**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ**
**ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ**
**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΜΕ ΕΦΑΡΜΟΓΕΣ ΣΤΗ ΒΙΟΙΑΤΡΙΚΗ**

# Συλλογή και ανάλυση διαχεόμενου περιεχομένου λογαριασμών Κοινωνικών Δικτύων που ανήκουν στους ίδιους χρήστες

**Γεωργιλάς Στυλιανός**

**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**
**Υπεύθυνος**
**Ραζής Γεράσιμος**

**Λαμία, 2021**


**ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ**

**ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ**

**ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΜΕ ΕΦΑΡΜΟΓΕΣ ΣΤΗ ΒΙΟΙΑΤΡΙΚΗ**


# Συλλογή και ανάλυση διαχεόμενου περιεχομένου λογαριασμών Κοινωνικών Δικτύων που ανήκουν στους ίδιους χρήστες


**Γεωργιλάς Στυλιανός**


**ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ**

**Επιβλέπων**
**Ραζής Γεράσιμος**

**Λαμία, 2021**

Με ατομική μου ευθύνη και γνωρίζοντας τις κυρώσεις [(1)], που προβλέπονται από της διατάξεις της παρ. 6 του άρθρου 22 του Ν. 1599/1986, δηλώνω ότι:

1. *Δεν παραθέτω κομμάτια βιβλίων ή άρθρων ή εργασιών άλλων αυτολεξεί **χωρίς να τα περικλείω σε εισαγωγικά** και χωρίς να αναφέρω το συγγραφέα, τη χρονολογία, τη σελίδα. Η αυτολεξεί παράθεση χωρίς εισαγωγικά χωρίς αναφορά στην πηγή, είναι λογοκλοπή. Πέραν της αυτολεξεί παράθεσης, λογοκλοπή θεωρείται και η παράφραση εδαφίων από έργα άλλων, συμπεριλαμβανομένων και έργων συμφοιτητών μου, καθώς και η παράθεση στοιχείων που άλλοι συνέλεξαν ή επεξεργάσθηκαν, χωρίς αναφορά στην πηγή. Αναφέρω πάντοτε με πληρότητα την πηγή κάτω από τον πίνακα ή σχέδιο, όπως στα παραθέματα.*

2. *Δέχομαι ότι η αυτολεξεί **παράθεση χωρίς εισαγωγικά**, ακόμα κι αν συνοδεύεται από αναφορά στην πηγή σε κάποιο άλλο σημείο του κειμένου ή στο τέλος του, είναι αντιγραφή. Η αναφορά στην πηγή στο τέλος π.χ. μιας παραγράφου ή μιας σελίδας, δεν δικαιολογεί συρραφή εδαφίων έργου άλλου συγγραφέα, έστω και παραφρασμένων, και παρουσίασή τους ως δική μου εργασία.*

3. *Δέχομαι ότι υπάρχει επίσης περιορισμός στο μέγεθος και στη συχνότητα των παραθεμάτων που μπορώ να εντάξω στην εργασία μου εντός εισαγωγικών. Κάθε μεγάλο παράθεμα (π.χ. σε πίνακα ή πλαίσιο, κλπ.), προϋποθέτει ειδικές ρυθμίσεις, και όταν δημοσιεύεται προϋποθέτει την άδεια του συγγραφέα ή του εκδότη. Το ίδιο και οι πίνακες και τα σχέδια*

4. *Δέχομαι όλες τις συνέπειες σε περίπτωση λογοκλοπής ή αντιγραφής.*

Ημερομηνία: 29 / 09 / 2021

Ο Δηλών.

(Υπογραφή)

(1) «Όποιος εν γνώσει του δηλώνει ψευδή γεγονότα ή αρνείται ή αποκρύπτει τα αληθινά με έγγραφη υπεύθυνη δήλωση του άρθρου 8 παρ. 4 Ν. 1599/1986 τιμωρείται με φυλάκιση τουλάχιστον τριών μηνών. Εάν ο υπαίτιος αυτών των πράξεων σκόπευε να προσπορίσει στον εαυτόν του ή σε άλλον περιουσιακό όφελος βλάπτοντας τρίτον ή σκόπευε να βλάψει άλλον, τιμωρείται με κάθειρξη μέχρι 10 ετών.

# Συλλογή και ανάλυση διαχεόμενου περιεχομένου λογαριασμών Κοινωνικών Δικτύων που ανήκουν στους ίδιους χρήστες


## Γεωργιλάς Στυλιανός


**Τριμελής Επιτροπή:**

Ραζής Γεράσιμος, Επιβλέπων

Αναγνωστόπουλος Ιωάννης, Καθηγητής

Τασουλής Σωτήριος, Επίκουρος Καθηγητής

# Περίληψη

Στον κόσμο του διαδικτύου υπάρχουν πολλά μέσα δικτύωσης και κοινωνικοποίησης, με τρία πολύ δημοφιλή να είναι το Twitter, το Facebook και το Instagram. Ο χρήστης χρειάζεται να έχει λογαριασμούς σε παραπάνω από ένα Κοινωνικό Δίκτυο, ώστε να έχει μία πιο ολοκληρωμένη εμπειρία. Αυτό οφείλεται στο ότι το κάθε Κοινωνικό Δίκτυο έχει διαφορετική φιλοσοφία ως προς τον τρόπο λειτουργίας του, το περιεχόμενό του καθώς και την εμπειρία που επιθυμεί να βιώνει ο χρήστης του. Ένα ερώτημα που γεννάται είναι αυτό των διαφορών συμπεριφοράς του χρήστη σε κάθε Κοινωνικό Δίκτυο και των διαφορών του περιεχομένου που ο ίδιος αναρτά σε αυτά.

Αυτό το ερώτημα θα διερευνηθεί στο πλαίσιο της παρούσας πτυχιακής εργασίας. Απαραίτητο κριτήριο της επιλογής των χρηστών για την μελέτη αυτή είναι να έχουν λογαριασμούς και στα τρία Κοινωνικά Δίκτυα (Twitter, Facebook, Instagram). Έτσι, οι διαφορές που θα παρατηρηθούν μεταξύ των Δικτύων θα έχουν μεγαλύτερη σημασία και βαρύτητα, καθώς θα προέρχονται από την ίδια ομάδα χρηστών. Η εξόρυξη των δεδομένων από τα τρία Κοινωνικά Δίκτυα έγινε μέσω εφαρμογών που υλοποιήθηκαν με τη γλώσσα Python. Τα δεδομένα, τα οποία περιλαμβάνουν οντότητες Κοινωνικών Δικτύων (π.χ. Υπερσυνδέσμους (Links), Πολυμέσα (Media) και Hashtags) και δημοσιεύσεις με ενσωματωμένες πληροφορίες τους (π.χ. Likes, Comments , Shares) αποθηκεύτηκαν σε μία σχεσιακή βάση δεδομένων MySQL ώστε να είναι διαθέσιμα για την περαιτέρω ανάλυσή τους. Τα προγράμματα εξαγωγής δεδομένων καθώς και η Βάση Δεδομένων είναι διαθέσιμα online σε όλους στο αποθετήριο μας στο GitHub[1].

Οι αναλύσεις που πραγματοποιήθηκαν χωρίζονται σε δύο ευρείες κατηγορίες: α) στο σύνολο των χρηστών που αναφέρθηκαν προηγουμένως, καθώς και β) σε διαφορετικές ομάδες τους, με βάση την κοινωνική τους επιρροή. Οι αναλύσεις αυτές εμφάνισαν την ύπαρξη μοτίβων συμπεριφοράς χρηστών ανάλογα με το Κοινωνικό Δίκτυο και την επιρροή που ασκούν. Συγκεκριμένα, μέσω της πρώτης κατηγορίας αναλύσεων δημιουργήθηκε μία κατάταξη των τριών Κοινωνικών Δικτύων αναφορικά με την τάση χρήσης τους και την χρήση των μεταδεδομένων που παρέχουν, καθώς και αναλύσεις αναφορικά με τα μοτίβα χρήσης αυτών των μεταδεδομένων. Αναφορικά με την δεύτερη κατηγορία αναλύσεων παρατηρήσαμε πως κάθε ομάδα χρηστών προτιμάει να χρησιμοποιεί διαφορετικό Κοινωνικό Δίκτυο σε σχέση με τις υπόλοιπες ομάδες. Επίσης μέσω των δεδομένων υπολογίστηκαν οι δείκτες δραστηριότητας των χρηστών. Συγκεκριμένα, παρατηρούνται παρόμοια μοτίβα συμπεριφοράς των χρηστών μεταξύ των δικτύων. Παρόλα αυτά, οι χρήστες που ασκούν την μεγαλύτερη Κοινωνική Επιρροή εμφανίζουν την μεγαλύτερη ασυνέπεια χρήσης μεταξύ των δικτύων.

**Λέξεις-κλειδιά**: Κοινωνικά Δίκτυα, Εξόρυξη Δεδομένων, Python, MySQL, Κοινωνική Επιρροή, Ανάλυση Δεδομένων

[1] https://github.com/SGeorgilas/Content-analysis-coming-from-social-media-accounts-of-specific-users

# Abstract

There are many means of networking and socialization in the internet world, with three very popular Social Networks being Twitter, Facebook, and Instagram. The user needs to have accounts in more than one Social Network in order to have a more complete experience. This is due to the fact that each Social Network has a different philosophy in terms of how it works, its content and the user experience that it provides. One question that arises is that of differences in user behavior on each Social Network and of differences in the content the user disseminates.

This question will be investigated in this dissertation. A necessary criterion for the selection of users for this study is to have accounts in all three Social Networks (Twitter, Facebook, and Instagram). Thus, the differences that will be observed between the Networks will be more important and significant, as they will come from the same group of users. The data from the three Social Networks was extracted through applications implemented in the Python language. The data, which includes Social Networking entities (e.g., Links, Media, and Hashtags) and posts with embedded information (e.g. Likes, Comments, Shares) is stored in a relational MySQL relational database to be available for further analysis. The implemented data export software as well as a Database export are available online for further use by the research community in our GitHub repository[1].

The analyses performed are divided into two broad categories: a) to all the users mentioned above, and b) to their different groups, based on their social influence. These analyses revealed the existence of user behavior patterns depending on the Social Network and the influence they exert. Specifically, through the first category of analyses, a ranking of the three Social Networks was created regarding their tendency to use and the use of the metadata they provide. Regarding the second category of analyses, we observed that each group of users prefers to use a different Social Network in relation to the other groups. Users' activity indicators were also calculated through the data. Similar patterns of user behavior are observed between networks. However, users with the greatest social influence exhibit the greatest inconsistency of use between networks.

**Keywords**: Social Media, Data Mining, Python, MySQL, Social Influence, Data Analysis

# Περιεχόμενα

# Πίνακας Περιεχομένων

# Ενότητα 1. Εισαγωγή

Το διαδίκτυο και τα Κοινωνικά Δίκτυα (ΚΔ) πλέον αποτελούν ένα αναπόσπαστο κομμάτι της καθημερινότητας των ανθρώπων. Μέσω των ΚΔ οι άνθρωποι μοιράζονται τις εμπειρίες τους και τις απόψεις τους με τον υπόλοιπο κόσμο. Κάθε νέο συμβάν της παγκόσμιας επικαιρότητας δημοσιεύεται και διαμοιράζεται σε όλο τον κόσμο σε ελάχιστο χρόνο. Η έμφυτη τάση του ανθρώπου να λαμβάνει μέρος σε οποιαδήποτε μορφή κοινωνικοποίησης δικαιολογεί το στατιστικό [2] το οποίο αναφέρει πως το 2021 περίπου 4,3 δισεκατομμύρια άνθρωποι παγκοσμίως είναι χρήστες ΚΔ. Επίσης για την ίδια χρονιά αναφέρεται[2] πως ένας χρήστης ηλικίας 18-34 ετών έχει στην κατοχή του κατά μέσο όρο 8,5 λογαριασμούς σε όλα τα ΚΔ. Αυτή η δικτύωση σε πολλαπλά κοινωνικά μέσα αποδίδεται στην εξειδίκευση των μεμονωμένων πλατφορμών. Οι ιστότοποί τους έχουν αναβαθμιστεί, διευρύνοντας την ποικιλία και την επιλογή του διαθέσιμου προσφερόμενου περιεχομένου.

Αυτό το φαινόμενο έχει αποτελέσει αφορμή πολλών ερευνών ανά τα χρόνια. Κάποιες από τις έρευνες αυτές αναφέρονται στην *"Ενότητα 2. Σχετική Βιβλιογραφία"* παρακάτω. Μερικά σημαντικά σημεία είναι η ένωση των λογαριασμών ενός χρήστη, η δημιουργία οντοτήτων, διάφοροι τρόποι ανάκτησης και επεξεργασίας των δεδομένων των χρηστών καθώς και η Κοινωνική Επιρροή (ΚΕ) των χρηστών η οποία αποτελεί σημαντικό παράγοντα για την συμπεριφορά τους.

Οι συνεισφορές της παρούσης πτυχιακής εργασίας είναι οι εξής:

1. Συλλογή στοιχείων ΚΔ ιδίων χρηστών από πολλαπλά δίκτυα
2. Ανάλυση μοτίβων συμπεριφορών ανά ΚΔ
3. Ανάλυση μοτίβων συμπεριφορών ανά κοινωνική οντότητα
4. Μελέτη χρηστών ανά Κοινωνική Επιρροή (ΚΕ) και σύγκριση μοτίβων συμπεριφοράς
5. Αποθήκευση ΒΔ και προγραμμάτων σε Github[1] αποθετήριο για ελεύθερη χρήση

Για τις ανάγκες των αναλύσεων, εξαρχής επιλέχθηκε μία ομάδα χρηστών με κριτήριο την ύπαρξη τους και στα τρία ΚΔ. Μελέτη συμπεριφοράς χρηστών σε τρία διαφορετικά ΚΔ καθώς και μελέτη συμπεριφοράς ομάδων χρηστών με βάση την ΚΕ δεν έχει ξαναγίνει σε τέτοιο βαθμό και αυτή είναι η πρωτοπορία και συνεισφορά της εργασίας αυτής.

Το υπόλοιπο του εγγράφου οργανώνεται ως εξής. Στην *"Ενότητα 2. Σχετική Βιβλιογραφία"* περιγράφεται η σχετική βιβλιογραφία η οποία μελετήθηκε πριν την δημιουργία της εργασίας αυτής. Στην *"Ενότητα 3. Βάση Δεδομένων"* γίνεται η ανάλυση της Βάσης Δεδομένων (ΒΔ) που χρησιμοποιήθηκε. Συγκεκριμένα, περιγράφονται αναλυτικά οι πίνακες της ΒΔ καθώς και οι

[2] https://www.statista.com/topics/1164/social-networks/

συσχετίσεις μεταξύ αυτών. Στην "*Ενότητα 4. Ανάλυση backend προγράμματος*" περιγράφονται τα προγράμματα εξαγωγής των δεδομένων από τα τρία ΚΔ. Στην "*Ενότητα 5. Αναλύσεις και αποτελέσματα"* παρατίθενται οι αναλύσεις που πραγματοποιήθηκαν και περιγράφονται τα αποτελέσματά τους. Η "*Ενότητα 6. Συμπεράσματα και μελλοντικές επεκτάσεις"* παρέχει τα συμπεράσματα της εργασίας μας, καθώς και τις εκτιμήσεις μας για μελλοντικές κατευθύνσεις. Τέλος στην "*Ενότητα 7. Βιβλιογραφία"* υπάρχει η βιβλιογραφία που αναλύθηκε στην "*Ενότητα 2. Σχετική Βιβλιογραφία".*

# Ενότητα 2. Σχετική Βιβλιογραφία

Σε αυτή την ενότητα θα παρουσιαστούν οι εργασίες οι οποίες μελετήθηκαν κατά τη δημιουργία της πτυχιακής εργασίας. Θα αναφερθούν οι στόχοι των δημοσιεύσεων καθώς και τα σημεία αυτών τα οποία εμείς επιλέξαμε να χρησιμοποιήσουμε ως θεωρητικές βάσεις της δικιάς μας εργασίας.

Οι συγγραφείς του [1] αναφέρουν την συνένωση των λογαριασμών ενός χρήστη για την δημιουργία ψηφιακού αποτυπώματός του ΚΔ. Τα ψηφιακά αποτυπώματα βοηθούν στην εξατομίκευση ενός χρήστη, στη διαχείριση των προφίλ του, καθώς και για την ευκολότερη ανίχνευση κακόβουλης συμπεριφοράς χρηστών ενός ΚΔ.

Η εργασία [2] πραγματεύεται την ενσωμάτωση δεδομένων ΚΔ και την δημιουργία νέων κοινωνικών οντοτήτων. Παρόλο που ο σκοπός τους είναι η προώθηση εφαρμογών επιχειρήσεων και μάρκετινγκ, η δημιουργία κοινωνικών οντοτήτων είναι μια έννοια την οποία ασπαστήκαμε και εμείς στην δικιά μας εργασία.

Ο σκοπός της [3] είναι να αναπτυχθεί ένα μοντέλο βάσης γνώσεων ενός συστήματος πληροφοριών που συλλέγει πληροφορίες από διάφορα ΚΔ. Το μοντέλο αυτό είναι σημαντικό για εμάς, καθώς οι πληροφορίες που συλλέγει είναι παρόμοιες με τις πληροφορίες που επιλέξαμε εμείς να συλλέξουμε. Το έργο παρουσιάζει ένα οντολογικό μοντέλο για την ενοποίηση των προφίλ δεδομένων διαφορετικών ΚΔ. Επιπλέον, προτείνεται μια προσέγγιση για την ανάκτηση πληροφοριών με τη χρήση συνταγματικών προτύπων στο σχηματισμό ενός δέντρου βάσης δεδομένων για αναρτήσεις χρηστών ενός ΚΔ.

Το πρόβλημα της προετοιμασίας δεδομένων ΚΔ καλής ποιότητας για ανάλυση δεδομένων και εξόρυξη τους μελετάται στην εργασία [4]. Πραγματοποιείται διερευνητική ανάλυση δειγμάτων δεδομένων που αποκτήθηκαν από API ροής ΚΔ για να κατανοηθεί η αντιπροσωπευτικότητα των δειγμάτων στα αντίστοιχα πλήρη δεδομένα και την δυνατότητά τους για χρήση σε διάφορες εργασίες εξόρυξης δεδομένων. Η σωστή εξόρυξη δεδομένων καθώς και η καλή ποιότητά τους είναι βασικές προϋποθέσεις της πτυχιακής μας και για να τις εκπληρώσουμε βασιστήκαμε σε μεγάλο βαθμό σε αυτή την εργασία.

Στόχος της [5] είναι να αναπτυχθεί ένα πρακτικό πλαίσιο για τη λήψη ενός ομοιόμορφου δείγματος χρηστών σε ένα διαδικτυακό ΚΔ ανιχνεύοντας το κοινωνικό του γράφημα. Για τη λήψη του δικού μας δείγματος χρηστών χρησιμοποιήσαμε πληροφορίες που αντλήσαμε από την εργασία αυτή.

Μια πρόταση που εμφανίζεται στην [6] είναι αυτή ενός ενοποιημένου σημασιολογικού μοντέλου για ανάλυση συμβάντων. Το μοντέλο περιέχει καλά σχεδιασμένες οντολογικές κλάσεις και ιδιότητες για την αντιμετώπιση της έλλειψης ενοποιημένης αναπαράστασης, η

οποία είναι και στόχος της δικιάς μας εργασίας. Λαμβάνονται επίσης υπόψη οι πληροφορίες προέλευσης. Οι μέθοδοι χαρτογράφησης και μετατροπής δεδομένων παρέχονται για τη διαχείριση των ετερογενών δεδομένων από διάφορες διαδικτυακές πλατφόρμες ΚΔ και σύνολα δεδομένων.

Στην εργασία [7] παρουσιάζεται μια μελέτη της τομής των δεδομένων FOAF (Friend of a Friend) που βρέθηκαν σε πολλά ΚΔ. Χρησιμοποιώντας την οντολογία FOAF και εφαρμόζοντας τις τεχνικές συλλογιστικής (reasoning) του Σημασιολογικού Ιστού, δείχνεται ότι ένα σημαντικό ποσοστό προφίλ μπορεί να συγχωνευτεί από πολλά δίκτυα. Παρουσιάζονται αποτελέσματα για το πώς αυτό επηρεάζει τη δομή του δικτύου και εκθειάζεται η επιτυχία του Σημασιολογικού Ιστού.

Οι παραπάνω δημοσιεύσεις περιέχουν ερευνητικά αντικείμενα, με τα οποία εμείς δημιουργήσαμε τις βάσεις της δικής μας εργασίας. Η εργασία [5] αναφέρει την δημιουργία ενός πλαισίου λήψης ομοιόμορφου δείγματος χρηστών. Στην εργασία μας αυτό είναι δεδομένο εξαρχής, καθώς είχαμε από πριν μια προκαθορισμένη ομάδα χρηστών από την οποία αντλήσαμε τα δεδομένα μας και πάνω σε αυτά πραγματοποιήσαμε τις αναλύσεις μας. Επίσης, η συνένωση λογαριασμών και η δημιουργία οντοτήτων θεωρείται στόχος των εργασιών [1] και [2], ενώ εμείς δημιουργώντας από πριν τις κοινωνικές οντότητες, πραγματοποιήσαμε τις αναλύσεις μας πάνω σε αυτές και βγάλαμε συμπεράσματα που αφορούν την συμπεριφορά τους και το περιεχόμενο που αναρτούν.

# Ενότητα 3. Βάση Δεδομένων

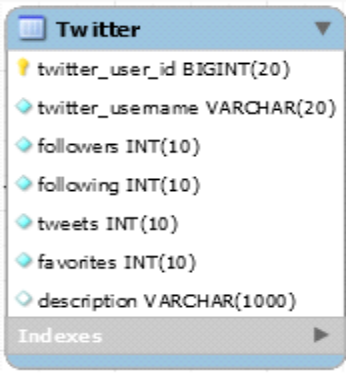
Στην ενότητα αυτή περιγράφονται σχεδιαστικά ζητήματα αναφορικά με την επιλεγμένη προσέγγιση αποθήκευσης των δεδομένων των ΚΔ. Συγκεκριμένα, χρησιμοποιήθηκε μία σχεσιακή βάση δεδομένων MySQL. Στις παρακάτω υποενότητες θα αναλυθούν οι πίνακες στους οποίους αποθηκεύονται τα δεδομένα, οι συσχετίσεις που επικρατούν μεταξύ των πινάκων καθώς και θα αναφερθούν τα κριτήρια σχεδιασμού της βάσης δεδομένων.

## 3.1. Ανάλυση Πινάκων

Στη Βάση Δεδομένων (ΒΔ) μας, για την αποθήκευση πληροφοριών σχετικά με τους λογαριασμούς των χρηστών, υπάρχουν τρεις πίνακες, ο πίνακας “Twitter”, ο “Facebook” και ο “Instagram”.

Ο πίνακας “Twitter” (*Εικόνα 1*) περιέχει κάποια βασικά χαρακτηριστικά των προφίλ των χρηστών στο Twitter και αναλύεται ως εξής :

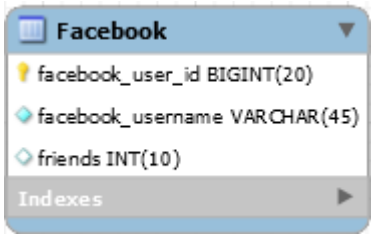
- twitter_user_id: Το κύριο κλειδί του πίνακα, είναι ο αναγνωριστικός κωδικός (ID) που δίνεται από το Twitter
- twitter_username: Το ψευδώνυμο που χρησιμοποιεί ο χρήστης στο Twitter
- followers: Ο αριθμός των ακολούθων του χρήστη
- following: Ο αριθμός των χρηστών που ακολουθεί ο χρήστης
- tweets: Ο αριθμός των δημοσιεύσεων που έχουν δημιουργηθεί από τον χρήστη
- favorites: Ο αριθμός των likes που έχει μαζέψει συνολικά ο χρήστης στις δημοσιεύσεις του
- description: Μια περιγραφή που έχει ορίσει ο χρήστης για το προφίλ του




***Εικόνα 1: Πίνακας Twitter***

Ο πίνακας “Facebook” (*Εικόνα 2*) περιέχει κάποια βασικά χαρακτηριστικά των προφίλ των χρηστών στο Facebook και αναλύεται ως εξής:

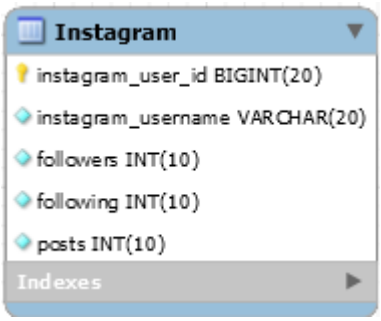
- facebook_user_id: Το κύριο κλειδί του πίνακα, είναι ο αναγνωριστικός κωδικός που δίνεται από το Facebook
- facebook_username: Το ψευδώνυμο που χρησιμοποιεί ο χρήστης στο Facebook
- friends: Ο αριθμός των φίλων που έχει ο χρήστης




***Εικόνα 2: Πίνακας Facebook***

Ο πίνακας “Instagram” (*Εικόνα 3*) περιέχει κάποια βασικά χαρακτηριστικά των προφίλ των χρηστών στο Instagram και αναλύεται ως εξής:

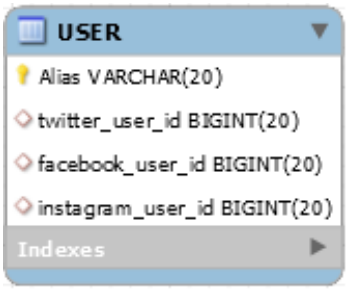
- instagram_user_id: Το κύριο κλειδί του πίνακα, είναι ο αναγνωριστικός κωδικός που δίνεται από το Instagram
- instagram_username: Το ψευδώνυμο που χρησιμοποιεί ο χρήστης στο Instagram
- followers: Ο αριθμός των ακολούθων του χρήστη
- following: Ο αριθμός των χρηστών που ακολουθεί ο χρήστης
- posts: Ο αριθμός των δημοσιεύσεων που έχουν δημιουργηθεί από τον χρήστη




***Εικόνα 3: Πίνακας Instagram***

Ο πίνακας “User” (*Εικόνα 4*) χρησιμοποιείται ως συνδετικός πίνακας των τριών προηγούμενων πινάκων και αναλύεται ως εξής:

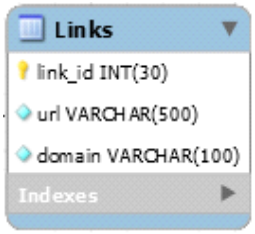
- Alias: Το κύριο κλειδί του πίνακα, ένα ψευδώνυμο που έχουμε δώσει εμείς στον χρήστη, ώστε να αναφερόμαστε με αυτό στην οντότητα του χρήστη από εδώ και στο εξής
- twitter_user_id: Ο αναγνωριστικός κωδικός που δίνεται από το Twitter στον χρήστη, ξένο κλειδί για τον πίνακα *“Twitter” (Εικόνα 1)*
- facebook_user_id: Ο αναγνωριστικός κωδικός που δίνεται από το Facebook στον χρήστη, ξένο κλειδί για τον πίνακα “Facebook” (*Εικόνα 2*)
- instagram_user_id: Ο αναγνωριστικός κωδικός που δίνεται από το Instagram στον χρήστη, ξένο κλειδί για τον πίνακα “Instagram” (*Εικόνα 3*)




***Εικόνα 4: Πίνακας User***

Ο πίνακας “Links” (*Εικόνα 5*) περιέχει τους υπερσυνδέσμους που βρίσκονται στις δημοσιεύσεις των χρηστών και αναλύεται ως εξής:

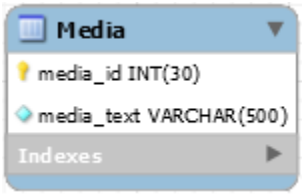
- link_id: Το κύριο κλειδί του πίνακα, ένας μοναδικός αριθμός, αύξων για κάθε εγγραφή
- url: Η διεύθυνση του υπερσυνδέσμου
- domain: Τομέας των διεθνών πόρων του διαδικτύου




***Εικόνα 5: Πίνακας Links***

Ο πίνακας “Media” (*Εικόνα 6*) περιέχει τα πολυμέσα (Εικόνες, Βίντεο) που βρίσκονται στις δημοσιεύσεις των χρηστών και αναλύεται ως εξής:

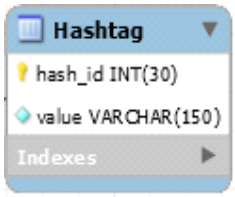
- media_id: Το κύριο κλειδί του πίνακα, ένας μοναδικός αριθμός, αύξων για κάθε εγγραφή
- media_text: Η διεύθυνση του πολυμέσου




***Εικόνα 6: Πίνακας Media***

Ο πίνακας “Hashtag” (*Εικόνα 7*) περιέχει τα hashtags που βρίσκονται στις δημοσιεύσεις των χρηστών και αναλύεται ως εξής:

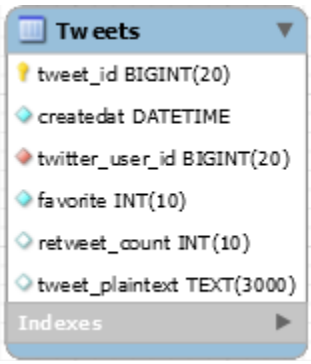
- hash_id: Το κύριο κλειδί του πίνακα, ένας μοναδικός αριθμός, αύξων για κάθε εγγραφή
- value: Το κείμενο που υπάρχει στο hashtag




***Εικόνα 7: Πίνακας Hashtag***

Ο πίνακας “Tweets” (*Εικόνα 8*) περιέχει τις δημοσιεύσεις των χρηστών στο Twitter και αναλύεται ως εξής:

- tweet_id: Το κύριο κλειδί του πίνακα, ο αναγνωριστικός κωδικός που δίνεται στη δημοσίευση από το Twitter
- createdat: Η ημερομηνία που δημιουργήθηκε η δημοσίευση
- twitter_user_id: Ο αναγνωριστικός κωδικός που δίνεται από το Twitter στον χρήστη, ξένο κλειδί για τον πίνακα “Twitter”(*Εικόνα 1*)
- favorite: Ο αριθμός των likes που έχει συλλέξει η δημοσίευση
- retweet_count: Ο αριθμός των shares που έχει η δημοσίευση
- tweet_plaintext: Το κείμενο που υπάρχει στη δημοσίευση




***Εικόνα 8: Πίνακας Tweets***

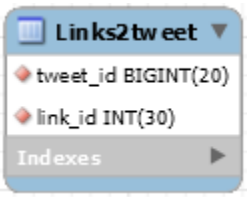
Ο πίνακας "TwitterUserMentions" (*Εικόνα 9*) περιέχει τις αναφορές σε χρήστες που υπάρχουν στις δημοσιεύσεις και αναλύεται ως εξής:

- tweet_id: Ο αναγνωριστικός κωδικός της δημοσίευσης στην οποία έγινε η αναφορά, ξένο κλειδί για τον πίνακα "Tweets" (*Εικόνα 8*)
- mention_name: Το όνομα του χρήστη που έχει αναφερθεί


***Εικόνα 9: Πίνακας TwitterUserMentions***

Ο πίνακας "Links2tweet" (*Εικόνα 10*) συνδέει τους πίνακες "Links" και "Tweets" και αναλύεται ως εξής:

- tweet_id: Ο αναγνωριστικός κωδικός της δημοσίευσης που υπάρχει ο υπερσύνδεσμος, ξένο κλειδί για τον πίνακα "Tweets" (*Εικόνα 8*)
- link_id: Ο μοναδικός αύξων αριθμός του υπερσυνδέσμου που υπάρχει στον πίνακα Links, ξένο κλειδί για τον πίνακα "Links" (*Εικόνα 5*)




***Εικόνα 10: Πίνακας Links2tweet***

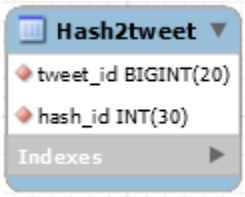
Ο πίνακας “Media2tweet” (*Εικόνα 11*) συνδέει τους πίνακες “Media” (*Εικόνα 6*) και “Tweets” (*Εικόνα 8*) και αναλύεται ως εξής:

- tweet_id: Ο αναγνωριστικός κωδικός της δημοσίευσης που υπάρχει το πολυμέσο, ξένο κλειδί για τον πίνακα “Tweets” (*Εικόνα 8*)
- media_id: Ο μοναδικός αύξων αριθμός του πολυμέσου που υπάρχει στον πίνακα “Media”, ξένο κλειδί για τον πίνακα “Media” (*Εικόνα 6*)


***Εικόνα 11: Πίνακας Media2tweet***

Ο πίνακας “Hash2tweet” (*Εικόνα 12*) συνδέει τους πίνακες “Hashtag” (*Εικόνα 7*) και “Tweets” (*Εικόνα 8*) και αναλύεται ως εξής:

- tweet_id: Ο αναγνωριστικός κωδικός της δημοσίευσης που υπάρχει το hashtag, ξένο κλειδί για τον πίνακα “Tweets” (*Εικόνα 8*)
- hash_id: Ο μοναδικός αύξων αριθμός του hashtag που υπάρχει στον πίνακα Hashtag, ξένο κλειδί για τον πίνακα “Hashtag” (*Εικόνα 7*)

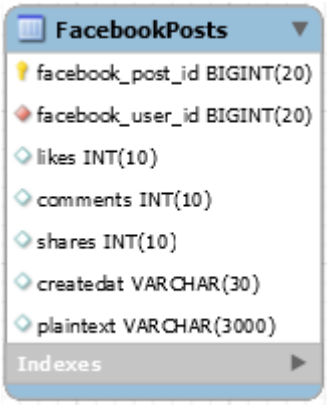



***Εικόνα 12: Πίνακας Hash2tweet***

Ο πίνακας “FacebookPosts” (*Εικόνα 13*) περιέχει τις δημοσιεύσεις των χρηστών στο Facebook και αναλύεται ως εξής:

- facebook_post_id: Το κύριο κλειδί του πίνακα, ο αναγνωριστικός κωδικός που δίνεται στη δημοσίευση από το Facebook
- facebook_user_id: Ο αναγνωριστικός κωδικός που δίνεται από το Facebook στον χρήστη, ξένο κλειδί για τον πίνακα “Facebook” (*Εικόνα 2*)
- likes: Ο αριθμός των likes που σύλλεξε η δημοσίευση
- comments: Ο αριθμός των σχολίων που έχει η δημοσίευση
- shares: Ο αριθμός των shares που έχει η δημοσίευση

- createdat: Η ημερομηνία δημιουργίας της δημοσίευσης
- plaintext: Το κείμενο που υπάρχει στη δημοσίευση




***Εικόνα 13: Πίνακας Facebook Posts***

Ο πίνακας "Links2Fb" (*Εικόνα 14*) συνδέει τους πίνακες "Links" (*Εικόνα 5*) και "Facebook_Posts" (*Εικόνα 13*) και αναλύεται ως εξής:

- facebook_post_id: Ο αναγνωριστικός κωδικός της δημοσίευσης που υπάρχει ο υπερσύνδεσμος, ξένο κλειδί για τον πίνακα "Facebook" (*Εικόνα 2*)
- link_id: Ο μοναδικός αύξων αριθμός του υπερσυνδέσμου που υπάρχει στον πίνακα Links, ξένο κλειδί για τον πίνακα "Links" (*Εικόνα 5*)


***Εικόνα 14: Πίνακας Links2fb***

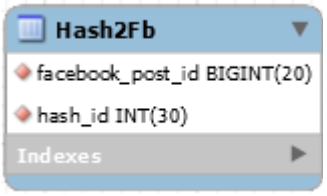
Ο πίνακας "Media2Fb" (*Εικόνα 15*) συνδέει τους πίνακες "Media" και "FacebookPosts" και αναλύεται ως εξής:

- facebook_post_id: Ο αναγνωριστικός κωδικός της δημοσίευσης που υπάρχει το πολυμέσο , ξένο κλειδί για τον πίνακα "Facebook" (*Εικόνα 2*)
- media_id: Ο μοναδικός αύξων αριθμός του πολυμέσου που υπάρχει στον πίνακα "Media", ξένο κλειδί για τον πίνακα "Media" (*Εικόνα 6*)


***Εικόνα 15: Πίνακας Media2fb***

Ο πίνακας "Hash2Fb" (*Εικόνα 16*) συνδέει τους πίνακες "Hashtag" (*Εικόνα 7*) και "Facebook_Posts" (*Εικόνα 13*) και αναλύεται ως εξής:

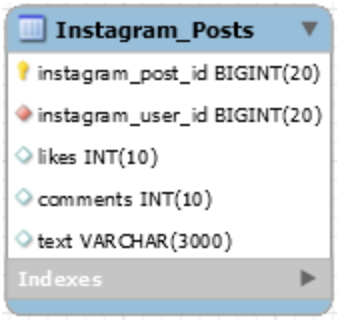
- facebook_post_id: Ο αναγνωριστικός κωδικός της δημοσίευσης που υπάρχει το hashtag, ξένο κλειδί για τον πίνακα "Facebook" (*Εικόνα 2*)
- hash_id: Ο μοναδικός αύξων αριθμός του hashtag που υπάρχει στον πίνακα Hashtag, ξένο κλειδί για τον πίνακα "Hashtag" (*Εικόνα 7*)




***Εικόνα 16: Πίνακας Hash2fb***

Ο πίνακας "Instagram_Posts" (*Εικόνα 17*) περιέχει τις δημοσιεύσεις των χρηστών στο Instagram και αναλύεται ως εξής:

- instagram_post_id: Το κύριο κλειδί του πίνακα, ο αναγνωριστικός κωδικός που δίνεται στη δημοσίευση από το Instagram
- instagram_user_id: Ο αναγνωριστικός κωδικός που δίνεται από το Instagram στον χρήστη, ξένο κλειδί για τον πίνακα "Instagram" (*Εικόνα 3*)
- likes: Ο αριθμός των likes που σύλλεξε η δημοσίευση
- comments: Ο αριθμός των σχολίων που έχει η δημοσίευση
- text: Το κείμενο που υπάρχει στη δημοσίευση




***Εικόνα 17: Πίνακας Instagram_Posts***

Ο πίνακας "Links2Insta" (*Εικόνα 18*) συνδέει τους πίνακες "Links" (*Εικόνα 5*) και "Instagram_Posts" (*Εικόνα 17*) και αναλύεται ως εξής:

- instagram_post_id: Ο αναγνωριστικός κωδικός της δημοσίευσης που υπάρχει ο υπερσύνδεσμος, ξένο κλειδί για τον πίνακα "Instagram" (*Εικόνα 3*)
- link_id: Ο μοναδικός αύξων αριθμός του υπερσυνδέσμου που υπάρχει στον πίνακα Links, ξένο κλειδί για τον πίνακα "Links" (*Εικόνα 5*)


***Εικόνα 18: Πίνακας Links2Insta***

Ο πίνακας "Media2Insta" (*Εικόνα 19*) συνδέει τους πίνακες "Media" (*Εικόνα 6*) και "Instagram_Posts" (*Εικόνα 17*) και αναλύεται ως εξής:

- instagram_post_id: Ο αναγνωριστικός κωδικός της δημοσίευσης που υπάρχει το πολυμέσο, ξένο κλειδί για τον πίνακα "Instagram" (*Εικόνα 3*)
- media_id: Ο μοναδικός αύξων αριθμός του πολυμέσου που υπάρχει στον πίνακα Media, ξένο κλειδί για τον πίνακα "Media" (*Εικόνα 6*)


***Εικόνα 19: Πίνακας Media2Insta***

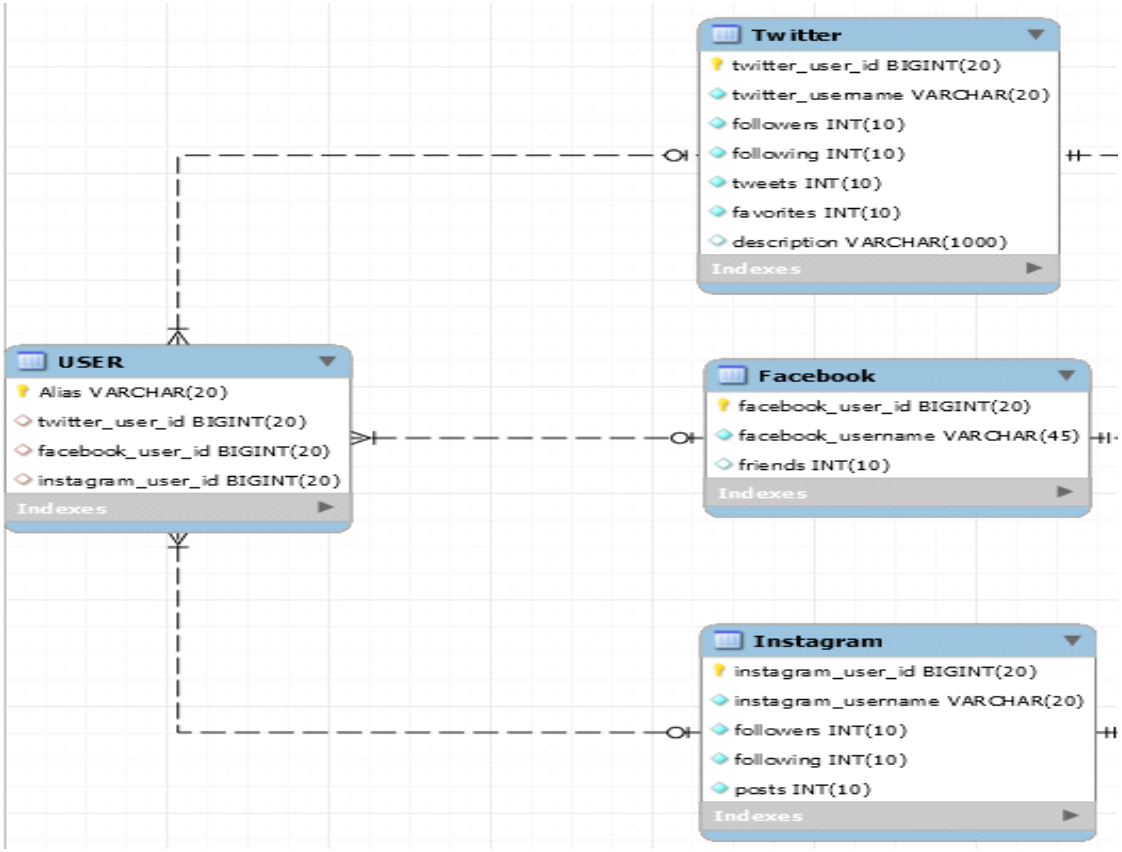
Ο πίνακας "Hash2Insta" (*Εικόνα 20*) συνδέει τους πίνακες "Hashtag" (*Εικόνα 7*) και "Instagram_Posts" (*Εικόνα 17*) και αναλύεται ως εξής:

- instagram_post_id: Ο αναγνωριστικός κωδικός της δημοσίευσης που υπάρχει το hashtag, ξένο κλειδί για τον πίνακα “Instagram” (*Εικόνα 3*)
- hash_id: Ο μοναδικός αύξων αριθμός του hashtag που υπάρχει στον πίνακα Hashtag, ξένο κλειδί για τον πίνακα “Hashtag” (*Εικόνα 7*)


***Εικόνα 20: Πίνακας Hash2Insta***

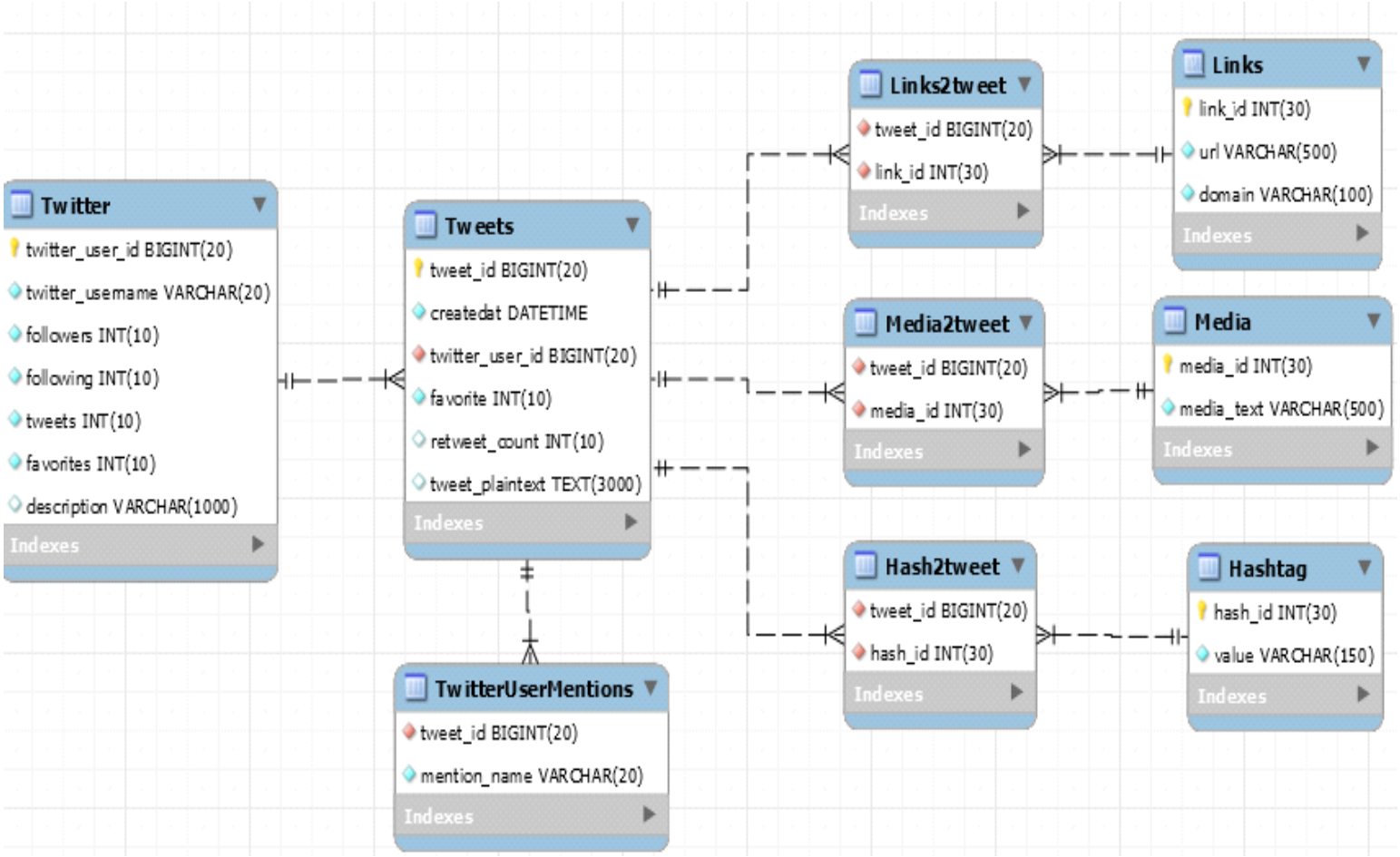
## 3.2. Ανάλυση συσχετίσεων πινάκων

Για να επιτευχθεί η ενοποίηση των τριών ΚΔ και να μπορούμε πλέον να μιλάμε για οντότητες όσον αφορά τους χρήστες, δημιουργήθηκε ο πίνακας "User" (*Εικόνα 4*) ο οποίος περιέχει *τους* αναγνωριστικούς κωδικούς του χρήστη στα τρία διαφορετικά ΚΔ. Μέσω των τριών αναγνωριστικών κωδικών γίνονται και οι σχέσεις μεταξύ των πινάκων, οι οποίες είναι της μορφής "ένα προς πολλά" (*Εικόνα 21*).



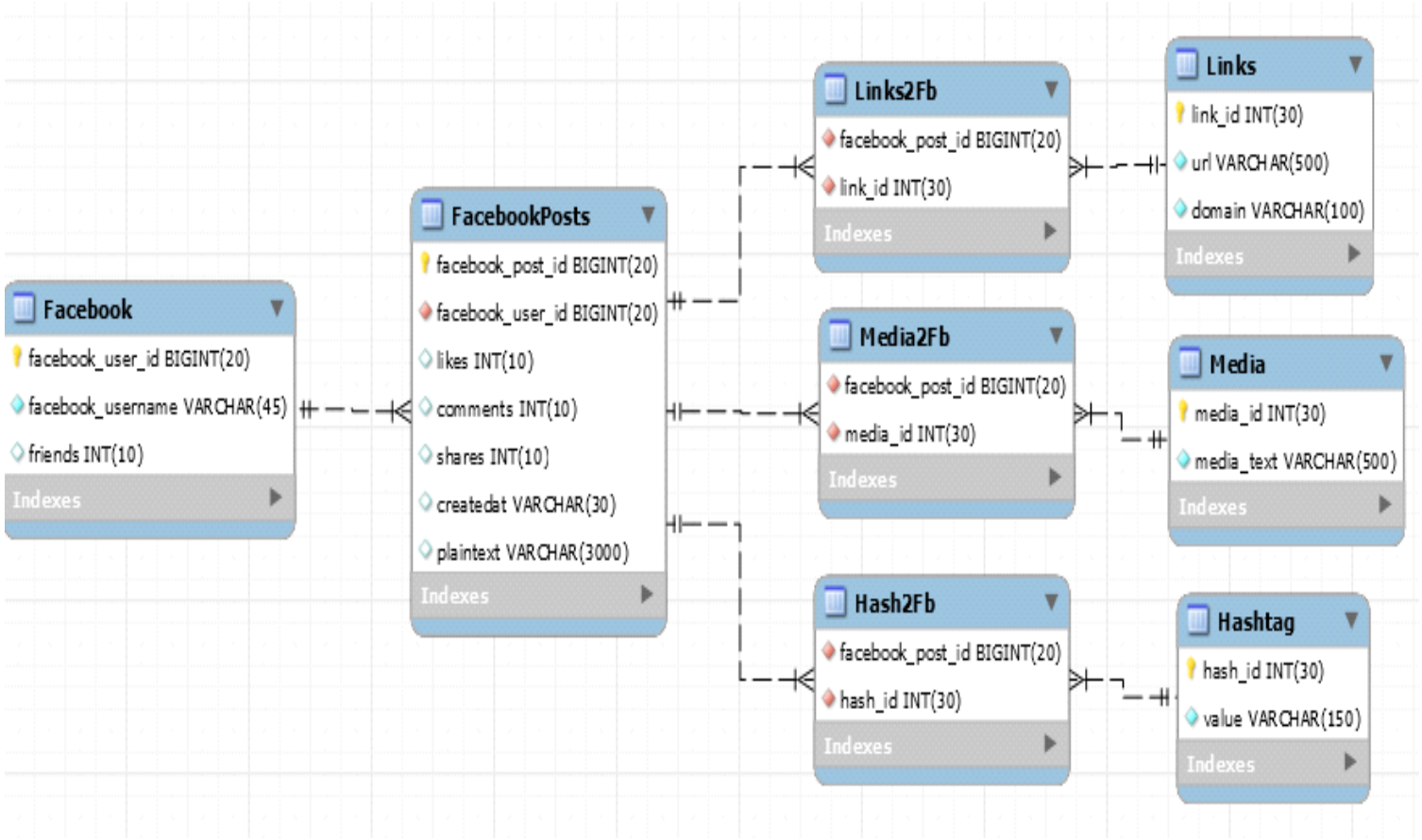

***Εικόνα 21: Σύνδεση User-Twitter-Facebook-Instagram***

*Στην Εικόνα 22 βλέπουμε την σύνδεση του πίνακα "Twitter" (Εικόνα 1) με όλους τους πίνακες σχετικούς με αυτόν.*



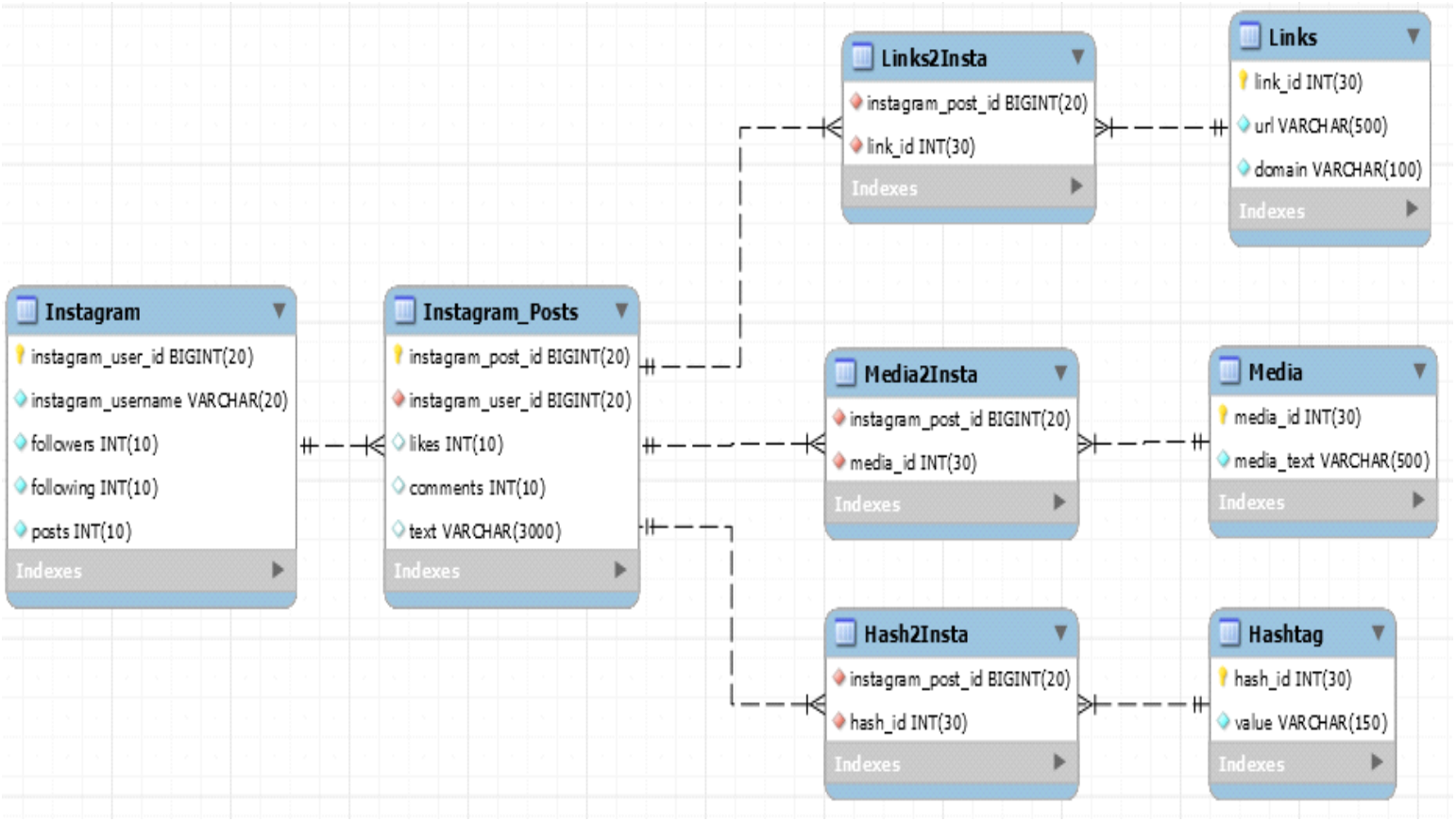

***Εικόνα 22: Πίνακες του κοινωνικού δικτύου Twitter***

Στην *Εικόνα 23* βλέπουμε την σύνδεση του πίνακα "Facebook" (*Εικόνα 2*) με όλους τους πίνακες σχετικούς με αυτόν.




***Εικόνα 23: Πίνακες του κοινωνικού δικτύου Facebook***

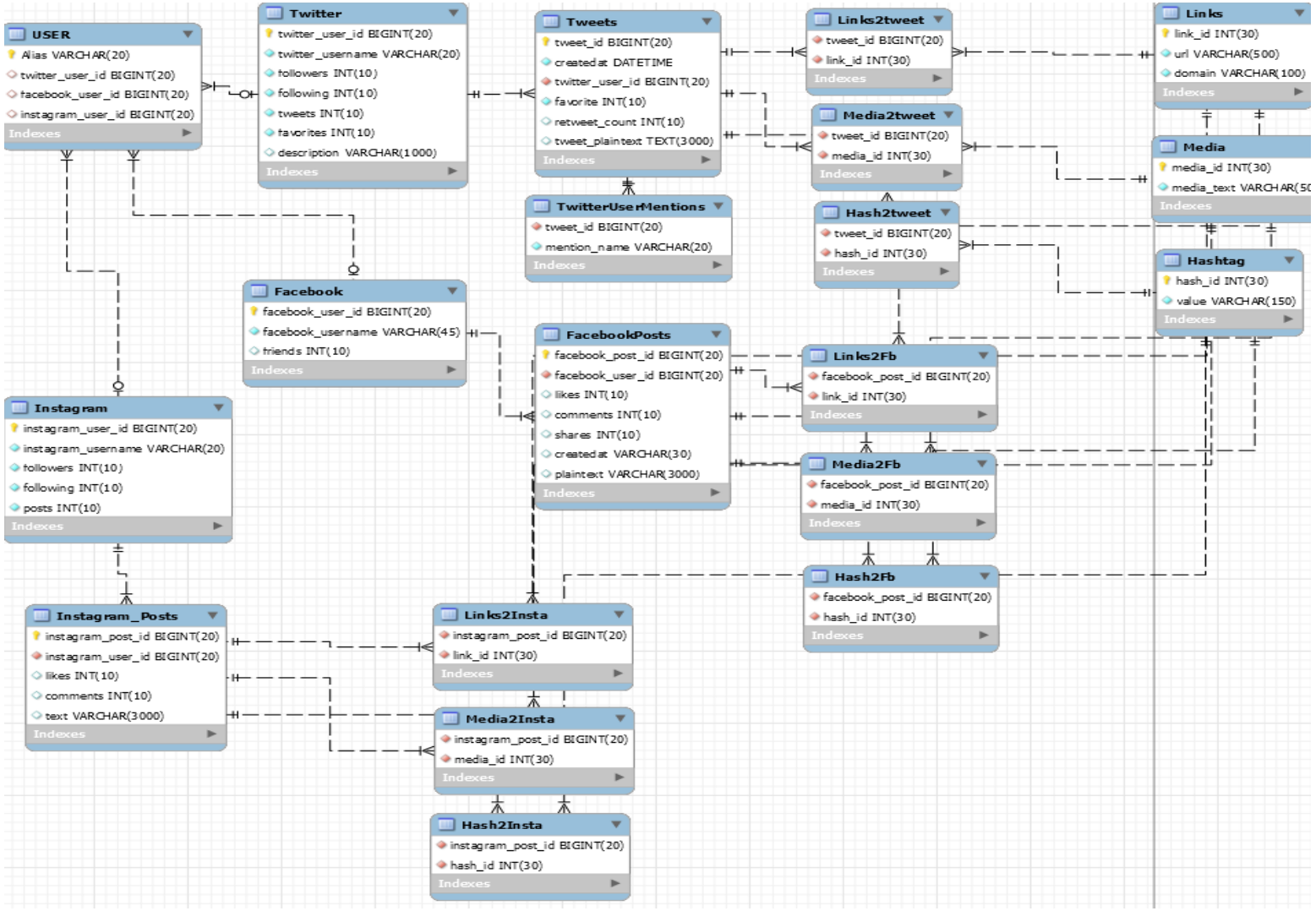
Στην *Εικόνα 24* βλέπουμε την σύνδεση του πίνακα “Instagram” (*Εικόνα 3*) με όλους τους πίνακες σχετικούς με αυτόν.




***Εικόνα 24: Πίνακες του κοινωνικού δικτύου Instagram***

Οι σχέσεις μεταξύ των πινάκων είναι της μορφής “ένα προς πολλά”.

Τέλος, στην *Εικόνα 25* βλέπουμε το διάγραμμα οντοτήτων-συσχετίσεων (Entity-Relation) στο σύνολο του.




***Εικόνα 25: Διάγραμμα Οντοτήτων-Συσχετίσεων***

## 3.3. Κριτήρια σχεδιασμού βάσης δεδομένων

Για τον σχεδιασμό της βάσης δεδομένων υπήρχαν αρκετοί τρόποι διαθέσιμοι. Θα μπορούσαμε για παράδειγμα να δημιουργήσουμε έναν πίνακα που να περιέχει όλα τα προφίλ και από τα τρία ΚΔ και έναν άλλον πίνακα που να περιέχει όλες τις δημοσιεύσεις των ΚΔ. Όμως, αυτός ο σχεδιασμός δημιουργεί κάποια θέματα αρνητικά για την δικιά μας περίπτωση.

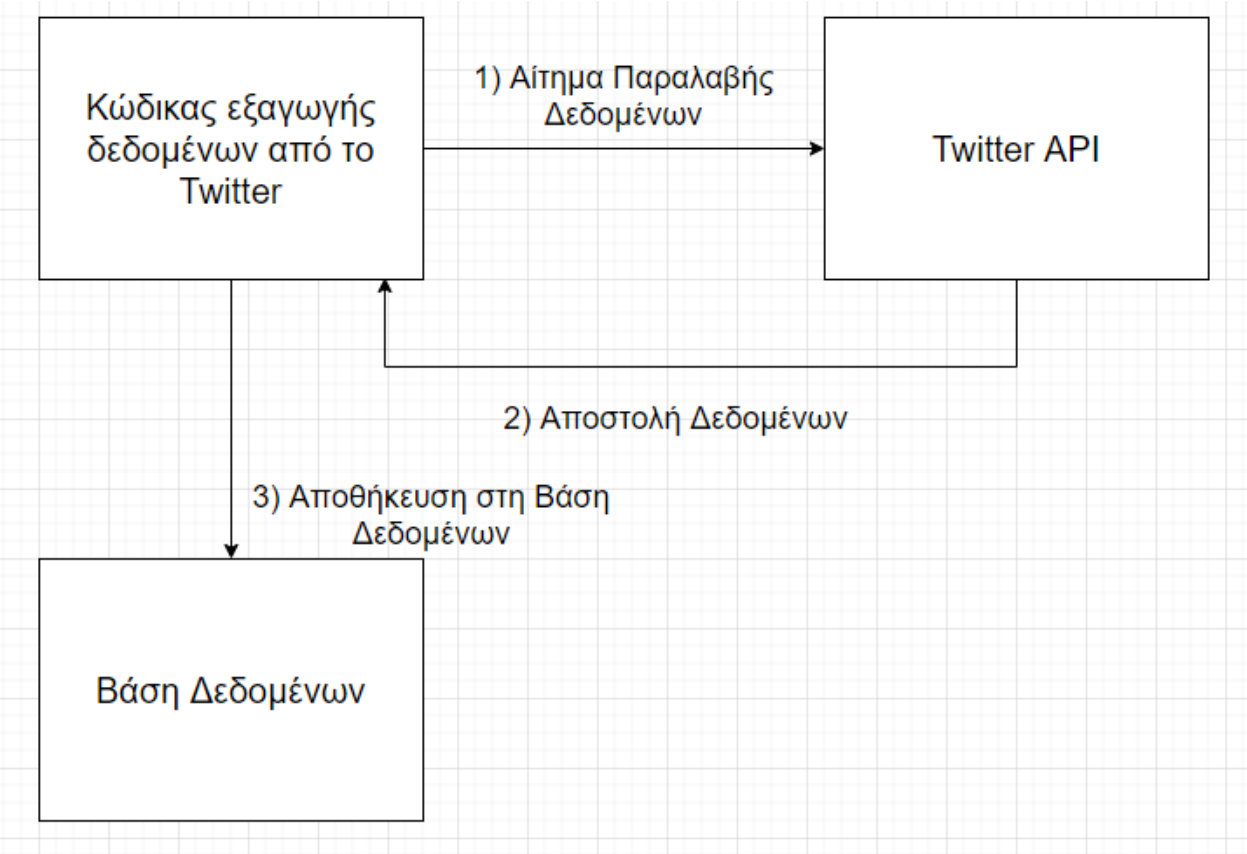
Οι πίνακες αυτοί μιας και θα περιείχαν όλα τα περιεχόμενα των πινάκων σαν στοιχεία, θα είχαν αρκετά κενά πληροφορίας σε κάθε τους εγγραφή, κάτι που εμείς το θεωρήσαμε αρνητικό. Επίσης η ύπαρξη ξεχωριστών πινάκων για κάθε ΚΔ κάνει σαφώς ευκολότερη μια διαχείριση λάθους, καθώς διευκολύνεται η διαδρομή ελέγχου. Τέλος, έχοντας ξεχωριστούς πίνακες για κάθε ΚΔ γίνεται πιο ξεκάθαρη η προσπάθεια ενοποίησης των πινάκων και η αναφορά του χρήστη πλέον ως μία οντότητα σε αυτά.

# Ενότητα 4. Ανάλυση backend προγράμματος

Στην ενότητα αυτή θα αναλυθούν οι τρόποι υλοποίησης των προγραμμάτων που χρησιμοποιήθηκαν για την συλλογή των δεδομένων από τα ΚΔ. Πέρα από τα διαγράμματα ροής κάθε προγράμματος θα υπάρχουν αναλύσεις για τις βιβλιοθήκες που χρησιμοποιούνται καθώς και για τις συναρτήσεις που αποτελούν τον κώδικα του προγράμματος. Για περισσότερες πληροφορίες μπορείτε να επισκεφτείτε το αποθετήριο της πτυχιακής μας εργασίας στο GitHub[1].

## 4.1 Συλλογή δεδομένων από το Twitter

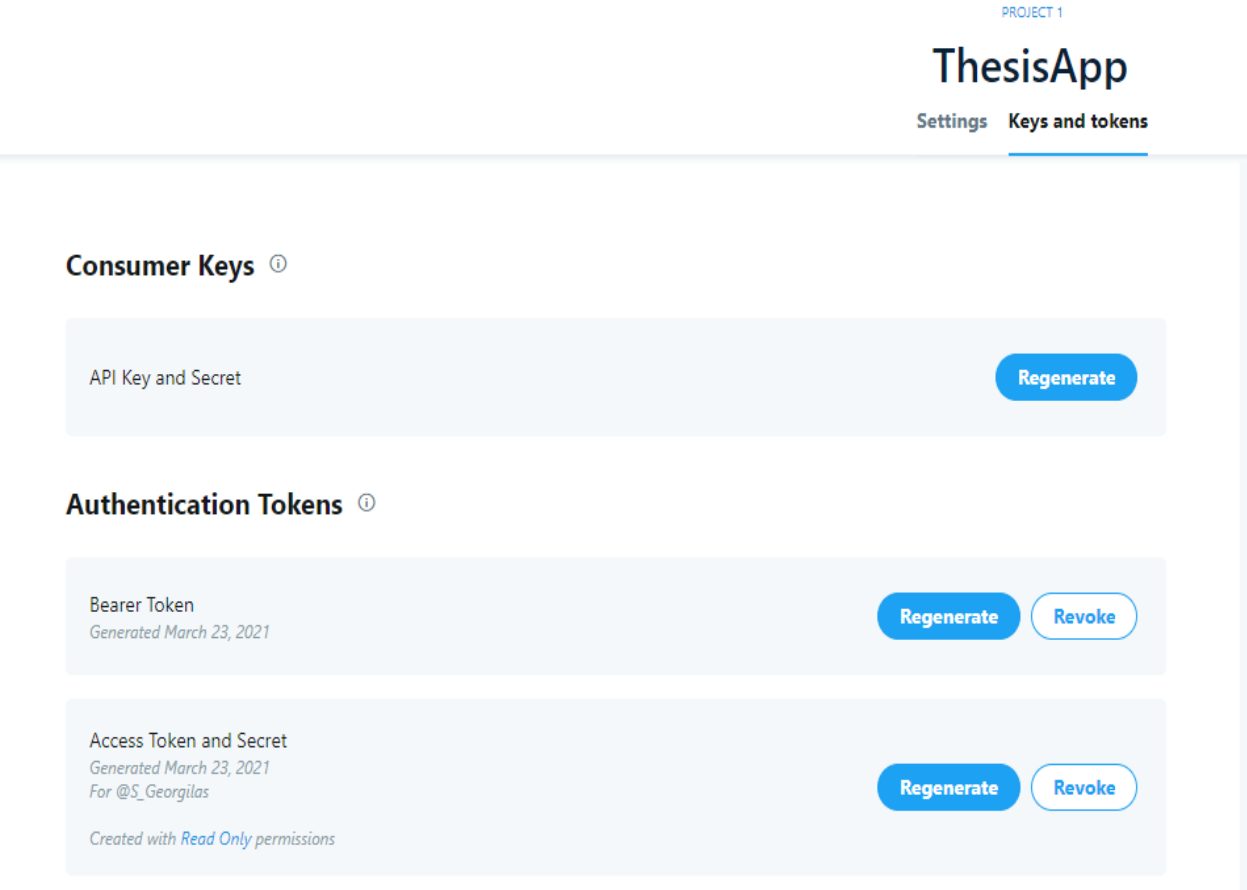
Για την συλλογή δεδομένων από το Twitter ακολουθήσαμε τη διαδικασία που φαίνεται στο διάγραμμα ροής (*Εικόνα 26*). Κάθε βήμα θα αναλυθεί στη συνέχεια.




**Εικόνα 26: Διάγραμμα ροής για το Twitter**

Για να εξάγει κάποιος δεδομένα από το Twitter χρειάζεται να λάβει άδεια από το ΚΔ. Για την επιβεβαίωση του χρήστη και την επικοινωνία του με το ΚΔ υπάρχει το Twitter API, ένα Application Programming Interface, δηλαδή μια διεπαφή προγραμματισμού εφαρμογών,

φτιαγμένη από το ίδιο το Twitter. Αρχικά, για να λάβουμε πιστοποίηση από το API, έπρεπε να στείλουμε μια αίτηση ανάπτυξης εφαρμογής, στην οποία εξηγούμε τους λόγους που θέλουμε να αναπτύξουμε μια τέτοια εφαρμογή (π.χ. για ακαδημαϊκούς σκοπούς). Στη συνέχεια, αν η αίτηση μας εγκριθεί, το Twitter API μας στέλνει τρία μοναδικά κλειδιά πρόσβασης τα οποία χρησιμοποιούμε στο πρόγραμμα μας για την ταυτοποίηση και επικοινωνία με τη διεπαφή (*Εικόνα 27*).




***Εικόνα 27: Twitter API - Authentication Tokens***

Ο κώδικας για την εξαγωγή των δεδομένων από το Twitter γράφτηκε με την γλώσσα Python και συγκεκριμένα την 2.7 έκδοση της, καθώς αυτή ήταν συμβατή με τις βιβλιοθήκες που χρησιμοποιήθηκαν. Οι βιβλιοθήκες αυτές είναι οι εξής:

- PyMySQL: Διεπαφή η οποία χρησιμοποιείται για την σύνδεση του προγράμματός μας με την MySQL ΒΔ μας. Προτιμήθηκε αυτή από την MySQLdb καθώς είναι καθαρά φτιαγμένη σε Python ενώ η δεύτερη είναι επέκταση της γλώσσας C.
- Tweepy: Βιβλιοθήκη όπου η χρήση της είναι η σύνδεση και αλληλεπίδραση με το Twitter API.

- Pandas: Βιβλιοθήκη φτιαγμένη για τον χειρισμό και την ανάλυση δεδομένων. Στον κώδικά μας χρησιμοποιήθηκε για την ανάγνωση ενός CSV (Comma Separated Values) αρχείου. Τις τιμές που η βιβλιοθήκη ανέγνωσε τις εισάγει σε ένα dataframe, δηλαδή σε ένα πλαίσιο δεδομένων.
- Sys: Ένα module το οποίο παρέχει συναρτήσεις και μεταβλητές για την χειραγώγηση πολλαπλών κομματιών του προγραμματιστικού περιβάλλοντος της Python. Στον κώδικά μας χρησιμοποιήθηκε για την αλλαγή της κωδικοποίησης από την προκαθορισμένη ASCII σε UTF-8.
- NumPy: Βιβλιοθήκη η οποία παρέχει υποστήριξη για μεγάλες, πολυδιάστατες συστοιχίες και πίνακες, μαζί με μια μεγάλη συλλογή μαθηματικών συναρτήσεων υψηλού επιπέδου. Στην δικιά μας περίπτωση χρησιμοποιήθηκε απλά για την αλλαγή ενός ακεραίου int σε ακέραιο 64bit, έπειτα από παρότρυνση του Twitter API.
- Re: Βιβλιοθήκη για τον έλεγχο εάν ένα string, δηλαδή μια σειρά χαρακτήρων ταιριάζει με μια δοσμένη κανονική έκφραση (Regular Expression). Στο πρόγραμμα μας χρησιμοποιήθηκε η συνάρτηση re.sub για τον καθαρισμό emoji και εισαγωγικών από τα tweets των χρηστών.

Το προαναφερθέν CSV αρχείο περιέχει τα usernames των χρηστών και στα τρία ΚΔ καθώς και το ψευδώνυμο που έχουμε δώσει σε κάθε χρήστη εμείς. Τα usernames χωρίζονται μεταξύ τους με ερωτηματικά. Παρακάτω παρουσιάζεται ένα μικρό κομμάτι του αρχείου αυτού, όπου πρώτα εμφανίζεται το ψευδώνυμο που έχουμε δώσει εμείς, και έπειτα εμφανίζονται κατά σειρά τα ονόματα του χρήστη στο Twitter,στο Facebook και στο Instagram αντίστοιχα (*Εικόνα 28*).

```
Alias;Twitter;Facebook;Instagram
Cnn;cnn;cnn;cnn
Time;time;time;time
Google;google;google;google
Barca;fcbarcelona;fcbarcelona;fcbarcelona
KatyPerry;katyperry;katyperry;katyperry
CR7;cristiano;Cristiano;Cristiano
AmericanAir;AmericanAir;AmericanAirlines;americanair
WhiteHouse;whitehouse;WhiteHouse;WhiteHouse
VSecret;VictoriasSecret;victoriassecret;victoriassecret
BBCSport;BBCSport;BBCSport;bbcsport
Starbucks;Starbucks;Starbucks;starbucks
McDonalds;McDonalds;McDonalds;mcdonalds
FTimes;FinancialTimes;financialtimes;financialtimes
RedDevils;ManUtd;manchesterunited;manchesterunited
9GAG;9GAG;9gag;9gag
Huffpost;HuffPost;HuffPost;huffpost
Chelsea;ChelseaFC;ChelseaFC;chelseafc
Marvel;Marvel;Marvel;marvel
MLS;MLS;MLS;mls
BMW;BMW;BMW;bmw
Yahoo;yahoo;yahoo;yahoo
NRJ;NRJhitmusiconly;NRJANTILLES97;nrjhitmusiconly
Bulls;chicagobulls;chicagobulls;chicagobulls
Liverpool;LFC;LiverpoolFC;liverpoolfc
```

***Εικόνα 28: Csv Αρχείο***

Στην *Εικόνα 29* βλέπουμε την κύρια συνάρτηση του προγράμματός μας.

```
1   import pandas as pd
2   import sys
3   reload(sys)
4   sys.setdefaultencoding('utf-8')
5   import backend
6
7   connection=backend.connectionopen()
8   cursor= backend.connectioncursor(connection)
9   try:
10      df = pd.read_csv(r'C:\Users\steli\Desktop\Thesis\usernames.csv', encoding = 'ISO-8859-7', delimiter=';')
11      print("Usernames have been loaded!\n")
12  except:
13      print ("Usernames file error!\n")
14      df = []
15  api = backend.twitter_api_authorization()
16  counter=0
17  for twitter_user in df["Twitter"]:
18      alias=df.iloc[counter,0]
19      backend.twitter_fetch_data(alias,api, twitter_user,cursor,connection)
20      counter = counter+1
21  backend.connectionclose(connection)
```

***Εικόνα 29: Συνάρτηση main του προγράμματος - Twitter***

- Στις γραμμές 1-5 γίνονται οι εισαγωγές των παραπάνω βιβλιοθηκών.
- Στις γραμμές 7-8 γίνεται η σύνδεση του προγράμματος με την ΒΔ και η δημιουργία ενός cursor ο οποίος θα διασχίζει την ΒΔ και θα υλοποιεί τα αιτήματά μας (*Εικόνα 30*).

```
def connectionopen():
    try:
        connection = pymysql.connect(host='localhost', user='root', password= [redacted], database='socialmediadb', charset='utf8')
        print("Connection with the database has been established!\n")
    except Exception as e:
        print("Connection error!\n")
    return connection

def connectioncursor(connection):
    cursor = connection.cursor()
    cursor.execute('SET NAMES utf8mb4')
    cursor.execute("SET CHARACTER SET utf8mb4")
    return cursor
```

***Εικόνα 30: Συναρτήσεις σύνδεσης βάσης δεδομένων***

- Στη συνέχεια, στις γραμμές 9-14 διαβάζουμε τα ονόματα των χρηστών από το CSV αρχείο και τα περνάμε σε ένα dataframe για την ευκολότερη πρόσβασή τους.

- Στη γραμμή 15 επιτυγχάνεται η ταυτοποίησή μας και η σύνδεση με το Twitter API (*Εικόνα 31*).

```
def twitter_api_authorization():
    auth = tweepy.OAuthHandler(                                                    )
    auth.set_access_token(                                  ,
                                                            )
    api = tweepy.API(auth,wait_on_rate_limit=True,
    wait_on_rate_limit_notify=True)
    return api
```

***Εικόνα 31: Twitter API Authorization***

- Στις γραμμές 16-18 διαβάζουμε το dataframe και παίρνουμε τα Twitter usernames των χρηστών ένα-ένα και τα φορτώνουμε στη συνάρτηση twitter_fetch_data της γραμμής 19 που θα αναλυθεί εκτενέστερα παρακάτω.
- Στη γραμμή 21 τερματίζουμε την σύνδεσή μας με την ΒΔ και τερματίζουμε το πρόγραμμα.

Επειδή η συνάρτηση twitter_fetch_data είναι μεγάλη θα αναλυθεί σε ξεχωριστά κομμάτια της.

Στην *Εικόνα 32* βλέπουμε το κομμάτι της που διαχειρίζεται δεδομένα καθαρά για τα προφίλ των χρηστών.

```
35  def twitter_fetch_data(alias,api, twitter_user,cursor,connection):
36      user = api.get_user(twitter_user)
37      user_id = user.id
38      cursor.execute("SELECT * FROM Twitter")
39      myresult = cursor.fetchall()
40      exist_user = False
41      for exist_users in myresult:
42          if (user_id == exist_users[0]):
43              exist_user = True
44              break
45          else:
46              exist_user = False
47      if(exist_user== False):
48          insertvaluetwitterusers(user_id, user.screen_name, user.followers_count, user.friends_count, user.statuses_count, user.favourites_count,
49                                  user.description,cursor,connection)
50          inserttwitteruserid(alias,user_id,cursor,connection)
51      else:
52          updatevaluetwitterusers(user_id, user.screen_name, user.followers_count, user.friends_count, user.statuses_count, user.favourites_count,
53                                  user.description,cursor,connection)
```

***Εικόνα 32: Twitter_fetch_data - Profiles***

Στην γραμμή 36 αιτούμαστε από το Twitter API να μας επιστρέψει το JSON αντικείμενο που σχετίζεται με τον χρήστη που έχουμε δώσει ως όρισμα. Ένα παράδειγμα ενός JSON αντικειμένου, όπως επιστέφεται από το API, υπάρχει στην *Εικόνα 33*.

```
{
	"id": 6253282,
	"id_str": "6253282",
	"name": "Twitter API",
	"screen_name": "TwitterAPI",
	"location": "San Francisco, CA",
	"profile_location": null,
	"description": "The Real Twitter API. Tweets about API changes, service issues and our Developer Platform. Don't get an answer? It's on my website.",
	"url": "https:\/\/t.co\/8IkCzCDr19",
	"entities": {
		"url": {
			"urls": [{
				"url": "https:\/\/t.co\/8IkCzCDr19",
				"expanded_url": "https:\/\/developer.twitter.com",
				"display_url": "developer.twitter.com",
				"indices": [
					0,
					23
				]
			}]
		},
		"description": {
			"urls": []
		}
	}
}
```


***Εικόνα 33: Παράδειγμα Json αντικειμένου***

Στη συνέχεια ελέγχουμε για το αν ο χρήστης υπάρχει στη βάση μας ή όχι. Αν δεν υπάρχει εισάγουμε τα δεδομένα μας στους πίνακες “Twitter” και “User” της ΒΔ (*Εικόνα 1* και *Εικόνα 4*). Αλλιώς, απλά ανανεώνουμε την εγγραφή μας στη βάση με τα δεδομένα που μόλις συλλέξαμε.

Έπειτα, μέσω της συνάρτησης twitter_fetch_data (*Εικόνα 34*) ζητάμε από το Twitter API να μας επιστρέψει τα τελευταία 3200 tweets που έχει δημοσιεύσει ο χρήστης τα οποία είναι και αυτά σε μορφή JSON. Έπειτα κάθε tweet περνάει από έλεγχο για την ήδη ύπαρξη του στη βάση ή όχι.

```
60      for status in tweepy.Cursor(api.user_timeline, twitter_user, tweet_mode="extended").items():
61          sql1 = """SELECT * FROM TWEETS WHERE twitter_user_id = '%d'""" % \
62                 (int(user_id))
63          cursor.execute(sql1)
64          myresult1 = cursor.fetchall()
65          exist_tweet = False
66          for exist_tweets in myresult1:
67              try:
68                  if (int(status.id) == int(exist_tweets[0])):
69                      exist_tweet = True
70                      break
71                  else:
72                      exist_tweet = False
73              except:
74                  exist_tweet = False
```

***Εικόνα 34: Twitter_fetch_data – Tweets***

Αν το tweet δεν υπάρχει στη βάση μας, ελέγχεται για το αν είναι retweet ή όχι, καθώς σε κάθε περίπτωση τα στοιχεία από το JSON αντικείμενο που είναι συμπληρωμένα είναι διαφορετικά. Σε κάθε μία από τις δύο περιπτώσεις, η λογική του κώδικα είναι ίδια απλά αλλάζουν τα δεδομένα που εξάγουμε από τα tweets. Για αυτό θα αναλύσουμε περαιτέρω την περίπτωση που το tweet δεν είναι retweet (*Εικόνα 35*).


***Εικόνα 35: Twitter_fetch_data – User Mentions***

- Στις γραμμές 150-151 προσθέτουμε στον πίνακα "Tweets" (*Εικόνα 8*) της βάσης μας την εγγραφή αυτού του tweet.
- Στις γραμμές 152-155 ελέγχουμε για το αν έχουν υπάρχει αναφορές σε άλλους χρήστες σε αυτό το tweet. Αν υπάρχουν τους εισάγουμε στον πίνακα "TwitterUserMentions" της βάσης (*Εικόνα 9*).

Στην *Εικόνα 36* βλέπουμε το κομμάτι του κώδικα που χρησιμοποιείται για την διαχείριση των υπερσυνδέσμων που βρίσκονται στα tweets. Συγκεκριμένα, κάθε υπερσύνδεσμος που βρίσκεται μέσα στο JSON αντικείμενο του tweet καθαρίζεται από οποιαδήποτε εισαγωγικά έχει. Αμέσως μετά ελέγχεται η ύπαρξη του στη βάση.

Αν δεν υπάρχει, παίρνουμε το πλήρες URL καθώς και το domain του, το οποίο προκύπτει από ειδικό διαχωρισμό του υπερσυνδέσμου ο οποίος γίνεται στις γραμμές 166-169, και τα εισάγουμε στον πίνακα "Links" της βάσης (*Εικόνα 5*). Στην συνέχεια παίρνουμε τον μοναδικό αύξων αριθμό link_id που προέκυψε στην εγγραφή της βάσης και *τον* αναγνωριστικό κωδικό του tweet και τα εισάγουμε στον ενδιάμεσο συνδετικό πίνακα "Links2tweet" (*Εικόνα 10*).

Αν ο υπερσύνδεσμος υπάρχει ήδη , λαμβάνουμε το link_id της πρώτης εμφάνισής του στη βάση μας και το εισάγουμε μαζί με τον αναγνωριστικό κωδικό του tweet στον ενδιάμεσο πίνακα "Links2Tweet".

```
156                 if 'urls' in status.entities:
157                     for link in status.entities["urls"]:
158                         expandedurl = link["expanded_url"]
159                         b=cleantextfromquotes(expandedurl)
160                         sqllink = """SELECT link_id FROM Links WHERE url = '%s'""" % \
161                                   (b)
162                         cursor.execute(sqllink)
163                         myresultlink = cursor.fetchone()
164                         if myresultlink == None:
165                             try:
166                                 urlsplit = expandedurl.split("//", 1)
167                                 domainlink = urlsplit[1]
168                                 domainsplit = domainlink.split("/", 1)
169                                 domain = domainsplit[0]
170                                 insertvaluetwitterlinks(expandedurl, domain,cursor,connection)
171                                 sqllinkid = """SELECT link_id FROM Links WHERE url = '%s'""" % \
172                                             (str(expandedurl))
173                                 cursor.execute(sqllinkid)
174                                 myresultlinkid = cursor.fetchone()
175                                 insertvaluetwitterlinks2tweet(status.id, myresultlinkid[0],cursor,connection)
176                             else:
177                                 insertvaluetwitterlinks2tweet(status.id, myresultlink[0],cursor,connection)
```

***Εικόνα 36: Twitter_fetch_data – Links***

Παρόμοια είναι και η διαχείριση των πολυμέσων στον κώδικα (*Εικόνα 37*). Για κάθε εικόνα ή βίντεο που υπάρχει στο tweet και αντίστοιχα στο JSON αντικείμενο, εξάγεται από αυτό το URL του πολυμέσου. Αμέσως μετά ελέγχεται η ύπαρξη του στη βάση και αντίστοιχα είτε εισάγουμε μία εγγραφή στους πίνακες “Media” και “Media2tweet” (*Εικόνα 6* και *Εικόνα 11*), είτε εισάγουμε μία εγγραφή στον πίνακα “Media2tweet” με το media_id της πρώτης εμφάνισής του πολυμέσου στη βάση και τον αναγνωριστικό κωδικό του tweet.

```
178                 if 'media' in status.entities:
179                     for image in status.entities["media"]:
180                         mediaurl = (image["media_url_https"])
181                         sqlmedia = """SELECT media_id FROM Media WHERE media_text = '%s'""" % \
182                                    (str(mediaurl))
183                         cursor.execute(sqlmedia)
184                         myresultmedia = cursor.fetchone()
185                         if myresultmedia == None:
186                             try:
187                                 insertvaluetwittermedia(mediaurl,cursor,connection)
188                                 sqlmediaid = """SELECT media_id FROM Media WHERE media_text = '%s'""" % \
189                                              (str(mediaurl))
190                                 cursor.execute(sqlmediaid)
191                                 myresultmediaid = cursor.fetchone()
192                                 insertvaluetwittermedia2tweet(status.id, myresultmediaid[0],cursor,connection)
193                             except:
194                         else:
195                             insertvaluetwittermedia2tweet(status.id, myresultmedia[0],cursor,connection)
```

***Εικόνα 37: Twitter_fetch_data – Media***

Τέλος, στην *Εικόνα 38* βλέπουμε την διαχείριση του προγράμματος για την περίπτωση των hashtags. Στην περίπτωση μη ύπαρξης του hashtag στη βάση προσθέτουμε μια εγγραφή στους πίνακες “Hashtag” και “Hash2tweet” (*Εικόνα 7* και *Εικόνα 12*), αλλιώς προσθέτουμε μια εγγραφή στον πίνακα “Hash2tweet”.

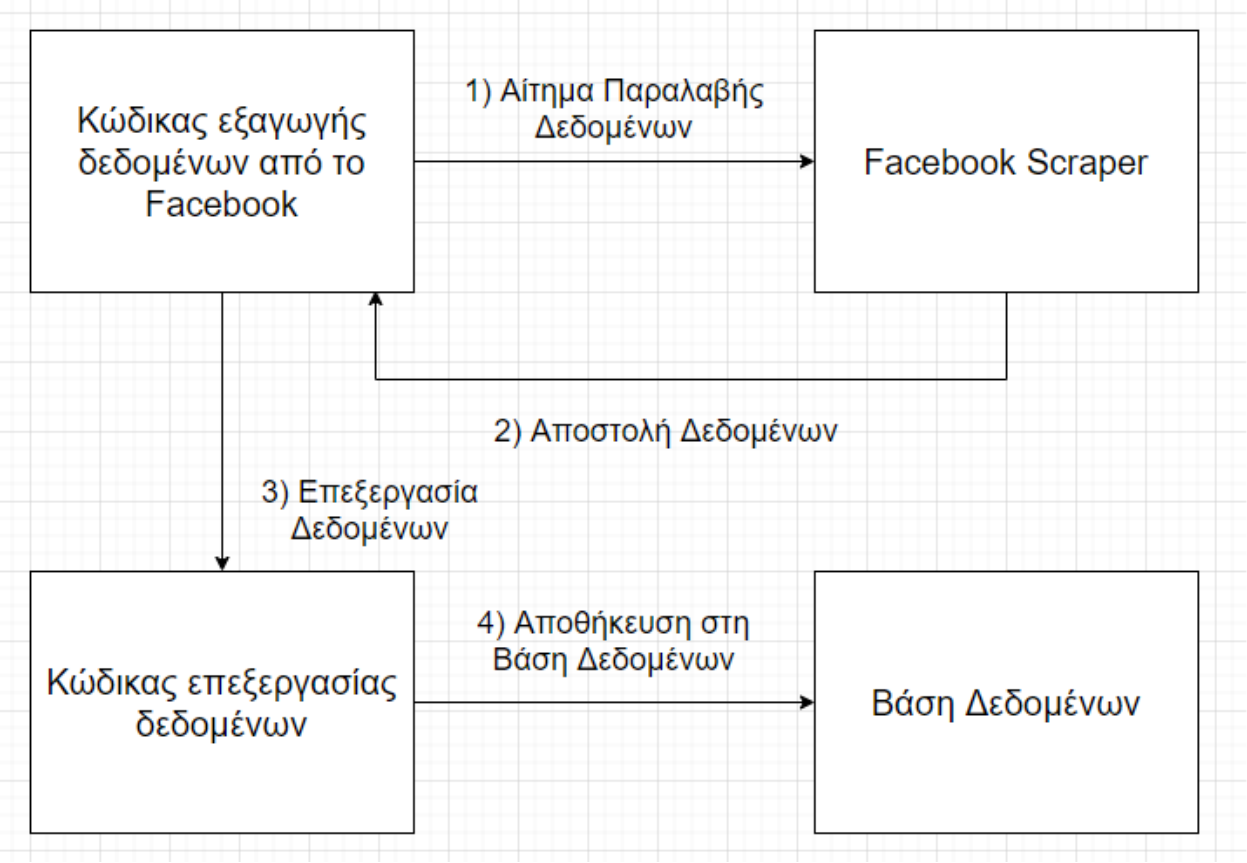
Κλείνοντας, μας έμεινε η περίπτωση που το tweet υπάρχει ήδη στη βάση μας. Τότε το μόνο που κάνουμε είναι να ενημερώσουμε την εγγραφή που υπάρχει στον πίνακα “Tweets” (*Εικόνα 8*) με τα καινούργια δεδομένα.

```
196                 if 'hashtags' in status.entities:
197                     for hashtag in status.entities["hashtags"]:
198                         text = hashtag["text"]
199                         sqlhashtag = """SELECT hash_id FROM Hashtag WHERE value = '%s'""" % \
200                                     (str(text))
201                         cursor.execute(sqlhashtag)
202                         myresulthashtag = cursor.fetchone()
203                         if myresulthashtag == None:
204                             try:
205                                 insertvaluetwitterhashtag(text,cursor,connection)
206                                 sqlhashtagid = """SELECT hash_id FROM Hashtag WHERE value = '%s'""" % \
207                                             (str(text))
208                                 cursor.execute(sqlhashtagid)
209                                 myresulthashtagid = cursor.fetchone()
210                                 insertvaluetwitterhash2tweet(status.id, myresulthashtagid[0],cursor,connection)
211                             except:
212 
213                         else:
214                             insertvaluetwitterhash2tweet(status.id, myresulthashtag[0],cursor,connection)
215             else:
216                 updatevaluetwitterposts(status.id, status.retweet_count, status.favorite_count,cursor,connection)
```

***Εικόνα 38: Twitter_fetch_data – Hashtags***

## 4.2. Συλλογή δεδομένων από το Facebook

Για την συλλογή δεδομένων από το Facebook ακολουθήσαμε την πορεία που φαίνεται στο διάγραμμα ροής παρακάτω (*Εικόνα 39*). Στη συνέχεια κάθε βήμα θα αναλυθεί περαιτέρω.




**Εικόνα 39: Διάγραμμα ροής για το Facebook**

Για το Facebook, όπως και για το Instagram που θα αναλυθεί στη συνέχεια, δεν υπάρχει κάποιο διαθέσιμο API, οπότε αναγκαστήκαμε να προβούμε σε άλλες λύσεις για να εξάγουμε δεδομένα. Συγκεκριμένα για το Facebook χρησιμοποιήθηκε ένας Scraper[3] του χρήστη Kevinzg στο GitHub στον οποίο αποδίδονται όλα τα credits.

Το μέγεθος των δεδομένων είναι αρκετά μικρό και η χρήση τους είναι καθαρά για ακαδημαϊκούς σκοπούς, σεβόμενοι τους Όρους Χρήσης των ΚΔ.

Ο κώδικας του προγράμματός μας υλοποιήθηκε με την έκδοση 3.6 της Python και πέρα από τις βιβλιοθήκες PyMySQL, Pandas, Sys, και Re που αναλύθηκαν στην Ενότητα 4.1 χρησιμοποιήθηκαν και οι βιβλιοθήκες:

3 https://github.com/kevinzg/facebook-scraper

- Facebook_scraper: Ο προαναφερθείς Scraper ο οποίος μπορεί να εξάγει δεδομένα από δημόσια προφίλ χρηστών του Facebook χωρίς κάποιο API κλειδί.
- Time: Module το οποίο μας παρέχει συναρτήσεις σχετικές με την διαχείριση του χρόνου. Στον κώδικά μας χρησιμοποιείται η time.sleep για να καθυστερήσουμε τον scraper ώστε να μην ξεπεράσουμε το «χρονικά επιτρεπόμενο όριο» που έχει ορίσει το Facebook πριν θεωρήσει ύποπτα τα αιτήματα για δεδομένα.

Στην *Εικόνα 40* βλέπουμε την κύρια συνάρτηση του προγράμματός μας.

```
1  import pymysql
2  import pandas as pd
3  import sys
4  import facebook_scraper
5  import fb_backend
6
7  connection=fb_backend.connectionopen()
8  cursor= fb_backend.connectioncursor(connection)
9  try:
10     df = pd.read_csv(r'C:\Users\steli\Desktop\Thesis\usernames.csv', encoding = 'ISO-8859-7', delimiter=';')
11     print("Usernames have been loaded!\n")
12 except:
13     print ("Usernames file error!\n")
14     df = []
15 counter = 0
16 for facebook_user in df["Facebook"]:
17     post_list = fb_backend.fb_scraper(facebook_user)
18     alias = df.iloc[counter, 0]
19     fb_backend.RecordFBValuesToDB(facebook_user, post_list, alias, cursor, connection)
20     counter = counter + 1
21 fb_backend.connectionclose(connection)
```

***Εικόνα 40: Συνάρτηση main του προγράμματος – Facebook***

- Στις γραμμές 1-5 γίνεται η εισαγωγή των βιβλιοθηκών.

Στις γραμμές 7-15 γίνεται η σύνδεση του προγράμματος με την βάση μας όπως δείξαμε στην *Εικόνα 30*, καθώς και η ανάγνωση των Facebook usernames των χρηστών από το csv αρχείο (*Εικόνα 28*) και φόρτωση τους σε ένα dataframe.

- Στην γραμμή 17, παίρνουμε κάθε χρήστη και τον εισάγουμε ως όρισμα στην συνάρτηση fb_scraper (*Εικόνα* 41).

```
27 def fb_scraper(fb_user):
28     get_posts = facebook_scraper.get_posts(fb_user, pages=16,timeout=10)
29     post_list = []
30     for post in get_posts:
31         post_list.append(post)
32     return post_list
```

***Εικόνα 41: Συνάρτηση fb_scraper***

Στη συνάρτηση αυτή κάνουμε αίτημα στον Facebook scraper να μας επιστρέψει 16 σελίδες από δημοσιεύσεις οι οποίες αναλογούν σε περίπου 64 δημοσιεύσεις. Αυτές οι δημοσιεύσεις έρχονται σε μορφή JSON αντικειμένου για ευκολότερη πρόσβαση του περιεχομένου τους (*Εικόνα 42*). Στη συνέχεια κάθε τέτοιο JSON αντικείμενο το βάζουμε σε μια λίστα την οποία θα την διατρέξουμε στις επόμενες συναρτήσεις.

```
{'post_id': '10162133195231509',
 'text': "They belt out songs of victory, their boots adding the drumbeat as r
 'post_text': "They belt out songs of victory, their boots adding the drumbeat
 'shared_text': 'CNN.COM\nInside the Myanmar mountain camp where rebels train
 'time': datetime.datetime(2021, 7, 8, 21, 30, 16),
 'image': 'https://external.fskg1-2.fna.fbcdn.net/safe_image.php?d=AQHqutWQ0_c
 'images': ['https://external.fskg1-2.fna.fbcdn.net/safe_image.php?d=AQHqutWQ0
 'video': None,
 'video_thumbnail': None,
 'video_id': None,
 'likes': 0,
 'comments': 29,
 'shares': 0,
 'post_url': 'https://facebook.com/cnn/posts/10162133195231509',
 'link': 'https://cnn.it/3jYzM1G',
 'user_id': '5550296508',
 'username': 'CNN',
 'is_live': False,
 'factcheck': None,
 'shared_post_id': None,
 'shared_time': None,
 'shared_user_id': None,
 'shared_username': None,
 'shared_post_url': None,
 'available': True,
 'comments_full': None,
 'reactors': None,
 'w3_fb_url': None}
```

***Εικόνα 42: Facebook JSON αντικείμενο***

- Στην γραμμή 19 της συνάρτησης main καλούμε την συνάρτηση RecordFBValuestoDB η οποία θα αναλυθεί σε βάθος παρακάτω.
- Στη γραμμή 21 τερματίζουμε την σύνδεση μας με τη ΒΔ και τερματίζουμε το πρόγραμμα.

Όπως και στο Twitter έτσι και στο Facebook η συνάρτηση διαχείρισης και επεξεργασίας των δεδομένων θα αναλυθεί τμηματικά.

Στην *Εικόνα 43* αρχικά παίρνουμε από το JSON αντικείμενο το facebook_id του χρήστη και με αυτό κάνουμε αναζήτηση στη βάση για το αν υπάρχει το προφίλ του χρήστη στη βάση ή όχι. Αν ο χρήστης δεν υπάρχει, τότε με τις συναρτήσεις insertValueFBUsers και insertfacebookuserid προσθέτουμε μια εγγραφή στους πίνακες “Facebook” και “User” αντίστοιχα (*Εικόνα 2* και

*Εικόνα 4*). Αλλιώς, αν ο χρήστης υπάρχει ήδη δεν χρειάζεται να κάνουμε τίποτα γιατί στον πίνακα “Facebook” δεν υπάρχει κάτι που χρειάζεται να ανανεωθεί.

```
193  def RecordFBValuesToDB(fb_user, post_list , alias , cursor , connection):
194      user_id = post_list[0]['user_id']
195      cursor.execute("SELECT * FROM Facebook")
196      myresult = cursor.fetchall()
197      exist_user = False
198      for exist_users in myresult:
199          if(fb_user == exist_users[1]):
200              exist_user = True
201          else:
202              exist_user = False
203
204      if(exist_user == False):
205          insertValueFBUsers(user_id, fb_user, cursor, connection)
206          insertfacebookuserid(alias, user_id, cursor, connection)
207      else:
208          print("User is already loaded in database!\n")
```

***Εικόνα 43: RecordFbValuesToDB - Users***

Ύστερα, διατρέχουμε τη λίστα με τις δημοσιεύσεις, και κάθε δημοσίευση περνάει από έλεγχο για το αν υπάρχει ήδη στη βάση, όπως φαίνεται και στην *Εικόνα 44*.

```
213      for i in range (0, len(post_list)):
214          time.sleep(2)
215          post = post_list[i]
216          exist_post = False
217          sql = """SELECT * FROM FacebookPosts WHERE facebook_user_id = '%s'""" % \
218                (user_id)
219          cursor.execute(sql)
220          myresult = cursor.fetchall()
221          for exist_posts in myresult:
222              try:
223                  if(int(post['post_id']) == int(exist_posts[0])):
224                      exist_post = True
225                      break
226                  else:
227                      exist_post = False
228              except:
229                  exist_post = False
```

***Εικόνα 44: RecordFbValuesToDB – Posts***

Στην *Εικόνα 45* βλέπουμε πως αν η δημοσίευση δεν υπάρχει στη βάση, παίρνουμε το κείμενο του από το JSON αντικείμενο, το καθαρίζουμε από πιθανά εισαγωγικά που μπορεί να έχει και μαζί με τα υπόλοιπα στοιχεία δημιουργούμε την εγγραφή που θα προσθέσουμε στον πίνακα "FacebookPosts" (*Εικόνα 13*).

```
231 if(exist_post == False):
232     text=post['text']
233     try:
234         cleantext=cleantextfromquotes(text)
235     except:
236         continue
237     try:
238         insertValueFBPosts(str(post['post_id']), user_id, str(cleantext), int(post['likes']), int(post['comments']), int(post['shares']),
239                            post['time'].strftime("%Y/%m/%d %H:%M:%S"),cursor, connection)
```

***Εικόνα 45: RecordFbValuesToDB – Posts 2***

Έπειτα έρχεται η διαχείριση των υπερσυνδέσμων που βρίσκονται στη δημοσίευση όπως φαίνεται στην *Εικόνα 46*. Αρχικά καθαρίζουμε τον υπερσύνδεσμο από πιθανά εισαγωγικά και έπειτα ελέγχουμε την ύπαρξη του στη βάση με την ίδια λογική που ελέγξαμε και στο πρόγραμμα του Twitter. Αν δεν υπάρχει, παίρνουμε το url από το JSON αντικείμενο, το επεξεργαζόμαστε για να απομονώσουμε το domain του και ύστερα και τα 2 μαζί τα προσθέτουμε ως εγγραφή στον πίνακα "Links" (*Εικόνα 5*). Το link_id που προκύπτει σε αυτή την εγγραφή μαζί με τον αναγνωριστικό κωδικό της δημοσίευσης δημιουργούν μια εγγραφή στον συνδετικό πίνακα "Links2fb" (*Εικόνα 14*).

Αν πάλι ο υπερσύνδεσμος υπάρχει ήδη στη βάση, δημιουργούμε μόνο την εγγραφή στον πίνακα "Links2fb" με στοιχεία του τον αναγνωριστικό κωδικό της δημοσίευσης και το link_id της πρώτης εμφάνισής του.

```
242                 if (post['link']!= None):
243                         link = post['link']
244                         try:
245                             b = cleantextfromquotes(link)
246                             sqllink = """SELECT link_id FROM Links WHERE url = '%s'""" % \
247                                     (b)
248                             cursor.execute(sqllink)
249                             myresultlink = cursor.fetchone()
250                             if myresultlink == None:
251                                 try:
252                                     urlsplit = b.split("//", 1)
253                                     domainlink = urlsplit[1]
254                                     domainsplit = domainlink.split("/", 1)
255                                     domain = domainsplit[0]
256                                     insertvaluefblinks(b, domain, cursor, connection)
257                                     sqllinkid = """SELECT link_id FROM Links WHERE url = '%s'""" % \
258                                                 (str(b))
259                                     cursor.execute(sqllinkid)
260                                     myresultlinkid = cursor.fetchone()
261                                     insertvaluelinks2fb(post['post_id'], myresultlinkid[0], cursor, connection)
262                                 except: print("Error at link insertion to database!\n")
263                         except: print("Error cleaning quotes!\n")
264                         else:
265                             insertvaluelinks2fb(post['post_id'], myresultlink[0], cursor, connection)
```

***Εικόνα 46: RecordFbValuesToDB – Links***

Ακολουθεί η διαχείριση των πολυμέσων η οποία εμφανίζεται στην *Εικόνα 47*. Παίρνουμε το url κάθε εικόνας η βίντεο που υπάρχει στο tweet από το JSON αντικείμενο και ελέγχουμε την ύπαρξη του στη βάση. Αν δεν υπάρχει το προσθέτουμε στον πίνακα “Media” (*Εικόνα 6*) και στον πίνακα “Media2fb” (*Εικόνα 15*) προσθέτουμε το media_id και τον αναγνωριστικό κωδικό της δημοσίευσης. Αν πάλι υπάρχει, απλά προσθέτουμε μια εγγραφή στον πίνακα “Media2fb” με το media_id της πρώτης εμφάνισης του πολυμέσου στη βάση και τον αναγνωριστικό κωδικό της δημοσίευσης.

```
268     if (post['image']!= None):
269         mediaurl = post['image']
270         sqlmedia = """SELECT media_id FROM Media WHERE media_text = '%s'""" % \
271                    (str(mediaurl))
272         cursor.execute(sqlmedia)
273         myresultmedia = cursor.fetchone()
274         if myresultmedia == None:
275             try:
276                 insertvaluefbmedia(mediaurl, cursor, connection)
277                 sqlmediaid = """SELECT media_id FROM Media WHERE media_text = '%s'""" % \
278                              (str(mediaurl))
279                 cursor.execute(sqlmediaid)
280                 myresultmediaid = cursor.fetchone()
281                 insertvaluemedia2fb(post['post_id'], myresultmediaid[0], cursor, connection)
282             except:
283                 print("Error at media insertion to database!\n")
284         else:
285             insertvaluemedia2fb(post['post_id'], myresultmedia[0], cursor, connection)
```

***Εικόνα 47: RecordFbValuesToDB – Media***

Στην *Εικόνα 48* μέσω της συνάρτησης extract_hashtags εξάγουμε τα hashtags της δημοσίευσης κάνοντας ειδική αναζήτηση για αυτές, και στη συνέχεια τις τοποθετούμε στη βάση όπως φαίνεται στην *Εικόνα 49*. Η δημιουργία συνάρτησης ειδικής διαχείρισης των hashtags προέκυψε από το γεγονός πως δεν υπάρχουν ως οντότητες μέσα στο JSON αντικείμενο της δημοσίευσης. Χωρίζουμε το κείμενο της δημοσίευσης σε λέξεις, και αναζητούμε αυτές που ξεκινάμε με το ειδικό σύμβολο της δίεσης ‘#’, το οποίο έχει καθιερωθεί ως το σύμβολο με το οποίο ξεκινάνε οι όροι των hashtags. Όσοι πληρούν το κριτήριο αναζήτησης εισάγονται σε μία λίστα. Κάθε hashtag της λίστας ελέγχεται για την ήδη ύπαρξή του στη βάση μας. Αν δεν υπάρχει, την εισάγουμε στον πίνακα “Hashtag” (*Εικόνα 7*), κρατάμε το hash_id που προκύπτει και το εισάγουμε στον πίνακα “Hash2Fb” (*Εικόνα 16*) μαζί με τον αναγνωριστικό κωδικό της δημοσίευσης. Αν το hashtag προϋπήρχε στη βάση , βρίσκουμε το hash_id της πρώτης εμφάνισης του και μαζί με τον αναγνωριστικό κωδικό της δημοσίευσης εισάγονται στον συνδετικό πίνακα “Hash2Fb”.

Για το τέλος έμεινε η πιθανότητα να υπάρχει η δημοσίευση ήδη στη βάση μας. Σε αυτή την περίπτωση απλά ενημερώνουμε τα δεδομένα της δημοσίευσης που μπορεί να έχουν αλλάξει χωρίς να ασχοληθούμε με τα υπόλοιπα δεδομένα όπως είναι οι υπερσύνδεσμοι, τα πολυμέσα, και τα hashtags αφού αυτά μπήκαν στη βάση κατά την διάρκεια της πρώτης εμφάνισης της δημοσίευσης.

```
288             text1 = post['text']
289             id = post['post_id']
290             extract_hashtags(id,text1,cursor,connection)
291         except:
292             print("Error on post insertion to database!\n")
293     else:
294         print("Post already loaded in database,waiting for update!\n")
295         updatevaluefbposts(str(post['post_id']), post['likes'], post['comments'], post['shares'], cursor, connection)
```

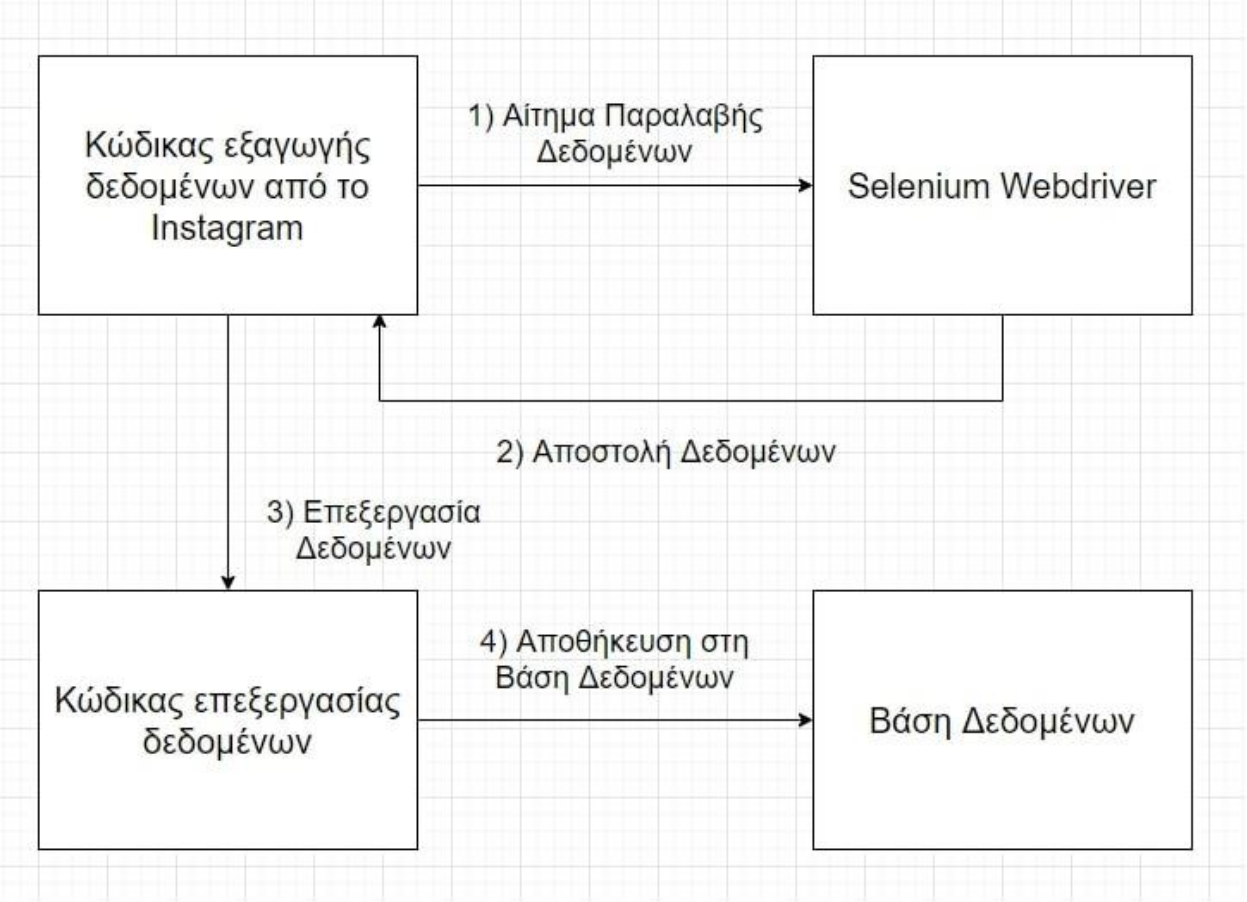
*Εικόνα 48: RecordFbValuesToDB – Hashtags*

```
102 def extract_hashtags(id,text,cursor,connection):
103     hashtag_list = []
104     for word in text.split():
105         if word[0] == '#':
106             hashtag_list.append(word[1:])
107     for hashtag in hashtag_list:
108         print(hashtag)
109         sqlhashtag = """SELECT hash_id FROM Hashtag WHERE value = '%s' """ % \
110                      (hashtag)
111         cursor.execute(sqlhashtag)
112         myresulthashtag = cursor.fetchone()
113         if myresulthashtag == None:
114             try:
115                 insertvaluefbhashtag(hashtag, cursor, connection)
116                 sqlhashtagid = """SELECT hash_id FROM Hashtag WHERE value = '%s' """ % \
117                                (hashtag)
118                 cursor.execute(sqlhashtagid)
119                 myresulthashtagid = cursor.fetchone()
120                 insertvaluehash2fb(id, myresulthashtagid[0], cursor, connection)
121             except:
122                 print("Error on hashtag insertion!\n")
123         else:
124             insertvaluehash2fb(id, myresulthashtag[0], cursor, connection)
```

*Εικόνα 49: RecordFbValuesToDB - Hashtag Συνάρτηση*

## 4.3. Συλλογή δεδομένων από το Instagram

Για την συλλογή δεδομένων από το Instagram ακολουθήσαμε την πορεία που φαίνεται στο διάγραμμα ροής παρακάτω (*Εικόνα 50*). Στη συνέχεια κάθε βήμα θα αναλυθεί περαιτέρω.




**Εικόνα 50: Διάγραμμα ροής για το Instagram**

Όπως αναφέραμε και πριν, το Instagram δεν παρέχει κάποιο API, οπότε έπρεπε να βρούμε κάποιον άλλο τρόπο εξαγωγής δεδομένων. Καταλήξαμε μέσω του Selenium να αυτοματοποιήσουμε την χρήση του ΚΔ δηλαδή την αναζήτηση των χρηστών, το σκρολλάρισμα στα προφίλ τους και τη φόρτωση κάθε δημοσίευσης ξεχωριστά και μέσα από τον κώδικα της σελίδας του ΚΔ να εξάγουμε τις πληροφορίες που θέλουμε. Συγκεκριμένα χρησιμοποιήσαμε έναν ChromeDriver[4] και μέσω της βιβλιοθήκης Selenium ορίσαμε τα αυτοματοποιημένα βήματα που ακολουθούν.

[4] https://chromedriver.chromium.org/downloads

Ο κώδικας του προγράμματος μας υλοποιήθηκε με την έκδοση 3.6 της Python και πέρα από τις βιβλιοθήκες PyMySQL, Pandas, Re και Time που αναλύθηκαν στις ενότητες 4.1 και 4.2 χρησιμοποιήθηκαν και οι βιβλιοθήκες:

- Selenium: Βιβλιοθήκη η οποία πέρα από το γεγονός πως μπορεί να κάνει scraping html και xml ιστοσελίδες, είναι ικανή να κάνει scraping και ιστοσελίδες δυναμικού κώδικα όπως είναι το Instagram. Ο Selenium Webdriver είναι μια συλλογή από APIs και αυτοματοποιεί οποιαδήποτε εφαρμογή διαδικτύου.
- Json: Ενσωματωμένο πακέτο της γλώσσας Python το οποίο χειρίζεται Json αντικείμενα. Στην δικιά μας περίπτωση τα κομμάτια του δυναμικού κώδικα που εξάγαμε από την ιστοσελίδα είχαν την μορφή Json αντικειμένων, οπότε τα μετατρέψαμε σε τέτοια για να έχουμε πρόσβαση στα δεδομένα με πιο εύκολο τρόπο.

Στην *Εικόνα 51* βλέπουμε την κύρια συνάρτηση του προγράμματός μας.

```
1  import pandas as pd
2  import instagram_backend
3
4  connection=instagram_backend.connectionopen()
5  cursor= instagram_backend.connectioncursor(connection)
6  try:
7      df = pd.read_csv(r'C:\Users\steli\Desktop\Thesis\usernames.csv', encoding = 'ISO-8859-7', delimiter=';')
8      print("Usernames have been loaded!\n")
9  except:
10     print ("Usernames file error!\n")
11     df = []
12 counter = 0
13 driver = instagram_backend.openwebdriver()
14 for instagram_user in df["Instagram"]:
15     alias = df.iloc[counter, 0]
16     instagram_backend.searchforuser(instagram_user, driver)
17     instagram_backend.scrolldown(driver)
18     post_hrefs = instagram_backend.takepostlinks(driver)
19     instagram_backend.RecordInstaValuesToDB(instagram_user, alias, cursor, connection, post_hrefs, driver)
20     counter = counter + 1
21 instagram_backend.connectionclose(connection)
```

***Εικόνα 51: Συνάρτηση main του προγράμματος – Instagram***

Στις γραμμές 1-2 γίνεται η εισαγωγή των βιβλιοθηκών. Στις γραμμές 4-12 γίνεται η σύνδεση του προγράμματος με την βάση μας όπως δείξαμε στην *Εικόνα 30*, καθώς και η ανάγνωση των Instagram usernames των χρηστών από το CSV αρχείο (*Εικόνα 28*) και φόρτωση τους σε ένα dataframe.

Στην γραμμή 13 μέσω της συνάρτησης openwebdriver (*Εικόνα 52*) ανοίγουμε τον Webdriver, φορτώνουμε την ιστοσελίδα του Instagram, κάνουμε σύνδεση, πατάμε όποια κουμπιά εμφανίζει η ιστοσελίδα όπως το να αποδεχτούμε τα cookies ή να αποθηκεύσουμε τα στοιχεία σύνδεσής μας και έτσι είμαστε έτοιμοι για τα επόμενα βήματα.

```
205  def openwebdriver():
206      driver = webdriver.Chrome('C:/Users/steli/Downloads/chromedriver_win32/chromedriver.exe')
207      time.sleep(5)
208      driver.get("https://www.instagram.com/")
209      accept_all = WebDriverWait(driver, 20).until(
210          EC.element_to_be_clickable((By.XPATH, '//button[contains(text(), "Accept All")]'))).click()
211      time.sleep(5)
212      username = WebDriverWait(driver, 10).until(EC.element_to_be_clickable((By.CSS_SELECTOR, "input[name='username']")))
213      password = WebDriverWait(driver, 10).until(EC.element_to_be_clickable((By.CSS_SELECTOR, "input[name='password']")))
214      username.clear()
215      username.send_keys( [redacted] )
216      time.sleep(5)
217      password.clear()
218      password.send_keys( [redacted] )
219      time.sleep(5)
220      button = WebDriverWait(driver, 30).until(
221          EC.element_to_be_clickable((By.CSS_SELECTOR, "button[type='submit']"))).click()
222      not_now = WebDriverWait(driver, 10).until(
223          EC.element_to_be_clickable((By.XPATH, '//button[contains(text(), "Not Now")]'))).click()
224      time.sleep(2)
225      not_now2 = WebDriverWait(driver, 10).until(
226          EC.element_to_be_clickable((By.XPATH, '//button[contains(text(), "Not Now")]'))).click()
227      return driver
```

***Εικόνα 52: Συνάρτηση openwebdriver***

Στις γραμμές 16-17 αφού λάβουμε το username του χρήστη, το χρησιμοποιούμε στη συνάρτηση searchforuser (*Εικόνα 53*) για να ψάξουμε το προφίλ του χρήστη και αφού το φορτώσουμε, μέσω της συνάρτησης scrolldown (*Εικόνα 53*) και του αυτοματοποιημένου scrolling φορτώνουμε από τον δυναμικό κώδικα της σελίδας τους υπερσυνδέσμους όλων των δημοσιεύσεων.

```
230  def searchforuser(username,driver):
231      time.sleep(10)
232      searchbox = WebDriverWait(driver, 10).until(
233          EC.element_to_be_clickable((By.XPATH, "//input[@placeholder='Search']")))
234      searchbox.clear()
235      searchbox.send_keys(username)
236      time.sleep(3)
237      searchbox.send_keys(Keys.ENTER)
238      time.sleep(3)
239      searchbox.send_keys(Keys.ENTER)
240      time.sleep(3)
241
242  def scrolldown(driver):
243      # scroll down to scrape more images
244      driver.execute_script("window.scrollTo(0, 1000);")
245      time.sleep(5)
246      driver.execute_script("window.scrollTo(0, 4000);")
247      time.sleep(5)
248      driver.execute_script("window.scrollTo(0, 4000);")
249      time.sleep(5)
250      driver.execute_script("window.scrollTo(0, 4000);")
251      time.sleep(5)
252      driver.execute_script("window.scrollTo(0, 3500);")
253      time.sleep(5)
```

***Εικόνα 53: Συναρτήσεις searchforuser και scrolldown***

Στην γραμμή 18 μέσω της συνάρτησης takepostlinks (*Εικόνα 54*) εξάγουμε από τον δυναμικό κώδικα τις διευθύνσεις των προφίλ με ειδική αναζήτηση. Αυτές τις διευθύνσεις τις φορτώνουμε σε μια λίστα για μετέπειτα χρήση.

```
255  def takepostlinks(driver):
256      post_xpath_str = "//a[contains(@href, '/p/')]"
257      post_links = driver.find_elements_by_xpath(post_xpath_str)
258      post_hrefs = []
259      if len(post_links) > 0:
260          for i in range(0, len(post_links)):
261              post_link_el = post_links[i]
262              if post_link_el != None:
263                  post_hrefs.append(post_link_el.get_attribute("href"))
264                  print(post_hrefs[i])
265      time.sleep(10)
266      return post_hrefs
```

***Εικόνα 54: Συνάρτηση takepostlinks***

Στη γραμμή 19 έχουμε τη συνάρτηση RecordInstaValuesToDB η οποία θα αναλυθεί παρακάτω και τέλος στη γραμμή 21 τερματίζουμε τη σύνδεση μας με τη ΒΔ και τερματίζουμε το πρόγραμμα.

Όπως και στα προηγούμενα δύο ΚΔ η συνάρτηση διαχείρισης και επεξεργασίας των δεδομένων θα αναλυθεί τμηματικά. Στην *Εικόνα 55* με βάση το Instagram username του χρήστη κάνουμε αναζήτηση στη βάση για το αν υπάρχει το προφίλ του χρήστη στη βάση ή όχι.

```
153  def RecordInstaValuesToDB(instagram_user, alias , cursor , connection, post_hrefs, driver):
154      cursor.execute("SELECT * FROM Instagram")
155      myresult = cursor.fetchall()
156      exist_user = False
157      for exist_users in myresult:
158          if(instagram_user == exist_users[1]):
159              exist_user = True
160          else:
161              exist_user = False
```

***Εικόνα 55: RecordInstaValuesToDB -Users***

Στη συνέχεια, στην *Εικόνα 56* διατρέχουμε την λίστα με τις δημοσιεύσεις του χρήστη.

```
for i in range (0, len(post_hrefs)):
    time.sleep(20)
    postlink = post_hrefs[i]
    dict = getjsonofpost(driver, postlink)
    if (dict == 0):
        continue
    post_id,user_id,likes,comments,text,followers,posts = getpostmetadata(dict)
    if (i == 0):
        if (exist_user == False):
            insertValueInstaUsers(user_id, instagram_user, followers, posts, cursor, connection)
            insertInstagramUserId(alias, user_id, cursor, connection)
        else:
            updatevalueInstausers(user_id, instagram_user, followers, posts, cursor, connection)
```

***Εικόνα 56: RecordInstaValuesToDB -Posts***

Με την συνάρτηση getjsonofpost (*Εικόνα 57*), φορτώνουμε την διεύθυνση της δημοσίευσης στον Webdriver, εξάγουμε το κομμάτι του δυναμικού κώδικα που επιθυμούμε με ειδική αναζήτηση και με την βοήθεια της βιβλιοθήκης Json το μετατρέπουμε σε JSON αντικείμενο.

```
def getjsonofpost(driver,post_href):
    driver.get(post_href)
    time.sleep(10)
    source_data = driver.page_source
    JSON = re.compile("\"graphql\":(\\{.+\\})", re.DOTALL)
    matches = JSON.search(source_data)
    html_text = matches.group(1)
    text1 = html_text.split('}}}]},')[0]
    text2 = '}}}]}}}'
    text_i_want = text1+text2
    try:
        dict = json.loads(text_i_want)
    except:
        return 0
    return dict
```

***Εικόνα 57: Συνάρτηση getjsonofpost***

Αυτό το JSON αντικείμενο μπαίνει ως όρισμα στην συνάρτηση getpostmetadata (*Εικόνα 58*). Σε αυτή την συνάρτηση διατρέχουμε το JSON αντικείμενο και λαμβάνουμε όσες πληροφορίες χρειαζόμαστε.

```
288 def getpostmetadata(dict):
289     post_id = dict["shortcode_media"]["id"]
290     user_id = dict["shortcode_media"]["owner"]["id"]
291     likes = dict["shortcode_media"]["edge_media_preview_like"]["count"]
292     comments= dict["shortcode_media"]["edge_media_to_parent_comment"]["count"]
293     try:
294         text= dict["shortcode_media"]["edge_media_to_caption"]["edges"][0]["node"]["text"]
295         text1= cleantextfromquotes(text)
296         finaltext = deEmojify(text1)
297     except:
298         finaltext = "Empty text"
299     followed = dict["shortcode_media"]["owner"]["edge_followed_by"]
300     ffollowed = str(followed)
301     followed1 =  ffollowed.split(' ')[1]
302     followers = followed1.split('}')[0]
303     nposts = dict["shortcode_media"]["owner"]["edge_owner_to_timeline_media"]
304     npostss = str(nposts)
305     nposts1 = npostss.split(' ')[1]
306     posts = nposts1.split('}')[0]
307     return post_id,user_id,likes,comments,finaltext,followers,posts
```

***Εικόνα 58: Συνάρτηση getpostmetadata***

Έπειτα, μόνο στην πρώτη δημοσίευση, αν ο χρήστης δεν υπάρχει ήδη στη βάση, καταχωρούμε από μια εγγραφή στους πίνακες “Instagram” και “User” (*Εικόνα 3* και *Εικόνα 4*) μέσω των συναρτήσεων insertValueInstaUsers και insertInstagramUserId αντίστοιχα. Αν ο χρήστης υπάρχει, απλά ενημερώνουμε κάποιες πληροφορίες στον πίνακα “Instagram” μέσω της συνάρτησης updatevalueInstausers.

Στη συνέχεια, μέσω της συνάρτησης RecordInstaValuesToDB (*Εικόνα 59*), ελέγχουμε αν η δημοσίευση υπάρχει ήδη στη βάση ή όχι.

```
180         exist_post = False
181         sql = """SELECT * FROM Instagram_Posts WHERE instagram_user_id = '%s'""" % \
182               (user_id)
183         cursor.execute(sql)
184         myresult = cursor.fetchall()
185         for exist_posts in myresult:
186             try:
187                 if(int(post_id) == int(exist_posts[0])):
188                     exist_post = True
189                     break
190                 else:
191                     exist_post = False
192             except:
193                 exist_post = False
```

***Εικόνα 59: RecordInstaValuesToDB -Posts 2***

Αν *η δημοσίευση* δεν υπάρχει, δημιουργούμε μια εγγραφή στον πίνακα “Instagram_Posts” (*Εικόνα 17*). Επίσης μέσω των συναρτήσεων getpostmedia και getposthashtags που θα αναλυθούν αμέσως μετά, λαμβάνουμε τα πολυμέσα και τα hashtags που υπάρχουν στη

δημοσίευση. Αν πάλι *η δημοσίευση* υπάρχει ήδη στη βάση, με τη συνάρτηση updatevalueinstaposts ανανεώνουμε κάποια δεδομένα της εγγραφής στον πίνακα “Instagram_Posts” (*Εικόνα 60*).

```
195         if(exist_post == False):
196             try:
197                 insertValueInstaPosts(post_id, user_id, text, likes, comments, cursor, connection)
198                 getpostmedia(dict,post_id,cursor,connection)
199                 getposthashtags(post_id, driver, cursor, connection)
200             except:
201                 print("To post den mporese na mpei sti vasi!\n")
202         else:
203             print("To insta post iparxei idi stin vasi,paw na to kanw update!\n")
204             updatevalueinstaposts(post_id, likes, comments, cursor, connection)
```

***Εικόνα 60: RecordInstaValueToDB - Posts 3***

Για να συλλέξουμε τα πολυμέσα της δημοσίευσης χρησιμοποιούμε τη συνάρτηση getpostmedia. Αρχικά ελέγχουμε αν υπάρχουν πολλαπλές εικόνες στη δημοσίευσή μας (*Εικόνα 61*). Αν υπάρχουν, για κάθε μία εικόνα ελέγχουμε την ύπαρξη της στη βάση μας και ανάλογα ή δημιουργούμε εγγραφές στους πίνακες “Media” και “Media2Insta” (*Εικόνα 6* και *Εικόνα 19*) ή απλά μία εγγραφή στο συνδετικό πίνακα “Media2Insta”.

```
310 def getpostmedia(dict,post_id,cursor,connection):
311     try:
312         for i in range(0, len(dict["shortcode_media"]["edge_sidecar_to_children"]["edges"])):
313             mediaurl = dict["shortcode_media"]["edge_sidecar_to_children"]["edges"][i]["node"]["display_url"]
314             sqlmedia = """SELECT media_id FROM Media WHERE media_text = '%s'""" % \
315                        (str(mediaurl))
316             cursor.execute(sqlmedia)
317             myresultmedia = cursor.fetchone()
318             if myresultmedia == None:
319                 try:
320                     insertvalueinstamedia(mediaurl, cursor, connection)
321                     sqlmediaid = """SELECT media_id FROM Media WHERE media_text = '%s'""" % \
322                                  (str(mediaurl))
323                     cursor.execute(sqlmediaid)
324                     myresultmediaid = cursor.fetchone()
325                     insertvaluemedia2insta(post_id, myresultmediaid[0], cursor, connection)
326                 except:
327                     print("Error at media insertion!\n")
328             else:
329                 insertvaluemedia2insta(post_id, myresultmedia[0], cursor, connection)
```

***Εικόνα 61: Συνάρτηση getpostmedia 1***

Αν ο πρώτος έλεγχος αποτύχει, ελέγχουμε για το αν υπάρχει βίντεο στη δημοσίευση (*Εικόνα 62*). Αν υπάρχει, ελέγχουμε την ύπαρξη του στη βάση και ακολουθούμε τα ίδια βήματα που ακολουθήσαμε στην προηγούμενη παράγραφο.

```
330     except:
331         if(dict["shortcode_media"]["is_video"] == True):
332             mediaurl = dict["shortcode_media"]["video_url"]
333             sqlmedia = """SELECT media_id FROM Media WHERE media_text = '%s'""" % \
334                       (str(mediaurl))
335             cursor.execute(sqlmedia)
336             myresultmedia = cursor.fetchone()
337             if myresultmedia == None:
338                 try:
339                     insertvalueinstamedia(mediaurl, cursor, connection)
340                     sqlmediaid = """SELECT media_id FROM Media WHERE media_text = '%s'""" % \
341                                  (str(mediaurl))
342                     cursor.execute(sqlmediaid)
343                     myresultmediaid = cursor.fetchone()
344                     insertvaluemedia2insta(post_id, myresultmediaid[0], cursor, connection)
345                 except:
346                     print("To media den mpike.\n")
347             else:
348                 insertvaluemedia2insta(post_id, myresultmedia[0], cursor, connection)
```

***Εικόνα 62: Συνάρτηση getpostmedia 2***

Τέλος, ελέγχουμε για το αν υπάρχει μόνο μία εικόνα στη δημοσίευση (*Εικόνα 63*). Αν υπάρχει, η συνέχεια είναι παρόμοια με τις προηγούμενες δύο περιπτώσεις.

```
349         else:
350             mediaurl = dict["shortcode_media"]["display_url"]
351             sqlmedia = """SELECT media_id FROM Media WHERE media_text = '%s'""" % \
352                       (str(mediaurl))
353             cursor.execute(sqlmedia)
354             myresultmedia = cursor.fetchone()
355             if myresultmedia == None:
356                 try:
357                     insertvalueinstamedia(mediaurl, cursor, connection)
358                     sqlmediaid = """SELECT media_id FROM Media WHERE media_text = '%s'""" % \
359                                  (str(mediaurl))
360                     cursor.execute(sqlmediaid)
361                     myresultmediaid = cursor.fetchone()
362                     insertvaluemedia2insta(post_id, myresultmediaid[0], cursor, connection)
363                 except:
364                     print("To media den mpike.\n")
365             else:
366                 insertvaluemedia2insta(post_id, myresultmedia[0], cursor, connection)
```

***Εικόνα 63: Συνάρτηση getpostmedia 3***

Για να συλλέξουμε τα hashtag της δημοσίευσης χρησιμοποιούμε τη συνάρτηση getposthashtags. Επειδή τα hashtags δεν υπάρχουν στον δυναμικό κώδικα που εξάγαμε και μετατρέψαμε σε JSON αντικείμενο, χρειάζεται μια επιπρόσθετη αναζήτηση στον δυναμικό κώδικα της ιστοσελίδας. Αφού βρούμε τα στοιχεία κώδικα που μας ενδιαφέρουν, τα επεξεργαζόμαστε και κρατάμε μόνο την φράση που υπάρχει σε κάθε hashtag (*Εικόνα 64*).

```
def getposthashtags(post_id,driver,cursor,connection):
    hashtag_xpath = "//a[contains(@href, '/explore/tags')]"
    time.sleep(5)
    post_hashtags = driver.find_elements_by_xpath(hashtag_xpath)
    if len(post_hashtags) > 0:
        for i in range(0, len(post_hashtags)):
            post_hashtag = post_hashtags[i]
            if post_hashtag != None:
                hashtagref = post_hashtag.get_attribute("href")
                hashtag1 = hashtagref.split('tags/')[1]
                hashtag = hashtag1.split('/')[0]
                sqlhashid1 = """SELECT hash_id FROM Hashtag WHERE value = '%s'""" % \
                             (str(hashtag))
                cursor.execute(sqlhashid1)
                myresulthashid = cursor.fetchone()
```

***Εικόνα 64: Συνάρτηση getposthashtags***

Στη συνέχεια ελέγχουμε αν το συγκεκριμένο hashtag υπάρχει στη βάση (*Εικόνα 65*). Αν δεν υπάρχει, προσθέτουμε από μία εγγραφή έκαστος στους πίνακες “Hashtag” και “Hash2Insta” (*Εικόνα 7* και *Εικόνα 20*). Διαφορετικά, προσθέτουμε μία εγγραφή στον πίνακα “Hash2Insta”.

```
                if myresulthashid == None:
                    try:
                        insertvalueinstahashtag(hashtag, cursor, connection)
                        sqlhashid2 = """SELECT hash_id FROM Hashtag WHERE value = '%s'""" % \
                                     (str(hashtag))
                        cursor.execute(sqlhashid2)
                        myresulthashid1 = cursor.fetchone()
                        insertvaluehash2insta(post_id, myresulthashid1[0], cursor, connection)
                    except:
                        print("To hashtag den mpike.\n")
                else:
                    insertvaluehash2insta(post_id, myresulthashid[0], cursor, connection)
```

***Εικόνα 65: Συνάρτηση getposthashtags 2***

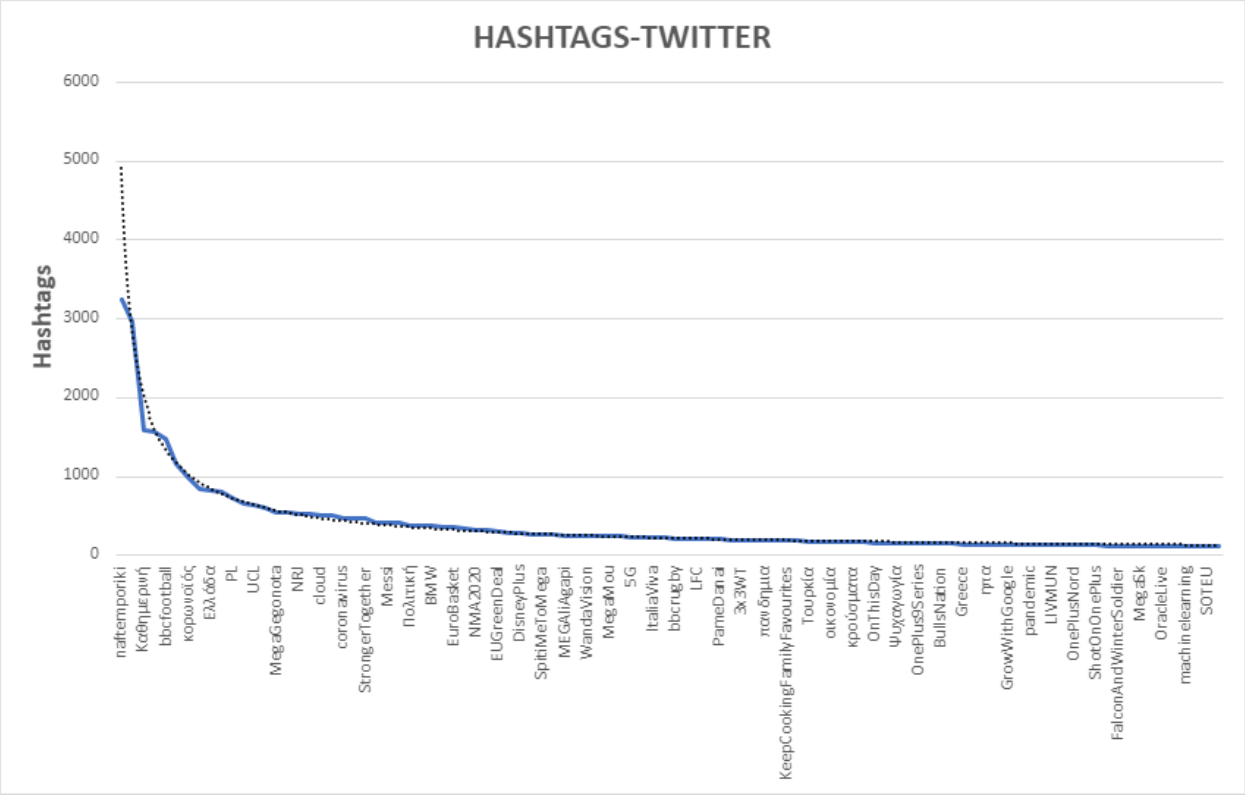
# Ενότητα 5. Αναλύσεις και αποτελέσματα

Σε αυτή την ενότητα θα παραθέσουμε τις αναλύσεις οι οποίες πραγματοποιήθηκαν πάνω στα δεδομένα που συλλέξαμε και αποθηκεύσαμε στη ΒΔ μας.

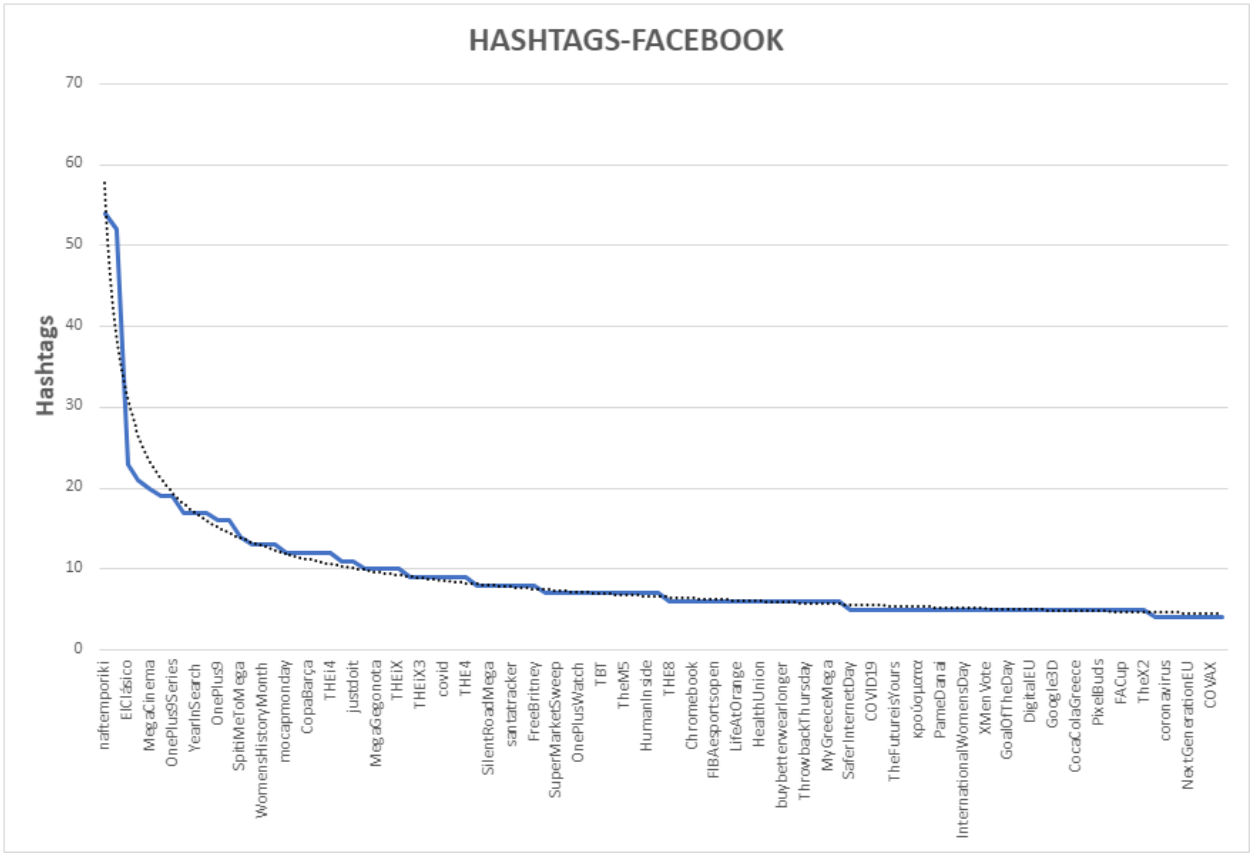
## 5.1. Αναλύσεις στην καθολικότητα των χρηστών

Θα ξεκινήσουμε την ενότητα με την ανάλυση πάνω στα hashtags. Αρχικά, θελήσαμε να αποτυπώσουμε μέσω γραφημάτων τα hashtags στα τρία ΚΔ. Για κάθε μία περίπτωση επιλέξαμε τα κορυφαία 100 hashtags με τις περισσότερες εμφανίσεις στη βάση μας (*Εικόνα 66*, *Εικόνα 67* και *Εικόνα 68*). Στον κάθετο άξονα απεικονίζεται το πλήθος των hashtags, ενώ στον οριζόντιο άξονα απεικονίζονται τα 100 κορυφαία hashtags. Με τη μπλε γραμμή απεικονίζονται τα πλήθη των hashtags, ενώ με την μαύρη διακεκομμένη γραμμή απεικονίζεται η κατανομή νόμου δυνάμεων.

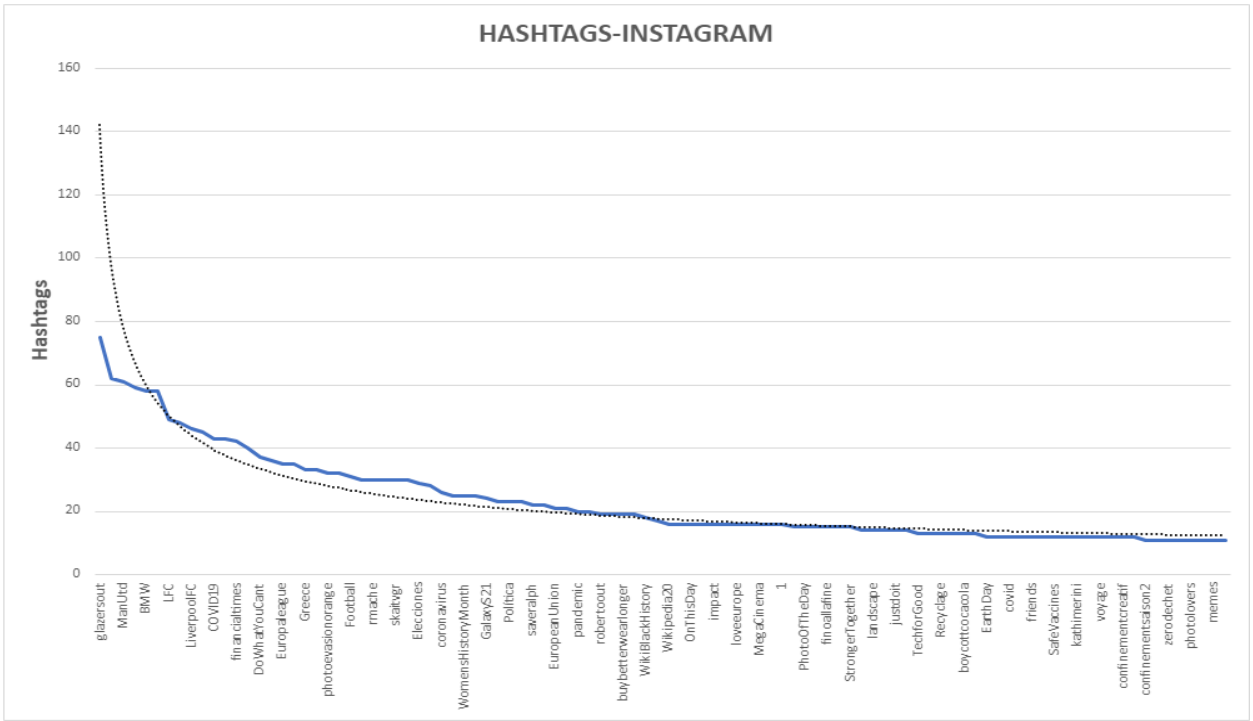
Παρατηρούμε πως και τα τρία γραφήματα ακολουθούν κατανομή δυνάμεων, με το γράφημα του Instagram να μην ακολουθεί πλήρως αυτήν την τάση στις υψηλές τιμές.




***Εικόνα 66: Hashtags - Twitter***



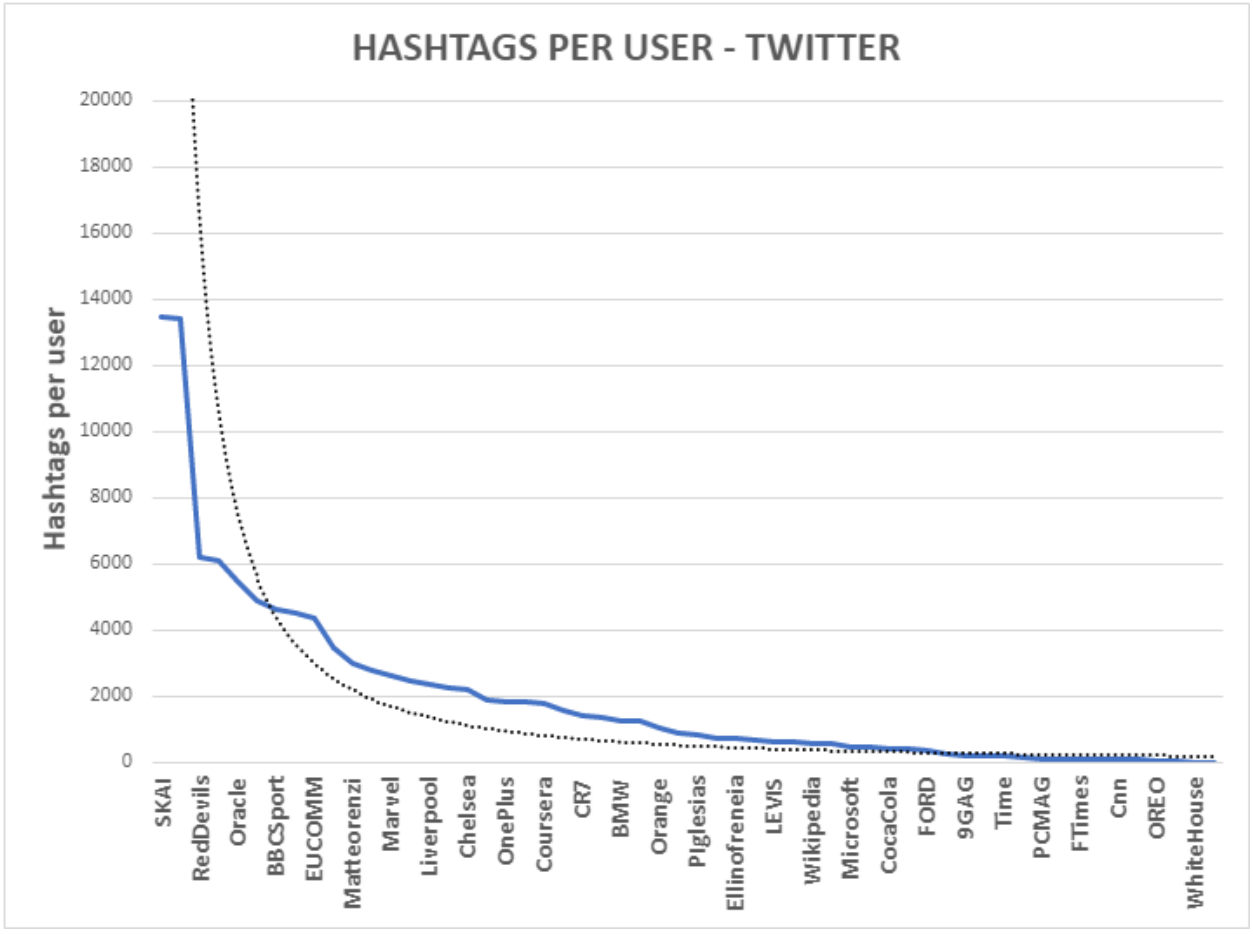

***Εικόνα 67: Hashtags - Facebook***



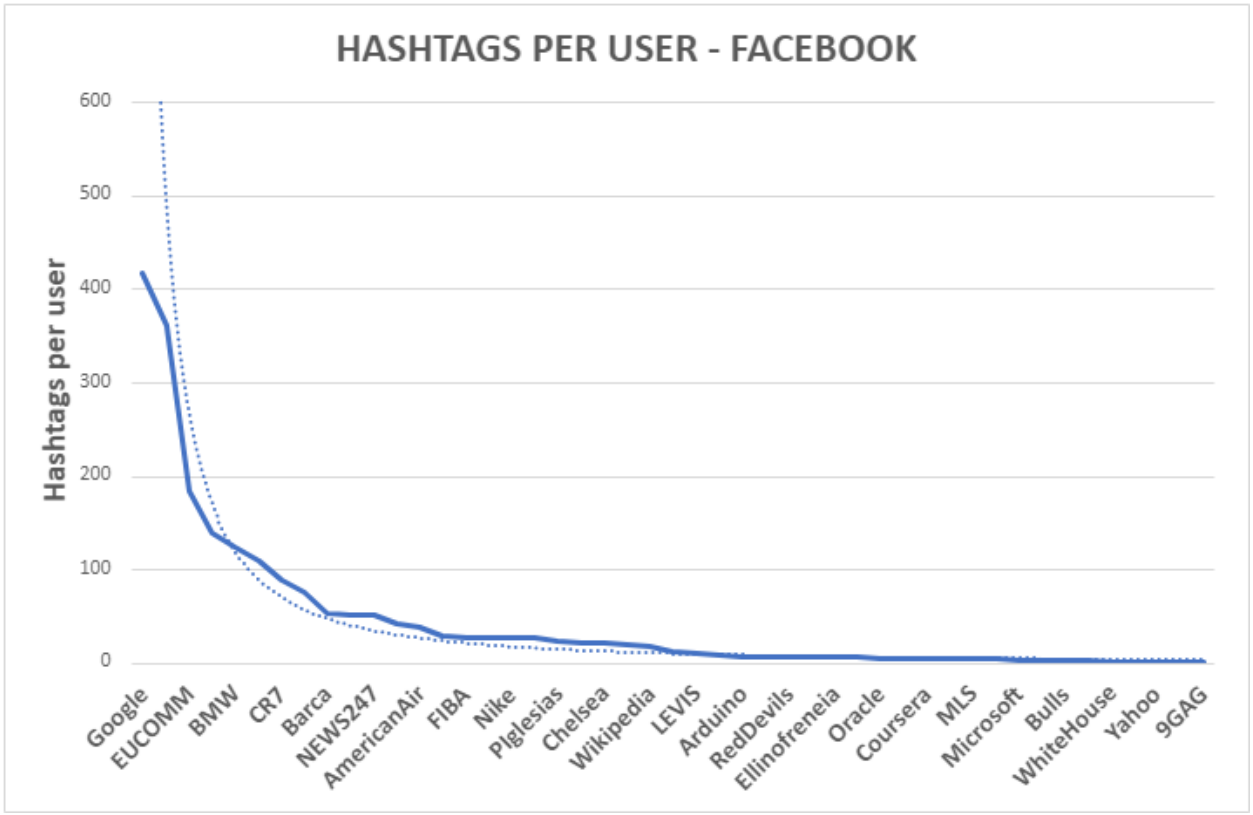

**Εικόνα 68: Hashtags - Instagram**

Την ίδια κατανομή ακολουθούν και τα ακόλουθα τρία γραφήματα (*Εικόνα 69, Εικόνα 70 και Εικόνα 71*). Σε αυτή την περίπτωση οι αναλύσεις γίνονται γύρω από τους χρήστες που δημιούργησαν τα hashtags. Για κάθε ΚΔ τους κατατάξαμε με βάση τον αριθμό των hashtags που σχετίζονται με τις δημοσιεύσεις τους.

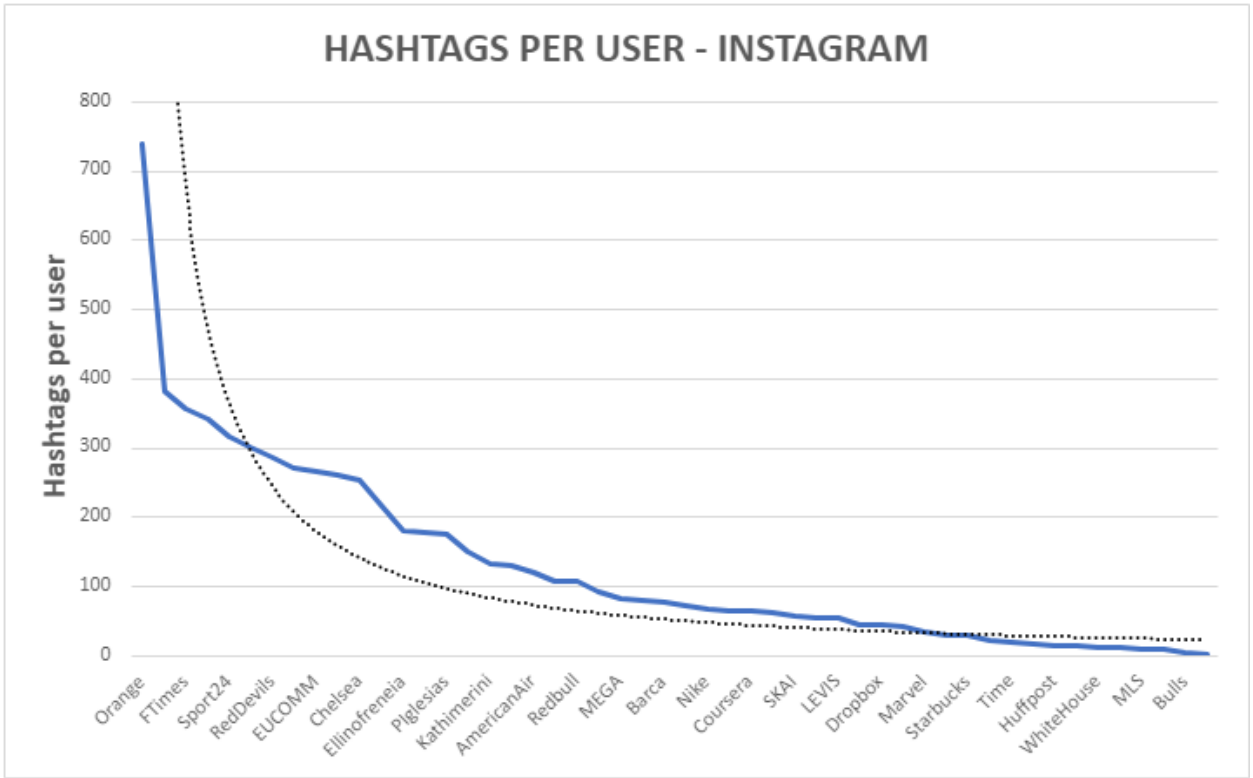
Αξιοπρόσεκτο είναι το γεγονός πως οι τάσεις των δεδομένων είναι παρόμοιες, παρόλο που το μέγεθος των δεδομένων στο Facebook και στο Instagram είναι σαφώς λιγότερο.




***Εικόνα 69: Hashtags Per User - Twitter***

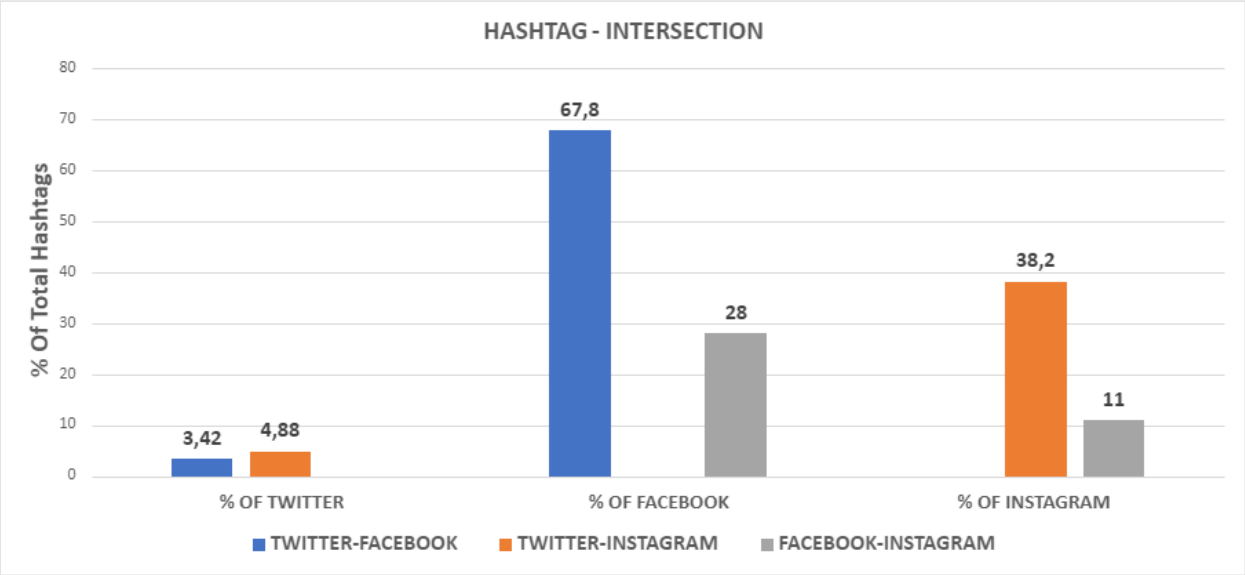



***Εικόνα 70: Hashtags Per User – Facebook***




***Εικόνα 71: Hashtags Per User - Instagram***

Στη συνέχεια, λαμβάνοντας υπόψη πως όλα τα hashtags έχουν προέλθει από τις δημοσιεύσεις των ιδίων χρηστών, θελήσαμε να δούμε πόσα hashtags που ανήκουν σε ένα ΚΔ υπάρχουν και στα άλλα δύο. Τα αποτελέσματα αυτής της ανάλυσης φαίνονται στην *Εικόνα 72*. Συγκεκριμένα βλέπουμε πως το 67,8% του συνολικού πλήθους hashtags του Facebook εμφανίζεται στο Twitter, όμως μόνο το 3,42% του πλήθους hashtags του Twitter εμφανίζεται στο Facebook. Αυτή η ποσοστιαία διαφορά οφείλεται στον μεγάλο όγκο δεδομένων που συλλέχθηκαν στην περίπτωση του Twitter. Η περίπτωση στην οποία αξίζει να επικεντρωθούμε είναι αυτή της τομής των hashtags του Facebook με των hashtags του Instagram. Παρατηρείται πως αυτά τα δύο ΚΔ παρουσιάζουν μικρότερες διαφορές μεταξύ τους (28% με 11%) σε σύγκριση με τις υπόλοιπες τομές.

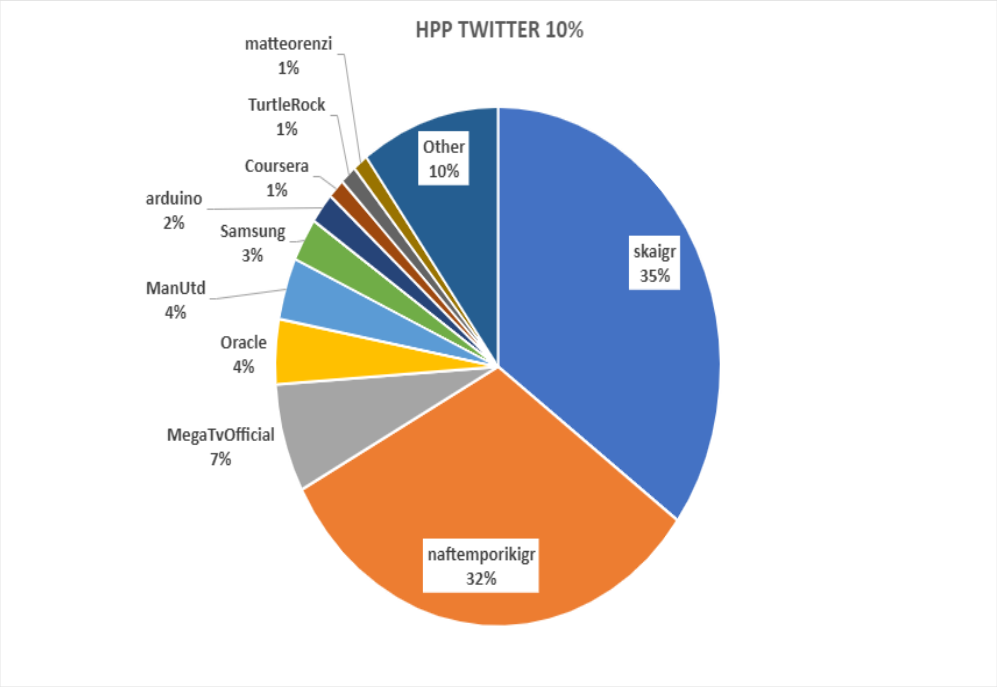



***Εικόνα 72: Intersection of Hashtags***

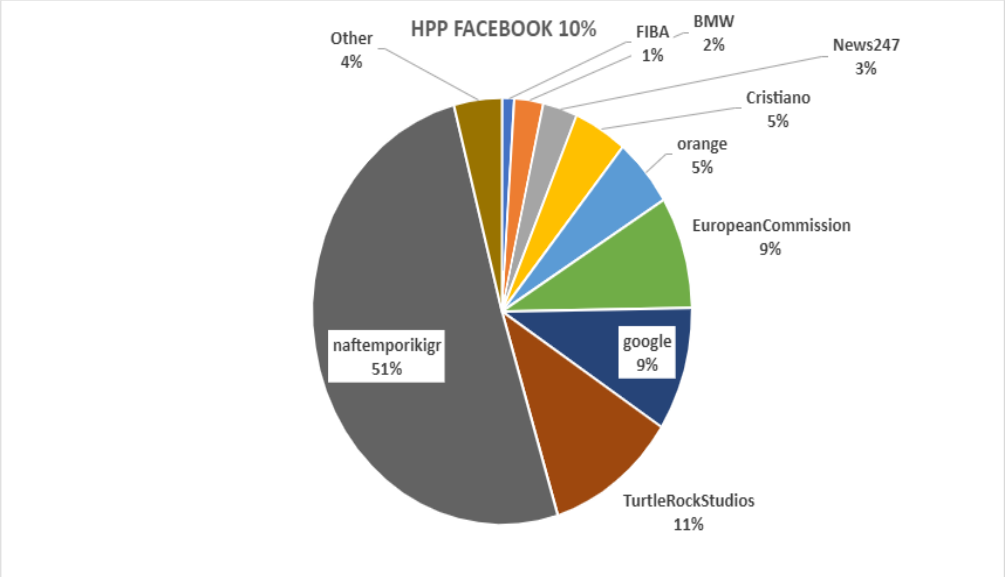
Στην επόμενη ανάλυση, θελήσαμε να δούμε ποιοι χρήστες συνδέονται με το 10% των δημοσιεύσεων με τα περισσότερα hashtags σε κάθε δίκτυο. Στην περίπτωση του Twitter 10 χρήστες έχουν μερίδιο αυτών των δημοσιεύσεων μεγαλύτερο από 1%, στο Facebook 9 χρήστες κατέχουν μερίδιο μεγαλύτερο του 1%, ενώ στο Instagram οι χρήστες είναι 14. Οι υπόλοιποι χρήστες που έχουν μερίδιο κάτω της μονάδας συμπεριλαμβάνονται στην κατηγορία “Other” κάθε γραφήματος πίτας (*Εικόνα 73, Εικόνα 74 και Εικόνα 75*).

Παρατηρούμε πως υπάρχει μόνο ένας χρήστης, ο “naftemporikigr”, ο οποίος κατέχει θέση και στα τρία γραφήματα, με 2 χρήστες να έχουν θέση στο Twitter και στο Instagram, ένας χρήστης να κατέχει θέση στο Twitter και στο Facebook, ενώ τρεις χρήστες κατέχουν παράλληλα θέση στο Facebook και στο Instagram.

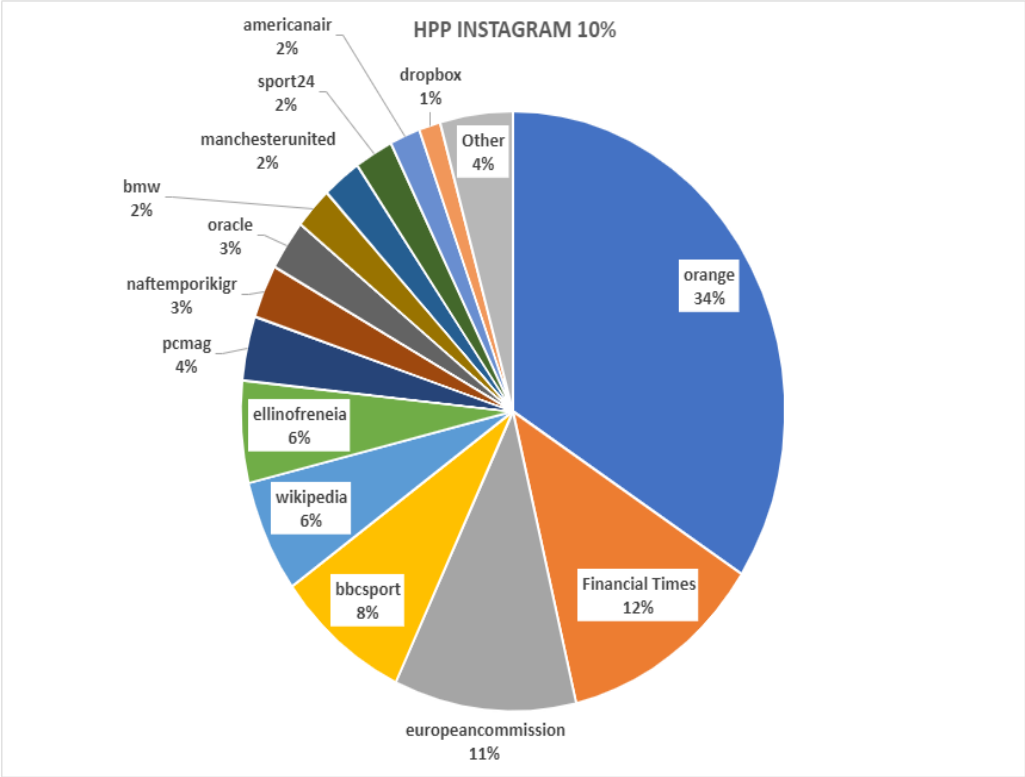
Ένα σημείο που πρέπει να σταθούμε είναι το γεγονός πως στην περίπτωση του Twitter, τις τρεις πρώτες θέσεις τις καταλαμβάνουν χρήστες οι οποίοι σχετίζονται με μέσα ενημέρωσης και ψυχαγωγίας, με συνολικό ποσοστό 74%.




***Εικόνα 73: Top 10% Hashtags per post - Users – Twitter***

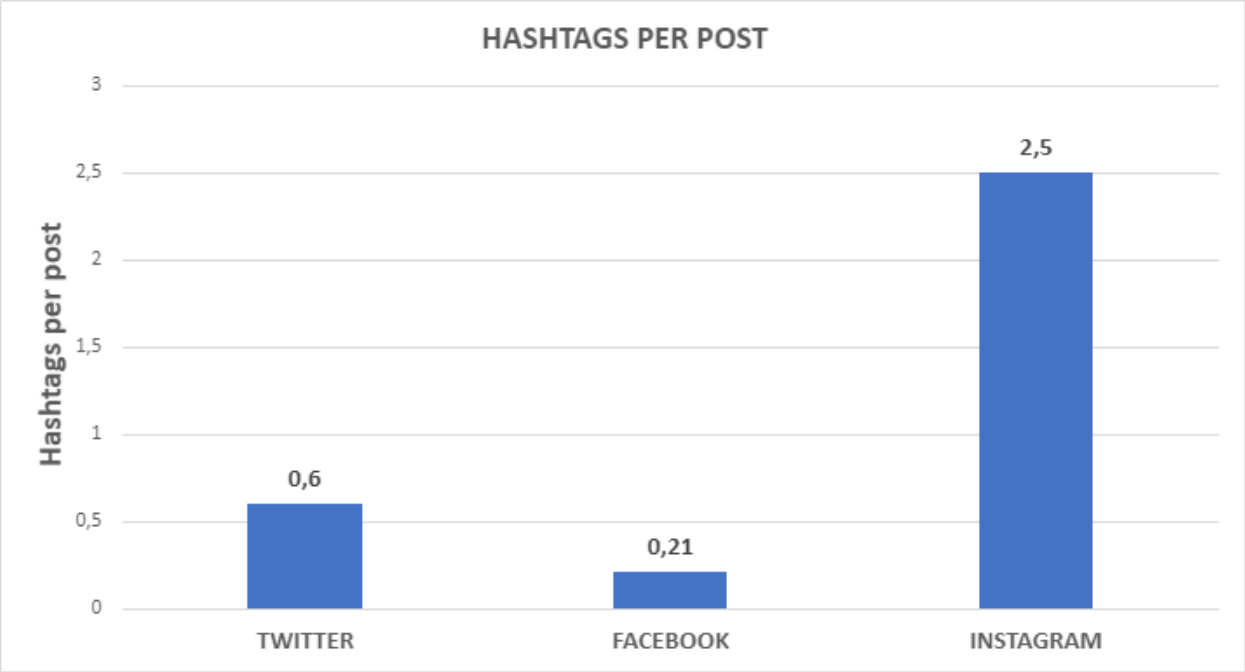



***Εικόνα 74: Top 10% Hashtags Per Post - Users - Facebook***




***Εικόνα 75: Top 10% Hashtags per post - Users – Instagram***

Στην ανάλυση της *Εικόνα 76* βλέπουμε τον αριθμό των hashtags που αναλογούν κατά μέσο όρο σε μία δημοσίευση των τριών ΚΔ. Παρατηρούμε πως η χρήση των hashtags στο Facebook δεν είναι ιδιαίτερα διαδεδομένη με μόλις 0.2 hashtag ανά δημοσίευση. Αντιθέτως στο Instagram βλέπουμε μια εμφατική διαφορά με 2.5 hashtag ανά δημοσίευση, τετραπλάσια σχεδόν από το Twitter που κατατάσσεται δεύτερο.

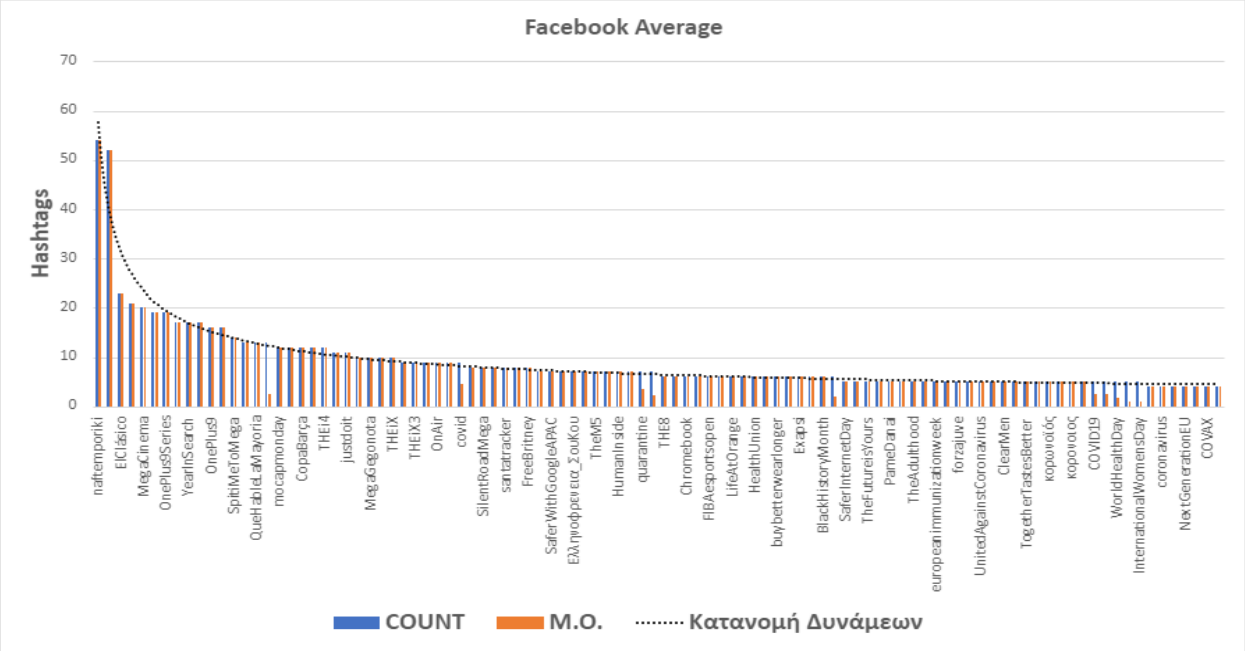



***Εικόνα 76: Hashtags Per Post***

Κλείνοντας με την ανάλυση των hashtags, θελήσαμε να αναλύσουμε το πόσοι χρήστες χρησιμοποιούν τα 100 κορυφαία hashtags σε κάθε ΚΔ. Ή αν το δούμε από την άλλη πλευρά πόσα από τα 100 κορυφαία hashtags είναι επαναχρησιμοποιούμενα.

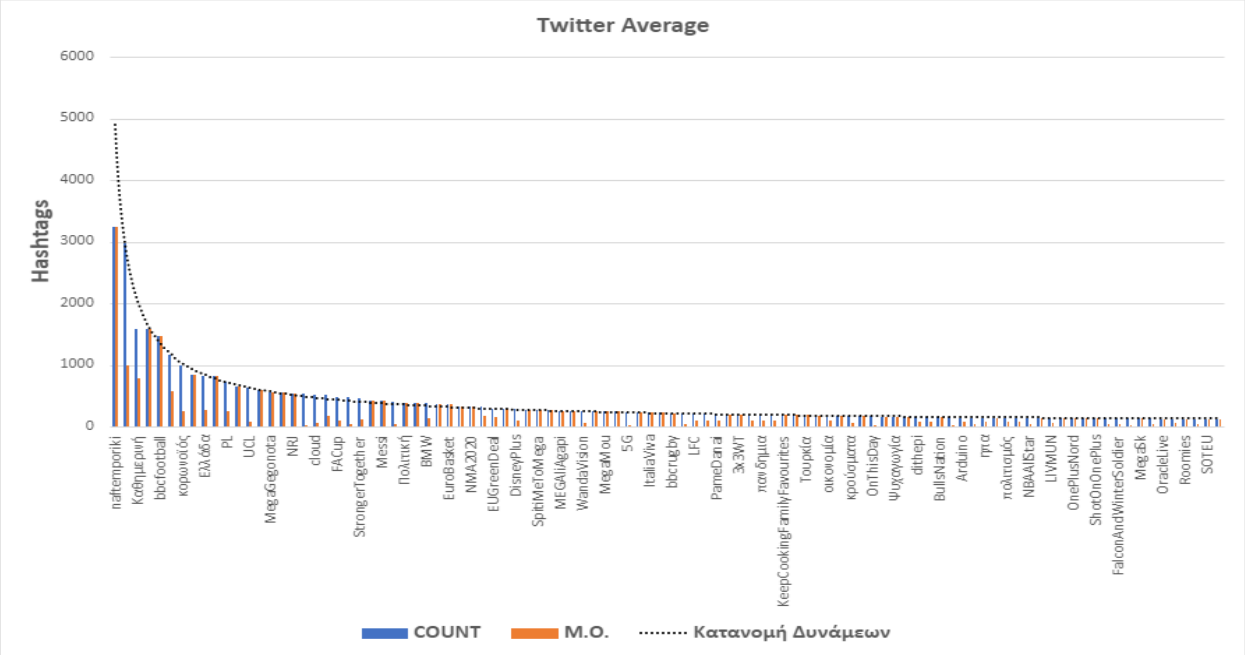
Στα παρακάτω τρία γραφήματα (*Εικόνα 77, Εικόνα 78, Εικόνα 79*), στον κάθετο άξονα παρουσιάζεται το πλήθος των hashtags, ενώ στον οριζόντιο άξονα παρουσιάζονται τα 100 κορυφαία hashtags. Με μπλε στήλες παρουσιάζονται οι συνολικές εγγραφές κάθε hashtag, ενώ με τις πορτοκαλί στήλες εμφανίζονται οι μέσοι όροι από εγγραφές που αναλογούν σε κάθε χρήστη που χρησιμοποίησε το εκάστοτε hashtag. Τέλος, με μαύρη διακεκομμένη γραμμή εμφανίζεται η κατανομή νόμου δύναμης.

Στην *Εικόνα 77* βλέπουμε την περίπτωση του Facebook, όπου παρατηρούμε πως πέρα από ελάχιστες εξαιρέσεις, οι τιμές των συνολικών εγγραφών με αυτές των μέσων όρων ταυτίζονται, γεγονός που οφείλεται στο ότι τα hashtags δεν είναι τόσο διαδεδομένα σε αυτό το ΚΔ , κάτι που δείξαμε στην προηγούμενη ανάλυση (*Εικόνα 76*). Πέρα από τις αυξομειώσεις παρατηρούμε πως σε αυτό το γράφημα έχουμε δύο τιμές οι οποίες διαφέρουν κατά πολύ από τις άλλες, αντί για μία που συνήθως ορίζει η κατανομή νόμου δυνάμεων.



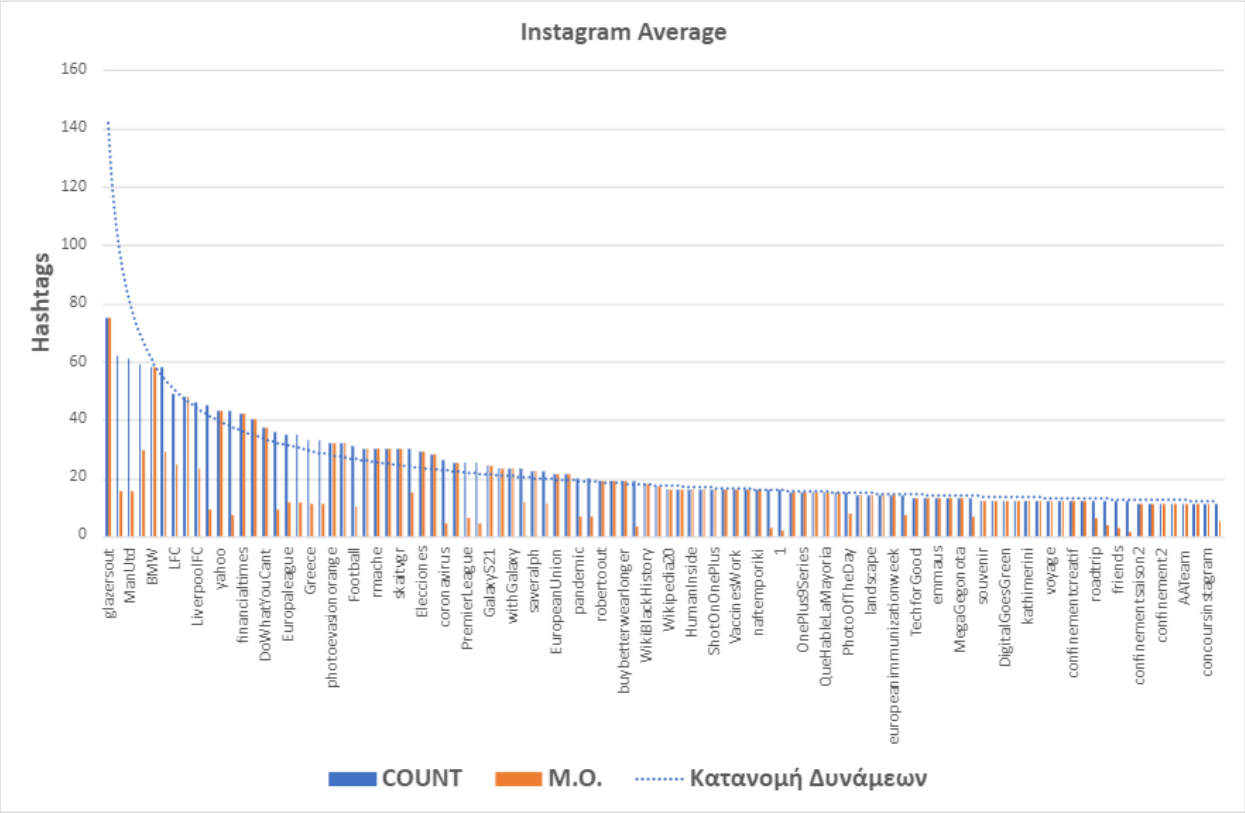

***Εικόνα 77: Average Hashtags - Facebook***

Στην *Εικόνα 78* βλέπουμε την περίπτωση του Twitter. Παρατηρούμε πως όσο ανεβαίνουμε στην κατάταξη, υπάρχουν μεγάλες μειώσεις των μέσων όρων hashtags ανά χρήστη. Αυτό σημαίνει πως τα hashtags με τις περισσότερες εμφανίσεις στη βάση είναι και πιο trending ανάμεσα στους χρήστες.



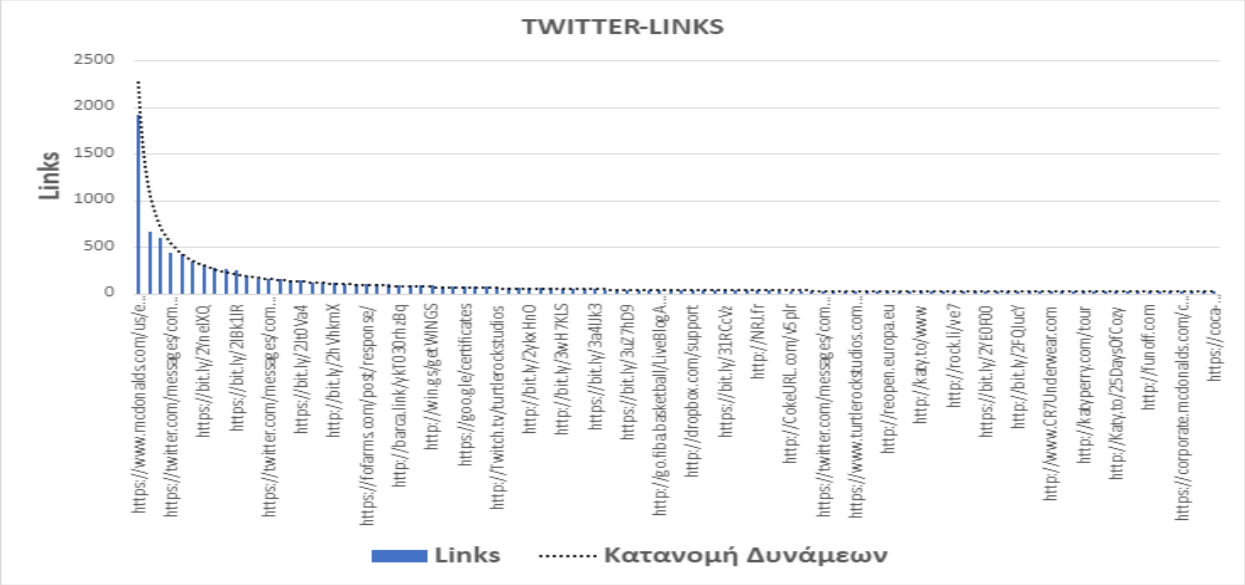

***Εικόνα 78: Average Hashtags - Twitter***

Στην *Εικόνα 79* βλέπουμε την περίπτωση του Instagram, του ΚΔ που όπως αναφέραμε προηγουμένως έχει την μεγαλύτερη χρήση hashtags σε σχέση με τα άλλα δύο δίκτυα. Εδώ παρατηρούμε πως σε όλο το εύρος του γραφήματος υπάρχουν διαφορές ανάμεσα στις τιμές των συνολικών εγγραφών με αυτές των μέσων όρων. Επίσης παρατηρούμε πως υπάρχουν τιμές που υπερβαίνουν το όριο της κατανομής νόμου δυνάμεων. Με αυτή την παρατήρηση συμπεραίνουμε πως στο δίκτυο που τα hashtags είναι ένα από τα κύρια metadata του, υπάρχει μια τάση να επιλέγονται ίδια hashtags τα οποία μάλλον έχουν υψηλό ενδιαφέρον και μεγαλύτερη τάση για επισκεψιμότητα.

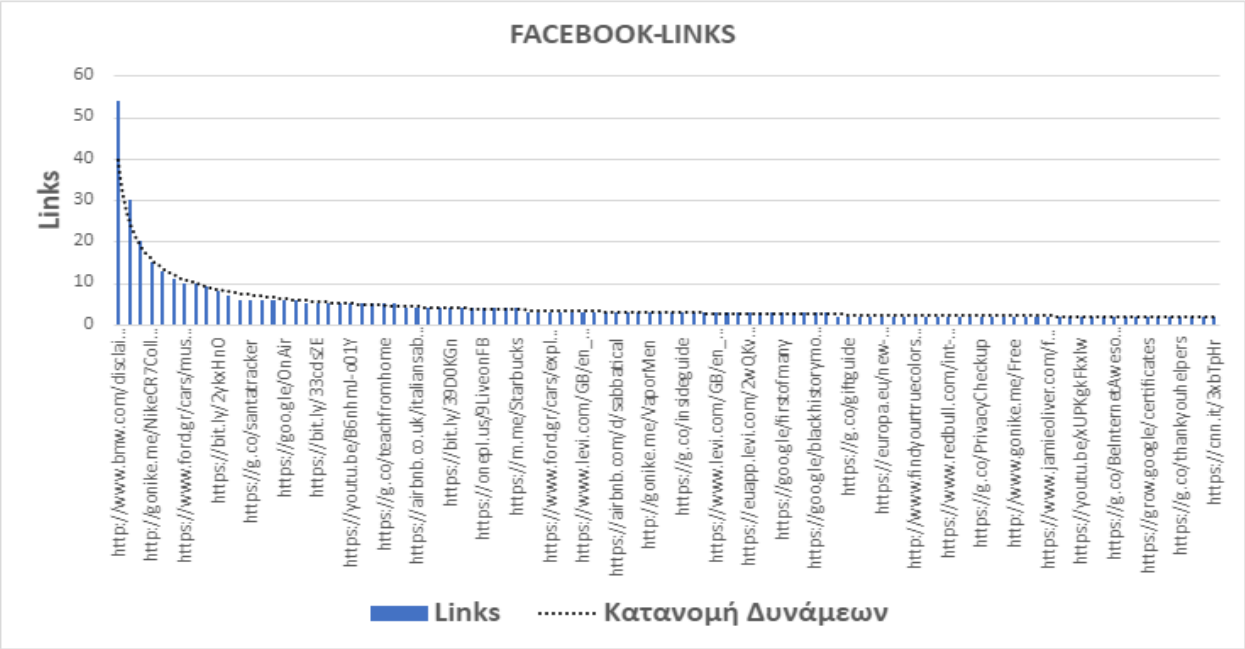



***Εικόνα 79: Average Hashtags - Instagram***

Στη συνέχεια, θα περιγράψουμε τις αναλύσεις που αφορούν τους υπερσυνδέσμους των δημοσιεύσεων της βάσης δεδομένων. Υπερσύνδεσμοι βρέθηκαν μόνο στο Twitter και το Facebook, καθώς το Instagram δεν παρέχει τέτοια πληροφορία στις δημοσιεύσεις των χρηστών. Στην *Εικόνα 80* καθώς και στην *Εικόνα 81*, στον κάθετο άξονα παρουσιάζουμε το πλήθος των εγγραφών ενός υπερσυνδέσμου και στον οριζόντιο άξονα παρουσιάζουμε τους υπερσυνδέσμους. Με μαύρη διακεκομμένη γραμμή ορίζεται η κατανομή νόμου δυνάμεων. Και στις δύο περιπτώσεις τα γραφήματα ακολουθούν την κατανομή.

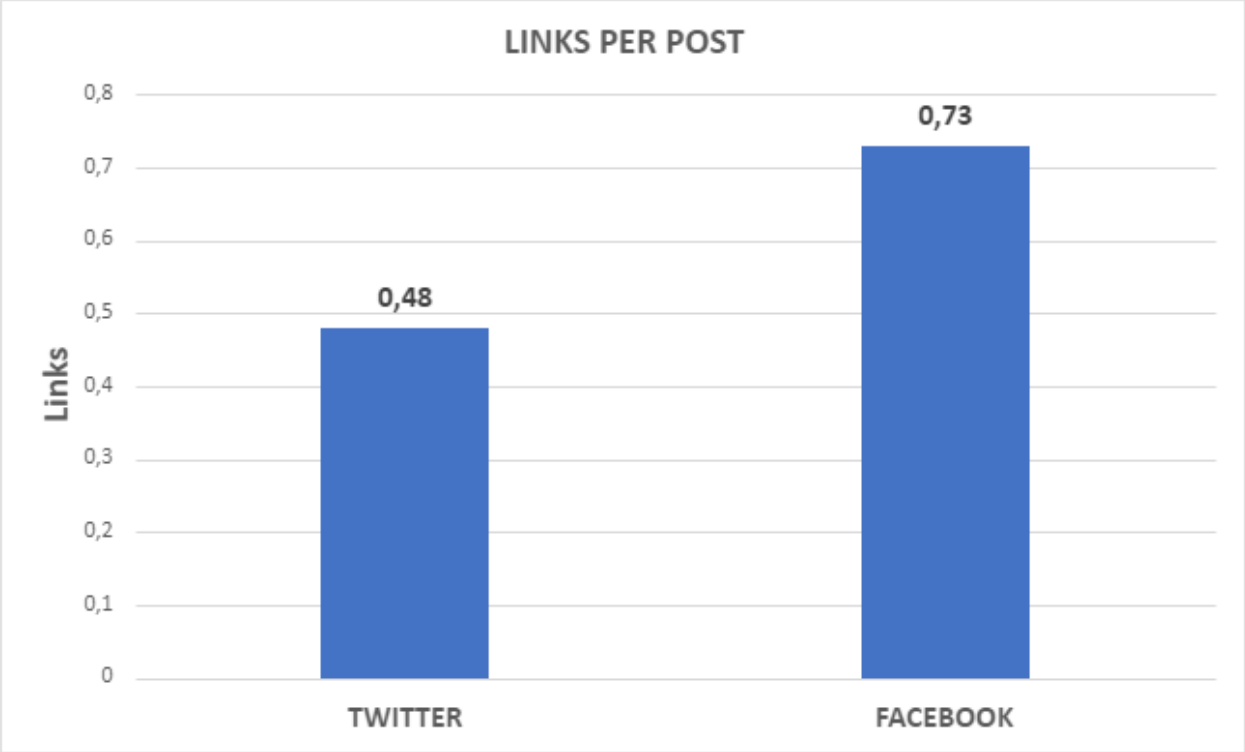



***Εικόνα 80: Links - Twitter***




***Εικόνα 81: Links - Facebook***

Στην *Εικόνα 82* βλέπουμε τον μέσο όρο από υπερσυνδέσμους που δύνανται να βρεθούν σε κάποια δημοσίευση κάθε ΚΔ. Παρατηρούμε πως υπάρχει μια σχετική διαφορά υπέρ του Facebook της τάξης του 52%, κάτι που ίσως υποδεικνύει την προτίμηση των χρηστών αυτού του ΚΔ προς αυτήν την οντότητα. Σε μελλοντική εργασία θα προσπαθήσουμε να συλλέξουμε περισσότερα δεδομένα από το Facebook ώστε μα μειωθεί η διαφορά όγκου δεδομένων για την πιστότερη μέτρηση των μέσων όρων.

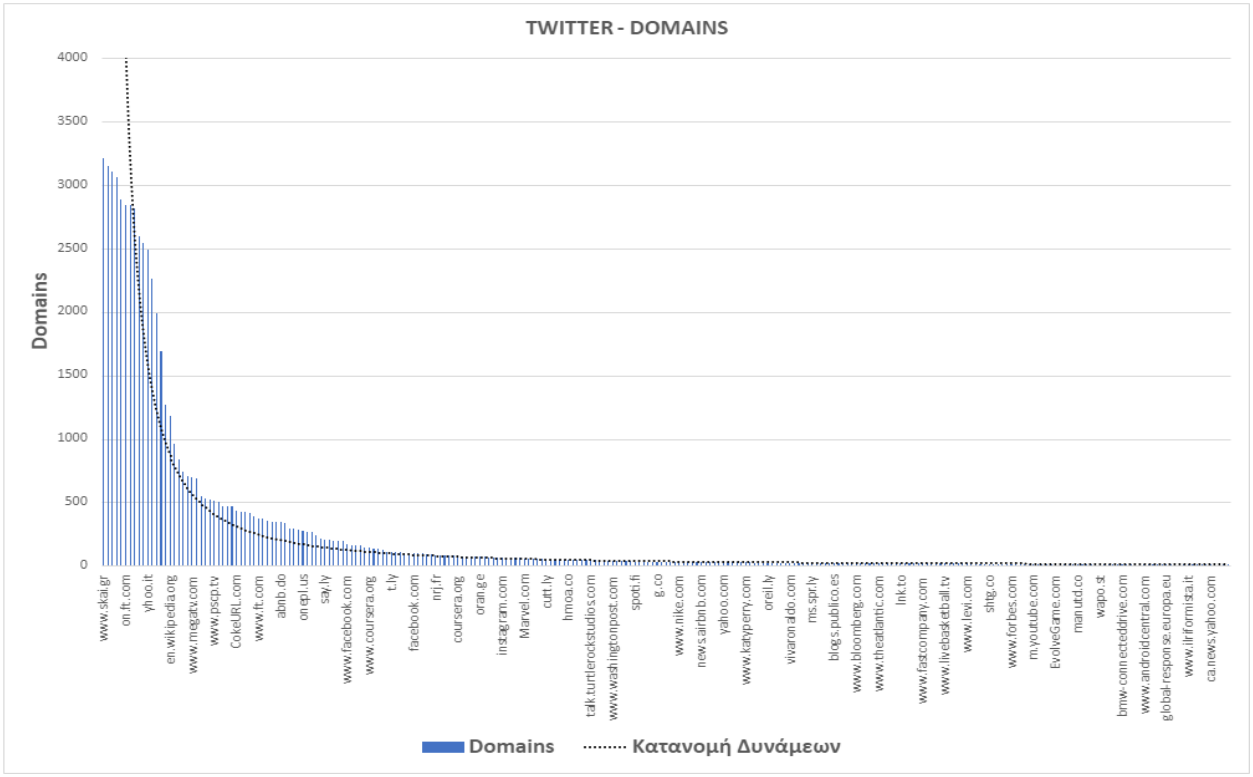



***Εικόνα 82: Links Per Post***

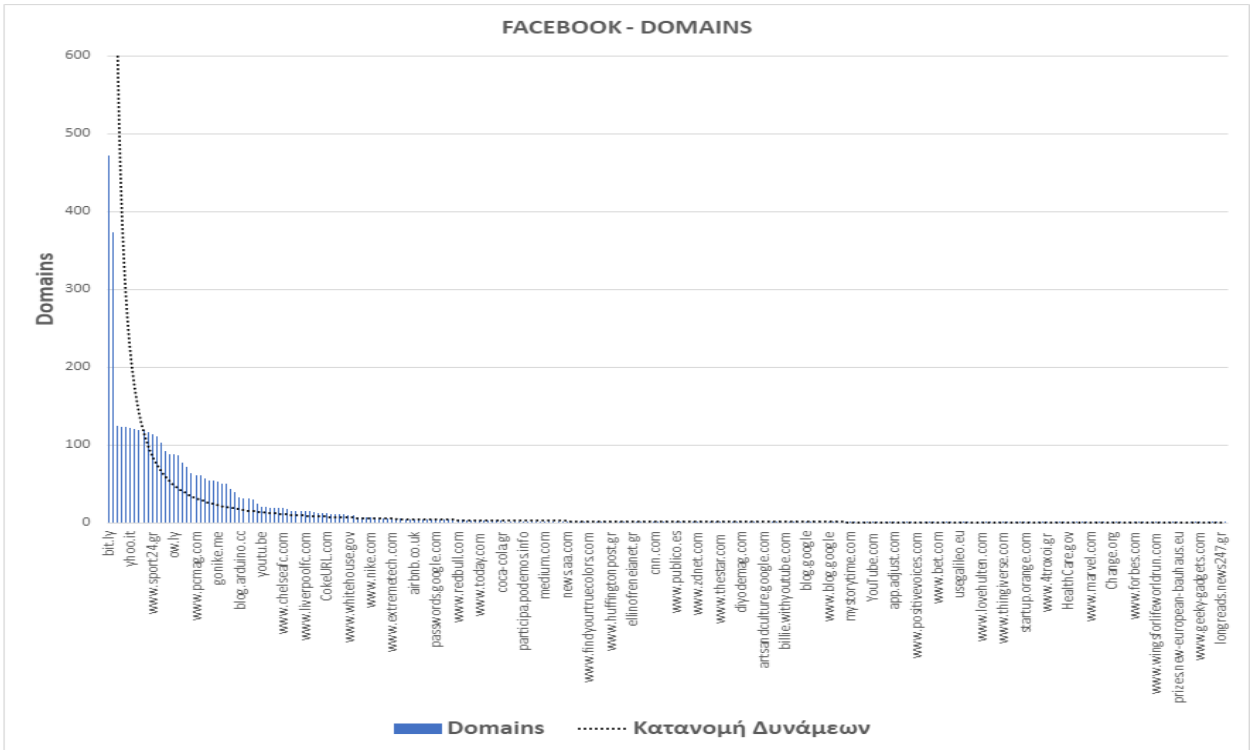
Μια φυσικά επακόλουθη ανάλυση ήταν αυτή γύρω από τα domains των συνδέσμων αυτών. Στην *Εικόνα 83* και στην *Εικόνα 84* βλέπουμε τα γραφήματα κατάταξης των domains ανάλογα με την εμφάνισή τους σε συνδέσμους. Στον κάθετο άξονα παρουσιάζεται το πλήθος εγγραφών ενός domain, ενώ στον οριζόντιο άξονα παρουσιάζονται τα domains. Με μαύρη διακεκομμένη γραμμή ορίζεται η κατανομή νόμου δύναμης. Σε αμφότερα τα γραφήματα παρατηρείται μια τάση των δεδομένων να ακολουθούν την κατανομή.

Πιο συγκεκριμένα, στο γράφημα του Twitter παρατηρείται πως τα πρώτα πέντε domains έχουν παρόμοιο πλήθος εγγραφών, ενώ ελάχιστα domains έχουν μεγαλύτερο πλήθος εγγραφών από αυτό που ορίζει η κατανομή.

Στο γράφημα του Facebook μόνο τα πρώτα δύο domains εμφανίζουν μεγάλες διαφορές σε σχέση με τα υπόλοιπα. Επίσης, και σε αυτή την περίπτωση, ελάχιστα domains εμφανίζουν πλήθος μεγαλύτερο από αυτό που ορίζει η κατανομή, τα οποία τοποθετούνται γύρω από το σημείο που η κατανομή νόμου δυνάμεων αρχίζει να αυξάνεται σημαντικά.

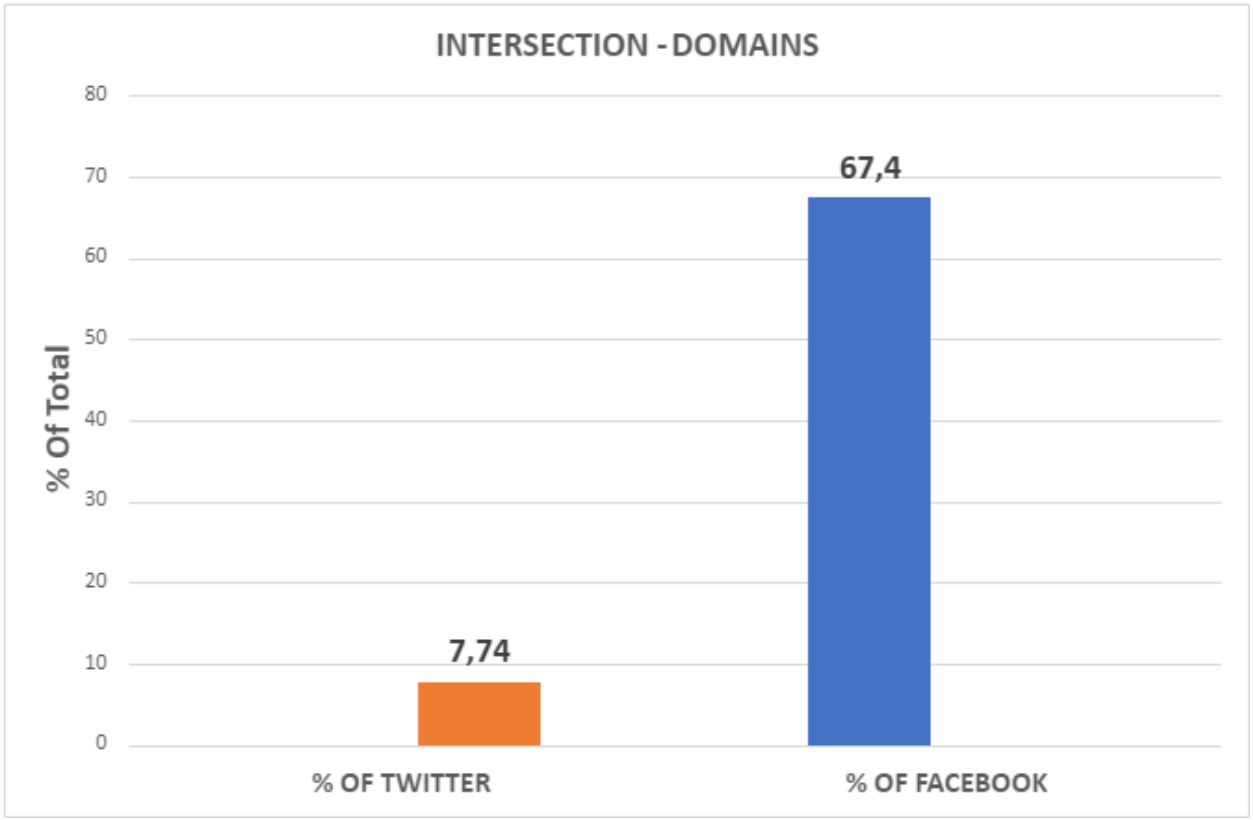



***Εικόνα 83: Domains - Twitter***



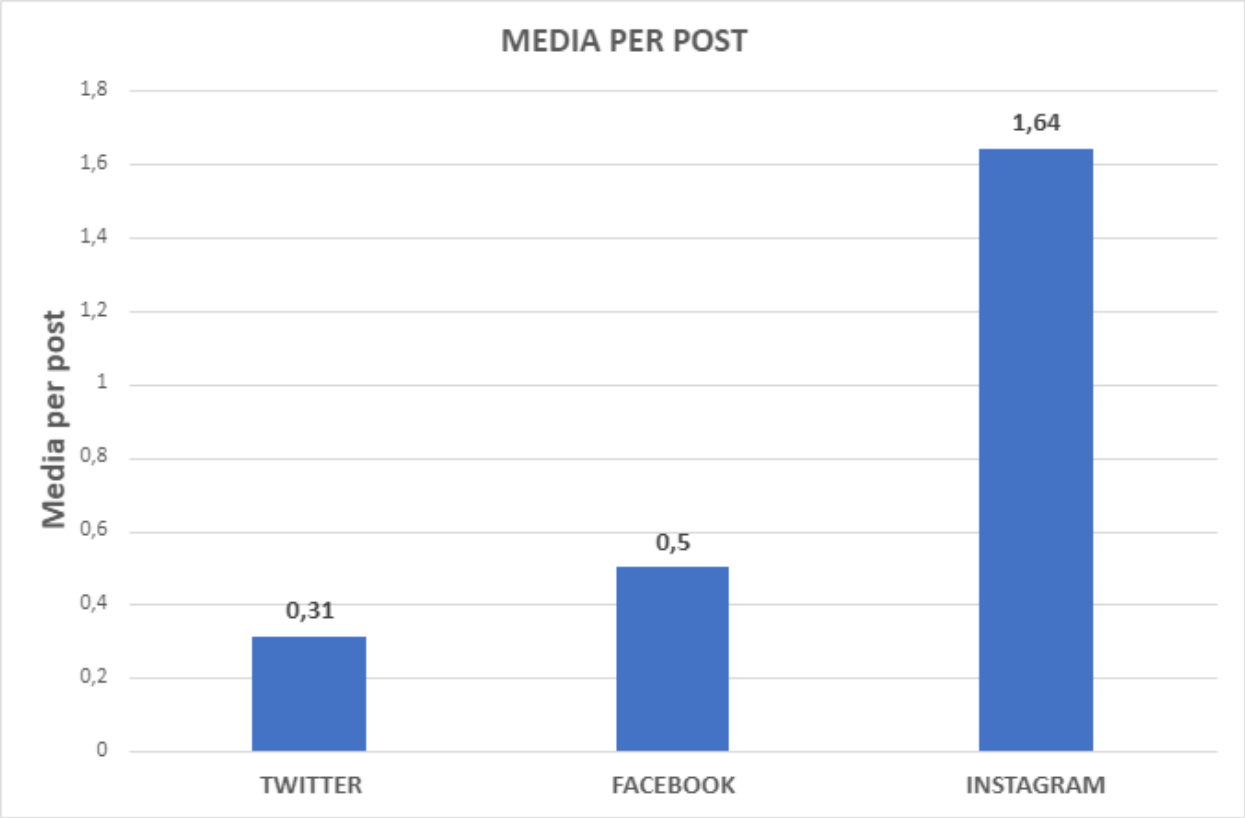

***Εικόνα 84: Domains - Facebook***

Στη συνέχεια, θελήσαμε να δούμε τα κοινά domains που βρέθηκαν και στα δύο ΚΔ. Όπως παρατηρούμε στην *Εικόνα 85*, το 67,4% του συνολικού πλήθους domains του Facebook εμφανίζεται στο Twitter, όμως μόνο το 7,74% του πλήθους domains του Twitter εμφανίζεται στο Facebook. Με αυτή την παρατήρηση συμπεραίνουμε πως στο Twitter υπάρχει μεγαλύτερη ποικιλομορφία όσον αφορά τα domains. Η διαφορά στην ποικιλομορφία αυτή σε μεταγενέστερη έρευνα όταν ο όγκος των δεδομένων του Facebook φτάσει τα επίπεδα του Twitter θα είναι πιο ξεκάθαρη.




***Εικόνα 85: Intersection of Domains***

Ακολουθεί η ανάλυση των πολυμέσων (Εικόνες και βίντεο) της ΒΔ. Στην *Εικόνα 86* απεικονίζονται οι μέσοι όροι πολυμέσων που μπορεί να υπάρχουν σε μια δημοσίευση καθενός από τα τρία ΚΔ. Παρατηρούμε πως η χρήση των πολυμέσων στο Twitter δεν είναι διαδεδομένη με μόλις 0,31 πολυμέσα ανά δημοσίευση. Αντιθέτως παρατηρούμε μια ξεκάθαρη επικράτηση του Instagram με 1,64 πολυμέσα ανά δημοσίευση, τριπλάσια σχεδόν από το Facebook που κατατάσσεται δεύτερο. Μέσα από αυτό το γράφημα φαίνεται και η πολιτική που ακολουθεί το Instagram γύρω από το περιεχόμενο και το ύφος των δημοσιεύσεων που υπάρχουν σε αυτό.

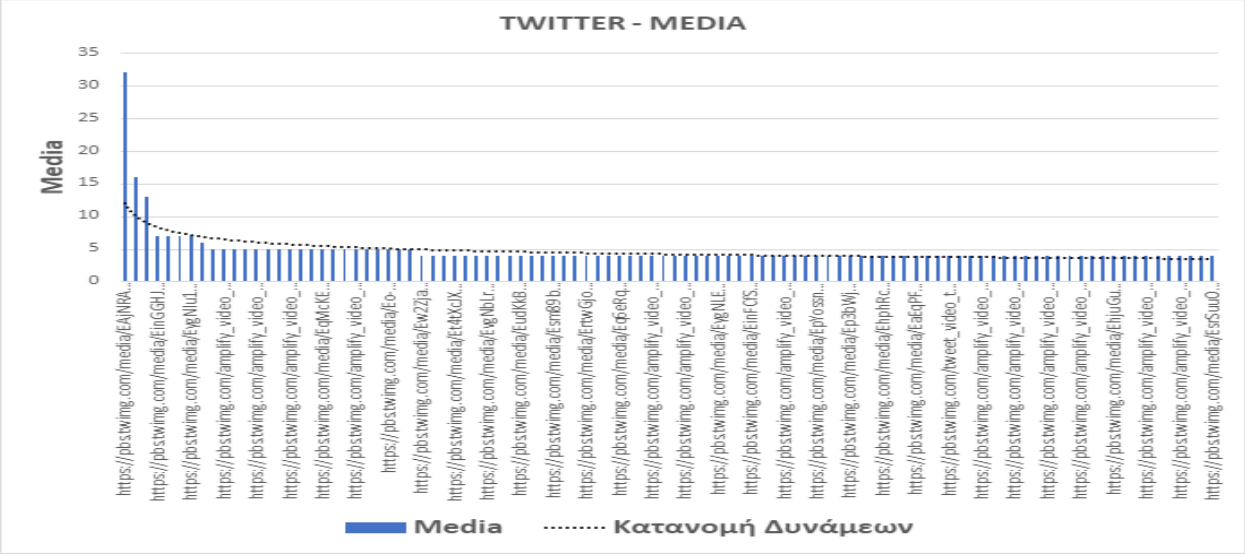



***Εικόνα 86: Media Per Post***

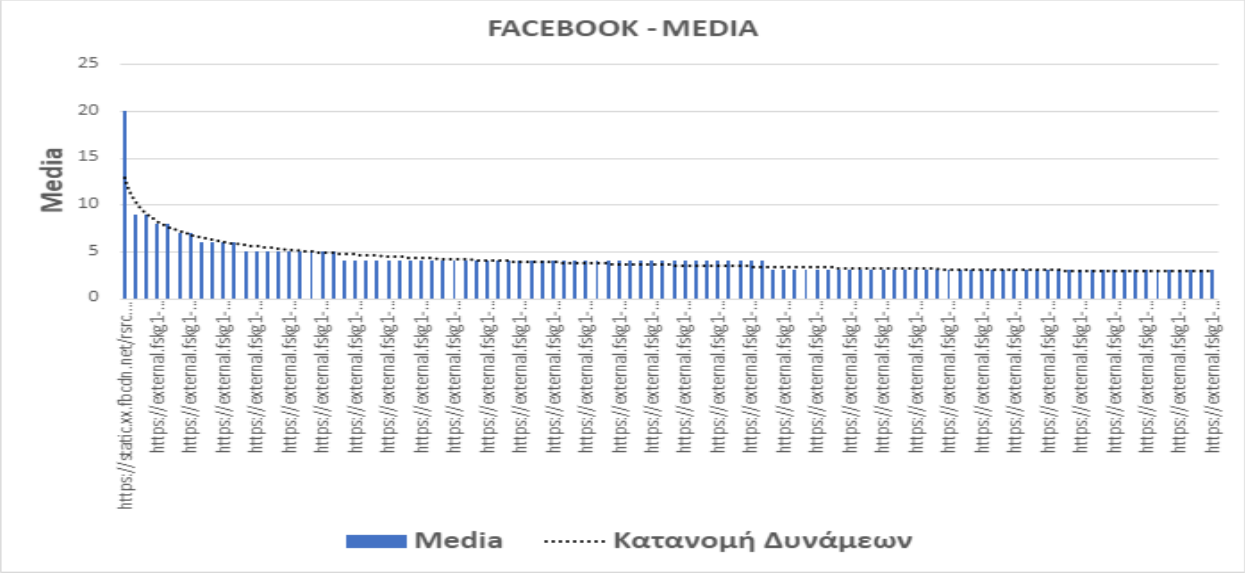
Στην Εικόνα 87 και *Εικόνα 88* βλέπουμε τα γραφήματα κατάταξης των πολυμέσων στο Twitter και το Facebook. Δεν δημιουργήθηκε γράφημα για την περίπτωση του Instagram, καθώς δεν υπάρχουν πολυμέσα του στη βάση με περισσότερες από μία εμφανίσεις, κατάληξη της επιλογής του ΚΔ να δημιουργεί ξεχωριστά URLs για κάθε εικόνα ή βίντεο.

Στον κάθετο άξονα των δύο γραφημάτων παρουσιάζεται το πλήθος των εγγραφών ενός πολυμέσου, ενώ στον οριζόντιο άξονα παρουσιάζονται τα πολυμέσα. Με μαύρη διακεκομμένη γραμμή ορίζεται η κατανομή νόμου δύναμης.

Και σε αυτή την περίπτωση παρατηρείται η τάση των δύο γραφημάτων να ακολουθούν την κατανομή.

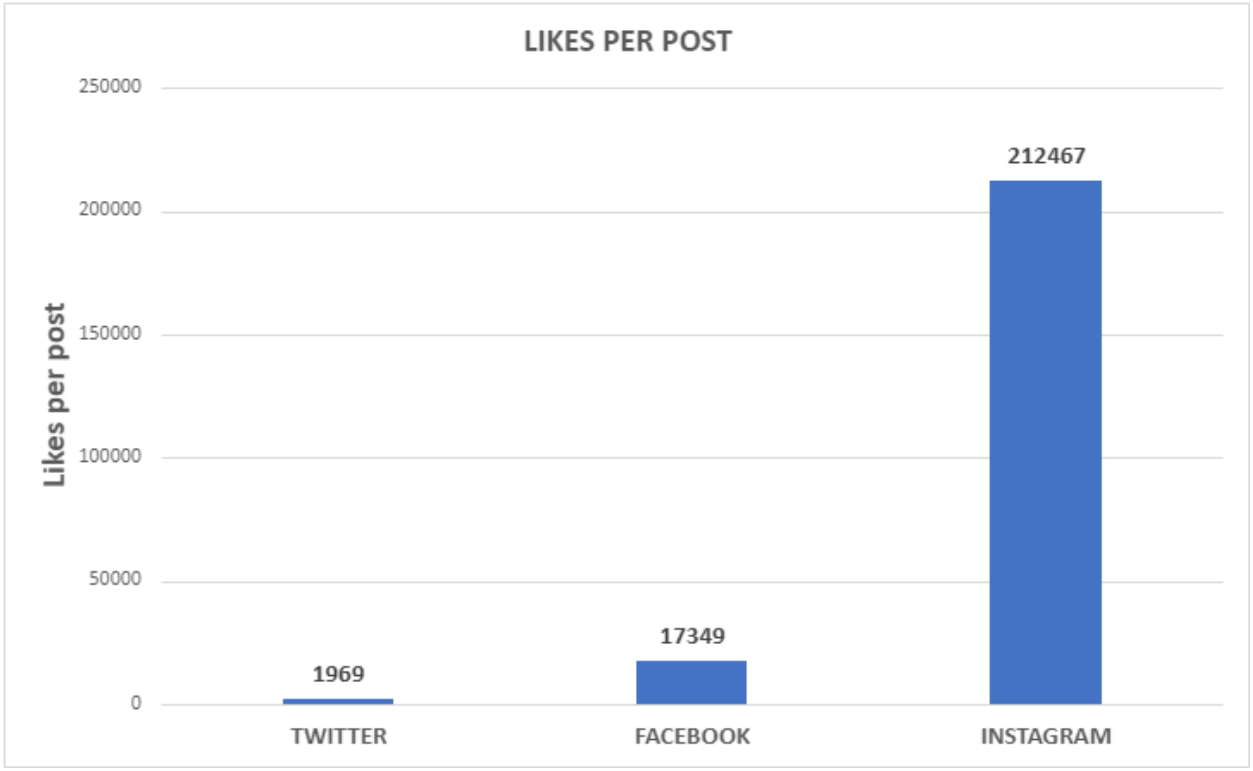



***Εικόνα 87: Media – Twitter***




***Εικόνα 88: Media - Facebook***

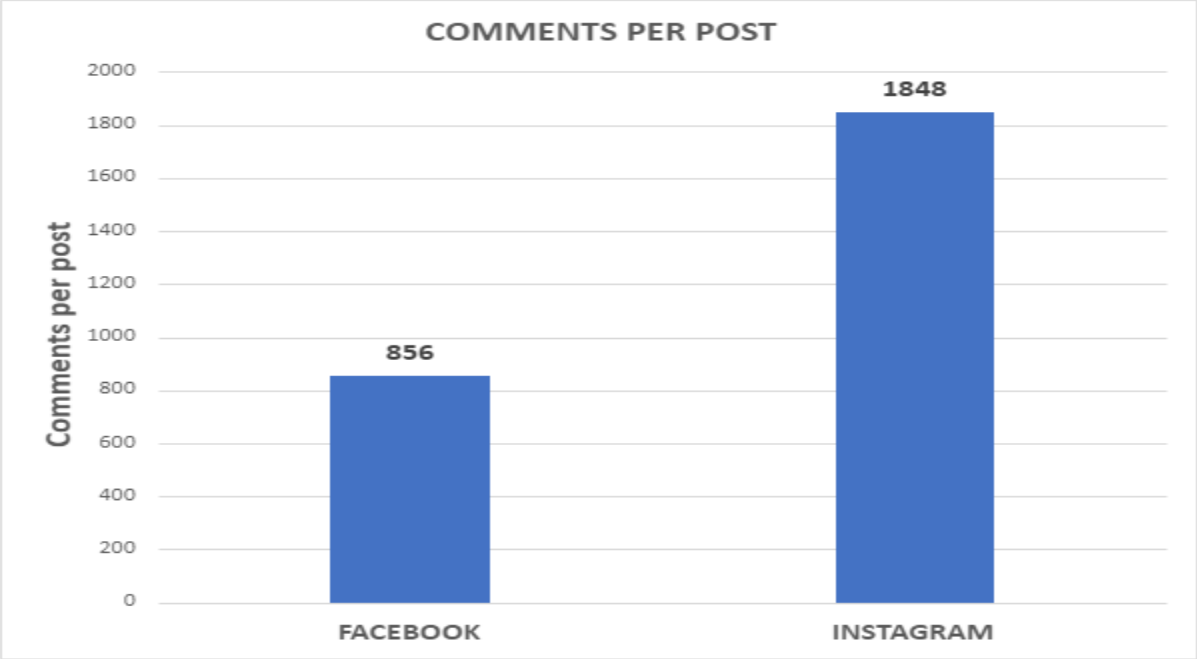
Στην *Εικόνα 89* βλέπουμε τον μέσο όρο likes που έχει μία δημοσίευση στα τρία ΚΔ. Άλλη μια ξεκάθαρη επικράτηση του Instagram με το δεύτερο Facebook να έχει λιγότερα από το 1/10 των likes και το τρίτο Twitter να έχει λιγότερα από το 1/100 του πρώτου Αυτή η ανάλυση κατέδειξε το Instagram ως το ΚΔ στο οποίο τα likes είναι κατά πολύ πιο χρησιμοποιούμενα από τα άλλα δύο.




***Εικόνα 89: Likes Per Post***

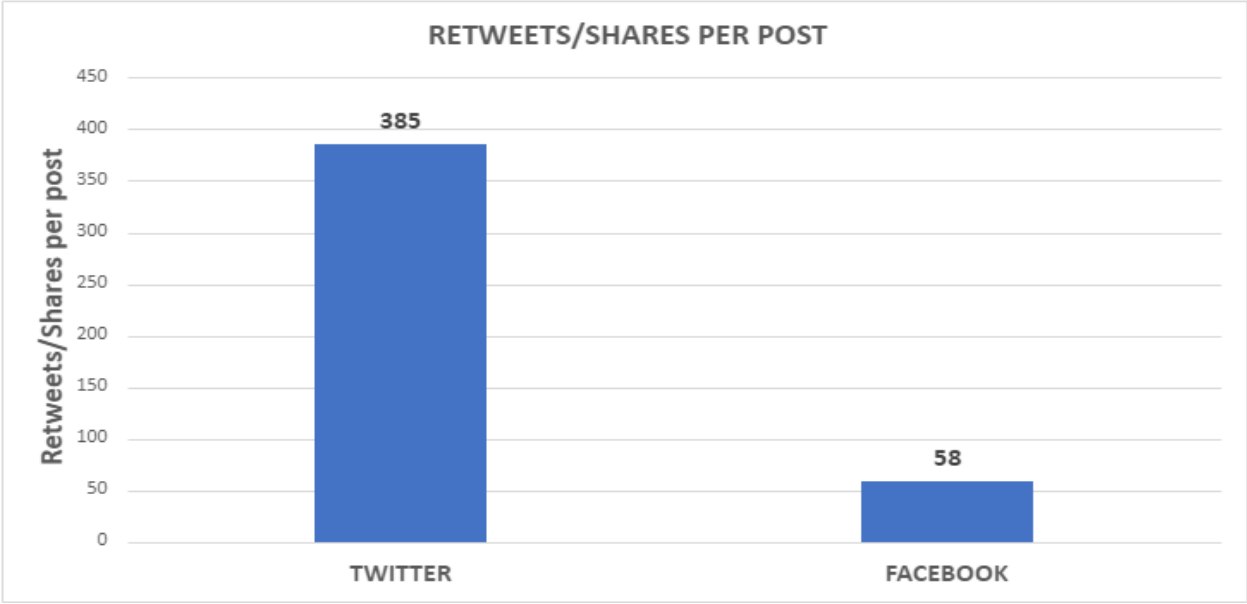
Στην *Εικόνα 90* βλέπουμε τα αποτελέσματα της ανάλυσης το οποίο σχετίζεται με τα σχόλια που μπορεί να περιέχει μια δημοσίευση κατά μέσο όρο. Σε αυτό το σημείο πρέπει να αναφερθεί πως οι δημοσιεύσεις στο Twitter περιείχαν και αυτές σχόλια, αλλά δεν κατέστη δυνατή η συλλογή τους ή το πλήθος τους και για αυτό το ΚΔ λείπει από την σύγκριση.

Εδώ παρατηρούμε μια ακόμη πρωτιά του Instagram, στο οποίο μία δημοσίευση κατά μέσο όρο έχει 1848 σχόλια, περίπου τα διπλάσια από ότι μια δημοσίευση στο Facebook. Γενικά οι συγκρίσεις μεταξύ των δύο αυτών ΚΔ που στη βάση μας έχουν δεδομένα παρόμοιου όγκου, δείχνουν έντονα την τάση των χρηστών να αντιδρούν σε δημοσιεύσεις του Instagram παρά σε δημοσιεύσεις του Facebook.




***Εικόνα 90: Comments Per Post***

Κλείνουμε την ενότητα με την ανάλυση που σχετίζεται με τα αναδημοσιεύσεις από άλλους χρήστες. Από αυτή την ανάλυση λείπει το Instagram, καθώς δεν υπάρχει η λειτουργία της αναδημοσίευσης σε αυτό το ΚΔ. Σε αυτή τη σύγκριση υπερτερεί το Twitter, έχοντας 385 αναδημοσιεύσεις ανά δημοσίευση, περίπου εφτά φορές περισσότερες σε σχέση με το Facebook (*Εικόνα 91*).




***Εικόνα 91: Retweets/Shares Per Post***

## 5.2. Αναλύσεις με βάση την κοινωνική επιρροή

Στη συνέχεια, θελήσαμε να χωρίσουμε τους χρήστες μας σε ομάδες, ανάλογα με την Κοινωνική Επιρροή (ΚΕ) που ασκούν και να εκτελέσουμε ξανά κάποιες από τις αναλύσεις. Χωρίστηκαν σε χρήστες με Μέτρια, Υψηλή και Πολύ Υψηλή ΚΕ.

### 5.2.1. Κοινωνική Επιρροή

Η έννοια της ΚΕ με τον οποίο χωρίσαμε τους χρήστες μας σε τρεις διαφορετικές ομάδες, περιγράφεται στην εργασία [8], και ο δείκτης μέτρησης επιρροής ενός χρήστη του Twitter ορίζεται ως εξής:

$$\text{Influence Metric} = \frac{\text{tweets}_k + \text{AdjustedTweets}_k}{\text{Hours}_{\text{since } k_{th}\text{tweet}}} * \text{OOM}(\text{Followers}) * \log_{10}\left(\frac{\text{Followers}}{\text{Following}} + 1\right), \text{ where OOM: Order Of Magnitude}$$

Για τον υπολογισμό του δείκτη επιρροής, λαμβάνονται υπόψη τα τελευταία 100 tweets του χρήστη, στα οποία προστίθενται όλα τα tweets που προήλθαν από αναδημοσιεύσεις τους. Ο αριθμός των tweets διαιρείται με τον αριθμό των ωρών που έχουν περάσει από το 100ο tweet. Ένας άλλος παράγοντας είναι ο αριθμός των ακολούθων καθώς και αυτός των χρηστών που ο ίδιος ο χρήστης ακολουθεί. Οι δύο τιμές διαιρούνται μεταξύ τους και στο πηλίκο τους προστίθεται μία μονάδα, ώστε να αποφευχθεί η πιθανότητα οι αριθμοί να είναι ίδιοι, καθώς θα υπάρχει θέμα όταν το νούμερο αυτό περαστεί σε έναν δεκαδικό λογάριθμο. Το αρχικό πηλίκο πολλαπλασιάζεται με τον λογάριθμο και με έναν δείκτη βαρύτητας των ακολούθων, και έτσι προκύπτει ο τελικός δείκτης μέτρησης επιρροής.

Παρακάτω εμφανίζονται οι χρήστες οι οποίοι με βάση τον δείκτη μέτρησης επιρροής κατατάσσονται στην ομάδα Μέτριας ΚΕ, της οποίας τα όρια είναι [30,47) (*Πίνακας 1*).

| **Alias** | **InfluenceMetric** |
|---|---|
| TurtleRock | 31.26 |
| Liverpool | 41.34 |
| Airbnb | 42.01 |
| Sport24 | 42.62 |
| DELL | 43.06 |
| OREO | 43.3 |
| Coursera | 43.56 |
| FIBA | 44.19 |
| MEGA | 44.34 |
| Orange | 44.39 |
| LEVIS | 44.48 |
| Oracle | 44.64 |
| Ellinofreneia | 45.07 |
| Arduino | 46.42 |
| Wikipedia | 46.52 |

***Πίνακας 1: Ομάδα Μέτριας ΚΕ***

Παρακάτω υπάρχουν οι χρήστες οι οποίοι με βάση τον δείκτη μέτρησης επιρροής κατατάσσονται στην ομάδα Υψηλής ΚΕ, της οποίας τα όρια είναι [48, 65) (*Πίνακας 2*).

| **Alias** | **InfluenceMetric** |
|---|---|
| PCMAG | 51.61 |
| CocaCola | 52.33 |
| FORD | 52.45 |
| JamieOliver | 53.09 |
| Nike | 53.25 |
| Redbull | 53.62 |
| Kathimerini | 53.75 |
| SKAI | 54.34 |
| Timoreiily | 54.36 |
| LKing | 55.07 |
| NEWS247 | 55.45 |
| Yahoo | 55.93 |
| Dropbox | 57.18 |
| PIglesias | 57.45 |
| MLS | 57.72 |
| Bulls | 58.09 |
| Naftemporiki | 59.04 |
| BMW | 59.07 |
| Microsoft | 59.38 |
| Matteorenzi | 59.45 |
| EUCOMM | 61.1 |
| Marvel | 61.29 |
| OnePlus | 61.6 |
| AmericanAir | 62.14 |
| NRJ | 62.52 |
| FTimes | 63.27 |
| Huffpost | 63.3 |
| McDonalds | 63.74 |
| BBCSport | 64.16 |

***Πίνακας 2: Ομάδα Υψηλής ΚΕ***

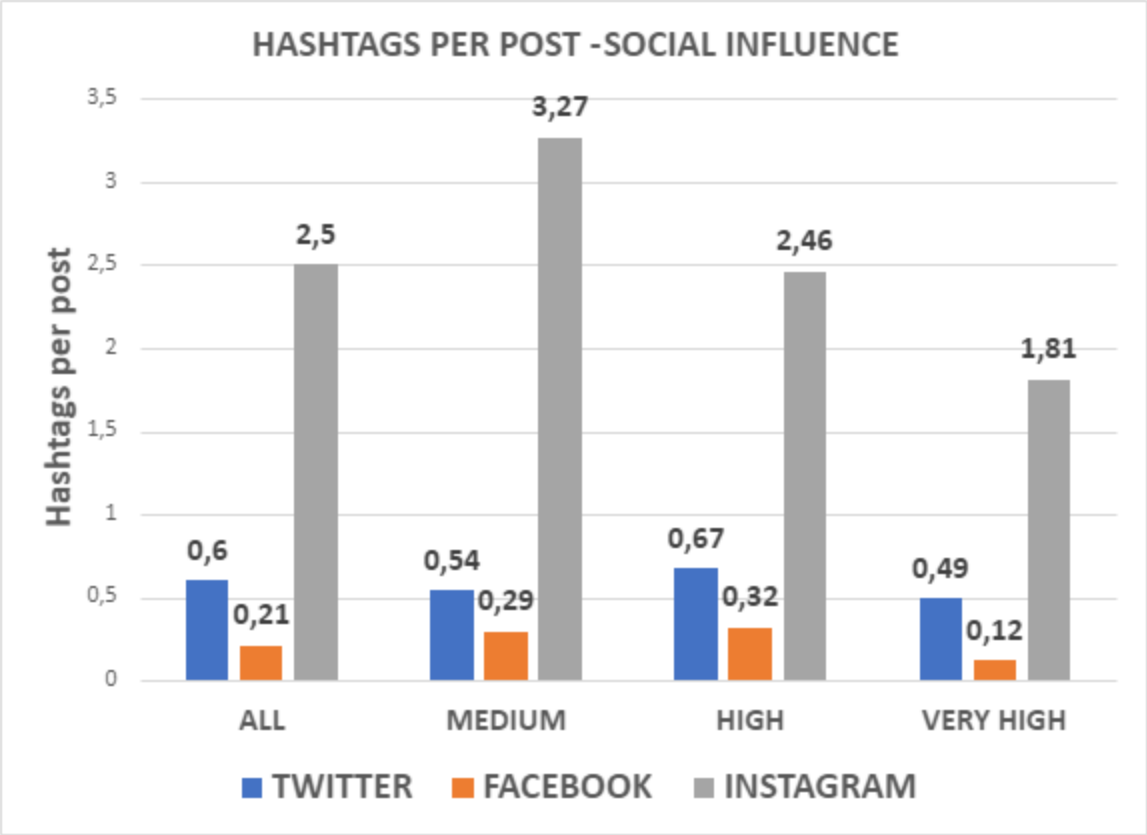
Παρακάτω υπάρχουν οι χρήστες οι οποίοι με βάση τον δείκτη μέτρησης επιρροής κατατάσσονται στην ομάδα Υψηλής ΚΕ, της οποίας τα όρια είναι [66, 82) (*Πίνακας 3*).

| **Alias** | **InfluenceMetric** |
|---|---|
| Starbucks | 66.3 |
| VSecret | 66.98 |
| Samsung | 71.03 |
| CR7 | 71.95 |
| Chelsea | 72.06 |
| Google | 72.2 |
| RedDevils | 72.31 |
| Barca | 72.34 |
| 9GAG | 72.56 |
| WhiteHouse | 73.01 |
| Time | 75.59 |
| Cnn | 78.78 |
| KatyPerry | 81.78 |

***Πίνακας 3: Ομάδα Πολύ Υψηλής ΚΕ***

## 5.2.2. Ανάλυση δεδομένων

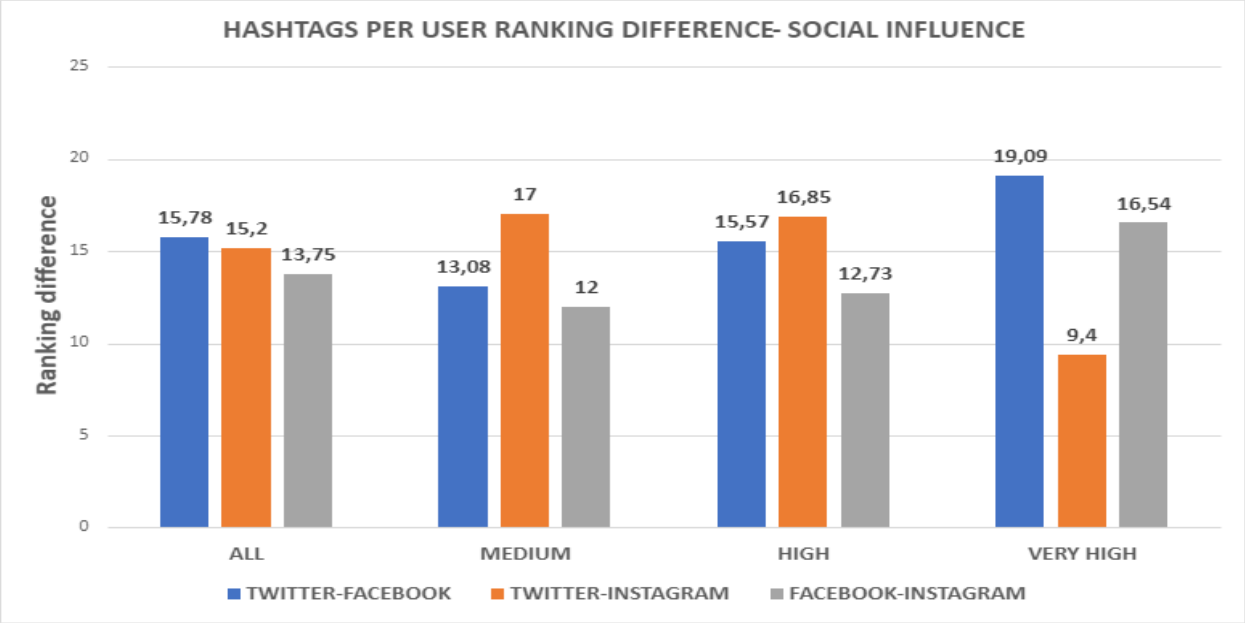
Η πρώτη ανάλυση σχετίζεται με τον μέσο όρο hashtags ανά δημοσίευση για κάθε ΚΔ. Τρέξαμε την ανάλυση τέσσερις φορές, μία για όλους τους χρήστες και τις υπόλοιπες τρεις με κάθε ομάδα ξεχωριστά με τα αποτελέσματα να εμφανίζονται στην *Εικόνα 92*. Στον κάθετο άξονα παρουσιάζονται οι μέσοι όροι hashtags ανά δημοσίευση, ενώ στον οριζόντιο άξονα παρουσιάζονται οι τέσσερις διαφορετικές κατηγορίες. Με μπλε στήλες παρουσιάζεται το Twitter, με πορτοκαλί το Facebook και με γκρι το Instagram. Παρατηρούμε πως οι χρήστες που ασκούν Μέτρια ΚΕ, χρησιμοποιούν 3,27 hashtags ανά δημοσίευση στο Instagram, 32% πιο πάνω από τον αντίστοιχο μέσο όρο της Υψηλής ΚΕ και 80% πάνω από την ομάδα της Υψηλής ΚΕ. Ακόμη, οι χρήστες με Υψηλή Επιρροή χρησιμοποιούν 0,67 hashtags ανά δημοσίευση στο Twitter, 24% περισσότερα από την Μέτρια ΚΕ, και 36% από την τρίτη ομάδα της Πολύ Υψηλής ΚΕ. Οι χρήστες με Πολύ Υψηλή ΚΕ τείνουν να κάνουν μέτρια χρήση των hashtags σε όλα τα ΚΔ και δεν υπερτερούν σε κάποια κατηγορία.




***Εικόνα 92: Hashtags Per Post - Social Influence***

Για την επόμενη ανάλυση, κατατάξαμε τους χρήστες μας σε τρεις λίστες, μία για κάθε ΚΔ, με βάση το πλήθος των hashtags που έχουν προσθέσει στις δημοσιεύσεις τους. Έπειτα, για κάθε χρήστη, υπολογίσαμε την απόλυτη διαφορά των θέσεων στις οποίες βρίσκεται. Τέλος, υπολογίσαμε τους τρεις μέσους όρους αυτών των διαφορών οι οποίοι μπορούν να βρεθούν στην *Εικόνα 93* στην ενότητα «ALL». Στη συνέχεια τρέξαμε την ίδια ανάλυση άλλες τρεις φορές, μία για κάθε ομάδα και τις προσθέσαμε στο γράφημα.

Παρατηρούμε πως οι χρήστες με Μέτρια και Υψηλή ΚΕ χρησιμοποιούν περισσότερο και σε παρόμοιο βαθμό Facebook και Instagram παρά τους άλλους δύο συνδυασμούς. Οι χρήστες με πολύ υψηλή επιρροή προτιμούν περισσότερο την χρήση του Twitter και του Instagram παρά το Facebook.

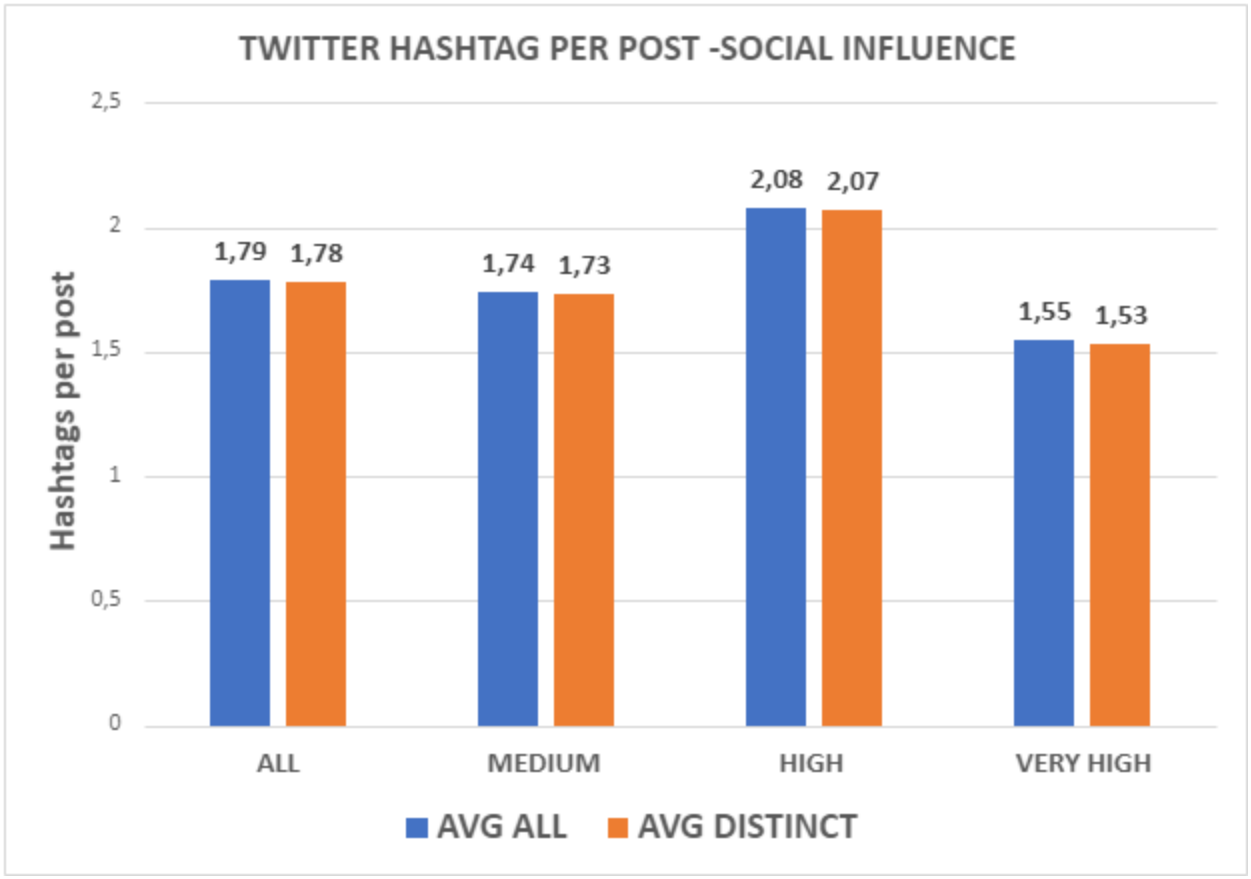



***Εικόνα 93: Hashtags Per User - Rankings - Social Influence***

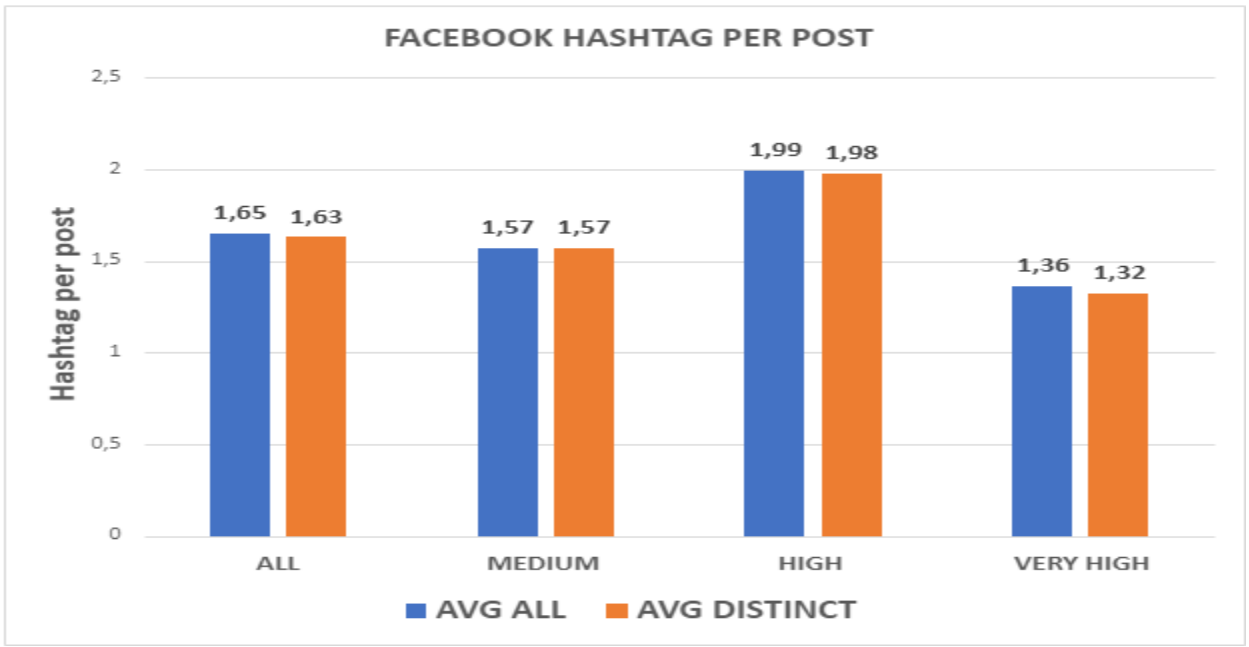
Στη συνέχεια, πήραμε ένα μεγάλο ποσοστό των κορυφαίων δημοσιεύσεων κάθε ΚΔ και σε αυτές τρέξαμε τις αναλύσεις παρακάτω (*Εικόνα 94, Εικόνα 95 και Εικόνα 96*). Συγκεκριμένα υπολογίσαμε τον μέσο όρο hashtag ανά δημοσίευση και στη συνέχεια υπολογίσαμε τον μέσο όρο hashtag ανά δημοσίευση, αλλά σε αυτή την περίπτωση κρατήσαμε μόνο μία εμφάνιση κάθε hashtag σε κάθε δημοσίευση. Με αυτή την ανάλυση στην ουσία ελέγχουμε αν υπάρχουν δημοσιεύσεις στις οποίες κάποιο hashtag χρησιμοποιείται πολλές φορές και από ποια ομάδα χρηστών προέρχεται.

Στον κάθετο άξονα παρουσιάζονται οι μέσοι όροι hashtag ανά δημοσίευση, ενώ στον οριζόντιο άξονα παρουσιάζονται οι τέσσερις κατηγορίες. Με μπλε στήλες παρουσιάζονται οι μέσοι όροι των hashtag συμπεριλαμβανομένων και των επαναλήψεων των εγγραφών. Με πορτοκαλί στήλες παρουσιάζονται οι μέσοι όροι των hashtag λαμβάνοντας υπόψη μόνο μία εμφάνιση κάθε hashtag ανά δημοσίευση.

Παρατηρούμε πως στο Twitter (*Εικόνα 94*) δεν παρατηρούνται μεγάλες διαφορές σε καμία ομάδα. Παρόμοια κατάσταση επικρατεί και στην περίπτωση του Facebook, χωρίς μεγάλες διαφορές (*Εικόνα 95*).

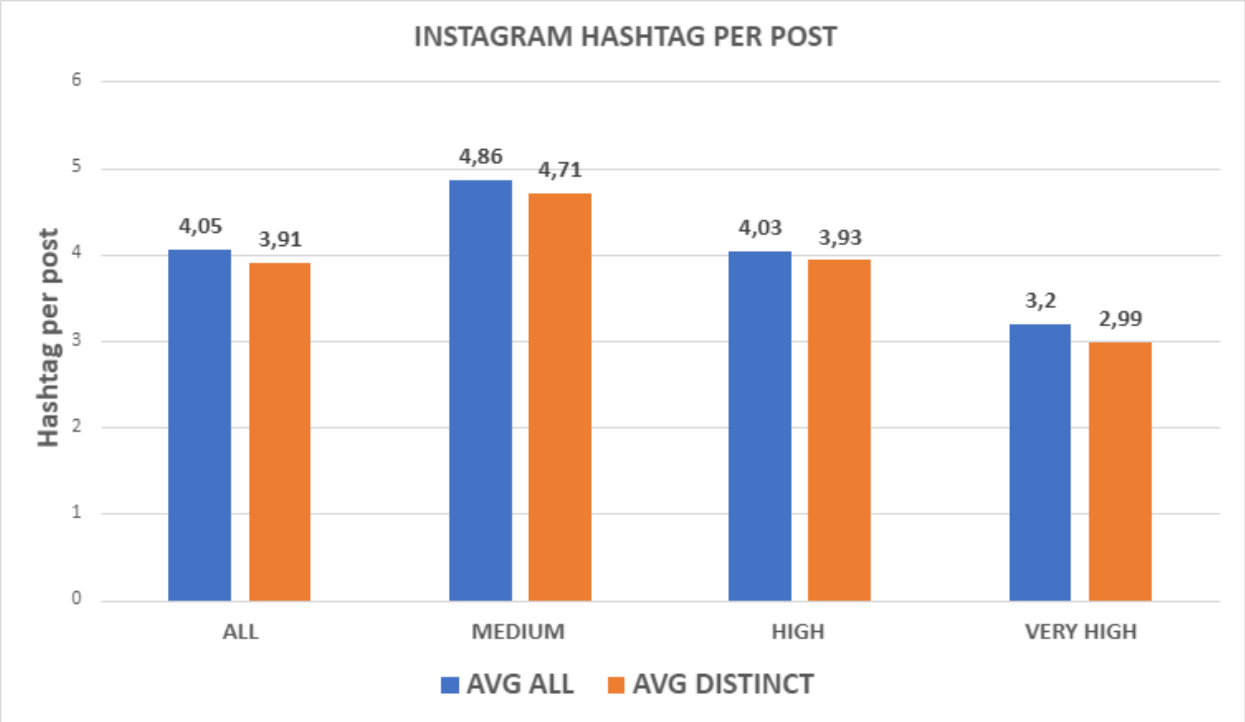



***Εικόνα 94: Hashtag Per Post - Social Influence - Twitter***




**Εικόνα 95: Hashtag Per Post - Social Influence - Facebook**

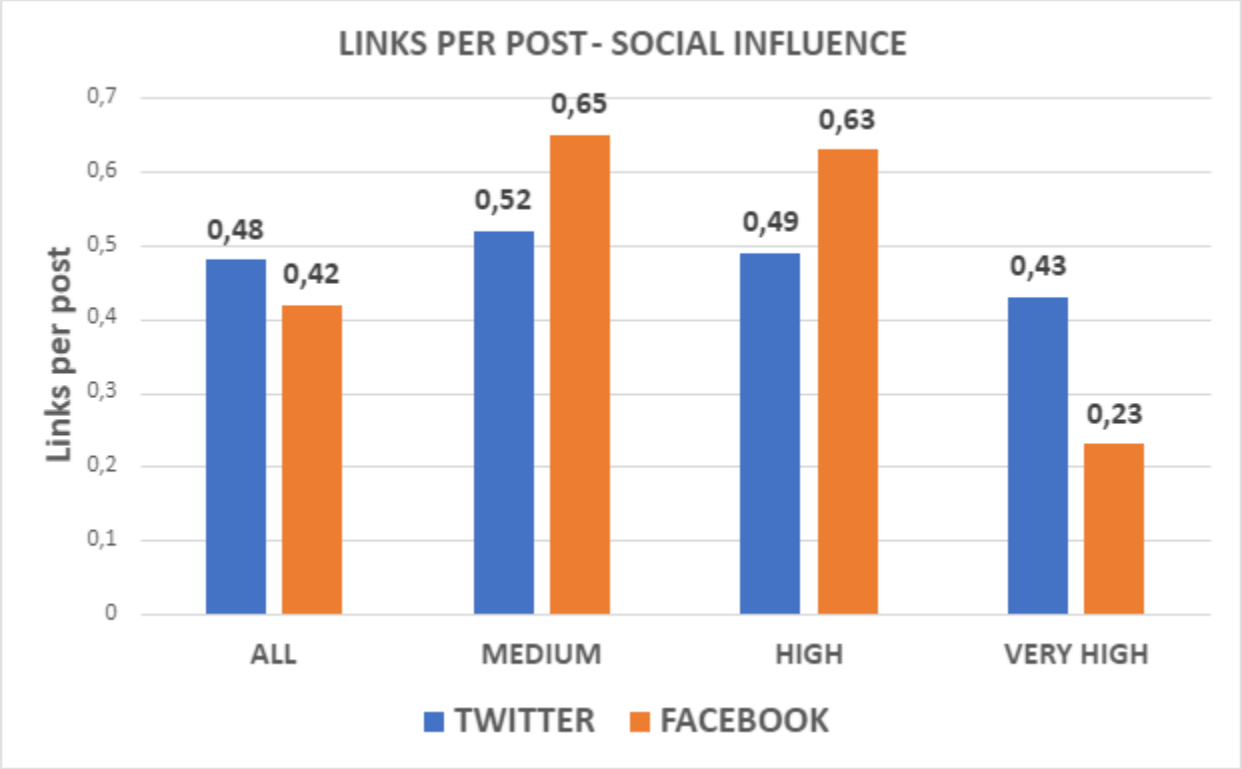
Στην περίπτωση του Instagram όμως, παρατηρούνται σημαντικές διαφορές σε όλες τις περιπτώσεις. Συγκεκριμένα η διαφορά των δύο μέσων όρων της Μέτριας ΚΕ είναι της τάξεως του 3,18% και η διαφορά της Υψηλής ΚΕ είναι της τάξεως του 2,54%. Η μεγαλύτερη διαφορά της τάξεως του 7% εμφανίζεται στην ομάδα χρηστών με Πολύ Υψηλή ΚΕ (*Εικόνα 96*). Με αυτή την ανάλυση, το Instagram ξεχωρίζει ως το ΚΔ στο οποίο οι χρήστες έχουν την τάση να χρησιμοποιούν ένα hashtag παραπάνω από μία φορές στις δημοσιεύσεις τους.




***Εικόνα 96: Hashtag Per Post - Social Influence - Instagram***

Η επόμενη οντότητα που αναλύεται είναι οι υπερσύνδεσμοι. Σε αυτή την ανάλυση υπολογίσαμε τους μέσους όρους υπερσυνδέσμων ανά δημοσίευση στο Twitter και το Facebook με το Instagram να απουσιάζει καθώς δεν υπάρχουν υπερσύνδεσμοι στις δημοσιεύσεις αυτού του ΚΔ. Στη συνέχεια τρέξαμε την ίδια ανάλυση και για τις τρεις ξεχωριστές ομάδες με τα αποτελέσματα των αναλύσεων να εμφανίζονται στην *Εικόνα 97*.

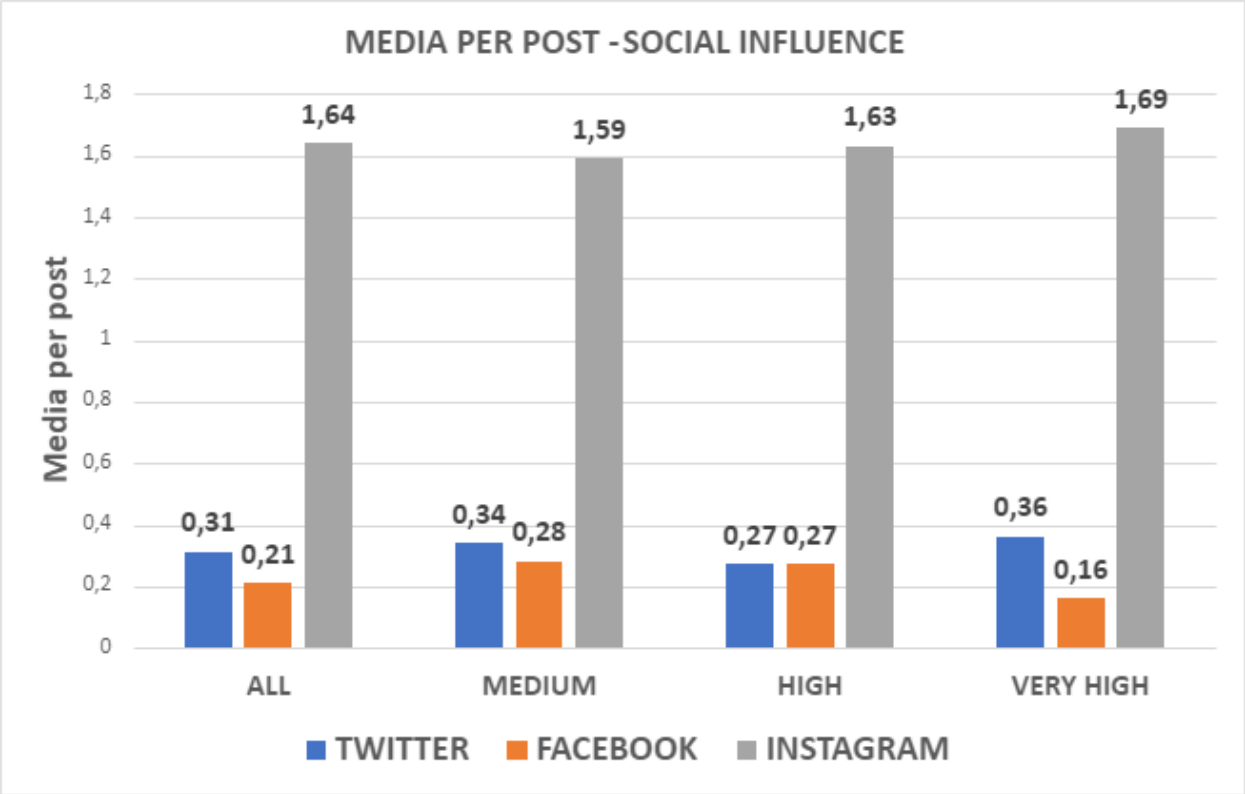
Παρατηρείται πως είναι πιο πιθανό να περιέχεται υπερσύνδεσμος σε μία δημοσίευση του Facebook παρά σε μία του Twitter, σε όλες τις περιπτώσεις εκτός από την ομάδα χρηστών με Πολύ Υψηλή ΚΕ. Συγκεκριμένα, στην Μέτρια ΚΕ παρατηρείται μια διαφορά υπέρ του Facebook κατά 25%, ενώ στην Υψηλή ΚΕ παρατηρείται διαφορά κατά 28%. Αντιθέτως στην Πολύ Υψηλή ΚΕ επικρατεί το Twitter με διαφορά της τάξεως του 86%. Γενικά, οι χρήστες με Μέτρια ΚΕ χρησιμοποιούν περισσότερους υπερσυνδέσμους στις δημοσιεύσεις τους, με μικρή διαφορά από την δεύτερη ομάδα των χρηστών με Υψηλή ΚΕ.




***Εικόνα 97: Links Per Post - Social Influence***

Στη συνέχεια ασχοληθήκαμε με τα πολυμέσα. Όπως φαίνεται και στην *Εικόνα 98*, υπολογίσαμε τους μέσους όρους από πολυμέσα ανά δημοσίευση σε κάθε ΚΔ, για κάθε ομάδα ΚΕ.

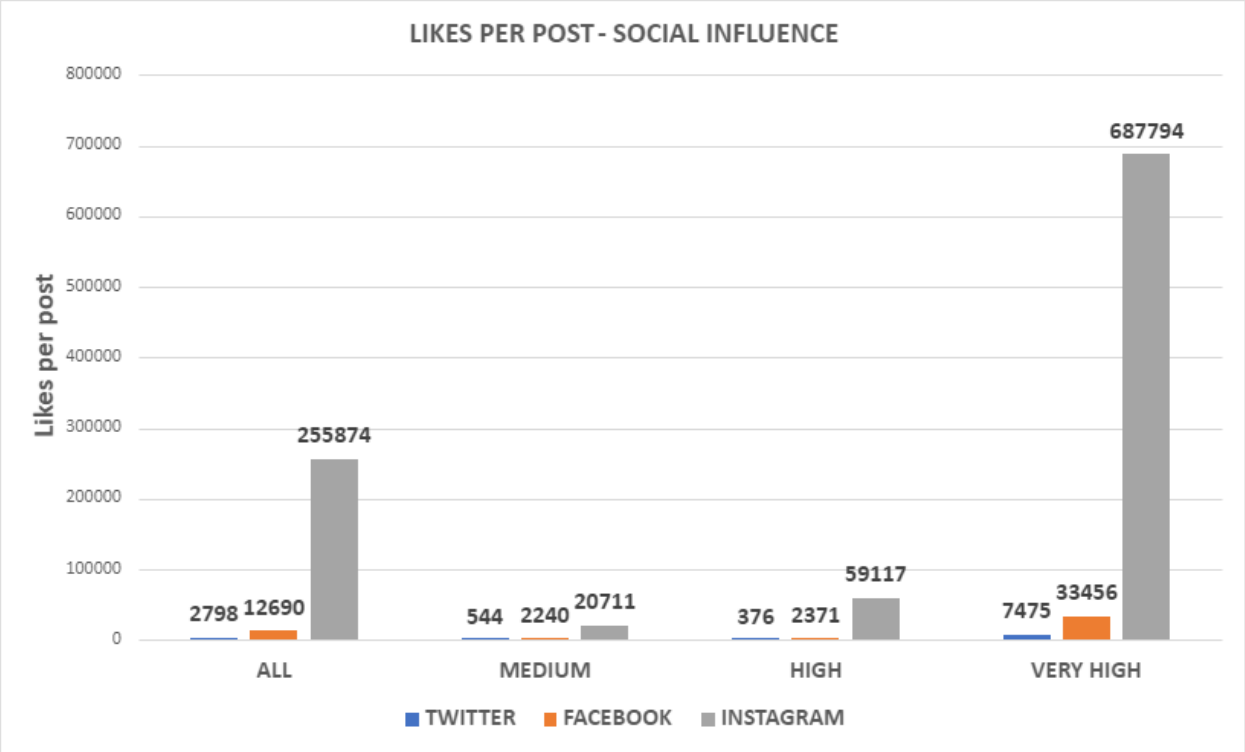
Μια πρώτη παρατήρηση είναι πως το Instagram επικρατεί με διαφορά σε κάθε μία από τις περιπτώσεις, με τους χρήστες Πολύ Υψηλής ΚΕ να περιέχουν περισσότερες εικόνες στις δημοσιεύσεις τους. Αναλυτικότερα, στην Μέτρια ΚΕ το Instagram επικρατεί με 1,59 πολυμέσα ανά δημοσίευση, 367% περισσότερα από το δεύτερο Twitter. Στην υψηλή ΚΕ το Instagram επικρατεί με 1,63 πολυμέσα ανά δημοσίευση, 503% περισσότερα από τα άλλα δύο ΚΔ. Τέλος στην Πολύ Υψηλή ΚΕ το Instagram επικρατεί με 1,69 πολυμέσα ανά δημοσίευση, 369% περισσότερα από το δεύτερο Twitter. Ακόμη βλέπουμε πως στην περίπτωση των χρηστών με Πολύ Υψηλή ΚΕ εμφανίζεται μια τάση αποφυγής του Facebook για την δημοσίευση των πολυμέσων τους, καθώς ο μέσος όρος αυτός είναι αρκετά χαμηλότερος σε σχέση με τους αντίστοιχους άλλους δύο. Συγκεκριμένα ο μέσος όρος της Πολύ Υψηλής ΚΕ για το Facebook είναι κατά 68% μικρότερος συγκριτικά με τους άλλους δύο.




***Εικόνα 98: Media Per Post - Social Influence***

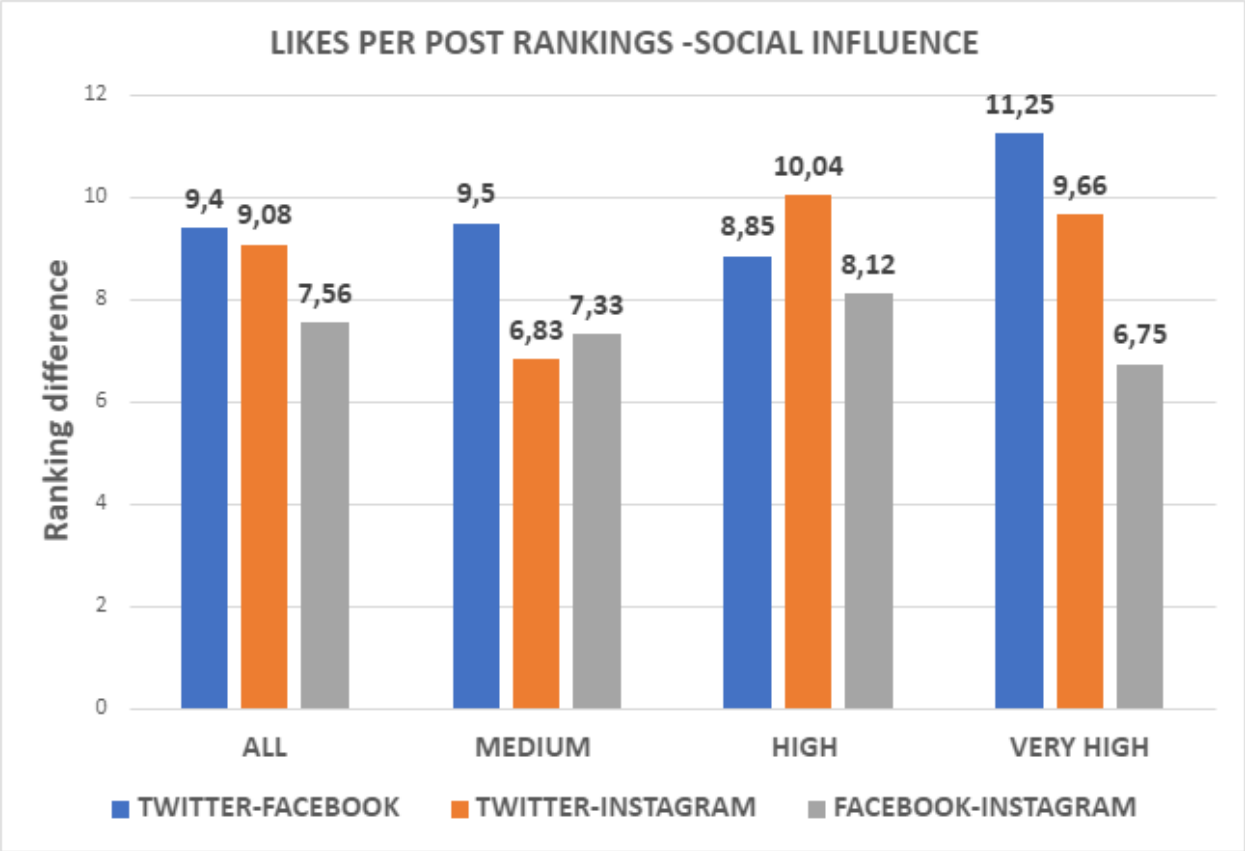
Η επόμενη ανάλυσή μας σχετίζεται με τα likes. Υπολογίσαμε τους μέσους όρους likes ανά δημοσίευση για κάθε ΚΔ και κάθε ξεχωριστή ομάδα. Τα αποτελέσματα εμφανίζονται στην *Εικόνα 99*. Στον κάθετο άξονα παρουσιάζονται οι μέσοι όροι likes ανά δημοσίευση, ενώ στον οριζόντιο άξονα παρουσιάζονται οι 4 κατηγορίες. Με μπλε στήλες εμφανίζονται τα στατιστικά του Twitter, με πορτοκαλί του Facebook και με γκρι στήλες του Instagram.

Παρατηρούμε πως το Instagram επικρατεί σε όλες τις περιπτώσεις κατά πολύ, κάνοντας το ΚΔ στο οποίο τα likes είναι πιο διαδεδομένα. Γενικά η ομάδα με την Πολύ Υψηλή ΚΕ μέσω αυτής της ανάλυσης δικαιολογεί την ιδιότητά της, καθώς οι μέσοι όροι της είναι πολλαπλάσια μεγαλύτεροι συγκριτικά με αυτούς των άλλων δύο ομάδων. Αναλυτικότερα, στη Μέτρια ΚΕ επικρατεί το Instagram με 20711 likes ανά δημοσίευση, σχεδόν 9 φορές περισσότερα από το δεύτερο Facebook. Στην Υψηλή ΚΕ επικρατεί πάλι το Instagram με 59117 likes ανά δημοσίευση, σχεδόν 25 φορές περισσότερα από το δεύτερο Facebook. Τέλος στην Πολύ Υψηλή ΚΕ επικρατεί και πάλι το Instagram με 687794 likes ανά δημοσίευση, 20 φορές περισσότερα από το δεύτερο Facebook.




***Εικόνα 99: Likes Per Post - Social Influence***

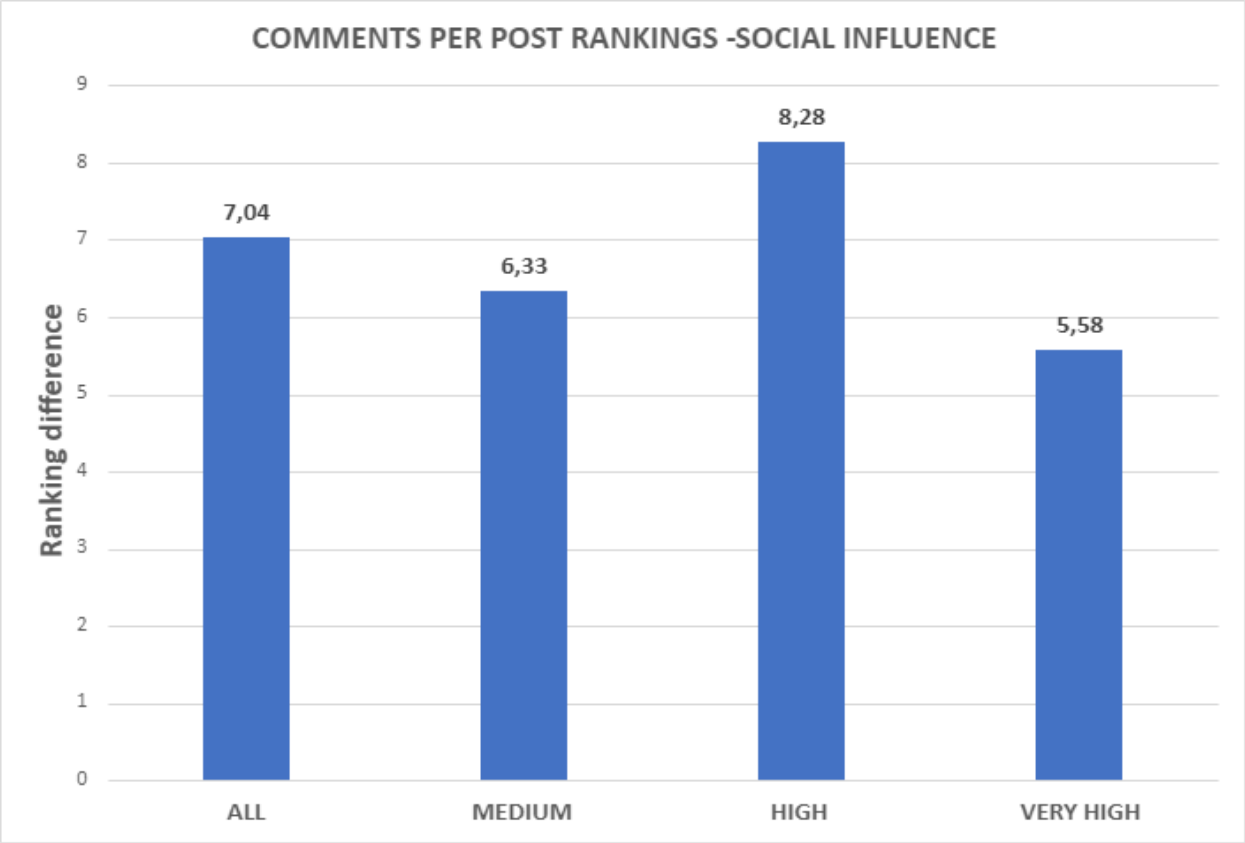
Συνεχίζοντας τις αναλύσεις γύρω από τα likes, κατατάξαμε τους χρήστες μας με βάση τον μέσο όρο likes ανά δημοσίευση που έχουν σε κάθε ΚΔ, και στη συνέχεια τρέξαμε ξανά την ανάλυση για κάθε ομάδα ξεχωριστά. Όπως βλέπουμε στην *Εικόνα 100*, η ομάδα χρηστών που ασκεί Μέτρια ΚΕ παρουσιάζει μεγαλύτερη συνέπεια στα ΚΔ Twitter και Instagram, με τις άλλες δύο ομάδες να εμφανίζουν την συνέπεια αυτή στα ΚΔ Facebook και Instagram. Συγκεκριμένα η Μέτρια ΚΕ εμφανίζει την μικρότερη διαφορά θέσεων μεταξύ Twitter-Instagram με 6,83 θέσεις έναντι 7,33 θέσεων στο Facebook-Instagram και 9,5 θέσεων μεταξύ Twitter-Facebook. Στην Υψηλή ΚΕ επικρατεί η διαφορά Facebook-Instagram με 8,12 θέσεις έναντι 8,85 θέσεων στο Twitter-Facebook και 10,04 θέσεων στο Twitter-Instagram. Τέλος στην Πολύ Υψηλή ΚΕ επικρατεί με 6,75 θέσεις το Facebook-Instagram, με δεύτερη να έρχεται η διαφορά Twitter-Instagram με 9,66 θέσεις και τρίτη η διαφορά Twitter-Facebook με 11,25 θέσεις.




***Εικόνα 100: Likes Per Post - Rankings - Social Influence***

Στην επόμενη ανάλυση ασχοληθήκαμε με τα σχόλια των χρηστών. Κατατάξαμε τους χρήστες με βάση τον μέσο όρο σχολίων που δέχονται σε μία δημοσίευσή τους στο Facebook και το Instagram. Το Twitter απουσιάζει καθώς δεν κατέστη δυνατή η συλλογή των δεδομένων αυτών. Στη συνέχεια υπολογίσαμε την απόλυτη διαφορά θέσεων κάθε χρήστη και τέλος τον μέσο όρο διαφορών σε κάθε ΚΔ. Αμέσως μετά πραγματοποιήσαμε την ίδια ανάλυση για τις τρεις ομάδες ξεχωριστά. Τα αποτελέσματα εμφανίζονται στην *Εικόνα 101*.

Παρατηρούμε πως οι χρήστες με Πολύ Υψηλή ΚΕ δεν έχουν μεγάλες διαφορές στην κατάταξή τους σε σχέση με τις άλλες δύο ομάδες, που σημαίνει πως σε Facebook και Instagram είναι πιο σταθερός ο αριθμός των σχολίων που δέχονται σε σχέση με τους υπόλοιπους. Στις άλλες δύο περιπτώσεις και ειδικότερα στην ομάδα με Υψηλή ΚΕ, τα σχόλια που δέχονται δεν είναι αναλογικά του ιδίου βαθμού σε όλα τα ΚΔ σε σχέση με τους υπόλοιπους χρήστες και για αυτό η μεγαλύτερη διαφορά θέσεων. Αναλυτικότερα, η Μέτρια ΚΕ εμφανίζει 6,33 θέσεις στην διαφορά της κατάταξης έναντι 8,28 θέσεων της Υψηλής ΚΕ, με την Πολύ Υψηλή ΚΕ να επικρατεί με μόλις 5,58 θέσεις.

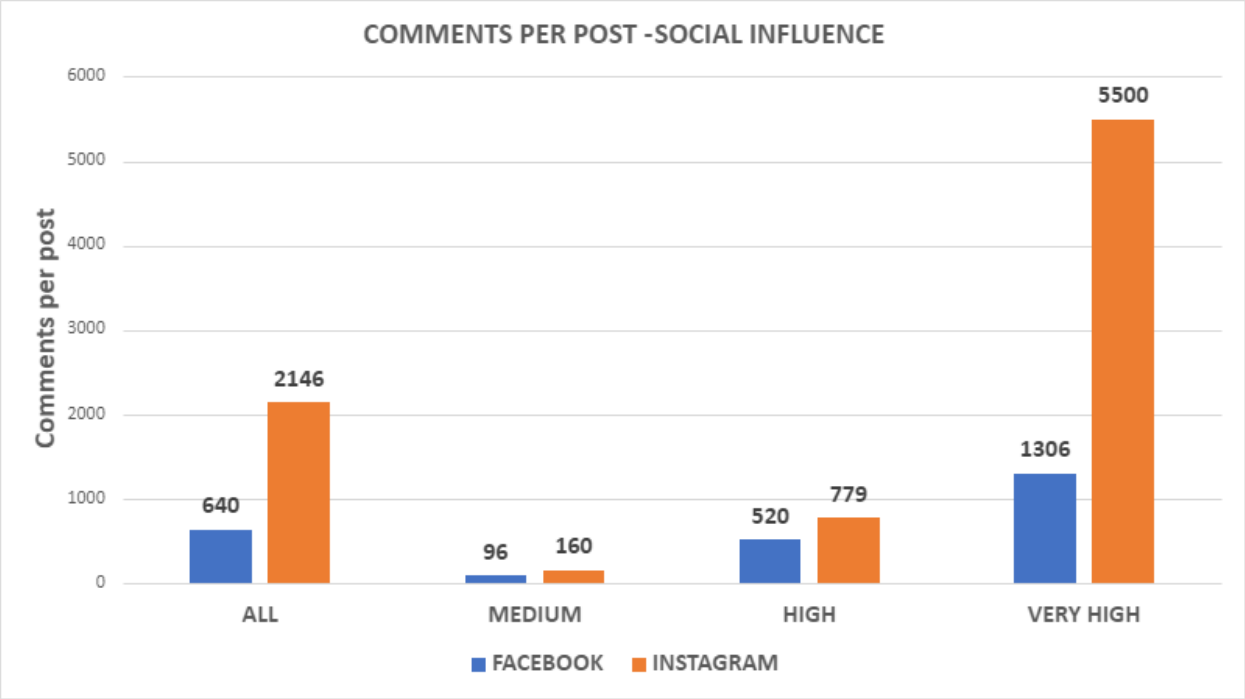



***Εικόνα 101: Comments Per Post - Rankings - Social Influence***

Έπειτα πραγματοποιήσαμε την ανάλυση που εμφανίζεται στην *Εικόνα 102*. Υπολογίσαμε τους μέσους όρους σχολίων ανά δημοσίευση για τα δύο ΚΔ και για τις τρεις διαφορετικές ομάδες. Στον κάθετο άξονα παρουσιάζονται οι μέσοι όροι σχολίων και στον οριζόντιο άξονα οι τέσσερις κατηγορίες. Με μπλε στήλες εμφανίζεται το Facebook και με πορτοκαλί το Instagram.

Οι χρήστες που ασκούν Πολύ Υψηλή ΚΕ επικρατούν και στα δύο ΚΔ συγκριτικά με τις άλλες δύο ομάδες. Γενικά τα νούμερα αυξάνονται αναλογικά με την ΚΕ που ασκεί ένας χρήστης, με το Instagram να υπερισχύει σε κάθε περίπτωση.

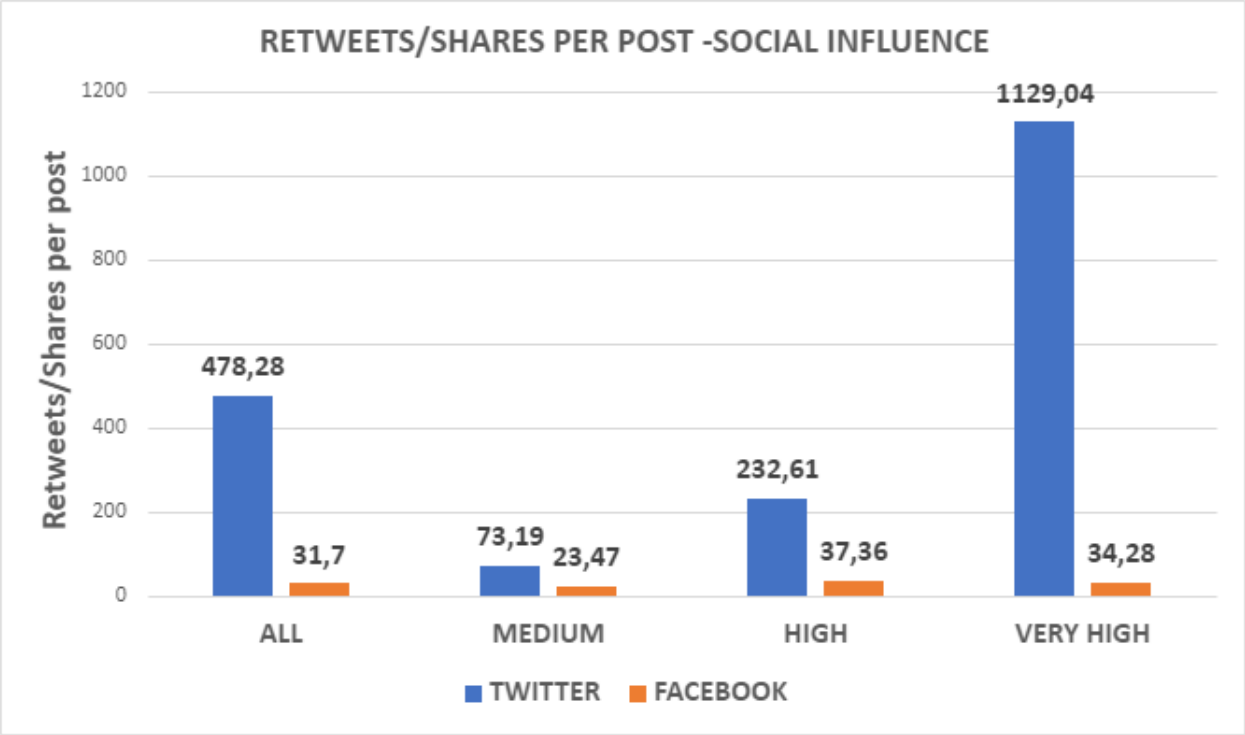
Συγκεκριμένα για το Facebook επικρατεί η Πολύ Υψηλή ΚΕ με 1306 σχόλια, δυόμιση φορές περισσότερα από την Υψηλή ΚΕ και δεκατρείς φορές περισσότερα από την Μέτρια ΚΕ. Για το Instagram επικρατεί πάλι η Πολύ Υψηλή ΚΕ με 5500 σχόλια ανά δημοσίευση, έναντι 779 της Υψηλής ΚΕ και 160 σχολίων της Μέτριας ΚΕ.




***Εικόνα 102: Comments Per Post - Social Influence***

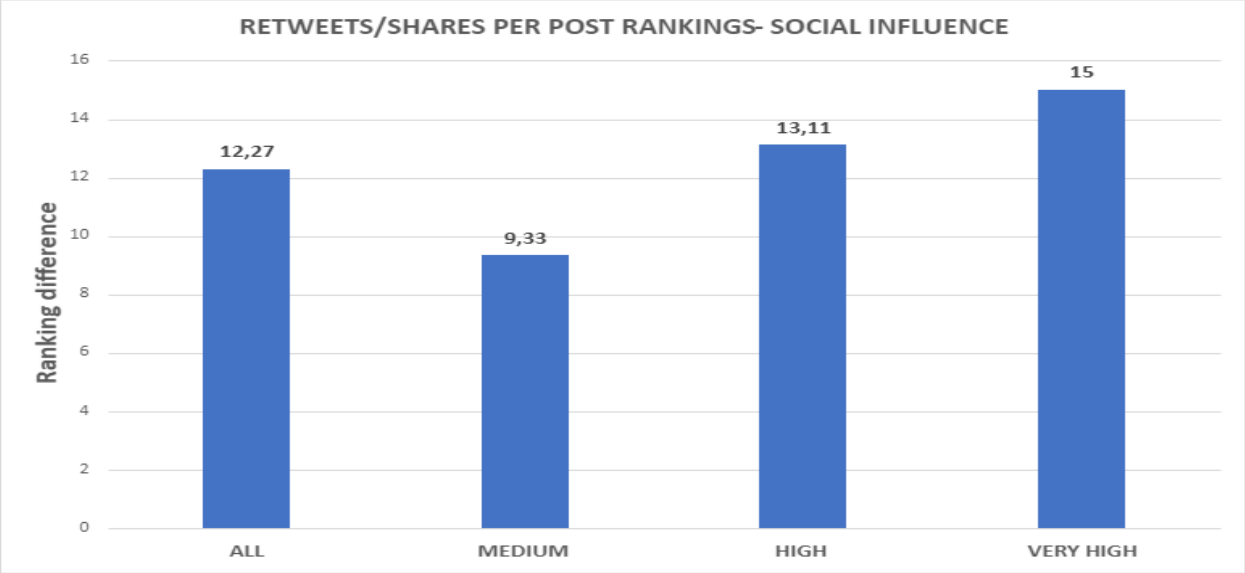
Η επόμενη ανάλυση πραγματεύεται τις αναδημοσιεύσεις που έχει κατά μέσο όρο μια δημοσίευση στο Twitter και το Facebook. Στο Instagram δεν υλοποιούνται οι αναδημοσιεύσεις και για αυτό απουσιάζει το ΚΔ από την ανάλυση.

Παρατηρούμε στην *Εικόνα 103* πως για το Twitter, η ομάδα χρηστών με Πολύ Υψηλή ΚΕ επικρατεί με μεγάλη διαφορά, ενώ στην περίπτωση του Facebook επικρατεί για λίγο η ομάδα της Υψηλής ΚΕ. Αναλυτικότερα, στο Twitter επικρατεί η Πολύ Υψηλή ΚΕ με 1129 αναδημοσιεύσεις ανά δημοσίευση, έναντι 232 της Υψηλής ΚΕ και 73 της Μέτριας ΚΕ. Στο Facebook επικρατεί η Υψηλή ΚΕ με 37 αναδημοσιεύσεις ανά δημοσίευση, με την Πολύ Υψηλή ΚΕ να ακολουθεί με 34 αναδημοσιεύσεις και τελευταία την Μέτρια ΚΕ με 23 αναδημοσιεύσεις. Με αυτή την ανάλυση συμπεραίνουμε πως όσο μεγαλύτερη ΚΕ ασκεί ένας χρήστης στους ακόλουθούς του, τόσο περισσότερες αναδημοσιεύσεις θα δέχονται οι δημοσιεύσεις του.




***Εικόνα 103: Retweets/Shares Per Post - Social Influence***

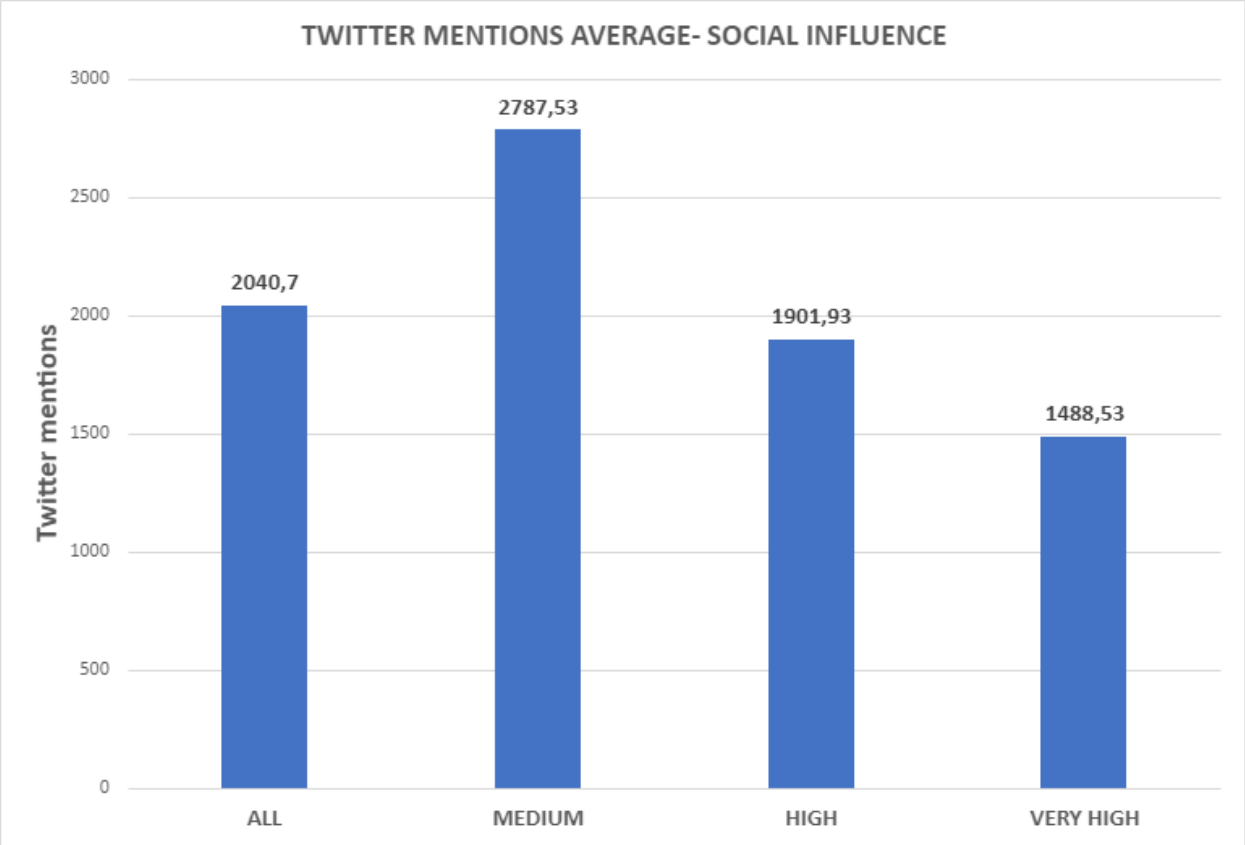
Συνεχίζοντας με τις αναδημοσιεύσεις, πραγματοποιήσαμε την παρακάτω ανάλυση (*Εικόνα 104)*. Κατατάξαμε τους χρήστες με βάση τον μέσο όρο αναδημοσιεύσεων ανά δημοσίευση για τα δύο ΚΔ και τις τρεις διαφορετικές ομάδες. Παρατηρώντας τα νούμερα συμπεραίνουμε πως οι χρήστες με Μέτρια ΚΕ εμφανίζουν μεγαλύτερη συνέπεια στον αριθμό αναδημοσιεύσεων που δέχεται μία δημοσίευση τους στα δύο ΚΔ, ενώ οι χρήστες με Πολύ Μεγάλη ΚΕ εμφανίζουν την μεγαλύτερη απόκλιση. Συγκεκριμένα, οι χρήστες με Μέτρια ΚΕ εμφανίζουν 9,33 θέσεις διαφορά, έναντι 13,11 θέσεων της Υψηλής ΚΕ και 15 θέσεων της Πολύ Υψηλής ΚΕ.




**Εικόνα 104: Retweets/Shares Per Post - Rankings - Social Influence**

Η επόμενη ανάλυση αφορά τις αναφορές (mentions) που βρίσκονται στις δημοσιεύσεις των χρηστών του Twitter. Υπολογίσαμε τον μέσο όρο από αναφορές ανά χρήστη για κάθε μία από τις τρεις ομάδες. Στον κάθετο άξονα παρουσιάζονται οι μέσοι όροι αναφορών ανά χρήστη, ενώ στον οριζόντιο άξονα παρουσιάζονται οι τέσσερις κατηγορίες.

Βλέποντας την *Εικόνα 105*, διαπιστώνουμε πως όσο μικρότερη είναι η ΚΕ των χρηστών, τόσο περισσότερα mentions περιέχονται στις δημοσιεύσεις τους. Αναλυτικότερα, πρώτη έρχεται η Μέτρια ΚΕ με 2787 αναφορές ανά χρήστη, με την Υψηλή ΚΕ να ακολουθεί με 1901 αναφορές και τελευταία την Πολύ Υψηλή ΚΕ με 1488 αναφορές ανά χρήστη.



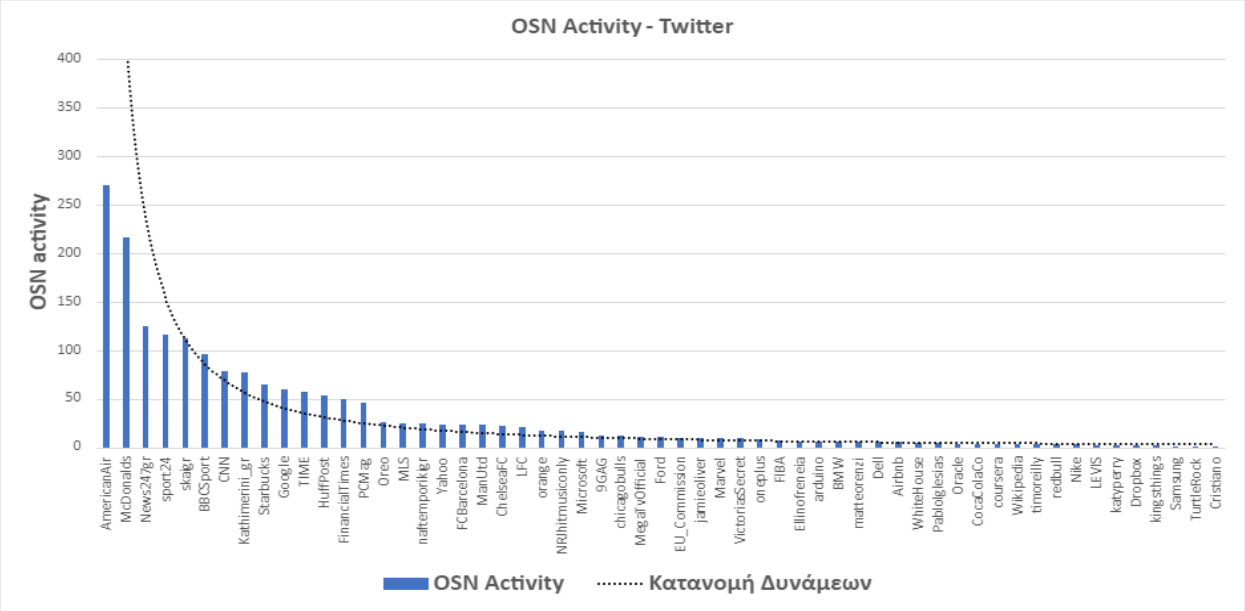

***Εικόνα 105: Twitter Mentions Average - Social Influence***

Κλείνοντας, οι τελευταίες τρεις αναλύσεις μας αφορούν το OSN activity των χρηστών, δηλαδή την δραστηριότητά τους στα ΚΔ. Συγκεκριμένα, για κάθε δημοσίευση των χρηστών στο Twitter και το Facebook (στο Instagram δεν κατέστη δυνατό) έχουμε αποθηκεύσει την ημερομηνία δημιουργίας της.

Για κάθε χρήστη, επιλέγουμε την δημοσίευση που έχει δημιουργηθεί παλιότερα καθώς και την πιο πρόσφατη ανάμεσα σε αυτές που είναι αποθηκευμένες στη ΒΔ. Υπολογίζουμε την διαφορά ημερών ανάμεσα στις δύο αυτές δημοσιεύσεις καθώς και τον αριθμό των δημοσιεύσεων του χρήστη σε αυτή την χρονική διάρκεια. Κάνοντας την διαίρεση των δημοσιεύσεων με τον αριθμό των ημερών προκύπτει ο μέσος όρος δημοσιεύσεων ανά ημέρα. Αυτός ο μέσος όρος ορίζει το OSN activity ενός χρήστη.

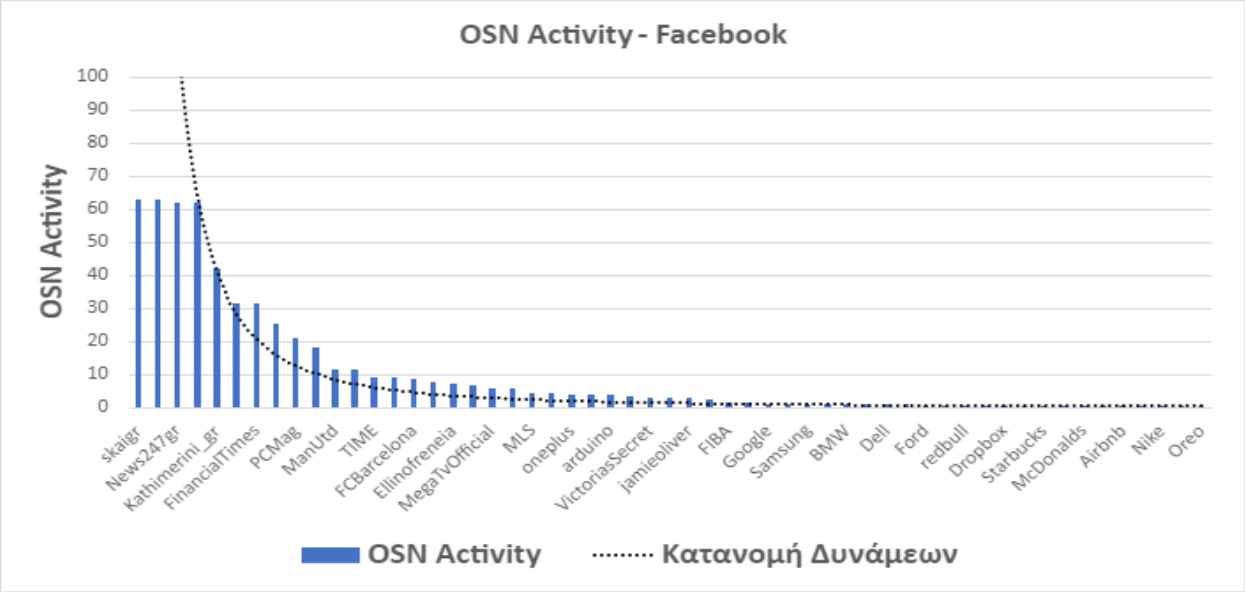
Στους κάθετους άξονες των δύο παρακάτω γραφημάτων παρουσιάζονται οι τιμές των OSN activities, ενώ στους οριζόντιους άξονες εμφανίζονται οι χρήστες. Με μαύρη διακεκομμένη γραμμή ορίζεται η κατανομή νόμου δυνάμεων.

Στην *Εικόνα 106* παρατηρούμε τα OSN activities των χρηστών στο Twitter καταταγμένα σε φθίνουσα σειρά. Βλέπουμε πως τα δεδομένα ακολουθούν την κατανομή δυνάμεων.




***Εικόνα 106: OSN Activity -Twitter***

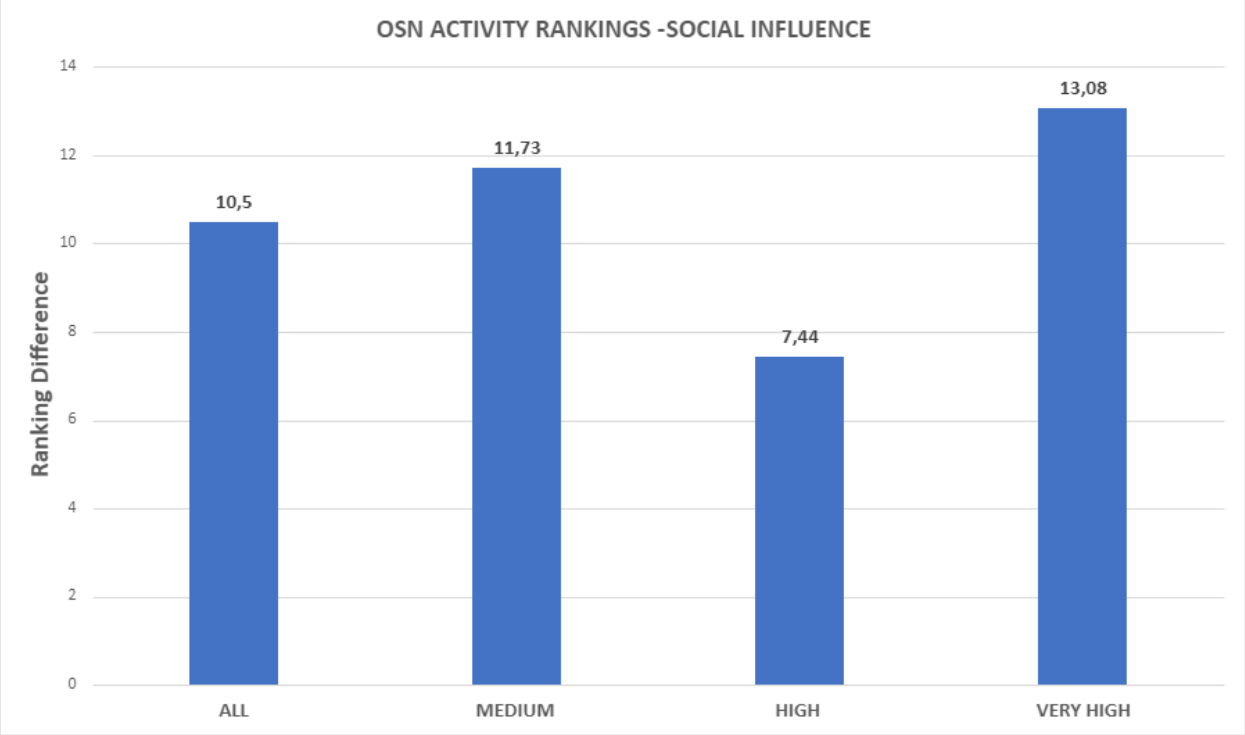
Η ίδια ανάλυση πραγματοποιήθηκε και για την περίπτωση του Facebook (*Εικόνα 107*). Και σε αυτό το διάγραμμα, τα δεδομένα ακολουθούν την κατανομή δυνάμεων.




**Εικόνα 107: OSN Activity - Facebook**

Στην τελευταία ανάλυση κατατάξαμε τους χρήστες με βάση τα OSN activities τους και υπολογίσαμε τις διαφορές στις θέσεις κάθε χρήστη και τους μέσους όρους των διαφορών για κάθε διαφορετική ομάδα. Όπως φαίνεται στην *Εικόνα 108*, οι χρήστες με Υψηλή ΚΕ παρουσιάζουν μεγαλύτερη συνέπεια στην δραστηριότητά τους μεταξύ των δύο ΚΔ. Αντίθετα,

την μεγαλύτερη απόκλιση στις δύο κατατάξεις παρουσιάζει η ομάδα χρηστών με Πολύ Υψηλή ΚΕ. Πιο συγκεκριμένα, η Υψηλή ΚΕ εμφανίζει 7,44 θέσεις διαφορά, με την Μέτρια ΚΕ να ακολουθεί με 11,73 θέσεις και τελευταία να είναι η Πολύ Υψηλή ΚΕ με 13,08 θέσεις.




***Εικόνα 108: OSN Activity - Rankings - Social Influence***

## 5.3. Αξιολόγηση Κοινωνικών Δικτύων

Παρακάτω προστέθηκαν οι έξι αναλύσεις που αφορούν τα metadata που βρίσκονται στα ΚΔ (*Πίνακας 4*). Με πράσινο χρώμα συνδυάζεται η πρώτη θέση, με κίτρινο η δεύτερη και με κόκκινο χρώμα η τρίτη θέση στις μεταξύ τους συγκρίσεις. Με γκρι χρώμα εμφανίζονται οι περιπτώσεις στις οποίες το ΚΔ δεν συμμετέχει στη σύγκριση.

Παρατηρούμε πως το Instagram έχει επικρατήσει και στις τέσσερις συγκρίσεις στις οποίες συμμετείχε. Αν δώσουμε 2 βαθμούς για κάθε πρώτη θέση, 1 για κάθε δεύτερη και 0 βαθμούς για κάθε τρίτη θέση, προκύπτει πως το Instagram είναι πρώτο με 8 βαθμούς, ακολουθεί το Facebook με 6, και τελευταίο το Twitter με 4 βαθμούς. Από αυτόν τον πίνακα καταλήγουμε στο συμπέρασμα πως το Instagram είναι το ΚΔ στο οποίο γίνεται η πιο εκτεταμένη χρήση μεταδεδομένων, ενώ το Twitter δεν δείχνει να προτιμάται τόσο σε σχέση με τα άλλα δύο.

| Metadata \ ΚΔ | Twitter | Facebook | Instagram |
|---|---|---|---|
| **Εικόνα 76: Hashtags Per Post** | | | |
| **Εικόνα 82: Links Per Post** | | | |
| **Εικόνα 86: Media Per Post** | | | |
| **Εικόνα 89: Likes Per Post** | | | |
| **Εικόνα 90: Comments Per Post** | | | |
| **Εικόνα 91: Retweets/Shares Per Post** | | | |

***Πίνακας 4: Σύγκριση Κοινωνικών Δικτύων ως προς τα Metadata***

Στη συνέχεια, μέσω των τριών πινάκων που ακολουθούν, παρουσιάζουμε μια γενική εικόνα των αναλύσεων που πραγματοποιήθηκαν με βάση την ΚΕ .

Ο *Πίνακας 5* αφορά το Twitter και τα μοτίβα συμπεριφορών των τριών κοινωνικών ομάδων σε αυτό.

Με την ίδια βαθμολόγηση με πριν, πρώτη προκύπτει η ομάδα Μέτριας ΚΕ με 8 βαθμούς, δεύτερη έρχεται η ομάδα της Υψηλής ΚΕ με 7 βαθμούς και τελευταία η ομάδα της Πολύ Υψηλής ΚΕ με 6 βαθμούς.

| **TWITTER**<br>**Metadata \ KE** | **Medium** | **High** | **Very High** |
|---|---|---|---|
| **Εικόνα 92: Hashtags Per Post - Social Influence** | | | |
| **Εικόνα 94: Hashtag Per Post - Social Influence - Twitter** | | | |
| **Εικόνα 97: Links Per Post - Social Influence** | | | |
| **Εικόνα 98: Media Per Post - Social Influence** | | | |
| **Εικόνα 99: Likes Per Post - Social Influence** | | | |
| **Εικόνα 103: Retweets/Shares Per Post - Social Influence** | | | |
| **Εικόνα 105: Twitter Mentions Average - Social Influence** | | | |

***Πίνακας 5: Σύγκριση ομάδων Κοινωνικής Επιρροής ως προς τα Metadata – Twitter***

Ο *Πίνακας 6* αφορά το Facebook. Σε αυτή την περίπτωση πρώτη έρχεται η ομάδα της Υψηλής ΚΕ με 10 βαθμούς, δεύτερη έρχεται η ομάδα της Μέτριας ΚΕ και τρίτη έρχεται η ομάδα της Πολύ Υψηλής ΚΕ.

| **FACEBOOK**<br>**Metadata \ ΚΕ** | **Medium** | **High** | **Very High** |
|---|---|---|---|
| **Εικόνα 92: Hashtags Per Post - Social InfluenceError! Reference source not found.** | | | |
| **Εικόνα 95: Hashtag Per Post - Social Influence - Facebook** | | | |
| **Εικόνα 97: Links Per Post - Social Influence** | | | |
| **Εικόνα 98: Media Per Post - Social Influence** | | | |
| **Εικόνα 99: Likes Per Post - Social Influence** | | | |
| **Εικόνα 102: Comments Per Post - Social Influence** | | | |
| **Εικόνα 103: Retweets/Shares Per Post - Social Influence** | | | |

***Πίνακας 6: Σύγκριση ομάδων Κοινωνικής Επιρροής ως προς τα Metadata – Facebook***

Ο *Πίνακας 7* αφορά το Instagram. Με την ίδια βαθμολόγηση, προκύπτει νικήτρια η ομάδα της Πολύ Υψηλής ΚΕ με 6 βαθμούς, δεύτερη η ομάδα της Υψηλής ΚΕ με 5 βαθμούς και τελευταία η ομάδα της Μέτριας ΚΕ με 4 βαθμούς.

| **INSTAGRAM**<br>**Metadata \ ΚΕ** | **Medium** | **High** | **Very High** |
|---|---|---|---|
| **Εικόνα 92: Hashtags Per Post - Social Influence** | | | |
| **Εικόνα 96: Hashtag Per Post - Social Influence - Instagram** | | | |
| **Εικόνα 98: Media Per Post - Social Influence** | | | |
| **Εικόνα 99: Likes Per Post - Social Influence** | | | |
| **Εικόνα 102: Comments Per Post - Social Influence** | | | |

***Πίνακας 7: Σύγκριση ομάδων Κοινωνικής Επιρροής ως προς τα Metadata - Instagram***

Μία πρώτη πολύ εύκολη παρατήρηση είναι πως κάθε ομάδα επιρροής επικρατεί σε κάποιο ΚΔ. Ακόμη συνδυάζοντας όλα τα αποτελέσματα, βλέπουμε πως η ομάδα της Πολύ Υψηλής ΚΕ επικρατεί στο Instagram, το οποίο έχει την μεγαλύτερη χρήση μεταδεδομένων , ενώ αντίθετα η ομάδα Μέτριας ΚΕ επιλέγει το Twitter που έρχεται τελευταίο.

Επίσης παρατηρούμε πως η ομάδα Πολύ Υψηλής ΚΕ επικρατεί στα δεδομένα που αφορούν αλληλεπιδράσεις των άλλων χρηστών στις δημοσιεύσεις τους, ενώ τείνει να αποφεύγει την χρήση hashtag συγκριτικά με τις άλλες δύο ομάδες.

# Ενότητα 6. Συμπεράσματα και μελλοντικές επεκτάσεις

Στόχος της πτυχιακής ήταν να αναλυθεί το περιεχόμενο των λογαριασμών ιδίων χρηστών από τρία διαφορετικά ΚΔ (Twitter, Facebook και Instagram) και να παρατηρηθούν μοτίβα συμπεριφοράς των χρηστών αλλά και των διαφορετικών ομάδων ΚΕ.

Οι συνεισφορές της παρούσης πτυχιακής εργασίας είναι οι εξής:

1. Συλλογή στοιχείων ΚΔ ιδίων χρηστών από πολλαπλά δίκτυα
2. Ανάλυση μοτίβων συμπεριφορών ανά ΚΔ
3. Ανάλυση μοτίβων συμπεριφορών ανά κοινωνική οντότητα
4. Μελέτη χρηστών ανά κοινωνική επιρροή (ΚΕ) και σύγκριση μοτίβων συμπεριφοράς
5. Αποθήκευση ΒΔ και προγραμμάτων σε Github[1] αποθετήριο για ελεύθερη χρήση

Μέσω των αναλύσεων καταλήξαμε στο συμπέρασμα πως η ανάλυση συμπεριφοράς ενός χρήστη ως οντότητα και όχι ως ένας μεμονωμένος λογαριασμός, είναι απολύτως δυνατή και πραγματοποιήσιμη. Επίσης, παρατηρήσαμε πως το Instagram είναι το ΚΔ στο οποίο οι οντότητες είναι πιο διαδεδομένες και περισσότερο χρησιμοποιούμενες στην καθολικότητα των χρηστών με το Facebook να ακολουθεί, και τελευταίο το Twitter. Παρά τις μεγάλες διαφορές στις αριθμητικές τιμές των κοινωνικών οντοτήτων, λαμβάνοντας υπόψιν τα γραφήματα κατανομής νόμου δυνάμεων, παρατηρούνται παρόμοια μοτίβα συμπεριφοράς μεταξύ των ΚΔ.

Στη συνέχεια ακολούθησαν οι αναλύσεις που αφορούσαν τις ομάδες ΚΕ. Καταλήξαμε στην αξιοσημείωτη παρατήρηση πως κάθε μία από τις τρεις ομάδες υπερτερεί των άλλων δύο σε διαφορετικό ΚΔ, όσον αφορά την χρήση των οντοτήτων. Συγκεκριμένα η ομάδα Μέτριας ΚΕ υπερτερεί στο Twitter, η ομάδα Υψηλής ΚΕ στο Facebook, και η ομάδα Πολύ Υψηλής ΚΕ στο Instagram. Συνδυάζοντας την προηγούμενη παρατήρηση με τα αποτελέσματα της Ενότητας 5.3, προκύπτει πως όσο μεγαλύτερη ΚΕ έχει μια ομάδα, προτιμάει να χρησιμοποιεί το ΚΔ που βρίσκεται πιο ψηλά στην κατάταξη χρήσης κοινωνικών οντοτήτων.

Επίσης πραγματοποιήθηκαν αναλύσεις που αφορούσαν την ένταση και συχνότητα της δραστηριότητας των χρηστών στα ΚΔ Twitter και Facebook. Μέσω των εικόνων (*Εικόνα 106* και *Εικόνα 107*), προέκυψε το συμπέρασμα πως υπάρχουν παρόμοια μοτίβα συμπεριφοράς των χρηστών ανάμεσα στα δύο ΚΔ όσον αφορά την συχνότητα δραστηριότητάς τους. Ακόμη, μέσω της *Εικόνα 108* παρατηρήσαμε και συμπεράναμε πως η ομάδα Πολύ Υψηλής ΚΕ εμφανίζει την μεγαλύτερη ασυνέπεια όσον αφορά την συχνότητα δραστηριότητάς τους μεταξύ των δύο ΚΔ.

Κλείνοντας αυτή την πτυχιακή, δημοσιοποιούμε μια ΒΔ και ένα πρωτοποριακό ερευνητικό υπόβαθρο στον τομέα της ανάλυσης δεδομένων από ΚΔ, το οποίο μπορεί να αποτελέσει βάση για μεταγενέστερη έρευνα.

Σε μελλοντική επέκταση της έρευνας αυτής, θα αυξηθούν και θα εξισωθούν οι όγκοι των δεδομένων των τριών ΚΔ. Επίσης θα μελετηθεί το περιεχόμενο των δημοσιεύσεων ως προς την σημασιολογία τους και θα αναζητηθούν μοτίβα συμπεριφορών των χρηστών που δημιουργούνται στα τρία διαφορετικά ΚΔ.

# Ενότητα 7. Βιβλιογραφία


[1] Anshu Malhotra, Luam Totti, Wagner Meira Jr., Ponnurangam Kumaraguru, and Virgilio Almeida. 2012 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining. 6p. 2012. “Studying User Footprints in Different Online Social Networks”.DOI:10.1109/ASONAM.2012.184

[2] Sebei Hiba, Hadj Taieb, Mohamed Ali and Ben Aouicha Mohamed. Knowledge & Information Systems . Nov2020, Vol. 62 Issue 11, p4297-4336. 40p. 2020. “SNOWL model: social networks unification-based semantic data integration”. DOI: 10.1007/s10115-020-01498-5

[3] Vadim Moshkin. 2020 IEEE 14th International Conference on Application of Information and Communication Technologies (AICT). 2020. "The approach to building a graph knowledge base using social media data,"DOI:10.1109/AICT50176.2020.9368794.

[4] Yazhe Wang. PhD Dissertation. 139p. 2014. “Data Preparation for Social Network Mining and Analysis”. 1-139.

[5] Minas Gjoka, Maciej Kurant, Carter T. Butts and Athina Markopoulou. IEEE Journal on Selected Areas in Communications ( Volume: 29, Issue: 9, October 2011). 20p. 2011. "Practical Recommendations on Crawling Online Social Networks" DOI: 10.1109/JSAC.2011.111011.

[6] Fang Mingzhe, Li Yang, Hu Ying, Mao Shuang and Shi Peng. IEEE Access ( Volume: 7). 17p. 2019. “A Unified Semantic Model for Cross-Media Events Analysis in Online Social Networks”. DOI: 10.1109/ACCESS.2019.2899989.

[7] Golbeck Jennifer and Rothstein Matthew. Proceedings of the Twenty-Third AAAI Conference on Artificial Intelligence (2008). 6p. 2008. “Linking Social Networks on the Web with FOAF: A Semantic Web Case”. DOI: 10.5555/1620163.1620249

[8] Gerasimos Razis and Ioannis Anagnostopoulos. 2014 9th International Workshop on Semantic and Social Media Adaptation and Personalization. 8p. 2014. "Semantifying Twitter: The Influence Tracker Ontology”. DOI: 10.1109/SMAP.2014.23.