions through social media. The direct communication that users have on a daily basis is an enormous amount of information exchanged with social networking platforms. Inevitably, the use of social networking means leads users to reveal their personal data as they are needed to create an account on any platform. Additionally for companies dealing with National Security information from user profiles where they can be used to prevent insider threats. In view of the large amount of information provided by Facebook users by providing their personal information and how they are expressed through social media tools, we are asking the question of how to extract information in an automated and undetectable way that will help companies help who are engaged in National Security to gather the information they need.

In the answer to the above research question, this dissertation explores information with a web crawler for OSINT, from the popular social media tool, Facebook. Additionally, by creating a database of data collected from the social networking platform we have chosen, a method is developed that creates a learning machine to allow some words to be filtered. The words that are filtered are used for a single purpose, to understand the user's psychology. Also, the results show that Facebook's use can be exploited by providing users with vulnerable personal data (photos, locations, and user feelings). Finally, it is worthwhile that National Security companies would like to know the users' psychology. The results for alternations of their psychosynthesis at different time periods are presented in Ngrams graphs and can be used to prevent Insider Threats.

# Ευχαριστίες

Θα ήθελα να εκφράσω τις θερμές μου ευχαριστίες στον επιβλέποντα καθηγητή μου Δρ. Σταύρο Σιαηλή για τη συνεχή καθοδήγηση, τις πολύτιμες πληροφορίες που μου έδωσε καθ' όλη τη διάρκεια της διατριβής.

Επιπρόσθετα θα ήθελα να ευχαριστήσω την οικογένεια μου και τους φίλους για την συνεχή υποστήριξη τα τελευταία δύομιση χρόνια.

# Περιεχόμενα

<b>1</b>	<b>Κεφάλαιο 1: Εισαγωγή</b> .....	<b>6</b>
1.1	Εισαγωγή .....	6-8
<b>2</b>	<b>Κεφάλαιο 2: Βιβλιογραφία</b> .....	<b>9-12</b>
<b>3</b>	<b>Κεφάλαιο 3: OSNIT</b> .....	<b>13</b>
3.1	Τι είναι το OSNIT .....	13-14
3.1.1	Web Crawler για το OSINT .....	14-15
3.1.2	Εργαλεία OSINT.....	15-20
3.1.3	Οφέλη και αδυναμίες του OSINT.....	21-22
3.2	Μέσα Κοινωνικής Δικτύωσης (Social Media) .....	22-23
3.2.1	Πλεονεκτήματα και Μειονεκτήματα που υπάρχουν στην χρήση των Μέσων Κοινωνικής Δικτύωσης (Social Media).....	23
3.2.2	Facebook, η δημοφιλέστερη πλατφόρμα Social Media.....	24
3.2.3	Διαδικτυακοί Κίνδυνοι.....	24
3.2.4	Εθισμός στα Μέσα Κοινωνικής Δικτύωσης (Social Media).....	25-26
<b>4</b>	<b>Κεφάλαιο 4 FaceWallGrap</b> .....	<b>27-28</b>
4.1	Εισαγωγή στην Python .....	29-30
4.1.1	Πρόγραμμα που χρησιμοποιήθηκε για την υλοποίηση του κώδικα .....	26-27
4.2	Βασικές λειτουργίες Python Facebook Crawler .....	30
4.2.1	Βιβλιοθήκη Python Selenium .....	30-32
4.2.2	Κλάση FacebookCrawler.py .....	32-33
4.2.3	Κλάση Facebookbot.py .....	34
4.3	Αποθήκευση των πληροφοριών .....	34
4.3.1	Κλάση Database.py .....	35
4.3.2	Ανάλυση Αποθήκευσης στην Βάση Δεδομένων .....	36
4.3.2.1	Φίλοι.....	36
4.3.2.2	Φωτογραφίες .....	37
4.3.2.3	Αναρτήσεις φίλων στα προφίλ τους (posts) .....	37-38
4.4	Βιβλιοθήκη Python ngrams .....	38

4.4.1	Κλάση NGrams.....	38-39
4.4.2	Πώς λειτουργεί.....	39-40
4.4.3	Ιστορία.....	40
4.4.4	Πως θα αξιοποιηθεί; .....	40-41
4.5	Εξόρυξη των πληροφοριών σε HTML Profiling για κάθε χρήστη .....	41-43
<b>5</b>	<b>Κεφάλαιο 5 Ανάλυση Αποτελεσμάτων</b> .....	<b>44</b>
5.1	Αποτελέσματα κλάσης Facebook_bot.py.....	44-47
5.2	Αποτελέσματα κλάσης view_posts.py.....	47-49
5.3	Αποτελέσματα κλάσης ngrams.py.....	49
5.3.1	Ψυχοσύνθεση χρηστών μέσα από το Facebook.....	49
5.3.1.1	Τι είναι ψυχοσύνθεση .....	49-50
5.3.1.2	Γιατί να χρησιμοποιήσει κάποιος την Εκφραστική γραφή; .....	50
5.3.2	Έκφραση συναισθημάτων στο Facebook .....	50-51
5.3.3	Έκφραση συναισθημάτων ανά χρονικές περιόδους στο Facebook .....	52
5.3.4	Ανάλυση γραφημάτων n-grams.....	52-55
<b>6</b>	<b>Κεφάλαιο 6 Συμπεράσματα</b> .....	<b>56</b>
6.1	Συμπεράσματα.....	56- 57
6.2	Μελλοντική δουλειά.....	57
	<b>Βιβλιογραφία</b> .....	<b>59-62</b>
<b>A</b>	<b>Python Classes</b> .....	<b>A-1</b>
A.1	Class Facebook_Bot.....	A-1
A.2	Class view_posts.....	A-3
A.3	Class FacebookCrawler.....	A-5
A.4	Class ngrams.....	A9
A.5	Class Database.....	A13



# Κεφάλαιο 1

## Εισαγωγή

### 1.1 Εισαγωγή

Στις μέρες μας οι περισσότεροι άνθρωποι είναι ενεργοί χρήστες στα μέσα κοινωνικής δικτύωσης και κυρίως στο Facebook όπου τα τελευταία χρόνια έχει ραγδαία εξέλιξη. Με αποτέλεσμα τα μέσα κοινωνικής δικτύωσης, μας δίνεται η δυνατότητα να έχουμε ένα τεράστιο όγκο πληροφοριών φανερώνει μια πρόκληση για την επεξεργασία και την ανάλυση των εν λόγω πληροφοριών. Οι πληροφορίες που παρέχει το Facebook είναι ένα ακατέργαστο υλικό που προσφέρεται δωρεάν και μπορεί να αξιοποιηθεί από εταιρίες που ασχολούνται με National Security για να καταλαβούν την ψυχολογία του χρήστη και στην συνέχεια να χρησιμοποιήσουν τις συγκεκριμένες πληροφορίες για να αποτρέψουν insider threads.

Ο όγκος των πληροφοριών που προσφέρεται από το Facebook είναι κυρίως προσωπικά δεδομένα των χρηστών. Η επεξεργασία των προσωπικών δεδομένων κάτω από προϋποθέσεις αποτελεί παράπτωμα χωρίς την συγκατάθεση των χρηστών και γι' αυτό υπάρχει σχετική νομοθεσία για την προστασία των προσωπικών δεδομένων στα κοινωνικά δίκτυα. Στις 14 Απριλίου 2016, το Ευρωπαϊκό Κοινοβούλιο ενέκρινε το Γενικό Κανονισμό για τα Προσωπικά δεδομένα. Ο κανονισμός έχει τεθεί σε ισχύ την

άνοιξη του 2016 και θα αρχίσει να εφαρμόζεται τον Μάιο του 2018. Ο κανονισμός περιγράφει τα δικαιώματα του υποκειμένου των δεδομένων, δηλαδή του ατόμου του οποίου τα προσωπικά δεδομένα υποβάλλονται σε επεξεργασία. [18] Τέλος, ο κανονισμός αναγνωρίζει το δικαίωμα των υποκειμένων των δεδομένων να υποβάλλουν καταγγελία σε εποπτική αρχή καθώς και το δικαίωμά τους για δικαστική προσφυγή και αποζημίωση έτσι οι εταιρίες είναι εκτεθειμένες σε αγωγές από τρίτους των οποίων χάθηκαν τα προσωπικά τους δεδομένα. [19]

Με αφορμή όλη αυτή την ακατέργαστη πληροφορία η παρούσα διατριβή έχει σκοπό να αξιολογήσει, να συλλέξει και να επεξεργαστεί για ότι πληροφορία παρέχετε από το Facebook, με την συγκατάθεση των χρηστών που υπάρχουν στον λογαριασμό του Facebook. Το υπόβαθρο του ερευνητικού ερωτήματος είναι με την χρήση Machine Learning θα παρακολουθεί και θα συλλέγει πληροφορίες χρησιμοποιώντας το Facebook. Στόχος λοιπόν είναι να γίνει επεξεργασία της ακατέργαστης ανοικτής πληροφορίας (OSINT) που παρέχει το Facebook χρησιμοποιώντας λέξεις ή/και σύμβολα φιλτράροντας τα για να έχουμε αποτελέσματα σε γραφήματα ngram. Οι λέξεις αυτές θα προέρχονται από την βάση δεδομένων όπου ένας web crawler για το OSINT έχει τραβήξει αρκετές πληροφορίες. Πιο κάτω αναφέρονται τα κεφάλαια που θα αναπτυχθούν στην παρούσα διατριβή για την υλοποίηση του σκοπού.

Αρχικά στο κεφάλαιο 2 αναφέρονται σχετικές έρευνες που σχετίζονται με τα συναισθήματα που εκφράζονται στα μέσα κοινωνικής και πώς παρουσιάζονται. Επίσης στο κεφάλαιο 3 παρουσιάζεται ο ορισμός του Open-source intelligence (OSINT) και πως συνδέετε με το Web-Crawler για την υλοποίηση του ερευνητικού ερωτήματος. Επιπλέον γίνεται αναφορά στα μέσα κοινωνικής δικτύωσης και κυρίως στο Facebook που βασίζεται η συλλογή και επεξεργασία πληροφοριών για την διεκπεραίωση της διατριβής. Τέλος, αναφέρονται εργαλεία OSINT που χρησιμοποιούνται από ερευνητές για διάφορους τρόπους συλλογής πληροφοριών. Στο επόμενο κεφάλαιο, το τέταρτο, το οποίο είναι και το μεγαλύτερο κεφάλαιο της παρούσας διατριβής αναφέρονται αναλυτικά οι κλάσεις που έχουν χρησιμοποιηθεί για την υλοποίηση του προγράμματος αλλά και οι λειτουργίες της κάθε μιας σε γραφήματα Ngrams. Τέλος γίνεται αναφορά στην γλώσσα προγραμματισμού και στα λογισμικά που χρησιμοποιήθηκαν. Στο κεφάλαιο 5 γίνεται ανάλυση της ψυχοσύνθεσης των χρηστών μέσα από τον γραπτό λόγο και την έκφραση των συναισθημάτων μέσα από λέξεις. Συνοψίζοντας παρουσιάζεται η ψυχοσύνθεση και τα συναισθήματα των χρηστών μέσα από την χρήση του Facebook. Τέλος γίνεται μια ανάλυση των αποτελεσμάτων που έχουν προκύψει από

τις μεθόδους που χρησιμοποιήθηκαν. Το τελευταίο κεφάλαιο της παρούσας διατριβής, το κεφάλαιο 6 κλείνει με γενικά συμπεράσματα και γίνεται αναφορά σε μελλοντικές εργασίες .

# Κεφάλαιο 2

## Βιβλιογραφία

Πολλές έρευνες έχουν γίνει για τα μέσα κοινωνικής δικτύωσης και πως επηρεάζουν τα συναισθήματα των ανθρώπων. Με αφορμή το δικό μου θέμα για την υλοποίηση της παρούσας διατριβής αναφέρονται πιο κάτω διάφορα θέματα που έχουν γίνει από άλλους ερευνητές.

Αρχικά, οι φοιτητές της Ψυχιατρικής Jeannette Golden, Ronán M. Conroy, Irene Bruce, Aisling Denihan, Elaine Greene, Michael Kirby και Brian A. Lawlor [39] έκαναν μια έρευνα με θέμα «Loneliness, social support networks, mood and wellbeing in community-dwelling elderly» για την Ψυχική υγεία ηλικιωμένων ανθρώπων. Σκοπός αυτής της μελέτης ήταν να εξεταστεί η σχέση μεταξύ κοινωνικού δικτύου, μοναξιάς, κατάθλιψης, άγχους και ποιότητας ζωής σε ηλικιωμένους που ζουν στο Δουβλίνο στην κοινότητα. Έλαβαν μέρος χίλια διακόσια ενενήντα εννέα άτομα ηλικίας 65 ετών και άνω, μέσω πρακτικών πρωτοβάθμιας περίθαλψης, ερωτήθηκαν στο σπίτι τους χρησιμοποιώντας το GMS-AGECAT. Το κοινωνικό δίκτυο αξιολογήθηκε χρησιμοποιώντας την τυπολογία του Wenger. Τα αποτελέσματα είχαν ως εξής το 35% των συμμετεχόντων ήταν μόνοι, με το 9% να το χαρακτηρίζει επώδυνο και το 6% ως

παρεμβατικό. Ομοίως, το 34% είχε ένα μη ολοκληρωμένο κοινωνικό δίκτυο. Ωστόσο, οι δύο δομές ήταν διακριτές: το 32% των συμμετεχόντων με ένα ολοκληρωμένο κοινωνικό δίκτυο ανέφεραν ότι ήταν μόνοι. Η μοναξιά ήταν μεγαλύτερη στις γυναίκες, στους χήρους και σε εκείνους με σωματική αναπηρία και αυξανόταν με την ηλικία, αλλά όταν οι μεταβλητές που σχετίζονται με την ηλικία ελέγχονταν για αυτή τη σχέση δεν ήταν σημαντική. Η ευεξία, η καταθλιπτική διάθεση και η απελπισία συνδέονταν ανεξάρτητα τόσο με τη μοναξιά όσο και με το μη ολοκληρωμένο κοινωνικό δίκτυο. Συγκεκριμένα, η μοναξιά εξήγησε τον υπερβολικό κίνδυνο κατάθλιψης στους χήρους. Ο κίνδυνος που οφείλεται στον πληθυσμό (PAR) που σχετίζεται με τη μοναξιά ήταν 61%, έναντι 19% για το μη ολοκληρωμένο κοινωνικό δίκτυο. Λαμβανόμενα από κοινού είχαν ένα ΠΑΑ 70%. Το συμπέρασμα της συγκεκριμένης έρευνας ήταν ότι η μοναξιά και τα κοινωνικά δίκτυα επηρεάζουν ανεξάρτητα τη διάθεση και την ευημερία των ηλικιωμένων, υποκρύπτοντας ένα πολύ σημαντικό ποσοστό καταθλιπτικής διάθεσης. [39]

Μια έρευνα από τους ερευνητές Kramer, A., Guillory, J. και Hancock, J. το 2014 [40] με θέμα «Experimental evidence of massive-scale emotional contagion through social networks» παρουσιάζει συναισθηματικές καταστάσεις που μπορούν να μεταφερθούν σε άλλους μέσω συναισθηματική μετάδοση στο Facebook, που οδηγεί τους ανθρώπους να βιώσουν τα ίδια συναισθήματα, χωρίς την ευαισθητοποίησή τους. Τα πειραματικά στοιχεία είναι ότι η συναισθηματική μόλυνση συμβαίνει χωρίς άμεση αλληλεπίδραση μεταξύ των ανθρώπων και στην πλήρη απουσία των μη λεκτικών σημείων. Η συναισθηματική μόλυνση είναι καθιερωμένη σε εργαστηριακά πειράματα, με τους ανθρώπους να μεταφέρουν θετικά και αρνητικά συναισθήματα σε άλλους μέσω του Facebook. Τα δεδομένα από ένα μεγάλο κοινωνικό δίκτυο πραγματικού κόσμου, που συλλέχθηκαν σε μια περίοδο 20 ετών, υποδεικνύουν ότι οι μακροχρόνιες διαθέσεις (π.χ. κατάθλιψη, ευτυχία) μπορούν να μεταφερθούν μέσω δικτύων [Fowler JH, Christakis NA (2008) BMJ337: a2338], αν και τα αποτελέσματα είναι αμφιλεγόμενα. Σε ένα πείραμα με ανθρώπους που χρησιμοποιούν το Facebook, δοκιμάζουμε εάν η συναισθηματική μόλυνση συμβαίνει έξω από την προσωπική αλληλεπίδραση μεταξύ των ατόμων, μειώνοντας το ποσό συναισθηματικού περιεχομένου στη ροή ειδήσεων. Όταν οι θετικές εκφράσεις μειώθηκαν, οι άνθρωποι παράγαγαν λιγότερες θετικές θέσεις και περισσότερες αρνητικές θέσεις. όταν οι αρνητικές εκφράσεις μειώθηκαν, συνέβη το αντίθετο πρότυπο. Αυτά τα αποτελέσματα δείχνουν ότι τα συναισθήματα που εκφράζουν οι άλλοι στο Facebook επηρεάζουν τα συναισθήματά μας, συνιστώντας

πειραματικές ενδείξεις για μαζική εξάπλωση μέσω κοινωνικών δικτύων. Αυτή η εργασία επίσης υποδηλώνει ότι, σε αντίθεση με τις επικρατούσες υποθέσεις, η αλληλεπίδραση μεταξύ των ανθρώπων και τα μη λεκτικά συνθήματα δεν είναι απολύτως απαραίτητα για τη συναισθηματική μόλυνση. [40]

Μια άλλη έρευνα βασισμένη στο θέμα «Social contagion theory: examining dynamic social networks and human behavior» από τους Christakis, N. και Fowler, J. το 2012 [41] εξετάζει την κοινωνική μόλυνση. Περιγράφει τις μεθόδους που χρησιμοποιήθηκαν για την εξέταση αρκετών δεδομένων με συμπληρωματικά δυνατά σημεία και αδυναμίες, όπως η Καρδιολογική Μελέτη Framingham, η Εθνική Διαχρονική Μελέτη της Εφηβικής Υγείας και άλλα σύνολα όπου γίνονται παρατήρησης και πειράματα που οι ερευνητές έχουν συγκεντρώσει. Περιγράφουν τις κανονικότητες που τους οδήγησαν να προτείνουν τα ανθρώπινα κοινωνικά δίκτυα και να παρουσιάζουν ιδιότητα «τριών επιπέδων επιρροής» και αναθεωρούν τις στατιστικές προσεγγίσεις που χρησιμοποιήσαμε για να χαρακτηρίσουμε τη διαπροσωπική επιρροή σε φαινόμενα τόσο διαφορετικά όπως η παχυσαρκία, το κάπνισμα, ευτυχία. Δεν ισχυρίζονται ότι αυτή η δουλειά είναι η τελευταία λέξη, αλλά πιστεύουν ότι παρέχει κάποια νέα, ενημερωτικά, και την τόνωση των στοιχείων σχετικά με την κοινωνική μόλυνση στα διαμήκης ακολουθούμενα δίκτυα. Μαζί με άλλους μελετητές, εργάζονται για την ανάπτυξη νέων μεθόδων για τον εντοπισμό των αιτιωδών επιπτώσεων χρησιμοποιώντας δεδομένα κοινωνικού δικτύου και πιστεύουν ότι ο τομέας αυτός είναι ώριμος για στατιστική εξέλιξη, καθώς οι τρέχουσες μέθοδοι έχουν γνωστούς και συχνά αναπόφευκτους περιορισμούς.

Τέλος μια άλλη έρευνα με θέμα «Happiness Is Assortative in Online Social Networks» [42] ενδέχεται να παρουσιάσει εξαιρετικά πολύπλοκες και προσαρμοστικές συλλογικές συμπεριφορές που προέρχονται από τα διαδικτυακά μέσα κοινωνικής δικτύωσης. Δεδομένου ότι τα συναισθήματα παίζουν τόσο σημαντικό ρόλο στην ανθρώπινη λήψη αποφάσεων, το πώς τα διαδικτυακά δίκτυα διαμορφώνουν τις ανθρώπινες συλλογικές καταστάσεις διάθεσης έχει γίνει ένα θέμα ιδιαίτερου ενδιαφέροντος από τους ερευνητές. Παρά την αυξανόμενη κοινωνική σημασία των διαδικτυακών μέσων κοινωνικών δικτύων, δεν είναι γνωστό αν η συνειρμική ανάμειξη ψυχολογικών καταστάσεων λαμβάνει χώρα σε καταστάσεις όπου οι κοινωνικοί δεσμοί μεσολαβούν αποκλειστικά από υπηρεσίες κοινωνικής δικτύωσης και ελλείψεις φυσικής

επαφής. Στην παρούσα έρευνα, παρουσιάζεται ότι η γενική ευτυχία, ή η υποκειμενική ευημερία(SWB), των χρηστών του Twitter, όπως μετράτε από ένα ρεκόρ 6 μηνών για τα μεμονωμένα tweets τους, είναι πράγματι ευαίσθητο σε όλο το κοινωνικό δίκτυο Twitter. Τα αποτελέσματά των ερευνητών υποδηλώνουν ότι τα διαδικτυακά μέσα κοινωνικής δικτύωσης μπορεί να υπόκεινται εξίσου στους κοινωνικούς μηχανισμούς που προκαλούν ανάμειξη σε πραγματικά κοινωνικά δίκτυα και ότι αυτή η ανάμειξη λαμβάνει χώρα στο επίπεδο της υποκειμενικής ευημερίας. Δεδομένου της αυξανόμενης επικράτησης των διαδικτυακών μέσων κοινωνικής δικτύωσης, η τάση τους να συνδέουν τους χρήστες με παρόμοια επίπεδα SWB μπορεί να είναι ένας σημαντικός παράγοντας για το πώς τα θετικά και αρνητικά συναισθήματα διατηρούνται και διαδίδονται μέσω της ανθρώπινης κοινωνίας. Η μελλοντική έρευνα μπορεί να επικεντρωθεί στο πώς οι καταστάσεις διάθεσης που αφορούν συγκεκριμένα γεγονότα μπορούν να μεταδώσουν και να επηρεάσουν τη συμπεριφορά των χρηστών στην "πραγματική ζωή". [42]

Συνοψίζοντας και λαμβάνοντας υπόψιν όλες τις έρευνες που αναφέρθηκαν πιο πάνω με οδήγησαν στην υλοποίηση της παρούσας διατριβής για την μελέτη της ψυχοσύνθεσης των χρηστών σε συνέχεια προηγούμενων ερευνών. Οι πιο πάνω ερευνητές αλλά και άλλοι ερευνητές αναφέρθηκαν στο πώς επηρεάζονται οι χρήστες μεταξύ τους. Έχει χρησιμοποιηθεί σε διάφορες έρευνες το OSINT και το Facebook αλλά σε λίγες περιπτώσεις για να δείξουν τις εναλλαγές στην ψυχοσύνθεση των χρηστών. Επιπλέον δεν έχουν γίνει αρκετές αναφορές από άλλους ερευνητές σε ερευνητικά ερωτήματα με learning machine, Web Crawler για την συλλογή και το φιλτράρισμα πληροφοριών σε βάση δεδομένων, αλλά όλες οι έρευνες έχουν γίνει πειραματικά με φυσική παρουσία των ανθρώπων και έρευνες από 6 μήνες μέχρι 2 χρόνια. Σε αντίθεση με την έρευνα που θα παρουσιάσω όπου οι χρήστες είναι οι φίλοι στο Facebook και είναι μια σύντομη έρευνα 4 περίπου μηνών.

# Κεφάλαιο 3

## OSINT

### 3.1 Τι είναι το OSINT;

Το Open-source intelligence (OSINT) ορίζεται από τον Αμερικανό Διευθυντή Εθνικής Υπηρεσίας Πληροφοριών και το Υπουργείο Άμυνας των ΗΠΑ (DoD), ως «Το Open-source intelligence (OSINT) παράγεται από πληροφορίες διαθέσιμες στο κοινό που συλλέγονται, αξιοποιούνται και διαδίδονται εγκαίρως σε κατάλληλο κοινό για την αντιμετώπιση μιας συγκεκριμένης απαίτησης πληροφοριών. »[1] Σύμφωνα με τον επιστήμονα Jeffrey T. Richelson, « η απόκτηση ανοικτής δωρεάν πληροφορίας (Open-source intelligence (OSINT)) περιλαμβάνει προφορικό, γραπτό ή ηλεκτρονικό μεταδιδόμενο υλικό που μπορεί να αποκτηθεί νόμιμα. Εκτός από έγγραφα και βίντεο που είναι διαθέσιμα μέσω του Διαδικτύου ή που παρέχονται από τις συμμαχικές δυνάμεις ελέγχου ενός χώρου ή ιστοτόπου που λειτουργούσε προηγουμένως μια ξένη κυβέρνηση ή τρομοκρατικής ομάδα ». [2]

Ο ερευνητής ασφάλειας Mark M. Lowenthal χαρακτηρίζει το OSINT ως «όλες οι πληροφορίες που μπορούν να αντληθούν από την συλλογή είναι όλα τα είδη μέσω ενημέρωσης, κυβερνητικές εκθέσεις και άλλα έγγραφα, επιστημονικής έρευνας και εκθέσεις, εμπορικοί προμηθευτές πληροφοριών, Διαδίκτυο κ.ο.κ. [34] Οι πληροφορίες ανοιχτού κώδικα δεν απαιτεί κανενός είδους τεχνικής παράνομης συλλογής για να τις αποκτήσει και ότι πρέπει να αποκτηθούν με μέσα που πληρούν εξ ολοκλήρου πνευματικές και εμπορικές απαιτήσεις των πωλητών. »[3]

Με λίγα λόγια το Open Source Intelligence (OSINT) είναι η συλλογή και ανάλυση πληροφοριών που συλλέγονται από δημόσιες ή ανοιχτές πηγές. Το OSINT χρησιμοποιείται κυρίως στις λειτουργίες εθνικής ασφάλειας, επιβολής του νόμου και επιχειρηματικής ευφυΐας και έχει αξία για τους αναλυτές που χρησιμοποιούν μη



ευαίσθητες πληροφορίες για την απάντηση σε διαβαθμισμένες, μη ταξινομημένες ή ιδιόκτητες απαιτήσεις πληροφοριών σε όλα τα προηγούμενα επιστημονικά πεδία. Το OSINT μπορεί να συλλέξει πληροφορίες από συγκεκριμένες πηγές όπως τα μέσα μαζικής ενημέρωσης όπου συμπεριλαμβάνουν εφημερίδες, περιοδικά, ραδιόφωνο και τηλεόραση από διάφορες χώρες. Επιπλέον ενδιαφέρων πηγές διαθέσιμων πληροφοριών είναι το διαδίκτυο, οι ηλεκτρονικές δημοσιεύσεις, οι ανοικτές ομάδες συζήτησης, το YouTube και τα μέσα κοινωνικής δικτύωσης όπως Facebook, Twitter, Instagram. Συγκεκριμένα η πηγή των μέσων κοινωνικών δικτύωσης και του διαδικτύου θεωρείται πολύ εύκολη από οποιαδήποτε άλλη ανοικτή πηγή αφού είναι εύκολα προσβάσιμες.

Στοιχεία δημόσιας διοίκησης, οι εκθέσεις δημόσιας κυβέρνησης, οι προϋπολογισμοί, ακροάσεις, οι τηλεφωνικοί κατάλογοι, οι συνεντεύξεις Τύπου, ιστοτόπους και ομιλίες. Οι πηγές αυτές προέρχεται από επίσημη πηγή, είναι προσιτές στο κοινό και μπορούν να χρησιμοποιηθούν ανοιχτά και ελεύθερα. Τέλος οι πηγές αυτές μπορούν να χρησιμοποιηθούν για να εξυπηρετήσουν National Security εταιρίες για την εξιχνίαση εγκλημάτων.

Το OSINT διακρίνεται από διάφορες έρευνες ότι εφαρμόζει τη διαδικασία νοημοσύνης για να δημιουργήσει προσαρμοσμένες γνώσεις που υποστηρίζουν μια συγκεκριμένη απόφαση από ένα συγκεκριμένο άτομο ή ομάδα. [4]

### **3.1.1 Web Crawler για το OSINT**

Ένας Web crawler είναι πρόγραμμα το οποίο διαβάζει τον ιστοτόπο και άλλες πληροφορίες με ένα αυτόματο τρόπο για να δημιουργήσει καταχωρήσεις για ένα ευρετήριο μηχανών αναζήτησης. Η διαδικασία αυτή αναφέρεται και ως crawling. Επίσης ένα τέτοιο πρόγραμμα είναι γνωστό ως bot, Web spider ή Web robot. Οι ανιχνευτές προγραμματίζονται συνήθως για να επισκέπτονται ιστότοπους που έχουν υποβληθεί από τους ιδιοκτήτες τους ως νέοι ή ενημερωμένοι. Ολόκληροι ιστότοποι ή συγκεκριμένες σελίδες μπορούν να επισκέπτονται και να αναπροσαρμόζονται επιλεκτικά. Ο Web crawler κέρδισε το όνομα του επειδή ανιχνεύει μέσω ιστότοπου μια σελίδα τη φορά, ακολουθώντας τις συνδέσεις με άλλες σελίδες του ιστότοπου μέχρι να διαβάσουν όλες οι σελίδες. Επιπλέον ένας Web crawler μπορεί να χρησιμοποιηθεί για την αυτόματη συντήρηση ενός ιστότοπου (Web site), όπως για παράδειγμα για τον έλεγχο συνδέσμων ή για την επιβεβαίωση HTML κώδικα, καθώς επίσης μπορούν να

χρησιμοποιηθούν για την συλλογή συγκεκριμένου τύπου πληροφοριών από το web, όπως για παράδειγμα την συλλογή e-mail διευθύνσεων (spamming, spambot). [5]  
Με αυτό τον τρόπο θα δημιουργηθεί μια βάση δεδομένων για την αποθήκευση των πληροφοριών που θα συλλέγει ο Web crawler για το OSINT για να μπορεί να χρησιμοποιηθεί την υλοποίηση του FaceWallGrap. [5]

### 3.1.2 Εργαλεία OSINT

Για έναν ερευνητή ασφαλείας είναι απαραίτητα τα εργαλεία OSINT αφού αναφέρεται σε πληροφορίες που προέρχονται από πηγές που είναι διαθέσιμες στο κοινό. Βασικά, τα εργαλεία OSINT χρησιμοποιούνται στη φάση αναγνώρισης για να συλλέξουν όσο το δυνατόν περισσότερες πληροφορίες σχετικά με ν χρήστη. Αυτά τα εργαλεία ανοικτού κώδικα χρησιμοποιούν λειτουργίες τεχνητής νοημοσύνης για τη συγκέντρωση δεδομένων από τον Ιστό με όλες τις πιθανές αντιστοιχίες την απόκτηση του στόχου. Με τα εργαλεία OSINT, η διαδικασία αναγνώρισης γίνεται χωρίς να το αντιλαμβάνεται ο χρήστης, επιτρέποντας μια αποδοτικότερη μείωση του στόχου. Με την χρήση εργαλείων ανοικτού κώδικα μειώνει δραστικά τον αριθμό των αλλαγών και των συνδυασμών που πρέπει να αντιμετωπίσουν όσον αφορά τις πληροφορίες που συλλέγονται. Αυτό οδηγεί σε έναν αποτελεσματικό συνδυασμό κλασικών επιθέσεων κοινωνικής μηχανικής στο χρήστη, οι οποίες μπορούν να χρησιμοποιηθούν για τη συλλογή περισσότερων πληροφοριών. Το OSINT μπορεί επίσης να χρησιμοποιηθεί για αποτελεσματική ανακάλυψη στόχων και επακόλουθες επιθέσεις phishing. [27]  
Μερικά από τα δημοφιλή εργαλεία του OSINT είναι τα πιο κάτω:

1. **Maltego:** Το Maltego είναι ένα εξαιρετικά ισχυρό εργαλείο OSINT, που καλύπτει την προσωπική αναγνώριση. Η υποδομή του Maltego επιτρέπει τη συγκέντρωση ευαίσθητων δεδομένων σχετικά με τον οργανισμό-στόχο, τις διευθύνσεις ηλεκτρονικού ταχυδρομείου των υπαλλήλων, τα εμπιστευτικά αρχεία που χειρίζονται απρόσεκτα, τους εσωτερικούς αριθμούς τηλεφώνου, τις εγγραφές DNS, τις πληροφορίες διεύθυνσης IP, τη γεωγραφική θέση του δικτύου, τους διακομιστές MX επί. Η συλλογή τέτοιων δεδομένων - γνωστή στο Maltego ως μετασχηματισμοί - πρέπει να δημιουργηθεί δημιουργικά ώστε να επιτευχθούν τα καλύτερα αποτελέσματα. Η προσωπική αναγνώριση του Maltego, από την άλλη πλευρά, βοηθά στη συλλογή πληροφοριών συγκεκριμένων προσώπων, όπως η δραστηριότητα κοινωνικής δικτύωσης, οι διευθύνσεις ηλεκτρονικού

ταχυδρομείου, οι ιστοτόποι που σχετίζονται με το άτομο, οι αριθμοί τηλεφώνου κλπ. Αυτό συμβαίνει με τη χρήση μηχανών αναζήτησης στο Διαδίκτυο, τις οποίες το Maltego επικοινωνεί αποτελεσματικά, για να συγκεντρώσει όλες τις πληροφορίες που χρειάζονται.



Εικόνα 1: Εργαλείο Maltego

2. **Shodan:** Το Shodan σημαίνει Sentient Hyper Optimized Data Access Network. Σε αντίθεση με τις παραδοσιακές μηχανές αναζήτησης που ανιχνεύουν τον ιστότοπο για την εμφάνιση αποτελεσμάτων, ο Shodan επιχειρεί να αρπάξει δεδομένα από τις θύρες. Αναπτύχθηκε από τον John Matherly, και είναι διαθέσιμο ως δωρεάν έκδοση, επαγγελματική έκδοση και πληρωμένη έκδοση. Η δωρεάν έκδοση παρέχει έως 50 αποτελέσματα, και εάν κάποιος χρειαστεί περισσότερα αποτελέσματα πρέπει να προμηθεύσει την πληρωμένη έκδοση. Η δημιουργική χρήση του εργαλείου Shodan OSINT συμβάλλει στην εύρεση των ευάλωτων υπηρεσιών ενός διακομιστή Web, ο οποίος αποτελεί μια πολύ σημαντική πτυχή της φάσης αξιολόγησης της ευπάθειας. Το Shodan χρησιμοποιά διάφορα φίλτρα όπως χώρα, θύρα, λειτουργικό σύστημα και ονόματα κεντρικού υπολογιστή που είναι διαθέσιμα με αυτό το εργαλείο.

3. Το **Metagoofil** είναι ένα πολύ ισχυρό εργαλείο συλλογής πληροφοριών OSINT, το οποίο αναπτύχθηκε από την Edge Security. Στην ουσία, το Metagoofil χρησιμοποιείται για την εξαγωγή μεταδεδομένων από το στόχο. Υποστηρίζει διάφορους τύπους αρχείων, όπως pdf, doc, xls και ppt. Αυτό το εργαλείο ανοιχτού κώδικα μπορεί να χρησιμοποιηθεί για την εξαγωγή διευθύνσεων MAC από αυτά τα αρχεία, δίνοντας έτσι στον εισβολέα μια ιδέα για το είδος του δικτύου που χρησιμοποιείται στην εγκατάσταση στόχου. Σε συνδυασμό με τα ένστικτα και τη νοημοσύνη του εισβολέα, το Metagoofil μπορεί να χρησιμοποιηθεί για να μαντέψει το είδος του λειτουργικού συστήματος, τα ονόματα δικτύων κ.ο.κ. Μια επίθεση μπορεί να εκτελεστεί, αφού συγκεντρωθούν αρκετές πληροφορίες από τα μεταδεδομένα των αρχείων. Με τα μεταδεδομένα που έχουν ληφθεί μέσω του Metagoofil, είναι δυνατή η εξαγωγή πληροφοριών της διαδρομής, οι οποίες μπορούν να χρησιμοποιηθούν για τη χαρτογράφηση του δικτύου. Τα αποτελέσματα εμφανίζονται σε μορφή HTML.

```
~$ metagoofil -h
*****
*MetaGooFil Ver. 1.4c          *
*Coded by Christian Martorella *
*Edge-Security Research      *
*cmartorella@edge-security.com *
*****

MetaGooFil 1.4

usage: metagoofil options

    -d: domain to search
    -f: filetype to download (all,pdf,doc,xls,ppt,odp,ods, etc)
    -l: limit of results to work with (default 100)
    -o: output file, html format.
    -t: target directory to download files.

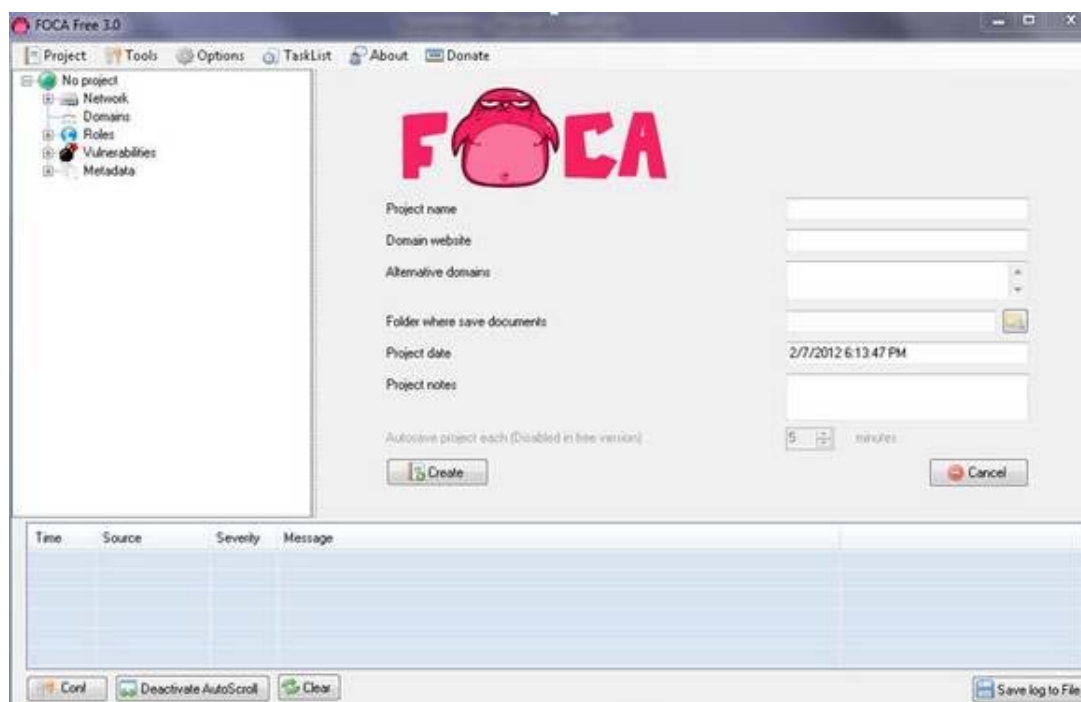
Example: metagoofil.py -d microsoft.com -l 20 -f all -o micro.html -t micro-files
```

Εικόνα 2: Έναρξη εργαλείου Metagoofil

4. Το Google συμβαίνει να είναι το πιο ισχυρό εργαλείο OSINT για να εκτελέσει επιθέσεις ενός χρήστη και αποτελεί τη βάση για εργαλείο **το GHDB - το Google Hacking DataBase**. Χρησιμοποιώντας το εργαλείο αυτό, μια βάση SQL σε έναν τυχαίο ιστότοπο μπορεί να πραγματοποιηθεί μέσα σε 0,2 δευτερόλεπτα. Οι

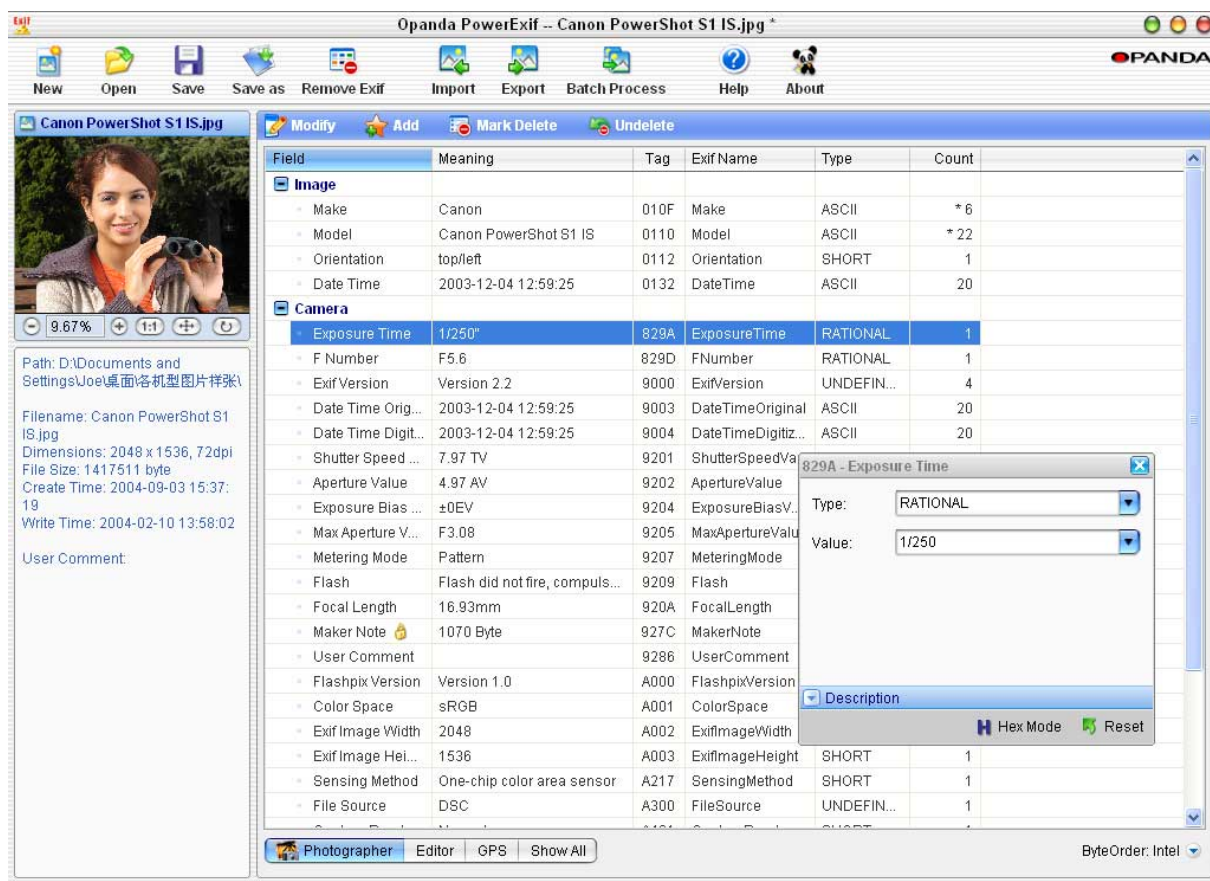
ειδικά επεξεργασμένες λέξεις που δίδονται ως είσοδος στην Google ονομάζονται dorks ή googledorks. Αυτά τα dorks GHDB μπορούν να χρησιμοποιηθούν για να αποκαλύψουν ευπαθείς διακομιστές στο Διαδίκτυο, να συλλέξουν ευαίσθητα δεδομένα και ευάλωτα αρχεία που τροποποιούνται. Η αποτελεσματική χρήση του GHDB μπορεί να κάνει τη διαδικασία hacking ευκολότερη. Το Exploit DB διατηρεί μια συλλογή από googledorks κάτω από ένα τμήμα που ονομάζεται GHDB

5. Το **FOCA** είναι ένα εργαλείο χαρτογράφησης δικτύου το οποίο μπορεί να χρησιμοποιηθεί για το OSINT. Μπορεί να αναλύσει τα μεταδεδομένα από διάφορα αρχεία, συμπεριλαμβανομένων αρχείων doc, pdf και ppt. Το FOCA μπορεί επίσης να απαριθμήσει τους χρήστες, τους φακέλους, τα μηνύματα ηλεκτρονικού ταχυδρομείου, το λογισμικό που χρησιμοποιείται, το λειτουργικό σύστημα και άλλες χρήσιμες πληροφορίες. Οι επιλογές προσαρμογής διατίθενται σε αυτό το εργαλείο OSINT. Για πιο συγκεκριμένες πληροφορίες και λεπτομέρειες σχετικά με τις ανασφαλείς μεθόδους, παρέχεται μια επιλογή ανίχνευσης. Τα μεταδεδομένα μπορούν να εξαχθούν από ένα μόνο αρχείο ή από πολλά αρχεία. Το FOCA είναι επομένως ένα εξαιρετικό εργαλείο στη φάση αναγνώρισης, για να εξαγάγει πληροφορίες από τα μεταδεδομένα.



**Εικόνα 3: Εργαλείο FOCA**

6. Οι συσκευές που έχουν την δυνατότητα να βγάλουν ψηφιακές φωτογραφίες χρησιμοποιούν ένα πρότυπο για να καθορίσουν τις μορφές για τις εικόνες και τους ήχους που καταγράφονται χρησιμοποιώντας αυτά. Αυτό το πρότυπο ονομάζεται **μορφή ανταλλαγής αρχείου εικόνας (EXIF)**. Υπάρχουν διάφοροι θεατές δεδομένων EXIF. Παρέχουν λεπτομέρειες όπως τύπος κάμερας, εστιακή απόσταση, τύπος φακού κλπ. Το πιο σημαντικό, οι ερευνητές δεδομένων EXIF παρέχουν τις πληροφορίες γεωγραφικής θέσης που είναι αποθηκευμένες για κάθε εικόνα. Στην πραγματικότητα, από όλα τα smartphones έχει ενεργοποιηθεί η ρύθμιση GPS. Επομένως, αυτό μπορεί να διαρρεύσει τη θέση όπου λήφθηκε η εικόνα. Η ακρίβεια είναι τέτοια ώστε το γεωγραφικό πλάτος και το γεωγραφικό μήκος θα παρέχονται από το πρόγραμμα προβολής δεδομένων EXIF κατά την εξαγωγή των δεδομένων EXIF, διαγράφοντας έτσι πολύ ιδιωτικές πληροφορίες.



**Εικόνα 4: Χρήση εργαλείου Opanda PowerEXIF**

7. Υπάρχουν διάφορες ιστοσελίδες και εργαλεία OSINT διαθέσιμα στο διαδίκτυο που μπορούν να χρησιμοποιηθούν για να βρουν δημόσιες πληροφορίες για ένα συγκεκριμένο άτομο. Οι ιστότοποι PeekYou και Lullar επιτρέπουν τη συγκέντρωση πληροφοριών σχετικά με ένα άτομο που είναι διαθέσιμο σε

διάφορους ιστότοπους κοινωνικής δικτύωσης. Το **Wayback Machine** είναι ένας ιστότοπος που μπορεί να χρησιμοποιηθεί για την εύρεση προηγούμενων εκδόσεων ιστοσελίδων, επιτρέποντας σε κάποιον να βλέπει ιστότοπους σε προηγούμενα εκδόσεις. Αυτά τα εργαλεία αναγνώρισης είναι χρήσιμα για την εκτέλεση επιθέσεων κοινωνικής μηχανικής. Το EDGAR, το σύστημα συλλογής, ανάλυσης και ανάκτησης ηλεκτρονικών δεδομένων, είναι ένας άλλος ιστότοπος που παρέχει πρόσβαση σε πληροφορίες μιας εταιρείας που διαφορετικά δύσκολα θα μπορούσαν να αποκτηθούν. Στη συνέχεια, υπάρχει YouGetSignal, παρέχοντας εργαλεία OSINT για να ελέγξετε τους αριθμούς τηλεφώνου, τις διευθύνσεις IP, τα στοιχεία των υπαλλήλων, τη γεωγραφική τοποθεσία, τον εντοπισμό κλπ. [27]

### 3.1.3 Οφέλη και αδυναμίες του OSINT

Το OSINT είναι συνήθως λιγότερο δαπανηρό και παρουσιάζει λιγότερους κινδύνους από άλλες μεθόδους συλλογής πληροφοριών και πηγές όπως η ανθρώπινη ευφυΐα (HUMINT). Το OSINT μπορεί επίσης να ευαισθητοποιήσει σχετικά με άλλες εξελίξεις που μπορεί να έχουν παραλειφθεί ή να μην θεωρούνται προτεραιότητα. Το μεγαλύτερο πλεονέκτημα του OSINT είναι η διαθεσιμότητά του. Απαιτεί επίσης λιγότερη εργασία από την HUMINT ή τα άλλα τεχνικά INTs. Το OSINT είναι εξαιρετικά ευεργετικό και για την οργάνωση λαθραίων δεδομένων σε ένα ευρύτερο πλαίσιο. Το OSINT είναι χρήσιμο για την αποκάλυψη ξένων ειδήσεων που δεν είναι διαθέσιμες σε επίσημες αγορές. Το OSINT υπήρξε επίσης εξαιρετικά πολύτιμο για την αντιμετώπιση των καταστροφών και την ανθρωπιστική βοήθεια. Οι πληροφορίες που συλλέγονται από τα κανάλια SMS (γραπτών μηνυμάτων) και τα κανάλια κοινωνικής δικτύωσης έχουν κάνει τη διαφορά μεταξύ ζωής και θανάτου μετά από καταστροφή. Ωστόσο, αυτές οι νέες χρήσεις του OSINT αποτελούν ένα νέο ζήτημα. Σήμερα, η συμμετοχή από τόσες κυβερνητικές υπηρεσίες, έθνη και οργανώσεις δημιουργούν ένα τεράστιο πρόβλημα οργάνωσης και συντονισμού. [28]

Από την άλλη πλευρά, ο όγκος είναι η κύρια αδυναμία του OSINT. Και ενώ οι υπολογιστές έχουν βελτιώσει τις δυνατότητες ανάλυσης της πληροφορίας, κάποιοι υποστηρίζουν ότι το σύνολο των αποτελεσμάτων της Intel δεν έχει βελτιωθεί. Επιπλέον, σε σύγκριση με άλλα INTs, το OSINT μπορεί να γίνει πρόκληση για να λειτουργήσει σε σχέση με την εξαπάτηση λόγω της μεταβλητότητάς του. Μια άλλη σημαντική αδυναμία του OSINT είναι η πιθανότητα παραπληροφόρησης, μυστικών



μηνυμάτων ή απλών σκουπιδιών. Ενδεχομένως ένα από τα χειρότερα μειονεκτήματα του OSINT είναι ότι η επανάσταση πληροφόρησης επηρέασε τις δυνατότητες της Intelligence Community (IC) προς την εκπλήρωση του συνήθους ρόλου της στην παροχή αποκλειστικής πληροφόρησης στους υπεύθυνους χάραξης πολιτικής. Όσον αφορά τις πιο πρόσφατες πηγές μέσων που βασίζονται στο Web, η ποσότητα, η ποικιλία και η ταχύτητα των πληροφοριών αυξάνονται ανά λεπτό. Η πρόκληση σήμερα δεν είναι να «συνδέσουμε τις τελείες» όπως πριν αλλά μάλλον να συστηματοποιήσουμε τη ροή των πληροφοριών. Πρόκειται για τη διαφοροποίηση μεταξύ "θορύβου και σημείων", συν τις πηγές επικύρωσης χωρίς καθυστέρηση και έγκαιρα, προκειμένου να υποστηριχθούν εξίσου οι υπεύθυνοι χάραξης πολιτικής και ο στρατός. Από την άλλη πλευρά, ο όγκος μπορεί επίσης να βοηθήσει τον αναλυτή να αποκαλύψει προκατειλημμένες ή παραπλανητικές πληροφορίες, καθώς μπορεί να συγκρίνει πολλές πηγές. Τα ατελείωτα αντίγραφα που πρέπει να αναλυθούν συντρίβουν τους σημερινούς αναλυτές και, κατά συνέπεια, η επιδίωξη για πιο αποτελεσματικά αναλυτικά εργαλεία εξακολουθεί να αποτελεί μεγάλη πρόκληση για την IC. Ως αποτέλεσμα, να προσπαθεί να δημιουργήσει εργαλεία οπτικοποίησης που θα επιτρέπουν στους αναλυτές να εντοπίζουν πιο κρίσιμα στοιχεία πιο αποτελεσματικά. [29]

### **3.2 Μέσα Κοινωνικής Δικτύωσης (Social Media)**

Με τον όρο Μέσα Κοινωνικής Δικτύωσης εννοούμε την κοινωνική διάδραση μεταξύ ανθρώπων που δημιουργούν, μοιράζονται ή ανταλλάσσουν πληροφορίες και ιδέες μέσα σε εικονικές κοινότητες και δίκτυα. Οι Andreas M. Kaplan και Michael Haenlein ορίζουν τα Μέσα Κοινωνικής Δικτύωσης ως "ένα σύνολο διαδικτυακών εφαρμογών που οικοδομούνται πάνω στα ιδεολογικά και τεχνολογικά θεμέλια του Web 2.0, κι επιτρέπουν τη δημιουργία και την ανταλλαγή περιεχομένου από χρήστες [33]. Επιπλέον, τα Μέσα Κοινωνικής Δικτύωσης εξαρτώνται από τεχνολογίες κινητών τηλεφώνων και διαδικτύου προκειμένου να δημιουργηθούν διαδραστικές πλατφόρμες μέσω των οποίων άτομα και κοινότητες μοιράζονται, συν-δημιουργούν, συζητούν, και αλλάζουν περιεχόμενο προερχόμενο από χρήστες. [23]

Σύμφωνα με τη Nielsen, οι χρήστες του Διαδικτύου εξακολουθούν να ξοδεύουν περισσότερο χρόνο στα Μέσα Κοινωνικής Δικτύωσης απ' ό,τι σε οποιαδήποτε άλλη ιστοσελίδα. Την ίδια ώρα, ο συνολικός χρόνος που ξοδεύεται στα Μέσα Κοινωνικής Δικτύωσης στις ΗΠΑ σε ηλεκτρονικούς υπολογιστές και κινητές συσκευές αυξήθηκε



κατά 37% σε 121 δισ. λεπτά τον Ιούλιο του 2012, συγκρινόμενο με τα 88 δισ. λεπτά τον Ιούλιο του 2011. [22]

Τα Social Media δεν έχουν ένα συγκεκριμένο ορισμό αφού μπορούν να περιορίσουν το εύρος τους που κυριαρχούν στην χρήση των μέσων κοινωνικής δικτύωσης, είναι το Facebook, MySpace, LinkedIn, Twitter.

Εναλλακτική ορολογία όπως είναι η κοινωνική δικτύωση ή online κοινωνικά δίκτυα, είναι οι web-based υπηρεσίες που επιτρέπουν στους χρήστες να κάνουν προσωπικά προφίλ, να δημιουργούν περιεχόμενο, και να μοιράζονται τα μηνύματα με άλλους χρήστες του συστήματος. [6] Το έργο του προσδιορισμού των κοινωνικών μέσων μαζικής ενημέρωσης γίνεται πιο δύσκολο από το γεγονός ότι είναι συνεχώς σε κατάσταση αλλαγής, καθώς η κάθε πλατφόρμα κοινωνικής δικτύωσης προσπαθεί να δημιουργήσει νέα ή βελτιωμένα χαρακτηριστικά που θα καλύψουν τις ανάγκες των χρηστών. Μια σύντομη περιγραφή των λειτουργιών του Facebook απεικονίζει τις πιο πρόσφατες δυνατότητες δικτύων κοινωνικής δικτύωσης. Στο εσωτερικό του Facebook, οι χρήστες μπορούν να στέλνουν μηνύματα, να προσθέτουν νέους φίλους, να ενημερώνουν το προσωπικό προφίλ, να συμμετέχουν σε ομάδες, να αναπτύσσουν εφαρμογές, καθώς επίσης να ενημερώνονται για τους άλλους χρήστες μέσω των online προφίλ τους.[7]

### **3.2.1 Πλεονεκτήματα και Μειονεκτήματα που υπάρχουν στην χρήση των Μέσων Κοινωνικής Δικτύωσης (Social Media)**

Τα πλεονεκτήματα των μέσων κοινωνικής δικτύωσης είναι η δυνατότητα δημιουργίας δεσμών με πολύ μεγάλο αριθμό ατόμων, εφόσον το διαδίκτυο συγκεντρώνει πλήθος άτομα από όλο τον κόσμο, με άτομα που μπορεί να βρίσκονται σε μεγάλη γεωγραφική απόσταση το ένα από το άλλο, εφόσον το Διαδίκτυο καταργεί τις αποστάσεις. Αυτή η δυνατότητα δεν υπάρχει στα στενά κοινωνικά δίκτυα του άμεσου πραγματικού περιβάλλοντος του ατόμου. Επιπλέον με την χρήση των μέσων κοινωνικής δικτύωσης οι χρήστες μπορούν να μάθουν διαφορετικούς πολιτισμούς και κουλτούρες, να διευρύνουν το πολιτικό τους πνεύμα και να γνωρίζουν τι συμβαίνει σε όλο τον κόσμο. Τέλος μπορούν να χρησιμοποιήσουν τα μέσα κοινωνικής δικτύωσης σαν ψυχαγωγικό περιεχόμενο παρακολουθώντας βίντεο, φωτογραφία και παίζοντας παιχνίδια μέσα από τις εφαρμογές που προσφέρονται. [30]

Πέραν από τα πλεονεκτήματα που υπάρχουν, τα μειονεκτήματα στην χρήση των μέσων κοινωνικής δικτύωσης είναι πολύ πιο περισσότερα, αλλά και σοβαρά γι' αυτό και πρέπει να ληφθούν υπόψιν από κάθε χρήστη. Αρχικά γίνεται παραβίαση ιδιωτικής ζωής αφού σε κάθε βήμα της περιδιάβασής μας στο Διαδίκτυο “προσφέρουμε” προσωπικές πληροφορίες. Αυτές οι πληροφορίες είναι σαν ένας γρίφος που πρέπει να συμπληρωθεί για να αποκαλυφθεί η εικόνα μας. Επιπλέον μπορεί να γίνει υποκλοπή των προσωπικών δεδομένων από την στιγμή που κάθε χρήστης τις παραθέτει στα μέσα κοινωνικής δικτύωσης. Επίσης, οι άνθρωποι που κάνουν πολύωρη χρήση των μέσων κοινωνικής δικτύωσης αποξενώνονται από τον πραγματικό κόσμο. Τέλος, γίνεται παραπλάνηση σε ανυποψίαστους χρήστες τις περισσότερες φορές από διάφορους κακόβουλους χρήστες για να επισκεπτόμαστε διάφορες ιστοσελίδες με αποτέλεσμα να επιφέρουν διάφορους ιούς και κακόβουλες ενεργείες. [30]

### **3.2.2 Facebook, η δημοφιλέστερη πλατφόρμα Social Media**

Το Facebook ανήκει στην κατηγορία των Social Media, ή αλλιώς των Μέσων Κοινωνικής Δικτύωσης, και πιο συγκεκριμένα στην κατηγορία των Κοινωνικών Δικτύων. Αυτή τη στιγμή είναι το δημοφιλέστερο στο είδος του, έχοντας πάνω από 829 εκατ. καθημερινούς ενεργούς χρήστες και πάνω από 2 δισ. μηνιαίους ενεργούς χρήστες, σύμφωνα με τα τελευταία οικονομικά του αποτελέσματα.[8] Η χρήση του Facebook έχει να κάνει κυρίως με τα νέα φίλων και γνωστών, με την καθημερινή επικοινωνία μέσω μηνυμάτων και με την ενημέρωση από επιχειρήσεις και σελίδες άλλων ενδιαφερόντων.[9]

Οι λειτουργίες τις οποίες χρησιμοποιούν περισσότερο οι χρήστες βρίσκονται στις δημοσιεύσεις και είναι το like (μου αρέσει), το share (η κοινοποίηση) και τα comments (σχόλια). Οι ενέργειες αυτές εμφανίζονται στην πρώτη σελίδα του Facebook και με αυτό τον τρόπο ο χρήστης ενημερώνετε πολύ πιο εύκολο για τις δραστηριότητες των φίλων του. Επιπλέον μέσα από το Facebook παρατηρούνται και οι συμπεριφορές και οι χαρακτήρες των χρηστών είτε από την φωτογραφία του προφίλ τους είτε από τις δημοσιεύσεις που κάνουν στον τοίχο τους. Εκτός από ενδιαφέροντα ένα μεγάλο ποσοστό των χρηστών του Facebook έχει έλλειψη ελέγχου για την προστασία της ιδιωτικής ζωής αφού παρουσιάζουν την ταυτότητα τους και πληροφορίες για την ιδιωτική τους ζωή.[10]

### 3.2.3 Διαδικτυακοί Κίνδυνοι

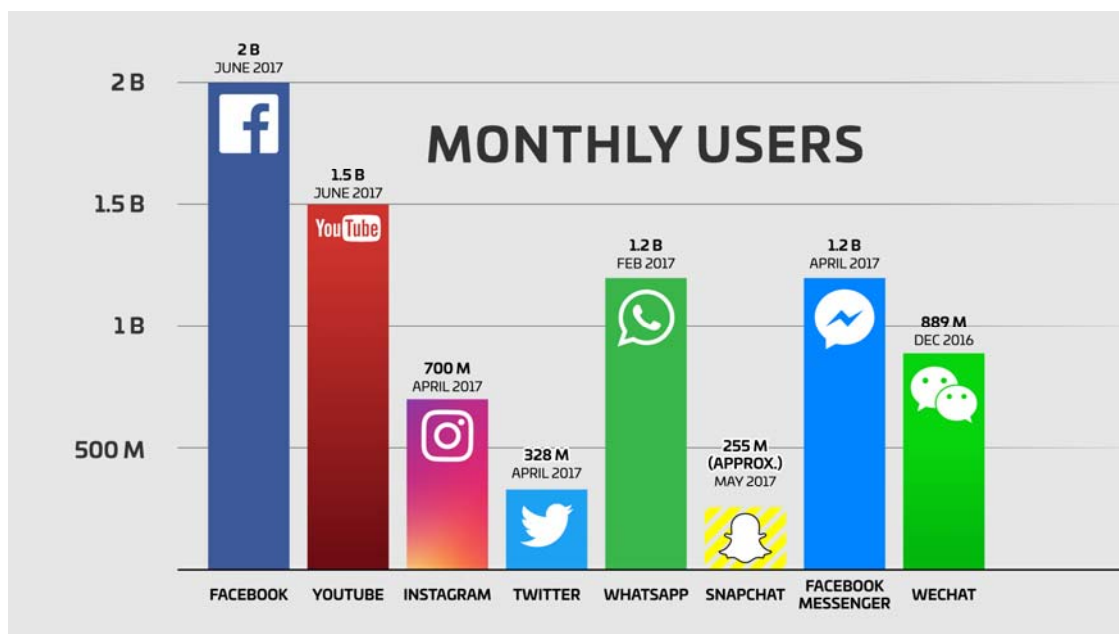
Οι δραστηριότητες όμως μπορεί να επηρεάσουν και τους άλλους χρήστες να προβούν σε ενέργειες που μπορεί να είναι κακόβουλες όπως για παράδειγμα διάφορους ιούς που ελλοχεύουν και πολλαπλασιάζονται ιδιαίτερα στο Facebook. Παρατηρείτε ότι δελεάζουν τον χρήστη να επισκεφθεί μια σελίδα με αφορμή ειδήσεις της επικαιρότητας αλλά και διαγωνισμούς. Επίσης, παρατηρούνται και φαινόμενα πλαστοπροσωπίας με την χρήση της φωτογραφίας διασύροντας τους ανυποψίαστους χρήστες. Το Facebook παρέχει αρκετές πληροφορίες που μπορεί να χρησιμοποιηθούν από κακόβουλους για διάφορους σκοπούς. Ωστόσο όλα τα δεδομένα που αντλούνται από το Facebook προσφέρουν την ανάγκη για επεξεργασία και για ανάλυση. [31]

### 3.2.4 Εθισμός στα Μέσα Κοινωνικής Δικτύωσης (Social Media)

Πρόσφατη έρευνα της Αμερικανικής Ψυχολογικής Εταιρείας για την επίδραση των μέσων κοινωνικής δικτύωσης στα επίπεδα άγχους των Αμερικανών έκρουε τον κώδωνα του κινδύνου για τον εθισμό των χρηστών στα social media. Σε πολύ μεγάλο ποσοστό οι χρήστες βρίσκονται σχεδόν πάντα online, ελέγχοντας συνεχώς τον λογαριασμό τους στο Facebook ή στο Instagram. Μεταξύ των νέων τα ποσοστά των «εμμονικών» χρηστών είναι ακόμη μεγαλύτερα και το χειρότερο είναι ότι τις περισσότερες φορές δεν συνειδητοποιούν ότι το πάθος τους αγγίζει τα όρια της εξάρτησης. Σε άλλη έρευνα του Πανεπιστημίου Κορνέλ ζητήθηκε από τους χρήστες να μείνουν offline για 99 ημέρες και να περιγράψουν τη διάθεσή τους ανά διάστημα 33 ημερών. Η έρευνα κατέδειξε ότι για τους περισσότερους είναι εξαιρετικά δύσκολο να επιβληθούν στον εαυτό τους και να μείνουν εκτός Διαδικτύου.

Υπάρχουν διάφορες εξηγήσεις για τον εθισμό στον θαυμαστό κόσμο των social media. Παγκοσμίως, υπάρχουν περισσότεροι από 2 δισ. μηνιαίοι ενεργοί χρήστες του Facebook, ένας αριθμός που αυξάνεται ετησίως με ρυθμούς πάνω από 10%. Και είναι πολύ σημαντικό να νιώθει κανείς ότι ανήκει σε αυτήν την τεράστια κοινότητα, μια κοινότητα με ενεργό δραστηριότητα 24 ώρες το 24ωρο, που επιτρέπει σε φίλους, κοινούς γνωστούς, αλλά και εχθρούς να αλληλεπιδρούν μεταξύ τους, να ανταλλάσσουν ιδέες και απόψεις, να ενημερώνονται -όχι πάντα από αξιόπιστες πηγές- για τα σημαντικότερα γεγονότα πάσης φύσεως, είτε πρόκειται για ειδήσεις, είτε για τα κουτσομπολιά από τον κόσμο των διασήμων. Μια κοινότητα που προσφέρει λίγες

στιγμές δημοσιότητας μέσα από τα δισεκατομμύρια «likes» στις 136.000 φωτογραφίες που αναρτώνται κάθε λεπτό της ημέρας.



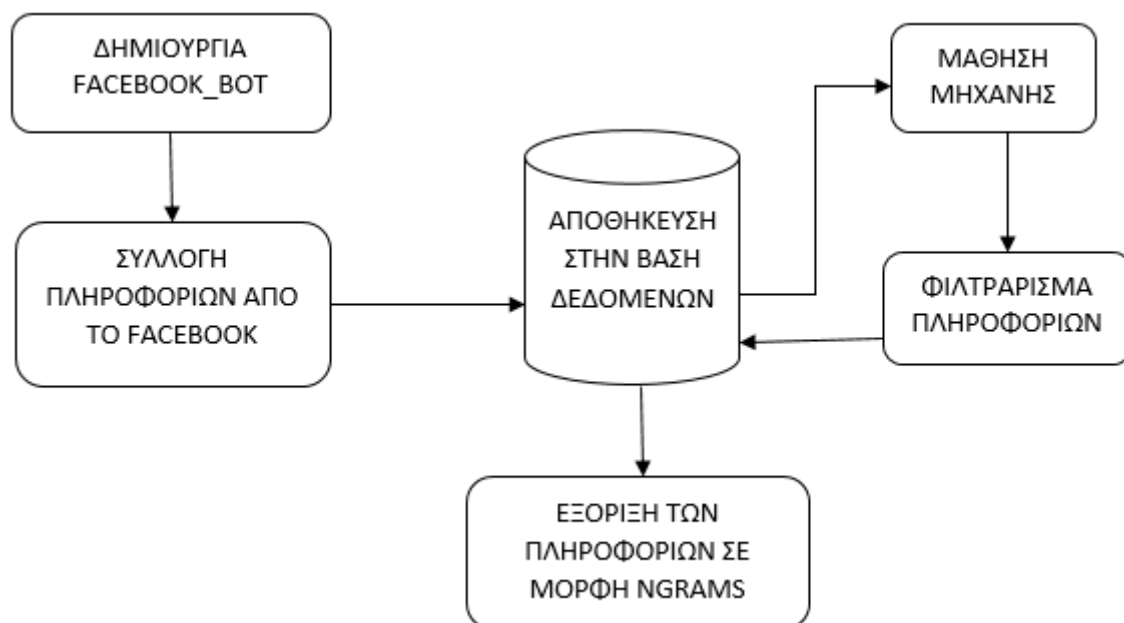
Εικόνα 34: Μηνιαίοι χρήστες στα μέσα κοινωνικής δικτύωσης

Κανείς δεν μπορεί να αμφισβητήσει τη δύναμη του Διαδικτύου, ούτε και το γεγονός ότι αποτελεί την τάση του μέλλοντος. Ούτε μπορεί κανείς να δαιμονοποιήσει ένα τόσο ισχυρό και συνάμα τόσο επιτυχημένο εργαλείο, που δημιουργήθηκε από την ανάγκη για πρόοδο και καλύτερη πληροφόρηση. Αρκεί το εργαλείο να χρησιμοποιείται σωστά και με σύνεση, χωρίς να υποκαθιστά βασικές αξίες των ανθρώπινων σχέσεων. Διότι, σε αντίθετη περίπτωση, απομένει ο εθισμός και τραγικά περιστατικά, όπως η ανάρτηση αποτρόπαιων και σοκαριστικών εικόνων, με πιο πρόσφατο παράδειγμα τη ζωντανή αναμετάδοση του ομαδικού βιασμού της 15χρονης στις ΗΠΑ. [32]

# Κεφάλαιο 4

## FaceWallGrap

Το Κεφάλαιο FacebookWallGrap παρουσιάζει την τεχνική η οποία χρησιμοποιεί ένα web-crawler για το OSINT για την η συλλογή και επεξεργασία των πληροφοριών. Όπως βλέπουμε και στο σχεδιάγραμμα η δημιουργία του Facebook\_Bot με την χρήση της γλώσσας προγραμματισμού Python ήταν το αρχικό στάδιο, όπου πιο κάτω αναφέρονται αναλυτικά οι κλάσεις που δημιουργήθηκαν για την υλοποίηση του FaceWallGrap. Στην συνέχεια οι πληροφορίες έχουν συλλεχθεί από profiles των φίλων που είναι στο λογαριασμό του Facebook και αποθηκεύτηκαν στην Βάση Δεδομένων. Με την χρήση των πληροφοριών από την βάση δεδομένων η μηχανή μάθησης (machine learning) έκανε φιλτράρισμα ορισμένων λέξεων που υπήρχαν μέσα στις πληροφορίες και απόστέλλε στην βάση δεδομένων τις φιλτραρισμένες πληροφορίες. Τέλος οι πληροφορίες αυτές που αφορούσαν την ψυχοσύνθεση των χρηστών έγιναν εξόρυξη σε ngrams σχεδιαγράμματα.



Σχεδιάγραμμα 1: Σχεδιάγραμμα της προτεινόμενης δουλειάς μου

## 4.1 Εισαγωγή στην Python

Η Python είναι μια γλώσσα προγραμματισμού γενικής χρήσης, πάρα πολύ υψηλού επιπέδου, απλή και εύκολη στην εκμάθηση, ισχυρή, δυναμική, αποδοτική, παραγωγική και επεκτάσιμη. Είναι κατάλληλη και για αρχάριους και για έμπειρους προγραμματιστές.

Η Python διαθέτει αποδοτικές δομές δεδομένων υψηλού επιπέδου και υποστηρίζει, χωρίς να αναγκάζει, μια απλή αλλά συνάμα αρκετά αποτελεσματική προσέγγιση στον αντικειμενοστραφή προγραμματισμό. Υποστηρίζει και άλλες γνωστές προγραμματιστικές προσεγγίσεις, όπως είναι ο διαδικαστικός και ο συναρτησιακός προγραμματισμός. Η σύνταξή της είναι κομψή και οι τύποι της δυναμικοί. Είναι διερμηνευόμενη (interpreted) γλώσσα προγραμματισμού και μπορεί να χρησιμοποιηθεί τόσο για τη δημιουργία σεναρίων εντολών όσο και για τη γρήγορη ανάπτυξη ολοκληρωμένων εφαρμογών σε διάφορες περιοχές ενδιαφέροντος (διαχείριση συστήματος υπολογιστή, ανάπτυξη εφαρμογών Διαδικτύου, επεξεργασία αρχείων κειμένου, επιστημονικές εφαρμογές, εκπαίδευση, ανάπτυξη παιχνιδιών, κ.λπ.) και στις περισσότερες πλατφόρμες υλικού υπολογιστών και Λειτουργικών Συστημάτων (Windows, Unix, Linux, Mac OS X, κ.λπ.). Διαθέτει πληθώρα έτοιμων βιβλιοθηκών που μπορούν να χρησιμοποιηθούν εύκολα και άμεσα. Τα προγράμματα σε Python είναι συμπαγή, ευανάγνωστα, και γράφονται και συντηρούνται γρηγορότερα σε σχέση με

άλλες δημοφιλείς γλώσσες προγραμματισμού όπως οι C, C++ και Java. Ο κώδικας μπορεί να ομαδοποιηθεί σε αρθρώματα (modules) και πακέτα (packages). Αναπτύσσεται ως λογισμικό ανοικτού κώδικα με βάση το μοντέλο της κοινότητας προγραμματιστών που εργάζονται για την ανάπτυξή της, και ο συντονισμός και η διαχείριση γίνεται από τον μη κερδοσκοπικό οργανισμό Python Software Foundation (PSF) ([www.python.org/psf](http://www.python.org/psf)). Ως αποστολή του PSF αναφέρεται η ανάδειξη, η προστασία και η προαγωγή της Python, καθώς και η υποστήριξη της ανάπτυξης μιας ποικίλης και διεθνούς κλίμακας κοινότητας προγραμματιστών της Python. Η κοινότητα αυτή αυξάνεται καθημερινά και η Python βρίσκεται σε υψηλή θέση στη λίστα των δημοφιλέστερων γλωσσών προγραμματισμού. Επίσης, χρησιμοποιείται από μεγάλες εταιρείες και οργανισμούς του χώρου της Πληροφορικής και όχι μόνο. Πέρα όμως από την πληθώρα των θετικών χαρακτηριστικών της Python, οφείλουμε να αναφέρουμε και μειονεκτήματα. Ο χρόνος εκτέλεσης των προγραμμάτων της Python μπορεί να μην είναι πάντα τόσο γρήγορος όσο είναι στις γλώσσες προγραμματισμού C και η C++. Αυτό οφείλεται στο ότι ένα πρόγραμμα σε Python δεν μεταγλωττίζεται σε δυαδικό κώδικα μηχανής που εκτελείται άμεσα από τον επεξεργαστή του υπολογιστή. Το μειονέκτημα αυτό αντισταθμίζεται από το ότι πολλές φορές είναι σημαντικότερη η εξοικονόμηση χρόνου που έχουμε κατά την ανάπτυξη μιας εφαρμογής σε Python. [11]



Εικόνα 5: Λογότυπο Python

#### **4.1.1 Πρόγραμμα που χρησιμοποιήθηκε για την υλοποίηση του κώδικα**

Για να υλοποιηθεί ο κώδικας για τις ανάγκες της διατριβής χρησιμοποιήσα το λογισμικό Spyder. Το Spyder είναι το Επιστημονικό Περιβάλλον Ανάπτυξης της γλώσσας Python όπου έχει ένα ισχυρό περιβάλλον διαδραστικής ανάπτυξης για τη γλώσσα Python με προηγμένες δυνατότητες επεξεργασίας, διαδραστικές δοκιμές, εντοπισμό σφαλμάτων

και χαρακτηριστικά ενδοσκόπησης. Επιπλέον έχει και ένα αριθμητικό περιβάλλον υπολογιστών χάρη στην υποστήριξη του IPython (ενισχυμένη αλληλεπιδραστική ερμηνεία Python) και δημοφιλείς βιβλιοθήκες Python όπως NumPy (γραμμική άλγεβρα), SciPy (επεξεργασία σήματος και εικόνας) ή matplotlib (διαδραστική γραφική παράσταση 2D / 3D). [12]



Εικόνα 6: Λογότυπο λογισμικού Spyder για Python

## 4.2 Βασικές λειτουργίες Python Facebook Crawler

Για να υλοποιήσουμε το Facebook Crawler χρησιμοποιούμε την βιβλιοθήκη Selenium στην κλάση FacebookCrawler.py. Με αυτές τις τεχνικές θα μπορούμε να τραβήξουμε αυτόματα τις πληροφορίες που χρειαζόμαστε για τον κάθε χρήστη και να τις αποθηκεύσουμε στην βάση δεδομένων έτσι ώστε να μπορούμε να γίνουν επεξεργασία σε μετέπειτα στάδιο.

### 4.2.1 Βιβλιοθήκη Python Selenium

Το Selenium WebDriver είναι ένα από τα πιο δημοφιλή εργαλεία για το Web UI Automation. Η γλώσσα προγραμματισμού Python μπορεί να το συμπληρώσει για να αυτοματοποιήσει ένα ευρύ φάσμα εφαρμογών web. Ως εκ τούτου, υπάρχει και το Selenium Webdriver Python για να γίνει πιο εύκολα το έργο. Ξεκινώντας με την κατανόηση του τι πρέπει να κάνουμε μόνο για τον έλεγχο αυτοματισμού UI. [13]

Με το Web UI Automation εννοούμε την αυτόματη εκτέλεση των ενεργειών που εκτελούνται σε ένα παράθυρο προγράμματος περιήγησης ιστού όπως την πλοήγηση σε



έναν ιστότοπο, την συμπλήρωση φόρμας που περιλαμβάνει την επεξεργασία κουτιών κειμένου, των ραδιοφωνικών κουμπιών και του drop down, την υποβολή των εντύπων, την περιήγηση σε ιστοσελίδες, το χειρισμό αναδυόμενων παραθύρων (pop-up windows). Το Selenium WebDriver είναι αυτό που μπορεί να αυτοματοποιήσει όλες αυτές τις εργασίες. Μπορεί να αλληλεπιδράσει με όλους τους τύπους προγραμμάτων περιήγησης Web που είναι διαθέσιμα όπως το Firefox, ο Internet Explorer, το Safari και το Chrome.

Τις περισσότερες φορές, χρησιμοποιείτε Java και Eclipse για να δημιουργήσουμε ένα έργο Selenium Webdriver. Σε αυτήν την ανάρτηση, θα χρησιμοποιήσουμε την Python για να αντικαταστήσουμε την Java και θα δείξουμε πώς λειτουργεί η ενοποίηση της Selenium Webdriver Python για την επίτευξη της αυτοματοποίησης του Web UI.

Το Selenium είναι μια ανοιχτή πηγή και η βιβλιοθήκη του διατίθεται σε διαφορετικές γλώσσες προγραμματισμού για να εκτελέσει τον έλεγχο Web UI Automation και η Python είναι μία από αυτές.

Η Βιβλιοθήκη πελατών Selenium WebDriver για Python μας επιτρέπει να αξιοποιήσουμε όλες τις λειτουργίες που προσφέρετε με το Selenium WebDriver και να αλληλοεπιδρά με το Selenium Standalone Server για να εκτελείτε αυτοματοποιημένες δοκιμές (τόσο απομακρυσμένες όσο και κατανεμημένες δοκιμές) των εφαρμογών που βασίζονται σε προγράμματα περιήγησης. [14]

```

5 from selenium.webdriver.common.by import By
6 from selenium.webdriver.support.ui import WebDriverWait
7 from selenium.webdriver.support import expected_conditions as EC
8 from selenium.common.exceptions import NoSuchElementException
9 import time
10 from selenium.webdriver.common.keys import Keys
11 from datetime import datetime
12
13 class FacebookCrawler:
14     def __init__(self,driver):
15         self.url = 'https://www.facebook.com'
16         self.email = 'emilia.panayiotou@hotmail.com'
17         self.password = 'Emily1234!'
18         self.driver = driver
19
20     def login(self,email,password):
21         wait = WebDriverWait(self.driver, 10)
22         wait.until(EC.presence_of_element_located((By.ID,'email')))
23         username = self.driver.find_element_by_id('email')
24         username.send_keys(email)
25         wait.until(EC.presence_of_element_located((By.ID,'pass')))
26         passwd = self.driver.find_element_by_id('pass')
27         passwd.send_keys(password,Keys.ENTER)
28
29     def getFriendList(self):
30
31         wait = WebDriverWait(self.driver, 10)
32         wait.until(EC.presence_of_element_located((By.CLASS_NAME,'_Safe')))
33         profile = self.driver.find_element_by_class_name("_Safe")
34         profile.click()
35
36         url = self.driver.current_url.split('?')[0]
37         url = url + '/friends'

```

Εικόνα 7: Απόσπασμα κώδικα για το Selenium

## 4.2.2 Κλάση FacebookCrawler.py

Η κλάση FacebookCrawler.py χρησιμοποιεί την βιβλιοθήκη Selenium για γίνετε αυτόματη σύνδεση στο Facebook με τον πιο κάτω κώδικα και με την χρήση του προγράμματος περιήγησης Chrome για να ξεκινά η συλλογή πληροφοριών για τον κάθε χρήστη.

```

class FacebookCrawler:
    def __init__(self, driver):
        self.url = 'https://www.facebook.com'
        self.email = 'emilia.panayiotou@hotmail.com'
        self.password = 'Emily1234!'
        self.driver = driver

    def login(self, email, password):
        wait = WebDriverWait(self.driver, 10)
        wait.until(EC.presence_of_element_located((By.ID, 'email'))))
        username = self.driver.find_element_by_id('email')
        username.send_keys(email)
        wait.until(EC.presence_of_element_located((By.ID, 'pass'))))
        passwd = self.driver.find_element_by_id('pass')
        passwd.send_keys(password, Keys.ENTER)

```

Εικόνα 8: FacebookCrawel code for automatic Login in

Η συλλογή των πληροφοριών για τους χρήστες γίνεται μέσω της κλάσης facebook\_bot.py καλώντας την κλάση FacebookCrawler.py. Όταν τρέξει ο κώδικας αρχίζει να συλλέγει τις πληροφορίες για τους φίλους που υπάρχουν στο Facebook. Ο κώδικας είναι προσαρμοσμένος έτσι ώστε να συλλέγει όλες τις πληροφορίες από την μέρα που οι χρήστες έχουν δημιουργήσει το Facebook τους. Η συλλογή είναι χρονοβόρα γι' αυτό χρειαζόμαστε ένα καλό δίκτυο και χρόνο για να γίνει μια ολοκληρωμένη συλλογή πληροφοριών.

Επιπλέον χρησιμοποιούνται τα πιο κάτω απαραίτητα functions για σκοπούς web scrapping:

1. Το function **init\_driver** όπου είναι ένα απαραίτητο driver για να:
  - δημιουργεί την παρουσία του driver του προγράμματος
  - προσθέτει τη λειτουργία **WebDriverWait** ως χαρακτηριστικό γνώρισμα του driver του προγράμματος, ώστε να είναι πιο εύκολη η πρόσβαση. Αυτή η λειτουργία χρησιμοποιείται για να περιμένει ένα συγκεκριμένο χρονικό διάστημα (εδώ 5 δευτερόλεπτα) για να συμβεί κάποιο συμβάν όπως για παράδειγμα απώλεια της σύνδεσης του δικτύου. Εάν συμβεί αυτό τότε προχωρά παρακάτω.
2. Το function **lookup** όπου απαιτεί δύο επιχειρήματα: μια παράσταση driver του προγράμματος και μια αναζήτηση ερωτήματος (μια συμβολοσειρά).
  - φορτώνει τη σελίδα του Facebook
  - περιμένει να εντοπιστεί το profil του ερωτήματος και να είναι clickable για το κουμπί. Σημειώστε ότι χρησιμοποιεί τη λειτουργία **WebDriverWait** για να περιμένουμε να εμφανιστούν αυτά τα στοιχεία.
  - Και τα δύο στοιχεία βρίσκονται με το όνομα.
  - Στη συνέχεια, στέλνει το ερώτημα στο profile και κάνει κλικ στο κουμπί αναζήτησης.
  - Αν το profil ή το κουμπί δεν βρίσκονται κατά τη διάρκεια της ώρας που έχει καθοριστεί στη λειτουργία αναμονής (εδώ, 5 δευτερόλεπτα), η επιλογή **TimeoutException** αυξάνεται. [14]

### 4.2.3 Κλάση Facebookbot.py

Η κλάση Facebookbot.py κάνει import όλα τα απαραίτητα στοιχεία από το FacebookCrawler.py έτσι ώστε να δημιουργήσει και να αποθηκεύσει τα στοιχεία στην Βάση δεδομένων. Έχει ως βασικό χαρακτηριστικό να χρησιμοποιά τον browser google chrome έτσι ώστε να είναι πιο εύκολο στον crawler να τραβήξει τις πληροφορίες. Τέλος η κλάση Facebookbot.py προσδιορίζει πως θα εμφανίζονται οι διαδικασίες των functions στην κονσόλα του προγράμματος έτσι ώστε να είναι πιο κατανοητές οι ενέργειες που γίνονται για όσους θα δουν το πρόγραμμα να τρέχει.

### 4.3 Αποθήκευση των πληροφοριών

Παράλληλα με το πρόγραμμα τις Python Spyder θα πρέπει να είναι ανοικτό και το πρόγραμμα για την αποθήκευση των πληροφοριών το MySQL Workbench 6.3 CE. Οι πληροφορίες που θα συλλέγονται, θα αποθηκεύονται στην βάση δεδομένων και αυτόματα θα γίνεται μια κατηγοριοποίηση έτσι ώστε να έχουμε μια πιο ολοκληρωμένη εικόνα για το τι συλλέγεται από τους χρήστες. Το λογισμικό MySQL Workbench 6.3 CE παρέχει ένα ολοκληρωμένο περιβάλλον εργαλείων για την Σχεδίαση και μοντελοποίηση βάσεων δεδομένων, την ανάπτυξη SQL και την διαχείριση βάσεων δεδομένων.



Εικόνα 9: Λογότυπο λογισμικού MySQL Workbench 6.3 CE

### 4.3.1 Κλάση Database.py

Με την κλάση Database περιορίζομαι τι πληροφορίες χρειαζόμαστε και τι προσθέσουμε στους πίνακες που δημιουργήθηκαν στο πρόγραμμα για την βάση δεδομένων.

Οι πληροφορίες που περιορίζονται μέσα από την κλάση Database είναι οι πιο κάτω:

- Δημοσιεύσεις των χρηστών (posts)
- Φωτογραφίες
- Τόπος που έγινε η κάθε δημοσίευση (εάν και εφόσον έγινε κοινοποίηση από τον χρήστη)
- Όνομα που έχει ο χρήστης στο Facebook (ελληνικά ή Αγγλικά γραμμένο)
- Ημερομηνίες και ώρες για την κάθε δημοσίευση

Όταν εισαχθούν στην βάση δεδομένων τότε ο κάθε χρήστης παίρνει ένα μοναδικό id όπου αυτό το id θα μας δείχνει πια δημοσίευση έχει γίνει.

```
1 # -*- coding: utf-8 -*-
2 """
3 Created on Wed Aug  9 12:52:29 2017
4
5 """
6 import pymysql
7
8 class Database:
9     def __init__(self):
10         self.host = 'localhost'
11         self.user = 'root'
12         self.password = 'root123'
13         self.db = 'facebookcrawler'
14
15     def insertFriend(self,i,name,gender,url):
16         query = "Insert into friends(id,name,gender,url) values(%s,%s,%s,%s)"
17         args = (i,name,gender,url)
18         connection = pymysql.connect(self.host, self.user, self.password, self.db, use_unicode =
19         cursor = connection.cursor()
20
21         try:
22             cursor.execute(query,args)
23             connection.commit()
24         except:
25             print("Error inserting friend to database")
26             connection.rollback()
27         finally:
28             connection.close()
29             cursor.close()
30
31     def deleteAllFriends(self):
32         query = "Delete from friends"
33         connection = pymysql.connect(self.host, self.user, self.password, self.db, use_unicode =
34         cursor = connection.cursor()
```

Εικόνα 10: Απόσπασμα κώδικα κλάσης Database

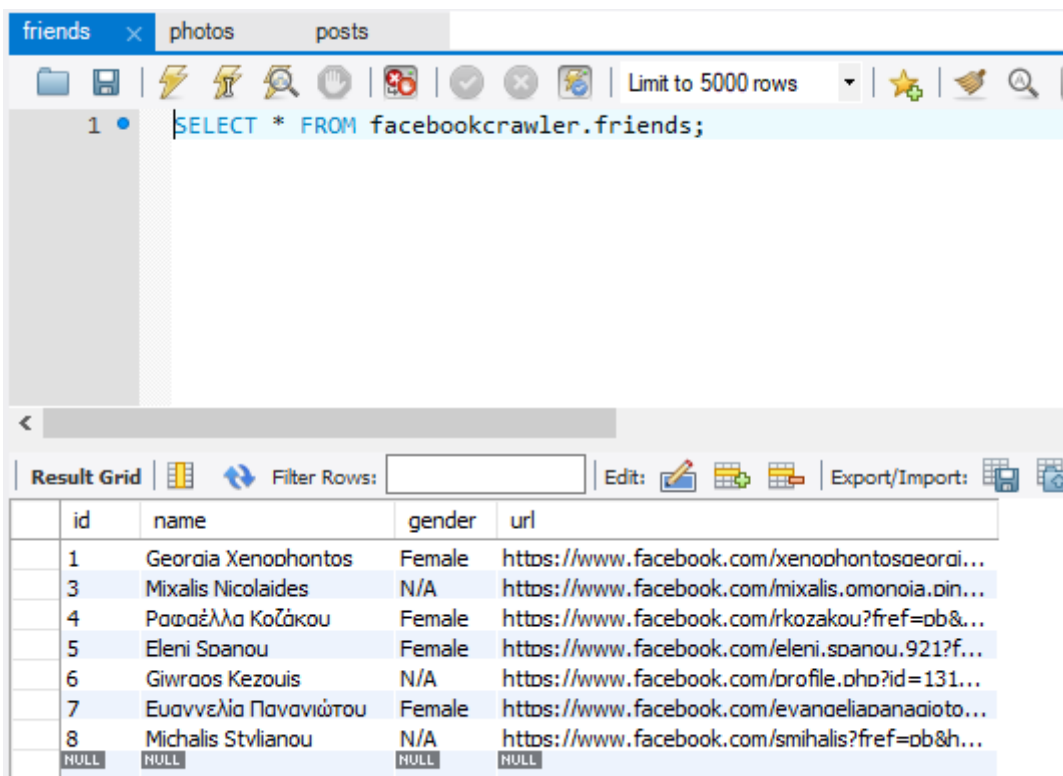
## 4.3.2 Ανάλυση Αποθήκευσης στην Βάση Δεδομένων

Οι πίνακες που έχουν δημιουργηθεί στην βάση δεδομένων είναι για Φίλους, δημοσιεύσεις και φωτογραφίες. Ο κάθε πίνακας έχει τα δικά του χαρακτηριστικά αλλά μεταξύ τους συνδέονται.

### 4.3.2.1 Φίλοι

Ο πίνακας με το όνομα “Friends” αποθηκεύει τα πιο κάτω στοιχεία:

1. **ID:** είναι τύπου integer και δημιουργείται αυτόματα το id του χρήστη και δεν μπορεί ποτέ το πεδίο του συγκεκριμένου πίνακα να είναι κενό.
2. **Name:** περιλαμβάνει το ονοματεπώνυμο του χρήστη όπως είναι γραμμένο στο Facebook με οποιαδήποτε γλώσσα χρησιμοποιεί ο χρήστης.
3. **Gender:** περιλαμβάνει το φίλο του κάθε χρήστη είτε το καθορίζει στο προφίλ του είτε όχι.
4. **URL:** παίρνει το url που δίνεται στο κάθε χρήστη του Facebook.



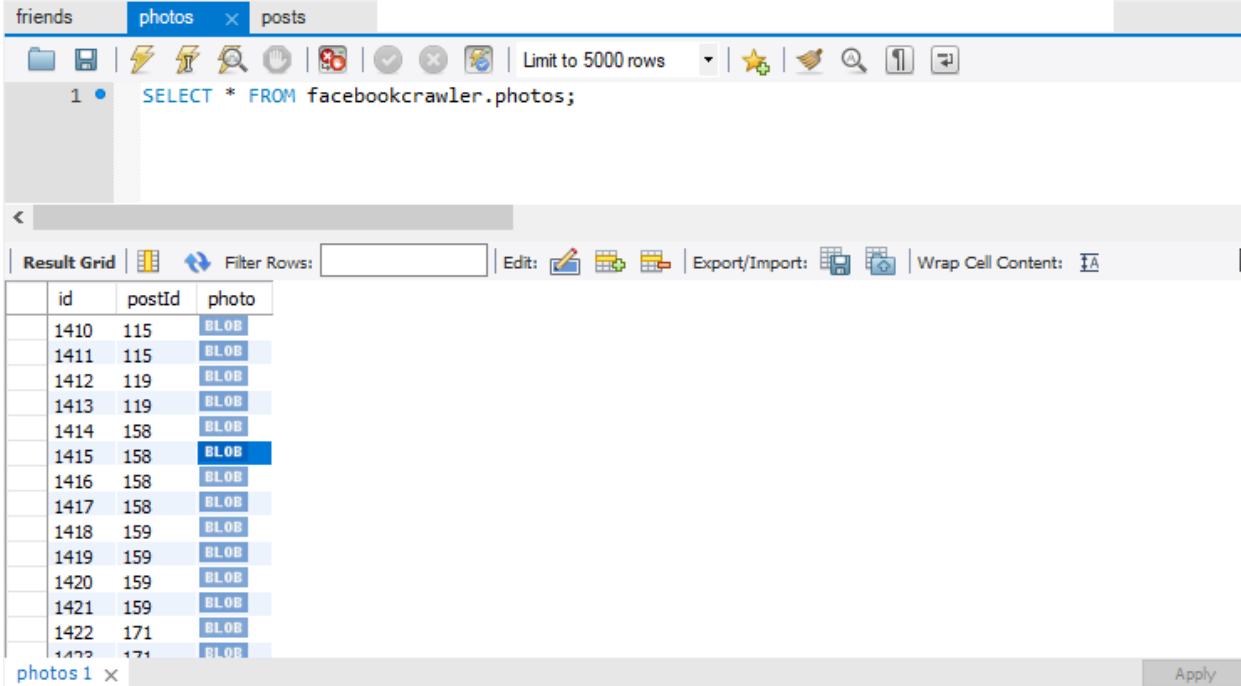
	id	name	gender	url
1	1	Georai Xenophontos	Female	https://www.facebook.com/xenophontosaeorai...
3	3	Mixalis Nicolaidis	N/A	https://www.facebook.com/mixalis.omonioia.pn...
4	4	Ραφαέλλα Κοζάκου	Female	https://www.facebook.com/rkozakou?fref=pb&...
5	5	Eleni Spanou	Female	https://www.facebook.com/eleni.spanou.921?f...
6	6	Giwraos Kezouis	N/A	https://www.facebook.com/profile.php?id=131...
7	7	Ευαννελία Πανανιώτου	Female	https://www.facebook.com/evangeliaapanaioto...
8	8	Michalis Stvlianou	N/A	https://www.facebook.com/smihalis?fref=pb&h...
	NULL	NULL	NULL	NULL

Εικόνα 11: Πίνακας friends από την βάση δεδομένων

### 4.3.2.2 Φωτογραφίες

Ο πίνακας με το όνομα “Photos” αποθηκεύει τα πιο κάτω στοιχεία:

1. **ID:** είναι τύπου integer και δημιουργείται αυτόματα το id του χρήστη και δεν μπορεί ποτέ το πεδίο του συγκεκριμένου πίνακα να είναι κενό.
2. **PostID:** είναι ένα foreign key στον πίνακα και χρησιμοποιείται για την ένωση του πίνακα με τον πίνακα posts.
3. Οι φωτογραφίες γίνονται συλλογή με τον τύπο BLOG στην βάση δεδομένων και όταν γίνει export σε κάποια σελίδα επαναφέρει τον τύπο της φωτογραφίας ξανά.
4. Το ID που έχει κάθε φωτογραφία αντιστοιχεί στο ID που έχει το post στο οποίο αναρτήθηκε η φωτογραφία. Για παράδειγμα ID:1410 and ID:1411 = post ID: 115, όπου ID είναι του πίνακα Photos και post ID του πίνακα Posts όπου το Post 115 έχει τις φωτογραφίες 1410 και 1411 (Εικόνα 12).



id	postId	photo
1410	115	BLOB
1411	115	BLOB
1412	119	BLOB
1413	119	BLOB
1414	158	BLOB
1415	158	BLOB
1416	158	BLOB
1417	158	BLOB
1418	159	BLOB
1419	159	BLOB
1420	159	BLOB
1421	159	BLOB
1422	171	BLOB
1423	171	BLOB

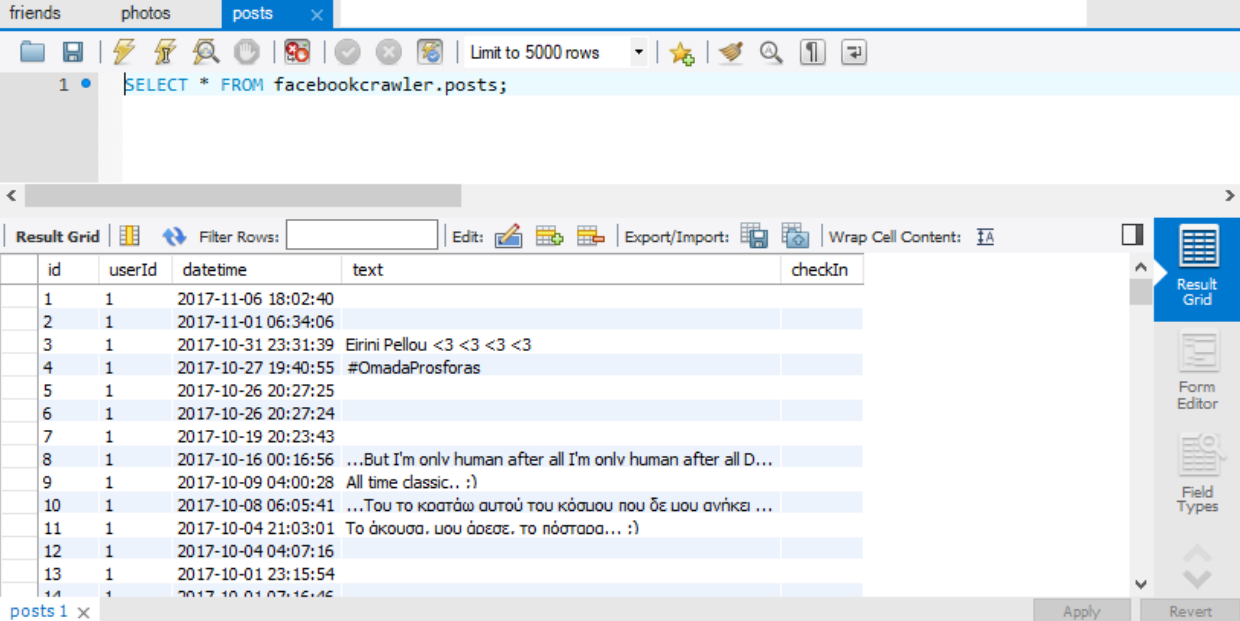
Εικόνα 12: Πίνακας photos από την βάση δεδομένων

### 4.3.2.3 Αναρτήσεις φίλων στα προφίλ τους (posts)

Ο πίνακας με το όνομα “Posts” αποθηκεύει τα πιο κάτω στοιχεία:

1. **ID:** αυξάνει το primary key κάθε φορά που συλλέγει νέο post και η σχέση του μεταξύ του πίνακα για τους φίλους είναι one-to-many δηλαδή ένας χρήστης μπορεί να κάνει πολλά posts.
2. **Date/Time:** Μέρα και ώρα που έχει το post

3. **Text:** το κείμενο που υπάρχει στο post
4. **Check in:** εάν ο χρήστης έχει συμπεριλάβει κάποια τοποθεσία που έκανε το post τότε την αποθηκεύει σε αυτή την στήλη.



The screenshot shows a database query tool interface. At the top, there are tabs for 'friends', 'photos', and 'posts'. Below the tabs is a toolbar with various icons and a 'Limit to 5000 rows' dropdown. The main area contains a SQL query: `SELECT * FROM facebookcrawler.posts;`. Below the query is a 'Result Grid' showing a table with columns: id, userId, datetime, text, and checkIn. The table contains 14 rows of data.

id	userId	datetime	text	checkIn
1	1	2017-11-06 18:02:40		
2	1	2017-11-01 06:34:06		
3	1	2017-10-31 23:31:39	Eirini Pellou <3 <3 <3 <3	
4	1	2017-10-27 19:40:55	#OmadaProsforas	
5	1	2017-10-26 20:27:25		
6	1	2017-10-26 20:27:24		
7	1	2017-10-19 20:23:43		
8	1	2017-10-16 00:16:56	...But I'm only human after all I'm only human after all D...	
9	1	2017-10-09 04:00:28	All time classic.. :)	
10	1	2017-10-08 06:05:41	...Του το κρατάω αυτού του κόσμου που δε μου ανήκει ...	
11	1	2017-10-04 21:03:01	Το άκουσα. μου άρεσε. το πόστασα... :)	
12	1	2017-10-04 04:07:16		
13	1	2017-10-01 23:15:54		
14	1	2017-10-01 07:16:46		

Εικόνα 13: Πίνακας posts από την βάση δεδομένων

## 4.4 Βιβλιοθήκη Python ngrams

### 4.4.1 Κλάση NGrams

Η κλάση NGrams έχει στατικές μεθόδους για να συγκρίνει ένα ζεύγος συμβολοσειρών. Τα N-grams δεν βασίζονται σε λέξεις και δεν εφαρμόζουν ένα γλωσσικό μοντέλο, απλώς αναζητούν μέλη με ομοιότητα συμβολοσειρών. Η τεκμηρίωση, οι οδηγίες και οι σημειώσεις έκδοσης βρίσκονται στην τοποθεσία τεκμηρίωσης του πακέτου PyPI. [16]



```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Wed Dec 02 11:06:59 2017
4
5 """
6 from Database import Database
7 import ngram
8 from datetime import datetime
9 import numpy as np
10 import matplotlib.pyplot as plt
11
12 db = Database()
13 posts = db.selectPosts()
14
15 dates = []
16 happy,sad,love,disappointed = [],[],[],[]
17 happyAvg,sadAvg,loveAvg,disappointedAvg = [],[],[],[]
18 for post in posts:
19     G = ngram.NGram(post[3].split())
20     dates.append(post[2].timestamp())
21
22     #happy keywords
23     happy1 = G.search('happy')
24     happy2 = G.search(':)')
25     if happy1 == []:
26         happy.append(0)
27     else:
28         for x in happy1:
29             happy.append(x[1])
30     if happy2 == []:
31         happy.append(0)
32     else:
33         for x in happy2:
34             happy.append(x[1])
35     avg = np.average(happy)
36     happyAvg.append(avg)
37     happy = []

```

Εικόνα 14: Απόσπασμα κώδικα κλάσης Ngrams

#### 4.4.2 Πώς λειτουργεί

Το set του ngram στην Python αποθηκεύει αυθαίρετα στοιχεία, αλλά για τα στοιχεία που δεν είναι στοιχειοσειρά, πρέπει να οριστεί μια λειτουργία κλειδιού (όπως το str) για την παροχή συμβολοσειράς. Η λειτουργία κλειδιού μπορεί επίσης να χρησιμοποιηθεί για την ομαλοποίηση στοιχείων πριν από την ευρετηρίαση N-gram.

Για την ευρετηρίαση μιας συμβολοσειράς, τοποθετεί μια σειρά με σύμβολα με ένα συγκεκριμένο ανιχνευτικό χαρακτήρα, και στη συνέχεια χωρίζει σε υποκείμενες αλληλουχίες με N (προεπιλεγμένες N = 3) χαρακτήρες και συσχετίζει κάθε N-gram με τα στοιχεία που το χρησιμοποιούν.

Για να βρει αντικείμενα παρόμοια με μια συμβολοσειρά ερωτήματος, χωρίζει το ερώτημα σε N-γραμμάκια, συλλέγει όλα τα αντικείμενα που μοιράζονται τουλάχιστον ένα N-gram με το ερώτημα και κατατάσσει τα στοιχεία με βάση την αναλογία των κοινών σε μη επιμερισμένα N-γραμμάκια μεταξύ χορδές. [15]

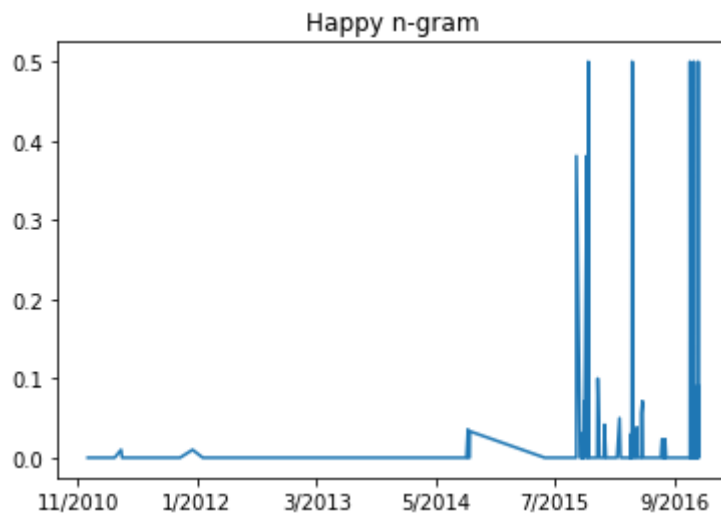
### **4.4.3 Ιστορία**

Το 2007, ο Michel Albert (exhuma) έγραψε την υπομονάδα python-ngram βασισμένη στην υπομονάδα Perl's String :: Trigram από τον Tarek Ahmed και δέσμευσε τον κώδικα για το 2.0.0b2 σε μια τώρα μη χρησιμοποιούμενη αναίρεσης του Sourceforge .

Από τα τέλη του 2008, Graham Poulter έχει διατηρήσει python-ngram, αρχικά refactoring να αξιοποιήσει το σύνολο της κατηγορίας, αλλά και προσθέτοντας χαρακτηριστικά, τεκμηρίωσης, δοκιμές, βελτιώσεις στην απόδοση και Python 3 υποστήριξη. [15]

### **4.4.4 Πως θα αξιοποιηθεί**

Τα NGrams θα χρησιμοποιηθούν για να δείξουμε τα αποτελέσματα από τα συναισθήματα που εκφράζουν οι χρήστες του Facebook. Σε επόμενο κεφάλαιο θα δούμε τι αποτελέσματα έχουμε με την χρήση αυτής της τεχνικής (Εικόνα 14).



Εικόνα 14: Παράδειγμα NGrams από το πρόγραμμα μου

## 4.5 Εξόρυξη των πληροφοριών σε HTML Profiling για κάθε χρήστη

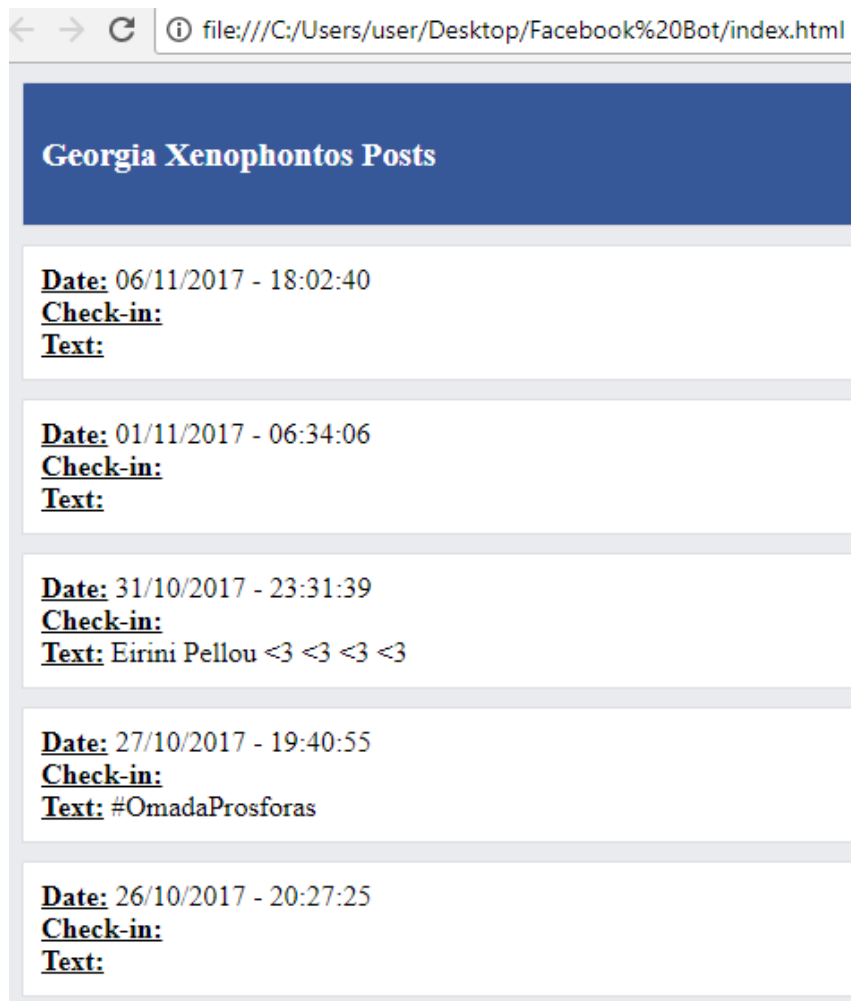
Η δημιουργία του HTML αρχείου προέρχεται από την κλάση `post_view.py`. Τρέχοντας την κλάση αυτή μας εμφανίζεται στην κονσόλα να επιλέξουμε ένα αριθμό από το 1 ως το 8 που αντιστοιχούν στα ονόματα των φίλων. Επιλέγοντας τον αριθμό, ανοίγουμε τον browser μας που έχει μετονομαστεί σε `index` και μας εμφανίζει το προφίλ του χρήστη που επιλέξαμε σε αρχείο HTML (Εικόνα 15 και Εικόνα 16).

```
IPython console
Console 1/A
Please select a person from the list to show his/her posts
1 - Georgia Xenophontos
2 - Mixalis Nicolaidis
3 - Ραφαέλλα Κοζάκου
4 - Eleni Spanou
5 - Giwrgos Kezouis
6 - Ευαγγελία Παναγιώτου
7 - Michalis Stylianos

Enter number: 1
Writing (1/16)
Writing (2/16)
Writing (3/16)
Writing (4/16)
Writing (5/16)
Writing (6/16)
Writing (7/16)
Writing (8/16)
Writing (9/16)
Writing (10/16)
Writing (11/16)
Writing (12/16)
Writing (13/16)
Writing (14/16)
Writing (15/16)
Writing (16/16)
Done! Open index.html file.

In [3]: |
```

Εικόνα 15: Η κονσόλα όταν τρέχουμε την κλάση `view_posts` για να επιλέξουμε τον χρήστη



Εικόνα 16: Το προφίλ του χρήστη σε html αρχείο βάση των πληροφοριών που σύλλεξε το πρόγραμμα

# Κεφάλαιο 5

## Ανάλυση Αποτελεσμάτων

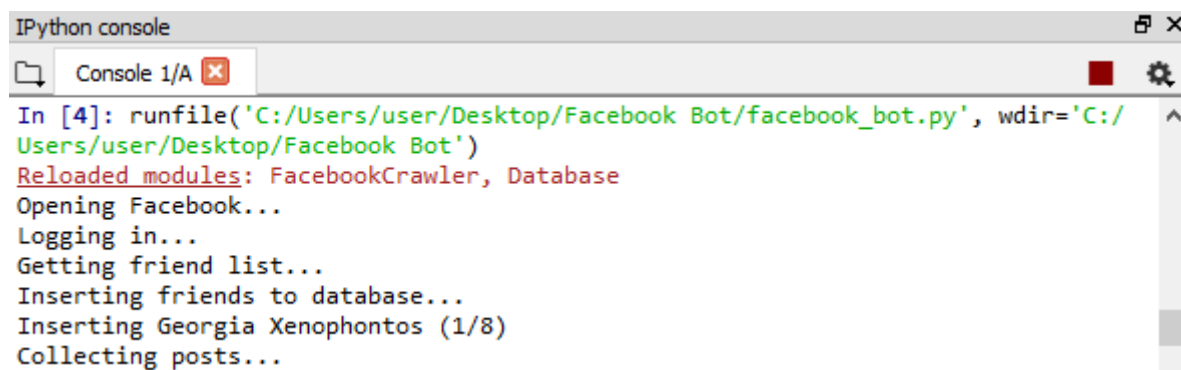
Στο παρόν κεφάλαιο θα γίνει μια αναλυτική περιγραφή για τα αποτελέσματα του θέματος. Στα προηγούμενα κεφάλαια έγινε μια αναφορά στα αποτελέσματα και τα χρησιμοποιήσα σαν παραδείγματα για τους σκοπούς της κάθε ενότητας. Πιο κάτω θα αναλύσω τα αποτελέσματα που είχα από την κλάση `Facebook_bot.py`, την κλάση `view_posts.py` και την κλάση `ngrams.py`.

### 5.1 Αποτελέσματα κλάσης `Facebook_bot.py`

Η κλάση `Facebook_bot.py` καλεί τις κλάσεις `FacebookCrawler.py` και `Database.py` για να μπορεί να ξεκινήσει να συλλέγει και να αποθηκεύει τις πληροφορίες για τον κάθε χρήστη.

Αρχικά, έτρεξα την κλάση `Facebook_bot.py` και ξεκίνησε να κάνει login στο Facebook και να συλλέγει δεδομένα. Ξεκινώντας από τον πρώτο χρήστη κάνει Inserting τα στοιχεία

του χρήστη στον πίνακα “friends” στην βάση δεδομένων. Στο παρακάτω παράδειγμα (Εικόνα 22) βλέπουμε ότι ξεκινά από τον πρώτο, πάντα με τυχαία σειρά μέχρι να φτάσει στον τελευταίο από τους 8 φίλους που υπάρχουν στον λογαριασμό μου.



```
IPython console
Console 1/A
In [4]: runfile('C:/Users/user/Desktop/Facebook Bot/facebook_bot.py', wdir='C:/Users/user/Desktop/Facebook Bot')
Reloaded modules: FacebookCrawler, Database
Opening Facebook...
Logging in...
Getting friend list...
Inserting friends to database...
Inserting Georgia Xenophontos (1/8)
Collecting posts...
```

Εικόνα 22: Παράδειγμα από τρέξιμο της κλάσης Facebook\_bot.py

Στην συνέχεια όταν κάνει collecting τα posts συμπεριλαμβανομένων και των προσωπικών στοιχείων του κάθε χρήστη (φίλο, όνομα, facebook url) αλλά και των φωτογραφιών που έχει κάνει ο κάθε φίλος. Όταν τελειώσει τα κάνει inserting στην βάση δεδομένων έτσι ώστε να φαίνονται στους πίνακες (Εικόνες 23,24 και 25).

Όπως βλέπουμε στην εικόνα 23 ο πίνακας έχει γεμίσει με τα στοιχεία των 8 φίλων που υπάρχουν στον λογαριασμό του Facebook μου. Έχει συλλέξει τις πληροφορίες όνομα (όπως ακριβώς αναγράφεται στον λογαριασμό του χρήστη στο Facebook), το φίλο του (εάν είναι άντρας ή γυναίκα ή δεν έχει δηλώσει κάποιο φίλο) και το url που έχει δημιουργηθεί αυτόματα από το Facebook για να μπορεί ο χρήστης να έχει ένα μοναδικό προσωπικό ιστότοπο.

id	name	gender	url
1	Georgia Xenophontos	Female	https://www.facebook.com/xenophontosaeordi...
3	Michalis Nicolaidis	N/A	https://www.facebook.com/michalis.omonioia.oin...
4	Ραφαέλλα Κοζάκου	Female	https://www.facebook.com/rkozakou?fref=pb&...
5	Eleni Soanou	Female	https://www.facebook.com/eleni.sohanou.921?f...
6	Givraos Kezouis	N/A	https://www.facebook.com/profile.php?id=131...
7	Ευαννελία Πανανιώτου	Female	https://www.facebook.com/evanaeliapanaaioto...
8	Michalis Stvlianou	N/A	https://www.facebook.com/smihalis?fref=pb&h...
NULL	NULL	NULL	NULL

Εικόνα 23: Ολοκληρωμένος ο πίνακας friends από την συλλογή πληροφοριών

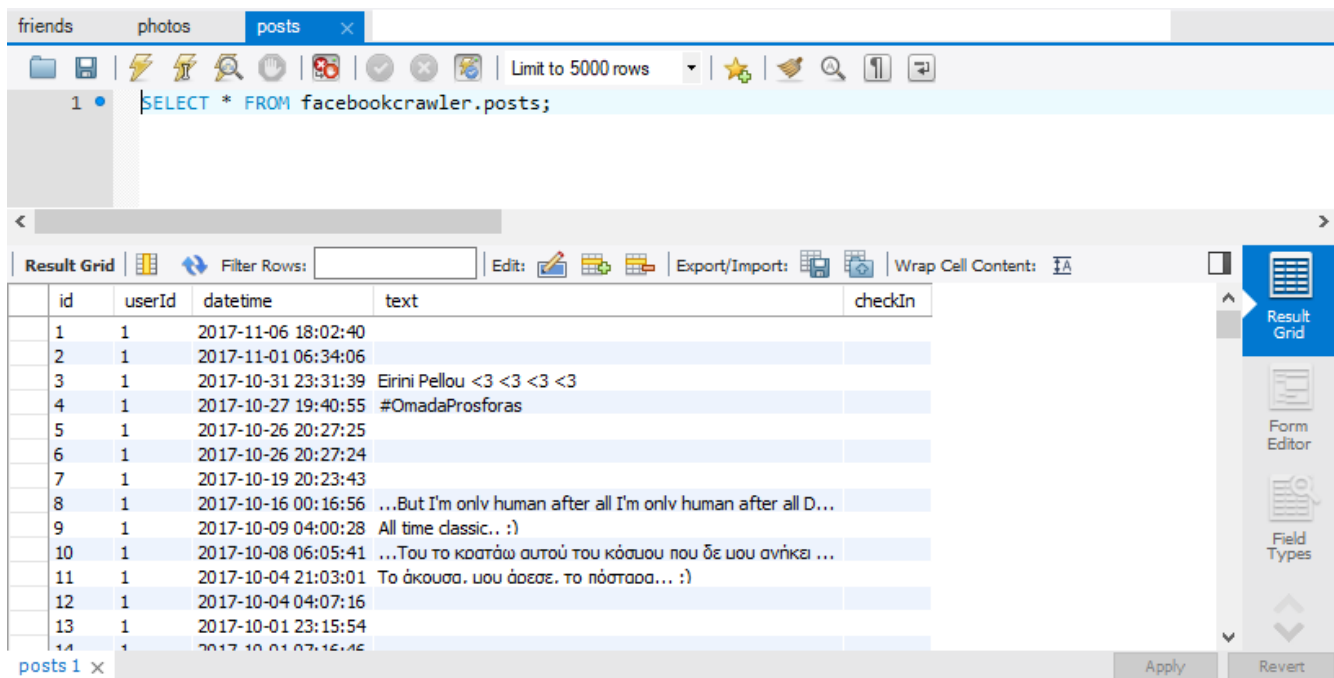
Στην συνέχεια βλέπουμε στην εικόνα 24 ο πίνακας photos έχει γεμίσει με τις φωτογραφίες των 8 φίλων που υπάρχουν στον λογαριασμό του Facebook μου. Οι φωτογραφίες που έχουν συλλεχθεί αντιστοιχούν σε κάθε post που έχει γίνει και συμπεριλάμβανε φωτογραφία για τον κάθε χρήστη ξεχωριστά.

id	postId	photo
1410	115	BLOB
1411	115	BLOB
1412	119	BLOB
1413	119	BLOB
1414	158	BLOB
1415	158	BLOB
1416	158	BLOB
1417	158	BLOB
1418	159	BLOB
1419	159	BLOB
1420	159	BLOB
1421	159	BLOB
1422	171	BLOB
1423	171	BLOB

Εικόνα 24: Ολοκληρωμένος ο πίνακας photos από την συλλογή πληροφοριών

Τέλος, στην εικόνα 25 παρατηρούμε ότι ο πίνακας posts έχει γεμίσει με τις posts των 8 φίλων που υπάρχουν στον λογαριασμό του Facebook μου. Τα posts που έχουν συλλεχθεί αντιστοιχούν σε κάθε φίλο με το userID του κάθε ενός και ακολούθως φαίνεται η ώρα και η μέρα, το κείμενο που υπάρχει στο post που έχει κάνει και εάν έχει κάνει check in (εικόνα 26).





Εικόνα 25: Ολοκληρωμένος ο πίνακας posts από την συλλογή πληροφοριών

id	userId	datetime	text	checkIn
9	5	2017-12-23 10:32:21	Χρόνια πολλά Ντίνα μας! ♥♥♥	
10	5	2017-12-21 09:13:56	Στα αστέρια πάει!!!! Άνοιξε τα φτερά σου πετάει....	NOIR Tapas & Bar

Εικόνα 26: Πίνακας posts με check in από την συλλογή πληροφοριών

Όπως αναφέρθηκε στο κεφάλαιο 4 αναλυτικά ο κάθε πίνακας στην βάση δεδομένων έχει τα δικά του χαρακτηριστικά.

## 5.2 Αποτελέσματα κλάσης view\_posts.py

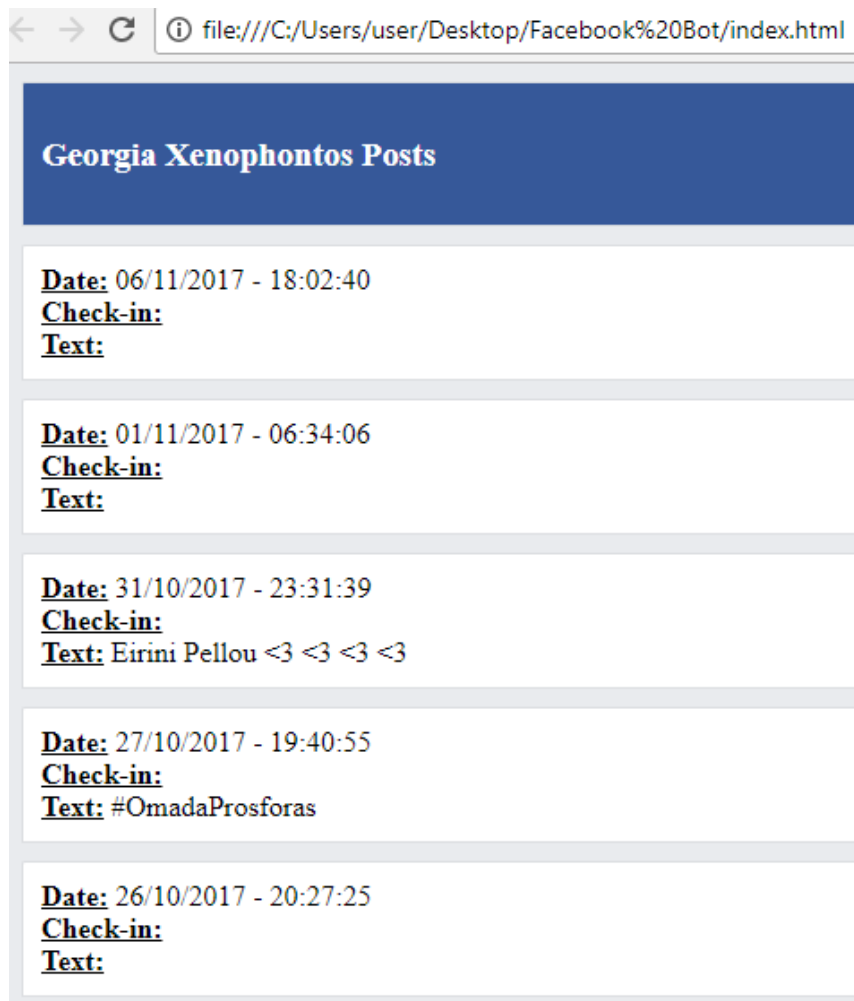
Με την κλάση view\_posts.py γίνεται μια δημιουργία σε HTML. Τρέχοντας την κλάση αυτή μας εμφανίζεται στην κονσόλα να επιλέξουμε ένα αριθμό από το 1 ως το 8 που αντιστοιχούν στα ονόματα των φίλων. Επιλέγοντας τον αριθμό, ανοίγουμε τον browser μας που έχει μετονομαστεί σε index (όπου βρίσκεται στο αρχείο του bot στον υπολογιστή μας) και μας εμφανίζει το προφίλ του χρήστη που επιλέξαμε σε αρχείο HTML (Εικόνα 27 και Εικόνα 28).

```
IPython console
Console 1/A
Please select a person from the list to show his/her posts
1 - Georgia Xenophontos
2 - Mixalis Nicolaides
3 - Ραφαέλλα Κοζάκου
4 - Eleni Spanou
5 - Giwrgos Kezouis
6 - Ευαγγελία Παναγιώτου
7 - Michalis Stylianos

Enter number: 1
Writing (1/16)
Writing (2/16)
Writing (3/16)
Writing (4/16)
Writing (5/16)
Writing (6/16)
Writing (7/16)
Writing (8/16)
Writing (9/16)
Writing (10/16)
Writing (11/16)
Writing (12/16)
Writing (13/16)
Writing (14/16)
Writing (15/16)
Writing (16/16)
Done! Open index.html file.

In [3]: |
```

Εικόνα 27: Η κονσόλα της κλάση view\_posts



Εικόνα 28: Το προφίλ του χρήστη σε html αρχείο βάση

### 5.3 Αποτελέσματα κλάσης ngrams.py

Τα NGrams χρησιμοποιήθηκαν για να δείξουν τα αποτελέσματα από τα συναισθήματα που εκφράζουν οι χρήστες του Facebook.

#### 5.3.1 Ψυχοσύνθεση χρηστών μέσα από το Facebook

##### 5.3.1.1 Τι είναι ψυχοσύνθεση

Με τον όρο ψυχοσύνθεση εννοούμε μια ψυχολογική θεωρία και πρακτική, που αναπτύχθηκε από την ψυχανάλυση και εξελίχθηκε στην Ανθρωπιστική-Υπαρξιακή ψυχολογία και ακολούθως στην Υπερπροσωπική ψυχολογία. Εμπνευστής της ο Ιταλός ψυχίατρος Roberto Assagioli (1888-1974). Φίλος του C.G. Jung και μέλος της Διεθνούς Ψυχαναλυτικής Εταιρείας, ο Assagioli

απομακρίνθηκε σταδιακά από τη φροϋδική σκέψη, που θεωρούσε δύσκαμπτη και περιορισμένη, σε σχέση με την ευρύτητα και πολυπλοκότητα του ανθρώπινου ψυχισμού. [17]

### **5.3.1.2 Γιατί να χρησιμοποιήσει κάποιος την Εκφραστική γραφή;**

Σύμφωνα με την ψυχοθεραπεύτρια και συγγραφέα Kathleen Adams υπάρχουν πολλοί λόγοι:

- Για ν' ανακαλύψει το συγγραφέα μέσα του/ της.
  - Για να κρατά ένα αρχείο με τις αλλαγές στη ζωή του/της.
  - Για να γνωρίσει διαφορετικά κομμάτια του εαυτού του/της.
  - Για να έχει ένα πολύτιμο εργαλείο στην ψυχοθεραπευτική διαδικασία.
  - Για να βελτιώσει τις σχέσεις του/της.
  - Για να έχει πρόσβαση στις συνειδητές και ασυνείδητες σκέψεις του/της.
  - Για να επεξεργάζεται τα όνειρά του/της.
  - Για ν' αναπτύξει τη διαίσθησή του/ της.
  - Για ν' αυξήσει τη δημιουργικότητά του/της.
  - Για να βελτιώσει την απόδοσή του στη δουλειά ή τις σπουδές.
  - Για ν' αναγνωρίζει του κύκλους και τα μοτίβα που επαναλαμβάνει στη ζωή του/της.
- (Adams, 1990) [17]

### **5.3.2 Έκφραση συναισθημάτων στο Facebook**

Στις μέρες μας οι νέοι άνθρωποι αλλά και οι περισσότεροι χρήστες του Facebook χρησιμοποιούν τον γραπτό λόγο για να εκφράσουν τα συναισθήματά τους. Η γραφή βασίζεται σε βιωματικές εμπειρίες ζωής για να έχουμε περαιτέρω (επιθυμητά) αποτελέσματα. Δεν αρκεί πάντα απλά να πιάσουμε ένα μολύβι ή το πληκτρολόγιο και ν' αρχίσουμε να γράφουμε (Adams, 1999). Χρειάζεται να υπάρχει κάποιος σκοπός και να εφαρμόζονται συγκεκριμένες τεχνικές. Η γραφή μπορεί να χρησιμοποιηθεί για την έκφραση των συναισθημάτων ή και ως συμπληρωματικό εργαλείο στην ψυχοθεραπεία. Αναλόγως με το σκοπό, λέγεται «εκφραστική» ή «θεραπευτική». Για θεραπευτικούς

σκοπούς μπορεί να χρησιμοποιηθεί σε περιπτώσεις μετατραυματικού στρες, κατάθλιψης, αγχώδων διαταραχών, διατροφικών διαταραχών κλπ. με την καθοδήγηση ειδικά εκπαιδευμένων συμβούλων, ψυχολόγων ή ψυχοθεραπευτών.

Όταν κάποιος αποτυπώνει στο χαρτί τις σκέψεις του, δίνει διέξοδο στα συναισθήματά του, που προκλήθηκαν από ένα πρόβλημα ή κάποιος γεγονός στη ζωή του, ευχάριστο ή δυσάρεστο. Ακόμη κι αν απλά καταγράφει σημειώσεις, ποίηση, αποφθέγματα ή όνειρα έρχεται πιο κοντά στον πραγματικό του εαυτό. Είναι ένα παράθυρο σε ό,τι έχει σημασία για κάποιον. [17]

Το Facebook πλέον έχει προσθέσει πρόσφατα μια επιλογή η οποία με ένα κουμπί ή ένα εικονίδιο εκφράζει τα συναισθήματα του κάθε χρήστη. Αυτή η τεχνική φαίνεται να εξηπηρετεί πολύ τους χρήστες αφού μπορούν χωρίς κόπο και πολλές λέξεις να εκφράζουν το πώς νοιώθουν.

Όπως για παράδειγμα ο πιο κάτω χρήστης νοιώθει υπέροχα επειδή είναι καλοκαίρι με βάση το τραγούδι που ανέβασε(Εικόνα 17).



Εικόνα 17: Παράδειγμα από post για συναισθήματα

### 5.3.3 Έκφραση συναισθημάτων ανά χρονικές περιόδους στο Facebook

Το Facebook τα τελευταία χρόνια, έχει πάρει μια διαφορετική τροπή. Ο αρχικός σκοπός του Facebook ήταν να φέρει κοντά ανθρώπους μέσω της επικοινωνίας, κάτι «αθώο». Σήμερα οι άνθρωποι χρησιμοποιούν το Facebook για να αναβιβάζουν τις απόψεις τους, να δείχνουν τον τρόπο ζωής που κάνουν, να δέχονται κριτικές από άγνωστα πρόσωπα και να έχουν τόσους πολλούς εικονικούς φίλους με αποτέλεσμα να χανθεί η ουσία της πραγματικής ζωής.

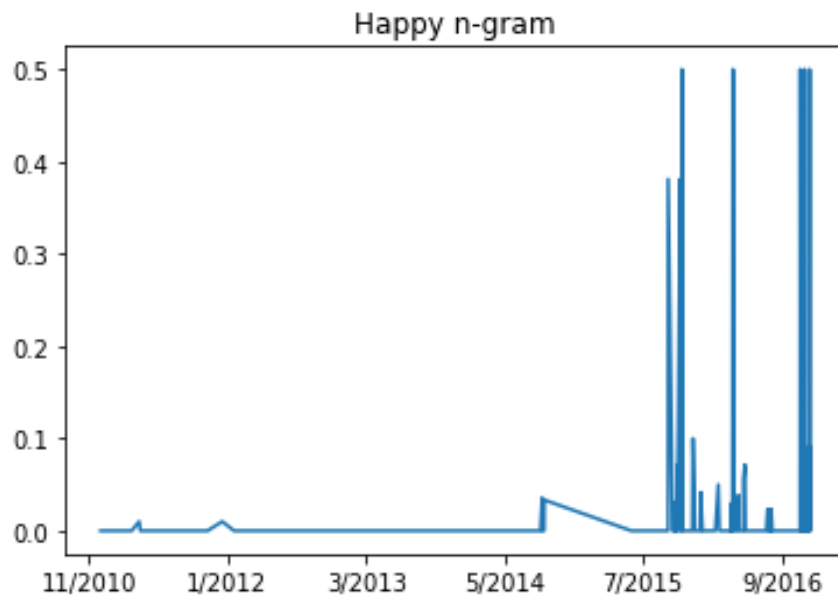
Στα πιο κάτω γραφήματα παρατηρούμε πώς οι χρήστες που έχω στο δικό μου λογαριασμό Facebook είχαν εναλλαγές ανά διαστήματα και τότε άρχισαν να εκφράζουν τα συναισθήματα είτε θετικά είτε αρνητικά. Αξιοσημείωτο είναι να αναφερθεί ότι τα γραφήματα έχουν δημιουργηθεί βάση κάποιων συγκεκριμένων λέξεων και χαρακτήρων όπου με την χρήση κώδικα για το OSNIT και βάσης δεδομένων έχω πάρει τα πιο κάτω αποτελέσματα. Επιπλέον ανάλυση θα γίνει στο κεφάλαιο 6 όπου θα αναλυθούν όλα τα αποτελέσματα αναλυτικά.

### 5.3.4 Ανάλυση γραφημάτων n-grams

Στα πιο κάτω γραφήματα παρατηρούμε πώς οι χρήστες που έχω στο δικό μου λογαριασμό Facebook είχαν εναλλαγές ανά διαστήματα και τότε άρχισαν να εκφράζουν τα συναισθήματα είτε θετικά είτε αρνητικά. Αξιοσημείωτο είναι να αναφερθεί ότι τα γραφήματα έχουν δημιουργηθεί βάση κάποιων συγκεκριμένων λέξεων και χαρακτήρων όπου με την χρήση κώδικα για το OSNIT και βάσης δεδομένων έχω πάρει τα πιο κάτω αποτελέσματα. Παραθέτω πιο κάτω μια επεξηγηματική ανάλυση για το κάθε γράφημα.

1. Το γράφημα «Happy n-gram» παρουσιάζει το συναίσθημα της χαράς σε χρονικές περιόδους από ένα συνολικό αποτέλεσμα για όλους τους χρήστες. Οι λέξεις που χρησιμοποιήθηκαν για την δημιουργία αυτού του ngram είναι λέξεις και σύμβολα που έδειχναν την χαρά όπως η λέξη “happy” ή τα σύμβολα ‘:’ και ‘)’ που όταν ενωθούν σχηματίζουν μια φατσούλα χαρούμενη που χρησιμοποιούν οι περισσότεροι χρήστες « :) ».

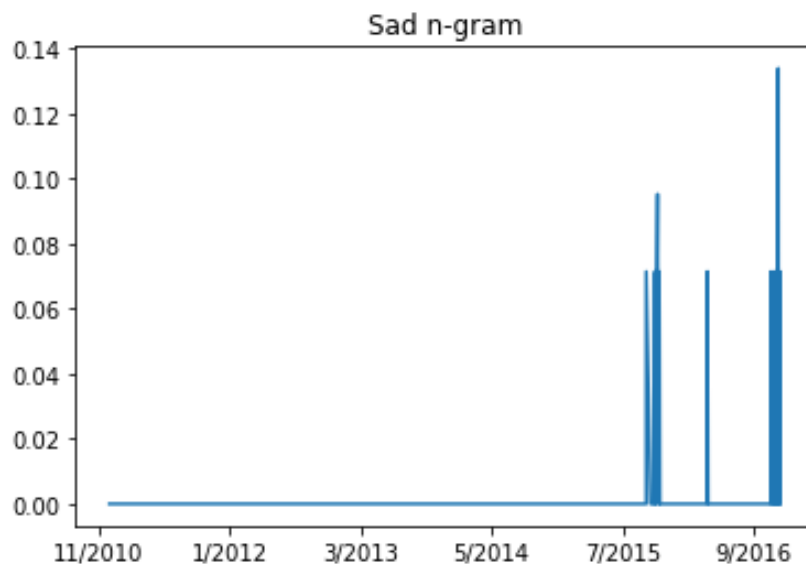
Επιπλέον φαίνεται ότι σε αυτό το γράφημα στατιστικά τα 2-3 τελευταία χρόνια οι χρήστες έχουν σκαμπανεβάσματα στο θέμα της χαράς ενώ τα προηγούμενα χρόνια ήταν στάσιμοι.



Εικόνα 29: Γράφημα «Happy n-gram»

2. Το γράφημα «Sad n-gram» παρουσιάζει το συναίσθημα της λύπης σε χρονικές περιόδους από ένα συνολικό αποτέλεσμα για όλους τους χρήστες. Οι λέξεις που χρησιμοποιήθηκαν για την δημιουργία αυτού του ngram είναι λέξεις και σύμβολα που έδειχναν την λύπη όπως η λέξη "sad" ή τα σύμβολα ':' και '(' που όταν ενωθούν σχηματίζουν μια φατσούλα λυπημένη που χρησιμοποιούν οι περισσότεροι χρήστες « :( ».

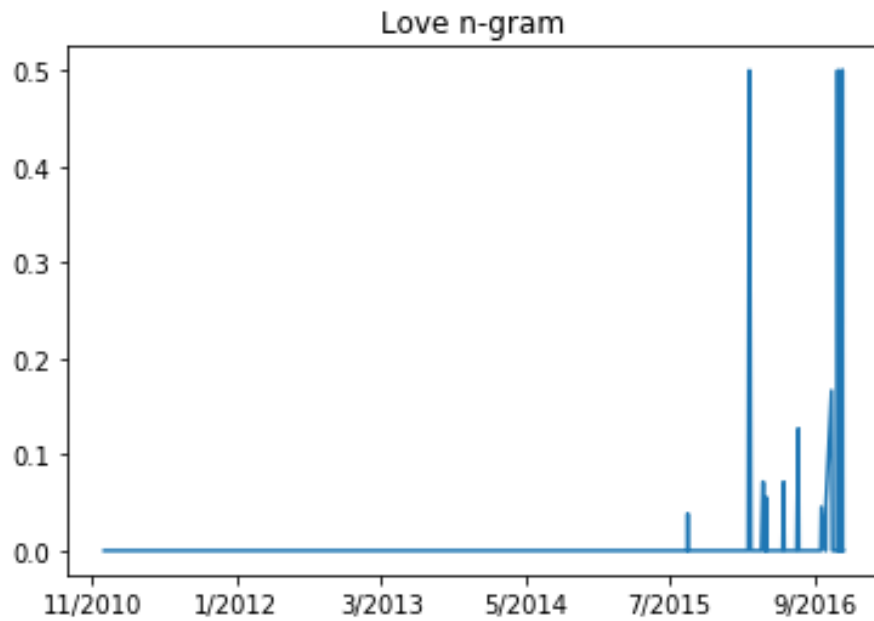
Επιπλέον φαίνεται ότι σε αυτό το γράφημα στατιστικά τα 2 τελευταία χρόνια οι χρήστες έχουν πολύ λίγες και συγκεκριμένες στιγμές που είναι λυπημένοι.



Εικόνα 30: Γράφημα «Sad n-gram»

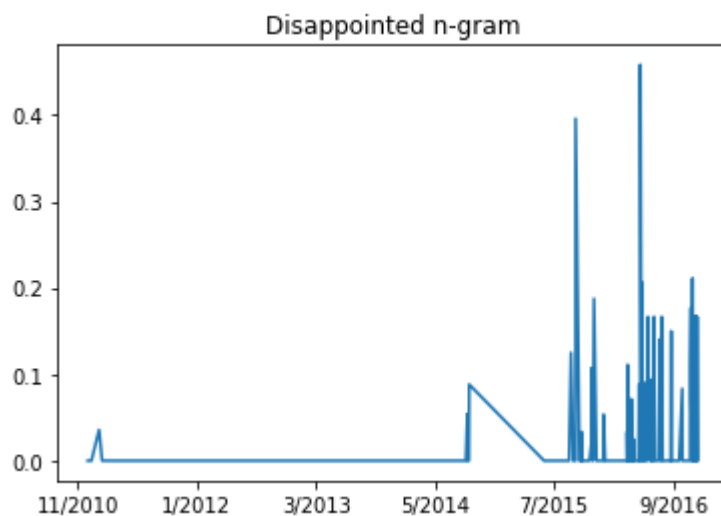
3. Το γράφημα «Love n-gram» παρουσιάζει το συναίσθημα της αγάπης σε χρονικές περιόδους από ένα συνολικό αποτέλεσμα για όλους τους χρήστες. Οι λέξεις που χρησιμοποιήθηκαν για την δημιουργία αυτού του ngram είναι λέξεις, αριθμούς και σύμβολα που έδειχναν την αγάπη ή τον έρωτα όπως η λέξη "love" ή το σύμβολο '<' και τον αριθμό '3' που όταν ενωθούν σχηματίζουν μια καρδιά που χρησιμοποιείτε από τους περισσότερους χρήστες « <3 ».
- Επιπλέον φαίνεται ότι σε αυτό το γράφημα στατιστικά τα 2 τελευταία χρόνια οι χρήστες έχουν αυξημένες στιγμές που είναι ερωτευμένοι ή αγαπούν τους γύρω τους.





Εικόνα 31: Γράφημα «Love n-gram»

4. Το γράφημα «Disappointed n-gram» παρουσιάζει το συναίσθημα της απογοήτευσης σε χρονικές περιόδους από ένα συνολικό αποτέλεσμα για όλους τους χρήστες. Οι λέξεις που χρησιμοποιήθηκαν για την δημιουργία αυτού του ngram είναι για παράδειγμα η λέξη «Disappointed» που φανερώνει την απογοήτευση. Επιπλέον φαίνεται ότι σε αυτό το γράφημα στατιστικά τα 3 τελευταία χρόνια οι χρήστες έχουν αυξημένες στιγμές που είναι απογοητευμένοι.



Εικόνα 32: Γράφημα «Disappointed n-gram»

# Κεφάλαιο 6

## Συμπεράσματα

### 6.1 Συμπεράσματα

Στόχος της παρούσας διπλωματικής ήταν η συλλογή και επεξεργασία των πληροφοριών που μπορούν να συλλεχθούν από το πιο δημοφιλές κοινωνικό δίκτυο το Facebook όπου στην συνέχεια να εξάγει γραφήματα n-grams που να φανερώνουν πώς αλλάζει ανά διαστήματα η ψυχοσύνθεση των χρηστών αλλά και δημιουργία προφίλ σε html αρχείο με μαζεμένες όλες τις αναρτήσεις που έκαναν τα τελευταία χρόνια συμπεριλαμβανομένων και των φωτογραφιών τους.

Χρησιμοποιώντας τεχνικές OSINT και δημιουργώντας μεθόδους που αναφέρθηκαν και αναλυθήκαν σε αντίστοιχα κεφάλαια καταφέραμε να συλλέξουμε πληροφορίες από το Facebook χωρίς να μας ανακαλύψει οποιοσδήποτε χρήστης. Οι τεχνικές αυτές λειτουργούσαν κατά διαστήματα για την εκπόνηση της συγκεκριμένης εργασίας δηλαδή από τον Αύγουστο έως τον Νοέμβριο του 2017.

Ο συνδυασμός των παραπάνω λειτουργιών έγινε μέσω της κλάσης FacebookBot στην γλώσσα προγραμματισμού Python. Τα αποτελέσματα για την ανάλυση της ψυχοσύνθεσης των ατόμων που διατηρούν προφίλ στο Facebook έγιναν από απλούς πολίτες ηλικίας 22-28 χρόνων από τους οποίους έχω πάρει συγκατάθεση για την συμμετοχή τους στην παρούσα έρευνα. Ο τρόπος που νέες ηλικίες χρησιμοποιούν τους λογαριασμούς τους ήταν αρκετά εκφραστικός κυρίως τα τελευταία 2-3 χρόνια. Τέλος

επικεντρωθήκαν μόνο στην συλλογή πληροφοριών από τα προφίλ τους αποδεικνύοντας ότι η παρακολούθηση του οποιοδήποτε χρήστη είναι εύκολη χωρίς να γίνεσαι αντιληπτός αλλά άκρως επικίνδυνη γιατί πολύ εύκολα ο οποιοσδήποτε συλλέγει πληροφορίες για το κάθε άτομο που μπορεί να τις χρησιμοποιήσει με κακόβουλο τρόπο.

Τα αποτελέσματα που έχω πάρει από τα γραφήματα Ngram που αναλύθηκαν στο Κεφάλαιο 5, είναι έγκυρα αλλά όχι τόσο ακριβές γιατί είναι συνολικό το ποσό για όλους τους χρήστες. Τα αποτελέσματα δεν τα έχω συγκρίνει με κάποια άλλη δουλειά αφού είναι και η μόνη που έγινε στο κομμάτι της ψυχολογίας. Θεωρώ ότι έχει γίνει αρκετά καλή δουλειά για την διεκπεραίωση του στόχου της παρούσας διπλωματικής εργασίας γιατί στις μέρες μας οι άνθρωποι εκφράζουν το πως νοιώθουν μέσα από το Facebook. Επιπλέον θα μπορούσε να αναπτυχθεί η παρούσα δουλειά με πιο εξειδικευμένους στόχους για να εξυπηρετήσει και άλλους οργανισμούς. Τέλος σύμφωνα με την παρούσα έρευνα, καταλήγουμε στο συμπέρασμα ότι μπορούν να αξιοποιηθεί από εταιρίες που ασχολούνται με Nation Security για να αποτρέπουν επιθέσεις από insider threats.

## **6.2 Μελλοντική δουλειά**

Μελλοντική βελτίωση της συγκεκριμένης εργασίας, θα μπορούσε να χρησιμοποιηθούν περισσότερα μέσα κοινωνικής δικτύωσης όπως το Twitter, LinkedIn και Instagram και να ενσωματώσουμε ένα script με web interface που να βγάζει αυτόματα n-grams ή γραφικές με επιλεγμένα συναισθήματα για επιλεγμένους χρήστες. Επιπλέον θα μπορούσαμε να επιλέγουμε διάσημους ή πολιτικούς στα μέσα κοινωνικής δικτύωσης και σαν κακόβουλος χρήστης να απομονώνονται μόνο οι σημαντικές πληροφορίες όπως ηλικία, τηλέφωνο, προσωπικές φωτογραφίες και αναρτήσεις. Επίσης, η παρούσα εργασία μπορεί να αξιοποιηθεί από μεγάλους οργανισμούς σε ενσωματωμένα script στα προγράμματα περιήγησης των υπαλλήλων για αποφυγή εσωτερικής διαρροής σημαντικών πληροφοριών. Τέλος, θα μπορούσε να χρησιμοποιηθεί σε διαδικτυακές ομάδες σχολείων για να αποτυπωθεί το να αποτυπώσουν το κοινωνικό προφίλ των μαθητών με τις περισσότερες απουσίες ή αποβολές στο σχολείο, όπου φανερώνεται μέσα από την κινητικότητα τους στα μέσα κοινωνικής δικτύωσης.

[ΛΕΥΚΗ ΣΕΛΙΔΑ]

## Βιβλιογραφία

- [01] As defined in Sec. 931 of Public Law 109-163, entitled, "National Defense Authorization Act for Fiscal Year 2006."
- [02] Richelson, Jeffrey T. The U.S. Intelligence Community. Avalon Publishing. ISBN 9780813349190. Retrieved 15 May 2017.
- [03] George, edited by Roger Z; Kline, Robert D; Lownethal, Mark M (2005). Intelligence and the national security strategist : enduring issues and challenges. Lanham: Rowman and Littlefield. ISBN 9780742540392.
- [04] "Spy Agencies Turn to Newspapers, NPR, and Wikipedia for Information: The intelligence community is learning to value 'open-source' information". Retrieved 2008-09-15.
- [05] SearchMicroservices. (2017). What is crawler? - Definition from WhatIs.com. [online] Available at: <http://searchmicroservices.techtarget.com/definition/crawler>
- [06] Ellison, N. B., & Boyd, D. (2007). Social network sites: Definition, history, and scholarship. Journal of Computer-Mediated Communication, 13(1), 210-230.
- [07] Quan-Haase, A., & Young, A. L. (2010). Uses and gratifications of social media: A comparison of Facebook and instant messaging. Bulletin of Science, Technology & Society, 30(5), 350-361.
- [08] Coughlan, T., & Perryman, L.-A. (2015). A Murky Business: Navigating the Ethics of Educational Research in Facebook Groups. Available from [http://www.eurodl.org/materials/special/2015/Coughlan\\_Perryman.htm](http://www.eurodl.org/materials/special/2015/Coughlan_Perryman.htm)
- [09] Facebook.com. (2015). Available from <http://www.facebook.com>
- [10] Gross, R., & Acquisti, A. (2005, November). Information revelation and privacy in online social networks. Paper presented at the WPES'05, Alexandria, Virginia.

- [11] Python.org. (2017). *What is Python? Executive Summary*. [online] Available at: <https://www.python.org/doc/essays/blurb/>.
- [12] Pythonhosted.org. (2017). *Spyder - Documentation — Spyder 3 documentation*. [online] Available at: <https://pythonhosted.org/spyder/>.
- [13] Seleniumhq.org. (2017). *Selenium - Web Browser Automation*. [online] Available at: <http://www.seleniumhq.org/>.
- [14] Selenium-python.readthedocs.io. (2017). *Selenium with Python — Selenium Python Bindings 2 documentation*. [online] Available at: <http://selenium-python.readthedocs.io/>.
- [15] Speech and Language Processing. Daniel Jurafsky & James H. Martin. Draft of September 1, 2014
- [16] Pythonhosted.org. (2017). Tutorial — Python NGram 3.3 documentation. [online] Available at: <https://pythonhosted.org/ngram/tutorial.html>.
- [17] E. (2017). ΨΥΧΟΣΥΝΘΕΣΗ. [online] [Psycosynthesis.blogspot.com.cy](http://psycosynthesis.blogspot.com.cy). Available at: <http://psycosynthesis.blogspot.com.cy/>.
- [18] Deptment, P. (2017).Pi.ac.cy. Available at: [http://www.pi.ac.cy/InternetSafety/sec\\_facebook.html](http://www.pi.ac.cy/InternetSafety/sec_facebook.html)
- [19] Cyberinsurancequote.gr. (2017). Νομοθεσία (GDPR) :: Cyber Insurance Quote.gr. [online] Available at: <http://www.cyberinsurancequote.gr/insurance/nomothesia/>
- [20] OSINT-Group, Overview, 2012. Retrieved 2 July 2012, <http://www.theosintgroup.com/overview.html>.
- [21] See for instance: The United Kingdom Intelligence Services Act 1994 or the Australian Intelligence Services Act 2001.
- [22] Kerpen, D. (2012). *Likeable Social Media*, McGraw-Hill Education – Europe, ISBN13: 9780071813723.

- [23] Kietzmann, H. J. and Kristopher, H. (2011). "Social media? Get serious! Understanding the functional building blocks of social media". *Business Horizons* 54: 241–251.
- [24] Lardi, K. and Fuchs, R. (2013). *Social Media Strategy - A step-by-step guide to building your social business* (1st ed.). Zurich: vdf. ISBN 978-3-7281-3557-5.
- [25] Maceda, M. (2017). *Dysfunction Junction: Intelligence, Peacekeeping, and the United Nations*. Monterrey, California, p.3.
- [26] Tise, V. (2007). *The Use of Open Source Information & Planning Terrorism Scenarios*. CDCA First Annual C4ISR Transformation and Fusion Government and Industry Conference, p.1.
- [27] Computerweekly.com. *Passive Recon - Nine must-have OSINT tools*. [online] Available at: <http://www.computerweekly.com/photostory/2240160136/Nine-must-have-OSINT-tools/10/9-Passive-Recon>.
- [28] Expertsystem.com. *Advantages and disadvantages of open source intelligence*. [online] Available at: <http://www.expertsystem.com/advantages-disadvantages-open-source-intelligence/>.
- [29] Globalresultspr.net. *The challenges of Open Source Intelligence (OSINT) in an increasingly connected age – Global Results Communications*. [online] Available at: <http://www.globalresultspr.net/blog/the-challenges-of-open-source-intelligence-osint-in-an-increasingly-connected-age/>.
- [30] Τσιλιγκιρίδης Θ. , Αποστολάκης Ι., Φούσκας Γ., Πατριαρχέας Κ. (2000). *Βασικές Υπηρεσίες Διαδικτύου – Βιβλίο Μαθητή* . (κεφ 2. σελ 54-60) ΑΘΗΝΑ . ΥΠΕΠΘ .
- [31] naftemporiki.gr. *Ο εθισμός στα social media*. [online] Available at: <http://www.naftemporiki.gr/story/1220853/o-ethismos-sta-social-media>.
- [32] Camp, L. (2001). *Trust and risk in Internet commerce*. Cambridge, MA.: MIT Press.

- [33] Kaplan, A. and Haenlein, M. (2012). Social media: back to the roots and back to the future. *Journal of Systems and Information Technology*, 14(2), pp.101-104.
- [34] Lowenthal, M. (2015). *Intelligence : from secrets to policy*. 5th ed. Los Angeles: CQ Press.
- [35] Harfield C & Harfield K, *Intelligence: Investigation, Community and Partnership* (2008) Oxford: Oxford University Press
- [36] Ratcliffe, J *Intelligence-Led Policing*, (2008) Devon: Willan Publishing
- [37] Harfield C & Harfield K, *Intelligence: Investigation, Community and Partnership* (2008) Oxford: Oxford University Press
- [38] Akhgar, B Bayerl, S Saathoff, G *Application of Big Data for National Security, A Practitioner's Guide to Emerging Technologies* London:Elsevier 2015.
- [39] Golden, J., Conroy, R., Bruce, I., Denihan, A., Greene, E., Kirby, M. and Lawlor, B. (2009). Loneliness, social support networks, mood and wellbeing in community-dwelling elderly. *International Journal of Geriatric Psychiatry*, 24(7), pp.694-700.
- [40] Kramer, A., Guillory, J. and Hancock, J. (2014). Experimental evidence of massive-scale emotional contagion through social networks. *Proceedings of the National Academy of Sciences*, 111(24), pp.8788-8790.
- [41] Christakis, N. and Fowler, J. (2012). Social contagion theory: examining dynamic social networks and human behavior. *Statistics in Medicine*, 32(4), pp.556-577.
- [42] Bollen, J., Gonçalves, B., Ruan, G. and Mao, H. (2011). Happiness Is Assortative in Online Social Networks. *Artificial Life*, 17(3), pp.237-251.



# Παράρτημα Α

## Python Classes

### A.1 Class Facebook\_Bot

```
# -*- coding: utf-8 -*-  
"""  
Created on Mon Jul 31 11:45:22 2017  
"""  
  
from selenium import webdriver  
from FacebookCrawler import FacebookCrawler  
from Database import Database  
import urllib.request  
  
if __name__ == "__main__":  
    #Initialize the crawler and block pop ups  
    chrome_options = webdriver.ChromeOptions()  
    prefs = {"profile.default_content_setting_values.notifications" : 2}  
    chrome_options.add_experimental_option("prefs",prefs)
```

```

crawler =
FacebookCrawler(webdriver.Chrome('chromedriver.exe',chrome_options=chrome_options))

#Initialize the databse

db = Database()

db.deleteAllFriends()

#Open facebook

print("Opening Facebook...")

crawler.driver.get(crawler.url)

#Log in

print("Logging in...")

crawler.login(crawler.email,crawler.password)

#Get friends list

print("Getting friend list...")

friendList = crawler.getFriendList()

print("Inserting friends to database...")

postId = 1

#Iterate over friends list

for i,friend in enumerate(friendList):

    try:

        print("Inserting {} ({} / {})".format(friend[0],i+1,len(friendList)))

        gender = crawler.getGender(friend[1])

        db.insertFriend(i+1,friend[0],gender,friend[1])

        print("Collecting posts...")

        posts = crawler.getPosts(friend[1])

        print("Inserting posts to database...")

        #Iterate over posts of friend

        for j,post in enumerate(posts):

            try:

                #Insert post to database

                print("Inserting posts ({} / {})".format(j+1,len(posts)))

                db.insertPost(postId,i+1,post[0],post[1],post[2])

```

```

#Iterate over possible photos in a post
for url in post[3]:
    try:
        urllib.request.urlretrieve(url,"temp.jpg")
        photo = open('temp.jpg','rb').read()
        #Insert photo to database
        db.insertPhoto(postId,photo)
    except:
        print("Error retrieving image at url: ",url)
    postId += 1
except:
    print("Error processing post with ID: ",j+1)
except:
    print("Error processing friend: ",friend[0])
print("\aExiting...")
crawler.driver.quit()

```

## A.2 Class view\_posts

```

# -*- coding: utf-8 -*-
"""
Spyder Editor
This is a temporary script file.
"""
import sys
import base64
from Database import Database

#Initialize the database and get friends names
db = Database()
names = list(db.selectFriendsNames())

```

```

print("Please select a person from the list to show his/her posts")
for i,name in enumerate(names):
    names[i] = name[0]
    print("{} - {}".format(i+1,name[0]))

try:
    choice = int(input("Enter number: "))
except ValueError:
    sys.exit("\ aWrong number!")

if choice < 1 or choice > len(names):
    sys.exit("\ aWrong number!")

#Select all posts of the friend selected
posts = list(db.selectPostsByName(names[choice-1]))

#Open an html file and write all posts in it
with open("index.html","w+",encoding='utf-8') as myFile:
    myFile.write("<!DOCTYPE html><html><head><link rel='stylesheet' type='text/css'
href='mystyle.css'></head><body><div class='user'><h3>{}
Posts</h3></div>".format(names[choice-1]))
    for count,post in enumerate(posts):
        print("Writing ({} / {})".format(count+1,len(posts)))
        div = "<div><b><u>Date:</b></u> {}<br><b><u>Check-in:</b></u>
{}<br><b><u>Text:</b></u> {}<br>".format(post[1].strftime("%d / %m / %Y -
%H:%M:%S"),post[3],post[2])
        photos = db.selectPhotosByPostId(post[0])
        for photo in photos:
            data_uri = base64.b64encode(photo[0]).decode('utf-8').replace('\n', '')
            div += ''.format(data_uri)
        div += "</div>"
    myFile.write(div)

```

```
myFile.write("</body></head></html>")
myFile.close()
print("Done! Open index.html file.")
```

## A.3 Class FacebookCrawler

```
# -*- coding: utf-8 -*-
"""
Created on Mon Jul 31 13:03:45 2017
"""
from selenium.webdriver.common.by import By
from selenium.webdriver.support.ui import WebDriverWait
from selenium.webdriver.support import expected_conditions as EC
from selenium.common.exceptions import NoSuchElementException
import time
from selenium.webdriver.common.keys import Keys
from datetime import datetime

class FacebookCrawler:
    def __init__(self,driver):
        self.url = 'https://www.facebook.com'
        self.email = 'emilia.panayiotou@hotmail.com'
        self.password = 'Emily1234!'
        self.driver = driver

    def login(self,email,password):
        wait = WebDriverWait(self.driver, 10)
        wait.until(EC.presence_of_element_located((By.ID,'email')))
        username = self.driver.find_element_by_id('email')
        username.send_keys(email)
        wait.until(EC.presence_of_element_located((By.ID,'pass')))
```

```

passwd = self.driver.find_element_by_id('pass')
passwd.send_keys(password,Keys.ENTER)

def getFriendList(self):

    wait = WebDriverWait(self.driver, 10)
    wait.until(EC.presence_of_element_located((By.CLASS_NAME,'_5afe')))
    profile = self.driver.find_element_by_class_name("_5afe")
    profile.click()

    url = self.driver.current_url.split('?')[0]
    url = url + '/friends'
    self.driver.get(url)

    flag = True

    uls_beforeScroll
=len(self.driver.find_elements_by_xpath("//div[@id='pagelet_timeline_app_collection_1155995189:23
56318349:2']/ul"))

    while(flag):

        self.driver.execute_script("window.scrollTo(0,document.body.scrollHeight)")

        time.sleep(1)

        uls_afterScroll =
len(self.driver.find_elements_by_xpath("//div[contains(@id,'pagelet_timeline_app_collection_')]
/ul"
)

        if(uls_afterScroll == uls_beforeScroll):

            flag = False

        else:

            uls_beforeScroll = uls_afterScroll

    friends = self.driver.find_elements(By.XPATH,"//div[@class='fsl fwb fcb']")
    result = []
    for friend in friends:

```

```

name = friend.find_element_by_tag_name("a").text
url = friend.find_element_by_tag_name("a").get_attribute("href")
result.append([name,url])

return result

def getGender(self,friendUrl):
    self.driver.get(friendUrl)
    wait = WebDriverWait(self.driver, 10)
    wait.until(EC.presence_of_element_located((By.LINK_TEXT,"About")))
    profile = self.driver.find_element_by_link_text('About')
    profile.click()
    wait.until(EC.presence_of_element_located((By.LINK_TEXT,"Contact and Basic Info")))
    profile = self.driver.find_element_by_link_text("Contact and Basic Info")
    profile.click()
    self.driver.implicitly_wait(5)
    try:
        profile = self.driver.find_element_by_xpath("//li[@class='_3pw9 _2pi4 _2ge8 _3ms8']")
        return profile.text.split('\n')[1]
    except NoSuchElementException:
        return 'N/A'

def removePopUps(self):
    try:
        wait = WebDriverWait(self.driver, 10)
        wait.until(EC.presence_of_element_located((By.CLASS_NAME,"layerCancel")))
        cancel = self.driver.find_element_by_class_name("layerCancel")
        cancel.click()
    except:
        return

def getPosts(self,userUrl):
    #postsList = "fbUserStory" " _4-u2 mbm _4mrt _5jmm _5pat _5v3q _4-u8"

```

```

postList = "_4mrt"
self.driver.get(userUrl)
flag = True
uls_beforeScroll = len(self.driver.find_elements_by_class_name(postList))

while(flag):
    self.driver.execute_script("window.scrollTo(0,document.body.scrollHeight)")
    time.sleep(3)
    uls_afterScroll = len(self.driver.find_elements_by_class_name(postList))
    if(uls_afterScroll == uls_beforeScroll):
        flag = False
    else:
        uls_beforeScroll = uls_afterScroll
posts = self.driver.find_elements(By.CLASS_NAME,postList)
result = []

for i,post in enumerate(posts):
    print("Collecting ({} / {})".format(i+1,len(posts)))
    try:
        date = "
        date =
datetime.fromtimestamp(int(post.find_element_by_class_name("_5ptz").get_attribute("data-
utime"))).strftime('%Y-%m-%d %H:%M:%S')
    except:
        pass
    try:
        txt = "
        txt = post.find_element_by_class_name("userContent").find_element_by_tag_name("p").text
    except:
        pass
    try:
        checkIn = "

```



```

    checkIn = post.find_element_by_class_name("fcg").text
    if ' at ' in checkIn:
        checkIn = checkIn.split("at",1)[1]
        checkIn = checkIn[:-1]
    else:
        checkIn = ""
except:
    pass
try:
    photos = ""
    photos = post.find_elements_by_class_name("_5dec")
    photoUrls = []
    for photo in photos:
        url = photo.get_attribute("data-ploi")
        if type(url[0])!= type(None):
            photoUrls.append(url)
except:
    pass
if(date != ""):
    result.append([date,txt,checkIn,photoUrls])

return result

```

## A.4 Class ngrams

```

# -*- coding: utf-8 -*-
"""
Created on Wed Dec 02 11:06:59 2017
"""
from Database import Database
import ngram

```

```

from datetime import datetime
import numpy as np
import matplotlib.pyplot as plt

db = Database()
posts = db.selectPosts()

dates = []
happy,sad,love,disappointed = [],[],[],[]
happyAvg,sadAvg,loveAvg,disappointedAvg = [],[],[],[]
for post in posts:
    G = ngram.NGram(post[3].split())
    dates.append(post[2].timestamp())

    #happy keywords
    happy1 = G.search('happy')
    happy2 = G.search(':')
    if happy1 == []:
        happy.append(0)
    else:
        for x in happy1:
            happy.append(x[1])
    if happy2 == []:
        happy.append(0)
    else:
        for x in happy2:
            happy.append(x[1])
    avg = np.average(happy)
    happyAvg.append(avg)
    happy = []

    #sad keywords
    sad1 = G.search('RIP')
    sad2 = G.search(':')
    if sad1 == []:
        sad.append(0)
    else:
        for x in sad1:
            sad.append(x[1])

```

```

if sad2 == []:
    sad.append(0)
else:
    for x in sad2:
        sad.append(x[1])
avg = np.average(sad)
sadAvg.append(avg)
sad = []

#love keywords
love1 = G.search('xxx')
love2 = G.search('<3')
if love1 == []:
    love.append(0)
else:
    for x in love1:
        love.append(x[1])
if love2 == []:
    love.append(0)
else:
    for x in love2:
        love.append(x[1])
avg = np.average(love)
loveAvg.append(avg)
love = []

#disappointed keywords
disappointed1 = G.search('...')
disappointed2 = G.search('-.')
if disappointed1 == []:
    disappointed.append(0)
else:
    for x in disappointed1:
        disappointed.append(x[1])
if disappointed2 == []:
    disappointed.append(0)
else:
    for x in disappointed2:
        disappointed.append(x[1])

```

```

    avg = np.average(disappointed)
    disappointedAvg.append(avg)
    disappointed = []

#prepare data
unq, unq_inv, unq_cnt = np.unique(dates, return_inverse=True, return_counts=True)
indices = np.split(np.argsort(unq_inv), np.cumsum(unq_cnt[:-1]))

happySum = 0
happy = []
for index in indices:
    for i in index:
        happySum = happySum + happyAvg[i]
    happy.append(happySum / len(index))
    happySum = 0

sadSum = 0
sad = []
for index in indices:
    for i in index:
        sadSum = sadSum + sadAvg[i]
    sad.append(sadSum / len(index))
    sadSum = 0

loveSum = 0
love = []
for index in indices:
    for i in index:
        loveSum = loveSum + loveAvg[i]
    love.append(loveSum / len(index))
    loveSum = 0

disappointedSum = 0
disappointed = []
for index in indices:
    for i in index:
        disappointedSum = disappointedSum + disappointedAvg[i]
    disappointed.append(disappointedSum / len(index))
    disappointedSum = 0

```

```

ts = np.linspace(unq[0],unq[-1],8)
xticks = []
for t in ts:
    xticks.append(str(datetime.fromtimestamp(t).month) + '/' + str(datetime.fromtimestamp(t).year))

#plot data
#happy
fig, ax = plt.subplots()
ax.plot(unq,happy)
ax.set_title('Happy n-gram')
ax.set_xticklabels(xticks)
plt.show()
#sad
fig, ax = plt.subplots()
ax.plot(unq,sad)
ax.set_title('Sad n-gram')
ax.set_xticklabels(xticks)
plt.show()
#love
fig, ax = plt.subplots()
ax.plot(unq,love)
ax.set_title('Love n-gram')
ax.set_xticklabels(xticks)
plt.show()
#disappointed
fig, ax = plt.subplots()
ax.plot(unq,disappointed)
ax.set_title('Disappointed n-gram')
ax.set_xticklabels(xticks)
plt.show()

```

## A.5 Class Database

```

# -*- coding: utf-8 -*-
"""

```

Created on Wed Aug 9 12:52:29 2017

```

"""
import pymysql

class Database:
    def __init__(self):
        self.host = 'localhost'
        self.user = 'root'
        self.password = 'root123'
        self.db = 'facebookcrawler'

    def insertFriend(self,i,name,gender,url):
        query = "Insert into friends(id,name,gender,url) values(%s,%s,%s,%s)"
        args = (i,name,gender,url)
        connection = pymysql.connect(self.host, self.user, self.password, self.db, use_unicode = True,
charset = "utf8")
        cursor = connection.cursor()

        try:
            cursor.execute(query,args)
            connection.commit()
        except:
            print("Error inserting friend to database")
            connection.rollback()
        finally:
            connection.close()
            cursor.close()

    def deleteAllFriends(self):
        query = "Delete from friends"
        connection = pymysql.connect(self.host, self.user, self.password, self.db, use_unicode = True,
charset = "utf8")
        cursor = connection.cursor()

        try:
            cursor.execute(query)
            connection.commit()
        except:
            print("Error deleting friends from database")

```

```

        connection.rollback()
    finally:
        connection.close()
        cursor.close()

def insertPost(self,i,uid,datetime,text,checkIn):
    query = "Insert into posts(id,userId,datetime,text,checkIn) values(%s,%s,%s,%s,%s)"
    args = (i,uid,datetime,text,checkIn)
    connection = pymysql.connect(self.host, self.user, self.password, self.db, use_unicode = True,
charset = "utf8mb4")
    cursor = connection.cursor()

    try:
        cursor.execute(query,args)
        connection.commit()
    except:
        print("Error inserting post to database")
        connection.rollback()
    finally:
        connection.close()
        cursor.close()

def deleteAllPosts(self):
    query = "Delete from posts"
    connection = pymysql.connect(self.host, self.user, self.password, self.db, use_unicode = True,
charset = "utf8")
    cursor = connection.cursor()

    try:
        cursor.execute(query)
        connection.commit()
    except:
        print("Error deleting posts from database")
        connection.rollback()
    finally:
        connection.close()
        cursor.close()

def insertPhoto(self,i,photo):

```

```

query = "Insert into photos(postId,photo) values(%s,%s)"
args = (i,photo)
connection = pymysql.connect(self.host, self.user, self.password, self.db, use_unicode = True,
charset = "utf8")
cursor = connection.cursor()

try:
    cursor.execute(query,args)
    connection.commit()
except:
    print("Error inserting photo to database")
    connection.rollback()
finally:
    connection.close()
    cursor.close()

def selectFriendsNames(self):
    query = "Select name from friends"
    connection = pymysql.connect(self.host, self.user, self.password, self.db, use_unicode = True,
charset = "utf8")
    cursor = connection.cursor()

    try:
        cursor.execute(query)
        return cursor.fetchall()
    except:
        print("Error selecting names from database")
        connection.rollback()
    finally:
        connection.close()
        cursor.close()

def selectPostsByName(self,name):
    query = "Select posts.id,datetime,text,checkIn from friends,posts where friends.id = posts.userId
and name = %s"
    args = (name)
    connection = pymysql.connect(self.host, self.user, self.password, self.db, use_unicode = True,
charset = "utf8")
    cursor = connection.cursor()

```



```

try:
    cursor.execute(query,args)
    return cursor.fetchall()
except:
    print("Error selecting names from database")
    connection.rollback()
finally:
    connection.close()
    cursor.close()

def selectNumberOfPostsByName(self,name):
    query = "Select count(*) from friends,posts where friends.id = posts.userId and name = %s"
    args = (name)
    connection = pymysql.connect(self.host, self.user, self.password, self.db, use_unicode = True,
charset = "utf8")
    cursor = connection.cursor()

    try:
        cursor.execute(query,args)
        return cursor.fetchone()[0]
    except:
        print("Error selecting names from database")
        connection.rollback()
    finally:
        connection.close()
        cursor.close()

def selectPhotosByPostId(self,i):
    query = "select photo from photos where postId = %s"
    args = (i)
    connection = pymysql.connect(self.host, self.user, self.password, self.db, use_unicode = True,
charset = "utf8")
    cursor = connection.cursor()

    try:
        cursor.execute(query,args)
        return cursor.fetchall()
    except:

```

```

        print("Error selecting photos from database")
        connection.rollback()
    finally:
        connection.close()
        cursor.close()

    def selectNumberOfPhotosByName(self,name):
        query = "select count(*) from friends,posts,photos where friends.id = posts.userId and posts.id =
photos.postId and name = %s"
        args = (name)
        connection = pymysql.connect(self.host, self.user, self.password, self.db, use_unicode = True,
charset = "utf8")
        cursor = connection.cursor()

        try:
            cursor.execute(query,args)
            return cursor.fetchone()[0]
        except:
            print("Error selecting number of photos from database")
            connection.rollback()
        finally:
            connection.close()
            cursor.close()

    def selectCountByGender(self,gender):
        query = "SELECT count(*) FROM friends where gender = %s"
        args = (gender)
        connection = pymysql.connect(self.host, self.user, self.password, self.db, use_unicode = True,
charset = "utf8")
        cursor = connection.cursor()

        try:
            cursor.execute(query,args)
            return cursor.fetchone()[0]
        except:
            print("Error selecting gender from database")
            connection.rollback()
        finally:
            connection.close()

```

```

        cursor.close()

    def selectPostsByMonth(self):
        query = "SELECT  EXTRACT(MONTH FROM datetime) as month,COUNT(EXTRACT(MONTH
FROM datetime)) as posts from posts group by EXTRACT(month from datetime)"
        connection = pymysql.connect(self.host, self.user, self.password, self.db, use_unicode = True,
charset = "utf8")
        cursor = connection.cursor()

        try:
            cursor.execute(query)
            return cursor.fetchall()
        except:
            print("Error selecting posts from database")
            connection.rollback()
        finally:
            connection.close()
            cursor.close()

    def selectPostsByHour(self):
        query = "SELECT  EXTRACT(HOUR FROM datetime),COUNT(EXTRACT(HOUR FROM
datetime)) FROM posts GROUP BY EXTRACT(HOUR FROM datetime)"
        connection = pymysql.connect(self.host, self.user, self.password, self.db, use_unicode = True,
charset = "utf8")
        cursor = connection.cursor()

        try:
            cursor.execute(query)
            return cursor.fetchall()
        except:
            print("Error selecting posts from database")
            connection.rollback()
        finally:
            connection.close()
            cursor.close()

    def selectPosts(self):
        query = "Select * from posts"

```

```
connection = pymysql.connect(self.host, self.user, self.password, self.db, use_unicode = True,
charset = "utf8")
cursor = connection.cursor()

try:
    cursor.execute(query)
    return cursor.fetchall()
except:
    print("Error selecting posts from database")
    connection.rollback()
finally:
    connection.close()
    cursor.close()
```