Πανεπιστήμιο Πειραιώς – Τμήμα Πληροφορικής
Πρόγραμμα Μεταπτυχιακών Σπουδών
«Ψηφιακός Πολιτισμός, Έξυπνες Πόλεις, IoT και Προηγμένες Ψηφιακές Τεχνολογίες»

Μεταπτυχιακή Διατριβή

Τίτλος Διατριβής	**Συναισθηματική ανάλυση δημοσιεύσεων χρηστών κοινωνικών δικτύων με εφαρμογή στο Twitter** **Sentiment Analysis on Twitter Greek Data**
Ονοματεπώνυμο Φοιτητή	**Τζάνα Ασημίνα**
Πατρώνυμο	**Ιωάννης**
Αριθμός Μητρώου	**ΨΠΟΛ/ 19058**
Επιβλέπων	**Ιωάννης Αναγνωστόπουλος, Καθηγητής**

Ημερομηνία Παράδοσης **Δεκέμβριος 2021**

Τριμελής Εξεταστική Επιτροπή

Ιωάννης Αναγνωστόπουλος Καθηγητής	Δημήτριος Βέργαδος Καθηγητής	Γεράσιμος Ραζής Διδάσκων ΠΜΣ

Πίνακας Περιεχομένων

Πίνακας Εικόνων

Εικόνα 50: Κατηγοριοποίηση αρνητικού αποτελέσματος

Ευχαριστίες

Η συγκεκριμένη διατριβή πραγματοποιήθηκε στα πλαίσια του μεταπτυχιακού προγράμματος «Ψηφιακός Πολιτισμός, Έξυπνες Πόλεις, IoT και Προηγμένες Ψηφιακές Τεχνολογίες» του Πανεπιστημίου Πειραιώς. Θα ήθελα σε αυτό το σημείο να ευχαριστήσω τον κύριο Ιωάννη Αναγνωστόπουλο που μου εμπιστεύτηκε το συγκεκριμένο θέμα. Ευχαριστώ ιδιαιτέρως τη μητέρα μου για όλη τη στήριξη και που μου έχει προσφέρει όλα τα χρόνια των σπουδών μου καθώς επίσης το Βίκτορ Ζουκόβσκι για την ενθάρρυνση και τη στήριξη που μου προσέφερε κατά τη διάρκεια των δύο αυτών χρόνων του μεταπτυχιακού προγράμματος.

Περίληψη

Η ανάλυση συναισθήματος η εξόρυξη γνώμης είναι ένα επιστημονικό πεδίο που ολοένα και κεντρίζει το ενδιαφέρον των εταιριών. Με την ανάλυση συναισθήματος μπορούμε να προβλέψουμε αποτελεσματικά πολλές συμπεριφορές μεγάλου πλήθους ανθρώπων. Τα κοινωνικά δίκτυα έχουν πλέον ενταχτεί στη ζωή του σύγχρονου ανθρώπου σε τέτοιο βαθμό που όχι απλά βοηθούν στην επικοινωνία αλλά πλέον επηρεάζουν και υποκινούν αφού πλέον αποτελούν μεγάλο μέρος της ζωής του ανθρώπου.

Ένα μεγάλο μέρος των δεδομένων που παράγονται καθημερινά, παράγεται στα μέσα κοινωνικής δικτύωσης. Συγκεκριμένα το Twitter αποτελεί ένα κοινωνικό δίκτυο μεγάλου επιστημονικού ενδιαφέροντος για τον τομέα της ανάλυσης συναισθήματος αφού το πλήθος των δημοσιεύσεων καθώς και η ευχρηστία του Twitter API, επιτρέπουν την εξόρυξη μεγάλης ποσότητας δεδομένων. Στη παρούσα μελέτη, θα κατασκευαστεί εξ' ολοκλήρου ένα σύστημα συναισθηματικής ανάλυσης από δεδομένα προερχόμενα από το Twitter. Τα δεδομένα αφορούν δημοσιεύσεις (tweets) στην Ελληνική γλώσσα που σχετίζονται με το εμβόλιο του κορονοϊού. Προτείνεται μια υβριδική μέθοδος ανάλυσης συναισθήματος στην οποία αρχικά μαζεύεται ένα ικανοποιητικό δείγμα δημοσιεύσεων οι οποίες αρχικά θα βαθμολογηθούν συναισθηματικά και στη συνέχεια θα χρησιμοποιηθούν σαν input για την εκπαίδευση ενός μοντέλου μηχανικής μάθησης. Τέλος, το μοντέλο που εκπαιδεύτηκε θα γίνει deploy με το web framework Flask.

Abstract

Sentiment analysis or opinion mining is a very interesting scientific field that is currently gaining a lot of interest in a lot of different fields. By using Sentiment Analysis techniques, we can predict efficiently people's mind and behavior around a specific field. Social Media nowadays are not only a way to communicate but they have become a powerful way to manipulate as well as influence individuals. Social media have become a big part of individual's everyday life.

A big percentage of nowadays data is located on social media. Specifically, Twitter is a very powerful tool with incredible scientific interest in sentiment analysis field because of the incredible amount of data that a researcher can mine easily through Twitter API.

This research is proposing a Greek sentiment analysis system from scratch. Our dataset consists of tweets collected from Twitter Streaming API in Greek language. The topic that those data are about is the Covid-19 vaccine that has been a very popular topic some months now. This thesis is proposing a hybrid approach of sentiment analysis that can be used in Greek data. We first collected the dataset that consists of tweets in Greek language. Secondly, we used a rule-based method for classify the tweets in the right sentiment class. Thirdly, we trained a machine learning classifier that used the rated tweets as an input in order to predict the sentiment of a new dataset that we collected. Finally, we made a sentiment analysis simple app by deploying the machine learning model using Flask.

Εισαγωγή

Χωρίς αμφιβολία, ζούμε στον αιώνα των δεδομένων. Οι άνθρωποι ξοδεύουν αμέτρητες ώρες από τη ζωή τους στο κινητό τους τηλέφωνο το οποίο είναι έξυπνο (smarphone). Πιο συγκεκριμένα, ο μέσος χρήστης περνάει πολλές ώρες της καθημερινότητας του στα κοινωνικά δίκτυα. Μέσω των αισθητήρων που διαθέτουν τα smartphones μαζεύουν διαρκώς δεδομένα τα οποία μπορούν να χρησιμοποιηθούν με ποικίλους τρόπους. Η εξέλιξη των smartphones είναι ραγδαία αφού μετρούν μόλις κάποια χρόνια στη ζωή μας. Στις μέρες μας, τα κοινωνικά δίκτυα πλέον δεν απευθύνονται μόνο σε απλούς χρήστες, αλλά βλέπουμε ότι μεγάλες εταιρίες επενδύουν μεγάλα ποσά για τη προώθηση προϊόντων αλλά και την εξόρυξη δεδομένων των χρηστών έτσι ώστε να προτείνουν τα κατάλληλα προϊόντα στον εκάστοτε χρήστη.

Η ραγδαία εξέλιξη της τεχνολογίας αλλά και η ολοένα αυξανόμενη ένταξη των κοινωνικών δικτύων στη ζωή μας, αυξάνουν την ανάγκη για σωστή διαχείριση των δεδομένων καθώς και για ανθρώπους που ξέρουν να τα διαχειρίζονται και να μπορούν να βγάζουν συμπεράσματα από αυτά. Γεγονός αποτελεί η πληθώρα νέων θέσεων εργασίας που να σχετίζονται με τα data. Το φαινόμενο των κοινωνικών δικτύων δεν αποτελεί κάτι καινούργιο στη ζωή μας αφού τα πρώτα κοινωνικά δίκτυα έκαναν την εμφάνιση τους εδώ και αρκετά χρόνια. Ωστόσο, με την άφιξη του Web 2.0 στη ζωή μας, τα κοινωνικά δίκτυα παρουσίασαν τεράστιες αλλαγές ως προς τη διαδραστικότητα τους αλλά και τη δημοφιλία τους. Η εξέλιξη αυτή του διαδικτύου συνέβαλε στο να μετατραπεί ο μέσος χρήστης ο οποίος μέχρι πρότινος ήταν ένας παθητικός αναγνώστης, σε έναν συνδρομητή περιεχομένου.

Βλέπουμε ότι τα κοινωνικά δίκτυα, εξελίσσονται συνεχώς, οι αλγόριθμοι και οι τεχνολογίες που χρησιμοποιούν αλλάζουν με ραγδαίους ρυθμούς και ο επαγγελματίας της τεχνολογίας θα πρέπει να προσαρμόζεται διαρκώς στα νέα δεδομένα. Με την άφιξη των smartphones ο αριθμός των χρηστών αυξήθηκε κατά τεράστιο βαθμό αφού πλέον περίπου το 83% των ενεργών χρηστών προτιμά να συνδεθεί από το smartphone του. Τα smartphones μέσω των αισθητήρων που διαθέτουν, συλλέγουν καθημερινά ένα τεράστιο όγκο δεδομένων των χρηστών τους. Στις μέρες μας, 3.96 δισεκατομμύρια άνθρωποι χρησιμοποιούν τα μέσα κοινωνικής δικτύωσης, αριθμός που αντιστοιχεί στο 50,64% του πληθυσμού ανεξαρτήτου ηλικίας παγκοσμίως. Αυτό σημαίνει λοιπόν ότι υπάρχει ένα σημαντικό ποσοστό δεδομένων στις πλατφόρμες κοινωνικής δικτύωσης.

Η δραματική εξέλιξη της τεχνολογίας που προαναφέραμε σε συνάρτηση με την αύξηση των διαδικτυωμένων χρηστών, έχει μετατρέψει το παγκόσμιο ιστό σε μια δυναμική πλατφόρμα αποθήκευσης, διάδοσης αλλά και εξόρυξης γνώσης μέσω των δεδομένων που συλλέγονται. Ωστόσο, η τεράστια ποσότητα μη δομημένων δεδομένων που συλλέγεται καθημερινά, δημιουργεί ποικίλες προκλήσεις για τον επαγγελματία της τεχνολογίας αλλά και για τους χρήστες οι οποίο έρχονται αντιμέτωποι με το φαινόμενο της υπερπληροφόρησης. Ακόμα υπάρχουν προκλήσεις που αφορούν το ηθικό κομμάτι της συλλογής των δεδομένων αυτών.

Πολλά από τα δεδομένα που συλλέγονται από τα κοινωνικά δίκτυα αποτελούν απόψεις και σκέψεις των χρηστών τους. Με αυτά τα δεδομένα χρησιμοποιώντας τις κατάλληλες τεχνικές μπορούμε να εξάγουμε χρήσιμα συμπεράσματα για τους χρήστες των κοινωνικών δικτύων. Η συναισθηματική ανάλυση των δημοσιεύσεων είναι μια από τις τεχνικές αυτές. Η εργασία αυτή έχει σαν στόχο την εξόρυξη και τη συναισθηματική ανάλυση δημοσιεύσεων χρηστών στο Twitter το οποίο αποτελεί μια πλατφόρμα κοινωνικής δικτύωσης μικρο-ιστολογίου. Συλλέχτηκε και αναλύθηκε ένα δείγμα δημοσιεύσεων χρηστών που αφορούν το εμβόλιο του κορονοϊού.

Η ανάλυση αυτή αποσκοπεί στο να ανακαλυφθεί μέσα από τη συναισθηματική ανάλυση των tweets η άποψη των Ελλήνων σχετικά με το εμβόλιο του κορονοϊού. Φυσικά η ανάλυση αυτή περιέχει πολλές προκλήσεις οι οποίες θα αναλυθούν στα επόμενα κεφάλαια. Το θεωρητικό υπόβαθρο των αλγορίθμων και των τεχνικών που θα χρησιμοποιηθούν αναλύεται διεξοδικά στα παρακάτω κεφάλαια. Θα αναλυθούν επίσης κάποιες τεχνικές εξόρυξης και ανάλυσης δεδομένων χρηστών κοινωνικών δικτύων καθώς και τέλος θα γίνει οπτικοποίηση των συμπερασμάτων που έχουν ανακαλυφθεί από την έρευνα μέσω γραφημάτων. Τέλος, από τα δεδομένα που συλλέχτηκαν και αναλύθηκαν συναισθηματικά θα δημιουργηθεί ένα μοντέλο πρόβλεψης μηχανικής μάθησης το οποίο είναι σε θέση να προβλέπει το συναίσθημα μιας πρότασης. Με βάση

το μοντέλο αυτό, θα δημιουργηθεί μια εφαρμογή με τη βοήθεια του web framework Flask η οποία προβλέπει το συναίσθημα της πρότασης την οποία πληκτρολογεί ο χρήστης.

Επισκόπηση εργασίας

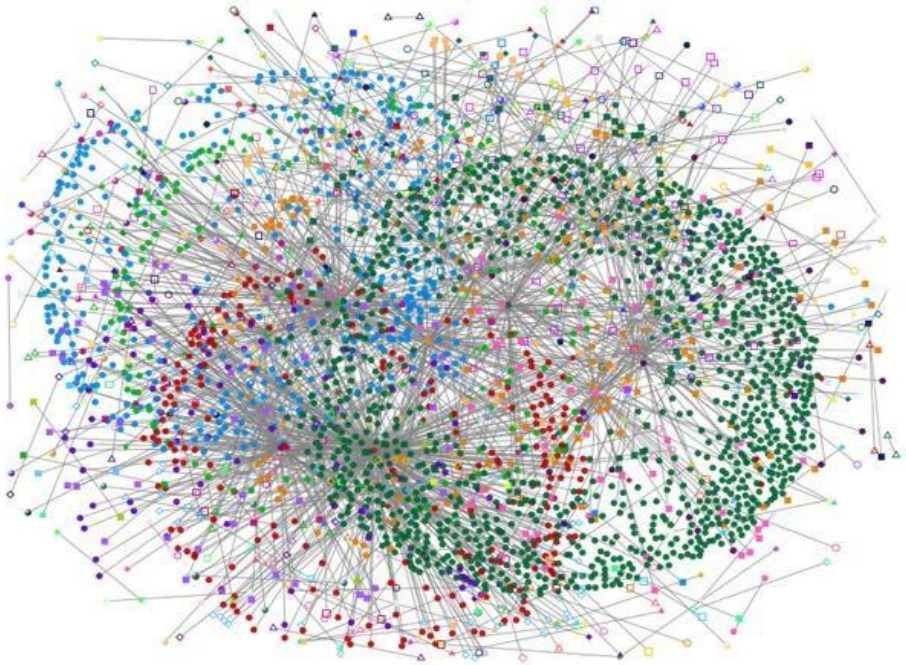
Η ανάλυση συναισθήματος στην Ελληνική γλώσσα αποτελεί μια πρόκληση για το τομέα της Επεξεργασίας της Φυσικής Γλώσσας. Η εργασία αυτή αποτελεί μια ερευνητική προσπάθεια που αφορά το συγκεκριμένο πρόβλημα. Καθ' όλη τη διάρκεια της παρούσας παρουσιάζεται η θεωρία αλλά και πρακτικά το πρόβλημα της συναισθηματικής ανάλυσης κειμένου και συγκεκριμένα στην Ελληνική γλώσσα. Αρχικά παρουσιάζεται το θεωρητικό υπόβαθρο του προβλήματος και αναλύονται οι αλγόριθμοι και οι μέθοδοι που θα χρησιμοποιηθούν καθ' όλη τη διάρκεια της εργασίας. Η γλώσσα που θα χρησιμοποιηθεί σε όλα τα στάδια της εργασίας είναι η γλώσσα Python. Το πρακτικό μέρος της εργασίας είναι χωρισμένο σε έξι διαφορετικά στάδια τα οποία επεξηγούνται διεξοδικά στα επόμενα κεφάλαια.

- Το πρώτο στάδιο της εργασίας περιλαμβάνει την εξόρυξη των δεδομένων από το κοινωνικό δίκτυο Twitter. Η εξόρυξη των δεδομένων πραγματοποιήθηκε με την επικοινωνία με το Streaming API του Twitter το οποίο επιτρέπει την εξόρυξη μεγάλου όγκου δεδομένων σε σχετικά σύντομο χρονικό διάστημα. Μαζεύτηκαν πάνω από 50.000 δημοσιεύσεις που αφορούν το εμβόλιο του κορονοϊού, κάτι το οποίο απασχολεί πολύ την ελληνική πραγματικότητα. Το κοινωνικό δίκτυο Twitter λοιπόν, αποτελεί μια διαδικτυωμένη κοινωνία η οποία κατατάσσει τις δημοσιεύσεις ανάλογα με την ετικέτα (hashtag) που ακολουθεί την εκάστοτε θεματική. Τα δεδομένα που συλλέχτηκαν περιείχαν θόρυβο γι' αυτό και δόθηκε ιδιαίτερη βάση στο στάδιο της προεπεξεργασίας τους.
- Το δεύτερο στάδιο της εργασίας περιλαμβάνει την "γνωριμία" με τα ανεπεξέργαστα δεδομένα τα οποία συλλέχτηκαν. Σε αυτό το στάδιο παρουσιάζονται τα δεδομένα έτσι ώστε να ανακαλυφθούν τα "προβλήματα" που περιέχονται σε αυτά. Τα δεδομένα παρουσιάζονται με γραφήματα έτσι ώστε να είναι πιο κατανοητά προς το ανθρώπινο μάτι.
- Στο τρίτο στάδιο πραγματοποιείται η προεπεξεργασία των δεδομένων, αφαιρείται κάθε είδος θορύβου από αυτά (ετικέτες, αναφορές, greeklish, σημεία στίξης). Σε αυτό το στάδιο επίσης αφαιρούνται οι επαναλαμβανόμενες δημοσιεύσεις και γεμίζονται η αφαιρούνται οι κενές εγγραφές. Στη συνέχεια αφαιρούνται οι τερματικοί όροι (stopwords) καθώς δεν προσδίδουν καμία συναισθηματική αξία στο κείμενο.
- Το τέταρτο στάδιο αποτελεί τη μετάφραση των δεδομένων στην αγγλική γλώσσα μέσω του Google Translate API και την εφαρμογή δύο μεθόδων κατηγοριοποίησης κειμένου (Vader, Textblob) οι οποίες μπορούν να κατανοήσουν μόνο αγγλικό κείμενο και να το κατατάξουν συναισθηματικά σύμφωνα με τη πολικότητα του. Τελικά επιλέγεται η τεχνική Vader για τη κατασκευή του μοντέλου που θα εκπαιδεύσουμε στη συνέχεια.
- Στο πέμπτο στάδιο εκπαιδεύεται ένας κατηγοριοποιητής μηχανικής μάθησης με σκοπό να δημιουργηθεί ένα μοντέλο πρόβλεψης το οποίο μπορεί να χρησιμοποιηθεί αργότερα σε άλλα δεδομένα προβλέποντας το συναίσθημα της εκάστοτε πρότασης.
- Στο έκτο και τελευταίο στάδιο δημιουργείται μια απλή εφαρμογή ανάλυσης συναισθήματος με το web framework Flask η οποία χρησιμοποιεί back end το μοντέλο πρόβλεψης που εκπαιδεύτηκε στο προηγούμενο στάδιο.

1. Κοινωνικά δίκτυα

1.1 Ορισμοί

Ο όρος Κοινωνικό δίκτυο μετρά ήδη την ύπαρξη του κάποιες δεκαετίες. Ήδη από το 1977 Οι Walker, MacBride και Vachon, όρισαν ως κοινωνικό δίκτυο το άθροισμα των προσωπικών επαφών μέσω των οποίων το άτομο διατηρεί την κοινωνική του ταυτότητα, λαμβάνει συναισθηματική υποστήριξη, υλική ενίσχυση και συμμετοχή στις υπηρεσίες, έχει πρόσβαση στις πληροφορίες και δημιουργεί νέες κοινωνικές επαφές. Ένα Κοινωνικό δίκτυο (Social Network) είναι μία κοινωνική δομή αποτελούμενη από ένα σύνολο δραστών (άτομα, οργανισμοί) και ένα σύνολο δεσμών μεταξύ αυτών. Με λίγα λόγια τα κοινωνικά δίκτυα είναι ένα σύνολο αλληλεπιδράσεων και διαπροσωπικών σχέσεων. Σύμφωνα με τη βιβλιογραφία ένα κοινωνικό δίκτυο είναι μία κοινωνική δομή αποτελούμενη από κόμβους (συνήθως άτομα ή επιχειρήσεις) οι οποίοι συνδέονται μμεταξύ τους με έναν ή περισσότερους τύπους αλληλεξάρτησης, όπως αξίες, οράματα, ιδέες, οικονομικές συναλλαγές, φιλία, συγγένεια, αντιπάθεια, συγκρούσεις, σεξουαλικές επαφές, μμεταφορά μμολυσματικών ασθενειών ή επιγραμμικές (web) επαφές. Ο Χτούρης (Χτούρης 2004) ορίζει ως κοινωνικά δίκτυα τα «πολυδιάστατα συστήματα επικοινωνίας και διαμόρφωσης της ανθρώπινης πρακτικής και της κοινωνικής ταυτότητας».

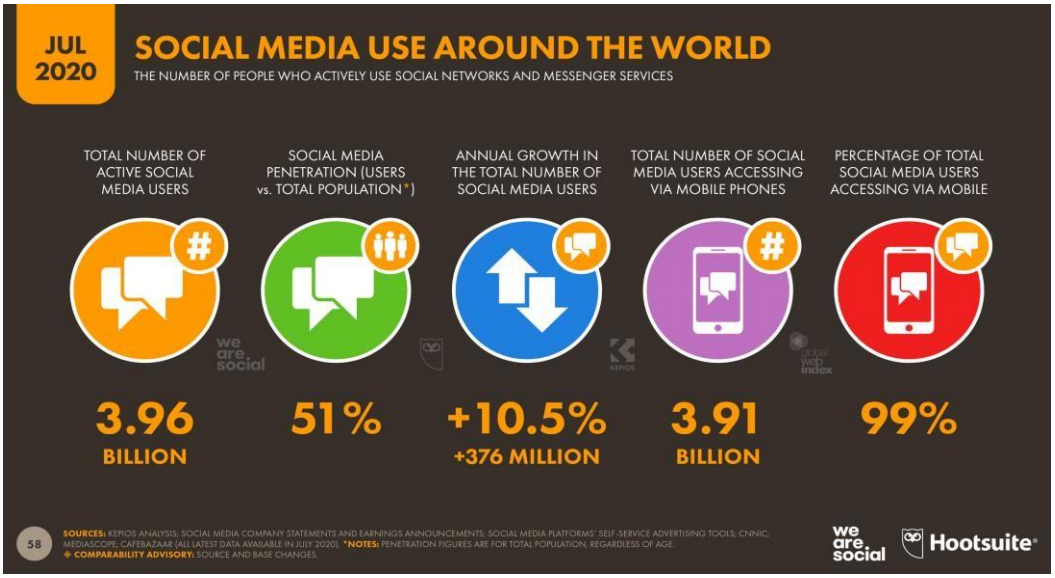


Εικόνα 1: Παράδειγμα γράφου τυχαίου κοινωνικού δικτύου

Σήμερα, ο όρος κοινωνικό δίκτυο (social media) είναι ευρέως γνωστός και χρησιμοποιείται για πλατφόρμες στις οποίες οι χρήστες μπορούν να έχουν διεπαφές μεταξύ τους με διάφορους τρόπους όπως π.χ. σχόλια, διαμοιρασμός φωτογραφιών, ανταλλαγή απόψεων κ.α. Πλέον τα κοινωνικά δίκτυα δεν είναι μόνο ιστοσελίδες επικοινωνίας αλλά έχουν και τεράστια ισχύ σε πολλούς τομείς της ζωής του σύγχρονου ανθρώπου. Οι χρήστες μπορούν να εκφράσουν απόψεις πολιτικές, προσωπικές αλλά ακόμα και να καταγγείλουν περιστατικά που στη πραγματική ζωή ενδεχομένως να μη μπορούσαν να εκφράσουν. Σύμφωνα με τα δεδομένα του dataportal (σχήμα 2) , 3.96 δισεκατομμύρια άνθρωποι είναι ενεργοί στις πλατφόρμες κοινωνικής δικτύωσης εκ των οποίων οι 99% συνδεμένοι από το smarphone τους.

Αυτή η κατακόρυφη αύξηση της χρήσης των smartphones, δημιουργεί την ανάγκη των χρηστών να βρίσκονται συνδεδεμένοι συνεχώς στο διαδίκτυο και συγκεκριμένα στα κοινωνικά

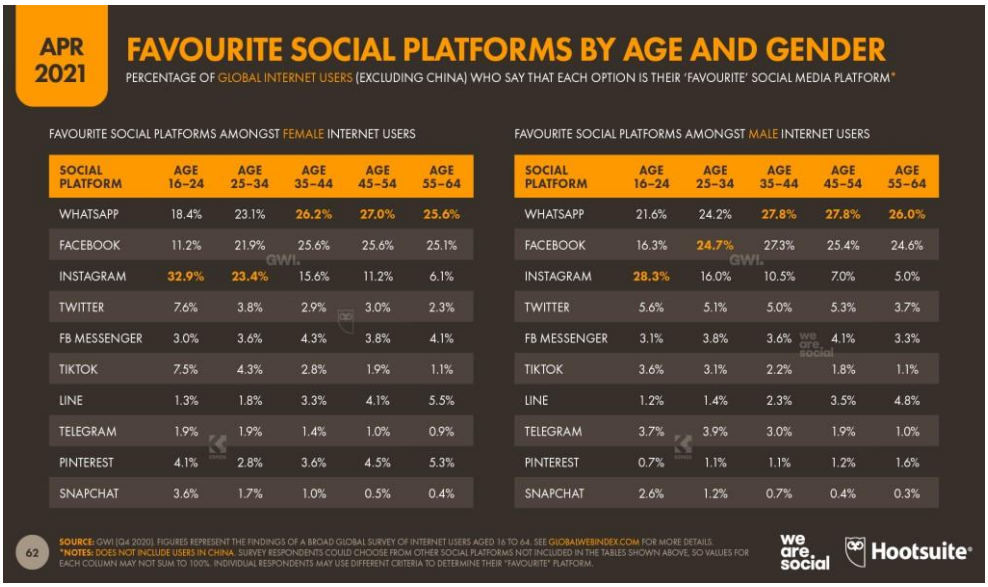
δίκτυα. Έτσι, μέσω των αισθητήρων τους, τα smartphones μαζεύουν τεράστιο όγκο δεδομένων ο οποίος μπορεί να χρησιμοποιηθεί και να αναλυθεί για εξαγωγή συμπερασμάτων αναφορικά με πληθυσμούς. Στα επόμενα κεφάλαια θα αναλύσουμε κάποιες τεχνικές εξόρυξης και ανάλυσης δεδομένων τα οποία συλλέχτηκαν από τυχαίους χρήστες κοινωνικών δικτύων.



Εικόνα 2: Στατιστικά χρηστών κοινωνικών δικτύων 2020

Οι ιστοσελίδες κοινωνικής δικτύωσης αποτελούν εικονικές κοινότητες, όπου οι χρήστες του Διαδικτύου έχουν τη δυνατότητα να δημιουργήσουν τα εικονικά τους προφίλ και να αναπτύξουν ένα δίκτυο επαφών, με τις οποίες μπορούν να επικοινωνούν μέσω της ιστοσελίδας. Μερικές διαδεδομένες πλατφόρμες κοινωνικής δικτύωσης παγκοσμίως αυτή τη στιγμή είναι το Instagram, το Facebook , το YouTube, το LinkedIn, το Twitter και τελευταία έχει γίνει πολύ δημοφιλές το TikTok. Υπάρχουν Κάθε πλατφόρμα έχει μια δικιά της ταυτότητα και γλώσσα και απευθύνεται σε συγκεκριμένες ηλικίες και κοινό. Σύμφωνα με στατιστικά στοιχεία της ιστοσελίδας Statista, το πιο δημοφιλές κοινωνικό δίκτυο είναι το Facebook με 2.797 δισεκατομμύρια χρήστες. Ακολουθεί το YouTube με 2.291 δισεκατομμύρια. Αυτό ωστόσο αναφέρεται στη δημοφιλία του Facebook σύμφωνα με τους ενεργούς χρήστες και όχι στο βαθμό επιρροής. Όσον αναφορά το Μάρκετινγκ και τη προώθηση προϊόντων, η πιο δημοφιλής πλατφόρμα είναι το Instagram.

Επιπλέον, υπάρχουν διακυμάνσεις στις προτιμήσεις χρήσης συγκεκριμένων κοινωνικών δικτύων όσον αναφορά την ηλικία αλλά και σε πολλές περιπτώσεις το φύλο των χρηστών. Βλέπουμε ότι οι γυναίκες 16-34 ετών δείχνουν μια προτίμηση στο Instagram ενώ την αντίστοιχη προτίμηση δείχνουν μόνο οι άντρες 16-24.



Εικόνα 3: Προτίμηση χρήσης κοινωνικών δικτύων ανά ηλικία και φύλο

Σήμερα υπάρχουν δεκάδες σελίδες κοινωνικής δικτύωσης και είναι αδύνατον να τις συμπεριλάβουμε όλες. Στα επόμενα κεφάλαια θα αναλυθούν εκτενέστερα κάποιες από αυτές. Στη παρούσα εργασία θα επικεντρωθούμε στο Twitter, στο Instagram, στο Facebook και στο LinkedIn. Ακόμα θα αναλυθούν τεχνικές εξόρυξης δεδομένων και ανάλυσης αυτών έτσι ώστε να παραχθούν χρήσιμες πληροφορίες για κάποιο πληθυσμό. Επίσης θα εφαρμόσουμε ανάλυση συναισθήματος έτσι ώστε να εξάγουμε πιο συγκεκριμένα συμπεράσματα ως προς τη συμπεριφορά χρηστών. Υπάρχουν πολλά είδη κοινωνικών δικτύων τα οποία εξυπηρετούν διαφορετικούς σκοπούς. Ο βασικός όμως σκοπός των κοινωνικών δικτύων είναι η διαδραστικότητα μεταξύ των χρηστών. Κάποιες από τις κατηγορίες κοινωνικών δικτύων που αναφέραμε είναι οι εξής:

- Δίκτυα επικοινωνίας π.χ. Facebook, VK
- Μικροιστολόγια π.χ. Twitter
- Επαγγελματικά π.χ LinkedIn
- Διαμοιρασμού οπτικοακουστικού περιεχομένου π.χ. Youtube, Instagram, Vimeo, Flickr
- Δίκτυα όπως π.χ. Stack Overflow και Quora στα οποία ο χρήστης μπορεί να κάνει μια ερώτηση η να απαντήσει σε κάποια άλλη.
- Δίκτυα επικεντρωμένα σε εικόνες (Image-based social networks) όπως π.χ. Pinterest, Tmblr, deviantART, We Heart It κ.α.

1.2 Κοινωνικά δίκτυα και Γράφοι

Η θεωρία των γράφων, είναι η μαθηματική θεωρία της μελέτης γράφων η γραφημάτων τα οποία αποτελούν τις σχέσεις μεταξύ διακριτών αντικειμένων. Η θεωρία των γράφων θεωρείται ότι ξεκίνησε από τον Euler στις αρχές του 18ου αιώνα (1736). Η εφαρμογή της θεωρίας των γράφων στα κοινωνικά δίκτυα ονομάζεται ανάλυση κοινωνικών δικτύων και είναι ένας πολύ δημοφιλής τρόπος αναπαράστασης των σχέσεων που διαδραματίζονται εντός των πλατφορμών κοινωνικής δικτύωσης. Στη μαθηματική θεωρία των γράφων, ένας γράφος αποτελείται από κόμβους (ή κορυφές), κάποιοι από τους οποίους (ενδεχομένως όλοι) συνδέονται μμεταξύ τους με μια η παραπάνω σχέσεις (ties). Οι γράφοι μπορούν να είναι:

- *Κατευθυνόμενοι (directed graph)* αν κάθε μια από τις ακμές του είναι προσανατολισμένη προς μία κατεύθυνση.
- *Μη-κατευθυνόμενοι (undirected)* αν οι ακμές του δεν είναι προσανατολισμένες.

Στην περίπτωση που όλοι οι κόμβοι συνδέονται μεταξύ τους έχουμε έναν πλήρη γράφο. Οι συνδέσεις (links) μμεταξύ των κόμβων, όταν υπάρχουν, μπορεί να είναι είτε τόξα ή γραμμές, αναλόγως του αν υπάρχει ή δεν υπάρχει κάποια κατεύθυνση στις συνδέσεις μεταξύ των κόμβων του γράφου. Ο όρος δίκτυο μπορεί να αναφέρεται σε οποιαδήποτε αλληλοσυνδεόμενη ομάδα η σύστημα. Υπάρχουν διάφορα είδη δικτύων όπως δίκτυα επιχειρήσεων, οικονομικά, ξενοδοχείων, κοινωνικά.

Ανατρέχοντας στη βιβλιογραφία μπορούμε να βρούμε πληθώρα ορισμών για να ορίσουμε τα κοινωνικά δίκτυα ως γράφους. Ένας από αυτούς μας παρουσιάζει τα κοινωνικά δίκτυα σαν μία κοινωνική δομή αποτελούμενη από κόμβους (συνήθως άτομα ή επιχειρήσεις) οι οποίοι συνδέονται μμεταξύ τους με έναν ή περισσότερους τύπους αλληλεξάρτησης, όπως αξίες, οράματα, ιδέες, οικονομικές συναλλαγές, φιλία, συγγένεια, αντιπάθεια, συγκρούσεις, σεξουαλικές επαφές, μμεταφορά μμολυσματικών ασθενειών ή επιγραμμικές (web) επαφές.

Οι Walker, MacBride, και Vachon (1977), όρισαν ως κοινωνικό δίκτυο το άθροισμα των προσωπικών επαφών μέσω των οποίων το άτομο διατηρεί την κοινωνική του ταυτότητα, λαμβάνει συναισθηματική υποστήριξη, υλική ενίσχυση και συμμετοχή στις υπηρεσίες, έχει πρόσβαση στις πληροφορίες και δημιουργεί νέες κοινωνικές και επαγγελματικές επαφές. Αυτός ο ορισμός αναμφίβολα περιγράφει τη ζωή του μέσου χρήστη στην εποχή μας αναλογιζόμενοι την ραγδαία αύξηση των κοινωνικών δικτύων αλλά και των χρηστών. Σε ένα κοινωνικά δίκτυο ουσιαστικά μοντελοποιείται η σχέση μεταξύ ανθρώπων, δηλαδή χρηστών. Η αναπαράσταση των κοινωνικών δικτύων σε γράφους είναι μια πολύ χρήσιμη προσέγγιση για την αναπαράσταση σχέσεων μεταξύ οντοτήτων.

Όπως ήδη έχουμε αναφέρει, οι χρήστες των κοινωνικών δικτύων αλληλοεπιδρούν μεταξύ τους. Οι αλληλεπιδράσεις αυτές δεν είναι το ίδιο από χρήστη σε χρήστη. Ένα κοινωνικό δίκτυο μπορεί να αναπαρασταθεί με πολλούς τρόπους. Ένας από αυτούς είναι η θεωρία των γράφων, όπως αναφερθήκαμε και παραπάνω. Στην ουσία σε ένα κοινωνικό δίκτυο, κάθε μονάδα ονομάζεται κοινωνικός δρών (social actor). Ο δρών μπορεί να είναι είτε ένα υποκείμενο, είτε ένα σύνολο ενεργών υποκειμένων με κοινά χαρακτηριστικά η ενδιαφέροντα. Επίσης μπορεί να αντιστοιχεί σε μια συλλογικότητα όπως ένας οργανισμός, μια ιστοσελίδα η ένα blog. Θα μπορούσε ενδεχομένως να είναι και κάποιο κράτος, πόλη η χωριό.

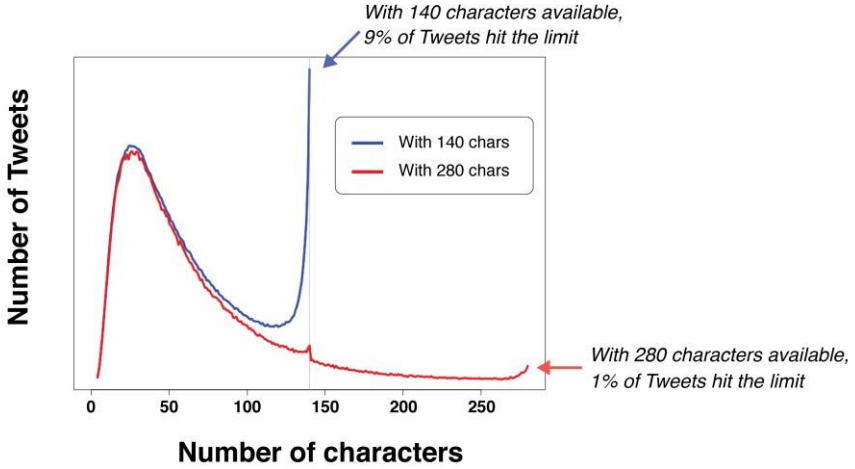
Σύμφωνα με τους Martino και Spoto, 2006, Ο κοινωνικός δρών αναπαρίσταται συνήθως ως ένα σημείο ή κόμβος (node) και η σχέση μμεταξύ δύο σημείων αναπαρίσταται ως μια σύνδεση ή μια ροή μμεταξύ αυτών. Οι δρώντες αναπτύσσουν διαδραστικές σχέσεις μεταξύ τους οι οποίες ονομάζονται interactions η ties. Επιπλέον, κάθε δεσμός διαφοροποιείται ως προς την ισχύ και τη δύναμη από κάποιον άλλον, ανάλογα τον δρών, δηλαδή τον χρήστη. Για την οπτικοποίηση τέτοιου τύπου δικτύων, υπάρχουν πολλά εργαλεία. Μια πολύ χρήσιμη βιβλιοθήκη της Python για τη δημιουργία, διαχείριση και το χειρισμό δικτύων είναι η NetworkX.

1.3 Twitter

Το Twitter αποτελεί μια παγκόσμια, δωρεάν υπηρεσία ανταλλαγής μηνυμάτων η οποία επιτρέπει την εύκολη και άμεση επικοινωνία των χρηστών. Με άλλα λόγια πρόκειται για μια υπηρεσία μικρο-ιστολογίου (microblogging) που επιτρέπει την άμεση δημοσίευση σκέψεων, απόψεων, ιδεών μέσω μηνυμάτων τα οποία λέγονται tweets. Ακόμα οι χρήστες μπορούν να δημοσιεύουν σύντομες ενημερώσεις κατάστασης οι οποίες εμφανίζονται στα timelines. Ακόμα μπορούν να κάνουν αναδημοσίευση το tweet κάποιου άλλου χρήστη μέσω της δυνατότητας του retweet. Ακόμα, η χρήστες του Twitter, οργανώνουν τις δημοσιεύσεις τους με τη βοήθεια του hashtag (#). Ακόμα ο χρήστης μπορεί να αναφέρει τη τοποθεσία από την οποία δημοσιεύει.

Σύμφωνα με τη Wikipedia, το Twitter είναι ένας ιστοχώρος κοινωνικής δικτύωσης που επιτρέπει στους χρήστες του να στέλνουν και να διαβάζουν σύντομα μηνύματα (μέχρι 280 χαρακτήρες), τα οποία ονομάζονται τιτιβίσματα (tweets). Τα μηνύματα μπορούν να αναγνωστούν και από μη συνδεδεμένους χρήστες, αλλά μόνο οι συνδεδεμένοι μπορούν να δημοσιεύσουν κείμενα.

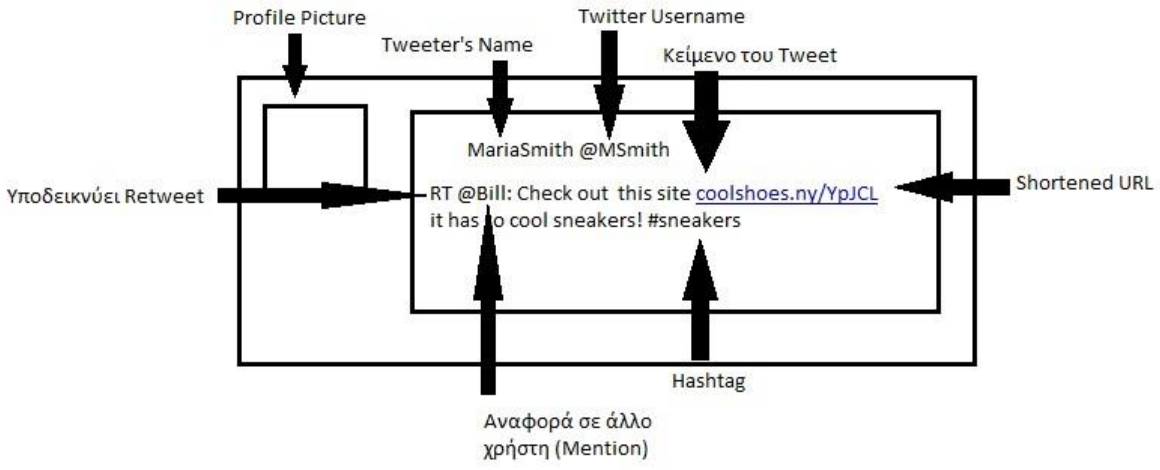
Δημιουργήθηκε την 21 Μαρτίου του 2006 από τον Τζακ Ντόρσεϊ και δημοσιεύθηκε τον Ιούλιο του ίδιου χρόνου. Η υπηρεσία έγινε γρήγορα δημοφιλής και το 2015 έφτασε τους 305 εκατομμύρια ενεργούς χρήστες . Σήμερα απαριθμεί 395 εκατομμύρια χρήστες και βρίσκεται ακόμα στα 17 δημοφιλέστερα κοινωνικά δίκτυα. Τα tweets αρχικά ξεκίνησαν με όριο τους 140 χαρακτήρες, το όριο αυτό όμως διπλασιάστηκε το 2017. Πλέον το όριο των 280 χαρακτήρων εξυπηρετεί πολύ περισσότερο τους χρήστες, οι οποίοι μπορούν πλέον να δημοσιεύουν πολύ πιο αναλυτικά αυτά που θέλουν.



Εικόνα 4: Μελέτη που αφορά τη συχνότητα των tweets ανάλογα τους χαρακτήρες

Οι διαχειριστές του Twitter αποφάσισαν να αλλάξουν αυτό το όριο, αφού ανάλυσαν τα δεδομένα της υπηρεσίας και είδαν ότι η ανάγκη των χρηστών για παραπάνω χαρακτήρες ήταν έντονη (σχήμα 4). Με την αύξηση του ορίου χαρακτήρων λύθηκαν πολλά προβλήματα που αντιμετώπιζαν οι χρήστες όπως π.χ. την υπερβολική προσπάθεια να χωρέσουν τις σκέψεις τους σε 140 χαρακτήρες και έτσι να χάνουν πολύ χρόνο και κατά συνέπεια να μην δημοσιεύουν αυτά που θέλουν να πουν. Στη παρακάτω εικόνα βλέπουμε πως η αύξηση του ορίου των χαρακτήρων επηρέασε τα tweets. Πλέον μόνο το 1% των tweets φτάνουν το όριο των χαρακτήρων σε αντίθεση με το 9% που επικρατούσε πριν. Σύμφωνα με στατιστικά στοιχεία της ιστοσελίδας InternetLiveStat, 9.540 tweets δημοσιεύονται κάθε δευτερόλεπτο.

Ακόμα, είναι σημαντικό να αναφέρουμε ότι το σχεσιακό μοντέλο του Twitter διαφοροποιείται από άλλων κοινωνικών δικτύων όπως π.χ. του Facebook, στο ότι επιτρέπει στους χρήστες να παρακολουθούν δημοσιεύσεις άλλων χρηστών ακόμα και αν δεν είναι άμεσα συνδεδεμένοι (αν δεν κάνουν follow ο ένας τον άλλον). Θα μπορούσαμε λοιπόν να χαρακτηρίσουμε το Twitter σαν ένα προσωπικό δημόσιο online περιοδικό του κάθε χρήστη αφού μέσω αυτού μπορούμε να ενημερωθούμε ακόμα και για σοβαρά θέματα όπως πολιτική η ακόμα και για τα κουτσομπολιά που επικρατούν αυτή τη στιγμή στη χώρα. Ανέκαθεν ο άνθρωπος υπήρξε ένα περίεργο ον και το Twitter ικανοποιεί κατά μεγάλο βαθμό τη περιέργεια αυτή. Εφόσον η πληροφορία στο Twitter δεν έχει ιδιαίτερους περιορισμούς, κάθε tweet είναι μια τεράστια πηγή πληροφορίας που αφορά το χρήστη. Μέσω των tweets μπορούμε να ανιχνεύσουμε και να αναλύσουμε διάφορα γεγονότα.



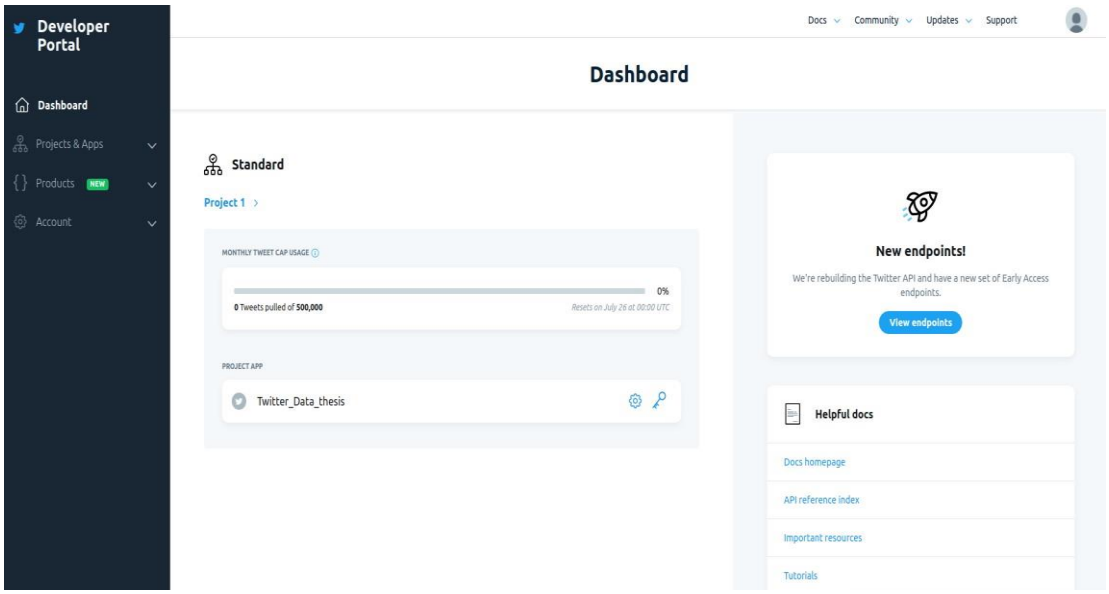
Εικόνα 5: Δομή ενός tweet

Η δομή ενός Tweet φαίνεται πολύ απλή, στη πραγματικότητα όμως υπάρχουν πολλά μεταδεδομένα (metadata) τα οποία μπορούν να χρησιμοποιηθούν για ανάλυση. Για παράδειγμα κάποια από αυτά είναι οι οντότητες και οι θέσεις. Ως οντότητες ορίζουμε τυχόν αναφορές χρηστών, hashtags, διευθύνσεις URL, και δεδομένα (media) που μπορεί να υπάρχουν σε κάποιο tweet. Οι θέσεις είναι οι τοποθεσίες στο πραγματικό κόσμο τις οποίες επισυνάπτει ο χρήστης στο tweet του. Οι τοποθεσίες αυτές μπορεί να είναι είτε οι πραγματικές είτε αυτές που για κάποιο λόγο προσωπικό μπορεί να δήλωσει στο tweet του. Στην εικόνα3 μπορούμε να δούμε τη δομή ενός tweet. Στη συνέχεια της εργασίας θα παρουσιάσουμε τα δεδομένα που μαζεύτηκαν μέσω του Twitter API. Ακόμα θα αναλύσουμε ένα μέρος αυτών και θα προσπαθήσουμε να εξάγουμε κάποια συμπεράσματα.

1.3.1 Twitter API

Σύμφωνα με τη Wikipedia, ένα API (Διεπαφή Προγραμματισμού Εφαρμογών) , από το Application Programming Interface), γνωστή και ως Διασύνδεση Προγραμματισμού Εφαρμογών (για συντομία διεπαφή ή διασύνδεση), είναι η διεπαφή των προγραμματιστικών διαδικασιών που παρέχει ένα λειτουργικό σύστημα, βιβλιοθήκη ή εφαρμογή προκειμένου να επιτρέπει να γίνονται προς αυτά αιτήσεις από άλλα προγράμματα ή/και ανταλλαγή δεδομένων. Τα APIs είναι σημαντικά γιατί μέσω αυτών, οι εταιρείες επιτρέπουν σε third-party developers να αναπτύξουν εφαρμογές που μπορούν να βελτιώσουν τη χρήση και τη λειτουργία της κεντρικής πλατφόρμας. Με αυτόν τον τρόπο, μια επιχείρηση μπορεί να οικοδομήσει ένα οικοσύστημα που εξαρτάται από τα δεδομένα του API της — μια δυναμική που συχνά οδηγεί σε πρόσθετες ευκαιρίες εσόδων.

Για τη χρήση του Twitter API είναι απαραίτητη η δημιουργία ενός λογαριασμού Developer όπως και η απόκτηση των απαραίτητων credentials. Παλαιότερα ήταν πολύ απλό κάποιος να φτιάξει ένα λογαριασμό, όμως πλέον αυτό είναι εφικτό μόνο για κάποιες συγκεκριμένες χρήσεις. Για να πάρει κάποιος λογαριασμό προγραμματιστή συμπληρώνει μια φόρμα και στη συνέχεια το Twitter επικοινωνεί μαζί του μέσω email και ζητάει περισσότερα στοιχεία για την εφαρμογή. Για τις ανάγκες αυτής της εργασίας, έχει δημιουργηθεί ένας λογαριασμός προγραμματιστή. Τα βήματα που ακολουθήθηκαν θα παρουσιαστούν σε επόμενο κεφάλαιο.



Εικόνα 6: Περιβάλλον Twitter Developer

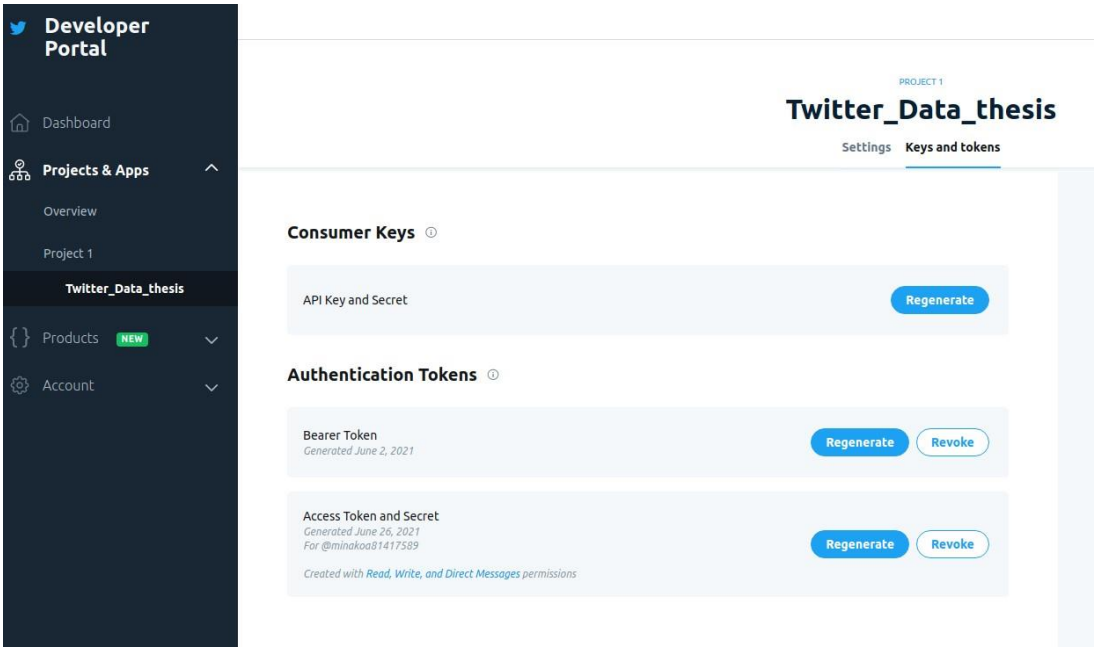
Το Twitter, παρέχει τρεις διαφορετικές διεπαφές προγραμματισμού εφαρμογών, το Search API, το REST API και το Streaming API. Η κάθε διεπαφή προσφέρει άλλες δυνατότητες και ο προγραμματιστής επιλέγει ποια θα χρησιμοποιήσει με βάση τις ανάγκες του Project το οποίο θέλει να υλοποιήσει. Απαραίτητο για τη χρήση κάποιας διεπαφής είναι η πιστοποίηση μέσω του πρωτοκόλλου επικοινωνίας Oauth. Οι απαντήσεις που λαμβάνονται από την εφαρμογή είναι σε μορφή JSON (JavaScript Object Notation). Οι αναζητήσεις μέσω του API μπορεί να περιλαμβάνουν κάποια λέξη κλειδί (hashtag) η κάποιο όνομα χρήστη. Επίσης υπάρχει η δυνατότητα αναζήτησης δεδομένων με βάση τη τοποθεσία.

Το *Streaming API*, παρέχει τη δυνατότητα εξόρυξης tweets σε πραγματικό χρόνο. Έχει τη δυνατότητα να εξάγει το 1% των tweets του δημόσιου χρονολογίου (public timeline), δηλαδή όλα τα δημόσια tweets τα οποία δημοσιεύονται τη συγκεκριμένη χρονική στιγμή στη πλατφόρμα. Να διευκρινίσουμε εδώ ότι το 1% προσφέρει η Sampled μορφή του Streaming API την οποία θα παρουσιάσουμε στη συνέχεια. Η εφαρμογή ανακτά δεδομένα σε πραγματικό χρόνο. Το ποσοστό ακούγεται μικρό όμως αν αναλογιστεί κανείς τη ποσότητα των tweets που δημοσιεύονται καθημερινά, πρόκειται για ένα εύλογο δείγμα δεδομένων που ο προγραμματιστής μπορεί να εξάγει για περαιτέρω ανάλυση.

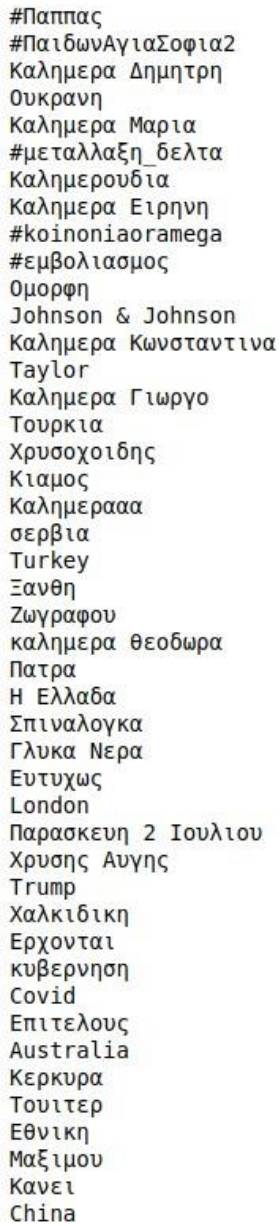
Για να ξεκινήσει λοιπόν η εξαγωγή δεδομένων από το Streaming API, θα δημιουργηθεί ο λογαριασμός προγραμματιστή και στη συνέχεια να ληφθούν τα κατάλληλα credentials. Στη συνέχεια μέσω κάποιας γλώσσας προγραμματισμού μπορεί να επιτευχθεί η σύνδεση και η επαλήθευση των στοιχείων μέσω του πρωτοκόλλου και στη συνέχεια να γραφτεί ο κώδικας για την εξαγωγή και αποθήκευση των δεδομένων. Η γλώσσα Python είναι η καταλληλότερη για τη περίπτωση αυτή αφού προσφέρει πληθώρα βιβλιοθηκών για αυτή τη χρήση. Το documentation του Twitter Developer Platform παρέχει λεπτομερώς όλες τις οδηγίες για το πως να επιτευχθούν αυτά που αναφέραμε. Σε αντίθεση με τις υπόλοιπες διεπαφές που προσφέρει το Twitter, η σύνδεση με το Streaming API είναι μόνιμη, γεγονός που σημαίνει ότι κλείνει μόνο όταν εμείς το θελήσουμε.

Το Streaming API, χρησιμοποιείται για ανάλυση δεδομένων π.χ. για μια εκδήλωση που λαμβάνει χώρα τη στιγμή της σύνδεσης π.χ. μια συναυλία η εκλογές. Το Steaming API χωρίζεται σε 2 παραμέτρους, το *Sampled stream* και το *Filtred stream*. Ουσιαστικά η διαφορά τους είναι ότι στο *Sampled stream* γίνεται ανάκτηση τυχαίων tweets σε πραγματικό χρόνο από το δημόσιο χρονολόγιο ενώ στο Filtred stream υπάρχει δυνατότητα τα δεδομένα να φιλτραριστούν και να επιλεχτούν tweets είτε μέσω λέξεων-κλειδιών(hashtags). Το *Filtred API* με λίγα λόγια επιτρέπει να προστεθούν κανόνες στο κώδικα έτσι ώστε να επιστρέφει συγκεκριμένα δεδομένα που αφορούν

κάποια θεματική ενότητα. Τέλος, στο Streaming API δεν υπάρχει περιορισμός όσον αναφορά τη χρήση και τον αριθμό συνδέσεων.



Το REST API του Twitter, είναι απλό και εύχρηστο και προσφέρει πρόσβαση σε tweets που έχουν ήδη δημοσιευτεί. Οι βιβλιοθήκες που προσφέρονται από τη γλώσσα Python, κάνουν τη διαδικασία επικοινωνίας με το API ακόμα ευκολότερη. Δύο πολύ γνωστές βιβλιοθήκες είναι η βιβλιοθήκη *twitter* και η βιβλιοθήκη *tweepy.* Το REST API του Twitter έχει κάποιους περιορισμούς όσον αναφορά τη χρήση του ο οποίοι αναγράφονται αναλυτικά στο Twitter API documentation. Το REST API δίνει τη δυνατότητα αναζήτησης με περισσότερες επιλογές φιλτραρίσματος των δεδομένων όπως και επίσης εξαγωγή tweets από περισσότερους χρήστες με κάποιους περιορισμούς όμως που αφορούν τη χρήση της υπηρεσίας. Μέσω του REST API μπορούμε να λάβουμε πληροφορίες για ένα χρήστη όπως πόσους ακόλουθους έχει, πόσους ακολουθεί, αλλά και το ιστορικό των tweets του. Ακόμα μπορούμε να εξάγουμε πολλές πληροφορίες που αφορούν ένα tweet. Τέλος, μέσω του WOEID, μπορούμε να βρούμε τα Twitter trends μιας συγκεκριμένης τοποθεσίας.



Εικόνα 7: Εντοπισμός σημαντικότερων θεμάτων της ημέρας μέσω Twitter API

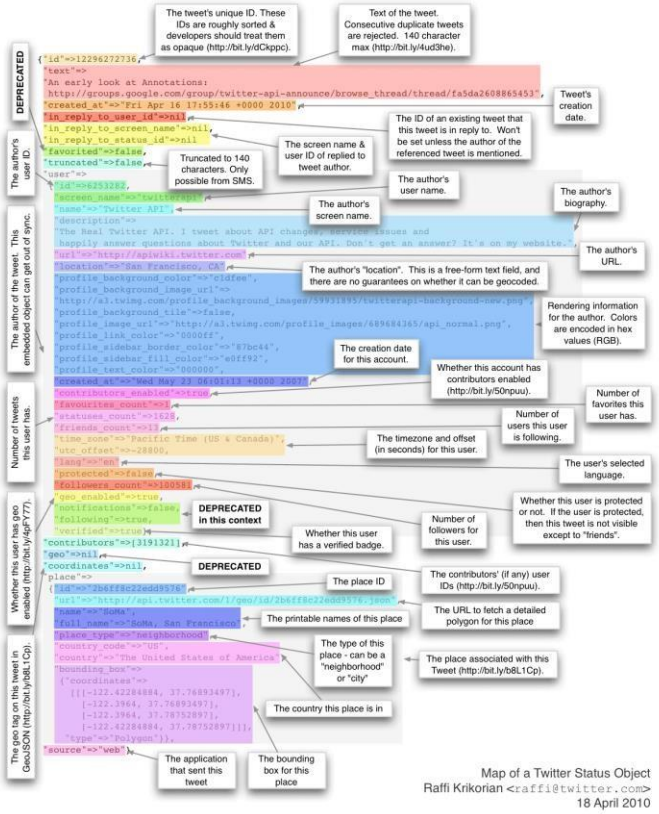
Μέσω του Search API μπορούμε να εξάγουμε tweets τα οποία έχουν δημοσιευτεί τις τελευταίες 7 ημέρες. Το συγκεκριμένο API διατίθεται επίσης και σε Premium και Enterprise έκδοση οι οποίες είναι επί πληρωμή. Οι εκδόσεις αυτές επιτρέπουν την εξαγωγή των tweets των τελευταίων 30 ημερών. Ακόμα επιτρέπουν την πρόσβαση στα tweets που χρονολογούνται από το 2006, τη χρονιά που δημοσιεύτηκε το Twitter. Το Premium API επίσης χωρίζεται σε δυο υποκατηγορίες: τη δωρεάν έκδοση Sandbox με αρκετούς περιορισμούς και την έκδοση Premium η οποία προσφέρει πολύ περισσότερες δυνατότητες. Η Enterprise έκδοση χωρίζεται επίσης σε δυο υποκατηγορίες εκ των οποίων η πρώτη δίνει πρόσβαση στα Tweets των τελευταίων 30 ημερών και η 2η σε όλα τα Tweets που έχουν δημοσιευτεί από το 2006 και μετά.

Feature summary

Category	Product name	Supported history	Query capability	Counts endpoint	Data fidelity
Standard	Standard Search API	7 days	Standard operators	Not available	Incomplete
Premium	Search Tweets: 30-day endpoint	30 days	Premium operators	Available	Full
Premium	Search Tweets: Full-archive endpoint	The entire archive	Premium operators	Available	Full
Enterprise	30-day Search API	30 days	Enterprise operators	Included	Full
Enterprise	Full-archive Search API	The entire archive	Enterprise operators	Included	Full

Εικόνα 8: Δυνατότητες Twitter API

Στην εικόνα (σχήμα 8) βλέπουμε τα μεταδεδομένα και τη δομή ενός tweet. Στα παρακάτω κεφάλαια θα αναλυθεί η διαδικασία που ακολουθήσαμε για την εξαγωγή δεδομένων από το Twitter.

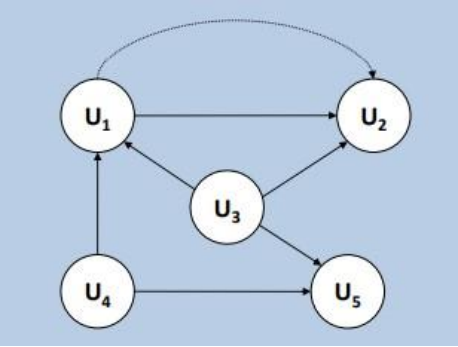


Εικόνα 9: Μεταδεδομένα που υπάρχουν σε ένα tweet

1.3.2 Τύποι σχέσεων στο Twitter

Το Twitter όπως αναφέραμε και σε προηγούμενο κεφάλαιο αποτελεί μια υπηρεσία μικρο-ιστολογίου. Αυτό σημαίνει ότι μεγάλο μέρος της λειτουργίας του Twitter αποτελούν οι σχέσεις που δημιουργούνται εντός του δικτύου. Αξίζει να αναφέρουμε επίσης ότι το σχεσιακό μοντέλο του Twitter δεν απαιτεί αμοιβαίες σχέσεις μεταξύ χρηστών, κάτι που δημιουργεί μεγάλη ευελιξία στο κομμάτι της διαδραστικότητας των χρηστών. Όπως ήδη αναφέραμε, στο Twitter οι κόμβοι αποτελούνται από τους χρήστες ενώ οι ακμές δείχνουν τις σχέσεις η ροές μεταξύ των κόμβων. Οι σχέσεις που αναπτύσσονται στο Twitter διακρίνονται σε τρια δίκτυα:

- *Δίκτυα αναδημοσίευσης (Retweets networks)*: στα δίκτυα αναδημοσίευσης ο κόμβος προέλευσης (source) αναδημοσιεύει τον κόμβο προορισμού (target). Γενικά, τα retweets σηνύθως δηλώνουν συμφωνία στη δημοσίευση του χρήστη. Ωστόσο, τα retweets δεν αντιπροσωπεύουν απαραίτητα υποστήριξη. Επομένως τα retweets μπορούν είτε να δηλώνουν συμφωνία είτε να είναι ουδέτερα και να αναδημοσιεύονται σαν μεταφορά κάποιας πληροφορίας. Στην εικόνα (σχήμα 8) αναπαρίσταται ένας γράφος αναδημοσίευσης όπου ο χρήστης U_1 αναδημοσιεύει το tweet του χρήστη U_2. Οι χρήστες του U_1 (U_3, U_4) θα δουν επίσης το αρχικό tweet.



Εικόνα 10: Αναπαράσταση γράφου δικτύου αναδημοσίευσης στο Twitter

- *Δίκτυα σχολιασμού (Quote networks)*: σε αυτό το τύπο δεσμού, ο χρήστης μπορεί να σχολιάσει κάποιο Tweet εκφράζοντας την άποψη του η απλά προσθέτοντας μια επιπλέον πληροφορίας στη δημοσίευση. Ο χρήστης που κάνει το σχόλιο αποτελεί τον κόμβο προέλευσης (source) ενώ ο χρήστης που έκανε τη δημοσίευση αποτελεί το κόμβο προορισμού (target).

- *Δίκτυα απαντήσεων (Reply networks)*: σε αυτό το δίκτυο ο χρήστης προέλευσης (source) απαντάει σε ένα tweet από κάποιον χρήστη-στόχο (target). Το tweet μπορεί να προέρχεται από κάποια φίλο η συγγενή είτε από κάποιον χρήστη με μεγάλη επιρροή στο δίκτυο. Η απάντηση μπορεί να εκφράζει συμφωνία είτε διαφωνία.

1.3.3 Ανάλυση δεδομένων στο Twitter

Το Twitter αποτελεί μια πολύ δυναμική πλατφόρμα κοινωνικής δικτύωσης η οποία μέσω της διεπαφής που προσφέρει μπορούμε να εξάγουμε και να αναλύσουμε πληθώρα δεδομένων. Καθημερινά δημοσιεύονται εκατομμύρια Tweets. Μέσω τον εργαλείων που προσφέρονται για αυτό το σκοπό μπορούμε να συγκεντρώσουμε και να αναλύσουμε πολύ μεγάλο όγκο δεδομένων

σε πολύ μικρό χρονικό διάστημα. Μια πολύ ενδιαφέρουσα προσέγγιση προτάθηκε από τον Rogers (2013), στην οποία ανέλυσε την εξέλιξη του Twitter και το πόσο αυτό προσελκύει τους ερευνητές. Σύμφωνα με αυτή τη προσέγγιση, το Twitter έχει περάσει από τρια στάδια εξέλιξης:

- *Twitter I:* αναφέρεται στο πρώιμο στάδιο της πλατφόρμας, όταν ακόμα οι χρήστες το χρησιμοποιούσαν μόνο για να συνδεθούν με ανθρώπους καθώς και σαν μέσο επικοινωνίας. • *Twitter II*: αποτελεί τα επόμενα χρόνια λειτουργίας του Twitter, όταν άρχισε να κινητοποιεί πλήθη ανθρώπων μέσω των Tweets. Το Twitter άρχισε να αποκτά ισχύ.
- *Twitter III*: αποτελεί την εξέλιξη του Twitter, όπως το γνωρίζουμε σήμερα. Σήμερα το Twitter αποτελεί μια τεράστια βάση δεδομένων, μέσο της οποίας μπορούμε να εξάγουμε χρήσιμες πληροφορίες που αφορούν τη κοινωνία και τις συμπεριφορές των χρηστών. Επιπλέον μπορούμε να ανατρέχουμε και να αναλύουμε ιστορικά γεγονότα. Η ανάλυση των δεδομένων που παρέχει το Twitter, μπορούν να βοηθήσουν τους επιστήμονες πολλών κλάδων σε σωστότερες αποφάσεις και καλύτερες προσεγγίσεις σε θέματα που αφορούν την ανθρωπότητα.

Το Twitter μπορεί να χρησιμοποιηθεί ως μέτρο της κοινής γνώμης για σημαντικά πολιτικά ή κοινωνικά θέματα. Τα δεδομένα του Twitter έχουν χρησιμοποιηθεί για την ανάλυση της πολιτικής πόλωσης, της κοινής γνώμης των παγκόσμιων ηγετών και της εξάπλωσης κινημάτων διαμαρτυρίας. Το Twitter ανήκει στη κατηγορία των έμμεσων δικτύων. Αυτό σημαίνει ότι οι συνδέσεις μεταξύ χρηστών αναπτύσσονται μέσω κοινών χαρακτηριστικών τους. Στο Twitter, το σχεσιακό μοντέλο δεν απαιτεί αμοιβαία συμφωνία όπως για παράδειγμα σε ένα άμεσο δίκτυο όπως το Facebook.

Στο Twitter οι χρήστες συνδέονται μεταξύ τους βάση κάποιον κοινών γνωρισμάτων. Στην ανάλυση δεδομένων αυτό μας δημιουργεί μεγαλύτερο εύρος δεδομένων αφού μπορούμε να έχουμε πρόσβαση σε οποιοδήποτε πρόσωπο θέλουμε και να αναλύσουμε τα Tweets του έτσι ώστε να οδηγηθούμε σε κάποια συμπεράσματα. Ακόμα μπορούμε να συγκρίνουμε δυο η περισσότερους χρήστες με βάση τα χαρακτηριστικά τους. Μέσω των μετρικών και των αλγορίθμων που προσφέρονται μπορούμε να αναλύσουμε:

- Τη κοινωνική επιρροή των χρηστών • Να αναλύσουμε συναισθήματα μιας ομάδας ανθρώπων για κάποιο συγκεκριμένο γεγονός μέσω των ετικετών (hashtags)
- Να εντοπίσουμε τα πιο δημοφιλή θέματα μέσω των ετικετών (hashtags)
- Να εντοπίσουμε γεγονότα μέσω της γεωγραφικής θέσης. Ακόμα μπορούμε να αναλύσουμε γεγονότα σε πραγματικό χρόνο. Τα γεγονότα αυτά μπορούν να προβλέψουν αποτελέσματα πριν αυτά συμβούν όπως για παράδειγμα αποτελέσματα εκλογών η κάποιου διαγωνισμού τραγουδιού.

Σκοπός αυτή της έρευνας είναι να μελετήσει τα συναισθήματα του ελληνικού λαού σχετικά με το εμβόλιο του κορονοϊού και να βγάλει συμπεράσματα σχετικά με το αν οι Έλληνες είναι θετικοί η όχι στο ενδεχόμενο εμβολιασμού. Τα δεδομένα προέρχονται κυρίως από χρήστες του Twitter, βλέπουμε ανάμεσα σε αυτούς πολιτικά πρόσωπα, κανάλια αλλά και προφίλ εφημερίδων οι οποίες αναδημοσιεύουν περιεχόμενο σχετικά. Στα παρακάτω κεφάλαια θα αναλύσουμε και θα οπτικοποιήσουμε τα δεδομένα που έχουν εξαχθεί από τη πλατφόρμα Twitter. Η γλώσσα που θα χρησιμοποιηθεί για την υλοποίηση του συστήματος είναι η Python. Η γλώσσα Python προσφέρει πληθώρα πακέτων και βιβλιοθηκών που βοηθούν για την εξόρυξη και ανάλυση δεδομένων.

2. Εξόρυξη Δεδομένων

2.1 Εξόρυξη γνώσης από δεδομένα

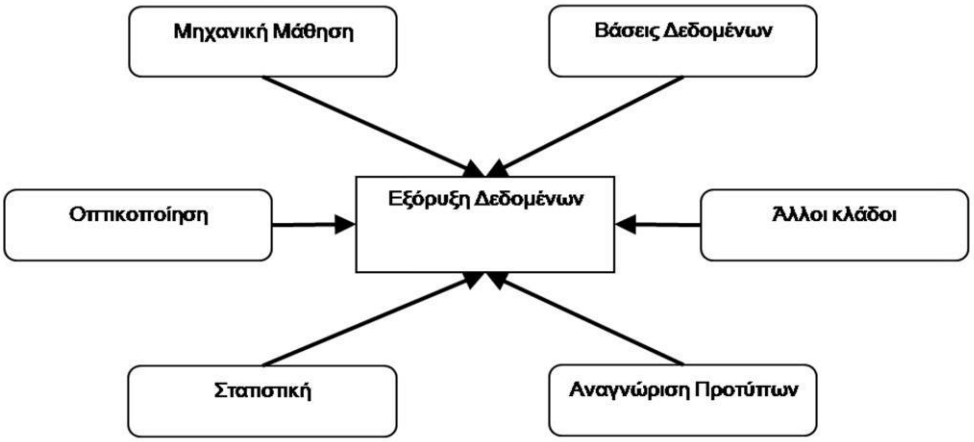
Όπως αναφέραμε σε προηγούμενα κεφάλαια, διανύουμε τον αιώνα της τεχνολογίας και κατά συνέπεια των δεδομένων. Ωστόσο, ο όγκος των δεδομένων που χρειάζονται ανάλυση, είναι πολύ μεγάλος γεγονός που δημιουργεί μεγάλες προκλήσεις «Έχουμε κατακλυστεί από δεδομένα, όμως μας λείπει η πληροφορία». Η έκφραση αυτή δηλώνει πόσο σημαντικός είναι ο τομέας της εξόρυξης γνώσης από δεδομένα αλλά και πόσο χρειαζόμαστε τη γνώση. Ακόμα αντικατοπτρίζει τη ποσότητα των δεδομένων που παράγεται από την ανθρωπότητα.

Στόχος της εξόρυξης δεδομένων είναι η πληροφορία που θα εξαχθεί και τα πρότυπα που θα προκύψουν να έχουν δομή κατανοητή προς τον άνθρωπο έτσι ώστε να τον βοηθήσουν να πάρει τις κατάλληλες αποφάσεις. (Κουρής, 2006). Η διαδικασία της εξόρυξης δεδομένων δεν είναι κάτι καινούργιο, ωστόσο όσο τα δεδομένα αυξάνονται τόσο αυξάνεται και η ανάγκη για γνώση, πρότυπα και συσχετίσεις προερχόμενες από αυτά. Φυσικά αυτή η διαδικασία δεν είναι δυνατόν να πραγματοποιηθεί χειροκίνητα, επομένως χρειάζονται τα κατάλληλα εργαλεία για να γίνει αυτή η διαδικασία όσο πιο σύντομη γίνεται.

Η εξόρυξη δεδομένων η αλλιώς εξόρυξη γνώσης είναι ένα σύμπλεγμα όλων αυτών που αναφέραμε. Είναι στην ουσία μία διαδικασία μη τετριμμένης εξαγωγής άγνωστων όπως συσχετίσεις, κανόνες γνώσης, πρότυπα, κανονικότητες και εύρεσης πληροφοριών από ένα μεγάλο όγκο δεδομένων (Λαδάς,2014). Ειδικότερα, πρόκειται για μια διαδικασία η οποία έχει ως στόχο την ανακάλυψη χρήσιμης γνώσης και προτύπων (patterns), τα οποία υπάρχουν σε μεγάλες βάσεις δεδομένων. Τα πρότυπα αυτά ουσιαστικά βρίσκονται κρυμμένα αντιστοιχούν σε στατιστικά στοιχεία ενός συνόλου δεδομένων.

Ο όρος Εξόρυξη Δεδομένων είναι ευρέως γνωστός και ως: ανακάλυψη γνώσης σε βάσεις δεδομένων (knowledge discovery in databases η KDD), εξόρυξη γνώσης από βάσεις δεδομένων (knowledge mining from databases), εξαγωγή γνώσης (knowledge extraction). Οι Maimon and Rokach (2005), χρησιμοποιούν τον όρο Ανακάλυψη Γνώσης σε Βάσεις Δεδομένων (Knowledge Discovery in Databases – KDD) για τη συνολική διαδικασία ανακάλυψης προτύπων μέσα από μεγάλα και περίπλοκα σύνολα δεδομένων. Για να γίνει σωστά η διαδικασία, είναι απαραίτητη η συνεργασία επαγγελματιών από πολλούς διαφορετικούς κλάδους της επιστήμης της πληροφορικής.

Σύμφωνα με τη βιογραφία η επιστήμη της Στατιστικής προσφέρει λύσεις ανάλυσης δεδομένων, δεν λαμβάνει όμως μέριμνα για το πρόβλημα του πολύ μεγάλου όγκου τους. Επίσης η Μηχανική Μάθηση και η Αναγνώριση Προτύπων διαθέτουν τις δικές τους μεθοδολογίες, όμως και πάλι δεν αντιμετωπίζουν το πρόβλημα του όγκου των δεδομένων. Ο κλάδος των Βάσεων Δεδομένων είναι ο κατ’ εξοχήν αρμόδιος για την τήρηση μεγάλου όγκου δεδομένων, όμως η σχεδιαστική φιλοσοφία του είναι προσανατολισμένη στην καταχώρηση, στη διαχείριση και στην ανάκτηση των δεδομένων, όχι όμως και στην ανάλυση τους. Πρόκειται λοιπόν για ένα διεπιστημονικό κλάδο ο οποίος δημιουργήθηκε χάρη στη συνεργασία των επιστημών που αναφέραμε. Στο παρακάτω σχήμα (σχήμα 9) αναφέρονται οι επιστημονικοί κλάδοι αυτοί:



Εικόνα 11: Η Εξόρυξη Δεδομένων ως αποτέλεσμα συμβολής άλλων κλάδων

2.2 Εξόρυξη Κειμένου

Στα προηγούμενα κεφάλαια της παρούσας εργασίας μιλήσαμε για την εξόρυξη γνώσης από δεδομένα. Η εξόρυξη γνώσης από κείμενα λοιπόν, δεν διαφέρει κατά πολύ ως προς τη διαδικασία και τη μεθοδολογία, ωστόσο διαφέρει ως το περιεχόμενο. Η εξόρυξη γνώσης από κείμενα είναι ένα ερευνητικό πεδίο το οποίο σχετίζεται άμεσα με την Επεξεργασία της Φυσικής Γλώσσας (NLP) η οποία ασχολείται με τις αλληλεπιδράσεις μεταξύ των υπολογιστών και των ανθρώπινων (φυσικών) γλωσσών (Βλαχάβας et al. , 2006).

Ανατρέχοντας στη βιβλιογραφία, βλέπουμε ότι υπάρχει πληθώρα ορισμών. Θα μπορούσαμε να πούμε ότι εξόρυξη γνώσης από κείμενα επομένως αποτελεί υποκατηγορία της εξόρυξης δεδομένων. Το λεξικό της Οξφόρντης ορίζει την εξόρυξη κειμένου ως τη διαδικασία ή την πρακτική της εξέτασης μεγάλων συλλογών γραπτών πόρων για τη δημιουργία νέων πληροφοριών. Στόχος της εξόρυξης κειμένων η αλλιώς της Ανακάλυψης Γνώσης από Κείμενα, αποτελεί η ανακάλυψη χρήσιμης γνώσης, προτύπων και τάσεων από δεδομένα φυσικής γλώσσας. Τα πρότυπα που αναζητούνται αποτελούν αδόμητα δεδομένα προερχόμενα από το παγκόσμιο ιστό και όχι δομημένα προερχόμενα από βάσεις δεδομένων. Η εξόρυξη κειμένων και δεδομένων γίνεται σε σώματα κειμένων ή ψηφιακών/ψηφιοποιημένων δεδομένων ηλεκτρονικής μορφής.

Η εξόρυξη κειμένων είναι μια περιοχή με πολύ μεγάλο επιστημονικό ενδιαφέρον, η οποία έχει αναπτυχθεί χάρη στην εξέλιξη της τεχνολογίας αλλά και τη ραγδαία ανάπτυξη των κειμένων που υπάρχουν σε ηλεκτρονική μορφή. Η αδόμητη μορφή των κειμένων αυτών αποτελεί μεγάλη πρόκληση για την επιστήμη της εξόρυξης κειμένου καθώς προϋποθέτει την σχεδίαση μεθόδων και αλγόριθμων οι οποίοι να μπορούν να διαχειριστούν το μεγάλο όγκο πληροφορίας αλλά το κυριότερο, να μπορούν να επεξεργαστούν τη πληροφορία αυτή.

Το Web 2.0 άλλαξε ριζικά το τρόπου που οι άνθρωποι επικοινωνούν και εκφράζονται στο διαδίκτυο. Σίγουρα έχει επίσης αλλάξει και το τρόπο σκέψης τους. Πλέον μεγάλο μέρος της ζωής του σύγχρονου ανθρώπου ξοδεύεται στο κοινωνικά δίκτυα, επικοινωνώντας με μηνύματα η δημοσιεύοντας αναρτήσεις με περιεχόμενο κειμένου. Οι αναρτήσεις αυτές είναι γραμμένες σε φυσική γλώσσα και η ανάλυση τους παρουσιάζει μεγάλο ενδιαφέρον στους αναλυτές. Η ανάλυση της φυσικής γλώσσας μπορεί να προσφέρει πολύ χρήσιμα συμπεράσματα για την ανθρώπινη συμπεριφορά σε πολλούς τομείς που παλαιότερα δεν ήταν εφικτό να αναλυθούν.

Τα δεδομένα κειμένου κρύβουν πολλές χρήσιμες πληροφορίες αλλά και πολλές προκλήσεις για τον αναλυτή οι οποίες θα αναλυθούν στη συνέχεια. Οι πηγές των κειμένων που

χρησιμοποιούνται για την εξόρυξη κειμένων είναι συλλογές κειμένων προερχόμενες από διάφορες πηγές. Μερικές από τις πηγές των κειμένων αυτών αποτελούν και τα κοινωνικά δίκτυα και συγκεκριμένα οι υπηρεσίες μικρο-ιστολογίου (π.χ. Twitter) οι οποίες θα μας απασχολήσουν στη συνέχεια της εργασίας. Στο μεγάλο ποσοστό τους όπως αναφέρθηκε και παραπάνω, τα δεδομένα αποτελούν αδόμητα σύνολα όπως HTML αρχεία, emails, έγγραφα πλήρους κειμένου κλπ.

2.3 Εφαρμογές Εξόρυξης Δεδομένων

Η εξόρυξη δεδομένων χρησιμοποιείται σήμερα σε πολλούς κλάδους όπως για παράδειγμα η ιατρική, η οικονομία, οι τηλεπικοινωνίες, το μάρκετινγκ. Πιο συγκεκριμένα μερικές εφαρμογές της εξόρυξης δεδομένων είναι:

- Ανάλυση αγοραστικής συμπεριφοράς π.χ. στενευμένο μάρκετινγκ (Δημιουργία συστάδων πελατών με ίδια χαρακτηριστικά (ενδιαφέρον, επίπεδο εισοδήματος, συνήθειες δαπανών κ.λπ.), ανάλυση καλαθιού αγοράς (market basket analysis), διασταύρωση πωλήσεων, τμηματοποίηση αγοράς.

- Ανάλυση κινδύνου: πρόβλεψη τάσεων, έλεγχος ποιότητας, ανάλυση ανταγωνισμού

- Ανάλυση δεδομένων τραπεζικών συναλλαγών: έλεγχος ποιότητας και Ανάλυση ανταγωνιστικότητας

- Ανάλυση ιατρικών δεδομένων: πρόβλεψη ασθένειας η μη, διάγνωση για ύπαρξη ασθένειας η όχι (π.χ. όγκων), μέτρηση της αποτελεσματικότητας κάποιου φαρμάκου.

- Ανίχνευση ψευδών ειδήσεων/απάτης: μέσω ομαδοποίησης και κατασκευής μοντέλου για ανίχνευση απάτης. Το μοντέλο αυτό θα μπορούσε να εφαρμοστεί σε πολλούς τομείς όπως π.χ. Υγειονομική περίθαλψη, υπηρεσία πιστωτικών καρτών, τηλεπικοινωνίες. Θα μπορούσε ακόμα να εφαρμοστεί σε OnLine blogs ώστε να διασταυρωθεί η αξιοπιστία της είδησης.

- Web Mining/Text Mining: ανάλυση δεδομένων που προέρχονται από το παγκόσμιο ιστό. Αυτά τα δεδομένα προέρχονται από οποιαδήποτε ιστοσελίδα και μπορεί να είναι σε οποιαδήποτε μορφή. Αυτά τα δεδομένα μπορεί να είναι π.χ. φωτογραφίες, κείμενα, cookies, email, κωδικοί πρόσβασης κλπ. Ο όρος text Mining αναφέρεται στην εξόρυξη γνώσης (ανάλυση συναισθήματος) από κείμενα που βρίσκονται στο παγκόσμιο ιστό π.χ. tweets. Τη συγκεκριμένη κατηγορία εξόρυξης γνώσης θα την αναλύσουμε στη συνέχεια της εργασίας.

- Ανάλυση δεδομένων κοινωνικών δικτύων: Ανάλυση των δεδομένων που παράγονται στο κοινωνικά δίκτυα. Τα δεδομένα αυτά μπορεί να προέρχονται από δημοσιεύσεις, φωτογραφίες, σχόλια, tweets κλπ.

Εικόνα 12: Εφαρμογές Εξόρυξης Δεδομένων

2.4 Μεθοδολογία Εξόρυξης Δεδομένων

Η εξαγωγή χρήσιμης γνώσης είναι μια περίπλοκη και χρονοβόρα διαδικασία. Προκειμένου όμως να εξαχθούν όσο το δυνατό σωστότερα συμπεράσματα είναι απαραίτητο να ακολουθηθεί η παρακάτω διαδικασία. Η εξόρυξη δεδομένων στη πραγματικότητα αποτελεί ένα βήμα στη συνολική διαδικασία. Η διαδικασία αυτή είναι ουσιαστικά ένας κύκλος διεργασιών. Κάθε στάδιο αυτής της διαδικασίας είναι σημαντικό αφού από το αποτέλεσμα του εξαρτάται το επόμενο. Για τη διαχείριση έργου εξόρυξης γνώσεις έχουν παρουσιαστεί αρκετά μοντέλα. Δυο δημοφιλείς μεθοδολογίες της εξόρυξης δεδομένων είναι οι εξής:

- Cross-Industry Standard Process for Data Mining (CRISP-DM) (Μεθοδολογία ανάπτυξης έργων Εξόρυξης Γνώσης CRISP-DM) Το μοντέλο CRISP-DM αναπτύχθηκε το 1996 από αναλυτές των εταιριών Integral Solutions Ltd (ISL), Teradata, Daimler AG, NCR Corporation και OHRA και παρουσιάστηκε το 1999. Το CRISP-DM αποτελεί μέχρι σήμερα το πιο δημοφιλές μοντέλο εξόρυξης γνώσης.

- Sample, Explore, Modify, Model, Assess (SEMMA), η οποία αναπτύχθηκε από το ινστιτούτο SAS των Ηνωμένων Πολιτιών Αμερικής

Το μοντέλο διαδικασίας εξόρυξης γνώσης *CRISP-DM* αποτελείται από έξι κυκλικά στάδια τα οποία περιγράφουν το κύκλο ζωής της διαδικασίας εξόρυξης γνώσης.

- *Business understanding* (Καθορισμός επαγγελματικών στόχων):
  στο στάδιο αυτό γίνεται εκτίμηση των προβλημάτων και προκλήσεων που έχει να αντιμετωπίσει η εταιρία, θέτονται οι στόχοι της εξόρυξης δεδομένων και αναπτύσσεται το πλάνο διαχείρισης έργου.

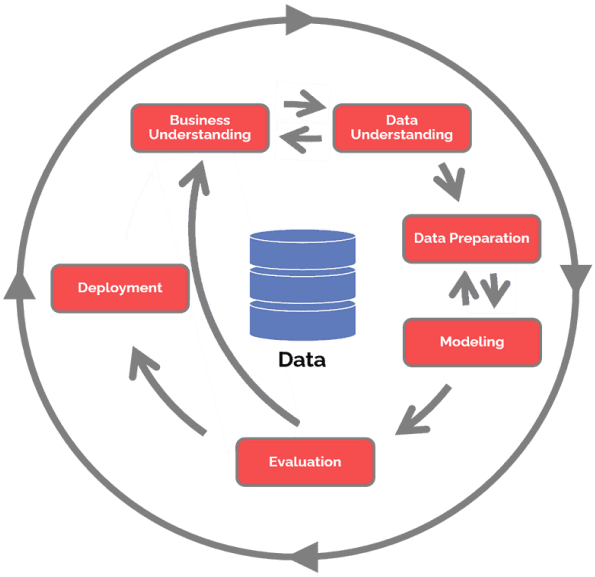
- *Data understanding* (κατανόηση δεδομένων): στο στάδιο αυτό γίνεται εκτίμηση των δεδομένων που θα χρειαστούν για την επίλυση του προβλήματος και περιλαμβάνει την συλλογή, τη διερεύνηση και την πιστοποίηση της ποιότητας των δεδομένων.

- *Data preparation* (προετοιμασία δεδομένων): Η φάση προετοιμασίας δεδομένων καλύπτει όλες τις δραστηριότητες που απαιτούνται για την κατασκευή του τελικού συνόλου δεδομένων. Σε αυτό το στάδιο γίνεται επεξεργασία των ακατέργαστων δεδομένων έτσι ώστε να είναι έτοιμα για τα προγράμματα εξόρυξης δεδομένων. Για παράδειγμα, μπορεί να γίνει αναγωγή αριθμητικών τιμών σε άλλες αριθμητικές τιμές ή μετατροπή αριθμητικών τιμών σε ονομαστικές τιμές. Επίσης, σε αυτό το στάδιο ανιχνεύονται πιθανά σφάλματα κωδικοποίησης, ελλιπή δεδομένα, κενές εγγραφές κλπ. Ακόμα ορισμένες μεταβλητές μπορεί να μετατρέπονται έτσι ώστε να μπορούν να χρησιμοποιηθούν στους αλγορίθμους. Αυτό το στάδιο είναι απαραίτητο για την αποφυγή σφαλμάτων.

- *Modeling (μοντελοποίηση)*: μόλις τα δεδομένα είναι έτοιμα, προχωράμε στο στάδιο της μοντελοποίησης το οποίο περιλαμβάνει τη κατασκευή ενός μοντέλου μέσω της εφαρμογής αλγορίθμων οι οποίοι οδηγούν στον εντοπισμό προτύπων στα δεδομένα. Αυτό το στάδιο πολλές φορές απαιτεί την επιστροφή στα προηγούμενα στάδια. Η μοντελοποίηση απαιτεί τη διαίρεση των δεδομένων σε δεδομένα εκπαίδευσης (training set) και ελέγχου (test set). Το μοντέλο εκπαιδεύεται σε πάνω από ένα αλγόριθμο συνήθως με σκοπό τον εντοπισμό του αποδοτικότερου.

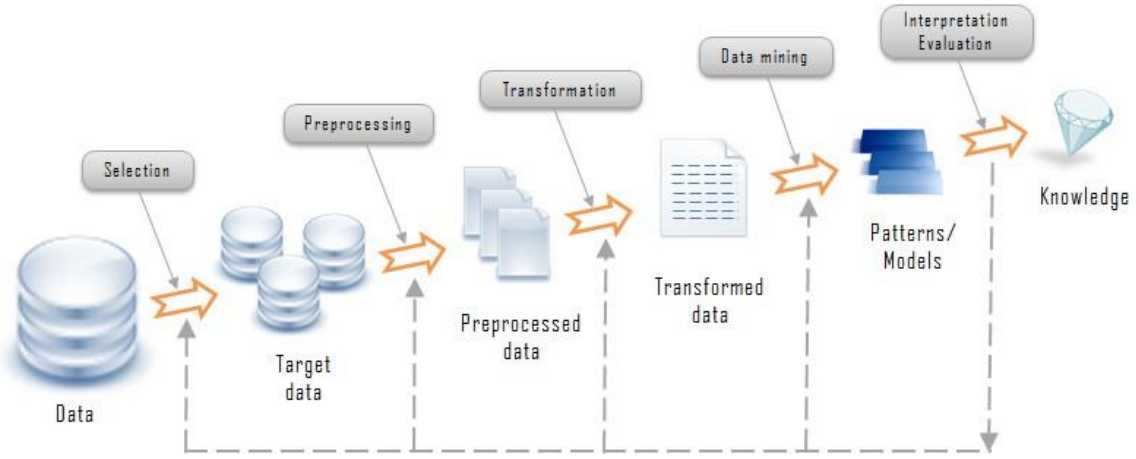
- *Evaluation (αξιολόγηση)*: το στάδιο αυτό περιλαμβάνει την αξιολόγηση των αποτελεσμάτων στο πλαίσιο που καθορίζεται από τους επιχειρηματικούς στόχους στο πρώτο βήμα. Αυτό οδηγεί στον εντοπισμό νέων αναγκών και με τη σειρά του οδηγεί στις προηγούμενες φάσεις στις περισσότερες περιπτώσεις.

- *Deployment (ανάπτυξη*): το στάδιο αυτό είναι το τελευταίο στάδιο ωστόσο η διαδικασία δεν τελειώνει εδώ αφού μπορεί να χρειαστεί επιστροφή σε κάποιο από τα προηγούμενα στάδια. Στο στάδιο αυτό φαίνονται ουσιαστικά τα αποτελέσματα όλων των παραπάνω βημάτων που ακολουθήθηκαν. Το στάδιο αυτό περιλαμβάνει αρχικά το σχεδιασμό της ανάπτυξης και χρήση των ανακαλύψεων, την αναφορά και την ανασκόπηση των τελικών αποτελεσμάτων. Στο σχήμα 12 φαίνεται το μοντέλο CRISP-DM.



Εικόνα 13: Μοντέλο εξόρυξης δεδομένων CRISP-DM

Το μοντέλο Sample, Explore, Modify, Model, Assess (SEMMA) έχει κάποιες διαφοροποιήσεις όσον αναφορά τα ονόματα των σταδίων, ωστόσο οι στόχοι και η διαδικασία παραμένει σχεδόν η ίδια. Τα στάδια αυτά είναι:



Εικόνα 14: Μοντέλο εξόρυξης δεδομένων SEMMA

- *Επιλογή των δεδομένων (data selection)*: το στάδιο αυτό περιλαμβάνει την αναγνώριση των διαθέσιμων πηγών και εξαγωγή των δεδομένων που θα χρειαστούν για την επίλυση του εκάστοτε προβλήματος. Σε αυτό το στάδιο γίνεται έλεγχος για κενές τιμές, τιμές μακριά από τις κανονικές (outliers) η τιμές που δεν φαίνονται αληθής.

- *Προεπεξεργασία (Preprocessing)*: στο στάδιο της προ-επεξεργασίας των δεδομένων γίνεται καθάρισμα των δεδομένων έτσι ώστε να επιβεβαιωθεί η ποιότητα των επιλεγμένων δεδομένων. Αυτό αποτελεί απαραίτητη προϋπόθεση για εξελιχθεί ομαλά η διαδικασία εξόρυξης. Το στάδιο αυτό περιλαμβάνει την επισκόπηση της δομής των

δεδομένων και τη μέτρηση της ποιότητας τους. Αυτό επιτυγχάνεται με στατιστικές μεθόδους καθώς και με οπτικοποίηση των δεδομένων. Η φάση της προεπεξεργασίας των δεδομένων αποτελεί αναμφίβολα την πιο χρονοβόρα φάση αφού για να ληφθούν σωστά αποτελέσματα τα δεδομένα πρέπει να είναι καθαρά και εύκολα αναγνώσιμα από το ανθρώπινο μάτι.

- *Μετασχηματισμός (Transformation)*: Το αποτέλεσμα αυτής της διαδικασίας οδηγεί σε ένα σύνολο δεδομένων το οποίο θα χρησιμοποιηθεί για την εξαγωγή προτύπων. Για παράδειγμα, μπορεί να γίνει αναγωγή αριθμητικών τιμών σε άλλες αριθμητικές τιμές ή μετατροπή αριθμητικών τιμών σε ονομαστικές τιμές. Αυτό το στάδιο είναι απαραίτητο για να δημιουργηθεί ένα σύνολο δεδομένων “καθαρό” ώστε να χρησιμοποιηθεί για την εξαγωγή προτύπων.

- *Εξόρυξη Δεδομένων (Data mining):* Στο στάδιο αυτό γίνεται η καθαυτό Εξόρυξη Δεδομένων, δηλαδή η εξαγωγή προτύπων. Έχοντας καθαρίσει και μετασχηματίσει τα δεδομένα, είναι έτοιμα να χρησιμοποιηθούν από κάποιον αλγόριθμο, ώστε να δημιουργηθεί κάποιο μοντέλο, συνήθως κατηγοριοποίησης ή πρόβλεψης. Η περιγραφική ανάλυση στοχεύει στην ομαδοποίηση των δεδομένων και στην εξαγωγή ιδιοτήτων. Η προγνωστική ανάλυση περιλαμβάνει την πραγματοποίηση προβλέψεων και τη κατασκευή κάποιου μοντέλου που να εξυπηρετεί την εκάστοτε πρόβλεψη. Υπάρχουν διάφορες τεχνικές εξόρυξης δεδομένων. Μερικές από αυτές είναι: η κατηγοριοποίηση, η ανάλυση συστάδων και η ανάλυση κανόνων συσχέτισης. Το ποια τεχνική θα χρησιμοποιηθεί εξαρτάται από τις ανάγκες του έργου αλλά και από την επιλογή του αναλυτή.

- *Ερμηνεία και αξιολόγηση (Interpretation and Evaluation)*: σε αυτό το στάδιο της εξόρυξης δεδομένων γίνεται η ερμηνεία και η αξιολόγηση των αποτελεσμάτων τα οποία παρήχθησαν από όλη τη διαδικασία.

2.5 Διαδικασία εξόρυξης κειμένου

Γενικά η διαδικασία της εξόρυξης κειμένου δεν διαφοροποιείται πολύ από την εξόρυξη δεδομένων. Ωστόσο σίγουρα υπάρχουν διαφοροποιήσεις οι οποίες αφορούν το τρόπου που τα δεδομένα επεξεργάζονται και καθαρίζονται αλλά και στους αλγορίθμους που χρησιμοποιούνται. Είναι φυσικά λογικό, όταν πρόκειται για ένα κείμενο να εμφανίζονται πολλές και διαφορετικές προκλήσεις που αφορούν τη ποιότητα και την αξιοπιστία του. Επίσης είναι βασικό ο αναλυτής να γνωρίζει τη γλώσσα την οποία αναλύει έτσι ώστε να μην υπάρξουν παρερμηνείες. Αξίζει επίσης να αναφερθεί ότι η διαφοροποίηση της καθημερινής γλώσσας που εκφράζονται οι άνθρωποι στα κοινωνικά δίκτυα και της γλώσσας που συναντάμε σε επιστημονικά βιβλία απέχει κατά πολύ. Η εξόρυξη κειμένου χωρίζεται στα παρακάτω στάδια:

Data collection (επιλογή δεδομένων): όπως είπαμε και στο προηγούμενο κεφάλαιο στο στάδιο αυτό επιλέγονται τα δεδομένα για την επίλυση του προβλήματος. Τα δεδομένα αυτά μπορεί να είναι HTML δεδομένα, αναρτήσεις η σχόλια σε κοινωνικά δίκτυα, κριτικές σε καταστήματα κλπ.

Text parsing (ανάλυση του κειμένου): το στάδιο αυτό αποτελεί το στάδιο προεπεξεργασίας των δεδομένων και αποτελεί ίσως το σημαντικότερο στάδιο της διαδικασίας εξόρυξης κειμένων. Αρχικά τα δεδομένα μετατρέπονται σε μια δομή στην οποία θα μπορεί να αναλυθεί στη συνέχεια. Σε αυτό το στάδιο γίνεται εξαγωγή του κειμένου και δημιουργία ενός λεξικού από τα δεδομένα με

τη χρήση της επεξεργασίας της φυσικής γλώσσας. Γίνεται λεξική ανάλυση και το κείμενο μετατρέπεται σε ακολουθία λέξεων. Ακόμα πρέπει να γίνει “καθάρισμα” των δεδομένων και να απαλλαχτεί από παύλες, σημεία στίξης και γενικά ό,τι χαλάει τη δομή του κειμένου. Ένας τρόπος εφαρμογής του συγκεκριμένου σταδίου είναι με το πακέτο NLTK της Python.

Αφαίρεση των τερματικών όρων (stopwords): Οι τερματικοί όροι αντιστοιχούν στις λέξεις τις οποίες δεν έχουν σημασιολογικό περιεχόμενο ως προς το θέμα του κειμένου. Οι λέξεις αυτές είναι απαραίτητες για τη σύνταξη της κάθε γλώσσας ωστόσο πρέπει να αφαιρεθούν έτσι ώστε να μην αλλοιωθεί το αποτέλεσμα της μελέτης. Αμέσως μετά γίνεται η μορφοσυντακτική ανάλυση (Part of Speech Tagging) στην οποία κάθε λέξη κατηγοριοποιείται στο μέρος του λόγου στο οποίο ανήκει.

Κανονικοποίηση (Stemming/Lemmatization): στο στάδιο αυτό τα δεδομένα κειμένου ετοιμάζονται έτσι ώστε να μπορούν στη συνέχεια να αναλυθούν. Η κανονικοποίηση δύναται να πραγματοποιηθεί επίσης σε δεδομένα ήχου. Η βιβλιοθήκη NLTK της Python προσφέρει μια γραφική διεπαφή για τη συγκεκριμένη διαδικασία. Η κανονικοποίηση διακρίνεται σε δυο τύπους:

- *Stemming (αποκατάληξη)*: πρόκειται για τη κανονικοποίηση των λέξεων με κάποιους κανόνες αποκατάληξης. Ουσιαστικά η λέξη γυρνάει στη “ρίζα” της ακόμα και αν δεν βγάζει κάποιο νόημα σημασιολογικά στο κείμενο. Πολλές λέξεις της ίδιας ομάδας μετατρέπονται στην ίδια κοινή λέξη-ρίζα. Για παράδειγμα:

  runs, running -> run colder colds
  -> cold
  beginning,begins→begin

- *Lemmatization (λημματοποίηση)*: Σε αντίθεση με την αποκατάληξη, λημματοποίηση μειώνει με σωστό τρόπο τις αρχικές λέξεις διασφαλίζοντας ότι η λέξη ανήκει στην συγκεκριμένη γλώσσα. Για παράδειγμα λημματοποίηση → λήμμα. Η ληματοποίηση πραγματοποιείται με τη βοήθεια μορφολογικού λεξικού. Στη διαδικασία αυτή γίνεται αναγωγή στο πρώτο κλιτικό τύπο της λέξης. Για παράδειγμα:

  am,was -> be has had
  -> have

Επιλογή αντιπροσωπευτικών όρων (Term frequency): σε αυτό το στάδιο γίνεται επιλογή των πιο αντιπροσωπευτικών όρων που υπάρχουν σε ένα κείμενο. Μια από τις πιο δημοφιλής τεχνικές για τον υπολογισμό των βαρών είναι η μέθοδος tf-idf (term frequency, inverse document frequency). Η μέθοδος αυτή αποτελεί μια στατιστική μέθοδο η οποία αξιολογεί πόσο αντιπροσωπευτική είναι μια λέξη σε ένα κείμενο η σε μια συλλογή κειμένων (corpus). Ακόμα χρησιμοποιείται για την εξαγωγή λέξεων-κλειδιών (keywords). Δημιουργείται ένα διάνυσμα κειμένων.

$$\begin{matrix} d1 & d2 \\ f_{1,1} & f_{1,2} \\ [f_{2,1} & f_{2,2}] \\ f_{3,1} & f_{3,2} \end{matrix}$$

Ο υπολογισμός των βαρών με χρήση του tf-idf γίνεται ως εξής:

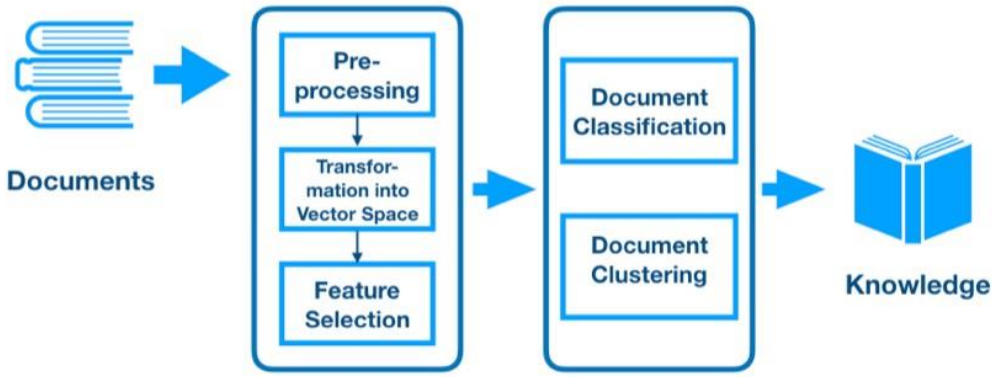
$$\boldsymbol{weight_i = \frac{tf_i \times idf_i}{\sqrt{\sum_{k \in vector} (tf_k \times idf_k)^2}}}$$

- Όπου tf (term frequency) αναφερόμαστε στη συχνότητα εμφάνισης ενός όρου σε ένα κείμενο.
- Όπου idf (inverse document frequency) αποτελεί την αντίστροφη συχνότητα στη συλλογή.

Ορίζουμε ως παρονομαστή το ευκλείδειο μήκος του διανύσματος για κάθε κείμενο. Όπου N ο συνολικός αριθμός κειμένων της συλλογής και n_i ο αριθμός των κειμένων της συλλογής στα οποία εμφανίζεται ο όρος i.

$$idf_i = \frac{N}{n_i}$$

Τα τελευταία στάδια του σχήματος περιλαμβάνουν την εφαρμογή αλγορίθμων για εκπαίδευση του μοντέλου και τη τελική γνώση των δεδομένων που παράχθηκαν.



Εικόνα 15: Διαδικασία εξόρυξης γνώσης από κείμενα

2.6 Κατηγοριοποίηση Μεθόδων Εξόρυξης δεδομένων

Η Εξόρυξη Δεδομένων αναζητά δομές δύο τύπων: πρότυπα και μοντέλα. Τα μοντέλα που παράγονται από το στάδιο της Εξόρυξης Δεδομένων διακρίνονται σε δυο βασικούς τύπους: τα μοντέλα πρόβλεψης (predictive) και τα περιγραφικά μοντέλα (descriptive). Οι εργασίες εξόρυξης δεδομένων μπορούν να ταξινομηθούν γενικά σε δύο τύπους με βάση το τι προσπαθεί να επιτύχει μια συγκεκριμένη εργασία. Τα περιγραφικά μοντέλα εξόρυξης δεδομένων χαρακτηρίζουν τις γενικές ιδιότητες των δεδομένων, ενώ οι εργασίες πρόβλεψης εξόρυξης δεδομένων πραγματοποιούν συμπεράσματα στο διαθέσιμο σύνολο δεδομένων για να προβλέψουν πώς θα συμπεριφερθεί ένα νέο σύνολο δεδομένων.

Επιπλέον, τα περιγραφικά μοντέλα έχουν ως στόχο τη περιγραφή όλου του συνόλου των δεδομένων ή της διαδικασίας παραγωγής των δεδομένων. Ουσιαστικά στοχεύουν στην εύρεση προτύπων (patterns) ή σχέσεις (relationships) που μπορεί να υπάρχουν στα δεδομένα. Ένας τρόπος που αυτό μπορεί να πραγματοποιηθεί είναι η σύνοψη (summarization). Η σύνοψη αποτελεί μέρος της διερευνητικής ανάλυσης δεδομένων (exploratory data analysis). Μια πολύ συμπαντική εφαρμογή των περιγραφικών μοντέλων είναι η συσταδοποίηση (clustering).

Για να καταλάβουμε καλύτερα τις δυο αυτές κατηγορίες μοντέλων, μπορούμε να αναφέρουμε κάποια παραδείγματα. Στην Ιατρική, ένας γιατρός που προσπαθεί να διαγνώσει μια ασθένεια με βάση τα αποτελέσματα των ιατρικών εξετάσεων ενός ασθενούς μπορεί να θεωρηθεί ως προγνωστική εργασία εξόρυξης δεδομένων. Οι περιγραφικές εργασίες εξόρυξης δεδομένων βρίσκουν συνήθως δεδομένα που περιγράφουν μοτίβα και έρχονται με νέες, σημαντικές πληροφορίες από το διαθέσιμο σύνολο δεδομένων. Ένας ιδιοκτήτης καταστήματος που προσπαθεί να αναγνωρίσει προϊόντα που αγοράζονται μαζί μπορεί να θεωρηθεί ως περιγραφική εργασία εξόρυξης δεδομένων.

Υπάρχουν γενικά πολλοί αλγόριθμοι εξόρυξης δεδομένων. Αυτοί οι αλγόριθμοι ουσιαστικά ανήκουν είτε στα περιγραφικά μοντέλα είτε στα μοντέλα πρόβλεψης. Ωστόσο, για να βγουν σωστά συμπεράσματα, ένα σύστημα εξόρυξης δεδομένων πιθανών να χρειαστεί να εκτελέσει ένα η παραπάνω αλγόριθμους εξόρυξης δεδομένων.

Τα μοντέλα πρόβλεψης που συνηθίζουμε να συναντάμε στην εξόρυξη δεδομένων είναι:

- **Κατηγοριοποίηση (Classification):**
  Η κατηγοριοποίηση αποτελεί μια προγνωστική μέθοδο της εξόρυξης δεδομένων. Στη συγκεκριμένη τεχνική δημιουργείται ένα μοντέλο-κατηγοριοποιητής (classifier) με βάση

τα υπάρχοντα δεδομένα. Η κατηγοποιοποίηση αποτελεί μια εργασία επιβλεπόμενης μάθησης η οποία έχει ως στόχο την ανακάλυψη της σχέσης ανάμεσα σε ένα γνώρισμα-στόχο με ονομαστικές τιμές και σε ένα σύνολο άλλων γνωρισμάτων. Ουσιαστικά, είναι η μάθηση μιας συνάρτησης, η οποία απεικονίζει ένα αντικείμενο (συνήθως αναπαρίσταται ως ένα διάνυσμα τιμών για τις χαρακτηριστικές του ιδιότητες) σε μία τιμή μιας κατηγορικής μεταβλητής, η οποία η οποία είναι γνωστή και ως κλάση (ή κατηγορία).

- **Παλινδρόμηση (Regression)**
  Με την ανάλυση παλινδρόμησης (regression analysis) εξετάζουμε τη σχέση μεταξύ δύο ή περισσοτέρων μεταβλητών με σκοπό την πρόβλεψη των τιμών της μιας, μέσω των τιμών της άλλης (ή των άλλων). Σύμφωνα με τη Wikipedia η παλινδρόμηση είναι μια τεχνική που χρησιμοποιείται για τη μοντελοποίηση και την ανάλυση αριθμητικών δεδομένων, μιας εξαρτημένης μεταβλητής και κάποιων ανεξάρτητων μεταβλητών. Το μοντέλο είναι μια συνάρτηση συσχέτισης της εξαρτημένης μεταβλητής από τις ανεξάρτητες. Με άλλα λόγια η ανάλυση παλινδρόμησης είναι μια αξιόπιστη μαθηματική μέθοδος προσδιορισμού των μεταβλητών που επηρεάζουν ένα θέμα ενδιαφέροντος. Η διαδικασία εκτέλεσης επιτρέπει να προσδιοριστεί με βεβαιότητα ποιοι παράγοντες έχουν μεγαλύτερη σημασία, ποιοι παράγοντες μπορούν να αγνοηθούν και πως αυτοί οι παράγοντες επηρεάζουν ο ένας τον άλλον. Προκειμένου να κατανοηθεί πλήρως η διαδικασία της ανάλυσης παλινδρόμησης είναι απαραίτητο να κατανοήσουμε τους ακόλουθους όρους:

  Εξαρτημένη μεταβλητή (dependent variable): αφορά τη μεταβλητή την οποία θέλουμε να κατανοήσουμε η να προβλέψουμε.
  Ανεξάρτητη μεταβλητή (independent variable): αφορά τη μεταβλητή η οποία υποθέτουμε ότι επηρεάζει την εξαρτημένη μεταβλητή μας.

Κάποια από τα περιγραφικά μοντέλα που συναντώνται είναι:

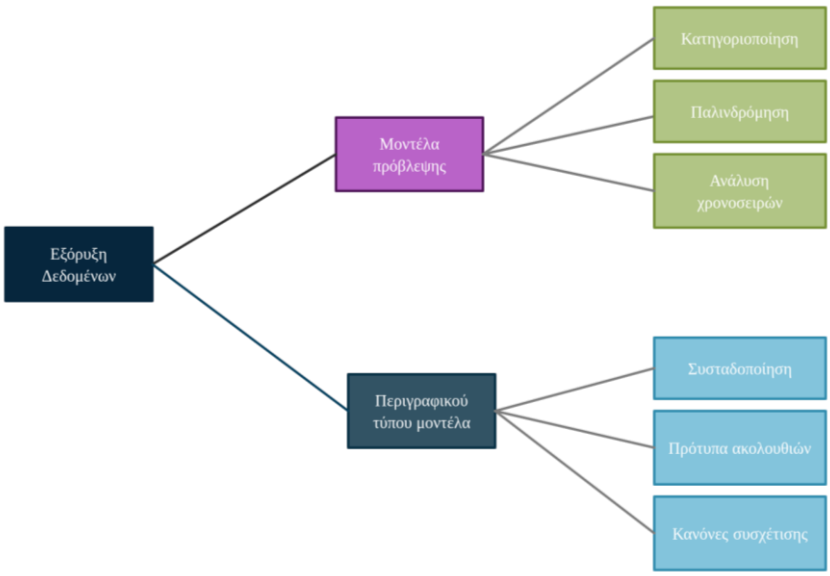
- Συσταδοποίηση (Clustering): Η συσταδοποίηση ανήκει στις περιγραφικές μεθόδους εξόρυξης δεδομένων. Στόχος της συσταδοποίησης είναι η διαίρεση του πληθυσμού η των σημείων των δεδομένων σε ομάδες έτσι ώστε τα σημεία που ανήκουν στην ίδια ομάδα να έχουν περισσότερα κοινά στοιχεία από αυτά που ανήκουν σε άλλη ομάδα. Στη συσταδοποίηση ο αριθμός των συστάδων επιλέγεται από τον ερευνητή. Οι ομοιότητες των στοιχείων στη συνέχεια επιλέγονται από τον αλγόριθμο. Επομένως ο αλγόριθμος είναι πιθανό να χρειαστεί να τρέξει πάνω από μια φορά για να καταλήξουμε στο σωστό συμπέρασμα. Οι αλγόριθμοι συσταδοποίησης είναι πολύ αποτελεσματικοί για μεγάλου όγκου δεδομένα. Η συσταδοποίηση είναι πολύ χρήσιμη σε πολλές εφαρμογές όπως: η λήψη αποφάσεων, η ανάκτηση πληροφορίας

- Η συσταδοποίηση διακρίνεται σε τρεις κατηγορίες:
  1. Διαμεριστική συσταδοποίηση
  2. Ιεραρχική συσταδοποίηση
  3. Συσταδοποίηση που βασίζεται στην πυκνότητα.

- *Κανόνες συσχέτισης (Association Rules):* Η εξαγωγή κανόνων συσχέτισης θεωρείται μια από τις σημαντικότερες διεργασίες δεδομένων. Οι κανόνες συσχέτισης αποτελούν ένα συνοπτικό τρόπο έκφρασης χρήσιμων πληροφοριών οι οποίες γίνονται εύκολα κατανοητές από τους χρήστες. Ουσιαστικά πρόκειται για κρυμμένες “συσχετίσεις” μεταξύ γνωρισμάτων ενός συνόλου. Γενικά, ένας κανόνας συσχέτισης έχει την μορφή $X \rightarrow Y$,

όπου $X \subset M$, $Y \subset M και X \cap Y = \emptyset$. Το X είναι ένα υποσύνολο του M, το Y είναι επίσης ένα υποσύνολο του M το οποίο δεν έχει κανένα κοινό στοιχείο με το X. Ο παραπάνω κανόνας μπορεί να μεταφραστεί πως « όποιος αγοράζει τα προϊόντα που ανήκουν στο X τότε αγοράζει και τα προϊόντα που ανήκουν στο Y. Ένας αλγόριθμος που βοηθά στην εύρεση κοινών συχνών στοιχειοσυνόλων είναι ο αλγόριθμος apriori. Ονομάζεται έτσι διότι χρησιμοποιεί προηγούμενη γνώση σχετικά με την συχνότητα κ-στοιχειοσυνόλων έτσι ώστε να βρει (κ+1) στοιχειοσύνολα. Ο συγκεκριμένος αλγόριθμος δέχεται ο είσοδο μια δομή από δεδομένα κα ορίζονται συνθήκες όπως η υποστήριξη δηλαδή το ποσοστό που περιέχουν το υποσύνολο και η εμπιστοσύνη.

- Πρότυπα ακολουθιών (Pattern discovery in sequences): Η ανακάλυψη προτύπων αναφέρεται στη διαδικασία στην οποία ανακαλύπτονται συσχετίσεις στα πρότυπα που ανακαλύπτονται στα δεδομένα. Αυτή η διαδικασία αξιολογεί κάποια κριτήρια όπως η συχνότητα εμφάνισης, η διάρκεια η οι τιμές σε ένα σύνολο ακολουθιών έτσι ώστε να βρεθούν ενδιαφέροντα κρυμμένα πρότυπα.



Εικόνα 16: Κατηγοριοποίηση μεθόδων εξόρυξης κειμένου

3. Επεξεργασία φυσικής γλώσσας

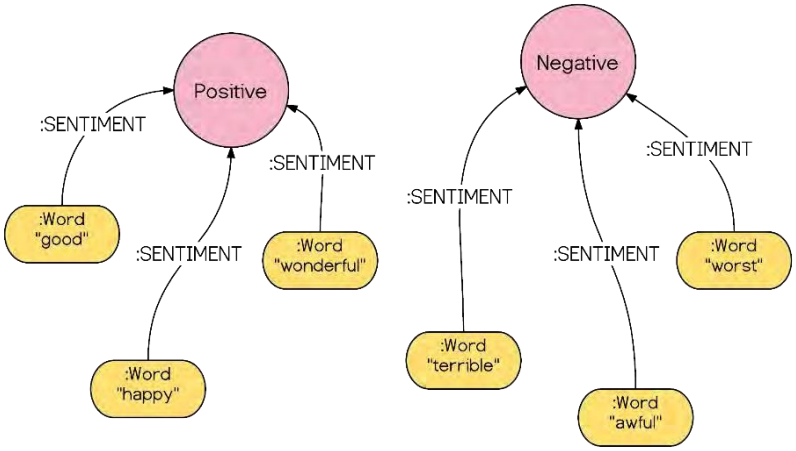
Η επεξεργασία φυσικής γλώσσας είναι ένα υποπεδίο της γλωσσολογίας, της επιστήμης των υπολογιστών και της τεχνητής νοημοσύνης που σχετίζεται με τις αλληλεπιδράσεις μεταξύ υπολογιστών και ανθρώπινης γλώσσας, ιδίως με τον τρόπο προγραμματισμού υπολογιστών για την επεξεργασία και ανάλυση μεγάλων ποσοτήτων φυσικών γλωσσικών δεδομένων. Θα μπορούσαμε να πούμε επίσης για να χαρακτηρίσουμε την επεξεργασία της φυσικής γλώσσας ως το σύνολο των εργαλείων που χρησιμοποιούνται ώστε να αντλούν σημαντικές και χρήσιμές πληροφορίες από πηγές της φυσικής γλώσσας, όπως ιστοσελίδες και έγγραφα κειμένου. Σημαντικές και χρήσιμες πληροφορίες θεωρούνται όλες οι πληροφορίες που μπορούν να έχουν κάποια εμπορική αξία, αν και πιο συχνά η επεξεργασία φυσικής γλώσσας χρησιμοποιείται για ακαδημαϊκά προβλήματα (Reese, 2015).

Οι εφαρμογές της φυσική γλώσσας δεν μετρούν πολλά χρόνια, λόγω των προκλήσεων που αυτές περιέχουν. Οι υπολογιστές απαιτούν από τους ανθρώπους να τους εκφράζουν έννοιες σε

μια γλώσσα προγραμματισμού που είναι ακριβείς, ξεκάθαρες και δομημένες ή μέσω ενός περιορισμένου αριθμού σαφώς εκφωνημένων φωνητικών εντολών. Ο ανθρώπινος λόγος, ωστόσο, δεν είναι πάντα ακριβής - είναι συχνά διφορούμενος και η γλωσσική δομή μπορεί να εξαρτάται από πολλές πολύπλοκες μεταβλητές, όπως αργκό, τοπικές διαλέκτους, κοινωνικό πλαίσιο και φωνητικές αποχρώσεις. Οι πρώτες προσεγγίσεις της Ανάλυσης Φυσικής Γλώσσας αποτελούσαν προσεγγίσεις περισσότερο βασισμένες σε κανόνες

όπου απλοποιημένοι αλγόριθμοι μηχανικής μάθησης λέγανε τι είδους λέξεις και φράσεις να αναζητηθούν σε ένα κείμενο και να δώσουν συγκεκριμένες απαντήσεις όταν εμφανίστηκαν οι φράσεις αυτές. Οι καινούργιες προσεγγίσεις της Ανάλυσης Φυσικής Γλώσσας, βασίζονται κυρίως στη μηχανική μάθηση η οποία χρησιμοποιεί μοντέλα προβλέψεις τα οποία εκπαιδεύονται πάνω σε ετικετοποιημένα δεδομένα.

Οι προσεγγίσεις που βασίζονται σε μοντέλα βαθιάς μάθησης (deep learning) απαιτούν τεράστιες ποσότητες ετικετοποιημένων δεδομένων προκειμένου να εκπαιδευτούν και να προσδιοριστούν συσχετίσεις. Αυτό αποτελεί πρόκληση για την Ανάλυση Φυσικής Γλώσσας. Παρά τις προκλήσεις, η μηχανική μάθηση είναι μια πιο ευέλικτη, διαισθητική προσέγγιση στην οποία οι αλγόριθμοι μαθαίνουν να προσδιορίζουν την πρόθεση των ομιλητών από πολλά παραδείγματα, σχεδόν όπως το πώς ένα παιδί μαθαίνει ανθρώπινη γλώσσα.



Εικόνα 17 : Ανάλυση συναισθήματος

3.1 Ανάλυση συναισθήματος

Η ανάλυση συναισθημάτων, γνωστή και ως εξόρυξη γνώμης, είναι από τις αρχές της δεκαετίας του 2000 ένας από τους πιο ενεργούς τομείς της έρευνας στην επεξεργασία φυσικής γλώσσας. (Cambria 2016). Η ανάλυση του συναισθήματος από τα κοινωνικά δίκτυα έχει γίνει ένα δημοφιλές ερευνητικό θέμα. Σύμφωνα με τους Pang και Lee (2008), τα πρώτα συστήματα αυτόματης ανάλυσης συναισθημάτων είχαν εφαρμοστεί ως επί το πλείστων σε παραδοσιακά κείμενα όπως διαδικτυακές κριτικές και άρθρα σε εφημερίδες . Με των όγκο των δεδομένων που δημιουργείται από τους χρήστες κοινωνικών δικτύων, παρατηρούμε όλο και περισσότερο ενδιαφέρον για την ανάλυση των συναισθημάτων που εκφράζονται μέσα από αυτά.

Ως ανάλυση συναισθήματος ορίζεται το επιστημονικό πεδίο της Επεξεργασίας της Φυσικής Γλώσσας (NLP) στο οποίο χαρακτηρίζουμε το συναισθηματικό τόνο που περιέχει ένα σύνολο λέξεων στο διαδίκτυο και το κατατάσσουμε σε μια κατηγορία συναισθήματος (θετικό, αρνητικό, ουδέτερο). Η ανάλυση συναισθήματος, η οποία είναι γνωστή και ως εξόρυξη γνώμης, είναι ένας από τους πιο ενεργούς κλάδους της επεξεργασίας της φυσικής γλώσσας από την αρχή του 21ου

αιώνα και έγινε ιδιαίτερα δημοφιλής χάρη στη ραγδαία αύξηση των ενεργών χρηστών στα κοινωνικά δίκτυα. (Cambria 2016)

Η ανάλυση συναισθήματος ουσιαστικά αποτελεί τη μορφή σημασιολογικής ανάλυσης των κειμένων. Στόχος της είναι να εξάγει απόψεις, συναισθήματα απέναντι σε διαφορετικά αντικείμενα ενδιαφέροντος. Τα κοινωνικά δίκτυα σήμερα αποτελούν πολύτιμες πηγές δεδομένων τα οποία μπορούν να οδηγήσουν σε σωστότερες αποφάσεις σε πολλές διαφορετικές επιστήμες. Ακόμα αποτελούν μια ψηφιοποιημένη μορφή της κοινωνίας μας και ίσως αληθινότερη καθώς οι άνθρωποι εκφράζονται πολύ πιο εύκολα και ξετυλίγουν πολύ περισσότερες πτυχές του εαυτού τους.

Υπάρχουν πολλές εταιρείες και οργανισμοί που επιθυμούν να γνωρίζουν τις απόψεις που ανταλλάσσονται για τις υπηρεσίες η προϊόντα τους στα μέσα κοινωνικής δικτύωσης. Το ενδιαφέρον είναι κυρίως εμπορικό. Στη μελέτη των Pang and Lee (2008), αναφέρει ότι οι σκέψεις των ανθρώπων ή απλά η γνώμη των άλλων υπήρξε πάντα σημαντικό πληροφοριακό στοιχείο για τους περισσότερους ανθρώπους ή εταιρείες, για της λήψη αποφάσεων. Πολύ πριν από τη χρήση των κοινωνικών δικτύων, οι άνθρωποι εξέφραζαν την επιθυμία τους να γνωρίζουν τα συναισθήματα των άλλων, των οικογενειών τους, των φίλων τους, προτού τελικά αποφασίσουν. Πριν τα κοινωνικά δίκτυα αυτό γινόταν με άλλους τρόπους όπως για παράδειγμα μέσω της ομιλίας η γραμμάτων. Ωστόσο, στα κοινωνικά δίκτυα βλέπουμε συμπεριφορές ανθρώπων που βγάζουν το πραγματικό τους εαυτό που δεν θα βγάζανε σε άλλες συνθήκες. Κατ' αυτό τον τρόπο Οι Pang και Lee (2008), υποστηρίζουν ότι η άποψη των χρηστών στα κοινωνικά δίκτυα είναι πιο αντικειμενική και έγκυρη.

Ακόμα, αξίζει να προσθέσουμε το ενδιαφέρον του πολιτικού κόσμου ως προς τις απόψεις των πολιτών στα κοινωνικά δίκτυα. Οι Pang και Lee (2008) επιπλέον αναφέρουν ότι ορισμένες εταιρείες η οργανισμοί αντί να χρησιμοποιούν βολιδοσκοπήσεις ή διαφημίσεις μεγάλου κόστους, έχουν εστιάσει σε αυτόν τον άμεσο και οικονομικό τρόπο γνώσης του ενδιαφέροντος ή του συναισθήματος που λαμβάνουν από την αίσθηση του συνόλου της κοινωνίας που είναι όλο και περισσότερο συνδεδεμένη στα κοινωνικά δίκτυα.

Ο στόχος της ανάλυσης συναισθημάτων λοιπόν είναι ο καθορισμός αυτόματων εργαλείων κατάλληλος να εξαγάγει υποκειμενικές πληροφορίες μέσα από κείμενο φυσικής γλώσσας τέτοιες όπως απόψεις και συναισθήματα, προκειμένου να δημιουργηθεί μια γνώση δομημένη και εκμεταλλεύσιμη από ένα σύστημα υποστήριξης αποφάσεων ή λήψης αποφάσεων. Μια από τις πολλές εφαρμογές των δεδομένων των κοινωνικών δικτύων είναι η ανάλυση συναισθήματος. Οι χρήστες των κοινωνικών δικτύων εκφράζουν τις απόψεις τους σε μεγάλη ποικιλία θεμάτων και αυτό κάνει την ανάλυση συναισθήματος από δημοσιεύσεις ένα πολύ αποτελεσματικό τρόπο μέτρησης της κοινής γνώμης αλλά και ένα τρόπο προώθησης προϊόντων.

Η ανάλυση συναισθήματος μπορεί να ωφελήσει πολλούς κλάδους όπως η πολιτική, η ψυχολογία, το μάρκετινγκ και η δημοσιογραφία. Οι Jansen et al. (2009) και Ghiassi et al. (2013) ανέλυσαν το πως οι χρήστες των κοινωνικών δικτύων εκφράζουν τα συναισθήματα τους σε διάφορες μάρκες δείχνοντας έτσι αποδοχή η δυσαρέσκεια απέναντι στα προϊόντα που προσφέρουν. Αυτό αποτελεί πολύ χρήσιμη τεχνική μάρκετινγκ αφού οι εταιρίες με τη βοήθεια της ανάλυσης συναισθήματος μπορούν να προτείνουν τα σωστά προϊόντα στο σωστό αγοραστικό κοινό.

Για τη κατασκευή όμως ενός συστήματος ανάλυσης συναισθήματος, απαιτούνται δυο στάδια: το πρώτο αποτελεί τον εντοπισμό και λήψη των δεδομένων που προέρχονται από τα μέσα κοινωνικής δικτύωσης, σύμφωνα με κριτήρια αναζήτησης όπως λέξεις-κλειδιά, ενώ το δεύτερο θα πρέπει να εφαρμόζει ανάλυση συναισθήματος, δηλαδή να είναι σε θέση να προσδιορίζει την πολικότητα κάθε κειμένου και να αξιολογεί τη γενική αίσθηση με σκοπό το συμπέρασμα του τι και πως σκέφτονται οι άνθρωποι. Στη βιβλιογραφία, ο R. Van't Ende (2013) χρησιμοποιεί αυτόν τον τύπο για να υπολογίσει το γενικό συναίσθημα:

$$\text{Αριθμός θετικών tweets} - \text{Αριθμός αρνητικών tweets} + \text{Αριθμός ουδέτερων tweets}$$

Συνολικος αριθμός tweets

Υπάρχουν δυο τεχνικές οι οποίες έχουν επικρατήσει για την ανάλυση συναισθήματος:

- ανάλυση συναισθήματος με Μηχανική μάθηση
- ανάλυση συναισθήματος με χρήση λεξικού

Στα πλαίσια της συγκεκριμένης εργασίας θα χρησιμοποιηθούν και οι δύο τρόποι.

3.1.1 Ανάλυση συναισθήματος με Μηχανική μάθηση

Η ανάλυση συναισθήματος βασισμένη σε λεξικό αν και εύκολη στη κατανόηση, υστερεί όμως σε πολλά πράγματα. Γι' αυτό το λόγο η ανάλυση συναισθήματος με μηχανική μάθηση παραμένει ένας πολύ αποτελεσματικός τρόπος πρόβλεψης συναισθήματος. Η ανάλυση συναισθήματος με Μηχανική μάθηση, αναφέρεται ουσιαστικά σε αυτοματοποιημένα συστήματα τα οποία μπορούν να προβλέψουν αποτελεσματικά το συναίσθημα μιας πρότασης. Τα συστήματα αυτά μπορούν στο μέλλον να χρησιμοποιηθούν για διαφορετικά αρχεία δεδομένων καθώς και να προστεθούν περισσότερα δεδομένα εκπαίδευσης έτσι ώστε να βελτιώνονται συνεχώς.

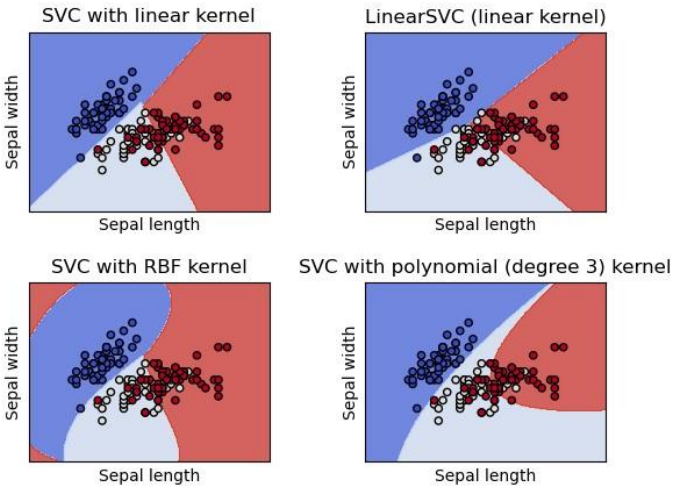
Τα μοντέλα αυτά προβλέπουν αποτελεσματικότερα από τα λεξικά συναισθήματος. Το μειονέκτημα τους όμως είναι ότι για να καταφέρουν να προβλέψουν το συναίσθημα μιας πρότασης θα πρέπει να τους δοθεί σαν input ένα αρχείο δεδομένων βαθμολογημένα συναισθηματικά. Προκειμένου να κατασκευαστεί ένα αποδοτικό μοντέλο, χρειάζεται και η ανάλογη ποσότητα δεδομένων. Το πρόβλημα της συναισθηματικής ανάλυσης κειμένου εντάσσεται στα προβλήματα κατηγοριοποίησης. Παρακάτω θα αναλυθούν κάποιοι από τους πιο σημαντικούς αλγόριθμους κατηγοριοποίησης κειμένου τους οποίους θα εφαρμόσουμε στη συνέχεια προκειμένου να κατασκευάσουμε το μοντέλο μηχανικής μάθησης για τη κατηγοριοποίηση των συναισθημάτων των δημοσιεύσεων.

Support Vector Machines (Μηχανές Διανυσμάτων Υποστήριξης)

Οι Μηχανές Διανυσμάτων Υποστήριξης (SupportVectorMachines,SVM) είναι μία οικογένεια αλγορίθμων επιβλεπόμενης μάθησης που αναπτύχθηκαν από τον Vladimir Vapni. Οι μηχανές διανυσματικής υποστήριξης (SVM) είναι ένα σύνολο μεθόδων που χρησιμοποιούνται για ταξινόμηση, παλινδρόμηση και έκτοπων παρατηρήσεων (outliers). Οι SVM εκτελούν κατηγοριοποίηση βρίσκοντας το υπερεπίπεδο που διαφοροποιεί τις κλάσεις που σχεδιάστηκαν στον n-διάστατο χώρο. Οι SVM σχεδιάζουν μια υπερπαράμετρο (hyperplane) μετασχηματίζοντας τα δεδομένα με τη βοήθεια μαθηματικών συναρτήσεων που ονομάζονται "πυρήνες" (kernels). Οι συναρτήσεις πυρήνα (kernel functions) είναι απεικονίσεις των διανυσμάτων εισόδου x στο σύνολο R, οι οποίες έχουν συγκεκριμένη μορφή και ιδιότητες και γενικεύουν σε μεγάλο βαθμό τις εφαρμογές των αλγορίθμων ταξινόμησης.

Τα πλεονεκτήματα της συγκεκριμένης ομάδας αλγορίθμων είναι: • Είναι αποτελεσματικοί σε χώρους υψηλής διάστασης (high dimentional spaces).

- Εξακολουθούν να είναι αποδοτικοί σε περιπτώσεις όπου ο αριθμός των διαστάσεων είναι μεγαλύτερος από τον αριθμό των δειγμάτων. • Είναι αποτελεσματικός ακόμα και σε περιπτώσεις όπου ο αριθμός των διαστάσεων είναι μεγαλύτερος από τον αριθμό των δειγμάτων
- Υπάρχει δυνατότητα επιλογής του κατάλληλου kernel προκειμένου να καταλήξουμε στο κατάλληλο αποτέλεσμα. Δημιουργούνται υπερεπίπεδα ανάλογα με το kernel ο οποίος χρησιμοποιείται.

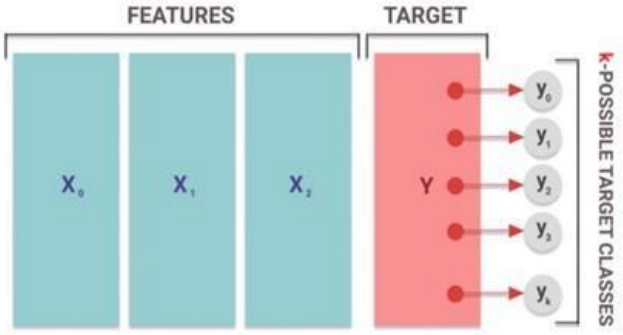


Εικόνα 18: Παράδειγμα Support Vector Machine (SVM)

Multinomial logistic regression (Πολυωνυμική Λογιστική Παλινδρόμηση)

Η Πολυωνυμική Λογιστική Παλινδρόμηση είναι η ανάλυση της παλινδρόμησης η οποία αποτελείται από εξαρτώμενες μεταβλητές με παραπάνω από δύο επίπεδα. Όπως και η πολλαπλή γραμμική παλινδρόμηση, η Πολυωνυμική Λογιστική Παλινδρόμηση αποτελεί ένα μοντέλο προγνωστικού τύπου. Η πολυωνυμική παλινδρόμηση χρησιμοποιείται για να εξηγήσει τη σχέση μεταξύ μιας ονομαστικής εξαρτημένης μεταβλητής και μίας ή περισσοτέρων ανεξάρτητων μεταβλητών. Η συνάρτηση πολυωνυμικής παλινδρόμησης είναι ένας αλγόριθμος στατιστικής ταξινόμησης. Αυτό σημαίνει ότι μόλις τροφοδοτήσουμε τη συνάρτηση με ένα σύνολο χαρακτηριστικών, το μοντέλο εκτελεί μια σειρά μαθηματικών πράξεων για να εξομαλύνει τις τιμές εισόδου σε ένα διάνυσμα τιμών που ακολουθεί μια κατανομή πιθανότητας.

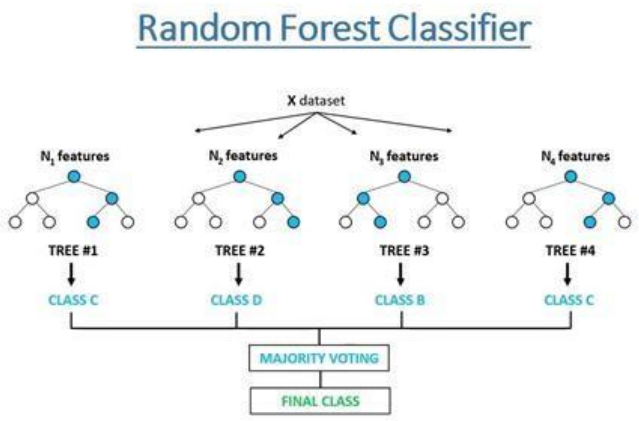
Στο μοντέλο δίνεται μια είσοδος διανυσμάτων X το οποίο αποτελείται από τα χαρακτηριστικά x_1, x_2, x_3..x_n. Σαν έξοδο λαμβάνουμε ένα διάνυσμα πιθανότητας Y που περιέχει πιθανότητες y_1, y_2, y_3 για τις κλάσεις στόχου k. Τέλος, το αποτέλεσμα με τη μεγαλύτερη πιθανότητα θα είναι το προβλεπόμενο αποτέλεσμα για το συγκεκριμένο σύνολο χαρακτηριστικών. Το αποτέλεσμα που προβλέπεται για ένα σύνολο χαρακτηριστικών είναι ένα από τα πιθανά αποτελέσματα k (σχήμα 18)



Εικόνα 19: Πολυωνυμική Λογιστική Παλινδρόμηση

Random Forest

Ο αλγόριθμος Random Forest, ο οποίος προτάθηκε επίσημα το 2001 από τους Leo Breiman και Adèle Cutler, είναι μέρος των τεχνικών αυτόματης μάθησης. Ο αλγόριθμος Random Forest εκπαιδεύεται σε πολλά δέντρα αποφάσεων που βασίζονται σε ελαφρώς διαφορετικά υποσύνολα δεδομένων. Αποτελεί έναν αλγόριθμο με πολύ καλή ακρίβεια σε προβλήματα κατηγοριοποίησης ο οποίος αποδίδει πολύ καλά σε πολύ μεγάλο αριθμό χαρακτηριστικών. Από το σύνολο δέντρων που εκπαιδεύονται επιλέγεται αυτό με τα καλύτερα χαρακτηριστικά (το πιο αποδοτικό).



Εικόνα 20: Αλγόριθμος Random Forest

Multinomial Naïve Bayes

Ο αλγόριθμος Naïve Bayes βασίζεται στο θεώρημα Bayes και αποτελεί έναν αλγόριθμο κατηγοριοποίησης. Ο αλγόριθμος Naïve Bayes είναι πολύ αποτελεσματικός στη ταξινόμηση κειμένου και για συστήματα εντοπισμού ανεπιθύμητων μηνυμάτων. Ένα από τα πλεονεκτήματα του αλγορίθμου είναι ότι είναι αποτελεσματικός ακόμα και με μικρή ποσότητα δεδομένων εκπαίδευσης. Η εκπαίδευση του είναι αρκετά γρήγορη σε σχέση με άλλες μεθόδους. Σύμφωνα με τον Segaran (2007). Ο συγκεκριμένος αλγόριθμος χρησιμοποιείται συχνά για την ταξινόμηση κειμένων επειδή απαιτεί πολύ λιγότερη υπολογιστική ισχύ από άλλες μεθόδους. Οι Pang και Lee (2008) συγκρίνουν την απόδοση τριών από τους πιο γνωστούς κατηγοριοποιητές: Naive Bayes, maximum Entropy και SVM (Support Vector Machine). Το αποτέλεσμα έδειξε ότι όταν το σύνολο δεδομένων (εκπαίδευση και δοκιμή) είναι μικρό, ο αφελής αλγόριθμος ταξινόμησης του Bayes αποδίδει καλύτερα. Σε αντίθεση με το SVM που είναι αποδοτικότερος για μεγαλύτερα σύνολα δεδομένων.

3.1.2 Τεχνικές αξιολόγησης κατηγοριοποιητών

Προκειμένου να σιγουρευτούμε ότι το μοντέλο το οποίο κατασκευάσαμε προβλέπει επαρκώς, θα πρέπει να ακολουθήσουμε κάποιες τεχνικές προκειμένου να δούμε κατά πόσο αυτό προβλέπει αποτελεσματικά.

Accuracy (ακρίβεια)

Αφού κατασκευαστεί ένα μοντέλο, θα θέλαμε να αξιολογήσουμε/εκτιμήσουμε την ποιότητα του/την ακρίβεια της κατηγοριοποίησης που αυτό πέτυχε. Η μέτρηση της ακρίβειας είναι μια από τις πιο σημαντικές μετρικές αξιολόγησης ενός μοντέλου κατηγοριοποίησης. Είναι η μέτρηση η οποία μας ενημερώνει σχετικά με το πόσες τιμές κατηγοριοποιήθηκαν στη σωστή κλάση και

πόσες όχι σε ένα σύνολο δεδομένων. Ωστόσο, δεν αποτελεί τη μοναδική μέτρηση που πρέπει να λαμβάνουμε υπόψιν όταν εκπαιδεύουμε ένα μοντέλο πρόβλεψης. Σε γενικές γραμμές η ακρίβεια ορίζεται ως:

$$\text{Ακρίβεια} = \frac{Αριθμός\ σωστών\ προβλέψεων}{Συνολικός\ αριθμός\ προβλέψεων}$$

Όταν πρόκειται για πρόβλημα δυαδικού τύπου, τότε η ακρίβεια δύναται να υπολογιστεί ως:

$$\text{Ακρίβεια} = \frac{TP+TN}{TP+TN+FP+FN}$$

Όπου: TP: ο αριθμός των σωστών θετικών προβλέψεων

FN: ο αριθμός των ψευδώς θετικών προβλέψεων

TN: ο αριθμός των σωστά αρνητικών προβλέψεων

FN: ο αριθμός των λανθασμένα αρνητικών προβλέψεων

Τύποι λαθών

Στα μοντέλα μηχανικής μάθησης συνηθίζουμε να συναντάμε δυο τύπους λαθών: *Σφάλμα τύπου Α:* είναι η εσφαλμένη απόρριψη μιας πραγματικής μηδενικής υπόθεσης ("ψευδές θετικό", δηλαδή, η απόρριψη μιας αληθινής υπόθεσης θεωρώντας την λάθος), ενώ

Σφάλμα τύπου Β: είναι η αδυναμία να απορριφθεί μια ψευδής μηδενική υπόθεση ("ψευδές αρνητικό", δηλαδή, η αποδοχή λανθασμένης υπόθεσης, θεωρώντας την σωστή) (Gambrill, 2006).

Recall (ανάκληση) – Precision (ακρίβεια)

Η ακρίβεια (Precision) προσπαθεί να απαντήσει στην ακόλουθη ερώτηση:

Τι ποσοστό των θετικών κατηγοριοποιήσεων είναι στη πραγματικότητα θετικό; Η ακρίβεια ορίζεται ως:

$$\text{Precision} = \frac{TP}{TP+FP}$$

Η ανάκληση δεν διαφέρει και πολύ από την ακρίβεια, όμως θέτει ένα διαφορετικό ερώτημα: Τι ποσοστό των πραγματικών θετικών εντοπίστηκε σωστά; Όσο πιο μεγάλη η ανάκληση, τόσο λιγότερα θετικά παραδείγματα έχουν κατηγοριοποιηθεί λάθος. Η ανάκληση ορίζεται ως εξής:

$$\text{Ανάκληση} = \frac{TP}{TP+FP}$$

Προκειμένου να αξιολογηθεί πλήρως την αποτελεσματικότητα ενός μοντέλου, πρέπει να εξεταστεί τόσο την ακρίβεια όσο και η ανάκληση. Δυστυχώς, η ακρίβεια και η ανάκληση βρίσκονται συχνά σε ένταση. Δηλαδή, η βελτίωση της ακρίβειας συνήθως μειώνει την ανάκληση και το αντίστροφο.

F1 score

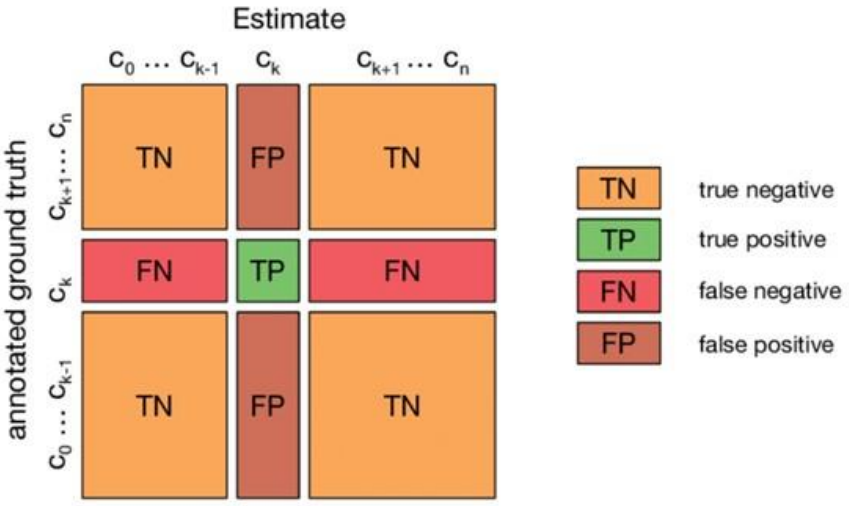
Η μέτρηση του f1 αποτελεί άλλη μια τεχνική αξιολόγησης του μοντέλου και αποτελεί ουσιαστικά τον αρμονικό μέσο της ακρίβειας (precision) και της ανάκλησης. Ορίζεται ως:

$$F1 = \frac{2rp}{r+p} = \frac{2TP}{2TP+FP+FN}$$

Confusion Matrix (πίνακας σύγχυσης)

Ο πίνακας σύγχυσης αποτελεί ένα πίνακα MxM (για δυαδικά προβλήματα) όπου τα στοιχεία (π.χ. i,j) ισούνται με το πλήθος των στοιχείων που ενώ προέρχονται από τη κλάση i, κατηγοριοποιούνται στη κλάση j. Αν έχουμε ένα πολυωνυμικό πρόβλημα τότε ο πίνακας μετατρέπεται σε MxMxM. Οι τέσσερεις διαφορετικές τιμές που μπορούν να προκύψουν σε ένα πίνακα σύγχυσης είναι:

- TP (True Positive)
- TN (True Negative)
- FP (False Positive)
- FN (False Negative)



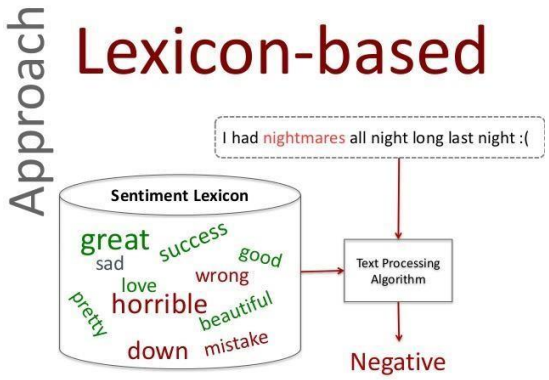
Εικόνα 21: Παράδειγμα πίνακα σύγχυσης (Confusion Matrix)

3.1.3 Ανάλυση συναισθήματος με χρήση λεξικού

Η λεξικολογική προσέγγιση είναι μία από τις δύο κύριες προσεγγίσεις στην ανάλυση συναισθημάτων και περιλαμβάνει τον υπολογισμό του συναισθήματος από τον σημασιολογικό προσανατολισμό της λέξης ή των φράσεων που εμφανίζονται σε ένα κείμενο. Η ανάλυση συναισθήματος με λεξικά (lexicon-based) αναφέρεται ουσιαστικά σε προκατασκευασμένα λεξικά συναισθήματος τα οποία περιέχουν βαθμούς “βαθμολόγησης” της εκάστοτε λέξης. Για παράδειγμα η λέξη “καλός” σε κάποιο λεξικό θα ήταν +2, η λέξη “καλούτσικος” +1 ενώ η λέξη “κακός” -3. Στη συνέχεια ο αλγόριθμος βρίσκει λέξεις από το λεξικό οι οποίες υπάρχουν στο κάθε κείμενο, βαθμολογώντας τες και ταξινομώντας τες με βάση το μέσο όρο της κάθε λέξης. Στη συνέχεια οι θετικές λέξεις αντιστοιχούνται με αυτές του κειμένου και οι αρνητικές με τις αρνητικές

του κειμένου. Η ανάλυση συναισθήματος με χρήση λεξικού συνήθως βασίζεται στον υπολογισμό της πολικότητας των λέξεων. Η πολικότητα των λέξεων ανακτάται από το εκάστοτε λεξικό με τη χρήση κάποιου αλγορίθμου και η πολικότητα της πρότασης υπολογίζεται χρησιμοποιώντας: Άθροισμα πολικότητας όλων των λέξεων σε μια πρόταση διαιρούμενο με τον συνολικό αριθμό λέξεων της πρότασης.

Η τελική κατηγοριοποίηση γίνεται μέσω μιας συνάρτησης η οποία κατατάσσει τις λέξεις σε συναισθήματα ανάλογα με το πόσες θετικές/αρνητικές λέξεις περιέχει κάθε πρόταση. Τέλος, γίνεται η τελική πρόβλεψη σχετικά με το σύνολο του συναισθήματος της εκάστοτε φράσης. Αυτή είναι η διαδικασία της λεξικολογικής προσέγγισης ανάλυσης συναισθήματος. Τα λεξικά συναισθήματος κατασκευάζονται είτε χειροκίνητα είτε αυτόματα. Στις λεξικολογικές προσεγγίσεις της ανάλυσης συναισθήματος οι προτάσεις αναπαρίστανται σαν μια “bag of words”, δηλαδή ένας “σάκος” με λέξεις ταξινομημένες σε τυχαία σειρά. Ένα πλεονέκτημα των λεξικολογικών προσεγγίσεων αποτελεί το γεγονός ότι δεν χρειάζονται σύνολο εκπαίδευσης ούτε βαθμολογημένες δημοσιεύσεις για να φτιαχτεί κάποιο μοντέλο πρόβλεψης. Οι λεξικολογικές προσεγγίσεις αν και είναι πολύ πιο απλές στη κατανόηση, προσφέρουν λιγότερο έγκυρα αποτελέσματα από τις μεθόδους μηχανικής μάθησης. Ενδιαφέρον παρουσιάζουν οι υβριδικές μέθοδοι στις οποίες τα κείμενα βαθμολογούνται με κάποιο αλγόριθμο βασισμένο σε λεξικό και στη συνέχεια με βάση τα αποτελέσματα φτιάχνεται ένα μοντέλο πρόβλεψης μηχανικής μάθησης.



Εικόνα 22: Προσέγγιση βασισμένη σε λεξικό συναισθήματος.

3.2 Προκλήσεις

Όπως όλα τα επιστημονικά πεδία, έτσι και η ανάλυση συναισθήματος έχει να αντιμετωπίσει πολλές προκλήσεις ως προς την υλοποίηση ενός συστήματος αλλά και ως προς την επεξεργασία των δεδομένων που χρησιμοποιούνται στο σύστημα αυτό. Στα κοινωνικά δίκτυα, οι φράσεις που χρησιμοποιούντα είναι συνήθως σύντομες. Το Twitter, αποτελεί μια πολύ καλή πηγή δεδομένων και αυτό έγκειται στο γεγονός ότι το μεγαλύτερο μέρος των δεδομένων τους είναι προσβάσιμο αφού όπως προαναφέραμε οι σχέσεις δεν απαιτούν αμοιβαία αποδοχή. Μέσω του Streaming API μπορούμε να αποκτήσουμε μεγάλη ποσότητα δεδομένων σε λίγο χρόνο.

Τα tweets περιέχουν χρονοσφραγίδα, γεγονός που επιτρέπει το φιλτράρισμα και την εξόρυξη tweets στη χρονική στιγμή που επιθυμούμε. Τέλος, μέσω των ετικετών μπορούμε να φιλτράρουμε και να εξάγουμε πολύτιμες πληροφορίες για οποιοδήποτε θέμα και σε οποιαδήποτε γλώσσα επιθυμούμε. Για το Twitter έχουν γίνει πληθώρα αναλύσεων που βασίζονται σε tweets και αυτό γιατί θεωρείται το πιο προσβάσιμο και εύχρηστο κοινωνικό δίκτυο από πλευράς δεδομένων κειμένου. Ωστόσο υπάρχουν και πολλές προκλήσεις οι οποίες θα πρέπει να ληφθούν υπόψη:

- *Ορθογραφικά/συντακτικά λάθη*: δεδομένου ότι οι δημοσιεύσεις γίνονται σε ένα πολύ μικρό χρονικό διάστημα, είναι αναπόφευκτο να γίνονται ορθογραφικά λάθη. Ειδικά στην ελληνική γλώσσα παρατηρείται ένα μεγάλο ποσοστό ορθογραφικών λαθών αφού η ελληνική αποτελεί μια δύσκολη ως προς την ορθογραφία της γλώσσα. Αυτό δημιουργεί πληθώρα προβλημάτων ως προς την ανάλυση του κειμένου αφού δημιουργείται θόρυβος και αλλοιώνεται η πολικότητα του κειμένου. Αυτό οδηγεί σε λάθη ταξινόμησης ως προς το συναίσθημα των συγκεκριμένων δημοσιεύσεων.

- *Νεολογισμοί/χρήση αργκό:* η χρήση νεολογισμών και της καθημερινής γλώσσας είναι ένα πολύ συνηθισμένο φαινόμενο στα κοινωνικά δίκτυα. Επίσης η υβρεολόγια είναι ακόμα ένα φαινόμενο που συναντάται συχνά δημιουργώντας ποικίλα προβλήματα όσον αναφορά την ανάλυση συναισθήματος.

- *Περιορισμένος χώρος δημοσίευσης*: το Twitter έχει αυξήσει πλέον το όριο των χαρακτήρων στους 280 όμως η ελληνική είναι μια γλώσσα με μεγάλες λέξεις και σύνθετο λεξιλόγιο. Ως αποτέλεσμα οι χρήστες χρησιμοποιούν πολλές φορές τις λέξεις κομμένες η με συντομογραφίες.
- *Πολυγλωσσία:* το περιεχόμενο στα tweets αποτελείται από greeklish η ξενόφωνες λέξεις, κάτι που δημιουργεί προκλήσεις στην συναισθηματική ανάλυση των λέξεων. Αυτό είναι ένα πρόβλημα που αφορά όλες τις γλώσσες και όχι μόνο τα ελληνικά.

4. Υλοποίηση συστήματος ανάλυσης συναισθήματος

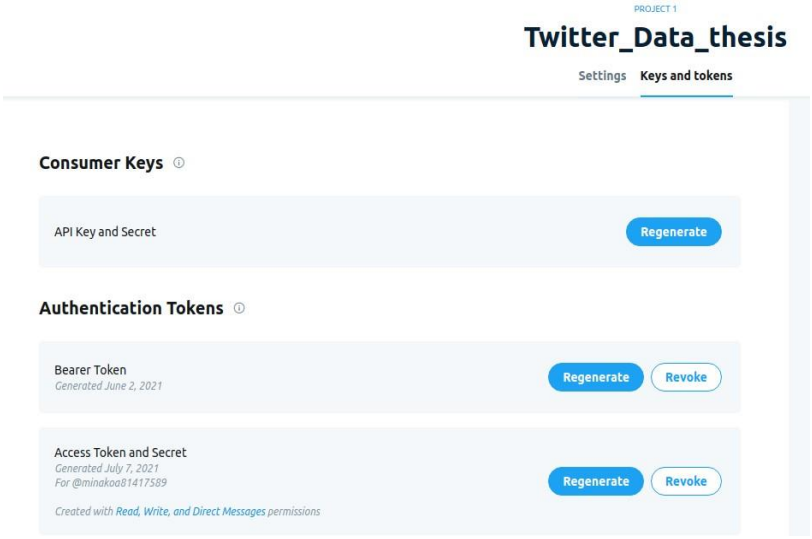
Η προσέγγιση που ακολουθήθηκε για την ανάλυση συναισθήματος στη συγκεκριμένη εργασία αποτελεί μια υβριδική μέθοδο ανάλυσης συναισθήματος αφού συνδυάζει τη προσέγγιση βασισμένη σε λεξικό και στη συνέχεια χρησιμοποιεί το αποτέλεσμα των προβλέψεων σαν input του μοντέλου πρόβλεψης που κατασκευάστηκε με αλγορίθμους μηχανικής μάθησης. Οι περισσότερες προσεγγίσεις που αφορούν την ανάλυση συναισθήματος βασίζονται μόνο σε μοντέλα μηχανικής μάθησης. Δυστυχώς αυτή η προσέγγιση απαιτεί βαθμολογημένα δεδομένα, κάτι που δεν μπορεί κάποιος να βρει εύκολα στην Ελληνική γλώσσα.

Το σύστημα που υλοποιήθηκε στη παρούσα εργασία έχει τη δομή που παρουσιάστηκε σε προηγούμενο κεφάλαιο και ακολουθεί τη διαδικασία εξόρυξης κειμένου που περιγράψαμε. Αρχικά πραγματοποιήθηκε η συλλογή και αποθήκευση των δεδομένων σε αρχείο csv, στη συνέχεια πραγματοποιήθηκε η διαδικασία της διερευνητικής ανάλυσης και αμέσως μετά η προεπεξεργασία τους και μετά ακολούθησε η συναισθηματική ανάλυσή τους. Στο τέλος πραγματοποιήθηκε η συναισθηματική ανάλυση τους μέσω των μεθόδων βασισμένων σε κανόνες Vader, Textblob. Από τις δύο μεθόδους επιλέχτηκες η Vader λόγω της ακριβέστερης πρόβλεψης της σε δεδομένα κοινωνικών δικτύων.

4.1 Συλλογή δεδομένων

Το σύστημα που παρουσιάζεται στη παρούσα εργασία αποτελείται από πέντε στάδια. Το πρώτο στάδιο του συστήματος περιλαμβάνει την διαδικασία της εξόρυξης δεδομένων από το κοινωνικό δίκτυο Twitter μέσω του Streaming API και την αποθήκευση τους σε αρχείο csv. Το πρώτο και σημαντικότερο στάδιο υλοποίησης του συστήματος περιλαμβάνει τη δημιουργία ενός Twitter Developer account και στη συνέχεια δημιουργία της εφαρμογής για την επικοινωνία με το API. Πλέον, μετά από αναβάθμιση στους όρους χρήσης της υπηρεσίας Twitter Developer Portal, η δημιουργία μιας εφαρμογής απαιτεί την αναλυτική επεξήγησή της στους διαχειριστές της

υπηρεσίας. Επιπλέον απαιτεί και την απαραίτητη έγκριση της. Τα επόμενα βήματα μετά τη δημιουργία της εφαρμογής απαιτούν τη λήψη των απαραίτητων credentials για τη σύνδεση με το Twitter API: “Access Token” “Access Token Secret” “Consumer Key” “Consumer Secret”.



Εικόνα 23: Twitter Developer Account

Το Streaming API του Twitter προσφέρει ένα εύκολο τρόπο να εξάγεις μια μεγάλη ποσότητα δεδομένων σε πολύ μικρό χρονικό διάστημα. Απαραίτητη προϋπόθεση είναι η συνεχής διατήρησης της σύνδεσης. Επιπλέον, όπως αναφέραμε και σε προηγούμενα κεφάλαια το Streaming API προσφέρει τη δυνατότητα φιλτραρίσματος των tweets που συλλέγονται σύμφωνα με τα hashtags η τη γλώσσα την οποία θέλουμε. Τα δεδομένα που συλλέχτηκαν αφορούν tweets τα οποία αφορούν το εμβόλιο του κορονοϊού προκειμένου να μελετηθούν και να ανακαλυφθεί μέσω της ανάλυσης συναισθήματος το αν οι Έλληνες νιώθουν θετικά η όχι απέναντι στο εμβόλιο του κορονοϊού.

Η πανδημία του κορονοϊού και συγκεκριμένα το εμβόλιο αποτελεί ένα θέμα της εποχής μας. Το πλούσιο περιεχόμενο των tweets μας δίνει τη δυνατότητα να γίνει μια εκτενής και αρκετά ρεαλιστική ανάλυση. Ακόμα, τα tweets που συλλέγονται μέσω του Streaming API συλλέγονται σε πραγματικό χρόνο, άρα τα συμπεράσματα που θα προκύψουν αφορούν τη παρούσα κατάσταση. Σε μελλοντικές μελέτες θα ήταν ενδιαφέρον να συμπεριληφθούν και παλαιότερα tweets συγκρίνοντας τα αποτελέσματα. Επιπλέον θα μελετηθούν τα δεδομένα προκειμένου να εξαχθούν κάποια στατιστικά στοιχεία σχετικά με την ελληνική κοινότητα του Twitter. Η γλώσσα που χρησιμοποιήθηκε ήταν εξ ολοκλήρου η Python λόγω της πληθώρας βιβλιοθηκών μηχανικής μάθησης και ανάλυσης και διαχείρισης δεδομένων που διαθέτει. Παρακάτω βλέπουμε ένα μικρό δείγμα των δεδομένων προτού καθαριστούν και επεξεργαστούν.

```
In [25]: 1 df.head()
```

Out[25]:

	user_name	created_at	followers_count	source	region	text
0	gna	27. July 2021	269.0	Twitter for iPhone	New York, NY	@Marka149133376 Είχαμε το μυστήριο του ΠόθενΈσ...
1	Κωστόπουλος Δημήτριος	27. July 2021	2.0	Twitter Web App	NaN	Ακούστε την παρέμβαση του τέως Διοικητή του No...
2	HERMES NEWS	27. July 2021	1756.0	Twitter for Android	Ελλάς	#COVID19greece #COVID19 #Covid_19 #covid19gr #...
3	Pfizer/mRNA Test Subject ID: 5e258d68-f600-40a1	27. July 2021	64.0	Twitter Web App	Zul Gurub	Τελικά ο #Αδωνις το έκανε το εμβόλιο; Η έκανε ...
4	Political	27. July 2021	294.0	Twitter Web App	NaN	Διαβάστε αύριο στην εφημερίδα POLITICAL 📲\nhtt...

Αρχικά προτού αρχίσουμε να γράφουμε οποιοδήποτε πρόγραμμα σε Python, είναι απαραίτητο να φορτώσουμε τις απαραίτητες βιβλιοθήκες που θα μας βοηθήσουν σε αυτό που θέλουμε να πραγματοποιήσουμε. Για το σύστημα εξόρυξης δεδομένων οι βιβλιοθήκες που χρησιμοποιήθηκαν είναι οι: tweepy, datetime, pandas, csv. Για την υλοποίηση του συστήματος εξόρυξης δεδομένων, χρησιμοποιήθηκε η βιβλιοθήκη tweepy της Python η οποία βοηθάει στο να πραγματοποιηθεί η επικοινωνία με το API του Twitter παρέχοντας μεθόδους διαχείρισης της σύνδεσης και της εξόρυξης tweets σε πραγματικό χρόνο. Με την εντολή OAuthHandler() του πακέτου OauthHandler της tweepy, πραγματοποιούμε τη σύνδεση με το πρωτόκολλο επικοινωνίας 0Auth.

Το εργαλείο που χρησιμοποιήθηκε για το σύστημα είναι το jupyter notebook το οποίο αποτελεί ένα πολύ χρήσιμο εργαλείο για οπτικοποίηση αλλά και για στατιστική ανάλυση δεδομένων μέσω της διαδραστικής διεπαφής χρήστη που προσφέρει. Η βιβλιοθήκη csv προσφέρει συναρτήσεις για τη διαχείριση αρχείων csv. Η βιβλιοθήκη Pandas προσφέρει πληθώρα συναρτήσεων για τη διαχείριση των δεδομένων που συλλέχτηκαν.

Η σύνδεση με το Streaming API πραγματοποιείται με την εντολή tweepy.Stream() την οποία αποθηκεύουμε στη μεταβλητή stream και το φιλτράρισμα των λέξεων κλειδιών πραγματοποιείται με την παράμετρο stream.filter(track=[‘λεξεις_κλειδιά’). Στη περίπτωσή μας οι λέξεις κλειδιά που χρησιμοποιήθηκαν είναι: ‘εμβολιο', 'εμβόλιο', 'κορονοϊός', 'κορονοϊός', 'κορoναϊός', 'εμβολιασμός', 'αντιεμβολιαστες', 'αντιεμβολιαστές'.

Μέσω του Streaming API υπάρχει δυνατότητα να γίνει φιλτράρισμα της γλώσσας στης οποία θέλουμε τα tweets. Ωστόσο στην ελληνική γλώσσα κάτι τέτοιο είναι περιττό καθώς με τις λέξεις κλειδιά που γράψαμε εμφανίστηκαν μόνο tweets στην ελληνική γλώσσα. Η επικοινωνία με το Streaming API μέσω της βιβλιοθήκης γίνεται με τη κλάση StreamListener(). Η μέθοδος on_data του StreamListener του Tweepy μεταβιβάζει εύκολα δεδομένα από δημοσιεύσεις στη μέθοδο on_status. Το αρχείο csv που δημιουργήθηκε περιέχει 6 στήλες: το id των χρηστών, το username, την ημερομηνία δημιουργίας του tweet, τον αριθμό ακολούθων του κάθε χρήστη, τη γεωγραφική περιοχή όπου αυτή υπάρχει και το κείμενο του tweet. Στο κώδικα που γράφτηκε θέσαμε ως όρο της αναζήτησης την αντικατάσταση των ειδικών χαρακτήρων με κενό με την εντολή replace() της βιβλιοθήκης csv επομένως τα tweets που συλλέχτηκαν είναι ευανάγνωστα.

Όταν χρησιμοποιούμε το Streaming API του Twitter, θα πρέπει η σύνδεση να παραμένει μονίμως ανοιχτή έτσι ώστε να έχουμε συνεχή ροή από Tweets. Εάν ο διακομιστής υπερβεί ένα περιορισμένο αριθμό προσπαθειών σύνδεσης στο API τότε θα ληφθεί σφάλμα και η σύνδεση θα διακοπεί. Θα πρέπει επίσης να αναφέρουμε ότι η σύνδεση με το API χάθηκε αρκετές φορές κατά τη διάρκεια συλλογής των tweets επομένως δημιουργήθηκαν πάνω από ένα csv αρχεία τα οποία ενώσαμε με τη μέθοδο DataFrame.merge() της βιβλιοθήκης pandas. Επομένως δημιουργήθηκε ένα dataset με το σύνολο των tweets τα οποία μαζεύτηκαν. Τέλος, αφού συλλέχτηκαν και αποθηκεύτηκαν τα tweets στη βάση δεδομένων μας, το επόμενο βήμα είναι η η διερευνητική ανάλυση έτσι ώστε να “γνωρίσουμε” τα δεδομένα. Στη συνέχεια θα γίνει η μορφοποίηση και το “καθάρισμά” τους έτσι ώστε να είναι έτοιμα για την ανάλυση.

4.2 Διερευνητική ανάλυση δεδομένων

Το πρώτο στάδιο της ανάλυσης των δεδομένων μας αποτελεί η κατανόηση του προβλήματος και των δεδομένων που έχουμε στη διάθεσή μας. Αμέσως μετά ακολουθεί η διερευνητική ανάλυση των δεδομένων έτσι ώστε να ανακαλύψουμε τι περιέχουν τα δεδομένα μας. Αυτό θα πραγματοποιηθεί με συναρτήσεις μέσω της γλώσσας Python και της βιβλιοθήκης pandas η οποία περιέχει συναρτήσεις για τη διευκόλυνση αυτής της διαδικασίας. Με την εντολή isnull() ελέγχουμε αν υπάρχουν κενές εγγραφές στα δεδομένα μας. Με την εντολή df.isna().sum() καταγράφεται ο αριθμός κενών εγγραφών που υπάρχουν στα δεδομένα και θα πρέπει να αποφασίσουμε τι θα κάνουμε με αυτά.

Στη περίπτωσή μας βρέθηκε μόνο μια κενή εγγραφή στο κείμενο των tweets επομένως θα σβήσουμε απλά αυτή την εγγραφή εφόσον δεν μπορούμε να το αντικαταστήσουμε με κάτι άλλο. Επίσης βρέθηκαν αρκετές κενές εγγραφές στη στήλη “Location” η οποία δηλώνει τη γεωγραφική περιοχή του χρήστη. Ουσιαστικά οι κενές εγγραφές αποτελούν τους χρήστες οι οποίοι επέλεξαν να μη δημοσιεύσουν τη γεωγραφική περιοχή τους. Αυτό δεν επηρεάζει καθόλου τη μελέτη μας επομένως θα κρατήσουμε τις εγγραφές αυτές όπως είναι και στη συνέχεια θα τις αντικαταστήσουμε. Το αρχείο δεδομένων που δημιουργήθηκε περιέχει 7 στήλες:

- *twitter_id*: αφορά το id του εκάστοτε χρήστη
- *Username*: αποτελεί το όνομα χρήστη που ανήκει το εκάστοτε tweet
- *Date*: είναι η στήλη στην οποία αναγράφεται η ημερομηνία στην οποία δημοσιεύτηκε το κάθε tweet
- *Followers:* αφορά τη καταμέτρηση των ακολούθων του κάθε χρήστη
- *Source:* αποτελεί τη συσκευή από την οποία δημοσιεύτηκε το εκάστοτε tweet.
- *Location:* αποτελεί τη γεωγραφική περιοχή την οποία έχει δηλώσει ο χρήστης σαν τόπο διαμονής του. Αυτή η στήλη δεν είναι απαραίτητα αντιπροσωπευτική καθώς αποτελεί μόνο αυτό το οποίο επιλέγει ο χρήστης να δημοσιεύσει. Στη στήλη αυτή υπήρξαν πολλές κενές εγγραφές οι οποίες προέρχονται από τους χρήστες οι οποίοι επιλέγουν να μη δημοσιεύσουν τη γεωγραφική τους περιοχή.
- *Tweet*: αποτελεί το κείμενο του κάθε tweet το οποίο και θα μας απασχολήσει για τη διαδικασία ανάλυσης συναισθήματος

Σε αυτό το στάδιο εξερευνούμε τα δεδομένα μας έτσι ώστε να τα "γνωρίσουμε καλύτερα". Τα ερωτήματα που θα απαντηθούν από τα δεδομένα σε αυτό το στάδιο:

- Ποιες είναι οι πιο συνηθισμένες συσκευές που επιλέγουν οι χρήστες;
- Ποιοι είναι οι χρήστες με τους πιο πολλούς ακολούθους;
- Σε ποιους χρήστες ανήκουν τα περισσότερα tweets;
- Σε ποια περιοχή διαμένουν οι περισσότεροι χρήστες;
- Πως διακυμαίνονται τα tweets σύμφωνα με την ημερομηνία; Ποια είναι η μέρα με τις πιο πολλές δημοσιεύσεις;
- Τι ποσοστό των tweets περιέχουν link, hashtag, mention;

4.3 Οπτικοποίηση δεδομένων

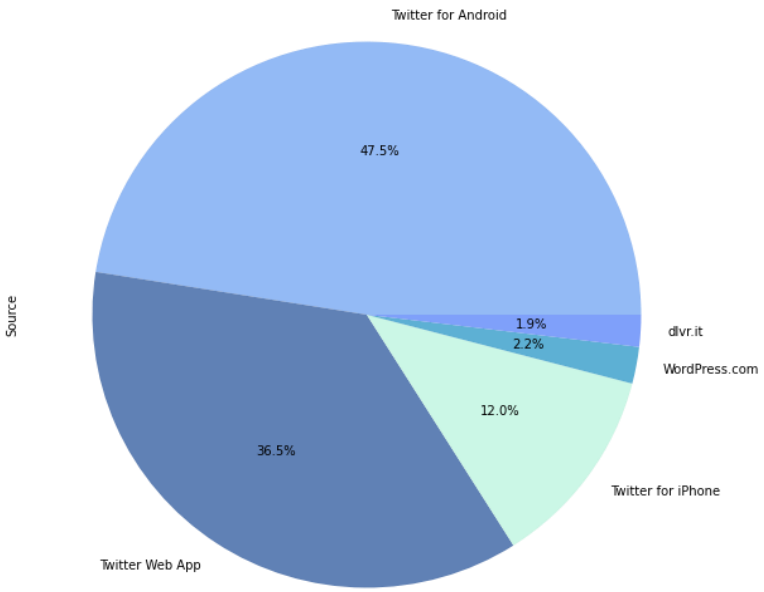
Οι τεχνικές οπτικοποίησης διαθέτουν μια σειρά από πλεονεκτήματα, τα οποία τις καθιστούν χρήσιμο εργαλείο για τον εντοπισμό και αναγνώριση δομών και ιδιοτήτων σε ένα σύνολο δεδομένων. (Κύρκος, Ε. 2015) Επιχειρηματική ευφυΐα και εξόρυξη δεδομένων. Τα γραφικά είναι ένα μέσο για τον εντοπισμό και αναγνώριση δομών και ιδιοτήτων σε ένα σύνολο δεδομένων (Card, Mackinlay & Shneiderman, 1999; Rober 2000). Με την οπτικοποίηση και τη διερευνητική ανάλυση των δεδομένων μπορούμε να βγάλουμε συμπεράσματα για τα δεδομένα με τη μορφή ενός γραφήματος, κάτι πολύ πιο εύκολο για κατανόηση από τον ανθρώπινο εγκέφαλο σε σχέση με ένα απλό πίνακα η ένα αρχείο δεδομένων.

Αρχικά σε οποιαδήποτε ανάλυση και να κάνουμε, το πρώτο στάδιο αποτελείται από την φόρτωση των δεδομένων στο αρχείο κώδικα. Στη Python υπάρχουν πολλές βιβλιοθήκες που βοηθούν αυτή τη διαδικασία. Εμείς θα χρησιμοποιήσουμε τη Pandas και θα το διαβάσουμε με την εντολή `read_csv()`. Στη συνέχεια θα ελέγξουμε τις πέντε πρώτες εγγραφές των δεδομένων με την εντολή `df.head()` Με την εντολή `nunique()` μπορούμε να δούμε τις μοναδικές εγγραφές των δεδομένων. Η στήλη source στο αρχείο δεδομένων μας αποτελεί τη στήλη στην οποία

καταγράφεται η συσκευή από την οποία συνδέεται και κάνει τις δημοσιεύσεις του ο εκάστοτε χρήστης. Για την οπτικοποίηση της στήλης αυτής έτσι ώστε να δούμε τα ποσοστά των δημοσιεύσεων που αφορούν διαφορετικές συσκευές χρησιμοποιήθηκε η βιβλιοθήκη seaborn. Η βιβλιοθήκη αυτή περιλαμβάνει συναρτήσεις για τη πραγματοποίηση στατιστικών απεικονίσεων των δεδομένων με λίγες γραμμές κώδικα.

Η βιβλιοθήκη seaborn προσφέρει ευανάγνωστες και ελκυστικές γραφικές αναπαραστάσεις των δεδομένων. Μπορούμε να υπολογίσουμε το ποσοστό διάφορων μετρήσεων μέσω της μεθόδου `value_counts()` της βιβλιοθήκης Pandas. Η μέθοδος `value_counts()` επιστρέφει μια σειρά που περιέχει τις μετρήσεις μοναδικών τιμών. Αυτό σημαίνει, για οποιαδήποτε στήλη σε ένα πλαίσιο δεδομένων, αυτή η μέθοδος επιστρέφει τον αριθμό των μοναδικών καταχωρήσεων σε αυτήν τη στήλη. Βλέπουμε λοιπόν ότι στα δεδομένα μας ένα πολύ μεγάλο ποσοστό χρηστών επιλέγει να δημοσιεύσει στο Twitter από το Android smarphone του. Το ποσοστό είναι αρκετά μεγάλο, φτάνει το 47.5%. Στη δεύτερη θέση βλέπουμε τη web εφαρμογή του Twitter με ποσοστό 36.5% και 3ο στη θέση το Twitter for iPhone με ποσοστό 12%.

Αυτή η καταμέτρηση θα μπορούσε επίσης να χρησιμοποιηθεί και για έρευνα σχετικά με τις προτιμήσεις των Ελλήνων σε συσκευές κινητών τηλεφώνων. Βλέπουμε ότι ένα μεγάλο ποσοστό ελλήνων επιλέγει Android ωστόσο δεν γνωρίζουμε αν το ποσοστό χρηστών που χρησιμοποιούν τη Web εφαρμογή συνδέονται από κινητό η υπολογιστή αφού και στα smartphones υπάρχει αυτή η δυνατότητα. Προκειμένου να πραγματοποιηθεί κάποια έρευνα σχετικά με το τις προτιμήσεις των χρηστών ανάμεσα σε Android/IPhone θα πρέπει αναμφίβολα να συλλεχτούν παραπάνω στοιχεία έτσι ώστε να εξαχθούν σωστά συμπεράσματα. Οπτικοποιούμε το αποτέλεσμα με ένα pie chart.

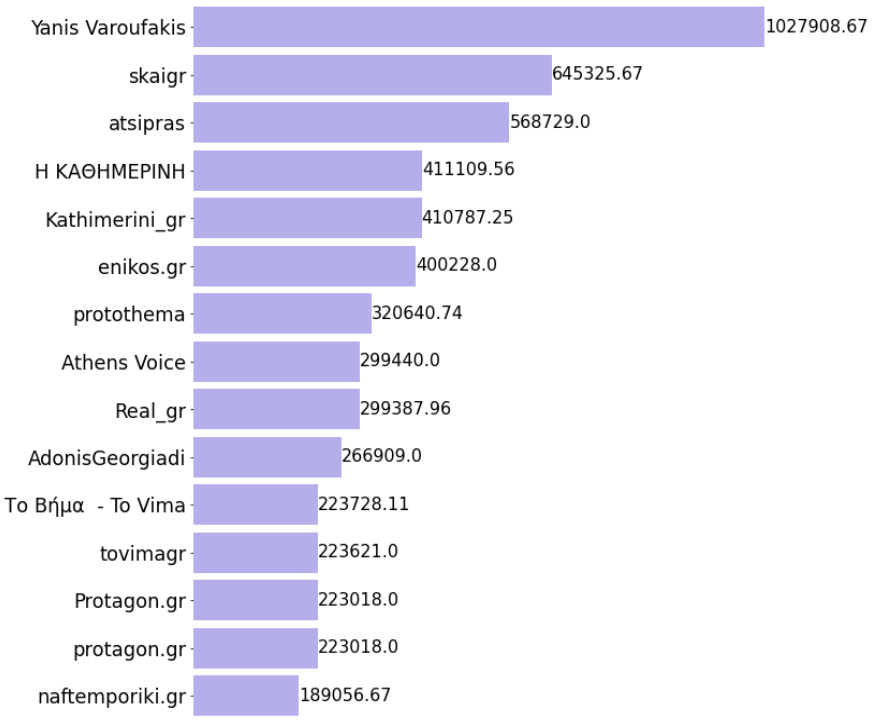


Εικόνα 24: Συσκευές που χρησιμοποιούν οι Έλληνες

Με ανάλογο τρόπο μπορούμε και να εντοπίσουμε τους χρήστες με τους πιο πολλούς ακολούθους. Προκειμένου να εντοπίσουμε και να οπτικοποιήσουμε τους χρήστες με τους περισσότερους ακολούθους μέσα στο αρχείο δεδομένων μας, θα δημιουργήσουμε με τη παρακάτω συνάρτηση την οποία αποθηκεύουμε σε μια μεταβλητή:

```
most_followers=df.groupby('name')['followers_count'].mean().nlargest(25)
```

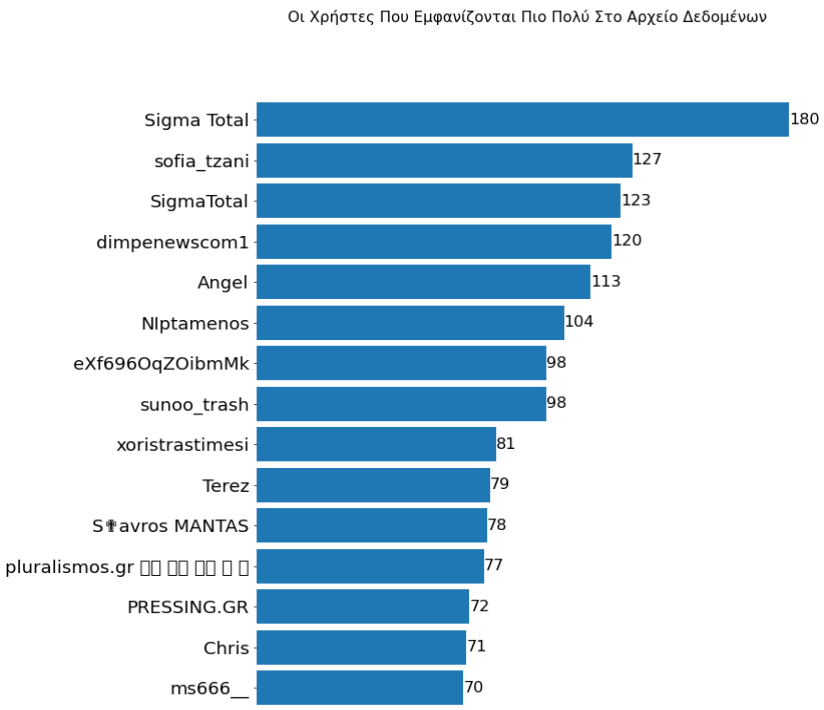
Η συγκεκριμένη συνάρτηση μετράει τους ακολούθους που έχει ο κάθε χρήστης και εντοπίζει τους χρήστες με τους περισσότερους ακολούθους. Στη συγκεκριμένη περίπτωση εντοπίστηκαν οι 25 χρήστες με τους περισσότερους ακολούθους και οπτικοποιήθηκαν με ένα γράφημα ράβδων (barchart). Βλέπουμε ότι ο χρήστης με τους περισσότερους ακόλουθους είναι ο Γιάννης Βαρουφάκης με πάνω από 1 εκατομμύριο ακόλουθους. Δεύτερο σε αριθμό ακολούθων είναι το κανάλι Skai με πάνω από 600 χιλιάδες ακολούθους.



Εικόνα 25: Χρήστες με τους περισσότερους ακολούθους

Προκειμένου να βρούμε τους χρήστες που εμφανίζονται πιο συχνά στο αρχείο δεδομένων, θα φτιάξουμε ένα πίνακα μέσω της βιβλιοθήκης Pandas που θα περιλαμβάνει το username του χρήστη και τη καταμέτρηση των δημοσιεύσεων του. Αυτό θα πραγματοποιηθεί με τις συναρτήσεις `value_counts().reset_index()`. Με την εντολή head(20) προβάλουμε τους 20 επικρατέστερους χρήστες. Φτιάχνουμε το παρακάτω πίνακα και στη συνέχεια οπτικοποιούμε το αποτέλεσμα με ένα barplot.

Username	counts
Sigma Total	180
sofia_tzani	127
SigmaTotal	123
dimpenewscom1	120
Angel	113
NIptamenos	104
eXf696OqZOibmMk	98
sunoo_trash	98
xoristrastimesi	81
Terez	79
S✝avros MANTAS	78
pluralismos.gr GR CY EU	77
PRESSING.GR	72
Chris	71

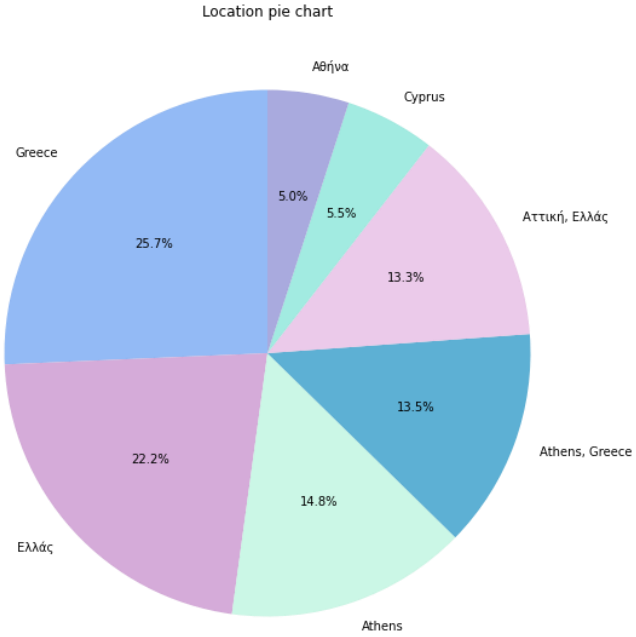


Εικόνα 26: Επικρατέστεροι χρήστες

Προκειμένου να βρούμε τη τοποθεσία από την οποία δημοσιεύθηκαν τα περισσότερα tweets, θα χρησιμοποιήσουμε τις συναρτήσεις `value_counts().reset_index()` της βιβλιοθήκης Pandas. Το αποτέλεσμα της συγκεκριμένης συνάρτησης είναι ο παρακάτω πίνακας. Ορίζουμε ως στήλες τη στήλη Location του αρχείου δεδομένων και counts, τον αριθμό των tweets.

Location	counts
Greece	1829
Ελλάς	1578
Athens	1056
Athens, Greece	961
Αττική, Ελλάς	949
Cyprus	393
Αθήνα	356
Ελλάδα	260
Θεσσαλονίκη, Ελλάς	255
Attiki, Greece	220

Thessaloniki	216

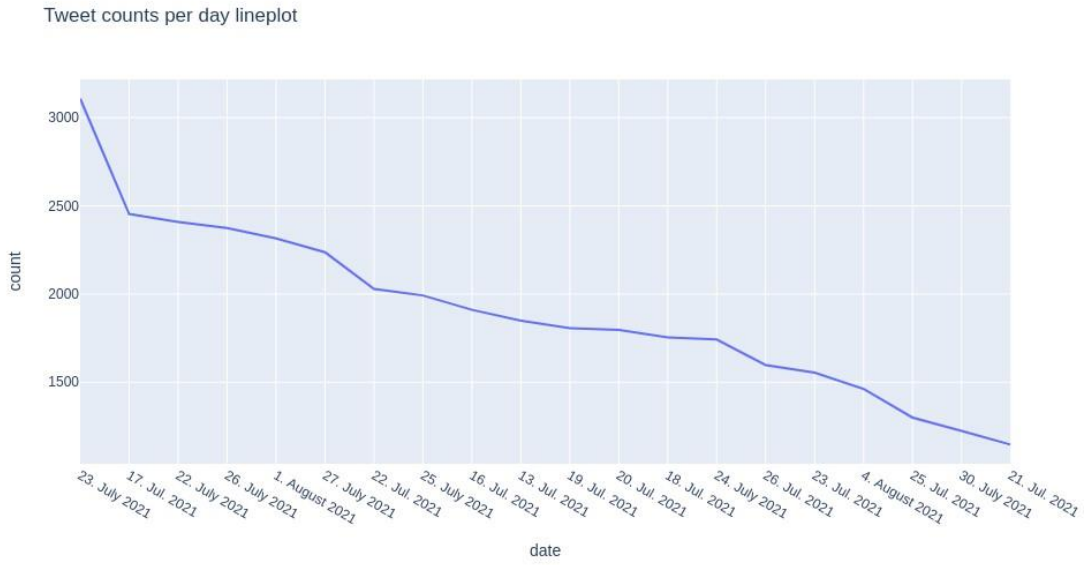


Εικόνα 27: Επικρατέστερες περιοχές αρχείου δεδομένων

Καταμετρούμε τα tweets σύμφωνα με την ημερομηνία που έχουν δημοσιευθεί. Στόχος είναι να εντοπιστεί η ημερομηνία με τα πιο πολλά δημοσιευμένα tweets. Δημιουργείται ο παρακάτω πίνακας. Η μέρα με τις πιο πολλές δημοσιεύσεις είναι η 23 Ιουλίου με 3108 δημοσιεύσεις. Οπτικοποιούμε το αποτέλεσμα της κατανομής των δημοσιεύσεων σε ένα γράφημα γραμμής (lineplot)

date	count
23. July 2021	3108
17. Jul. 2021	2454
22. July 2021	2408
26. July 2021	2374
1. August 2021	2315
27. July 2021	2237
22. Jul. 2021	2029
25. July 2021	1992
16. Jul. 2021	1910
13. Jul. 2021	1849
19. Jul. 2021	1807
20. Jul. 2021	1797
18. Jul. 2021	1754

24. July 2021	1742
26. Jul. 2021	1597
23. Jul. 2021	1554
4. August 2021	1461
25. Jul. 2021	1299
30. July 2021	1224
21. Jul. 2021	1146



Εικόνα 28: Πλήθος δημοσιεύσεων ανά ημέρα

Η Pandas είναι μια βιβλιοθήκη η οποία περιέχει ένα ευρύ φάσμα συναρτήσεων για ανάλυση δεδομένων. Μέσω της εντολής **`str.contains()`** μπορούμε να λάβουμε τη λίστα των tweets στα οποία περιέχεται κάποιο σύμβολο το οποίο μας ενδιαφέρει όπως για παράδειγμα link, hashtag η mention όσον αναφορά το Twitter. Στη συνέχεια μπορούμε να τα καταμετρήσουμε και να τα αριθμήσουμε με την εντολή `value_counts()`**`.sum()`**. Παρουσιάζει ενδιαφέρον το να εντοπίσουμε και να κατατάξουμε τα tweets σύμφωνα με το αν περιέχουν link, hashtag η mention. Ακόμα θα εντοπίσουμε τα tweets τα οποία αποτελούν retweets. Αυτό φυσικά θα πραγματοποιηθεί προγραμματιστικά αφού κάτι τέτοιο θα έπαιρνε πάρα πολύ χρόνο για να πραγματοποιηθεί χειροκίνητα. Δημιουργούμε λοιπόν 3 νέες στήλες στο αρχείο δεδομένων μας:

- *has_url*: αποτελεί τη στείλη στην οποία διακρίνουμε τα tweets σε αυτά που περιέχουν κάποιο link είτε σε αυτά που δεν περιέχουν. Η διαδικασία εντόπισης τους είναι πολύ απλή δεδομένου ότι για να περιέχει link ένα tweet, σίγουρα περιέχει τη λέξη http στο κείμενό του. Στη Python μέσω της βιβλιοθήκης Pandas αυτό πραγματοποιήθηκε μέσω των συναρτήσεων:

```
tweets_without_url = df[df['text'].str.contains('http')==False] για τα
tweets που περιέχουν url και
tweets_with_url = df[df['text'].str.contains('http')==True]
```

- *has_mention*: με αντίστοιχο τρόπο εντοπίστηκαν και τα tweets τα οποία περιέχουν mention. Γνωρίζοντας ότι το mention στο Twitter συμβολίζεται με @, εντοπίστηκαν εύκολα με τη συνάρτηση:

```
str.contains('@')==True, str.contains('http')==False
```

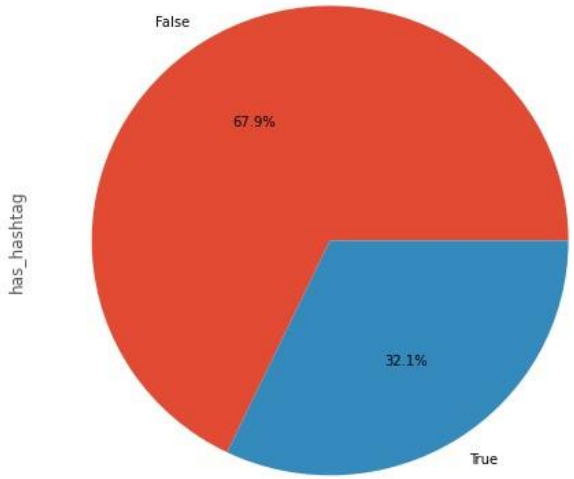
- *has_hashtag*: με τον ίδιο τρόπο διακρίναμε και τη στήλη hashtag, το οποίο στο Twitter συμβολίζεται με #. Επομένως προστέθηκε στη συνάρτηση μας το σύμβολο # και στη συνέχεια τα tweets ταξινομήθηκαν με βάση το αν περιέχουν hashtag η όχι.

- *is_retweet*: είναι τα tweets αυτά τα οποία έχουν αναδημοσιευτεί και δεν αποτελούν την αρχική δημοσίευση του χρήστη αλλά ο χρήστης που αναγράφεται είναι ο χρήστης που κάνει την αναδημοσίευση. Τα retweets θα εντοπιστούν με τον ίδιο τρόπο:

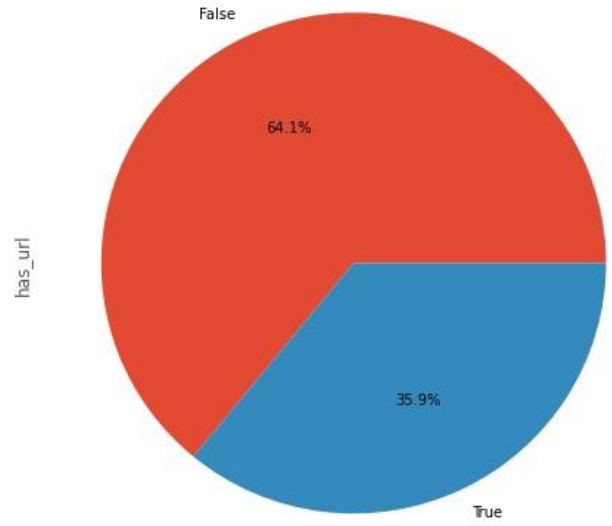
```
retweets = df[df['text'].str.contains('RT')== True]
```

για να βρούμε τον αριθμό των retweets. Τρέχοντας τη συγκεκριμένη εντολή είδαμε όμως ότι πολλά από αυτά τα αποτελέσματα που πήραμε δεν αποτελούν retweets απλά αναφέρουν τη λέξη αυτή η κάποια άλλη με τα δυο στοιχεία που δώσαμε. Για να εντοπιστούν τα πραγματικά retweets χρησιμοποιήθηκε η εντολή startswith('RT') αφού τα retweets στο twitter δηλώνονται με αυτό το συνδυασμό στην αρχή της πρότασης.

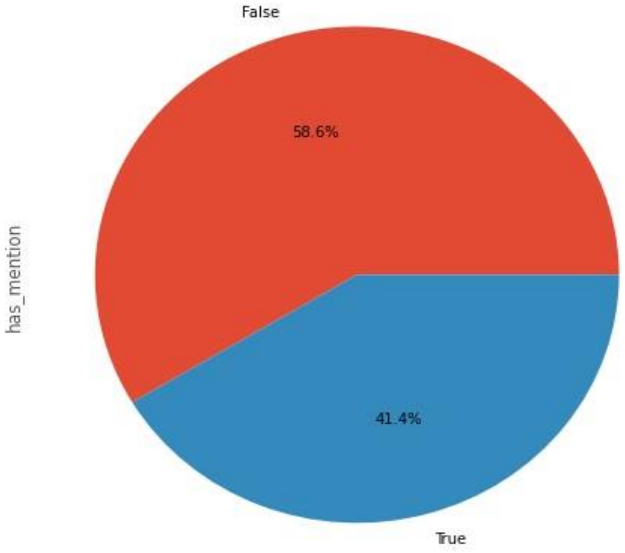
Όπως γίνεται αντιληπτό από το παρακάτω γράφημα, το μεγαλύτερο ποσοστό των δεδομένων περιέχει hashtag. 67.9% των δεδομένων περιέχουν ενώ μόλις 32.1% των δημοσιεύσεων δεν περιέχουν hashtag. Όσον αναφορά τις δημοσιεύσεις που περιέχουν σύνδεσμο, το ποσοστό είναι μικρό 35,9% και το 64.1% των δημοσιεύσεων δεν περιέχει κάποιο σύνδεσμο. Το ποσοστό των tweets που περιέχει αναφορά (mention) αποτελεί το 58.6% ενώ τα tweets που δεν περιέχουν κάποιο mention 41.4%. Τέλος, μόλις το 2.1% των δημοσιεύσεων αποτελούν αναδημοσιεύσεις (retweets).



Εικόνα 29: Ποσοστό δημοσιεύσεων με Hashtag



Εικόνα 30: Ποσοστό δημοσιεύσεων με link



Εικόνα 31: Ποσοστό δημοσιεύσεων με mention

4.4 Προεπεξεργασία δεδομένων

Το σημαντικότερο βήμα στην εξόρυξη δεδομένων κειμένου, όπως αναφέραμε και σε προηγούμενο κεφάλαιο είναι η προεπεξεργασία των δεδομένων. Η προεπεξεργασία των δεδομένων είναι απαραίτητη, καθώς τα αρχικά δεδομένα πάσχουν από διαφόρων ειδών προβλήματα. Σύμφωνα με το Κύρκο (2015), σε αυτά συγκαταλέγονται η ύπαρξη αλληλοσυγκρουόμενων πληροφοριών, η ύπαρξη ασυνεπειών ως προς την κωδικοποίηση, την ονοματοδοσία πεδίων και τις μονάδες μέτρησης, καθώς και η ύπαρξη χαμένων τιμών και θορύβου, τυχαία δηλαδή κυμαινόμενων χωρίς σημαντικό περιεχόμενο.

Οι διπλοτυπίες ή τα ελλιπή δεδομένα μπορεί να παράγουν λανθασμένα ή παραπλανητικά συμπεράσματα. Τα δεδομένα πρέπει να καθαριστούν και να αφαιρεθούν σημεία στίξης και χαρακτήρες που δυσκολεύουν την ανάγνωσή τους από τους αλγορίθμους που θα τρέξουμε στη συνέχεια. Επιπλέον, η ελληνική γλώσσα είναι μια γλώσσα η οποία περιέχει τόνους, κάτι που σημαίνει ότι θα πρέπει να αφαιρεθούν. Επίσης θα πρέπει να αφαιρεθούν οι συχνά επαναλαμβανόμενες λέξεις (stopwords) καθώς δεν προσδίδουν στο κείμενο καμία σημασιολογική αξία.

Όσον αναφορά τα δεδομένα προερχόμενα από το Twitter, θα πρέπει να λάβουμε υπόψιν μας και άλλα πράγματα όπως τους συνδέσμους και τις αναφορές (hashtags) τα οποία θα πρέπει να αφαιρεθούν έτσι ώστε να μείνει αυτούσιο το περιεχόμενο της πρότασης. Επιπλέον, το φαινόμενο των greeklish και των ξένων όρων δυσκολεύει την ανάλυση μας επομένως αφαιρέθηκαν και οι συγκεκριμένες λέξεις. Το γεγονός ότι στο Twitter υπάρχουν αναδημοσιεύσεις περιεχομένου, κάνουν τα δεδομένα να έχουν πολλές επαναλαμβανόμενες δημοσιεύσεις. Συνολικά μαζεύτηκαν 43675 tweets από τα οποία τελικά τα 25343 αποτελούν μοναδικά tweets και κρατήθηκαν για την ανάλυση συναισθήματος. Τα δεδομένα αυτά στη συνέχεια αφού τους αποδοθούν οι ανάλογες τιμές (θετικό/αρνητικό/ουδέτερο) θα χρησιμοποιηθούν ως είσοδος στο μοντέλο πρόβλεψης μας. Τα δεδομένα καθαρίστηκαν και διαμορφώθηκαν έτσι ώστε να είναι ευανάγνωστα από τον αλγόριθμο με τη βοήθεια βιβλιοθηκών της γλώσσας Python.

Αρχικά θα πρέπει να φορτώσουμε τις ανάλογες βιβλιοθήκες στο κώδικά μας οι οποίες περιέχουν τις συναρτήσεις που θα χρειαστούμε για το καθαρισμό των δεδομένων.

Στο συγκεκριμένο στάδιο χρησιμοποιήθηκαν οι εξής βιβλιοθήκες:

- pandas
- matplotlib
- seaborn
- plotly
- re
- string
- ntltk
- wordcloud
- tweet-preprocessor

Τα αρχικά δεδομένα διαμορφώνονται ως εξής:

	Username	Date	Followers	Source	Location	Tweet
0	CaeruleusCalva	4. August 2021	1088.0	Twitter for Android	Not So Ancient Greece	Όπως λένε και οι καθηγητές ΩΡΛ: Τα ανεμβολίαστ...
1	kambanellas	4. August 2021	316.0	Twitter for Android	Cyprus	@percentcrumpet @themist7 @Dr_kypri Δυστυχώς τ...
2	em56464	4. August 2021	116.0	Twitter Web App	NaN	RT @Gerasim73424653: Χειροκροτούν τους πυροσβέ...
3	zetakosmatougk	4. August 2021	2955.0	Twitter for Android	greece	RT @Faklana38: Έχετε μισό λεπτό να σας μιλήσω ...
4	PeterParker_84	4. August 2021	149.0	Twitter Web App	Timbuktu Κάτω Αχαίας	Σκέψου ούτε οι ίδιοι δεν εμπιστεύονται το εμβό...
5	EvgenTzavaras	4. August 2021	1813.0	Twitter for Android	NaN	@TheSupremacist_ @DeucalionEl Ημαρτον. Ξεστραβ...
6	Faklana38	4. August 2021	1091.0	Twitter for Android	Bachelor τσαρδί	Έχετε μισό λεπτό να σας μιλήσω για το εμβόλιο ...
7	e_thessalia	4. August 2021	2200.0	E-ThessaliaAutoPosting	Volos, Greece	Το εμβόλιο της Pfizer αναμένεται να εξασφαλίσε...
8	TheW0lfpup	4. August 2021	2988.0	Twitter for Android	NaN	@Lunaticus_9 Α δηλαδή για το εμβόλιο έχει μια ...
9	NemesisEfialtis	4. August 2021	110.0	Twitter Web App	Αθήνα	@Leo888Alex @lenitataz Η πιθανότητα να πάθει κ...

Η πρώτη τροποποίηση που εφαρμόστηκε ήταν να μετατραπούν όλες οι δημοσιεύσεις σε μικρά γράμματα. Αυτό πραγματοποιήθηκε με την εφαρμογή μιας lambda function. Η συναρτήσεις lambda δουλεύουν όπως μια κλασική συνάρτηση της Python όμως είναι ανώνυμες. Οι ανώνυμες αυτές συναρτήσεις είναι ιδανικές έτσι ώστε να μετατρέψουμε αποτελεσματικά ένα μεγάλο όγκο δεδομένων με πολύ λίγες γραμμές κώδικα. Η συνάρτηση εφαρμόστηκε στη στήλη Tweet και εφαρμόστηκε η εντολή `str.lower()` της βιβλιοθήκης Pandas. Στη συνέχεια θα αφαιρεθούν οι τόνοι από τα δεδομένα με μια συνάρτηση στην οποία αντικαθιστούμε τα φωνήεντα με τα αντίστοιχα φωνήεντα χωρίς το τονισμό.

Είναι επίσης πολύ σημαντικό να αφαιρεθούν οι λατινικοί χαρακτήρες που συνήθως αποτελούν ονόματα χρήστη η εκφράσεις γραμμένες σε greeklish. Επίσης θα αφαιρεθούν και όλοι οι αριθμοί από τα δεδομένα. Οι λατινικοί χαρακτήρες αποτελούν χαρακτήρες με κωδικοποίηση ASCII επομένως θα γράψουμε μια συνάρτηση για την αφαίρεση αυτών των χαρακτήρων. Με την εντολή `str.replace()` της Pandas θα αντικαταστήσουμε τους χαρακτήρες αυτούς με κενό μέσω μιας ανώνυμης συνάρτησης η οποία θα εφαρμοστεί σε όλη τη στήλη των δημοσιεύσεων. Τα δεδομένα διαμορφώθηκαν ως εξής:

```
1. όπως λένε και οι καθηγητές ωρλ: τα ανεμβολίαστα ζώα πεθαίνουν από #κορωνοϊό. τα εμβολιασμένα ζώα δεν πεθαίνουν από κορωνοϊό.\nσυμπέρασμα: δεν υπάρχει #εμβόλιο για το να μην είναι κανείς ζώο (οργανισμός από ζωϊκά κύτταρα τα οποία ο κορωνοϊός έχει στόχους).
```

```
2 @: χειροκροτούν τους πυροσβέστες μέχρι να σβήσει η φωτιά,  μετά κατάρες κ απειλές απόλυσης αν δεν κάνουν το εμβόλιο.
```

```
3 @: έχετε μισό λεπτό να σας μιλήσω για το εμβόλιο της ; #μητσοτακη_γαμιεσαι #μητσοτακη_παραιτησου #μητσοτακη_καθαρμα #φωτια
```

```
4 σκέψου ούτε οι ίδιοι δεν εμπιστεύονται το εμβόλιο τους.
```

Στη συνέχεια αφαιρούμε τους τόνους εφαρμόζοντας τη συνάρτηση που γράψαμε προηγουμένως. Όπως βλέπουμε, τα δεδομένα έχουν ακόμα κάποια σύμβολα όπως @, __ τα οποία θα πρέπει να αφαιρεθούν. Επιπλέον θα αφαιρεθούν και όλοι οι σύνδεσμοι καθώς δεν προσφέρουν σημασιολογική αξία στην ανάλυση μας. Αυτό θα επιτευχθεί με τη συνάρτηση `re.sub('https\S+', '', str(x))` την οποία αποθηκεύουμε σε μια lambda function. Προκειμένου να αφαιρεθούν όλα τα υπόλοιπα σημεία στίξης από τα δεδομένα θα χρησιμοποιηθεί η συνάρτηση `x.translate(str.maketrans('','',string.punctuation))` της βιβλιοθήκης String. Τα δεδομένα προς ανάλυση έχουν διαμορφωθεί ως εξής:

```
0 οπως λενε και οι καθηγητες ωρλ τα ανεμβολιαστα ζωα πεθαινουν απο κορωνοιο τα εμβολιασμενα ζωα δεν πεθαινουν απο κορωνοιο συμπερασμα δεν υπαρχει εμβολιο για το να μην ειναι κανεις ζωο οργανισμος απο ζωικα κυτταρα τα οποια ο κορωνοιος εχει στοχους
```

```
1 δυστυχως τα δεδομενα δεν αλλαζουν ειτε σου αρεσει ειτε οχι η δε θες εμβολιο προβλημα σου να μη θελεις να πιστεψεις δεδομενα επειδη ετσι σου καπνισε σε κανει προβληματικο σορρυ αλλα δεν υπαρχει κατι αλλο
```

2 χειροκροτουν τους πυροσβεστες μεχρι να σβησει η φωτια μετα καταρες κ απειλες απολυσης αν δεν κανουν το εμβολιο 3 εχετε μισο λεπτο να σας μιλησω για το εμβολιο της μητσοτακη γαμιεσαι μητσοτακη παραιτησου μητσοτακηκαθαρμα φωτια
4 σκεψου ουτε οι ιδιοι δεν εμπιστευονται το εμβολιο τους

Τα επαναλαμβανόμενα tweets είναι πάρα πολλά, θα πρέπει να αφαιρεθούν. Στη στήλη Location υπάρχουν πολλές κενές εγγραφές. Τις αντικαθιστούμε με 'prefer not to say' προκειμένου να σβήσουμε στη συνέχεια τις κενές εγγραφές από τη στήλη Tweet. Αφαιρούμε τη συνέχεια τα επαναλαμβανόμενα tweets καθώς και τις κενές εγγραφές. Το τελικό στάδιο περιλαμβάνει την αφαίρεση των τερματικών όρων.

	Username	Date	Followers	Source	Location	Tweet
0	CaeruleusCalva	4. August 2021	1088.0	Twitter for Android	Not So Ancient Greece	οπως λενε και οι καθηγητες ωρλ τα ανεμβολιαστα...
1	kambanellas	4. August 2021	316.0	Twitter for Android	Cyprus	δυστυχως τα δεδομενα δεν αλλαζουν ειτε σου ...
2	em56464	4. August 2021	116.0	Twitter Web App	prefer not to say	χειροκροτουν τους πυροσβεστες μεχρι να σβησε...
3	zetakosmatougk	4. August 2021	2955.0	Twitter for Android	greece	εχετε μισο λεπτο να σας μιλησω για το εμβολι...
4	PeterParker_84	4. August 2021	149.0	Twitter Web App	Timbuktu Κάτω Αχαΐας	σκεψου ουτε οι ιδιοι δεν εμπιστευονται το εμβο...
...	...	...	...	...	...	...
43663	GiorgosSahinis	3. August 2021	4759.0	Twitter Web App	prefer not to say	αληθειες και ψεματα για το εμβολιο
43664	Giorgosstefano2	3. August 2021	23938.0	Twitter for Android	Αττική, Ελλάς	καναμε το εμβολιο για να παμε τελικα απο πυρκα...
43665	MyPortalGR	3. August 2021	5435.0	dlvr.it	prefer not to say	εμβολιο η ανοσια διατηρειται τουλαχιστον για ...
43668	NTrachiotis	3. August 2021	4743.0	Twitter for Android	Ελευσινα	καυσωνας γυναικοκτονια εμβολιασμος βαρυμπομ...
43669	KyrEnotiadis	3. August 2021	2423.0	Twitter Web App	Ελλάς	η ελλαδα στη κοκκινη λιστα του ισραηλ οποιος ...

25344 rows × 6 columns

Προκειμένου να εντοπίσουμε τα tweets με τους πιο πολλούς αλλά και με τους λιγότερους χαρακτήρες, θα φτιάξουμε μια νέα στήλη στο αρχείο μας η οποία θα μετράει το μάκρος των χαρακτήρων των tweets. Αυτό θα πραγματοποιηθεί με τη συνάρτηση:

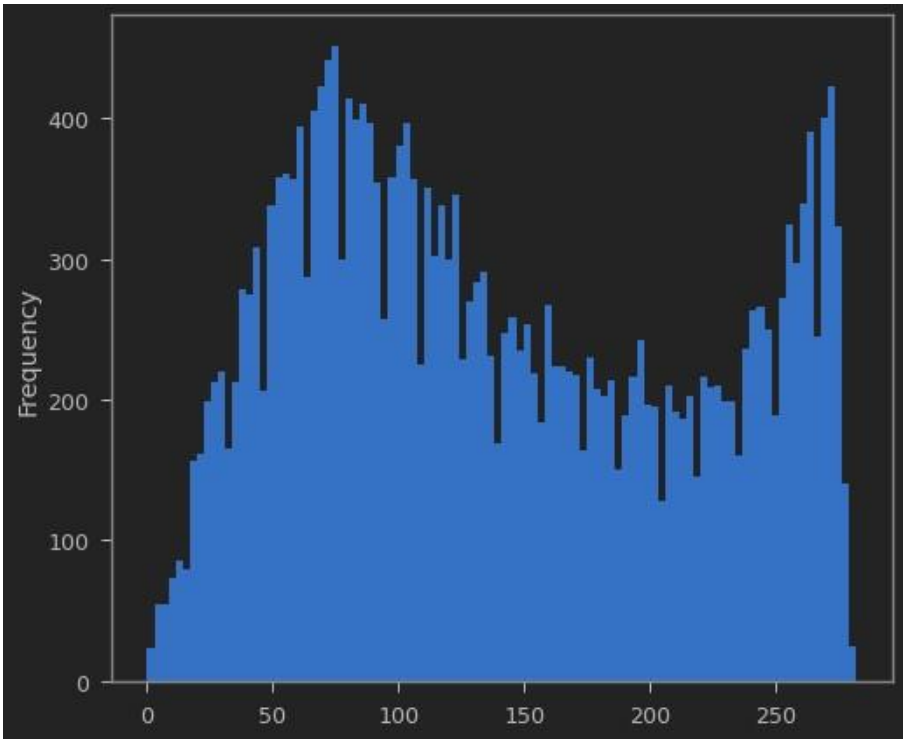
```
df['length'] = df['Tweet'].apply(len).
```

Στη συνέχεια θα εφαρμόσουμε τη συνάρτηση `describe()` προκειμένου να λάβουμε κάποια στατιστικά που αφορούν τα tweets.

count	25339.000000
mean	140.053633
std	77.204873
min	1.000000
25%	75.000000
50%	126.000000
75%	208.000000
max	282.000000

Ο μέσος όρος των δημοσιεύσεων διαμορφώνεται στους 140 χαρακτήρες, ενώ το μικρότερο και μεγαλύτερο tweet αποτελούνται από ένα και 282 χαρακτήρες αντίστοιχα. Παρακάτω βλέπουμε το μικρότερο και μεγαλύτερο tweet της λίστας καθώς και την οπτική κατανομή (σχήμα 32).

```
#Longest tweet
'μπορεις να εκφερεις αποψη εννοειται αλλα διαβασε πρωτα αυτο το εμβολιο που λες
ηταν κατι βιαστικο συγκριτικα με το πως πρεπει να βγαινει ενα εμβολιο και σε
μια πανδημια ειναι λογικο και επομενο να υπαρξουν μεταλλαξεις το εμβολιο το
βιαστικο δεν μπορει να μαντεψει τις μεταλλαξεις'
```



Εικόνα 32: Διακύμανση χαρακτήρων tweets

Τα tweets με τους λιγότερους χαρακτήρες αποτελούνται από emojis.

	Username	Date	Followers	Source	Location	Tweet	length
12255	uU1anaBa5tHy1d1	18. Jul. 2021	253.0	Twitter for Android	prefer not to say	😉	1
14797	the_most_ironic	19. Jul. 2021	192.0	Twitter Web App	Greece	👇	1
15818	zanoulitsa	20. Jul. 2021	794.0	Twitter for Android	Crete	🤣	1
22589	Dreamon_tweet	29. July 2021	483.0	Twitter for Android	Kavala, Macedonia, Greece	😀	1
24651	FrantseskaA	1. August 2021	555.0	Twitter for Android	prefer not to say	💖	1

4.5 Αφαίρεση τερματικών όρων (stopwords)

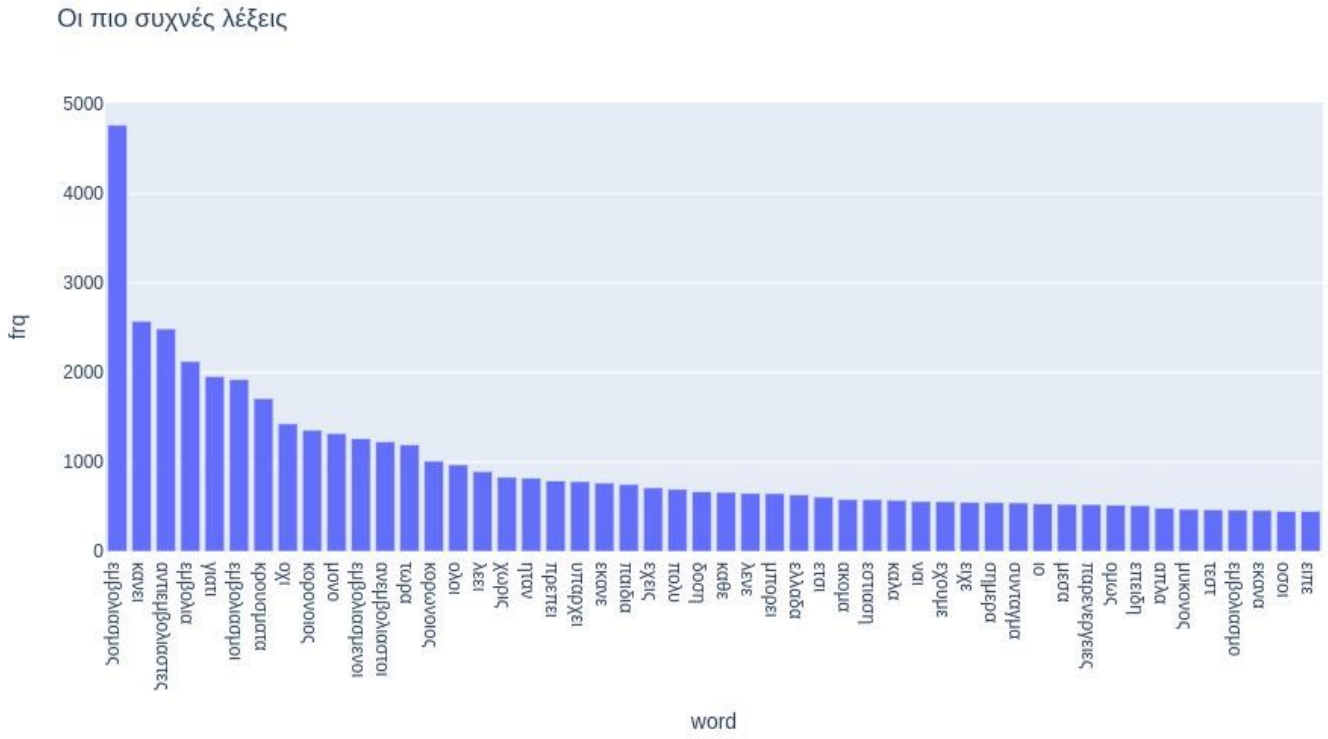
Είναι πολύ σημαντικό να αφαιρεθούν οι τερματικοί όροι οι οποίοι αποτελούν λέξεις που επαναλαμβάνονται συνεχώς και δεν προσδίδουν καμία συναισθηματική αξία στο κείμενο. Αυτό θα πραγματοποιηθεί με τη βιβλιοθήκη nltk της Python και το ελληνικό πακέτο τερματικών όρων.

Στη συνέχεια οι προτάσεις θα “σπάσουν” σε λέξεις οι οποίες θα είναι το input του κατηγοριοποιητή που θα φτιάξουμε στη συνέχεια. Επίσης θα καταμετρήσουμε τις πιο συνηθισμένες λέξεις που υπάρχουν στα δεδομένα μας με τη βιβλιοθήκη Counter και τη συνάρτιση `most_common()`.Η συγκεκριμένη συνάρτηση μετράει και κατηγοριοποιεί τις λέξεις σύμφωνα με το πόσες φορές περιέχονται σε ένα κείμενο:

```
[('εμβολιασμος', 4766),
 ('αντιεμβολιαστες', 2489),
 ('εμβολια', 2124),
```

```
 ('εμβολιασμοι', 1923),
 ('κρουσματα', 1710),
('οχι', 1428),
 ('κορονοιος', 1356),
 ('μονο', 1319),
 ('εμβολιασμενοι', 1263),
 ('ανεμβολιαστοι', 1225),
 ('τωρα', 1193),
 ('κορωνοιος', 1012),
 ('ολοι', 969),
 ('λεει', 895),
 ('χωρις', 831),
 ('πρεπει', 790),
 ('υπαρχει', 782),
 ('εκανε', 765),
 ('παιδια', 749),
 ('δοση', 670),
 ('καθε', 662),
 ('λενε', 650),
 ('μπορει', 648),
 ('ελλαδα', 634),
 ('ετσι', 609),
 ('ακομα', 581),
 ('εστιαση', 579),
 ('καλα', 573),
 ('ναι', 560),
 ('εχουμε', 558),
('ειχε', 550),
 ('σημερα', 547),
 ('συνταγμα', 544),
 ('μεσα', 527),
 ('παρενεργειες', 525),
 ('ομως', 519),
 ('επειδη', 513),
 ('απλα', 485),
 ('μυκονος', 473),
 ('τεστ', 468),
 ('εμβολιασμο', 463),
 ('εκανα', 460),
 ('οσοι', 450),
 ('ειπε', 450),
 ('κανουμε', 436),
 ('μεχρι', 435),
 ('ας', 433),
 ('μμεξεφτιλες', 427),
 ('χρονια', 425),
 ('μητσοτακης', 422)]
```

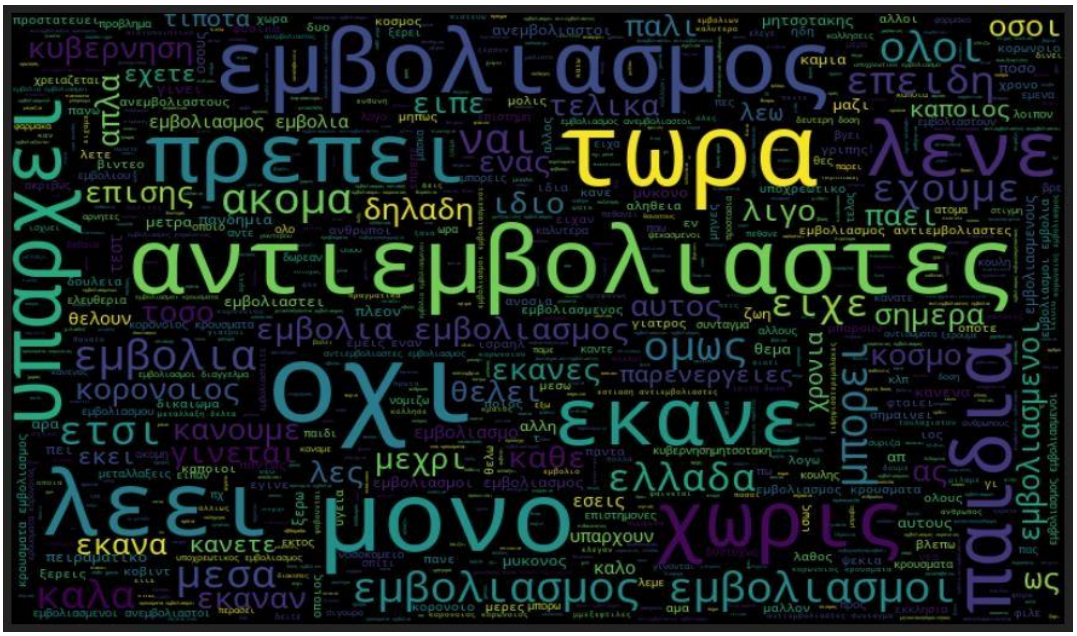
Δημιουργείται ένα γράφημα με τις καταμετρημένες λέξεις μέσω της βιβλιοθήκης plotly.express



Εικόνα 33: Συνηθέστεροι όροι στα δεδομένα

4.6 Κατασκευή σύννεφου λέξεων

Ένας τρόπος να οπτικοποιήσουμε τα δεδομένα κειμένου μας προκειμένου να οπτικοποιήσουμε τις πιο συνηθισμένες λέξεις είναι ένα σύννεφο λέξεων. Στη Python, αυτό θα πραγματοποιηθεί με τη βιβλιοθήκη wordcloud. Τα σύννεφα λέξεων ή σύννεφα ετικετών αποτελούν γραφικές αναπαραστάσεις συχνότητας λέξεων.



Εικόνα 34: Σύννεφο λέξεων με τις συνηθέστερες λέξεις

4.7 Συναισθηματική ανάλυση σε tweets στην ελληνική γλώσσα.

Η συναισθηματική ανάλυση των δημοσιεύσεων των κοινωνικών δικτύων είναι μια πολύ διαδεδομένη μέθοδος εξαγωγής συμπερασμάτων σχετικά με πληθυσμούς. Δυστυχώς είναι ένα πεδίο που δεν έχει μελετηθεί αρκετά στην ελληνική γλώσσα και ως αποτέλεσμα, δεν υπάρχουν αρκετές βιβλιοθήκες και αλγόριθμοι που να μπορούν να επεξεργαστούν επαρκώς το ελληνικό κείμενο. Αυτό κάνει τη διαδικασία πιο πολύπλοκη και αποτελεί πρόκληση για τον ερευνητή. Μια πολύ ενδιαφέρουσα ερευνητική προσπάθεια αποτελεί η μελέτη των Beleveslis, Tjortjis, Psaradelis και Nikoglou στην οποία παρουσιάζουν μια υβριδική προσέγγιση ανάλυσης συναισθήματος σε tweets που αφορούν τη πολιτική στην Ελληνική γλώσσα. Παρουσιάζεται σε πρώτη φάση ένα σύστημα εξόρυξης γνώμης βασισμένο σε λεξικό συναισθήματος και σε δεύτερη φάση εκπαιδεύεται ένα μοντέλο με αλγόριθμο Random Forest το οποίο είναι σε θέση να προβλέψει συναισθηματικά ελληνικό κείμενο.

Μερικές από τις προσεγγίσεις που αφορούν τη συναισθηματική ανάλυση που έχουν προταθεί σε άλλες γλώσσες, απαιτούν τη μετάφραση των δεδομένων στην αγγλική γλώσσα και στη συνέχεια την εκπαίδευση ενός μοντέλου έτσι ώστε να καταταχτούν τα κείμενα στην αντίστοιχη κλάση. Φυσικά αυτή η μέθοδος δεν αποτελεί τη πιο σωστή μέθοδο ανάλυσης συναισθήματος, ωστόσο είναι η απλούστερη μέθοδος καθώς η Python διαθέτει πολλές βιβλιοθήκες και λεξικά για το σκοπό αυτό. Δυστυχώς οι προσεγγίσεις που αφορούν την ελληνική γλώσσα διαθέτουν λεξικά με περιορισμένο αριθμό λέξεων, κάτι που δυσκολεύει τη μελέτη συναισθήματος και κατά συνέπεια το αποτέλεσμα. Οι κλάσεις που ορίζονται για τη κατάταξη συναισθημάτων μπορεί να είναι είτε δυαδικές μέθοδοι κατηγοριοποίησης (θετικό/αρνητικό) είτε μέθοδοι με παραπάνω κλάσεις (θετικό/αρνητικό/ουδέτερο). Πολλές προσεγγίσεις που αφορούν τη συναισθηματική ανάλυση επιλέγουν να αφαιρέσουν τις δημοσιεύσεις που περιέχουν έντονα φορτισμένη γλώσσα και πολλές

φορές υβρεολόγια. Ωστόσο, στη συγκεκριμένη ανάλυση κάτι τέτοιο δεν θα γίνει διότι η υβρεολόγια είναι κάτι που συναντάται σε ένα μεγάλο ποσοστό δημοσιεύσεων στα κοινωνικά δίκτυα και θα χαθεί πολύτιμη πληροφορία. Επιπλέον καθώς η υβρεολόγια αποτελεί μέρος της καθημερινότητας των δημοσιεύσεων των κοινωνικών δικτύων και δεν αφορά μόνο τα αρνητικά tweets, θα αποτελέσει πρόκληση για το μοντέλο που θα δημιουργήσουμε. Γενικότερα, εφόσον η ανάλυση μας περιλαμβάνει δημοσιεύσεις κοινωνικών δικτύων, το περιεχόμενο αποτελεί αναρτήσεις με μεγάλο ποσοστό θορύβου αλλά και προτάσεων ασυνάρτητων. Επομένως η ανάλυση δημοσιεύσεων κοινωνικών δικτύων αποτελεί από μόνη της μια πρόκληση καθώς το αποτέλεσμα πολλές φορές μπορεί να παρερμηνευτεί από τον αλγόριθμο.

Στα πλαίσια της συγκεκριμένης εργασίας, θα δοκιμαστούν και οι δυο προσεγγίσεις που αναφέραμε έτσι ώστε να δημιουργηθεί ένα ολοκληρωμένο σύστημα ανάλυσης συναισθήματος. Πολύ σημαντικό ρόλο στη τελική απόδοση της προσέγγισης που αφορά τη μεταφορά των δημοσιεύσεων στην αγγλική γλώσσα αποτελεί η μέθοδος αυτόματης μετάφρασης η οποία θα επιλέξουμε. Ένας αποδοτικότερος τρόπος να γίνει αυτή η ανάλυση θα ήταν να βαθμολογηθούν δημοσιεύσεις από ανθρώπους έτσι ώστε να δοθούν σαν είσοδο στο μοντέλο που θα κατασκευάσουμε. Στα πλαίσια της συγκεκριμένης μελέτης, προκειμένου να λάβουμε τη κλάση στην οποία ανήκει η κάθε δημοσίευση, θα χρησιμοποιηθούν δυο διαφορετικές βιβλιοθήκες της Python οι οποίες αναλύουν δημοσιεύσεις κοινωνικών δικτύων στην αγγλική γλώσσα. Στη συνέχεια θα κατασκευαστεί ένα μοντέλο πρόβλεψης, το οποίο δέχεται σαν είσοδο βαθμολογημένα tweets στην ελληνική γλώσσα. Τέλος το μοντέλο θα εφαρμοστεί επιπλέον και σε ένα άλλο αρχείο δεδομένων με διαφορετικές δημοσιεύσεις. Στο ποσοστό επιτυχίας/αποτυχίας του μοντέλου πρόβλεψης που θα κατασκευαστεί, μεγάλο ποσοστό θα στηριχτεί στην επιτυχία είτε αποτυχία της αυτόματης μετάφρασης που θα πραγματοποιήσουμε καθώς και η ποιότητα των δεδομένων που θα επιλέξουμε σαν δεδομένα εκπαίδευσης.

4.7.1 Αυτόματη μετάφραση δημοσιεύσεων

Προκειμένου να μεταφραστούν τα δεδομένα στα αγγλικά χρησιμοποιήθηκε η βιβλιοθήκη Googletrans της Python η οποία επιτρέπει την επικοινωνία με το Google Translate API. Η μετάφραση που πραγματοποιήθηκε είναι αρκετά αποδοτική, αν και είναι δύσκολο να μεταφραστούν νεολογισμοί και εκφράσεις καθημερινής γλώσσας. Η αλγόριθμοι που χρησιμοποιούνται στην ανάλυση συναισθήματος είναι σε θέση να αναλύουν και να κατηγοριοποιούν τα δεδομένα βασιζόμενοι στις λέξεις και όχι στις προτάσεις κάτι που πραγματοποιεί πλέον πολύ αποδοτικά η υπηρεσία Google translate. Δυστυχώς η ελληνική γλώσσα δεν έχει πολλά λεξικά που να μπορούν να χρησιμοποιηθούν στην ανάλυση συναισθήματος. Προκειμένου να εκπαιδευτεί στη συνέχεια της εργασίας μας το μοντέλο πρόβλεψης, τα δεδομένα θα πρέπει να κατηγοριοποιηθούν και να τους αποδοθεί το ανάλογο συναίσθημα. Υπάρχουν πολλές προσεγγίσεις που αφορούν τη κατηγοριοποίηση συναισθήματος. Στη παρούσα επιλέχτηκαν οι προσεγγίσεις Vader και Textblob καθώς κατηγοριοποιούν αποδοτικά κείμενο προερχόμενο από τα κοινωνικά δίκτυα.

4.8 Κατηγοριοποίηση συναισθήματος δεδομένων

Οι μέθοδοι που θα χρησιμοποιήσουμε για την κατηγοριοποίηση των δεδομένων οι Vader και Textblob. Οι συγκεκριμένες μέθοδοι βασίζονται σε μοντέλα κατηγοριοποίησης τα οποία αναλύουν αποδοτικά περιεχόμενο προερχόμενο από τα κοινωνικά δίκτυα και το κατηγοριοποιούν στην ανάλογη κλάση μετρώντας τη πολικότητα του κειμένου. Οι κλάσεις που θα κατατάξουμε τα δεδομένα είναι: θετικό/αρνητικό/ουδέτερο. Το αρνητικό τους είναι ότι δέχονται σαν είσοδο μόνο αγγλικό κείμενο. Η μέθοδος Textblob ωστόσο δίνει τη δυνατότητα να εκπαιδεύσει κάποιος ένα δικό του μοντέλο κατηγοριοποίησης βασισμένο σε Naive Bayes μέσω της μεθόδου textblob.classifiers. Ωστόσο προϋπόθεση για να εκπαιδευτεί ένα μοντέλο είναι να υπάρχουν κατηγοριοποιημένα tweets επομένως στη παρούσα θα χρησιμοποιηθεί το μοντέλο που προσφέρει η βιβλιοθήκη Textblob έτσι ώστε να κατηγοριοποιήσουμε τα tweets.

4.8.1 Μέθοδος Vader

Το λεξικό Vader αποτελεί ένα πλούσιο λεξικό συναισθήματος που χρησιμοποιείται ευρέως στην ανάλυση συναισθήματος στα κοινωνικά δίκτυα. Διαθέτει ανάλυση συναισθήματος emojis, κάτι που είναι πολύ σύνηθες στα κοινωνικά δίκτυα. Επίσης μπορεί να αναγνωρίσει και να αναλύσει κείμενο με θόρυβο καθώς και εκφράσεις αργκό που χρησιμοποιούνται στην αγγλική γλώσσα.

Είναι διαθέσιμο μέσω της βιβλιοθήκης vaderSentiment της Python. Η προσέγγιση ανάλυσης συναισθήματος με το λεξικό Vader ανήκει στις προσεγγίσεις βασισμένες σε κανόνες (rule-based sentiment analysis). Οι προσεγγίσεις βασισμένες σε κανόνες δεν απαιτούν την εκπαίδευση του κατηγοριοποιητή αλλά βασίζονται σε μια σειρά κανόνων οι οποίοι κατηγοριοποιούν το κείμενο σύμφωνα με τη πολικότητα του. Οι συγκεκριμένες προσεγγίσεις είναι εύκολες στη χρήση και αποδοτικές όσον αναφορά τη μέτρηση της πολικότητας του κειμένου. Η μέθοδος Vader αναλύει το συναίσθημα της πρότασης βασισμένη σε ένα λεξικό συναισθήματος πάνω από 7000 όρων. Δυστυχώς προς το παρόν η μέθοδος είναι σε θέση να αναγνωρίζει μόνο το αγγλικό κείμενο, ωστόσο υπάρχει η δυνατότητα μετάφρασης του λεξικού και προσαρμογής του σε οποιαδήποτε γλώσσα θέλουμε.

Η διαδικασία αυτή πιθανόν να είναι αποδοτική όμως πέρα από τους όρους, θα πρέπει να προσαρμοστούν στην γλώσσα επιλογής εκφράσεις αργκό καθώς και διαδεδομένες εκφράσεις που χρησιμοποιούν οι χρήστες των κοινωνικών δικτύων καθώς και άλλοι γραμματικοί κανόνες που αφορούν τη γλώσσα στόχο. Σε μελέτη τους οι Karsten Michael Tymann, Matthias Lutz, Patrick Palsbr¨oker, and Carsten Gips πρότειναν τη μέθοδο Vader προσαρμοσμένη στην Γερμανική γλώσσα μεταφράζοντας τους όρους και τους ιδιωματισμούς μέσω του google translate API, προσαρμόζοντας το περιεχόμενο και το κώδικα στη Γερμανική γλώσσα. Η μέθοδος αυτή λειτούργησε αποδοτικά και κατέταξε επιτυχώς ένα μεγάλο ποσοστό των δημοσιεύσεων.

Επιπλέον, υπάρχει η βιβλιοθήκη vader-multi της Python η οποία επικοινωνεί με το google translate API μεταφράζοντας το κώδικα του πακέτου στη γλώσσα στόχο και αναλύοντας έτσι κείμενο σε άλλες γλώσσες. Στα πλαίσια της παρούσας εργασίας, τα tweets μεταφράστηκαν στην αγγλική γλώσσα μέσω του google translate API και κατηγοριοποιήθηκαν σύμφωνα με το συναίσθημα τους σε θετικό/αρνητικό/ουδέτερο. Η μέθοδος Vader προέβλεψε σωστά ένα πολύ μεγάλο ποσοστό των δημοσιεύσεων που υπάρχουν στα δεδομένα μας σε αντίθεση με ένα ελληνικό λεξικό συναισθήματος που δοκιμάστηκε δυστυχώς όμως κατηγοριοποίησε λάθος το μεγαλύτερο ποσοστό των δημοσιεύσεων.

4.8.2 Μέθοδος TextBlob

Η βιβλιοθήκη TextBlob είναι μια βιβλιοθήκη της Python για τη κατηγοριοποίηση δεδομένων κειμένου. Παρέχει μια απλή στη χρήση διεπαφή χρήστη για την επεξεργασία της φυσικής γλώσσας για διεργασίες όπως μέρος της ομιλίας Tagging (part-of-speech tagging) το οποίο αποτελεί μια διαδικασία επισήμανσης των προτάσεων βασισμένη στους ορισμούς τους. Επίσης προσφέρει εξαγωγή φράσης ουσιαστικών (noun phrase extraction), την εξαγωγή δηλαδή των φράσεων που αποτελούν είτε ουσιαστικό είτε αντωνυμία.

Επίσης η TextBlob προσφέρει ανάλυση συναισθήματος, κατηγοριοποίηση κειμένου, μετάφραση και πολλές άλλες δυνατότητες όπως λημματοποίηση (lemmatization) και Tokenization δηλαδή το ‘σπάσιμο’ μιας πρότασης σε λέξεις. Η textblob δίνει τη δυνατότητα στο χρήστη να εκπαιδεύσει ένα custom μοντέλο κατηγοριοποίησης βασισμένο σε Naive Bayes. Η textBlob δίνει τη δυνατότητα ενσωμάτωσης της βάσης δεδομένων WordNet η οποία αποτελεί μια βάση δεδομένων λέξεων στην αγγλική γλώσσα. Όσον αναφορά την ανάλυση συναισθήματος, τα δεδομένα κατηγοριοποιούνται σύμφωνα με τη πολικότητα τους η οποία κυμαίνεται από -1 έως 1 και τα δεδομένα κατατάσσονται ως :

- Θετικά για πολικότητα μεγαλύτερη από 0

- Ουδέτερα, όταν η πολικότητα είναι ίση με το μηδέν
- Αρνητικά, για πολικότητα μικρότερη από 0

Μια επιπλέον μέτρηση της ανάλυσης συναισθήματος που προσφέρει η textBlob είναι η υποκειμενικότητα η οποία αποτελεί τη προσωπική γνώμη του ομιλητή ενός κειμένου η οποία διαφέρει από άτομο σε άτομο και επηρεάζεται από διάφορους παράγοντες όπως κοινωνικούς η πολιτικούς. Στην ανάλυση μας θα αρκεστούμε στην ανάλυση της πολικότητας και κατά αυτό το τρόπο θα κατατάξουμε τα δεδομένα στις τρεις κλάσεις που προαναφέραμε.

4.9 Συναισθηματική ανάλυση δημοσιεύσεων

Τα δεδομένα μεταφράστηκαν στην αγγλική γλώσσα και στη συνέχεια αναλύθηκαν συναισθηματικά μέσω των δυο βιβλιοθηκών της Python που προαναφέραμε. Στη συνέχεια δημιουργήθηκαν δυο διαφορετικά αρχεία δεδομένων, ένα για τη μέθοδο TextBlob και ένα για τη μέθοδο Vader. Τα μεταφρασμένα tweets μετατράπηκαν όλα σε μικρά γράμματα και αφαιρέθηκαν τυχόν σημεία στίξης που μπορεί να περιείχαν. Ανατρέχοντας στις πρώτες και στις τελευταίες γραμμές των δεδομένων, βλέπουμε ότι η μετάφραση στην αγγλική γλώσσα έχει γίνει αποδοτικά.

Στη συνέχεια εφαρμόζουμε τη συναισθηματική ανάλυση με τη μέθοδο Textblob. Αυτό θα επιτευχθεί με τη μέθοδο `sentiment.polarity()` της βιβλιοθήκης Textblob. Ορίζουμε ως αρνητικά tweets τα tweets με πολικότητα μικρότερη από 0, ουδέτερα όσα είναι ίσα με 0 και θετικά όσα είναι πάνω από 0. Δημιουργούνται δυο νέες στήλες στο αρχείο δεδομένων Polarity και label. Προκειμένου να βρούμε τα πιο θετικά και αρνητικά tweets θα εφαρμόσουμε την εντολή describe() στη στήλη Polarity προκειμένου να λάβουμε στατιστικά στοιχεία για τη πολικότητα των tweets.

count	25339.000000
mean	0.028353
std	0.238317
min	-1.000000
25%	0.000000
50%	0.000000
75%	0.100000
max	1.000000

Δημιουργείται ένα καινούργιο αρχείο δεδομένων προκειμένου να εξάγουμε τις κλάσεις και τη πολικότητα των tweets σύμφωνα με τη μέθοδο Vader. Η Vader προσφέρει ανάλυση συναισθήματος με τη συνάρτηση `SentimentIntensityAnalyzer()`. Δημιουργώντας μια συνάρτηση μπορούμε να κατατάξουμε τα tweets ανάλογα τη πολικότητα τους.

	Username	Date	Followers	Source	Location	Tweet	length	Tweet_without_stopwords	Translated	Polarity	label
0	CaeruleusCalva	4. August 2021	1088.0	Twitter for Android	Not So Ancient Greece	οπως λενε και οι καθηγητες ωρλ τα ανεμβολιαστα...	245	λενε καθηγητες ωρλ ανεμβολιαστα ζωα πεθαινουν ...	say teachers unvaccinated animals die coronavi...	0.000000	neutral
1	kambanellas	4. August 2021	316.0	Twitter for Android	Cyprus	δυστυχως τα δεδομενα δεν αλλαζουν ειτε σου αρε...	206	δυστυχως δεδομενα αλλαζουν ειτε αρεσει ειτε οχ...	unfortunately data changes whether you like it...	-0.500000	negative
2	em56464	4. August 2021	116.0	Twitter Web App	prefer not to say	χειροκροτουν τους πυροσβεστες μεχρι να σβησει ...	114	χειροκροτουν πυροσβεστες μεχρι σβησει φωτια κα...	applause firefighters until fire extinguished ...	0.000000	neutral
3	zetakosmatougk	4. August 2021	2955.0	Twitter for Android	greece	εχετε μισο λεπτο να σας μιλησω για το εμβολιο ...	113	εχετε μισο λεπτο μιλησω μητσοτακηκαθαρμα φωτι...	you have half a minute to talk mitsotaki fucki...	-0.166667	negative
4	PeterParker_84	4. August 2021	149.0	Twitter Web App	Timbuktu Κάτω Αχαίας	σκεψου ουτε οι ιδιοι δεν εμπιστευονται το εμβο...	55	σκεψου ιδιοι εμπιστευονται	they dont even trust the vaccine	0.000000	neutral

Στη συνέχεια ορίζουμε τις κλάσεις που θα καταταχτούν τα tweets σύμφωνα με τη πολικότητα τους με τον ίδιο τρόπο που κατατάξαμε για τη μέθοδο Textblob.

neg	neu	pos	compound	label
0.302	0.698	0.000	-0.8316	negative
0.303	0.570	0.127	-0.6395	negative
0.371	0.429	0.200	-0.3400	negative
0.275	0.725	0.000	-0.5849	negative
0.351	0.649	0.000	-0.4023	negative

Εξάγουμε πληροφορίες σχετικά με τη πολικότητα των δημοσιεύσεων εντοπίζοντας το πιο θετικό και πιο αρνητικό tweet της λίστας:

count	25339.000000
mean	140.053633
std	77.204873
min	1.000000
25%	75.000000
50%	126.000000
75%	208.000000
max	282.000000

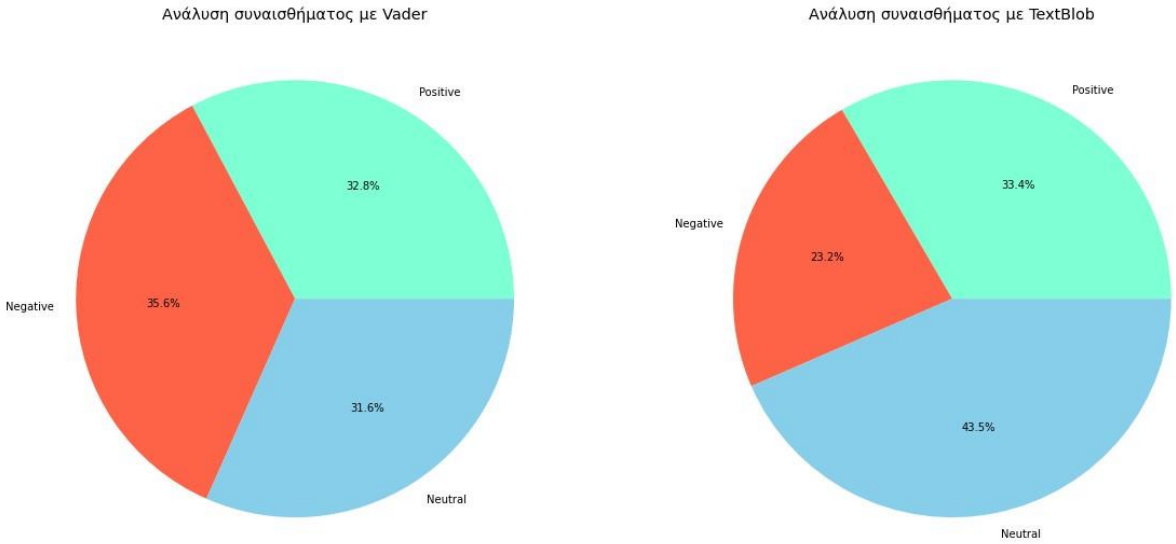
Βλέπουμε το πιο αρνητικό tweet το οποίο έχει πολικότητα -0.98. Το συγκεκριμένο tweet έχει κατηγοριοποιηθεί σωστά ως αρνητικό αφού το περιεχόμενο του είναι ιδιαίτερα αρνητικό.

Username	Date	Followers	Source	Location	Tweet	length	Tweet_without_stopwords	Tokens	compound	label
Dimos	25. July 2021	15.0	Twitter Web App	prefer not to say	απο το θανατηφορο εμβολιο φυσικααπο τι αλλο μπορει να πεθανε ενας χρονος απο τον ψευτοιο δεν πεθανε κανεις αλλωστεενας ψευτοιος δεν σκοτωνει αλλωστετο λεω γιατι εχουμε επαναληψη της προηγουμενης απατης του ψευτοιου ηναυτα λεγανε τοτε για πανδημια κουραφεξαλα	261	θανατηφορο φυσικααπο μπορει πεθανε ενας χρονος ψευτοιο πεθανε αλλωστεενας ψευτοιος σκοτωνει αλλωστετο λεω εχουμε επαναληψη προηγουμενης απατης ψευτοιου ηναυτα λεγανε πανδημια κουραφεξαλα	['θανατηφορο', 'φυσικααπο', 'μπορει', 'πεθανε', 'ενας', 'χρονος', 'ψευτοιο', 'πεθανε', 'αλλωστεενας', 'ψευτοιος', 'σκοτωνει', 'αλλωστετο', 'λεω', 'εχουμε', 'επαναληψη', 'προηγουμενης', 'απατης', 'ψευτοιου', 'ηναυτα', 'λεγανε', 'πανδημια', 'κουραφεξαλα']	-0.9847	negative

Όσον αναφορά το πιο θετικό tweet, βλέπουμε ότι είναι ένα tweet με πολλά emojis. Τα emojis παίζουν μεγάλο ρόλο στην έκφραση συναισθήματος των χρηστών των κοινωνικών δικτύων. Το λεξικό Vader μπορεί να κατηγοριοποιεί τα emojis με βάση τη πολικότητα του

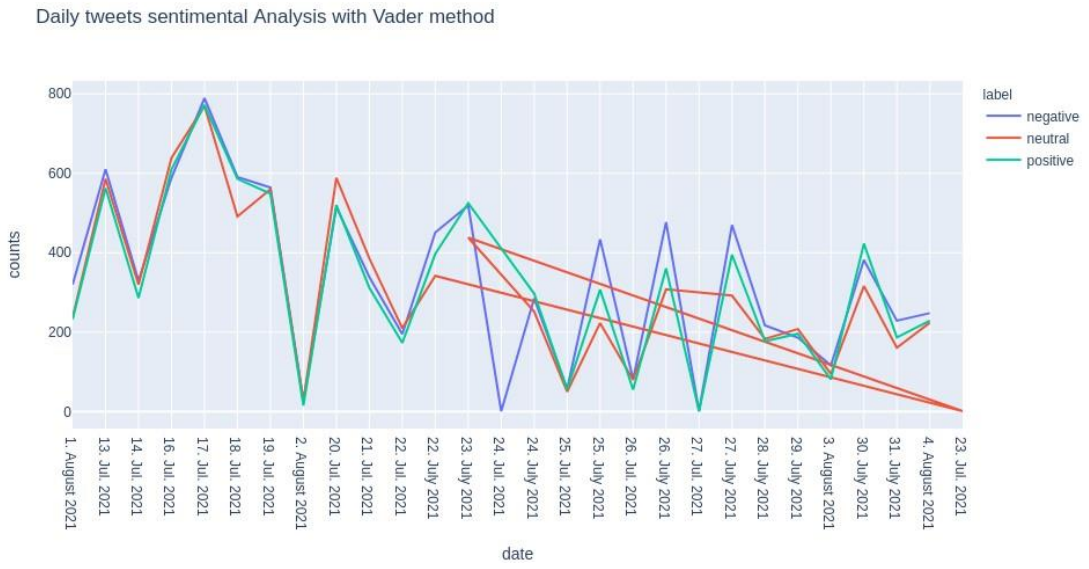
Username	Date	Followers	Source	Location	Tweet	length	Tweet_without_stopwords	Tokens	compound	label
Tsiribim tsiribom	27. July 2021	1658.0	Twitter Web App	prefer not to say	για να μην πουμε και αυτο το οσοι εχουν κανει το εμβολιο δεν νοσουν βαρια τελικα το εχει κανει το εμβολιο οχι τιποτα αλλο δεν εχει ανεβασει σελφι με το μπρατσο εξω🤣🤣🤣🤣🤣🤣🤣🤣🤣🤣🤣🤣🤣🤣🤣🤣🤣🤣🤣🤣🤣🤣🤣🤣🤣🤣	188	πουμε οσοι νοσουν βαρια τελικα οχι τιποτα ανεβασει σελφι μπρατσο εξω🤣🤣🤣🤣🤣🤣🤣🤣🤣🤣🤣🤣🤣🤣🤣🤣🤣🤣🤣🤣🤣🤣🤣🤣🤣🤣	['πουμε', 'οσοι', 'νοσουν', 'βαρια', 'τελικα', 'οχι', 'τιποτα', 'ανεβασει', 'σελφι', 'μπρατσο', 'εξω']	0.997	positive

Συγκρίνοντας τις δυο μεθόδους, βλέπουμε ότι έχουν διαφοροποιήσεις όσον αναφορά τα αρνητικά tweets, ωστόσο το ποσοστό θετικών tweets έχει ελάχιστη διαφορά. Αυτό οφείλεται στο γεγονός ότι το λεξικό Vader κατατάσσει αποδοτικότερα μερικές εκφράσεις argot καθώς και υβρεολόγια η οποία είναι πολύ διαδεδομένη στα κοινωνικά δίκτυα. Επομένως η μέθοδος Vader κατέταξε τις δημοσιεύσεις αποδοτικότερα σε σύγκριση με τη TextBlob (σχήμα 35). Ένας τρόπος να οπτικοποιήσουμε και να μετρήσουμε τα tweets ανά μέρα είναι ένα line plot (σχήμα 36, σχήμα 37).



Εικόνα 35: Αποτελέσματα ανάλυσης συναισθήματος

Sentiment	Vader	Textblob
negative	9022	11010
positive	8305	8461
neutral	8012	5868



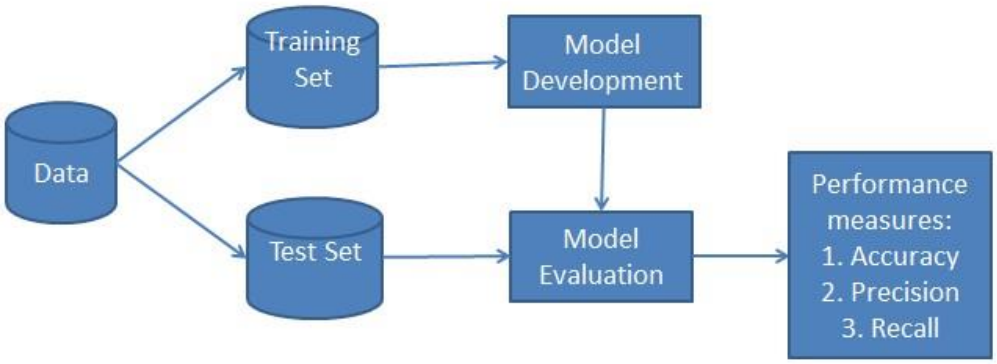
Εικόνα 36: Συναίσθημα ανά μέρα Vader

Εικόνα 37: Συναίσθημα ανά μέρα Textblob

4.10 Κατασκευή μοντέλου κατηγοριοποίησης συναισθήματος

Προκειμένου να κατασκευαστεί ένα αποδοτικό μοντέλο πρόβλεψης θα πρέπει να έχουμε ένα μεγάλο αριθμό βαθμολογημένων δεδομένων καθώς και τα δεδομένα αυτά να είναι όσο πιο ποιοτικά γίνεται. Στη περίπτωσή μας τα δεδομένα αποτελούνται από αληθινά δεδομένα κοινωνικών δικτύων, γεγονός που δυσκολεύει κατά πολύ την απόδοση του μοντέλου πρόβλεψης καθώς τα δεδομένα αυτά πολλές φορές υστερούν σε ποιότητα λόγω του θορύβου που περιέχουν. Επιπλέον τα δεδομένα αποτελούνται από κείμενα λίγων χαρακτήρων, κάτι που δυσκόλεψε πολύ το μοντέλο πρόβλεψης να πετύχει μεγάλο ποσοστό ακρίβειας σε πρώτη φάση. Το πρόβλημα επιλύθηκε όταν προστέθηκαν περισσότερες εγγραφές. Ο στόχος είναι να κατασκευαστεί ένα

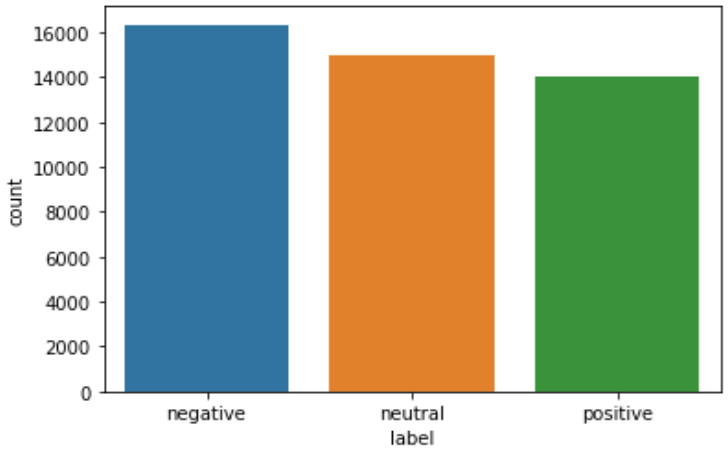
μοντέλο το οποίο να κατατάσσει όσο το δυνατόν σωστότερα τις κλάσεις τις οποίες θέλουμε να προβλέψουμε. Αυτό σημαίνει ότι αποζητάμε τουλάχιστον 75% ακρίβεια για να θεωρηθεί το μοντέλο αξιόπιστο. Για να πραγματοποιηθεί αυτό προστέθηκαν στο επεξεργασμένο αρχείο δεδομένων 20000 επιπλέον δημοσιεύσεις οι οποίες κατηγοριοποιήθηκαν με τη τεχνική Vader αφού πρώτα πραγματοποιήθηκε το στάδιο της προεπεξεργασίας. Το μοντέλο εκπαιδεύτηκε με τη χρήση τεσσάρων διαφορετικών αλγορίθμων κατηγοριοποίησης και έγινε deploy ο κατηγοριοποιητής με τη καλύτερη απόδοση. Τέλος έγινε αξιολόγηση του μοντέλου πρόβλεψης με την εφαρμογή των μεθόδων απόδοσης (performance measures). Η μεθοδολογία που ακολουθήθηκε είναι η εξής:



Εικόνα 38: Μεθοδολογία κατασκευής μοντέλου πρόβλεψης συναισθήματος

Αρχικά φορτώθηκε το διαμορφωμένο αρχείο δεδομένων το οποίο αποτελείται από 45.348 εγγραφές βαθμολογημένων δημοσιεύσεων με τη τεχνική Vader. Τα δεδομένα αποτελούνται από:

- 36% αρνητικά tweets
- 33% ουδέτερα tweets
- 30.9% θετικά tweets



Εικόνα 39: Συναισθηματική κατανομή τελικού αρχείου δεδομένων

Προκειμένου να χωρίσουμε τα δεδομένα σε δεδομένα εκπαίδευσης και δεδομένα ελέγχου, θα πρέπει αρχικά να εφαρμόσουμε τη τεχνική TF-IDF που επεξηγήσαμε σε προηγούμενο κεφάλαιο. Μετά τη διανυσματοποίηση των δεδομένων είναι έτοιμα να χωριστούν σε δεδομένα εκπαίδευσης και ελέγχου. Τα δεδομένα χωρίστηκαν σε 80% δεδομένα εκπαίδευσης και 20% δεδομένα ελέγχου. Αυτή η διαδικασία πραγματοποιήθηκε με τη μέθοδο train_test_split της βιβλιοθήκης scikit-learn της Python. Τα δεδομένα εκπαιδεύτηκαν με τέσσερεις διαφορετικούς αλγόριθμους με σκοπό να επιλεχτεί ο αποδοτικότερος ως προς τη πρόβλεψη και το χρόνος διεκπεραίωσης του. Τα δεδομένα διαμορφώθηκαν ως:

```
Train sample size     (36276, 45000
Test sample size      (9069, 45000)
```

4.10.1 Κατηγοριοποιητής Support Vector Machine

Βλέπουμε ότι τα δεδομένα εκπαίδευσης αποτελούνται από 36.276 εγγραφές ενώ τα δεδομένα ελέγχου από 9069 εγγραφές. Στη συνέχεια εκπαιδεύτηκε ένας κατηγοριοποιητής με τον αλγόριθμο Support Vector Machine. Αυτό πραγματοποιήθηκε στη γλώσσα Python με τη μέθοδο LinearSVC η οποία ενσωματώνει ένα κατηγοριοποιητή γραμμικού μοντέλου SVM. Κανονικά, ο αλγόριθμος Support Vector Machine είναι ένας αλγόριθμος κατηγοριοποίησης δυαδικών προβλημάτων. Ωστόσο, η βιβλιοθήκη Scikit-learn παρέχει τη δυνατότητα εκπαίδευσης ενός κατηγοριοποιητή SVM με πολλαπλές κλάσεις. Για την ταξινόμηση πολλαπλών τάξεων, η βιβλιοθήκη scikitlearn χρησιμοποιεί τη παράμετρο ‘ovr’ η οποία πραγματοποιεί τη διάσπαση του προβλήματος της πολυταξινόμησης σε πολλαπλά προβλήματα δυαδικής ταξινόμησης. Ο κατηγοριοποιητής SVM έφτασε το 79% ακρίβεια (accuracy) κάτι που το καθιστά ικανό έτσι ώστε να προβλέψει ικανοποιητικά μεγάλο μέρος των δημοσιεύσεων. Παρακάτω εξάγουμε τις στατιστικές πληροφορίες αξιολόγησης σχετικά με το μοντέλο που εκπαιδεύσαμε. Επιπλέον το μοντέλο δοκιμάστηκε σε φράσεις έτσι ώστε να δούμε την ικανότητα του να προβλέψει φράσεις που δεν ανήκουν στα δεδομένα που εκπαιδεύτηκε.

	precision	recall	f1-score	support
negative	0.79	0.78	0.79	3265
neutral	0.78	0.84	0.81	2992
positive	0.79	0.73	0.76	2812
accuracy			0.79	9069
Macro avg	0.79	0.78	0.78	9069
weighted avg	0.79	0.79	0.79	9069

```
sample = ["χαλια μερα σημερα"] sample =
tfidf.transform(sample).toarray()
sentiment = svm.predict(sample) print('Η
πρόταση είναι',":",sentiment)

Η πρόταση είναι : ['negative']

sample = ["τελεια μερα σημερα"] sample =
tfidf.transform(sample).toarray()
sentiment = svm.predict(sample) print('Η
πρόταση είναι',":",sentiment)
```

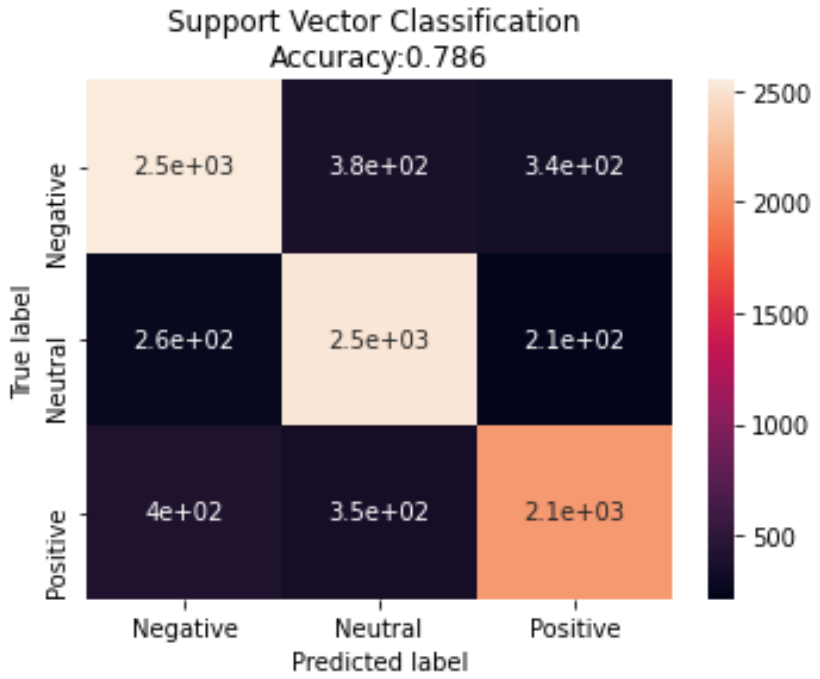
```
Η πρόταση είναι : ['positive']
```

```
sample = ["μετρια μερα σημερα"] sample =
tfidf.transform(sample).toarray()
sentiment = svm.predict(sample) print('Η
πρόταση είναι',":",sentiment)
```

```
Η πρόταση είναι : ['neutral']
```

Βλέπουμε λοιπόν ότι το μοντέλο προέβλεψε σωστά και τις 3 προτάσεις που του δόθηκαν. Μπόρεσε να ξεχωρίσει τις διαφορές ανάμεσα στις λέξεις «τέλεια», «χάλια», «μέτρια» και κατηγοριοποίησε τις προτάσεις στη σωστά κλάση. Επιπλέον εξάγουμε τα αποτελέσματα του πίνακα σύγχυσης (confusion matrix)

```
[[2548,  377,  340],
 [ 265, 2514,  213],
 [ 402,  346, 2064]]
```



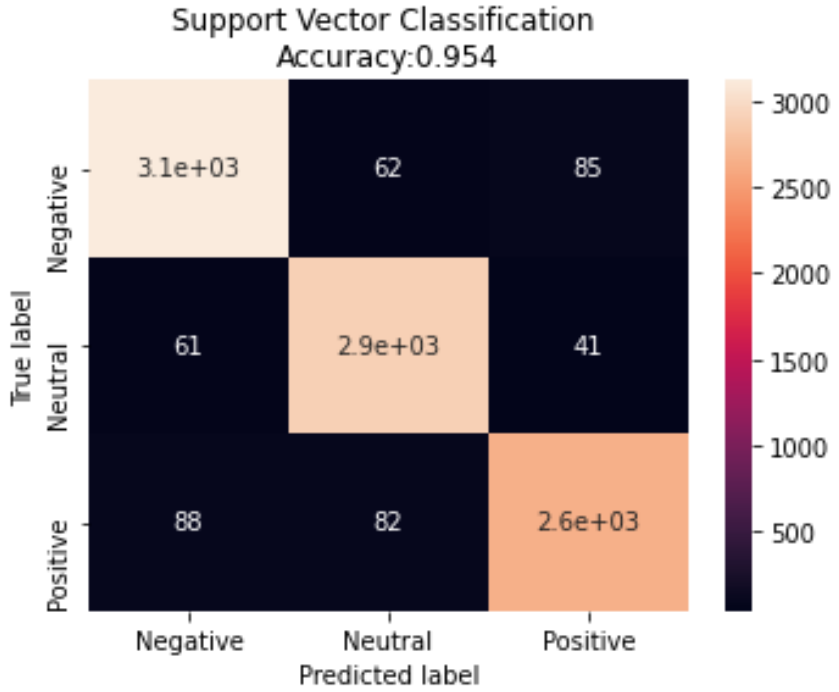
Εικόνα 40: Confusion Matrix Support Vector Machine

Προκειμένου να φορτώσουμε το κατηγοριοποιητή σε μελλοντικό αρχείο θα το αποθηκεύσουμε σαν “pickle” αρχείο. Η αποθήκευση θα γίνει μέσω της βιβλιοθήκης pickle η οποία κάνει σειριοποίηση των δεδομένων πριν τα μετατρέψει σε bytes. Η αποθήκευση αρχείων σε μορφή pickle είναι γρήγορος και αποδοτικός τρόπος αποθήκευσης, άμεση αποθήκευση δεδομένων από σύνθετες δομές (λεξικά, λίστες αντικειμένων, κλπ). Αποθηκεύουμε επίσης το μοντέλο TF-IDF γιατί θα το χρειαστούμε στη συνέχεια για το σύστημα πρόβλεψης συναισθήματος.

4.10.2 Support Vector Machine με OneVsRestClassifier

Φτιάχνουμε ένα νέο κατηγοριοποιητή που βασίζεται στον αλγόριθμο Support Vector Machine όμως αυτή τη φορά ενσωματώνουμε τη τεχνική one-vs-rest της sckit-learn. Αυτή η τεχνική συνίσταται στην τοποθέτηση ενός δυαδικού ταξινομητή ανά κλάση. Στη συνέχεια εκπαιδεύεται ο κάθε κατηγοριοποιητής ξεχωριστά για όλες τις κλάσεις. Ο κατηγοριοποιητής με την εφαρμογή της μεθόδου έφτασε 95% ακρίβεια.

	precision	recall	f1-score	support
negative	0.95	0.95	0.95	3265
neutral	0.95	0.97	0.96	2992
positive	0.95	0.94	0.95	2812
accuracy			0.95	9069
Macro avg	0.95	0.95	0.95	9069
weighted avg	0.95	0.95	0.95	9069



Εικόνα 41: Confusion Matrix με τεχνική One Vs Rest

Δοκιμάζουμε το κατηγοριοποιητή σε κάποια τυχαία δείγματα έτσι ώστε να δούμε αν προβλέπει αποδοτικά. Παρατηρούμε ότι κατηγοριοποίησε σωστά και τα δύο δείγματα που του δώθηκαν.

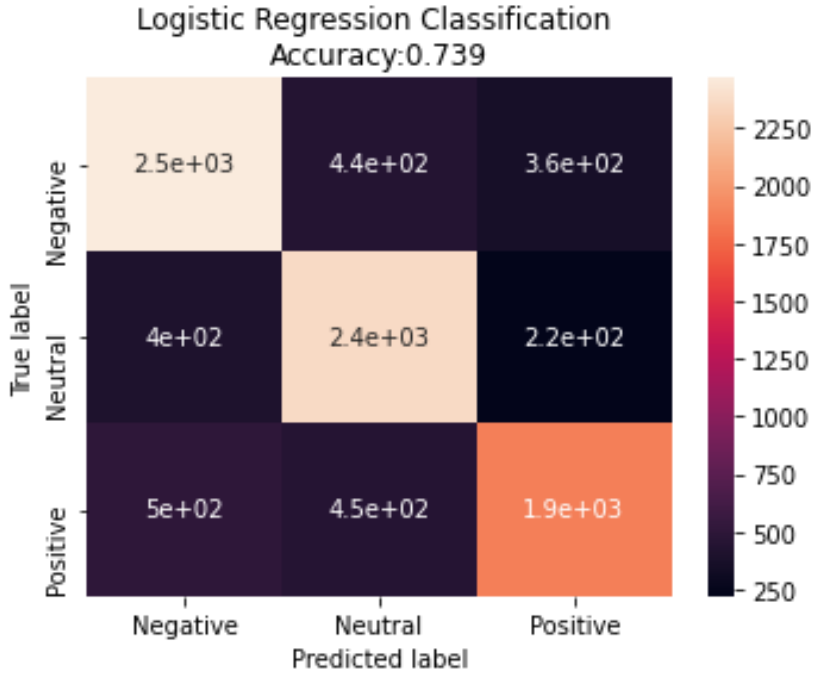
```
sample = ["Πήρα το τεστ για το σπίτι να το εχω άμεσα διαθεσιμο. Έχω ηδη
υποβληθεί σε 3 τεστ (λόγω εργασίας σε νοσοκομείο) τα οποία έδειξαν ακριβώς
τα ίδια αποτελέσμα τα με τη μεθοδο RT-PCR και τα υπόλοιπα rapid test που
κάνω εκεί."] sample = tfidf.transform(sample).toarray() sentiment =
ovsrc.predict(sample) print('Η πρόταση είναι',":",sentiment)

sample = ["ΜΕΤΡΙΑ ΜΕΡΑ ΣΗΜΕΡΑ"] sample =
tfidf.transform(sample).toarray()
sentiment = svm.predict(sample) print('Η
πρόταση είναι',":",sentiment)
```

4.10.3 Κατηγοριοποιητής Multinomial Logistic Regression

Εφόσον το πρόβλημά μας αφορά μια κατηγοριοποίηση με τρεις κλάσεις, θα πρέπει να χρησιμοποιηθεί η πολυωνυμική λογιστική παλινδρόμηση. Η πολυωνυμική παλινδρόμηση είναι μια τροποποιημένη έκδοση λογιστικής παλινδρόμησης που προβλέπει μια πολυωνυμική πιθανότητα (δηλαδή περισσότερες από δύο κατηγορίες) για κάθε παράδειγμα εισόδου. Ο κατηγοριοποιητής έφτασε το 73% ακρίβεια δηλαδή πιο χαμηλή ακρίβεια από το κατηγοριοποιητή Support Vector Machine που εκπαιδεύσαμε πριν.

	precision	recall	f1-score	support
negative	0.73	0.76	0.74	3265
neutral	0.73	0.79	0.76	2992
positive	0.76	0.66	0.71	2812
accuracy			0.74	9069
Macro avg	0.74	0.74	0.74	9069
weighted avg	0.74	0.74	0.74	9069

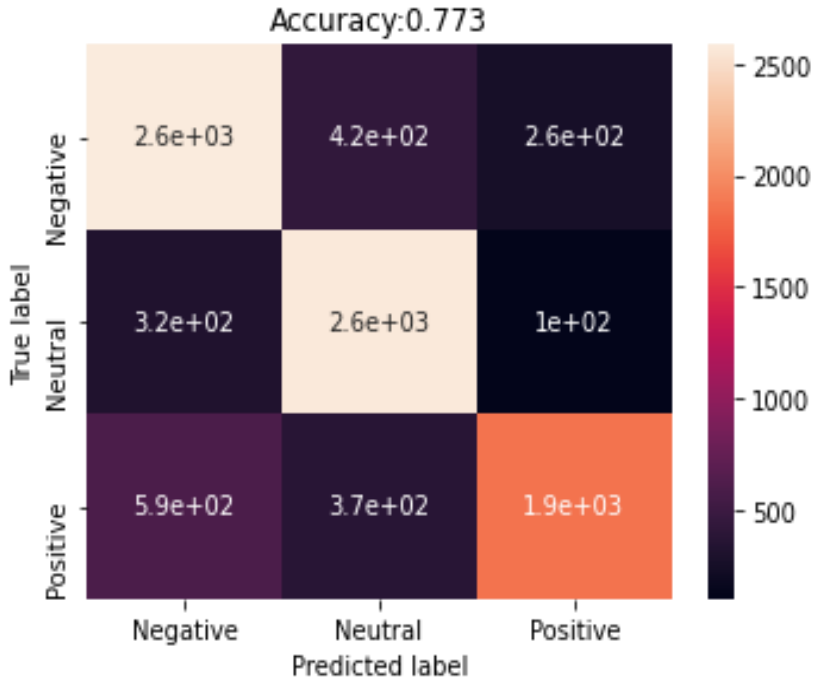


Εικόνα 42: Confusion Matrix Πολυωνυμικής Λογιστικής Παλινδρόμησης

4.10.4 Κατηγοριοποιητής Random Forest

Ο αλγόριθμος random forest αποτελεί ένα είδος μάθησης που λειτουργεί με την κατασκευή ενός πλήθους δέντρων απόφασης κατά το χρόνο εκπαίδευσης. Ο συγκεκριμένος αλγόριθμος είναι πολύ αποδοτικός σε μεγάλο αριθμό ανεξάρτητων μεταβλητών και έχει σχετικά μικρό απαιτούμενο χρόνο εκτέλεσης. Ο συγκεκριμένος αλγόριθμος αποδείχθηκε ιδιαίτερα αποτελεσματικός φτάνοντας το 77% ακρίβεια

	precision	recall	f1-score	support
negative	0.74	0.79	0.76	3265
neutral	0.76	0.86	0.81	2992
positive	0.84	0.66	0.74	2812
accuracy			0.77	9069
Macro avg	0.78	0.77	0.77	9069
weighted avg	0.78	0.77	0.77	9069

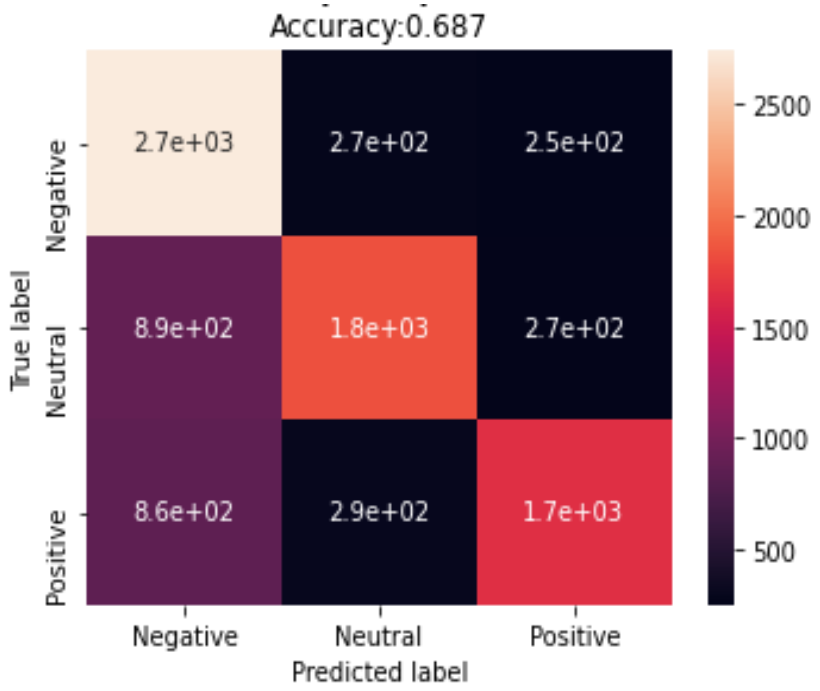


Εικόνα 43: Confusion Matrix με Random Forest

4.10.5 Κατηγοριοποιητής Multinomial Naive Bayes

Ο πολυωνυμικός αλγόριθμος Naive Bayes είναι μια στατιστική μέθοδος εκμάθησης που χρησιμοποιείται κυρίως στην Επεξεργασία Φυσικής Γλώσσας (NLP). Ο αλγόριθμος βασίζεται στο θεώρημα Bayes και προβλέπει την ετικέτα ενός κειμένου, όπως ένα email (Spam/ham) ή άρθρο εφημερίδας. Ο πολυωνυμικός Bayes έφτασε μόνο 68% ακρίβεια επομένως δεν είναι ο ιδανικός για τη κατηγοριοποίηση που ζητάμε στο συγκεκριμένο πρόβλημα. Παραμένει όμως ένας πολύ απλός στη κατανόηση αλγόριθμος με πολύ καλά αποτελέσματα σε πληθώρα προβλημάτων κατηγοριοποίησης.

	precision	recall	f1-score	support
negative	0.61	0.84	0.71	3265
neutral	0.76	0.61	0.68	2992
positive	0.76	0.59	0.66	2812
accuracy			0.69	9069
Macro avg	0.71	0.68	0.68	9069
weighted avg	0.71	0.69	0.68	9069



Εικόνα 44: Confusion Matrix Naive Bayes

4.11 Εφαρμογή του μοντέλου πρόβλεψης

Προκειμένου να δοκιμάσουμε το μοντέλο πρόβλεψης που κατασκευάσαμε, θα το εφαρμόσουμε σε νέα δεδομένα τα οποία θα εξάγουμε μέσω του Twitter API. Οι αλγόριθμοι που τρέξαμε και εκπαιδεύσαμε τα μοντέλα πραγματοποιήθηκαν μέσω των συναρτήσεων της βιβλιοθήκης scikitlearn της Python η οποία είναι μια βιβλιοθήκη μηχανικής μάθησης. Διαθέτει διάφορους αλγορίθμους ταξινόμησης, παλινδρόμησης και ομαδοποίησης και έχει σχεδιαστεί για να λειτουργεί με τις αριθμητικές και επιστημονικές βιβλιοθήκες της Python NumPy και SciPy.

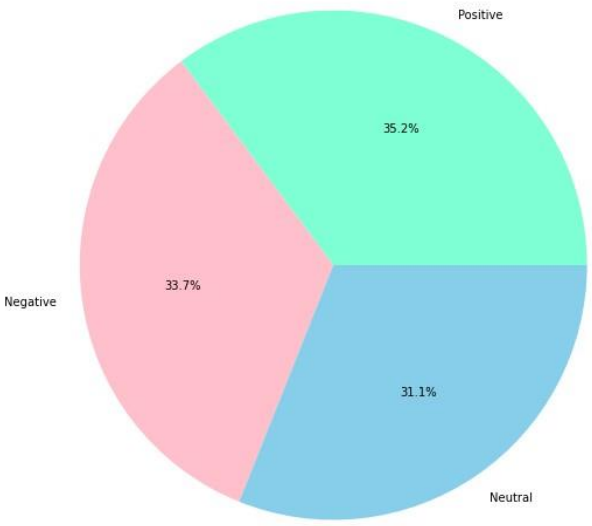
Δημιουργείται ένα καινούργιο αρχείο δεδομένων με βαθμολογημένα Tweets. Το μοντέλο προβλέπει αρκετά αποδοτικά φτάνοντας το 95% ακρίβεια με τη μέθοδο OneVsRest της βιβλιοθήκης scikit-learn η οποία εφαρμόζεται σε προβλήματα που αφορούν πολλαπλές κλάσεις. Η τεχνική αυτή εφαρμόζεται στον αλγόριθμο που εκπαιδεύτηκε το μοντέλο (Support Vector Machine). Η ακρίβεια του αλγορίθμου πριν την εφαρμογή της τεχνικής έφτασε το 79%.

Ο χρόνος καθώς και η απόδοση του συγκεκριμένου αλγορίθμου ήταν το κριτήριο επιλογής του. Το μοντέλο έχει εκπαιδευτεί σε ελληνικά tweets στην ελληνική γλώσσα. Οι δημοσιεύσεις αυτές αποτελούν δεδομένα με πολύ θόρυβο το οποία επικεντρώνονται στις δημοσιεύσεις που αφορούν το εμβόλιο του κορονοϊού. Τα συναισθήματα που αποτυπώνονται σχετικά με τη συγκεκριμένη θεματική, αλλάζουν σύμφωνα με τις εξελίξεις που γίνονται καθημερινά. Επομένως προκειμένου το μοντέλο στο μέλλον να προβλέπει και άλλου είδους δεδομένα διαφόρων θεματικών, θα πρέπει να προστεθούν επιπλέον δημοσιεύσεις που αφορούν και άλλα θέματα και να επαναλάβουμε τη διαδικασία της εκπαίδευσης.

Το μοντέλο προβλέπει επαρκώς τα αρνητικά και ουδέτερα tweets, όμως δυστυχώς λόγο της έντονης έλλειψης σε θετικά tweets, το μοντέλο υστερεί στη πρόβλεψη των θετικών. Αυτό μπορεί να διορθωθεί στο μέλλον προσθέτοντας επιπλέον θετικά tweets στα δεδομένα. Ένα ακόμα πρόβλημα του μοντέλου είναι η πρόβλεψη στις εκφράσεις αργκό καθώς και η ειρωνεία. Το αρχείο test_model.py εξάγει δεδομένα από το Twitter μέσω του Twitter API, σύμφωνα με τη λέξη κλειδί

που θα ορίσουμε, τα οποία κατηγοριοποιεί συναισθηματικά με το μοντέλο πρόβλεψης που κατασκευάστηκε και στη συνέχεια τα αποθηκεύει σε ένα αρχείο csv.

name	created_at	sentiment	text
GeniaTou	08/23/21	neutral	επιμενει στη συνδεση του εμβολιου με το ισχαιμικο επεισοδιο χαρδαλιας εμβολιο
GeniaTou	08/23/21	positive	που δεν δεχονται να κανουν το εμβολιο.το σου θα σε συμβουλευα να το κατεβασεις γιατι κινδυνευεις να βρεθεις αντιμετωπος με τον πραγματικο εισαγγελεα στην ελλαδα για διασπορα ψευδων ειδησεων φιλικα
GeniaTou	08/23/21	neutral	δεν χρειαζεται καν σημειωσεις τα γραφω ολα μεσω στο τσιπακι που μου φορεσαν με το εμβολιο 🤣
GeniaTou	08/23/21	neutral	κορονοιος ποιοι και ποτε κανουν την τριτη δοση αποκλειστικα με εμβολιο
GeniaTou	08/23/21	negative	στις ηπα με την πληρη εγκριση του εγινε υποχρεωτικος ο εμβολιασμος στον στρατο ας το κανουμε και εμεις και επειτα επιστρατευουμε τους υγειονομικους. ουτε γατα ουτε ζημια ααα και το μισθο τους ολοι εμβολιασμος εμβολιο εμβολιασμενοι
GeniaTou	08/23/21	negative	ειτε εμβολιο ειτε οχι παλι με μασκες καταρρεει ο μυθος της χουντοκυβερνησης για προνομια
GeniaTou	08/23/21	negative	ρε εσεις δεν ηξερα οτι οσοι καναμε το εμβολιο θα πεθανουμε σε δυο χρονια. αποψε το εμαθα κι αυτο 🤦🤦🤦🤦🤦να βαλουμε κατω τις ημερομηνιες να ξερουμε σε ποιες κηδειες θα παμε κι ποιοι θα ερθουν στις δικες μας.
GeniaTou	08/23/21	positive	λες να μαννεψαμε που το εμβολιο μιχαλη ισως μπορει και να φταιει το ντουνλοουντ της τελευταιας δεσμης δονησεων απο το συμπαν να μεν μας εκατσε καλα.
GeniaTou	08/23/21	negative	ουτε ειχαν στην πανδημια κυβερνηση που δεν ηξερε πως λειτουργουν τα εμβολια κι εδινε λαθος εντυπωσεις που εκαναν τον κοσμο να τα φοβαται. δεν εκανε διαφημισεις του στυλ "κανουμε το εμβολιο για να βγαλουμε αμεσως τις μασκες και να αρχισουμε αγκαλιες και φιλια με τους δικους μας".
GeniaTou	08/23/21	negative	κι εγω εχω κανει το εμβολιο και η γυναικα μου χωρις υποχρεωση και μετα απο πολλη σκεψη νιωθοντας οτι ρισκαρω η ζυγαρια εγειρε υπερ του εμβολιου το παιδι μου ομως δεν θα το κανει κι ας τα σκαω στα μοριακα μονο οταν μου υπογραψουν οτι δεν θα εχει κανενα προβλημα



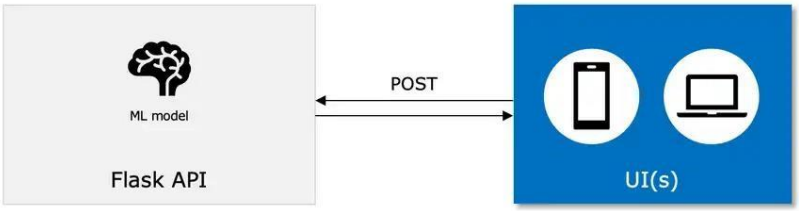
Εικόνα 45: Αποτελέσματα ανάλυσης συναισθήματος σε νέα δεδομένα

5. Εφαρμογή πρόβλεψης συναισθήματος με Flask

Το Flask είναι ένα web framework το οποίο μας επιτρέπει να δημιουργούμε APIs, backend applications σε python και να κάνουμε deploy μοντέλα πρόβλεψης. Οι εφαρμογές αυτές μπορούν να βασίζονται σε κάποιο μοντέλο πρόβλεψης έτσι ώστε ο χρήστης να μπορεί να κάνει κάποια πρόβλεψη. Μέσω του Flask λοιπόν δημιουργήθηκε μια απλή εφαρμογή ανάλυσης συναισθήματος έτσι ώστε να δοκιμαστεί το μοντέλο πρόβλεψης το οποίο κατασκευάσαμε. Η εφαρμογή προβλέπει

το συναίσθημα της φράσης σε πραγματικό χρόνο στέλνοντας μια μέθοδο POST στην εφαρμογή η οποία επιστρέφει το αποτέλεσμα της πρόβλεψης. Το front-end της εφαρμογής γράφτηκε με HTML και CSS.

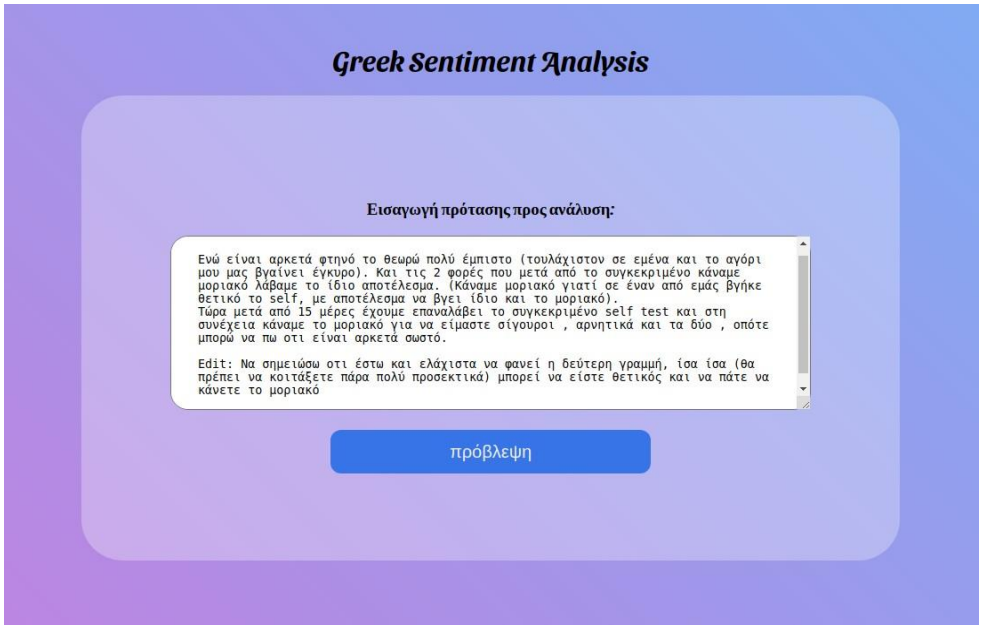
Το μοντέλο δοκιμάστηκε σε ελληνικές αξιολογήσεις της πλατφόρμας skroutz που αφορούν τα self test του κορονοϊού. Το μοντέλο λειτουργεί αποδοτικότερα για μεγάλες προτάσεις και υστερεί ως προς τη κατάταξη συναισθήματος σε εκφράσεις που περιέχουν υβρεολόγια. Η εφαρμογή παίρνει σαν είσοδο μια πρόταση την οποία πληκτρολογεί ο χρήστης και στη συνέχεια την αναλύει συναισθηματικά και τη κατατάσσει στις τρεις κλάσεις σύμφωνα με τη σημασιολογική της αξία Στη συνέχεια επιστρέφει το αποτέλεσμα της πρότασης. Η διεπαφή χρήστη (UI) είναι πολύ απλή όμως είναι αποτελεσματική στο να δούμε κατά πόσο το μοντέλο πρόβλεψης μας δουλεύει ικανοποιητικά.



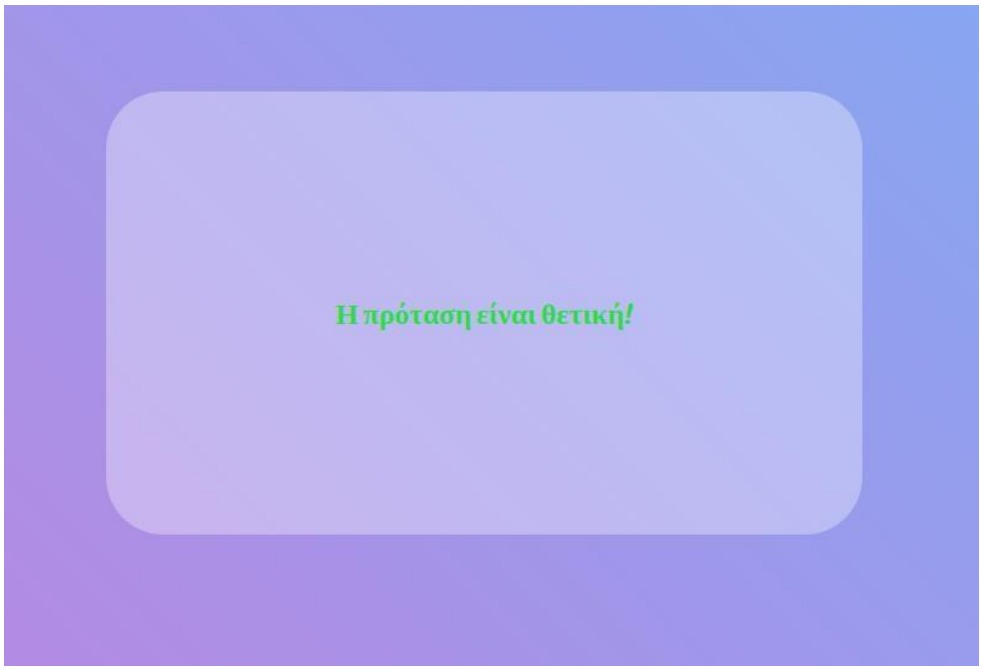
Εικόνα 46: Εφαρμογή Flask

5.1 Συναισθηματική κατάταξη αξιολογήσεων

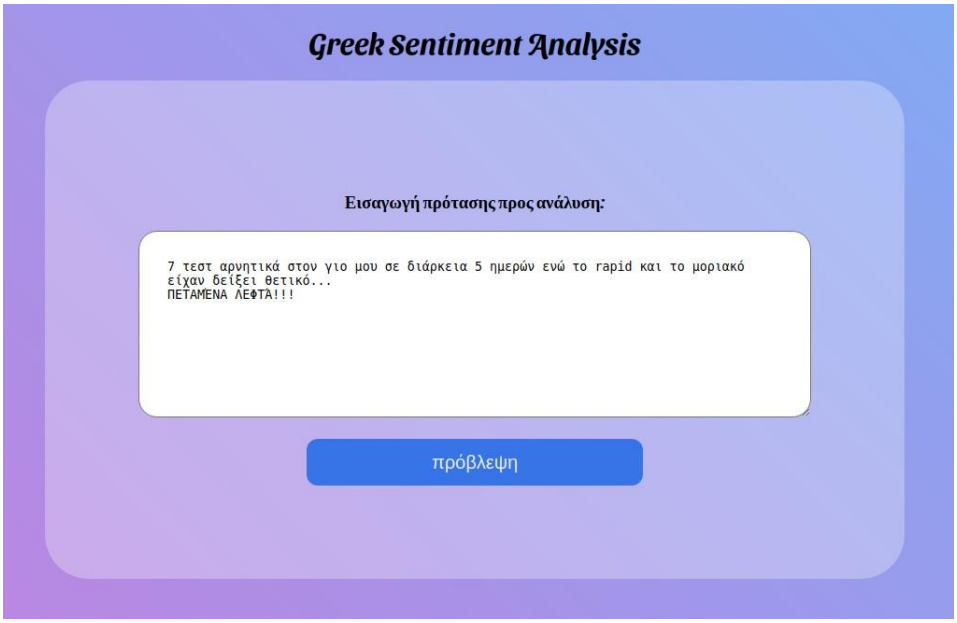
Προκειμένου να αξιολογήσουμε το μοντέλο, επιλέχτηκαν τυχαία κάποιες θετικές και κάποιες αρνητικές κριτικές οι οποίες είναι βαθμολογημένες με 1 αστέρι και 5 αντίστοιχα. Τα αποτελέσματα παρατίθενται παρακάτω:



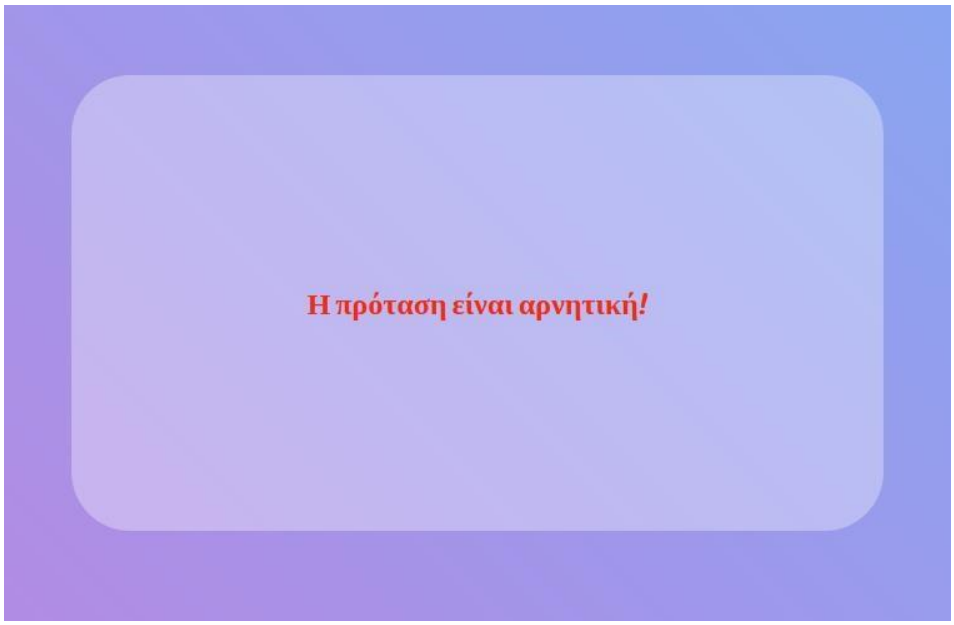
Εικόνα 47: Παράδειγμα εφαρμογής 1



Εικόνα 48: Επιστροφή θετικού αποτελέσματος



Εικόνα 49: Παράδειγμα αρνητικής πρότασης



Εικόνα 50: Κατηγοριοποίηση αρνητικού αποτελέσματος

Όπως βλέπουμε, η εφαρμογή είναι σε θέση να προβλέψει σωστά ένα μεγάλο ποσοστό των προτάσεων που της δίνονται. Μια αδυναμία του αλγορίθμου είναι να κατατάξει σωστά λέξεις που δεν έχει ξανασυναντήσει όπως και greeklish η άλλες γλώσσες πέρα των ελληνικών. Αυτό θα διορθωθεί εκπαιδεύοντας το μοντέλο σε περισσότερα δεδομένα. Σημειώνεται ότι το UI της εφαρμογής είναι πολύ απλό καθώς ο σκοπός είναι η ανάδειξη της χρηστικότητας της. Υπάρχουν πολλές βελτιώσεις που δύναται να πραγματοποιηθούν ως προς τη χρηστικότητα της εφαρμογής αλλά και τελειοποίησης του μοντέλου πρόβλεψης εκπαιδεύοντας το σε μεγαλύτερο εύρος δεδομένων.

6. Συμπεράσματα - Προτάσεις για μελλοντική έρευνα

Στη παρούσα διπλωματική συλλέχτηκαν πάνω από 50.000 ελληνικές δημοσιεύσεις (tweets). Τα δεδομένα συλλέχτηκαν σε πραγματικό χρόνο μέσω του Streaming API του κοινωνικού δικτύου Twitter. Οι δημοσιεύσεις που συλλέχτηκαν αφορούσαν το εμβόλιο του κορονοϊού με σκοπό να εξαχθεί κάποιο συμπέρασμα σχετικά με την αντιμετώπιση των Ελλήνων πάνω σε αυτό το θέμα. Σε γενικές γραμμές, υπήρξε μεγάλος διχασμός στο συγκεκριμένο θέμα όπως και πλήθος αναληθών και επαναλαμβανόμενων ειδήσεων (fake news). Τα δεδομένα συλλέχτηκαν και αποθηκεύτηκαν σε csv αρχεία και στη συνέχεια ακολούθησε η διαδικασία της προεπεξεργασίας όπου αφαιρέθηκαν οι επαναλαμβανόμενες δημοσιεύσεις και κάθε είδους θορύβου που μπορεί να υπήρχε.

Τα δεδομένα μεταφέρθηκαν στην αγγλική γλώσσα και στη συνέχεια προκειμένου να κατηγοριοποιηθούν συναισθηματικά σε τρεις κατηγορίες (θετικό, αρνητικό, ουδέτερο) χρησιμοποιήθηκαν οι τεχνικές Vader και Textblob μέσω των βιβλιοθηκών τους στη γλώσσα Python. Από τις δυο αυτές τεχνικές παρατηρήθηκε μεγαλύτερη ακρίβεια ως προς τα δεδομένα προερχόμενα από κοινωνικά δίκτυα στη μέθοδο Vader. Τα κατηγοριοποιημένα συναισθηματικά δεδομένα στη συνέχεια χρησιμοποιήθηκαν για την εκπαίδευση ενός κατηγοριοποιητή ανάλυσης συναισθήματος. Εφαρμόστηκε η τεχνική TF-IDF η οποία αποτελεί μια αριθμητική στατιστική που προορίζεται να αντικατοπτρίζει τη σημασία μιας λέξης για ένα έγγραφο σε μια συλλογή δεδομένων. Στη συνέχεια τα δεδομένα χωρίστηκαν σε δεδομένα εκπαίδευσης και δεδομένα

ελέγχου και εκπαιδεύτηκαν με τέσσερεις διαφορετικούς αλγόριθμους με σκοπό να βρεθεί ο πιο αποδοτικός ως προς το αποτέλεσμα αλλά και ως προς τη ταχύτητα. Οι αλγόριθμοι που δοκιμάστηκαν είναι:

1. Μηχανή διανυσματικής υποστήριξης (Support vector machine)
2. Πολυωνυμική λογιστική Παλινδρόμηση (Multinomial Logistic Regression)
3. Random Forest
4. Πολυωνυμικός Αφελής Bayes (Multinomial Naive Bayes)

Ο αλγόριθμος Support vector machine αποδείχτηκε ο πιο αποδοτικός ως προς την ακρίβεια των αποτελεσμάτων φτάνοντας ακρίβεια σχεδόν 80% και 95% με την εφαρμογή της τεχνικής Onevs-the-rest (OvR) αλλά και ως προς τη ταχύτητα οπότε επιλέχτηκε για τη κατασκευή του κατηγοριοποιητή συναισθήματος. Το μοντέλο δοκιμάστηκε σε ένα καινούργιο αρχείο δεδομένων από το κοινωνικό δίκτυο Twitter και τα βαθμολόγησε ανάλογα με το συναίσθημα τους. Τέλος, φτιάχτηκε μια απλή εφαρμογή συναισθηματικής ανάλυσης με το web framework Flask η οποία κατατάσσει συναισθηματικά προτάσεις στην Ελληνική γλώσσα. Η εφαρμογή δοκιμάστηκε και κατέταξε σωστά ένα μεγάλο ποσοστό προτάσεων ωστόσο υστερεί ως προς τη κατάταξη προτάσεων που περιέχουν υβρεολόγια και λέξεις οι οποίες δεν περιέχονται στα δεδομένα εκπαίδευσης.

Όσον αναφορά τον τομέα της επεξεργασίας της φυσικής γλώσσας στην Ελληνική γλώσσα, δυστυχώς τα εργαλεία που υπάρχουν είναι ελάχιστα. Επομένως, μια καλή ιδέα για μελλοντική έρευνα είναι η κατασκευή μιας βιβλιοθήκης η μετάφραση κάποιας υπάρχουσας (π.χ. Vader) για τη γλώσσα Python η οποία να είναι σε θέση να κατατάσσει συναισθηματικά ελληνικό κείμενο. Όσον αναφορά το μοντέλο πρόβλεψης, υπάρχει περιθώριο βελτίωσης και αυτό αφορά την εκπαίδευση του μοντέλου σε περισσότερα δεδομένα που αφορούν και άλλα θεματικά πεδία. Τα δεδομένα που χρησιμοποιήθηκαν προέρχονται από ένα κοινωνικό δίκτυο και είναι κατά συνέπεια περιέχουν ορθογραφικά λάθη καθώς και νεολογισμούς επομένως πρέπει να ληφθεί και αυτό υπόψιν για τη μελλοντική βελτίωση του μοντέλου. Όσον αναφορά την εφαρμογή πρόβλεψης συναισθήματος μια πρόταση για μελλοντική βελτίωση αποτελεί την επικοινωνία της εφαρμογής με το Twitter API έτσι ώστε να αντλεί δεδομένα σε πραγματικό χρόνο μέσω κάποιας λέξης κλειδί που θα ορίζει ο χρήστης. Επιπλέον μπορεί να μετατραπεί σε εφαρμογή για Windows, Linux, Mac.

Συνοψίζοντας, η ελληνική είναι μια γλώσσα που αποτελεί πρόκληση για το τομέα της Επεξεργασίας της Φυσικής Γλώσσας, μιας και είναι μια γλώσσα με πολύ πλούσιο λεξιλόγιο και σύνθετη γραμματική. Επομένως είναι πολλά αυτά τα οποία πρέπει να ληφθούν υπόψιν. Τα μοντέλα μηχανικής μάθησης είναι ένας αποτελεσματικός τρόπος πρόβλεψης συναισθήματος αρκεί τα δεδομένα να είναι όσο πιο “ποιοτικά” γίνεται καθώς και να δοθεί ιδιαίτερη σημασία στο κομμάτι της προεπεξεργασίας των δεδομένων. Τέλος τα αρχεία δεδομένων θα πρέπει να έχουν όσο το δυνατόν περισσότερες εγγραφές έτσι ώστε να επαρκούν για την εκπαίδευση και τον έλεγχο του μοντέλου.

Βιβλιογραφία

1. Tinati, Ramine & Carr, Les & Hall, Wendy & Bentwood, Johnny. (2012). Scale Free: Twitter's Retweet Network Structure.
2. Κύρκος, Ε. 2015. Εξόρυξη Γνώσης από Δεδομένα. [Κεφάλαιο Συγγράμματος]. Στο Κύρκος, Ε. 2015. Επιχειρηματική ευφυΐα και εξόρυξη δεδομένων. [ηλεκτρ. βιβλ.] Αθήνα: Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. κεφ. 6.
3. D. Beleveslis, C. Tjortjis, D. Psaradelis and D. Nikoglou, "A Hybrid Method for Sentiment Analysis of Election Related Tweets," 2019 4th South-East Europe Design Automation, Computer Engineering, Computer Networks and Social Media Conference (SEEDA-CECNSM), 2019, pp. 1-6, doi: 10.1109/SEEDA-CECNSM.2019.8908289.
4. Βερύκιος, Β., Καγκλής, Β., Σταυρόπουλος, Η., 2015. Η επιστήμη των δεδομένων μέσα από τη γλώσσα R. [ηλεκτρ. βιβλ.] Αθήνα: Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών.
5. ΕΞΟΡΥΞΗ ΔΕΔΟΜΕΝΩΝ (DATA MINING) ΚΑΙ ΚΑΤΗΓΟΡΙΚΑ ΔΕΔΟΜΕΝΑ Γεράσιμος Ε. Σταυλιώτης
6. Kydros, Dimitrios & Argyropoulou, Maria & Vrana, Vasiliki. (2021). A Content and Sentiment Analysis of Greek Tweets during the Pandemic. Sustainability. 13. 6150. 10.3390/su13116150.
7. Spatiotis, Nikolaos & Mporas, Iosif & Paraskevas, Dr & Perikos, Isidoros. (2016). Sentiment Analysis for the Greek Language. 1-4. 10.1145/3003733.3003769.
8. Bahrawi, Bahrawi. (2019). SENTIMENT ANALYSIS USING RANDOM FOREST ALGORITHM ONLINE SOCIAL MEDIA BASED. 2. h.29-33.
9. Matthew A Russell & Mikhail Klassen, "Mining the Social Web: Data Mining Facebook, Twitter, LinkedIn, Instagram, GitHub, and More" 3rd Edition. O'Reilly, 2019
10. Pang, B. et Lee, L., Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval, 2(1-2), 2008, p1-135
11. Liu B. Sentiment analysis and opinion mining. Synthesis Lectures on Human Language Technologies. 2012;5(1):1–167. 2. Liu B. Sentiment Analysis: Mining Opinions, Sentiments, and Emotions. Cambridge University Press; 2015
12. Κύρκος, Ε., 2015. Επιχειρηματική ευφυΐα και εξόρυξη δεδομένων. [ηλεκτρ. βιβλ.] Αθήνα: Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών.
13. Segaran, T., Programming Collective Intelligence. Sebastopol: O'Reilly Media, Inc, 2007
14. Taboada M, Brooke J, Tofiloski M, Voll K, Stede M (2011) Lexicon-based methods for sentiment analysis. Comput Linguist J 267–307
15. Κύρκος, Ε. 2015. Επιχειρηματική ευφυΐα και εξόρυξη δεδομένων. [ηλεκτρ. βιβλ.] Αθήνα: Σύνδεσμος Ελληνικών Ακαδημαϊκών Βιβλιοθηκών. κεφ 6-7.
16. Van't Ende, R., Sentiment analyse van gazouillis: de correlatie tussen de merkbeoodeling op Twitter en de werkelijke merbeoordeling van consumenten. Master thesis, Tilburg University, 2013
17. Karageorgou, Ioanna & Liakos, Panagiotis & Delis, Alex. (2021). Just-in-Time Sentiment Analysis for Streamed Data in Greek. 10.1007/978-3-030-73203-5_19.
18. Chatterjee, Deb & Mukherjee, Anirban & Mukhopadhyay, Sabyasachi & Panday, Mrityunjoy & Panigrahi, Prasanta & Goswami, Saptarsi. (2021). A Survey on Sentiment Analysis. 10.1007/978-981-33-4367-2_26.
19. Applied Natural Language Processing in the Enterprise. Ankur A. Patel, Ajay Uppili Arasanipalai. "O'Reilly Media, Inc.", 2021
20. Prakash, Aditya. (2020). Twitter Sentimental Analysis. International Journal for Modern Trends in Science and Technology. 6. 355-359. 10.46501/IJMTST061266.
21. "Ανάλυση Παλινδρόμησης" Εργαστήριο. Μαθηματικών & Στατιστικής / Γ. Παπαδόπουλος (www.aua.gr/gpapadopoulos)

22. Jurafsky D., Martin, J.H.., Speech and Language Processing: An Introduction to Natural
23. Language Processing, Computational Linguistics, and Speech Recognition. New Jersey:
24. Pearson Education, Inc, 2009.
25. Εξόρυξη Γνώσης από Δεδομένα Στο Κύρκος, Ε. 2015. Επιχειρηματική ευφυΐα και εξόρυξη δεδομένων. Αθήνα: Σύνδεσμος Ελληνικών Ακαδημαϊκών
26. https://eclass.ails.ece.ntua.gr/modules/document/file.php/103/%CE%94%CE%B9%CE%B1%CE%BB%CE%AD%CE%BE%CE%B5%CE%B9%CF%82/1_DM_Introduction.pdf
27. https://www.dit.uoi.gr/e-class/modules/document/file.php/179/less_1_chap%201.pdf
28. https://github.com/dimosbele/sentiment_analysis_greek
29. https://www.cs.uoi.gr/~pitoura/courses/dm/classification11b.pdf
30. https://scikit-learn.org/stable/modules/svm.html#svm-kernels
31. http://www.eng.ucy.ac.cy/gmitsis/ece795/lectures/ECE795_Lectures19_20_2012.pdf

Παράρτημα

```
#!/usr/bin/env python
# coding: utf-8
#0.Collect_tweets.py
 import tweepy from tweepy.streaming import
StreamListener from tweepy import
OAuthHandler from tweepy import Stream
import datetime import csv import pandas as
pd
 consumer_key = '********************' consumer_secret =
'**********************************' access_token =
*****************************************'
access_token_secret = ************************************
 auth = tweepy.OAuthHandler(consumer_key,
consumer_secret) auth.set_access_token(access_token,
access_token_secret) api = tweepy.API(auth)
  class
MyStreamListener(tweepy.StreamListener):
    def on_status(self,status):
        if hasattr(status, "retweeted_status"):
            try:
                current_status =
str(status.retweeted_status.extended_tweet["full_text"])
                current_status = current_status.replace('\n', ' ').replace('\r', '')
print(current_status)

except AttributeError:
                current_status = str(status.retweeted_status.text)
                current_status = current_status.replace('\n', ' ').replace('\r', '')
print(current_status)

else:
try:
                current_status = str(status.extended_tweet["full_text"])
current_status = current_status.replace('\n', ' ').replace('\r', '')
print(current_status)
                        except
AttributeError:
                current_status = str(status.text)
                current_status = current_status.replace('\n', ' ').replace('\r', '')
print('Gathering tweets for hashtag - ', current_status)

csvw.writerow([status.id,
status.user.screen_name,

# created_at is a datetime object, converting to just grab the month/day/year
status.created_at.strftime('%m/%d/%y'),
status.favorite_count,                              status.user.followers_count,
status.source,                          status.user.location,
current_status])
                    def
on_error(self,status_code):
if status_code == 420:
```

```
            print('You have been rate-limited for making too many requests')
return False
  if __name__ ==
'__main__':

    # This handles Twitter authetification and the connection to Twitter Streaming API
l = MyStreamListener()      auth = OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)      stream =
tweepy.Stream(auth, l)

    # Filter based on listed items      csvw =
csv.writer(open("5augoust", "a"))
csvw.writerow(['twitter_id', 'name', 'created_at',
                   'followers_count', 'source', 'region', 'text'])
    stream.filter(track=['εμβολιο','εμβόλιο', 'εμβολιασμος', 'αντιεμβολιαστες',
'αντιεμβολιαστές', '#ανεμβολιαστοι'])

#!/usr/bin/env python
# coding: utf-8
#1.Analyse_tweets.py
 import pandas as pd import
numpy as np import
matplotlib.pyplot as plt import
seaborn as sns import re
df = pd.read_csv('data/final.csv') df.head()
print("Number of tweets: {}".format(len(df)))

#Visualize the Source column percentages. colors = ['#93baf5','#6081b5',
'#cbf7e6', '#5db0d4', '#7fa0fa'] p =
df.Source.value_counts().head(5).plot.pie(x='lab', y='val', autopct='%1.1f%%',
rot=2, colors=colors, figsize=(15,10));
p.set_title("Συνηθισμένες συσκευές που χρησιμοποιούν οι Έλληνες του Twitter");
tweet_id = 222 tweet = df.iloc[tweet_id] print("Tweet:
{}".format(tweet["Tweet"]))

# Με τη συνάρτηση df.groupby().mean().nlargest() εντοπίζουμε τους χρήστες
# με τους περισσότερους ακολούθους και τους κατατάσουμε με το όνομα χρήστη τους.
 most_followers =
df.groupby('Username')['Followers'].mean().nlargest(15) most_followers
 ax = most_followers.plot(kind='barh', figsize=(10, 12), color='#b4aeeb',
zorder=2, width=0.85) plt.gca().invert_yaxis()
sns.despine(bottom = True, left = True)
plt.ylabel(None) plt.xticks(None)
plt.xticks([]) plt.yticks(fontsize=18,
rotation=0)
 for index, value in
enumerate(most_followers):
    plt.text( value, index, str(round(value, 2)), va = 'center', ha='left', fontsize=16)
     plt.suptitle('Οι χρήστες με τους περισσότερους followers'.title(),
fontsize=20) plt.show()  most_tweets_users =
df.Username.value_counts().reset_index() most_tweets_users.columns =
['Username','counts'] most_tweets_users.head(20)
 users =
df['Username'].apply(pd.Series).stack().value_counts().head(10) ax =
```

```
users.plot(kind='barh', figsize=(10, 12), zorder=2, width=0.85)
plt.gca().invert_yaxis()
sns.despine(bottom = True, left = True)
plt.ylabel(None) plt.xticks(None)
plt.xticks([]) plt.yticks(fontsize=18,
rotation=0)
 for index, value in
enumerate(users):
    plt.text( value, index, str(round(value, 2)), va = 'center', ha='left', fontsize=16)
     plt.suptitle('Οι χρήστες που εμφανίζονται πιο πολύ στο αρχείο δεδομένων'.title(),
fontsize=15) plt.show()


# Βρίσκουμε τη περιοχή από την οποία δημοσιεύτηκαν τα περισσότερα tweets.
most_tweets_users = df.Location.value_counts().reset_index()
most_tweets_users.columns = ['Location','counts'] most_tweets_users.head(20)

colors = ['#93baf5','#d5abd9', '#cbf7e6', '#5db0d4', '#ebcaea','#a2ebe1','#a9aade']

p = df.Location.value_counts().head(7).plot.pie(x='Location', autopct='%1.1f%%',
startangle=90,
                                                rot=2, colors=colors, figsize=(14,10));
p.set_title("Location pie chart"); plt.ylabel(None);

date_counts = df[['Tweet', 'Date']].groupby(['Date']).count().reset_index()
most_tweets = date_counts.groupby('Date')['Tweet'].mean().nlargest(20).reset_index()
most_tweets.columns = ['date','count'] most_tweets.head(20)
 import plotly.express as px px.line(most_tweets, x = 'date', y = 'count', title =
'Tweet counts per day lineplot')


# Υπολογίζουμε τα Tweets που περιέχουν hashtag, το οποίο συμβολίζεται με '#'.
tweets_with_hashtag = df[df['Tweet'].str.contains('#')==True]
print("Ο αριθμός των tweets που περιέχουν hashtag: {}".format(len(tweets_with_hashtag)))


# Με ανάλογο τρόπο μπορούμε να εντοπίσουμε και τα tweets που δεν περιέχουν hashtag σε μια
συλλογή με tweets.
tweets_without_hashtag = df[df['Tweet'].str.contains('#')==False]
print("Ο αριθμός των tweets που δεν περιέχουν hashtag:
{}".format(len(tweets_without_hashtag)))

#Save the file hastag_frame =
pd.read_csv('HASHTAG.csv')
 hashtag = hastag_frame.has_hashtag.value_counts().plot.pie(x='lab', y='val',
autopct='%1.1f%%', rot=2,figsize=(12,8)); plt.suptitle('Ποσοστό δημοσιεύσεων με
hashtag'.title(), fontsize=20);


# Υπολογίζουμε τα tweets που περιέχουν link και στη συνέχεια βλέπουμε με την εντολή sum()
τον αριθμό τους.
tweets_with_url = df[df['Tweet'].str.contains('http')==True]
print("Ο αριθμός των tweets που περιέχουν link: {}".format(len(tweets_with_url)))
```

```
# Επιβεβαιώνουμε την ύπαρξη link σε κάποιες από τις εγγραφές
tweets_with_url['Tweet'][37756]
# Υπολογίζουμε τα tweets που δεν περιέχουν link και στη συνέχεια βλέπουμε με την εντολή
sum() τον αριθμό τους.
tweets_without_url = df[df['Tweet'].str.contains('http')==False]
print("Ο αριθμός των tweets που δεν περιέχουν link: {}".format(len(tweets_without_url)))
 url_frame =
pd.read_csv('url_frame.csv')
tag = url_frame.has_url.value_counts().head(5).plot.pie(x='lab', y='val',
autopct='%1.1f%%', rot=2,figsize=(12,8)); plt.suptitle('Ποσοστό δημοσιεύσεων
με link'.title(), fontsize=20);


# Υπολογίζουμε τα tweets που περιέχουν αναφορά (mention) η οποία στο Twitter συμβολίζεται
με @.
tweets_with_mention = df[df['Tweet'].str.contains('@')== True]
print("Ο αριθμός των tweets που περιέχουν mention: {}".format(len(tweets_with_mention)))

tweets_without_mention = df[df['Tweet'].str.contains('@')== False]
print("Ο αριθμός των tweets που δεν περιέχουν mention:
{}".format(len(tweets_without_mention)))

mention_frame = pd.read_csv('MENTIONS.csv')

tag = mention_frame.has_mention.value_counts().head(5).plot.pie(x='lab', y='val',
autopct='%1.1f%%', rot=2,figsize=(12,8)); plt.suptitle('Ποσοστό δημοσιεύσεων με
mention'.title(), fontsize=20); tweets_with_emojis =
df[df['Tweet'].str.contains('ϖ§)== True]
print("Ο αριθμός των tweets που περιέχουν το emoji: {}".format(len(tweets_with_emojis)))
 tweets_with_emojis['Tweet'][198]

# Υπολογίζουμε τα tweets που είναι retweets και όχι αυτούσια. Στο twitter αυτό
συμβολίζεται με RT

retweets = df[df['Tweet'].str.startswith('RT')== True] print("Ο
αριθμός των retweets: {}".format(len(retweets)))
 retweets =
pd.read_csv('Retweeted.csv')

no_retweets = df[df['Tweet'].str.startswith('RT')== False]
print("Ο αριθμός των tweets που δεν είναι retweet: {}".format(len(no_retweets)))
 retweets_plot = retweets.is_retweet.value_counts().head().plot.pie(x='lab', y='val',
autopct='%1.1f%%', rot=2,figsize=(12,8)); plt.suptitle('Ποσοστό δημοσιεύσεων που είναι
retweets'.title(), fontsize=20);


#!/usr/bin/env python
# coding: utf-8
#2.Preprocess_tweets.py
 import pandas as pd import
matplotlib.pyplot as plt import seaborn
as sns import plotly.express as px import
re import string import nltk from
nltk.corpus import stopwords from
```

```
collections import Counter from wordcloud
import WordCloud import preprocessor as
preproc from nltk.tokenize import
RegexpTokenizer

df = pd.read_csv('data/final.csv') df.head(4)  def
delete_tonous(df, column_to_process, processed_column='Tweet'):
         if (processed_column !=
column_to_process):
        df[processed_column] = df[column_to_process]  # create new column

    # replace greek hyphend letters     replacements = {processed_column: {'ά': 'α',
'έ': 'ε', 'ή': 'η', 'ί': 'ι', 'ό': 'ο',
'ύ': 'υ', 'ώ': 'ω', 'ϊ': 'ι'}}
df.replace(replacements, regex=True, inplace=True)
         return
(df)
 df['Tweet'][6]
# Αφαιρούμε
όλους τους μη
ελληνικούς
χαρακτήρες από
τα tweets.
df['Tweet'] = df['Tweet'].str.replace(r'[a-zA-Z0-9]', '', regex=True)

# Αφαιρούμε τους τόνους καθώς και τα διαλυτικά από τα tweets. Τα διαλυτικά είναι σπάνια
στην ελληνική γλώσσα, ωστόσο μια πολύ συχνή λέξη στα δεδομένα είναι η λέξη κορονοϊός η
οποία συναντάται πολύ συχνά στα δεδομένα επομένως είναι απαραίτητο να αφαιρεθεί.
delete_tonous(df=df, column_to_process='Tweet', processed_column='Tweet').head(2)
# Αφαιρούμε όλα τα links από τα tweets. remove_url =
lambda x: re.sub('https\S+', '', str(x)) df['Tweet']
= df.Tweet.apply(remove_url)

# Μετατρέπουμε τα tweets σε μικρά γράμματα to_lowercase = lambda x : x.lower()
df['Tweet'] = df.Tweet.apply(to_lowercase) delete_tonous(df=df,
column_to_process='Tweet', processed_column='Tweet').head(3)
# Αφαιρούμε όλα τα σημεία στίξης από τα δεδομένα remove_puncs = lambda x:
x.translate(str.maketrans('','',string.punctuation)) df['Tweet'] =
df.Tweet.apply(remove_puncs) df['Tweet']

# Στη στήλη Location υπάρχουν πολλές κενές εγγραφές. Τις αντικαθηστούμε με 'prefer not to
say' προκειμένου να σβήσουμε στη συνέχεια τις κενές εγγραφές από τη στήλη Tweet.
Αφαιρούμε τη συνέχεια τα επαναλαμβανόμενα tweets καθώς και τις κενές εγγραφές
df["Location"].fillna('prefer not to say', inplace = True)

# Επιβεβαιώνουμε την αλλαγή df[df['Location'] ==
"prefer not to say"].head()
 df['Tweet'] =
df.Tweet.drop_duplicates() df =
df.dropna()

# Βλέπουμε αν υπάρχουν τυχόν υπολοιπόμενες κενές εγγραφές στη στήλη Tweet
df['Tweet'].isna().sum()

# Μετράμε το μέσο μάκρος των Tweets μετά την αφαίρεση των links και των mentions.
```

```
df['length'] = df['Tweet'].apply(len)

# Βλέπουμε τα στατιστικά στοιχεία του μάκρους των tweets. Υπάρχει κάποιο tweet με 0
χαρακτήρες. Εφόσον είναι μόνο ένα, θα το σβήσουμε στη συνέχεια από το αρχείο μας.
df.length.describe()

#Average tweet
df[df['length'] == 140]['Tweet'].iloc[0]

#Longest tweet
df[df['length'] == 282]['Tweet'].iloc[0]

#tweet with 0 characters
df[df['length'] == 0]

#Βλέπουμε τη κατανομή του μήκους των tweets from
jupyterthemes import jtplot
jtplot.style(theme='monokai', context='notebook', ticks=True, grid=False)
df['length'].plot(bins=100, kind='hist');

# ### Removing Stopwords
#
# Είναι πολύ σημαντικό να αφαιρέσουμε τις stopwords γιατί αποτελούν τις λέξεις που
επαναλαμβάνονται συνεχώς και δεν δίνουν κάποιο νόημα στη φράση. Ενημερώνουμε το πακέτο
της nltk με λέξεις που εντοπίστηκαν στα δεδομένα μας και θα ήταν καλό να αφαιρεθούν.
 from nltk.corpus import
stopwords
 stop_words = stopwords.words('greek') + ['μητσοτακηπαραιτησου', 'η', 'ειναι',
'με', 'θα', 'απο', 'τα', 'ο', 'την', 'του', 'σε', 'οτι', 'της', 'τον', 'οι', 'στο',
'αν',
'τις', 'τη', 'κ', 'σας', 'νδ', 'οι', 'ο', 'η', 'μου', 'σου', 'τα', 'απο', 'βαρυμπομπη',
'μμεξεφτιλες', 'μητσοτακη', 'αδωνις', 'εμβολιο', 'χαχαχα', 'αν', 'astrazeneca', 'astra',
'zeneca', 'phizer', 'α', 'ε', 'ν', 'via', 'τι', 'marka149133376', 'ο', 'χ', 'length',
'rt', 'political', 'fediuld76', 'πιο', 'ποθενεσχες', 'frq', 'COVID19greece', 'COVID19',
'Covid19', 'Marka149133376', 'covid19gr', 'covid19greece', 'covid', 'εγω', 'εσυ',
'adonisgeorgiadi', 'κανω','αλλο', 'κανεις',  'σαν', 'κατι', 'πριν', 'ολα', 'εχουν',
'κανουν', 'εχει', 'έχουν', 'κανουν', 'οπως', 'μια', 'ενα', 'amp', 'στις', 'στα',
'στους', 'εδω', 'της', 'τους', 'μας', 'ρε', '-', 'ουτε', 'εχω', 'οταν', 'σου',
'μητσοτακηγαμιεσαι', 'μου'] remove_words = lambda x : ' '.join([word for word in
x.split() if word not in stop_words]) df['Tweet_without_stopwords'] =
df.Tweet.apply(remove_words) df['Tweet_without_stopwords']

#Tokenize tweets tokenizer = RegexpTokenizer(r'\w+') df['Tokens'] =
df['Tweet_without_stopwords'].apply(lambda text: tokenizer.tokenize(text))

#inspecting some tweets
df['Tokens'][7]
 words_list = [word for line in df.Tweet_without_stopwords for word in
line.split()] words_list[:20]

# Προκειμένου να βρούμε τις πιο συνηθισμένες λέξεις που εντοπίζονται στα tweets θα
χρησιμοποιήσουμε το πακέτο collections από τη βιβλιοθήκη Counter. Μέσω της συνάρτησης
most_common() μπορούμε να εντοπίσουμε τις πιο συνηθισμένες λέξεις στα δεδομένα μας,
δηλαδή στη μεταβλητή word_list που δημιουργήθηκε. Θα μετατρέψουμε αυτές τις λέξεις σε ένα
νέο αρχείο δεδομένων.
word_counts = Counter(words_list).most_common(50)
word_counts
```

```
# Δημιουργούμε ένα καινούργιο dataFrame με στήλες την λέξη και τη συχνότητα της λέξης.
words_df = pd.DataFrame(word_counts)
words_df.columns = ['word', 'frq'] words_df.head()
px.bar(words_df, x='word', y='frq', title='Οι πιο συχνές λέξεις')
 #WordCloud
wordcloud = WordCloud(width = 800, height = 400, random_state = 21, max_font_size = 100,
collocations = False).generate(str(df.Tweet_without_stopwords))
 plt.figure(figsize = (20,
10))
plt.imshow(wordcloud, interpolation = 'bilinear') plt.axis('off');

#Save the new file
df.to_csv('data/modified.csv')



#!/usr/bin/env python
# coding: utf-8
#3.Sentiment_analysis.py
 import pandas as pd import numpy as np import matplotlib.pyplot as
plt from nltk.tokenize import TweetTokenizer from
vaderSentiment.vaderSentiment import SentimentIntensityAnalyzer from
textblob import TextBlob import seaborn as sns import string import
plotly.express as px

#Load the modified dataset with the translations tweets_df
= pd.read_csv('modified_with_length.csv') tweets_df.head()

#Make english text lowercase to_lowercase = lambda x :
str(x).lower() tweets_df['Translated'] =
tweets_df.Translated.apply(to_lowercase)

#Remove english punctuations remove_puncs = lambda x:
x.translate(str.maketrans('','',string.punctuation)) tweets_df['Translated']
= tweets_df.Translated.apply(remove_puncs) tweets_df['Translated']



# ### Συναισθηματική ανάλυση με τη μέθοδο TextBlob
#Make a copy of our data textblob_analysis
= tweets_df.copy()

#Create polarity with textblob column
textblob_analysis["Polarity"] = textblob_analysis["Translated"].apply(lambda word:
TextBlob(str(word)).sentiment.polarity)
labelize_textblob = lambda x : 'neutral' if x==0 else('positive' if x>0 else 'negative')
textblob_analysis['label'] = textblob_analysis.Polarity.apply(labelize_textblob)
textblob_analysis.head() textblob_analysis.Polarity.describe()

#drop english column
textblob_analysis.drop('Translated', axis=1, inplace=True)

# Τα πιο θετικά και πιο αρνητικά tweets που εντόπησε η μέθοδος.
textblob_analysis[textblob_analysis.Polarity == -1.000000]
textblob_analysis[textblob_analysis.Polarity == 1.000000] textblob_analysis.describe()
```

```
#Save dataframe
# textblob_analysis.to_csv('textblob_label.csv')

# ### Προσέγγιση Vader #Make a
copy of our data vader_analysis =
tweets_df.copy()
 sid = SentimentIntensityAnalyzer() ps = lambda x
: sid.polarity_scores(str(x)) vader_scores =
vader_analysis.Translated.apply(ps) vader_scores

#Make it a dataframe sentiment_df_vader = pd.DataFrame(data
= list(vader_scores)) sentiment_df_vader.head()
 labelize = lambda x : 'neutral' if x==0 else('positive' if x>0 else
'negative') sentiment_df_vader['label'] =
sentiment_df_vader.compound.apply(labelize) sentiment_df_vader.head()

#Join dataframes vader_data_with_compound =
tweets_df.join(sentiment_df_vader.compound)
vader_data_with_compound.head()
 vader_data_with_label =
vader_data_with_compound.join(sentiment_df_vader.label)
vader_data_with_label.head(20)

#Drop english column vader_data_with_label.drop('Translated',
axis=1, inplace=True)

# Καλώντας τη συνάρτηση describe() μπορούμε να πάρουμε μερικά στατιστικά στοιχεία για τη
πολικότητα των tweets.  Βρίσκοντας τα tweets με τη πιο μικρή και πιο μεγάλη πολικότητα
μπορούμε να βρούμε το πιο αρνητικό και το πιο θετικό tweet.
vader_data_with_label.compound.describe()

# Βλέπουμε ότι το πιο θετικό tweet έχει χαρακτηριστεί με τη κλάση θετικό λόγω των πολλών
emojis τα οποία περιέχει. Αυτό αποτελεί ψευδώς θετικό tweet αφού στην ουσία περιέχει
ειρωνεία.
vader_data_with_label[vader_data_with_label.compound == 0.997000]
vader_data_with_label['Tweet_without_stopwords'][1788]
vader_data_with_label[vader_data_with_label.compound == -0.984700]
vader_data_with_label['Tweet_without_stopwords'][3353]

#Save vader data file
vader_data_with_label.to_csv('vader_with_label.csv')

# ### Οπτικοποίηση δεδομένων
vader_pie = [len(vader_data_with_label[vader_data_with_label['label'] == 'positive']),
len(vader_data_with_label[vader_data_with_label['label'] == 'negative']),
len(vader_data_with_label[vader_data_with_label['label'] == 'neutral'])]

blob_pie = [len(textblob_analysis[textblob_analysis['label'] == 'positive']),
len(textblob_analysis[textblob_analysis['label'] == 'negative']),
len(textblob_analysis[textblob_analysis['label'] == 'neutral'])] labels =
['Positive', 'Negative', 'Neutral'] colors = ['aquamarine', 'tomato',
'skyblue']
```

```
# Ανάλυση ποσοστού tweets ανά κατηγορία συναισθήματος με δυο διαφορετικές προσεγγίσεις.
Βλέπουμε ότι οι δυο προσεγγίσεις που ακολουθήσαμε έχουν σχεδόν το ίδιο ποσοστό θετικών
δημοσιεύσεων ωστόσο υπάρχει μεγάλη διαφοροποίηση στα αρνητικά αποτελέσματα.
 plt.style.use('ggplot') plt.figure(figsize = (20, 10)) plt.subplot(1, 2,
1) plt.pie(vader_pie, labels = labels, colors = colors, autopct =
'%1.1f%%') plt.title('Ανάλυση συναισθήματος με Vader') plt.subplot(1, 2,
2) plt.pie(blob_pie, labels = labels, colors = colors, autopct =
'%1.1f%%') plt.title('Ανάλυση συναισθήματος με TextBlob');


#Count the labels counts_df_vader =
vader_data_with_label.label.value_counts().reset_index() counts_df_vader
counts_df_textblob = textblob_analysis.label.value_counts().reset_index()
counts_df_textblob


# Βλέπουμε τη διακύμανση του συναισθήματος των tweets καταμετρώντας τα σύμφωνα με την
ημερομηνία και τη κλάση τους και για τις δυο προσεγγίσεις.


#Vader counts data_agg_vader = vader_data_with_label[['Tweet', 'Date',
'label']].groupby(['Date', 'label']).count().reset_index() data_agg_vader.columns =
['date','label','counts'] data_agg_vader.head()


#Vader line visualisation px.line(data_agg_vader, x = 'date', y = 'counts', color =
'label', title = 'Daily tweets sentimental Analysis with Vader method')
data_agg_textblob = textblob_analysis[['Tweet', 'Date', 'label']].groupby(['Date',
'label']).count().reset_index() data_agg_textblob.columns = ['date','label','counts']
data_agg_textblob.head() px.line(data_agg_textblob, x = 'date', y = 'counts', color =
'label', title = 'daily tweets sentimental Analysis with Textblob')
```

```
#!/usr/bin/env python
# coding: utf-8
#4.Modeling.py
 import pandas as pd import
matplotlib.pyplot as plt import
seaborn as sns import pickle
 from sklearn.feature_extraction.text import TfidfVectorizer from
sklearn.model_selection import train_test_split from sklearn.svm import
LinearSVC from sklearn.metrics import classification_report from
sklearn.metrics import accuracy_score, confusion_matrix,
precision_recall_fscore_support from nltk.tokenize import RegexpTokenizer
pd.set_option('display.max_colwidth', 1)
pd.set_option('display.max_columns', 500)
 df =
pd.read_csv('data/sentiment_analysis_data.csv')
df.head() df[df['label'] == 'negative'].head()
df['label'].value_counts(normalize=True) * 100
 df.dropna(inplace =
True)
 sns.countplot(data=df, x =
'label');
 from sklearn.feature_selection import SelectKBest,
chi2 from sklearn.model_selection import
cross_val_score
```

```
tfidf = TfidfVectorizer(max_features=45000)
X = df['Tweet_without_stopwords'] y =
df['label']

X = tfidf.fit_transform(df['Tweet_without_stopwords'].values.astype('U'))

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20,  shuffle =
True, random_state = 10)


# ### Support vector machines
svm = LinearSVC(max_iter=40000)
svm.fit(X_train, y_train) y_pred
= svm.predict(X_test)
accuracy_score(y_test, y_pred, normalize=True) print(classification_report(y_test,
y_pred))
 sample = ["ΤΕΛΕΙΑ ΜΕΡΑ ΣΗΜΕΡΑ"] sample =
tfidf.transform(sample).toarray()
sentiment = svm.predict(sample) print('Η
πρόταση είναι',":",sentiment)

from sklearn.metrics import confusion_matrix
import matplotlib.pyplot as plt import
seaborn as sns
from sklearn.metrics import accuracy_score, confusion_matrix,
precision_recall_fscore_support


# Creates a confusion matrix cm =
confusion_matrix(y_test, y_pred) #
Transform to df for easier plotting
cm_df = pd.DataFrame(cm,
                     index = ['Negative','Neutral', 'Positive'],
columns = ['Negative','Neutral', 'Positive'])

plt.figure(figsize=(5.5,4)) sns.heatmap(cm_df,
annot=True) plt.title('Support Vector
Classification
\nAccuracy:{0:.3f}'.format(accuracy_score(y_te
st, y_pred))) plt.ylabel('True label')
plt.xlabel('Predicted label') plt.show()
sample = ["πολυ κακες οδηγιες χρησεως,
επομενως δεν γνωριζεις αν το εκανες σωστα.
Κακες οδηγιες ερμηνειας του αποτελεσματος. Στα
σκιτσα η εικονα διαφερει πολυ απο την
πραγματικη, οποτε δεν μπορεις να εισαι
σιγουρος αν ερμηνευεις σωστα το αποτελεσμα.
Εμενα μου εβγαλε μια πολυ χονδρη γραμμη στο C
και μια πολυ λεπτη και αδιορατη στο T (στις
οδηγιες για την ερμηνεια ολες οι γραμμες των
σκιτσων ειναι λεπτες.). να το παρω σαν θετικο
η να το παρω ως αρνητικο? Τωρα με εβαλε σε
ανησυχια και πρεπει να κανω μοριακο. Μη
```

```
αξιοπιστο λοιπον"] sample =
tfidf.transform(sample).toarray() sentiment =
svm.predict(sample) print('Η πρόταση
είναι',":",sentiment)
 import pickle pickle.dump(svm,
open('models/model_svm.pickle', 'wb')) pickle.dump(tfidf,
open('models/model_tfidf.pickle', 'wb'))

# ### LinearSVC with OneVsRestClassifier from
sklearn.multiclass import OneVsRestClassifier from
sklearn.svm import SVC
 ovsrc = OneVsRestClassifier(LinearSVC(random_state=0)).fit(X,
y) linear_ovsrc = ovsrc.predict(X_test)
accuracy_score(y_test, linear_ovsrc, normalize=True) print(classification_report(y_test,
linear_ovsrc))
  from sklearn.metrics import confusion_matrix import matplotlib.pyplot as
plt import seaborn as sns from sklearn.metrics import accuracy_score,
confusion_matrix, precision_recall_fscore_support

# Creates a confusion matrix cm =
confusion_matrix(y_test, y_pred) #
Transform to df for easier plotting
cm_df = pd.DataFrame(cm,
                     index = ['Negative','Neutral', 'Positive'],
columns = ['Negative','Neutral', 'Positive'])

plt.figure(figsize=(5.5,4)) sns.heatmap(cm_df,
annot=True)
plt.title('Support Vector Classification
\nAccuracy:{0:.3f}'.format(accuracy_score(y_test, linear_ovsrc)))
plt.ylabel('True label') plt.xlabel('Predicted label') plt.show()
sample = ["Πήρα το τεστ για το σπίτι να το εχω άμεσα διαθεσιμο. Έχω ηδη υποβληθεί σε 3
τεστ (λόγω εργασίας σε νοσοκομείο) τα οποία έδειξαν ακριβώς τα ίδια αποτελέσμα τα με τη
μεθοδο RT-PCR και τα υπόλοιπα rapid test που κάνω εκεί."] sample =
tfidf.transform(sample).toarray() sentiment = ovsrc.predict(sample)
print('Η πρόταση είναι',":",sentiment)
 pickle.dump(ovsrc, open('models/OneVsRestClassifier.pickle',
'wb'))



# ### Logistic Regression Classifier # Training the
classifier from sklearn.linear_model import
LogisticRegression classifier =
LogisticRegression(multi_class='multinomial')
classifier.fit(X_train, y_train)

# Testing model performance sent_pred_log =
classifier.predict(X_test) accuracy_score(y_test,
sent_pred_log, normalize=True)
print(classification_report(y_test, sent_pred_log))
 sample = ["ΤΕΛΕΙΑ ΜΕΡΑ ΣΗΜΕΡΑ"] sample =
tfidf.transform(sample).toarray()
sentiment = classifier.predict(sample)
print('Η πρόταση είναι',":",sentiment)
```

```
  print(classification_report(y_test,
sent_pred_log))

# Creates a confusion matrix cm = confusion_matrix(y_test,
sent_pred_log) # Transform to df for easier plotting cm_df =
pd.DataFrame(cm,                      index =
['Negative','Neutral', 'Positive'],                        columns
= ['Negative','Neutral', 'Positive'])

plt.figure(figsize=(5.5,4)) sns.heatmap(cm_df,
annot=True)
plt.title('Logistic Regression Classification
\nAccuracy:{0:.3f}'.format(accuracy_score(y_test, sent_pred_log)))
plt.ylabel('True label') plt.xlabel('Predicted label') plt.show()



# ### Random Forest classifier
from sklearn.ensemble import RandomForestClassifier from
sklearn.model_selection import train_test_split, GridSearchCV from
sklearn.model_selection import cross_val_score

rfc = RandomForestClassifier(n_estimators= 200, max_depth=None).fit(X_train,y_train)
sent_pred_random_forest = rfc.predict(X_test)
accuracy_score(y_test, sent_pred_random_forest, normalize=True)
print(classification_report(y_test, sent_pred_random_forest))

# Creates a confusion matrix
cm = confusion_matrix(y_test, sent_pred_random_forest)
# Transform to df for easier plotting
cm_df = pd.DataFrame(cm,
                     index = ['Negative','Neutral', 'Positive'],
columns = ['Negative','Neutral', 'Positive'])

plt.figure(figsize=(5.5,4)) sns.heatmap(cm_df, annot=True)
plt.ylabel('True label') plt.xlabel('Predicted label')
plt.show()  sample = ["Τι ΟΜΟΡΦΗ μέρα που έχει σήμερα. Θα πάω
για ποδήλατο"] sample = tfidf.transform(sample).toarray()
sentiment = rfc.predict(sample) print('Η πρόταση
είναι',":",sentiment)



# ### Random Forest with OneVsRestClassifier from
sklearn.multiclass import OneVsRestClassifier from
sklearn.svm import SVC
 ovsrc = OneVsRestClassifier(RandomForestClassifier(random_state=0)).fit(X,
y) rf_ovsrest = ovsrc.predict(X_test)

accuracy_score(y_test, rf_ovsrest, normalize=True) print(classification_report(y_test,
rf_ovsrest))

# Creates a confusion matrix cm = confusion_matrix(y_test,
rf_ovsrest) # Transform to df for easier plotting cm_df =
pd.DataFrame(cm,                      index =
```

```
['Negative','Neutral', 'Positive'],                    columns
= ['Negative','Neutral', 'Positive'])

plt.figure(figsize=(5.5,4))
sns.heatmap(cm_df, annot=True)
plt.ylabel('True label')
plt.xlabel('Predicted label')
plt.show()
sample = ["Τι ΟΜΟΡΦΗ μέρα που έχει σήμερα. Θα πάω για ποδήλατο"]
sample = tfidf.transform(sample).toarray() sentiment =
ovsrc.predict(sample) print('Η πρόταση είναι',":",sentiment)

# ### Multinomial Naive Bayes

from sklearn.naive_bayes import MultinomialNB
clf = MultinomialNB() clf.fit(X_train,
y_train)

from sklearn.naive_bayes import MultinomialNB

sent_pred_bayes = clf.predict(X_test)
accuracy_score(y_test, sent_pred_bayes, normalize=True)
print(classification_report(y_test, sent_pred_bayes))  from sklearn.metrics import
confusion_matrix

# Creates a confusion matrix cm = confusion_matrix(y_test,
sent_pred_bayes) # Transform to df for easier plotting cm_df =
pd.DataFrame(cm,                    index =
['Negative','Neutral', 'Positive'],                    columns
= ['Negative','Neutral', 'Positive'])

plt.figure(figsize=(5.5,4)) sns.heatmap(cm_df,
annot=True)
plt.title('Nayve Bayes \nAccuracy:{0:.3f}'.format(accuracy_score(y_test,
sent_pred_bayes))) plt.ylabel('True label') plt.xlabel('Predicted
label') plt.show()  sample = ["τελεια μερα"] sample =
tfidf.transform(sample).toarray() sentiment = clf.predict(sample)
print('Η πρόταση είναι',":",sentiment)
#!/usr/bin/env python
# coding: utf-8
#5.Test_model.py
 import tweepy import re import
pickle from tweepy import
OAuthHandler import pandas as
pd import matplotlib.pyplot as
plt import string

pd.set_option('display.max_colwidth', 1)
pd.set_option('display.max_columns', 500)

consumer_key = 'zRLze0lH2AKxY1bkVqpXfuhDZ'
consumer_secret = 'hAbqGAiTjwvTVBdUX0MMdl7Eqcb5eQPPY5jwx4SkGj07I4hGaD'
access_token = '1397618616653209602-3LUD2QVsxS5xjznEcTQJ6545G2lUko'
access_secret = 'DALjzoSMrU71FoULD4VOYN46yCtl9c1xsKghGtQGPvNxa'
```

```
# Loading the vectorizer and classfier with
open('OneVsRestClassifier.pickle','rb') as f:
    svm_model = pickle.load(f)
     with open('model_tfidf.pickle','rb')
as f:
    tfidf = pickle.load(f)
 import csv auth = OAuthHandler(consumer_key,
consumer_secret)
auth.set_access_token(access_token, access_secret)
args = ['εμβολιο'];
api = tweepy.API(auth,timeout=10)

# Fetching the tweets list_tweets
= []
 query = args[0]
if len(args) == 1:
    for status in tweepy.Cursor(api.search, q=query+" -
filter:retweets",lang='el',result_type='recent', tweet_mode = 'extended').items(500):
        list_tweets.append(status.full_text)
         for tweet in list_tweets:     tweet = tweet.lower()
tweet = re.sub('https\S+', '', tweet)     tweet = re.sub(r'@[a-
zA-Z0-9]', '', tweet)     tweet = re.sub(r'[a-zA-Z0-9]', '',
tweet)     tweet = re.sub(r"ά","α",tweet)     tweet =
re.sub(r"έ","ε",tweet)     tweet = re.sub(r"ή","η",tweet)
tweet = re.sub(r"ί","ι",tweet)     tweet =
re.sub(r"ύ","υ",tweet)     tweet = re.sub(r"ώ","ω",tweet)
tweet = re.sub(r"ΐ","ι",tweet)       tweet =
re.sub(r"ό","ο",tweet)     tweet = re.sub(r"ό","ο",tweet)
tweet = re.sub(r"#","",tweet)     tweet = re.sub(r"!","",tweet)
tweet = re.sub(r"«","",tweet)     tweet = re.sub(r"»","",tweet)
tweet = re.sub(r";","",tweet)     tweet = re.sub(r"ϋ","",tweet)
tweet = re.sub(r"/","",tweet)         tweet = re.sub(r',', ' ',
tweet)     sent =
svm_model.predict(tfidf.transform([tweet]).toarray())
print(tweet,":",sent)

    # Filter based on listed items
    csvw = csv.writer(open("predicted_new2", "a"))
csvw.writerow([status.user.screen_name,
                   # created_at is a datetime object, converting to just grab the
month/day/year
                   status.created_at.strftime('%m/%d/%y'),
sent,                    tweet])

df = pd.read_csv('predict.csv')

#remove punctuations from sentiment column
remove_puncs = lambda x: x.translate(str.maketrans('','',string.punctuation))
df['sentiment'] = df.sentiment.apply(remove_puncs) df['sentiment']
df.head(10)
df['sentiment'].value_counts(normalize=True) * 100
```

```
sentiment = [len(df[df['sentiment'] == 'positive']),
len(df[df['sentiment'] == 'negative']),
len(df[df['sentiment'] == 'neutral'])]
 labels = ['Positive', 'Negative',
'Neutral'] colors = ['aquamarine', 'pink',
'skyblue']
 t = df['sentiment'].value_counts(normalize=True) *
100 plt.style.use('ggplot') plt.figure(figsize = (20,
10))
plt.pie(t, labels = labels, colors = colors, autopct = '%1.1f%%') plt.title('Ανάλυση
συναισθήματος με μοντέλο πρόβλεψης');
```

Κώδικας εφαρμογής συναισθηματικής ανάλυσης

```
from flask import Flask,render_template,url_for,request import pandas as pd
import pickle from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.model_selection import train_test_split from sklearn.svm
import LinearSVC from sklearn.metrics import classification_report from
sklearn.metrics import accuracy_score, confusion_matrix,
precision_recall_fscore_support from sklearn.multiclass import
OneVsRestClassifier from sklearn.svm import SVC
  app =
Flask(__name__)

@app.route('/')
def home():
        return render_template('home.html')

@app.route('/predict',methods=['POST']) def
predict():
        df = pd.read_csv("sentiment_analysis_data.csv")
             # Features and Labels
        df['label'] = df['label'].map({'negative': -1, 'neutral': 0, 'positive':1})
# Extract Feature With CountVectorizer

        tfidf = TfidfVectorizer(max_features=45000)
X = df['Tweet_without_stopwords']         y =
df['label']
        X = tfidf.fit_transform(df['Tweet_without_stopwords'].values.astype('U'))

        from sklearn.model_selection import train_test_split
        X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.20,
shuffle = True, random_state = 10)         ovsrc =
OneVsRestClassifier(LinearSVC(random_state=0)).fit(X, y)
         if request.method ==
'POST':
                message = request.form['message']
        data = [message]
                vect = tfidf.transform(data).toarray()
my_prediction = ovsrc.predict(vect)         return
render_template('result.html',prediction = my_prediction)
 if __name__ ==
'__main__':
app.run(debug=True)
```