



ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

Πρόγραμμα Μεταπτυχιακών Σπουδών (ΠΜΣ) στις
«Ψηφιακές Εφαρμογές και την Καινοτομία»

Capstone Project

«Ανάλυση δεδομένων κοινωνικών δικτύων με εφαρμογή στο
Twitter χρησιμοποιώντας εργαλεία της γλώσσας Python»

Κορφιάτης Αντώνιος

A.M. ΠΚΜ201903

Επιβλέπουσα

Χριστοπούλου Ελένη

Κέρκυρα, Φεβρουάριος 2021

Τριμελής Επιτροπή:

Χριστοπούλου Ελένη
ΕΔΠ Ιόνιο Πανεπιστήμιο

Κουρουθανάσης Παναγιώτης
Καθηγητής Ιόνιο Πανεπιστήμιο

Ρίγγας Δημήτριος
ΕΔΠ Ιόνιο Πανεπιστήμιο

Περίληψη

Η επιστήμη της ανάλυσης των δεδομένων βρίσκεται σε άνθηση. Η εμφάνιση των κοινωνικών μέσων ενημέρωσης έδωσε στους χρήστες του διαδικτύου έναν χώρο για να εκφράσουν και να μοιραστούν τις σκέψεις και τις απόψεις τους για κάθε είδους θέματα και εκδηλώσεις. Το Twitter, με σχεδόν 600 εκατομμύρια χρήστες και πάνω από 250 εκατομμύρια μηνύματα την ημέρα αποτελεί μια τεράστια δεξαμενή άντλησης δεδομένων που προσφέρονται για επεξεργασία. Η πανδημία Covid-19 δεν έχει δημιουργήσει μόνο όλεθρο στις οικονομικές συνθήκες, στην φυσική υγεία, και στις συνθήκες εργασίας, αλλά έχει επίσης δημιουργήσει μια θέση στο μυαλό όλων των ανθρώπων παγκοσμίως.

Ένα από τα θέματα που προκάλεσε πληθώρα σχολίων και δημοσιεύσεων αποτέλεσε ο εμβολιασμός και η στάση των ανθρώπων ως προς αυτόν. Η παρούσα εργασία συγκεντρώνει μηνύματα από την πλατφόρμα Twitter τα οποία σχετίζονται με τα εμβόλια που βρίσκονται σε στάδιο ανάπτυξης για την καταπολέμηση της πανδημίας, προκειμένου να πραγματοποιηθεί ανάλυση συναισθήματος και να κατηγοριοποιηθούν αυτά σε κλάσεις ώστε να εξαχθούν συμπεράσματα για την στάση της κοινής γνώμης.

Με την χρησιμοποίηση διαφορετικών μεθόδων και τεχνικών επιχειρείται η ανίχνευση και εξαγωγή συναισθήματος από υποκειμενικές πηγές τις οποίες αποτελούν οι χρήστες του Twitter. Η στάση των ανθρώπων αυτών, οι απόψεις τους, οι επιθυμίες τους καθώς και τα σχόλια τους, διαμορφώνουν τα συναισθήματα τους ως προς το εξεταζόμενο θέμα. Η κατανόηση της κοινής γνώμης ως προς ένα θέμα δημόσιας υγείας κρίνεται ιδιαίτερα χρήσιμη. Δεδομένου ότι η πανδημία έχει άμεσες επιπτώσεις στην ζωή μας, η ανάπτυξη ενός ελπιδοφόρου εμβολίου θεωρητικά θα έπρεπε ως επί το πλείστον να δημιουργεί θετικά συναισθήματα και προσμονή για επιστροφή στην κανονικότητα. Αυτό αποτελεί και την υπόθεση, η οποία διερευνάται στα πλαίσια της εργασίας αυτής.

Για τις ανάγκες της συλλογής και επεξεργασίας των δεδομένων γίνεται η χρησιμοποίηση της γλώσσας python στην πλατφόρμα του Jupyter Notebooks σε συνδυασμό με τα πακέτα TextBlob, Gensim και Scikit-learn.

Λέξεις κλειδιά:

Big Data, Machine Learning, TextBlob, Gensim, NLP, Bag of Words, TF-IDF, Word2Vec

Abstract

The science of data analytics is booming. The advent of social media has given internet users a space to express and share their thoughts and opinions on all sorts of topics and events. Twitter, with almost 600 million users and over 250 million messages per day, is a huge platform ideal for data mining and processing. The Covid-19 pandemic does not only affect economic conditions, physical health, and working conditions, but has also created a place in the minds of people all over the world.

One of the topics that caused a lot of comments and publications was the vaccination and the attitude of people towards it. This paper collects messages from the Twitter platform related to vaccines under development to combat the pandemic, in order to analyze emotion and categorize them into classes to draw conclusions about the attitude of the world.

Using different methods and techniques, attempts are made to detect and extract emotion from subjective sources that are Twitter users. The attitude of these people, their views, their desires as well as their comments, shape their feelings regarding the issue under consideration. Understanding public opinion on a public health issue is particularly useful. Since the pandemic has a direct impact on our lives, the development of a promising vaccine should theoretically mostly create positive emotions and anticipation of a return to normalcy. This is the case and is being investigated in the context of this work.

For the needs of data collection and processing, the python language is used on the Jupyter Notebooks platform in combination with the TextBlob, Gensim and Scikit-learn packages.

Keywords:

Big Data, Machine Learning, TextBlob, Gensim, NLP, Bag of Words, TF-IDF, Word2Vec

Ευχαριστίες

Θα ήθελα να ευχαριστήσω την καθηγήτρια μου κυρία Χριστοπούλου Ελένη για την εμπιστοσύνη που μου έδειξε δίνοντάς μου τη δυνατότητα να εκπονήσω αυτή τη διπλωματική εργασία σε ένα τόσο ενδιαφέρον θέμα. Η καθοδήγηση, οι συζητήσεις που είχαμε και οι συμβουλές της ήταν ιδιαίτερα σημαντικές κατά την εκτέλεση της εργασίας.

Περιεχόμενα

Περίληψη.....	2
Abstract.....	3
Ευχαριστίες.....	5
Κατάλογος Εικόνων:	8
Κατάλογος Πινάκων:.....	8
1 Εισαγωγή	9
1.1 Big Data and Social media	9
1.2 Ανάλυση συναισθήματος.....	10
1.3 Προβλήματα και προκλήσεις.....	10
1.4 Σκοπός εργασίας.....	11
1.5 Εφαρμογές.....	12
2 Βιβλιογραφική Ανασκόπηση.....	13
2.1 Ανάλυση συναισθημάτων κοινωνικών δικτύων λόγω Covid-19 με τεχνητή νοημοσύνη	13
2.2 Ανάλυση συναισθήματος στον κορωνοϊό με την χρήση του Textblob	14
2.3 Ανάλυση συναισθήματος σε δεδομένα του Twitter	14
2.4 Ανάλυση συναισθήματος του Twitter	15
2.5 Συχνότητα των λέξεων και ανάλυση συναισθήματος κατά την διάρκεια του κορωνοϊού	16
3 Θεωρητικό υπόβαθρο	17
3.1 Python.....	17
3.2 Βιβλιοθήκες της Python	17
3.3 Twitter API.....	19
3.4 Μηχανική Μάθηση.....	19
3.5 Αλγόριθμοι Μηχανικής Μάθησης.....	20
3.4.1 K-NN	20
3.4.2 Δέντρο απόφασης- Random Forest	21
3.4.3 Multinomial Naïve Bayes.....	22
3.4.4 Neural Network- MLP	22
3.4.5 Support Vector Machine.....	23
3.4.6 Λογιστική Παλινδρόμηση	24
3.6 Εργαλείο επεξεργασίας φυσικής γλώσσας (NLTK).....	25
3.7 Textblob.....	26
3.8 Δημιουργία Διανυσμάτων	26
3.8.1 Μοντέλο Bag of Words	27
3.8.2 Μοντέλο TF-IDF (Term Frequency–Inverse Document Frequency)	27

3.8.3 Μοντέλο Word2Vec	27
4 Υλοποίηση	28
4.1 Συλλογή και αποθήκευση δεδομένων	28
4.2 Προεπεξεργασία δεδομένων	31
4.3 Ανάλυση συναισθήματος με την χρήση του TextBlob.....	33
4.4 Εύρεση ομοιοτήτων με την χρήση του Gensim.....	34
4.5 Ταξινόμηση	35
5 Αποτελέσματα	36
5.1 Μετρικές αξιολόγησης αλγορίθμων	36
5.2 Textblob.....	37
5.3 Word2Vec.....	42
5.4 Ταξινόμηση	46
5.4.1 Bag of Words.....	46
5.4.2 TF-IDF.....	51
5.4.3 Doc2vec	53
5.4.4 Word2vec.....	53
5.4.4 Δεκαπλή διασταυρωτική μέθοδος	55
6 Συμπεράσματα-Προτάσεις	56
Αναφορές.....	59
Παράρτημα	60

Κατάλογος Εικόνων:

Εικόνα 1 Σχηματική απεικόνιση κ-NN	21
Εικόνα 2 Δέντρο απόφασης.....	21
Εικόνα 3 Random Forest	22
Εικόνα 4 Κρυφό στρώμα MLP.....	23
Εικόνα 5 Δημιουργία υπερεπιπέδου	24
Εικόνα 6 Σιγμοειδής συνάρτηση	25
Εικόνα 7 Σχηματική απεικόνιση μεθοδολογίας.....	28
Εικόνα 8 Προεπεξεργασία Tweet.....	33
Εικόνα 9 TEXTBLOB.....	34
Εικόνα 10 Διάγραμμα πολικότητας και αντικειμενικότητας.....	40
Εικόνα 11 Συνδοιαστικό διάγραμμα πολικότητας και αντικειμενικότητας	41
Εικόνα 12 κατανομή συναισθήματος σε 3 κατηγορίες.....	41
Εικόνα 14 αποτέλεσμα διασταυρωτικής μεθόδου	56
Εικόνα 13 Ιεράρχηση μοντέλων ως προς αποτελεσματικότητα.....	58

Κατάλογος Πινάκων:

Πίνακας 1 Πολικότητα συναισθημάτων του Twitter	16
Πίνακας 2 Βιβλιοθήκες Python	18
Πίνακας 3 Διαδικασία αυθεντικοποίησης API.....	19
Πίνακας 4 Φράσεις κλειδιά	29
Πίνακας 5 Διαδικασία εξαγωγής Tweet	30
Πίνακας 6 Διαδικασία προ-επεξεργασίας.....	31
Πίνακας 7 Διαδικασία TextBlob	33
Πίνακας 8 Συνάρτηση δημιουργίας token.....	34
Πίνακας 9 Δημιουργία διγραμμάτων.....	35
Πίνακας 10 Συνάρτηση αντιστροφής token	36
Πίνακας 11 Συνάρτηση παραγωγής πίνακα σύγχυσης.....	36
Πίνακας 12 Πίνακας Σύγχυσης	36
Πίνακας 13 Αριθμός tweet κατά φάση.....	38
Πίνακας 14 Πολικότητα φάση 1.....	38
Πίνακας 15 Πολικότητα φάση 2.....	40
Πίνακας 16 Στάση κοινής γνώμης.....	42
Πίνακας 17 Θετικοί συσχετισμοί μοντέλου Word2Vec	43
Πίνακας 18 Ισοκατανομή δεδομένων εισόδου	47

1 Εισαγωγή

1.1 Big Data and Social media

Η εξέλιξη της τεχνολογίας με την ανάπτυξη των ηλεκτρονικών υπολογιστών, την ψηφιοποίηση της πληροφορίας και ειδικότερα την ανάπτυξη του διαδικτύου, οδήγησαν σταδιακά την ανθρωπότητα, στο να παράγει καθημερινά και να αποθηκεύει πολύ μεγάλο όγκο δεδομένων ή αλλιώς big data. Ο όρος πρωτοεμφανίστηκε το 1997 όταν επιστήμονες της Nasa ανέφεραν ότι αδυνατούσαν να αναπαραστήσουν γραφικά τα σύνολα δεδομένων που κατείχαν εξαιτίας του μεγάλου όγκου τους και της αδυναμίας αποθήκευσης τους στην κύρια μνήμη. Ψηφιακά δεδομένα τέτοιου μεγέθους δεν συναντώνται μόνο σε κάθε εμπορική δραστηριότητα όπως στους οργανισμούς αλλά πλέον και σε ατομικό επίπεδο.

Το φαινόμενο αυτό ενίσχυσε η δημιουργία κοινωνικών δικτύων όπου εκατομμύρια χρήστες αλληλοεπιδρούν μεταξύ τους με διάφορους τρόπους όπως μηνύματα, σχόλια και πολυμεσικό υλικό. Επομένως δημιουργείται μια μεγάλη τεχνολογική και επιχειρηματική πρόκληση που αφορά την αποτελεσματική διαχείριση και αξιοποίηση του τεράστιου όγκου αυτών των δεδομένων, τα οποία ξεπερνούν τις παραδοσιακές μεθόδους επεξεργασίας. Επομένως μιλάμε ουσιαστικά για δεδομένα τα οποία φέρουν 4 βασικά χαρακτηριστικά(IBM,2013):

- Όγκος (μεγάλο μέγεθος των δεδομένων)
- Ταχύτητα (ταχύτατος ρυθμός παραγωγής)
- Ποικιλία (διαφορετικοί τύποι δεδομένων)
- Πιστότητα (αξιοπιστία των δεδομένων)

Σε αυτά έχουν προστεθεί επιπλέον:

- Αξία (αν από την επεξεργασία της πληροφορίας προκύπτει αξία)
- Μεταβλητότητα (ασυνέπειες στα δεδομένα)

Το μέσο κοινωνικής δικτύωσης που χρησιμοποιείται για τους σκοπούς της παρούσα εργασίας είναι το Twitter. Αποτελεί ένα από τα πιο δημοφιλή του είδους και επιτρέπει στους χρήστες να κοινοποιούν μηνύματα (tweets) μεγέθους που δεν υπερβαίνει τους 140 χαρακτήρες. Το Twitter εξυπηρετεί 330 εκατομμύρια ενεργούς χρήστες (Statista 2020) και παράγει:

- 350.000 tweets κάθε λεπτό
- 500 εκατομμύρια tweets κάθε μέρα
- 200 δισεκατομμύρια tweets τον χρόνο

Συνεπώς αποτελεί μια τεράστια δεξαμενή δεδομένων κειμένου που προσφέρονται για συστηματική ανάλυση και εξαγωγή πληροφοριών που σχετίζονται με τις απόψεις των χρηστών και παράγουν χρήσιμα συμπεράσματα. Αυτό το αντικείμενο αποτελεί υποπεδίο της επιστήμης των δεδομένων, η οποία βρίσκεται σε άνθηση καθώς η ενσωμάτωση της τεχνολογίας στην καθημερινή ζωή έχει δημιουργήσει τεράστιες ποσότητες δεδομένων. Τα αποτελέσματα της ανάλυσης των δεδομένων χρησιμοποιούνται για εμπορικούς σκοπούς όπως η ανάλυση των επιχειρηματικών τάσεων σε μια αγορά αλλά και επιστημονικούς σκοπούς όπως η πρόληψη ασθενειών. Επί πλέον η ανάλυση δεδομένων υιοθετείται από κυβερνήσεις για διερεύνηση και διατύπωση της κοινής γνώμης αλλά και τομείς όπως η καταπολέμηση του εγκλήματος.

1.2 Ανάλυση συναισθήματος

Η ανάλυση συναισθήματος ή εξόρυξη γνώμης σε δεδομένα μεγάλου όγκου χρηστών είναι ένα δημοφιλές και αναπτυσσόμενο πεδίο ευρύτατης μελέτης, που κερδίζει όλο και μεγαλύτερο έδαφος τόσο στον επιχειρηματικό χώρο με εμπορικές εφαρμογές όσο και στον ακαδημαϊκό χώρο με πληθώρα δημοσιεύσεων. Αποτελεί σε γενικό πλαίσιο, μια διαδικασία κατά την οποία προσδιορίζεται η άποψη του συγγραφέα ενός κειμένου σε κλάσεις που εκφράζουν συναισθήματα (Despoina Chatzakou, 2015):

- Χαρά
- Λύπη
- Φόβο
- Θυμό
- Αγάπη
- Μίσος

Η δεύτερη μορφή η οποία συναντάται και στα πλαίσια αυτής της εργασίας είναι η ανίχνευση πολικότητας (θετική, ουδέτερη, αρνητική). Για να επιτευχθεί η ανάλυση συναισθήματος χρησιμοποιούνται τεχνικές επεξεργασίας **φυσικού λόγου** (natural language processing – **NLP**), στατιστικές μεθόδους και μεθόδους μηχανικής μάθησης.

Η ανάλυση συναισθήματος μπορεί να εφαρμοσθεί σε τρία επίπεδα:

- **Σε επίπεδο εγγράφου**, ακολουθώντας τον συλλογισμό ότι κάθε κείμενο εκφράζει μια μοναδική άποψη πάνω σε ένα συγκεκριμένο θέμα. Ένα τέτοιο παράδειγμα αποτελεί ένα άρθρο με κριτική ταινίας.
- **Σε επίπεδο πρότασης**, όπου κάθε πρόταση του κειμένου αποτελεί και μια ξεχωριστή άποψη. Η αναγνώριση συναισθήματος σε αυτή την περίπτωση είναι αρκετά πιο δύσκολη καθώς το νόημα είναι συμπυκνωμένο και τα δεδομένα είναι λιγότερα. Η ανάλυση των tweets που διαπραγματεύεται αυτή η εργασία ανήκει σε αυτή την κατηγορία καθώς τα tweets αποτελούνται συνήθως από δύο ή τρεις προτάσεις.
- **Σε μεμονωμένες φράσεις**. Κάθε εξεταζόμενη φράση δεδομένης μιας οντότητας παράγει ένα συναισθήμα, μια τέτοια περίπτωση εφαρμογής είναι η αναζήτηση σε χαρακτηριστικά ενός αντικειμένου σε κριτικές χρηστών π.χ. χαρακτηριστικό → οθόνη, οντότητα → κινητό

1.3 Προβλήματα και προκλήσεις

Η πολυπλοκότητα και ο θόρυβος που συναντώνται στον γραπτό λόγο, αλλά και το γεγονός ότι ακόμα και για τον ίδιο τον άνθρωπο είναι δύσκολο να εξάγει τον παλμό ενός κειμένου ή να πάρει θέση ως προς την άποψη του συγγραφέα, καθιστούν κάποια εμπόδια στο να διδάξουμε σε μια μηχανή την ανάλυση συναισθήματος. Αυτός είναι και ο λόγος που γίνονται αποδεκτές από την επιστημονική κοινότητα και έρευνες με ακρίβεια ακόμα και χαμηλότερη του 90%. Τα προβλήματα αυτά που συναντώνται στον γραπτό λόγο μπορούν να κατηγοριοποιηθούν ως εξής:

- **Αμφισημία**. Μια λέξη αναλόγως την φράση στην οποία εμπεριέχεται μπορεί να επιδέχεται πολλαπλές ερμηνείες. Η ερμηνεία της είναι εξαρτημένη από τα συμφραζόμενα.

- **Χειρισμός της Άρνησης.** Η άρνηση πολλές φορές χρησιμοποιείται στον λόγο όχι για να αντιστρέψει την πολικότητα της άποψης, αλλά για να προσδώσει έμφαση.
- **Ανάλυση της αναφοράς.** Ένα επίθετο ή μια αντωνυμία σε πιο σημείο της πρότασης αναφέρεται.
- **Υπονοούμενες δηλώσεις.** Ο σαρκασμός, η ειρωνεία και το χιούμορ δεν δημιουργούν σαφές συνθήκες διαχωρισμού και μπορεί να επηρεάζονται ανάλογα με τη κουλτούρα και το κοινωνικοπολιτικό υπόβαθρο.
- **Αναγνώριση ονομαστικών οντοτήτων.** Αφορά την κατανόηση σε τι πραγματικά αναφέρεται ένας άνθρωπος.
- **Υποθετικές προτάσεις.** Λόγω της υπόθεσης είναι δύσκολο να καθοριστεί το συναισθήμα καθώς και πολλαπλές υποθέσεις μπορεί να μην οδηγούν σε κάποια ξεκάθαρη συναισθηματική κατάσταση.
- **Σύγκρουση πολικότητας.** Όταν οι λέξεις ελέγχονται μεμονωμένα μπορεί ένας αρνητικός προσδιορισμός μιας θετικής λέξης να παράγει αρνητικό αποτέλεσμα.
- **Πολλαπλές αντικρουόμενες απόψεις.**

1.4 Σκοπός εργασίας

Η σύγχρονη εποχή με τις ραγδαίες εξελίξεις στις Τ.Π.Ε.¹ έχει διαμορφώσει μια παγκόσμια κοινωνία της πληροφορίας. Κάθε δευτερόλεπτο διακινείται τεράστιο πλήθος δεδομένων σε οποιαδήποτε μορφή. Η πρόσβαση στο διαδίκτυο έχει γίνει προσιτή για μεγάλο ποσοστό του πληθυσμού παγκοσμίως με 4.66 δισεκατομμύρια ενεργούς χρήστες (Statista, 2020). Σημαντικός παράγοντας διεύρυνσης αυτού του κοινού, αποτέλεσαν οι εφαρμογές κοινωνικής δικτύωσης, όπου αποτελούν μηχανισμούς που προσφέρουν δεδομένα πολύτιμα για επεξεργασία και εκμετάλλευση σε διάφορους τομείς όπως οικονομικές αναλύσεις, ανάλυση των τάσεων τόσο σε εμπορικό αλλά και σε κοινωνικο-πολιτικό επίπεδο καθώς και αντιμετώπιση έκτακτων καταστάσεων.

Όπως ήταν αναμενόμενο η πανδημία της νόσου του κορωνοϊού (COVID-19) οδήγησε σε ουσιαστική δημόσια συζήτηση στα κοινωνικά δίκτυα. Ένα από τα θέματα που προκάλεσε πληθώρα σχολίων και δημοσιεύσεων αποτέλεσε ο εμβολιασμός και η στάση των ανθρώπων ως προς αυτόν. Αυτές οι πληροφορίες που διακινούνται μεταξύ των χρηστών στα μέσα κοινωνικής δικτύωσης, αποτελούν ευκαιρία για εξόρυξη δεδομένων και την ανάλυση τους. Η παρούσα εργασία συγκεντρώνει μηνύματα από την πλατφόρμα Twitter τα οποία σχετίζονται με τα εμβόλια βρίσκονται σε στάδιο ανάπτυξης για την καταπολέμηση της πανδημίας, προκειμένου να πραγματοποιηθεί ανάλυση συναισθήματος και να κατηγοριοποιηθούν αυτά σε κλάσεις ώστε να εξαχθούν συμπεράσματα για την στάση του κόσμου.

Με την χρησιμοποίηση διαφορετικών μεθόδων και τεχνικών επιχειρείται η ανίχνευση και εξαγωγή συναισθήματος από υποκειμενικές πηγές τις οποίες αποτελούν οι χρήστες του Twitter. Η στάση των ανθρώπων αυτών, οι απόψεις τους, οι επιθυμίες τους καθώς και τα σχόλια τους, διαμορφώνουν τα συναισθήματα τους ως προς το εξεταζόμενο θέμα. Η κατανόηση της κοινής γνώμης ως προς ένα θέμα δημόσιας υγείας κρίνεται ιδιαίτερα χρήσιμη. Δεδομένου ότι η πανδημία έχει άμεσες επιπτώσεις στην ζωή μας, η ανάπτυξη ενός ελπιδοφόρου εμβολίου θεωρητικά θα έπρεπε ως επί το πλείστο να δημιουργεί θετικά συναισθήματα και προσμονή για επιστροφή στην κανονικότητα. Αυτή η υπόθεση διερευνάται στα πλαίσια της εργασίας αυτής.

¹ Τεχνολογίες της πληροφορίας και της επικοινωνίας

1.5 Εφαρμογές

Έντονη παρατηρείται τα τελευταία χρόνια η δημιουργία εφαρμογών με την χρήση τεχνικών ανάλυσης συναισθήματος σε **αδόμητο κείμενο**. Μάλιστα τα έτοιμα εργαλεία όπως το Quick Search² παρέχονται σε ένα μηνιαίο κόστος της τάξης των 800\$. Για πολλές εταιρείες η εξόρυξη γνώμης από κείμενο αποτελεί αναπόσπαστο μέρος της έρευνας τους, προκειμένου να κατανοήσουν καλύτερα το καταναλωτικό κοινό ως προς τα προϊόντα που προωθούν. Σκοπός είναι να βρίσκονται σε συνεχή ενημέρωση για τις απόψεις και το ενδιαφέρον του κοινού τους απέναντι στα προϊόντα τους. Οι κυριότερες εφαρμογές που λαμβάνει η ανάλυση συναισθήματος για εμπορικούς σκοπούς είναι:

- Παρακολούθηση κοινωνικών δικτύων.

Πολλές από τις δημοσιεύσεις που λαμβάνουν χώρα σε αυτά τα δίκτυα αφορούν εμπορικά προϊόντα. Επί πλέον μπορούν να αντλούνται πληροφορίες για πολιτικούς σκοπούς και βολιδοσκόπησης της κοινής γνώμης. Μέσω αυτής της μεθόδου λαμβάνονται τακτικές, αξιόπιστες πληροφορίες 24/7 και σε πραγματικό χρόνο ώστε μια εταιρεία να μπορεί να παρακολουθεί την πρόοδο της.

- Υποστήριξη πελατών.

Οι προκλήσεις που προέρχονται λόγω του μεγάλου όγκου αιτημάτων που δέχονται οι εταιρείες έχουν οδηγήσει σε αυτοματοποιημένες μεθόδους κατανόησης της φυσικής γλώσσας και του επείγοντος χαρακτήρα του συγκεκριμένου αιτήματος ώστε να διευκολύνεται η διαδικασία και να δρομολογούνται στον κατάλληλο αποδέκτη.

- Σχόλια πελατών.

Η ανάλυση συναισθήματος μπορεί να χρησιμοποιηθεί επίσης για την απόκτηση πληροφοριών από τα διαθέσιμα σχόλια των πελατών και να εξαγεί άμεσα συμπεράσματα που αντανακλούν στον σχεδιασμό του συγκεκριμένου προϊόντος ή υπηρεσίας.

- Παρακολούθηση επωνυμίας.

Η παρακολούθηση μιας επωνυμίας είναι μια από τις πιο δημοφιλείς εφαρμογές της ανάλυσης συναισθημάτων στον χώρο των επιχειρήσεων. Όχι μόνο παρέχετε η δυνατότητα να παρακολουθείται δυναμικά η φήμη της εταιρείας, αλλά παρακολουθούνται και οι ανταγωνιστές. Κατά αυτό τον τρόπο μπορεί να προκύψουν επιχειρηματικές ευκαιρίες και να τις εκμεταλλευτούν τα διοικητικά στελέχη την κατάλληλη στιγμή.

- Ανάλυση προϊόντος.

Άμεση γνώση για την άποψη του καταναλωτικού κοινού πάνω σε ένα νέο προϊόν αμέσως μετά την κυκλοφορία του μέσω της ανάλυσης των σχολίων και της αναφοράς του σε δημοσιεύσεις χρηστών. Μέσω αυτής της μεθόδου εξοικονομείται χρόνος και χρήμα από μακροσκελείς

² Εργαλείο αξιολόγησης feedback χρηστών

παραδοσιακές έρευνες και λαμβάνονται άμεσα αποτελέσματα για το πια στοιχεία του προϊόντος πρέπει να βελτιωθούν με βάση τα αποτελέσματα της ανάλυσης.

- Έρευνα αγοράς και ανταγωνιστών.

Παρέχεται η δυνατότητα ανάλυσης των ανταγωνιστών και σύγκρισης των επιμέρους προσπαθειών μάρκετινγκ πάνω στην πελατειακή βάση του ανταγωνισμού. Έτσι ανακαλύπτονται τρόποι που ενδεχομένως δεν είχαν εξεταστεί και βοηθάνε καλύτερα τις επιχειρήσεις να αντιληφθούν τα δυνατά σημεία αλλά και τις αδυναμίες τους.

- Διαχείριση μιας κρίσης.

Επειδή η ανάλυση συναισθήματος επιτρέπει την παρακολούθηση των καταναλωτών σε πραγματικό χρόνο, αν προκύψει μια ξαφνική πτώση ως προς την εξεταζόμενη επωνυμία τότε η εταιρεία θα χτυπήσει καμπανάκια προειδοποίησης και θα λάβει μέτρα ώστε να αποφύγει την καταστροφή.

2 Βιβλιογραφική Ανασκόπηση

Στο κεφάλαιο αυτό παρατίθενται επιστημονικά άρθρα με συνάφεια ως προς το θέμα της παρούσας έρευνας και την ανάλυση κειμένου.

2.1 Ανάλυση συναισθημάτων κοινωνικών δικτύων λόγω Covid-19 με εφαρμογή τεχνητής νοημοσύνης (Hung et al,2020)

Σκοπός του άρθρου είναι να κατανοηθούν οι συζητήσεις που δημιουργήθηκαν στα κοινωνικά δίκτυα εξαιτίας της πανδημίας, ώστε να μπορέσουν να βοηθήσουν τα ιδρύματα, τις κυβερνήσεις και τα άτομα για να μπορέσουν να ξεπεράσουν την πανδημία. Στην έρευνα αυτή εφαρμόστηκαν μέθοδοι μηχανικής μάθησης στον τομέα της τεχνητής νοημοσύνης για την ανάλυση δεδομένων που συλλέχθηκαν από το Twitter. Τα δεδομένα προέχονται αποκλειστικά από τις Ηνωμένες Πολιτείες και είναι γραμμένα στην αγγλική γλώσσα από αρχές Μαρτίου έως τέλη Απριλίου 2020.

Αποτελέσματα: Υπήρχαν συνολικά 14.180.603 επισημάνσεις "μου αρέσει", 863.411 απαντήσεις, 3.087.812 retweets και 641.381 αναφορές σε tweets κατά τη διάρκεια του χρονοδιαγράμματος της μελέτης. Από τα 902.138 tweets που αναλύθηκαν, η ανάλυση συναισθημάτων ταξινόμησε 434.254 (48,2%) tweets ως θετικά συναισθήματα, 187.042 (20,7%) ως ουδέτερο και 280.842 (31,1%) ως αρνητικά. Η μελέτη εντόπισε 5 κυρίαρχα θέματα που σχετίζονται με το COVID-19: περιβάλλον υγειονομικής περίθαλψης, συναισθηματική υποστήριξη, επιχειρηματική οικονομία, κοινωνική αλλαγή και ψυχολογική υγεία.

Συμπεράσματα: Τα «Trump», «η μάσκα» και το «νοσοκομείο» κυριάρχησαν στα tweets της υγείας. Στο περιβάλλον φροντίδας τα «Εβδομάδα», «άτομα», «σπίτι», «εργασία» και στην οικονομία των επιχειρήσεων «ανάγκη». Στο ψυχολογικό άγχος, οι «άνθρωποι», «θα» και «ιός» κυριάρχησαν στη συζήτηση. Συνολικά, τα θετικά tweets υπερέβησαν τα αρνητικά tweets. Τα συναισθήματα μπορούν να αποσαφηνίσουν τη δημόσια συζήτηση στο COVID-19 και να βοηθήσουν στην καθοδήγηση κυβερνητικούς αξιωματούχους, ιδιωτικούς φορείς και το κοινό με πληροφορίες καθώς διανύουμε την πανδημία.

2.2 Ανάλυση συναισθήματος στον κορωνοϊό με την χρήση του Textblob (Chhinder Kaur, 2020)

Σκοπός του άρθρου είναι να αναλυθούν τα συναισθήματα που παράγονται καθημερινά στα μέσα κοινωνικής δικτύωσης σχετικά με την πανδημία του κορωνοϊού (Covid-19). Επειδή πολλοί λαοί διαφορετικών χωρών επηρεάζονται από την πανδημία, αναλύονται απόψεις διαφορετικών ανθρώπων και για αυτό χρησιμοποιείται το Api για την απόκτηση των δεδομένων και στη συνέχεια αναλύονται τα tweets με τεχνικές μηχανικής μάθησης ως αρνητικά, θετικά και ουδέτερα.

Τα αποτελέσματα της ταξινόμησης των tweets σε πλήθος 2058:

- Θετικά 24%
- Αρνητικά 32.1%
- Ουδέτερα 43.9%

Ο κύριος λόγος αυτής της εργασίας είναι ανάλυση δεδομένων που προέρχονται από το Twitter. Το API του Twitter συγκεντρώνει αυτά τα δεδομένα και έχει εφαρμοστεί η γλώσσα προγραμματισμού python στο περιβάλλον του Jupyter notebook για να εφαρμόσουν τη συναισθηματική ανάλυση με κατηγοριοποίηση ως θετική, αρνητική και ουδέτερη. Τα αποτελέσματα εμφανίζονται χρησιμοποιώντας τη βιβλιοθήκη Matplotlib.

Συμπεράσματα: Διαπιστώθηκε ότι τα ουδέτερα συναισθήματα είναι σημαντικά υψηλά, γεγονός που δείχνει ότι υπάρχει ανάγκη βελτίωσης της ανάλυσης συναισθημάτων του Twitter.

2.3 Ανάλυση συναισθήματος σε δεδομένα του Twitter (Apoorv Agarwal, 2011)

Τα κοινωνικά δίκτυα έχουν εξελιχθεί σε πηγή ποικίλου είδους πληροφοριών. Αυτό οφείλεται στο ότι άνθρωποι δημοσιεύουν μηνύματα σε πραγματικό χρόνο σχετικά με τις απόψεις τους για διάφορα θέματα, συζητούν τρέχοντα ζητήματα, παραπονούνται και εκφράζουν θετικά συναισθήματα για προϊόντα που χρησιμοποιούν στην καθημερινή ζωή. Στην πραγματικότητα, οι εταιρείες που κατασκευάζουν τέτοια προϊόντα έχουν αρχίσει να κάνουν δημοσκοπήσεις στα κοινωνικά δίκτυα για να πάρουν μια αίσθηση του γενικού αισθήματος για το προϊόν τους. Πολλές φορές αυτές οι εταιρείες μελετούν τις αντιδράσεις των χρηστών και απαντούν σε χρήστες.

Σκοπός του άρθρου είναι να οικοδομήσει μια τεχνολογία για να ανιχνευθεί και να καταγραφεί ένα συνολικό συναίσθημα. Σε αυτό το άρθρο, εξετάζετε το Twitter και δημιουργούνται μοντέλα για την ταξινόμηση των tweets σε θετικά, αρνητικά και ουδέτερα συναισθήματα. Υλοποιούνται δυο διαδικασίες ταξινόμησης. Μια δυαδική σε αρνητικά και θετικά και μια κατηγορική σε θετικά αρνητικά και ουδέτερα συναισθήματα. Τα μοντέλα που χρησιμοποιούνται είναι: μοντέλο unigram, μοντέλο βασισμένο σε πυρήνα δέντρου και μοντέλο βασισμένο σε χαρακτηριστικά. Για το μοντέλο που βασίζεται στον πυρήνα δέντρων σχεδιάζετε μια νέα αναπαράσταση δέντρων για τα tweets.

Αποτελέσματα: Τα πειράματα δείχνουν ότι το μοντέλο unigram είναι πράγματι μια τεχνική η οποία λειτουργεί καλά για την ανάλυση συναισθημάτων σε μικρά μηνύματα όπως του Twitter. Το μοντέλο που βασίζεται σε χαρακτηριστικά και χρησιμοποιεί μόνο 100 επιτυγχάνει παρόμοια ακρίβεια με το μοντέλο unigram που χρησιμοποιεί πάνω από 10.000 χαρακτηριστικά. Το μοντέλο που βασίζεται στον πυρήνα του δέντρου ξεπερνά και τα δύο αυτά μοντέλα με σημαντικό περιθώριο. Πειραματισμός επίσης με ένα συνδυασμό μοντέλων:

συνδυασμός unigrams και συνδυασμός των χαρακτηριστικών με τον πυρήνα του δέντρου. Και οι δύο αυτοί συνδυασμοί ξεπερνούν την αρχική γραμμή unigram κατά περισσότερο από 4% και για τις δύο εργασίες ταξινόμησης. Τα πειράματά μας δείχνουν ότι οι λειτουργίες που σχετίζονται με συγκεκριμένες λειτουργίες του Twitter (emoticon, hashtags κ.λπ.) προσθέτουν αξία στον ταξινομητή αλλά μόνο οριακά.

Συμπεράσματα: Αναλύθηκαν και παρουσιάστηκαν τα αποτελέσματα 3 διαφορετικών μοντέλων. Προτείνεται η χρησιμοποίηση του μοντέλου unigram ως βάση και αναφέρεται ένα συνολικό κέρδος άνω του 4% για τις δύο εργασίες ταξινόμησης. Παρουσιάστηκε μια ολοκληρωμένη σειρά πειραμάτων για τις δύο αυτές εργασίες σε δεδομένα με μη αυτόματο σχολιασμό που είναι ένα τυχαίο δείγμα ροής tweets. Η διερεύνηση των μοντέλων του πυρήνα δέντρου και μοντέλου βασισμένο σε χαρακτηριστικά απόδειξε ότι ξεπερνούν την αρχική γραμμή του unigram. Η ανάλυση χαρακτηριστικών αποκαλύπτει ότι τα πιο σημαντικά χαρακτηριστικά είναι αυτά που συνδυάζουν την προηγούμενη πολικότητα των λέξεων και τις ετικέτες των μερών τους. Καταλήγουν προσωρινά στο συμπέρασμα ότι η ανάλυση συναισθημάτων για τα δεδομένα του Twitter δεν διαφέρει από την ανάλυση συναισθημάτων για άλλα είδη κειμένου.

2.4 Ανάλυση συναισθήματος του Twitter (Saif Hassan, 2012)

Η εμφάνιση των κοινωνικών μέσων ενημέρωσης έδωσε στους χρήστες του διαδικτύου έναν χώρο για να εκφράσουν και να μοιραστούν τις σκέψεις και τις απόψεις τους για κάθε είδους θέματα και εκδηλώσεις. Twitter, με σχεδόν 600 εκατομμύρια χρήστες και πάνω από 250 εκατομμύρια μηνύματα την ημέρα, έχουν γίνει γρήγορα ένα ορυχείο χρυσού για τους οργανισμούς να παρακολουθούν τη φήμη και τις επωνυμίες τους εξάγοντας και αναλύοντας το συναίσθημα των Tweets που δημοσιεύονται από το κοινό σχετικά με αυτά, τις αγορές τους και τους ανταγωνιστές τους.

Η ανάλυση συναισθημάτων μέσω του Twitter προσφέρει στους οργανισμούς έναν γρήγορο και αποτελεσματικό τρόπο για να παρακολουθούν τα συναισθήματα των πολιτών έναντι της επωνυμίας τους, των επιχειρήσεων, των διευθυντών τους κ.λπ. Ένα ευρύ φάσμα χαρακτηριστικών και μεθόδων για την ταξινόμηση συναισθημάτων για ομάδες δεδομένων του Twitter έχουν ερευνηθεί τα τελευταία χρόνια με ποικίλα αποτελέσματα.

Σε αυτό το άρθρο, εισάγεται μια νέα προσέγγιση της προσθήκης σημασιολογίας ως πρόσθετων χαρακτηριστικών στο εκπαιδευτικό σύνολο για την ανάλυση συναισθημάτων. Για κάθε εξαχθείσα οντότητα (π.χ. iPhone) από τα tweets, προστίθεται η σημασιολογική της έννοια (π.χ. "προϊόν Apple") ως πρόσθετο χαρακτηριστικό και μετριέται η συσχέτιση της αντιπροσωπευτικής έννοιας με αρνητικό / θετικό συναίσθημα. Εφαρμόζουμε αυτήν την προσέγγιση για την πρόβλεψη συναισθημάτων για τρία διαφορετικά σύνολα δεδομένων Twitter. Συγκρίνεται επίσης μια προσέγγιση που βασίζεται στην ανάλυση θεμάτων συναισθημάτων και διαπιστώνεται ότι τα σημασιολογικά χαρακτηριστικά παράγουν καλύτερη βαθμολογία ανάκλησης και F-αρμονικής κατά την ταξινόμηση αρνητικών συναισθημάτων και καλύτερη ακρίβεια με χαμηλότερη βαθμολογία ανάκλησης και F-αρμονικής στην ταξινόμηση θετικών συναισθημάτων.

Τα αποτελέσματά μας δείχνουν μια μέση αύξηση της βαθμολογίας ακρίβειας της F-αρμονικής για τον εντοπισμό τόσο του αρνητικού όσο και του θετικού συναισθήματος περίπου 6,5% και 4,8% κατά τη διάρκεια των βασικών χαρακτηριστικών των unigrams και των λειτουργιών της μερικής ομιλίας αντίστοιχα.

Τα συμπεράσματα δείχνουν ότι το σημασιολογικό μοντέλο χαρακτηριστικών ξεπερνά τη γραμμή βάσης των Unigram και POS για τον εντοπισμό και των δύο (αρνητικών και θετικών συναισθημάτων). Αποδείχθηκε ότι η προσθήκη σημασιολογικών χαρακτηριστικών παράγει υψηλότερα αποτελέσματα. Επίσης η χρήση σημασιολογικών χαρακτηριστικών ξεπερνά τα χαρακτηριστικά συναισθημάτων-θεμάτων για ταξινόμηση

θετικών συναισθημάτων όσον αφορά την ακρίβεια, αλλά όχι όσον αφορά την ανάκληση και το F1-score. Κατά μέσο όρο, τα σημασιολογικά χαρακτηριστικά φαινόταν να είναι πιο ακριβή μεταξύ των τεσσάρων άλλων επιλογών χαρακτηριστικών στον πειραματικό χώρο. Τα αποτελέσματά μας δείχνουν ότι η σημασιολογική προσέγγιση είναι πιο κατάλληλη όταν τα σύνολα δεδομένων που αναλύονται είναι μεγάλα και καλύπτουν ένα ευρύ φάσμα θεμάτων, ενώ η προσέγγιση συναίσθημα-θέμα ήταν πιο κατάλληλη για σχετικά μικρά σύνολα δεδομένων με συγκεκριμένες τοπικές εστίες.

2.5 Συχνότητα των λέξεων και ανάλυση συναισθήματος κατά την διάρκεια της πανδημίας του κορωνοϊού (Nikhil Rajput, 2020)

Οι καταστροφικές παγκόσμιες συνθήκες έχουν έντονη επίδραση στη ζωή των ανθρώπων σε ολόκληρο τον κόσμο. Οι συνέπειες ενός τέτοιου σεναρίου βιώνονται με διαφορετικούς και πολλαπλασιαστικούς τρόπους που καλύπτουν εργασίες, μέσα και αναφορές ειδήσεων, επιζήμια σωματική και ψυχική υγεία, καθώς και συνομιλίες ρουτίνας. Η επιδημία δεν έχει δημιουργήσει μόνο όλεθρο στις οικονομικές συνθήκες, φυσική υγεία, συνθήκες εργασίας και στον μεταποιητικό τομέα, αλλά έχει επίσης δημιουργήσει μια θέση στο μυαλό των ανθρώπων παγκοσμίως.

Η συνειδητοποίηση του κόσμου σχετικά με την πανδημία και η αναγνώριση της παρουσίας της, προκάλεσε αισθητή αύξηση στον αριθμό των αναρτήσεων στα κοινωνικά δίκτυα. Αυτή η εργασία βρίσκει τόπο εφαρμογής στο Twitter και παρουσιάζει μια στατιστική ανάλυση των μηνυμάτων του twitter που σχετίζονται με την ασθένεια του κορωνοϊού.

Υλοποιούνται δύο μέρη, το πρώτο αφορά την συχνότητα των λέξεων με την χρησιμοποίηση ν-γραμμάτων όπου το ν λαμβάνει τιμές από 1 έως 3 και το δεύτερο μέρος αφορά τα συναισθήματα των μεμονωμένων μηνυμάτων. Η επιθεώρηση της συχνότητας είναι χρήσιμη για τον χαρακτηρισμό των τάσεων που κυριαρχούν στο εν λόγω κοινωνικό δίκτυο, ενώ η αναγνώριση συναισθημάτων σε αυτά αντανακλάει την ψυχολογία των χρηστών.

Τα αποτελέσματα δείχνουν ότι η πλειονότητα των tweets είχε θετική πολικότητα και μόνο περίπου το 15% ήταν αρνητικά.

Πίνακας 1 Πολικότητα συναισθημάτων του Twitter

Sentiment Polarity	Positive	Neutral	Negative
Tweets by WHO ³	60.27%	24.05%	15.68%
Tweets by general public	29.33%	54.92%	15.75%

Ο αριθμός των χρηστών που κάνει tweet για τον Coronavirus αυξήθηκε ραγδαία με αρκετές κορυφές κατά τη διάρκεια του Φεβρουαρίου και του Μαρτίου. Μια εμπειρική ανάλυση των λέξεων στα μηνύματα υλοποιείται καθορίζοντας τις συχνότητες εμφάνισής τους. Οι πιο συχνές λέξεις ήταν Coronavirus, Covid19 και Wuhan. Ως προς την πολικότητα παρατηρήθηκε ότι τα περισσότερα από τα μηνύματα ανήκαν σε ουδέτερες και θετικές κατηγορίες. Με τα μηνύματα του ΠΟΥ, το 60% των μηνυμάτων βρέθηκε να είναι θετικά και γενικά μηνύματα, το 29% βρέθηκε να είναι θετικό και το 54% ουδέτερο. Και στις δύο περιπτώσεις, μόλις το 15% των μηνυμάτων

³ World health organization

ήταν αρνητικά. Τα αποτελέσματα που προέκυψαν αποτελούν μια υπέροχη αντανάκλαση των συναισθημάτων που εκφράστηκαν παγκοσμίως κατά τη διάρκεια αυτής της πανδημίας.

3 Θεωρητικό υπόβαθρο

Στο κεφάλαιο αυτό αναλύονται τα εργαλεία και οι βασικές έννοιες πίσω από αυτά, που χρησιμοποιήθηκαν για τους σκοπούς της εκπόνησης της συγκεκριμένης εργασίας. Η επιστήμη των big data και πιο συγκεκριμένα η ανάλυση συναισθήματος, απαιτεί γνώσεις από διαφορετικά επιστημονικά πεδία σε αρμονική συνεργασία με προγραμματισμό. Βέβαια υπάρχουν έτοιμα λογισμικά όπως το WEKA⁴, τα οποία έχουν αναπτυχθεί ειδικά για τον σκοπό της ανάλυσης δεδομένων μέσω της μηχανικής μάθησης. Ωστόσο ένα προγραμματιστικό περιβάλλον όπως αυτό της Python προσφέρει ευελιξία στον χρήστη και μπορεί να συνδυάσει πλήθος διαφορετικών εργαλείων ανοικτού κώδικα, προσδίδοντας κατά αυτόν τον τρόπο ένα πιο ολοκληρωμένο τρόπο.

3.1 Python

Η Python είναι μια δημοφιλής γλώσσα προγραμματισμού υψηλού επιπέδου - γενικού σκοπού που παρέχει πολλές δυνατότητες. Χαρακτηρίζεται ως μια απλή γλώσσα που στην σύνταξη της μοιάζει αρκετά με ψευδογλώσσα. Παρέχει το πλεονέκτημα ότι ο προγραμματιστής δεν χρειάζεται να μεταγλωττίσει το πρόγραμμα, αλλά μπορεί να το εκτελεί εντολή προς εντολή. Ο ιδιαίτερος χειρισμός των strings από την γλώσσα και τα διάφορα modules ανοικτού κώδικα με συνεχείς βελτιώσεις την καθιστούν ιδιαίτερα επιλέξιμη ακόμα και σε διαδικασίες όπως η ανάλυση κειμένου και αποτελεί συνήθης επιλογή στην ανάλυση δεδομένων.

Για τις ανάγκες της συγγραφής του κώδικα χρησιμοποιήθηκε το περιβάλλον του Jupyter Notebook⁵ όπου είναι μια πλατφόρμα που λαμβάνει ευρύ κοινό στην ανάλυση δεδομένων. Είναι ένα διαδραστικό περιβάλλον όπου έχεις την δυνατότητα να παρεμβάλεις ανάμεσα στα κελιά του κώδικα, κείμενο σε μορφοποιημένη μορφή κάνοντας πιο περιγραφική την ανάλυση και προσδίδοντας λεπτομέρειες που θες να καταγράψεις αμέσως μετά την εκτέλεση των πειραμάτων.

3.2 Βιβλιοθήκες της Python

Για να μπορέσει να λάβει χώρα η ανάλυση των δεδομένων, είναι απαραίτητη η εισαγωγή κάποιων βιβλιοθηκών της Python οι οποίες μας παρέχουν τα κατάλληλα εργαλεία ώστε να μπορέσουμε να υλοποιήσουμε την επεξεργασία. Οι βιβλιοθήκες που χρησιμοποιήθηκαν στον κώδικα παρουσιάζονται περιγραφικά υπό μορφή πίνακα:

⁴ Waikato Environment for Knowledge Analysis

⁵ Περιβάλλον ανάπτυξης ανοικτού κώδικα

Πίνακας 2 Βιβλιοθήκες Python

Όνομα βιβλιοθήκης Python	Περιγραφή
Sklearn	Παρέχει τους αλγορίθμους μηχανικής μάθησης που χρησιμοποιούνται στην εργασία για την ταξινόμηση των tweets σε κλάσεις. Για τη λειτουργία του είναι βασισμένο στα πακέτα Numpy, Pandas και Matplotlib.
Gensim	Για την μοντελοποίηση θεμάτων και την επεξεργασία φυσικής γλώσσας, χρησιμοποιώντας σύγχρονη στατιστική μηχανική εκμάθηση. Έχει σχεδιαστεί για να χειρίζεται μεγάλες συλλογές κειμένων χρησιμοποιώντας ροή δεδομένων και αυξητικούς διαδικτυακούς αλγόριθμους.
Pandas	Χρησιμοποιείται για τον χειρισμό και την ανάλυση δεδομένων. Προσφέρει δομές δεδομένων και λειτουργίες για χειρισμό πινάκων και χρονοσειρών.
Numpy	Δημιουργία και διαχείριση πινάκων τύπου Numpy array, καθώς διευκολύνουν προηγμένους μαθηματικούς τύπους σε μεγάλο αριθμό δεδομένων. Μειώνουν την ταχύτητα εκτέλεσης και εξοικονομούν κώδικα.
Multiprocessing	Για την εκτέλεση υπολογισμών σε περισσότερο από έναν πυρήνες των υπολογιστικών πόρων του συστήματος που τρέχει ο κώδικας.
Matplotlib/Seaborn	Εργαλεία απεικόνισης δεδομένων και αποτελεσμάτων σε διαγράμματα. Διαθέτουν πληθώρα επιλογών για βελτίωση των οπτικών αποτελεσμάτων και εφαρμογή διαφορετικών επιλογών απεικόνισης.
WordCloud	Ειδικός τύπος διαγράμματος για την απεικόνιση String σε διάγραμμα. Χρησιμοποιείται για την απεικόνιση των πιο συχνών λέξεων σε ένα δείγμα.
Warnings	Πακέτο για την απαλοιφή προειδοποιήσεων κατά την διαδικασία της απεικόνισης των αποτελεσμάτων σε διαγράμματα. Χρησιμοποιείται μόνο για αισθητικούς σκοπούς για βελτίωση της απεικόνισης στον χρήστη.
Re	Κανονικές εκφράσεις για την διαχείριση των String. Έχουν δικό τους συντακτικό το οποίο επιτρέπει στον χρήστη ενέργειες όπως η απαλοιφή των αριθμών από ένα κείμενο, ειδικών συμβόλων κλπ.
Emoji	Αποτελεί έναν από τους τρόπους μέσω έτοιμης μεθόδου για την απαλοιφή των emoticon από το κείμενο.
Os	Απαραίτητη βιβλιοθήκη για τη καθοδήγηση στο σύστημα, κατά παρόμοιο τρόπο με το τερματικό του υπολογιστικού συστήματος.
Termcolor/Sys	Για προσθήκη χρώματος, έντονης γραφής, υπογράμμισης και πλάγιας γραφής στην εκτύπωση ενός string στην οθόνη.

3.3 Twitter API

Ένα API (Application Programming Interface) είναι το ενδιάμεσο λογισμικό που επιτρέπει την επικοινωνία μεταξύ δύο εφαρμογών. Αναλαμβάνει να παραδώσει ένα αίτημα από τον χρήστη στον πάροχο, να αντλήσει πληροφορίες και στη συνέχεια να τις επιστρέψει στον χρήστη. Υπάρχουν πολλά API ανοιχτού κώδικα αναλόγως το τι θέλει να κάνει ο χρήστης. Ένα Twitter API παρέχει την δυνατότητα στον χρήστη να διαβάζει και να γράφει δεδομένα από την πλατφόρμα. Με αυτόν τον τρόπο μπορεί να αναζητήσει Tweets με συγκεκριμένο περιεχόμενο και σε μεγάλο αριθμό.

Για την επικοινωνία με το API του Twitter χρησιμοποιήθηκε το Tweepy (Roesslein, 2009-2021). Αποτελεί μια βιβλιοθήκη ανοικτού κώδικα της Python, η οποία παρέχει εύκολη πρόσβαση στις υπηρεσίες του Twitter μέσω των διαφορετικών μεθόδων της. Απαραίτητο στάδιο για την χρησιμοποίηση της είναι η απόκτηση των API Key, API Secret καθώς και ένα access token διότι προϋποθέτει OAuth Authentication⁶. Για την απόκτηση των παραπάνω κλειδιών δημιουργήθηκε ένα Twitter application στον ιστότοπο <https://developer.twitter.com>

Πίνακας 3 Διαδικασία αυθεντικοποίησης API

```
# Authenticate to Twitter
auth = tweepy.OAuthHandler(consumer_key, consumer_secret)
auth.set_access_token(access_token, access_token_secret)
api = tweepy.API(auth, wait_on_rate_limit=True, wait_on_rate_limit_notify=True,
parser=tweepy.parsers.JSONParser())# test authentication
try:
    api.verify_credentials()
    print("Authentication OK")
except:
    print("Error during authentication")
```

Ως είσοδος δίνονται οι παράμετροι που δημιουργήθηκαν για τον authentication handler, ενώ οι τιμές wait_on_rate=True και wait_on_rate_limit_notify=True αφορούν τους περιορισμούς που επιβάλλει το Twitter API σε λογαριασμό με μόνο δικαιώματα την αναζήτηση. Πιο συγκεκριμένα τα rate limits ενεργοποιούνται όταν το API στέλνει αιτήματα και υπερβεί τα 15 λεπτά. Τότε το API επιστρέφει response code HTTP 429.

3.4 Μηχανική Μάθηση

Η Μηχανική Μάθηση είναι ένα πεδίο της Επιστήμης των Υπολογιστών που παρέχει την δυνατότητα στους υπολογιστές να υλοποιούν αλγόριθμους σε δεδομένα, να μαθαίνουν και να κάνουν προβλέψεις πάνω στα δεδομένα αυτά. Οι αλγόριθμοι αυτοί χωρίζονται σε 2 κατηγορίες αναλόγως με το είδος της μάθησης:

- **Επιβλεπόμενη.** Οι αλγόριθμοι εκπαιδεύονται πάνω σε δείγματα δεδομένων για τα οποία γνωρίζουν την επιθυμητή έξοδο τους και χρησιμοποιούν την γνώση που απέκτησαν σε καινούργια δεδομένα εισόδου για τα οποία η έξοδος είναι άγνωστη και κάνουν προβλέψεις.
- **Μη επιβλεπόμενη.** Στην περίπτωση αυτή οι αλγόριθμοι εκπαιδεύονται χωρίς να γνωρίζουν τις επιθυμητές εξόδους αλλά προσπαθούν να βρουν συσχετίσεις χωρίζοντας τα δεδομένα σε ομάδες. Στην

⁶ <https://oauth.net/articles/authentication>

συνέχεια λαμβάνουν καινούργια είσοδο και προσπαθούν να τα τοποθετήσουν τα νέα δεδομένα στις ομάδες που δημιούργησαν στο προηγούμενο στάδιο.

Ένα από τα προβλήματα που καλούνται να επιλύσουν οι αλγόριθμοι της επιβλεπόμενης μάθησης είναι αυτό της **ταξινόμησης**. Σκοπός της ανάλυσης της εργασίας είναι να ταξινομηθούν τα tweets αναλόγως σε μια κατηγορία συναισθήματος ανήκουν (αρνητικά-θετικά-ουδέτερα). Κατά τη διαδικασία της ταξινόμησης, ο αλγόριθμος που χρησιμοποιείται για την ταξινόμηση προσπαθεί να αναγνωρίσει σε μια κατηγορία ανήκει ένα δείγμα δεδομένων. Αρχικά ο αλγόριθμος τροφοδοτείται με ένα μεγάλο δείγμα tweets που γνωρίζει σε μια κατηγορία ανήκουν και στην συνέχεια προσπαθεί να αναγνωρίσει την κατηγορία των νέων μηνυμάτων.

3.5 Αλγόριθμοι Μηχανικής Μάθησης

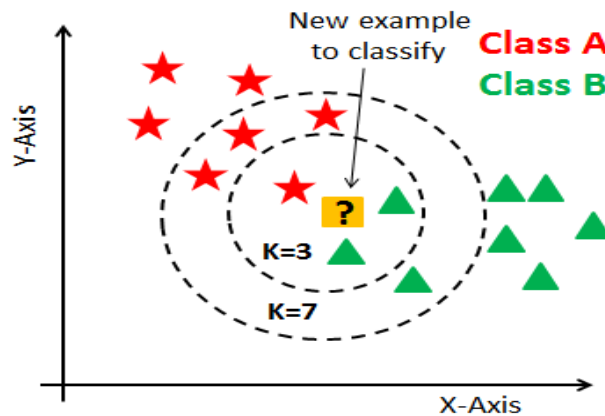
Για τις ανάγκες της ανάλυσης των δεδομένων γίνεται χρήση διαφορετικών αλγορίθμων ώστε να βρεθεί ο αλγόριθμος που παρουσιάζει τις καλύτερες μετρικές αξιολόγησης στα εξεταζόμενα δεδομένα. Στην ανάλυση δεδομένων δεν υπάρχει a priori συνταγή της επιτυχίας στην επιλογή ενός μοντέλου αλλά χρειάζονται πολλές δοκιμές και χάραξη στρατηγικής.

3.4.1 K-NN

Μια πολύ γνωστή και απλή τεχνική κατηγοριοποίησης που βασίζεται στη χρήση μέτρων βασισμένων στην απόσταση, είναι αυτή των k κοντινότερων γειτόνων (**k-nearest neighbors algorithm**). Σε αυτή τη μέθοδο της μάθησης κατά περίπτωση, όταν το σύστημα καλείται να αποφασίσει για την κατηγορία μιας νέας περίπτωσης εξετάζει εκείνη τη στιγμή τη σχέση της με τα ήδη αποθηκευμένα παραδείγματα. Απαραίτητη προϋπόθεση είναι ότι το σύνολο της εκπαίδευσης πρέπει να περιλαμβάνει και την επιθυμητή κατηγοριοποίηση για κάθε στοιχείο. Για να γίνει η κατηγοριοποίηση ενός νέου στοιχείου πρέπει να καθοριστεί η μέθοδος της απόστασης του από κάθε στοιχείο του συνόλου εκπαίδευσης και ο αριθμός των k πιο κοντινών στοιχείων εκπαίδευσης, τα οποία αποτελούν τους γείτονες του. Για την μέτρηση της απόστασης στον πειραματικό χώρο ορίστηκε αυτή της Ευκλείδειας απόστασης (1).

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2} \quad (1)$$

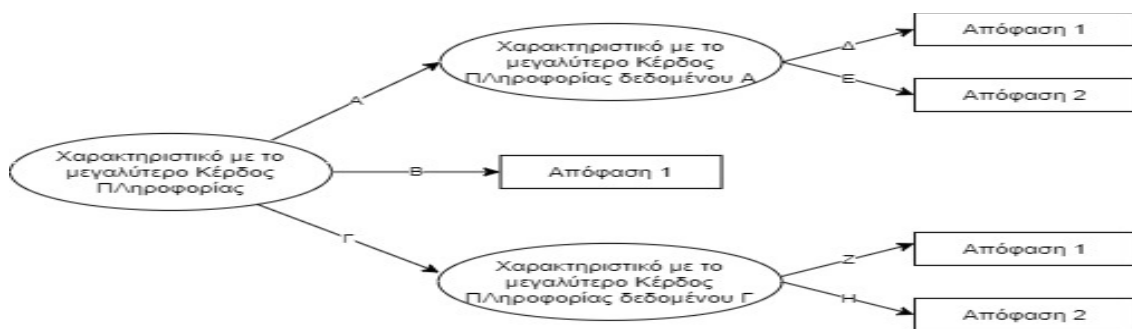
Το κύριο μειονέκτημα αυτών των αλγορίθμων είναι το υπολογιστικό κόστος κατά την κατηγοριοποίηση νέων παραδειγμάτων. Ο λόγος είναι πως λαμβάνουν χώρα σχεδόν όλοι οι υπολογισμοί. Ως εκ τούτου η μάθηση βασισμένη σε παραδείγματα αναφέρεται και ως «οκνηρή» μάθηση (lazy learning). Για τον αλγόριθμο k-NN η κεντρική ιδέα είναι πως η τιμή της συνάρτησης στόχου για ένα νέο παράδειγμα στηρίζεται στις αντίστοιχες τιμές των k πιο κοντινών παραδειγμάτων. Συνεπώς το παράδειγμα ελέγχου παίρνει την ετικέτα που έχει η πλειοψηφία των k πιο κοντινών γειτόνων.



Εικόνα 1 Σχηματική απεικόνιση κ-NN

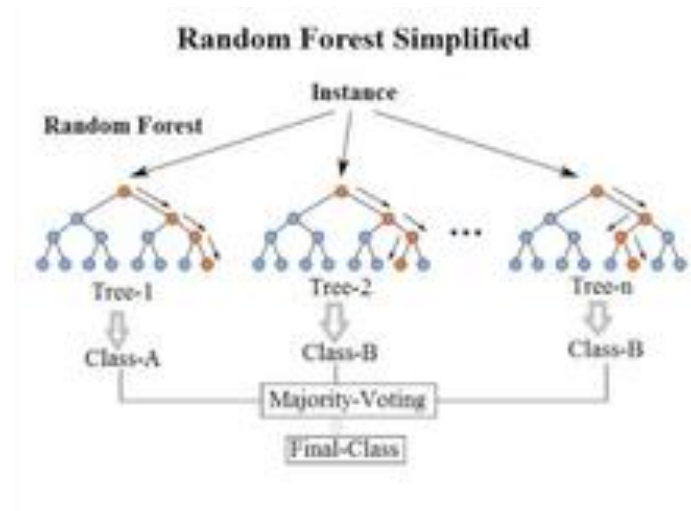
3.4.2 Δέντρο απόφασης- Random Forest

Ένα δέντρο απόφασης κατασκευάζεται συνήθως σε δυο φάσεις. Στην πρώτη φάση της ανάπτυξης κατασκευάζεται ένα μεγάλο δέντρο το οποίο είναι υπερπροσαρμοσμένο στα παραδείγματα εκπαίδευσης και τα παρουσιάζει με μεγάλη ακρίβεια. Στη δεύτερη φάση, η οποία ονομάζεται κλάδεμα (pruning), προσδιορίζεται το τελικό μέγεθος του δέντρου δημιουργώντας κατά αυτόν τον τρόπο πιο γενικευμένους κανόνες. Ένα πλεονέκτημα της χρήσης τέτοιων αλγορίθμων είναι η οπτικοποίηση του αποτελέσματος του δέντρου, διευκολύνοντας τον χρήστη στην εύκολη κατανόηση και ερμηνεία των παραχθέντων κανόνων.



Εικόνα 2 Δέντρο απόφασης

Τα τυχαία δάση (Random Forest) (Wikipedia, 2007) είναι μια μέθοδος μηχανικής μάθησης για ταξινόμηση, παλινδρόμηση και άλλες εργασίες που λειτουργούν κατασκευάζοντας ένα πλήθος δέντρων αποφάσεων κατά το χρόνο εκπαίδευσης και εξάγοντας την τάξη που είναι ο τρόπος των τάξεων (ταξινόμηση) ή μέση / μέση πρόβλεψη (παλινδρόμηση) των μεμονωμένων δέντρων. Τα δάση τυχαίας απόφασης είναι σωστά για τη συνήθεια των δέντρων αποφάσεων να ταιριάζουν με το εκπαιδευτικό τους σετ. Τα τυχαία δάση συνήθως ξεπερνούν τα δέντρα αποφάσεων, αλλά η ακρίβειά τους είναι χαμηλότερη από τα δέντρα που ενισχύονται με κλίση. Ωστόσο, τα χαρακτηριστικά των δεδομένων μπορούν να επηρεάσουν την απόδοσή τους.



Εικόνα 3 Random Forest https://en.wikipedia.org/wiki/Random_forest

3.4.3 Multinomial Naïve Bayes

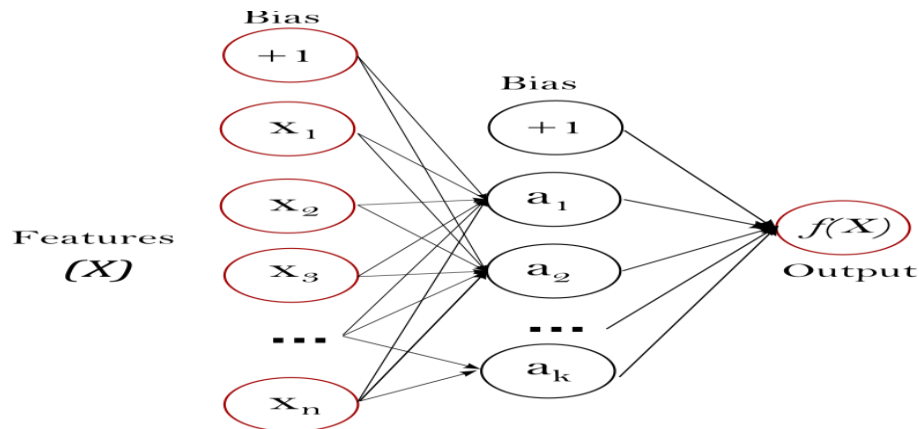
Ο Naïve Bayes ακόμα ένας δημοφιλής αλγόριθμος που ανήκει στην κατηγορία των πιθανολογικών ταξινομητών που βασίζονται στην εφαρμογή του θεωρήματος του Bayes (2), με ισχυρές παραδοχές ανεξαρτησίας μεταξύ των χαρακτηριστικών. Η αφέλεια (Naïve) στηρίζεται στην υπόθεση ότι η τιμή ενός συγκεκριμένου χαρακτηριστικού είναι ανεξάρτητη από την τιμή οποιαδήποτε άλλου δεδομένης της μεταβλητής κλάσης. Η Μπαϊεζιανή συλλογιστική παρέχει μια πιθανοτική προσέγγιση, η οποία στηρίζεται στην υπόθεση πως οι υπό μελέτη ποσότητες ακολουθούν πιθανοτικές κατανομές και πως οι βέλτιστες αποφάσεις μπορούν να παρθούν βάσει αυτών των κατανομών και των παρατηρούμενων δεδομένων.

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \quad (2)$$

Με ένα μοντέλο Multinomial Naïve Bayes, τα δείγματα (διανύσματα χαρακτηριστικών) αντιπροσωπεύουν τις συχνότητες με τις οποίες έχουν δημιουργηθεί συγκεκριμένα συμβάντα από μια κατηγορία ($p_1, \dots, p_n$) όπου το p_i είναι η πιθανότητα να συμβεί το συμβάν i . Ο αλγόριθμος Naïve Bayes προτιμάται να χρησιμοποιείται σε δεδομένα που διανέμονται κατηγορικά. Είναι ένας από τους τυπικούς αλγόριθμους που χρησιμοποιείται στην ταξινόμηση κατηγοριοποίησης κειμένου.

3.4.4 Neural Network- MLP

Το Multi-layer Perceptron (MLP) είναι ένας εποπτευόμενος αλγόριθμος μάθησης που μαθαίνει μια συνάρτηση $f(\cdot): \mathbb{R}^m \rightarrow \mathbb{R}^o$ εκπαιδεύοντας σε ένα σύνολο δεδομένων m όπου είναι ο αριθμός των διαστάσεων για την είσοδο και o είναι ο αριθμός των διαστάσεων για την έξοδο. Με δεδομένο ένα σύνολο χαρακτηριστικών $X = x_1, x_2, \dots, x_m$ και έναν στόχο y , μπορεί να μάθει ένα μη γραμμικό εργαλείο προσέγγισης είτε για ταξινόμηση είτε για παλινδρόμηση. Είναι διαφορετικό από τη λογιστική παλινδρόμηση, καθώς μεταξύ του επιπέδου εισόδου και εξόδου, μπορεί να υπάρχουν ένα ή περισσότερα μη γραμμικά στρώματα, που ονομάζονται κρυφά στρώματα. Το σχήμα 1 δείχνει ένα κρυφό στρώμα MLP με κλιμακωτή έξοδο.



Εικόνα 4 Κρυφό στρώμα MLP https://scikit-learn.org/stable/modules/neural_networks_supervised.html

Το αριστερότερο στρώμα, γνωστό ως στρώμα εισόδου, αποτελείται από ένα σύνολο νευρώνων $\{x_i: x_1, x_2, \dots, x_m\}$ που αντιπροσωπεύουν τα χαρακτηριστικά εισαγωγής. Κάθε νευρώνας στο κρυφό στρώμα μετατρέπει τις τιμές από το προηγούμενο στρώμα με σταθμισμένο γραμμικό άθροισμα $w_1x_1 + w_2x_2 + \dots + w_mx_m$, ακολουθούμενη από μια μη γραμμική συνάρτηση ενεργοποίησης $g(\cdot): \mathbb{R} \rightarrow \mathbb{R}$. Το επίπεδο εξόδου λαμβάνει τις τιμές από το τελευταίο κρυφό επίπεδο και τις μετατρέπει σε τιμές εξόδου.

Η βιβλιοθήκη MLP περιέχει τα δημόσια χαρακτηριστικά `coefs` και `intercepts_`. `coefs`. Είναι μια λίστα με πίνακες βάρους, όπου η μήτρα βάρους στο δείκτη αντιπροσωπεύει τα βάρη μεταξύ στρώματος i και στρώματος $i+1$. Το `intercepts_` είναι μια λίστα διανυσμάτων μεροληψίας, όπου το διάνυσμα στο ευρετήριο αντιπροσωπεύει τις τιμές πόλωσης που προστίθενται στο επίπεδο $i+1$.

Τα **πλεονεκτήματα** του Multi-layer Perceptron είναι:

- Δυνατότητα εκμάθησης μη γραμμικών μοντέλων.
- Δυνατότητα εκμάθησης μοντέλων σε πραγματικό χρόνο (on-line learning) χρησιμοποιώντας `partial_fit`.

Τα **μειονεκτήματα** του Multi-layer Perceptron (MLP) περιλαμβάνουν:

- Το MLP με κρυφά στρώματα έχει μια λειτουργία μη κυρτής απώλειας όπου υπάρχουν περισσότερα από ένα τοπικά ελάχιστα. Επομένως, διαφορετικές τυχαίες αρχικοποιήσεις βάρους μπορούν να οδηγήσουν σε διαφορετική ακρίβεια επικύρωσης.
- Το MLP απαιτεί συντονισμό ενός αριθμού υπερπαραμέτρων όπως ο αριθμός των κρυφών νευρώνων, των στρωμάτων και των επαναλήψεων.
- Το MLP είναι ευαίσθητο στην κλιμάκωση χαρακτηριστικών.

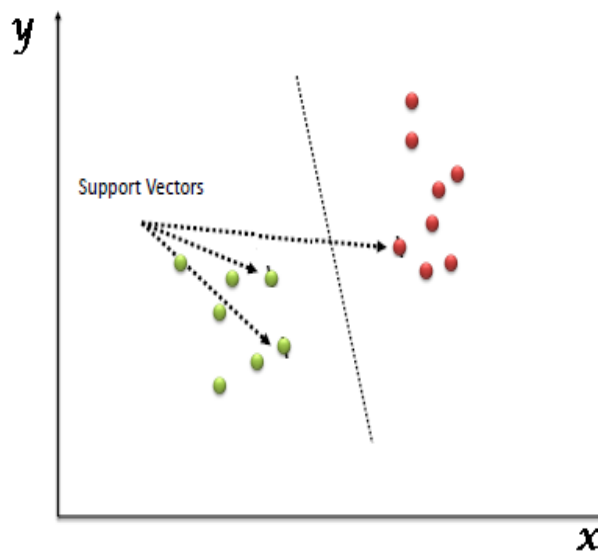
3.4.5 Support Vector Machine

Ο αλγόριθμος Μηχανές Διανυσμάτων Υποστήριξης (Support Vector Machine - SVM) αποτελεί έναν αλγόριθμο εποπτευόμενης μηχανικής μάθησης που μπορεί να χρησιμοποιηθεί τόσο για ταξινόμηση όσο και για προκλήσεις παλινδρόμησης. Δημιουργεί ένα μοντέλο που εκχωρεί τα παραδείγματα σε μια κατηγορία ή την άλλη. Πρόκειται

δηλαδή για μια δυαδική μηχανή μάθησης η οποία κατασκευάζει ένα υπερεπίπεδο ως επιφάνεια απόφασης με τρόπο τέτοιο ώστε το περιθώριο διαχωρισμού μεταξύ θετικών και αρνητικών παραδειγμάτων να μεγιστοποιείται (3). Λαμβάνοντας υπόψη ένα σύνολο παραδειγμάτων εκπαίδευσης, το SVM στην απλοϊκή του μορφή δημιουργεί ένα μοντέλο όπου εκχωρεί νέα παραδείγματα στη μια κατηγορία ή την άλλη εκτελώντας έτσι μια γραμμική ταξινόμηση.

$$\text{Margin} = \frac{2}{\|\vec{w}\|^2} \quad (3)$$

Στον αλγόριθμο SVM, σχεδιάζουμε κάθε στοιχείο δεδομένων ως σημείο στον n-διάστατο χώρο (όπου το n είναι ο αριθμός των χαρακτηριστικών που έχουμε) με την τιμή κάθε χαρακτηριστικού να είναι η τιμή μιας συγκεκριμένης συντεταγμένης. Στη συνέχεια, πραγματοποιούμε ταξινόμηση βρίσκοντας το υπερ-επίπεδο που διαφοροποιεί πολύ καλά τις δύο κατηγορίες όπως απεικονίζεται παρακάτω.



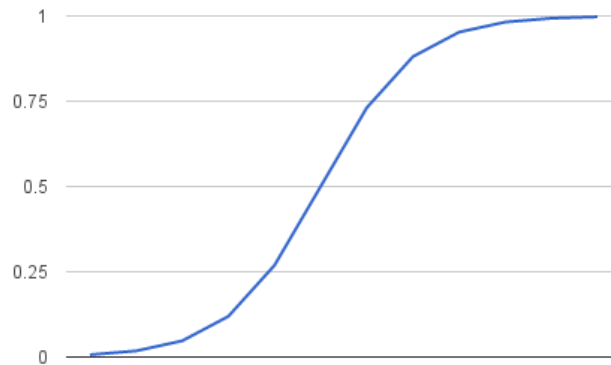
Εικόνα 5 Δημιουργία υπερεπιπέδου <https://www.analyticsvidhya.com/blog/2017/09/understaing-support-vector-machine-example-code/>

3.4.6 Λογιστική Παλινδρόμηση

Η λογιστική συνάρτηση, που ονομάζεται επίσης και σιγμοειδής, αναπτύχθηκε από στατιστικούς για να περιγράψει τις ιδιότητες της αύξησης του πληθυσμού στην οικολογία, αυξάνοντας γρήγορα και μεγιστοποιώντας τη φέρουσα ικανότητα του περιβάλλοντος. Στα στατιστικά στοιχεία, το λογιστικό μοντέλο (ή logit model) χρησιμοποιείται για να μοντελοποιήσει την πιθανότητα μιας συγκεκριμένης τάξης ή γεγονότος που υπάρχει, όπως πάσο / αποτυχία, νίκη / ήττα, ζωντανό / νεκρό ή υγιές / ασθενές. Αυτό μπορεί να επεκταθεί για να μοντελοποιήσει διάφορες κατηγορίες συμβάντων, όπως να προσδιορίσει εάν μια εικόνα περιέχει γάτα, σκύλο, λιοντάρι κ.λπ. Σε κάθε αντικείμενο που ανιχνεύεται στην εικόνα θα εκχωρηθεί πιθανότητα μεταξύ 0 και 1, με ένα άθροισμα ενός. Είναι μια καμπύλη σχήματος S που μπορεί να πάρει οποιονδήποτε πραγματικό αριθμό και να τον αντιστοιχίσει σε τιμή μεταξύ 0 και 1, αλλά ποτέ ακριβώς σε αυτά τα όρια.

$$1 / (1 + e^{-\text{value}}) \quad (4)$$

Όπου το e είναι η βάση των φυσικών λογάριθμων (αριθμός Euler) και η τιμή είναι η πραγματική αριθμητική τιμή που θέλουμε να μεταμορφώσουμε. Ακολουθεί μια γραφική παράσταση των αριθμών μεταξύ -5 και 5 που μετατρέπονται στο εύρος 0 και 1 χρησιμοποιώντας τη λογιστική συνάρτηση.



Εικόνα 6 Σιγμοειδής συνάρτηση <https://machinelearningmastery.com/logistic-regression-for-machine-learning/>

Η λογιστική παλινδρόμηση (Wikipedia, 2003) είναι ένα στατιστικό μοντέλο που στη βασική του μορφή χρησιμοποιεί μια λογιστική συνάρτηση για τη μοντελοποίηση μιας δυαδικής εξαρτώμενης μεταβλητής. Μαθηματικά, ένα δυαδικό λογιστικό μοντέλο έχει μια εξαρτημένη μεταβλητή με δύο πιθανές τιμές, όπως το πέρασμα / αποτυχία που αντιπροσωπεύεται από μια μεταβλητή δείκτη, όπου οι δύο τιμές φέρουν την ένδειξη "0" και "1". Στο λογιστικό μοντέλο, το log-odds (ο λογάριθμος των αποδόσεων) για την τιμή με την ένδειξη "1" είναι ένας γραμμικός συνδυασμός μιας ή περισσότερων ανεξάρτητων μεταβλητών ("προβλέψεις"). Οι ανεξάρτητες μεταβλητές μπορούν καθεμιά να είναι μια δυαδική μεταβλητή (δύο κατηγορίες, κωδικοποιημένες από μια μεταβλητή δείκτη) ή μια συνεχής μεταβλητή (οποιαδήποτε πραγματική τιμή). Η αντίστοιχη πιθανότητα της τιμής με την ένδειξη "1" μπορεί να κυμαίνεται μεταξύ 0 (σίγουρα η τιμή "0") και 1 (σίγουρα η τιμή "1"), εξ ου και η επισήμανση.

Η συνάρτηση που μετατρέπει log-odds σε πιθανότητα είναι η λογιστική συνάρτηση, εξ ου και το όνομα. Η μονάδα μέτρησης για την κλίμακα log-odds ονομάζεται logit, από τη λογιστική μονάδα, εξ ου και τα εναλλακτικά ονόματα. Το καθοριστικό χαρακτηριστικό του λογιστικού μοντέλου είναι ότι η αύξηση μίας από τις ανεξάρτητες μεταβλητές κλιμακώνει πολλαπλασιαστικά τις πιθανότητες του δεδομένου αποτελέσματος με σταθερό ρυθμό, με κάθε ανεξάρτητη μεταβλητή να έχει τη δική της παράμετρο. για μια δυαδική εξαρτώμενη μεταβλητή αυτό γενικεύει την αναλογία αποδόσεων.

3.6 Εργαλείο επεξεργασίας φυσικής γλώσσας (NLTK)

Το περιεχόμενο των tweets αποτελείται από μηνύματα γραμμένα σε φυσική γλώσσα με την προσθήκη ειδικών χαρακτήρων και emoticon. Για την επεξεργασία της φυσικής γλώσσας έγινε η χρησιμοποίηση της βιβλιοθήκης της python NLTK (Natural language Toolkit). Σχεδιάστηκε και αναπτύχθηκε στο Πανεπιστήμιο της Pennsylvania (Loper and Bird, 2002) σε συνδυασμό με ένα μάθημα γλωσσικής υπολογιστικής. Παρέχει μια σειρά από βιβλιοθήκες και προγράμματα για συμβολική και στατιστική επεξεργασία της φυσικής γλώσσας (NLP-

Natural Language Process) για την γλώσσα των αγγλικών. Τα βασικά εργαλεία που χρησιμοποιήθηκαν για τις ανάγκες της εργασίας είναι:

- **Tokenization.** Η διαδικασία κατά την οποία τεμαχίζεται μια συμβολοσειρά χαρακτήρων εισόδου σε ξεχωριστά μέρη με κάποιο κριτήριο όπως τα κενά ανάμεσα από τις λεκτικές μονάδες. Σκοπός της διαδικασίας αυτής είναι η αποδόμηση ενός κειμένου σε λεκτικές μονάδες ώστε να υποστούν περαιτέρω επεξεργασία.
- **Stemming.** Η επιστροφή μιας λέξης στην ρίζα της ή στην βάση προέλευσής της αν είναι παραγόμενη. Μια σειρά αλγορίθμων για να το επιτύχουν αυτό, όπως και ο Porter Stemmer που χρησιμοποιήθηκε στην εργασία, αφαιρούν την κατάληξη της λέξης. Η χρησιμοποίηση του αποτελεί μέρος της κανονικοποίησης των λέξεων του κειμένου.
- **Lemmatize.** Μια λέξη μπορεί να συναντάται σε διαφορετικές μορφές όπως πληθυντικός αριθμός και χρονικούς μετασχηματισμούς. Έτσι κατά την διαδικασία αυτή επιστρέφεται η λέξη στο αρχικό λήμμα. Η ειδοποιός διαφορά με το stemming είναι ότι σε κάποιες μορφές επεξεργασίας όπως στο μοντέλο Word2Vec χρειάζεται τα αποτελέσματα που παίρνουμε να μπορούμε να τα κατανοήσουμε. Η ρίζα μιας λέξης ενδεχομένως να δυσκολεύει αυτό το σκοπό καθώς κάνει ακαθόριστη την προέλευση της, έτσι σε αυτές της περιπτώσεις είναι προτιμότερος τρόπος κανονικοποίησης το Lemmatize.
- **StopWords.** Λέξεις όπως άρθρα, αντωνυμίες και σύνδεσμοι απαλείφονται από το αρχικό κείμενο γιατί δεν προσφέρουν κάποια χρήσιμη πληροφορία. Το StopWords αποτελείται από καθορισμένες λέξεις και όταν καλείται φιλτράρει την είσοδο με βάση το σετ λέξεων που περιέχει.
- **N-gram.** Συνεχόμενη ακολουθία n λεκτικών μονάδων. Πολλές φορές το νόημα μιας λέξης είναι διαφορετικό αναλόγως την λέξη που την συνοδεύει. Για να μπορέσουμε να μην αποκλείσουμε αυτή την δυνατότητα από την επεξεργασία μας χρησιμοποιούμε αυτήν την μέθοδο για να παίρνουμε συνδυασμούς συνήθως 2 ή 3 λέξεων.

3.7 Textblob

Το Textblob (Loria, 2012) αποτελεί μια βιβλιοθήκη της Python, η οποία χρησιμοποιείται για αρκετές μορφές επεξεργασίας της φυσικής γλώσσας (NLP) όπως είναι και η ανάλυση συναισθήματος. Παρέχει μεθόδους με τις οποίες μπορούμε να ταξινομήσουμε κείμενο με βάση την πολικότητα και υποκειμενικότητα του, λαμβάνοντας τιμές αντίστοιχα από $[-1,1]$ και $[0,1]$. Στην περίπτωση της πολικότητας ότι είναι κοντά στο -1 θεωρείται πολύ αρνητικό, ότι ισούται με το 0 θεωρείται ως ουδέτερο, ενώ ότι είναι κοντά στο $+1$ θεωρείται πολύ θετικό. Αντίστοιχα για την υποκειμενικότητα, για τιμές κοντά στο 0 το κείμενο θεωρείται αντικειμενικό ενώ κοντά στο 1 πολύ υποκειμενικό. Το Textblob στηρίζει την λειτουργία του σε λεξικό στο οποίο οι λέξεις που εκφράζουν συναίσθημα είναι βαθμονομημένες και χρησιμοποιεί μαθηματική συνάρτηση για να υπολογίσει το συνολικό αποτέλεσμα. Επιπλέον παρέχει ενσωματωμένη τεχνική για την διαχείριση της άρνησης.

3.8 Δημιουργία Διανυσμάτων

Για να μπορέσει η μηχανική μάθηση να λάβει χώρα, απαιτούνται τα δεδομένα μας να περιγράφονται με χαρακτηριστικά. Επειδή η επεξεργασία μας αφορά κείμενο, κάθε λέξη του κειμένου πρέπει με κάποια μεθοδολογία να απεικονίζεται ως ένα χαρακτηριστικό και το σύνολο αυτών των χαρακτηριστικών να αποδίδει ένα μήνυμα κειμένου. Για τον σκοπό αυτό χρησιμοποιήθηκαν τα μοντέλα που αναλύονται σε αυτή την ενότητα.

3.8.1 Μοντέλο Bag of Words

Το μοντέλο «σάκος από λέξεις» ή Bag-of-Words (BOW) είναι ένας τρόπος εξαγωγής χαρακτηριστικών από κείμενο για χρήση στη μοντελοποίηση, όπως με αλγόριθμους μηχανικής μάθησης. Είναι απλό στην κατανόηση και την εφαρμογή και έχει σημειώσει μεγάλη επιτυχία σε προβλήματα όπως η μοντελοποίηση γλωσσών και η ταξινόμηση εγγράφων. Η προσέγγιση είναι πολύ απλή και ευέλικτη και μπορεί να χρησιμοποιηθεί με πολλούς τρόπους εξαγωγής χαρακτηριστικών από έγγραφα.

Ένας σάκος από λέξεις είναι μια αναπαράσταση κειμένου που περιγράφει την εμφάνιση των λέξεων μέσα σε ένα έγγραφο. Πιο συγκεκριμένα περιλαμβάνει ένα λεξιλόγιο με όλες τις γνωστές λέξεις και ένα μέτρο της παρουσίας των λέξεων αυτών. Η αναπαράσταση αυτή του περιεχομένου ενός κειμένου με το μέτρο των γνωστών λέξεων δίνει τη διαίσθηση ότι τα έγγραφα είναι παρόμοια εάν έχουν παρόμοια αναπαράσταση λέξεων. Η πολυπλοκότητα σε αυτή την τεχνική έγκειται στον τρόπο σχεδίασης του λεξιλογίου των γνωστών λέξεων.

3.8.2 Μοντέλο TF-IDF (Term Frequency–Inverse Document Frequency)

Αυτή είναι μια άλλη μέθοδος που βασίζεται στη μέθοδο της συχνότητας, αλλά είναι διαφορετική από την προηγούμενη προσέγγιση των λέξεων με την έννοια ότι λαμβάνει υπόψη όχι μόνο την εμφάνιση μιας λέξης σε ένα μόνο έγγραφο (ή tweet), αλλά σε ολόκληρο το σώμα κειμένου. Το TF-IDF λειτουργεί «τιμωρώντας» τις κοινές λέξεις, δίνοντάς τους χαμηλότερα βάρη, ενώ παράλληλα δίνει σημασία σε λέξεις που είναι σπάνιες σε ολόκληρο το σώμα αλλά εμφανίζονται σε καλό αριθμό σε λίγα έγγραφα.

Έτσι λοιπόν το TF-IDF είναι ένα στατιστικό μέτρο που αξιολογεί πόσο σχετική είναι μια λέξη με ένα έγγραφο σε μια συλλογή εγγράφων. Αυτό γίνεται πολλαπλασιάζοντας δύο μετρήσεις: πόσες φορές μια λέξη εμφανίζεται σε ένα έγγραφο και τη αντίστροφη συχνότητα του κειμένου της λέξης σε ένα σύνολο εγγράφων. Ας ρίξουμε μια ματιά στους σημαντικούς όρους που σχετίζονται με το TF-IDF:

$TF = (\text{Πόσες φορές εμφανίζεται ο όρος } t \text{ σε ένα έγγραφο}) / (\text{Αριθμός όρων στο έγγραφο})$

$IDF = \log(N / n)$, όπου, N είναι ο συνολικός αριθμός των εγγράφων και n είναι ο αριθμός των εγγράφων στα οποία εμφανίστηκε ένας όρος t .

$TF-IDF = TF * IDF$

3.8.3 Μοντέλο Word2Vec

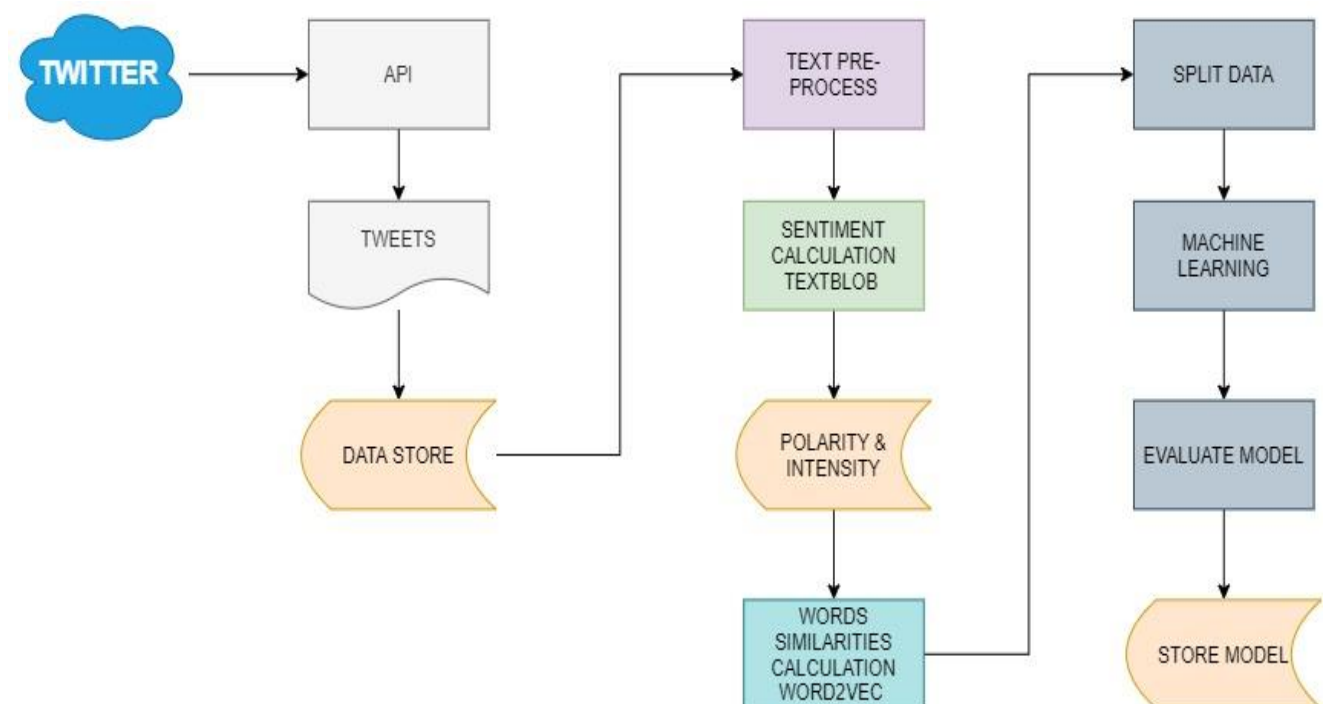
Το Word2Vec (Tomas Mikolov, 2013) αποτελεί ένα μοντέλο διανυσματικής αναπαράστασης λέξεων το οποίο χρησιμοποιείται για επεξεργασία φυσικής γλώσσας (NLP). Ο αλγόριθμος Word2Vec χρησιμοποιεί ένα μοντέλο νευρωνικού δικτύου για να μάθει συσχετισμούς λέξεων από ένα μεγάλο σώμα κειμένου. Μόλις εκπαιδευτεί, ένα τέτοιο μοντέλο μπορεί να ανιχνεύσει συνώνυμες λέξεις ή να προτείνει επιπλέον λέξεις για μια μερική πρόταση, διατηρώντας την σημασιολογική πληροφορία της λέξης. Κατά αυτόν τον τρόπο λέξεις που συνδέονται

σημασιολογικά ή βρίσκονται σε παρεμφερή περιβάλλοντα θα έχουν και παρόμοια διανύσματα π.χ. (σκύλος-κουτάβι).

Από αυτήν την υπόθεση, το Word2Vec μπορεί να χρησιμοποιηθεί για να ανακαλύψει τις σχέσεις μεταξύ λέξεων σε ένα σύνολο δεδομένων, να υπολογίσει την ομοιότητα μεταξύ τους ή να χρησιμοποιήσει τη διανυσματική αναπαράσταση αυτών των λέξεων ως εισαγωγή για άλλες εφαρμογές, όπως ταξινόμηση κειμένου ή ομαδοποίηση.

4 Υλοποίηση

Στο κεφάλαιο αυτό καταγράφονται οι τρόποι με τους οποίους χρησιμοποιήθηκαν τα εργαλεία που αναπτύχθηκαν προηγουμένως για την ανάλυση του συναισθήματος. Η πλατφόρμα που αναπτύχθηκε ο κώδικας στη γλώσσα Python είναι το Jupyter Notebooks, το οποίο παρέχει στον χρήστη ένα διαδραστικό περιβάλλον και είναι ιδανικό για την ανάπτυξη κώδικα σε συνδυασμό με την περιγραφική γλώσσα Markdown⁷ ταυτόχρονα, πράγμα που τον βοηθάει στο να αναπτύξει ένα βιβλίο με κεφάλαια, σημειώσεις και block κώδικα.



Εικόνα 7 Σχηματική απεικόνιση μεθοδολογίας

4.1 Συλλογή και αποθήκευση δεδομένων

Η διαδικασία της εξόρυξης των δεδομένων κρίνεται μείζονος σημασίας στην ανάλυση τους. Ένα από τα προβλήματα των μεγάλων δεδομένων αποτελεί η ποιότητα αυτών και συγκεκριμένα αν προκύπτει κάποια αξία από την ανάλυσή τους. Επι πλέον ο θόρυβος στα δεδομένα είναι κάτι που χρειάζεται ιδιαίτερη μεταχείριση. Όσο

⁷ γλώσσα σήμανσης για τη δημιουργία μορφοποιημένου κειμένου

πιο στοχευμένη είναι η αναζήτηση των δεδομένων τόσο καλύτερα θα είναι και τα αποτελέσματα. Σκοπός είναι από τα 6000⁸ tweets κατά μέσω όρο τα οποία δημοσιεύονται κάθε δευτερόλεπτο, τι ακριβώς χρειάζεται να εξάγουμε για τους σκοπούς της συγκεκριμένης ανάλυσης. Για την εξόρυξη αυτών των δεδομένων χρησιμοποιήθηκε το Tweepy API.

Το Tweepy API διαθέτει διάφορες μεθόδους για την αναζήτηση των tweets, όπως με βάση το όνομα ενός ή περισσότερων χρηστών, με βάση την ημερομηνία και με βάση την τοποθεσία. Ωστόσο για τις ανάγκες της αναζήτησης tweets που συνδέονται με κάποιο από τα εμβόλια που αναπτύχθηκαν για την καταπολέμηση της πανδημίας του κορωνοϊού, επιλέχθηκε η αναζήτηση με βάση λέξη κλειδί (keyword). Κατά αυτόν τον τρόπο το ερώτημα (query), εξασφαλίζει την άμεση συνάφεια του tweet με την λέξη κλειδί, εφόσον αυτή αναφέρεται μέσα σε αυτό.

Η χρησιμοποίηση γεοεντοπισμού δεν χρησιμοποιήθηκε, εφόσον η πανδημία είναι ένα οικουμενικό φαινόμενο που εμπλέκει το σύνολο των χωρών. Ο εκτροχιασμός της καθημερινότητας του κόσμου με τα διάφορα μέτρα που πάρθηκαν όπως η χρησιμοποίηση μάσκας, η αποφυγή συγχρωτισμού και τα κατά τόπους lockdown, είχε ως αποτέλεσμα αυτό να αντανakλάται στην ποσότητα και στην συχνότητα των tweets σχετικά με την πανδημία.

Ιδιαίτερος λόγος γίνεται μέσα σε αυτό το πλαίσιο της πανδημίας, σχετικά με τα εμβόλια και ιδιαίτερα την νέα τεχνολογία που χρησιμοποιείται για πρώτη φορά, αυτή του mRNA. Σε αυτό το πλαίσιο συναντώνται μηνύματα των χρηστών που εκφράζουν αντίστοιχα συναισθήματα ως προς την ασφάλεια αυτών των εμβολίων, την καθολικότητα της εφαρμογής τους, αν τηρήθηκαν τα προβλεπόμενα στάδια στην ανάπτυξη των εμβολίων, η πρόθεση τους στο να εμβολιαστούν και το κίνημα των αντιεμβολιαστών - αρνητών της ύπαρξης της πανδημίας αποτελούν κάποιες από τις κατηγορίες του περιεχομένου που διαπραγματεύονται τα tweets των χρηστών.

Η αναζήτηση των tweets έγινε με βάση τα πιο πρόσφατα στην ημερομηνία που εκτελέστηκε η αναζήτηση (4/12/20). Για την μείωση του θορύβου επιλέχθηκε φίλτρο ώστε να μην συλλέγονται οι αναδημοσιεύσεις. Επί πλέον καθώς τα εργαλεία τα οποία χρησιμοποιούνται για την ανάλυση φυσικής γλώσσας, στηρίζονται ως επί το πλείστο στην αγγλική γλώσσα, αναζητήθηκαν tweets με φίλτρο να είναι γραμμένα στα αγγλικά.

Οι φράσεις κλειδιά που χρησιμοποιήθηκαν για την αναζήτηση είναι:

Φράση κλειδί	Αριθμός tweets
covid vaccine	215.081
Pfizer vaccine	114.347
moderna vaccine	32.387
mRna vaccine	30.913
oxford vaccine	12.265
Σύνολο	404.993

Πίνακας 4 Φράσεις κλειδιά

Επειδή η άδεια χρήσης του Twitter API, που πάρθηκε για την συγκεκριμένη εργασία έχει κάποιους περιορισμούς τόσο χρονικούς όσο και ποσοτικούς χρησιμοποιήθηκε επί προσθέτως μια δομή ελέγχου στον κώδικα η οποία όταν υπερβαίνει κάποιον περιορισμό να θέτει το πρόγραμμα σε αναμονή για 15 λεπτά και κατόπιν συνεχίζει η αναζήτηση. Προφανώς κάθε φορά που βρίσκεται ένα Tweet ελέγχεται αν ήδη υπάρχει με βάση το id του.

Η εξαγωγή ενός tweet είναι ένα ανοικτό μορφότυπο το οποίο χρησιμοποιεί κείμενο, που μπορεί να διαβαστεί από τον άνθρωπο, και αποτελείται από ζεύγη χαρακτηριστικών-τιμών. Περιλαμβάνει στοιχεία όπως το όνομα χρήστη, το id του, την τοποθεσία, το κείμενο, μεταδομένα κτλπ. Για τις ανάγκες της εργασίας και για λόγους

⁸ Statista 2019

προστασίας των προσωπικών δεδομένων, συλλέχθηκαν tweets ανώνυμα και κρατήθηκε μόνο το κείμενο τους και κανένα άλλο στοιχείο.

Η αποθήκευση των δεδομένων γίνεται σε αρχείο τύπου text (csv), ώστε να διευκολύνεται η μετέπειτα επεξεργασία τους. Για την ανάκτησή τους χρησιμοποιείται το Panda Dataframe με το οποίο γίνεται έλεγχος για ελλιπείς τιμές και διπλότυπα.

Πίνακας 5 Διαδικασία εξαγωγής Tweet

```
searchQuery = 'mrna'
retweet_filter='-filter:retweets'

q=searchQuery+retweet_filter
tweetsPerQry = 200
fName = 'mrna'
sinceId = None

max_id = -1
maxTweets = 400000

tweetCount = 0
print("Downloading max {0} tweets".format(maxTweets))
with open(fName, 'w') as f:
    time_limit= 10
    start = dt.datetime.now()
    end = start + dt.timedelta(hours=time_limit)
    count, exitcount = 0, 0
    while dt.datetime.now() < end:
        print('remaining time:',str(end-dt.datetime.now()))
        while tweetCount < maxTweets:
            tweets = []
            try:
                if (max_id <= 0):
                    if (not sinceId):
                        new_tweets = api.search(q=q, lang ="en", count=tweetsPerQry, tweet_mode='extended')

                    else:
                        new_tweets = api.search(q=q, lang ="en", count=tweetsPerQry,
                                                since_id=sinceId, tweet_mode='extended')

                else:
                    if (not sinceId):
                        new_tweets = api.search(q=q, lang ="en", count=tweetsPerQry,
                                                max_id=str(max_id - 1), tweet_mode='extended')

                    else:
                        new_tweets = api.search(q=q, lang ="en", count=tweetsPerQry,
                                                max_id=str(max_id - 1),
                                                since_id=sinceId, tweet_mode='extended')

            if not new_tweets:
                print("No more tweets found")
                break
            for tweet in new_tweets:
                f.write(str(tweet.full_text.replace("\n","").encode("utf-8"))+"\n")

            tweetCount += len(new_tweets)
            print("Downloaded {0} tweets".format(tweetCount))
```

```

max_id = new_tweets[-1].id

except tweepy.TweepError as e:
    # Just exit if any error
    print("some error : " + str(e))
    print("zzzZZZ for 15 minutes...")
    time.sleep(15*60)
    break

print ("Downloaded {0} tweets, Saved to {1}".format(tweetCount, fName))

```

4.2 Προεπεξεργασία δεδομένων

Ο πυρήνας των δεδομένων αυτής της εργασίας είναι το tweet, το οποίο αποτελεί ένα τρόπο δημοσίευσης και μεταφοράς πληροφορίας ανάμεσα στους χρήστες. Η δομή του αποτελείται από **140 χαρακτήρες** και ως εκ τούτου εκτός από κείμενο το οποίο χρειαζόμαστε για τις ανάγκες της επεξεργασίας, συνηθίζεται να περιέχει σχεδόν πάντα τα σύμβολα:

- “#” ονομάζεται hashtag και χρησιμοποιείται για να γίνει αναφορά σε κάποιο θέμα. Κατά αυτόν τον τρόπο παρέχεται η δυνατότητα στους χρήστες να ομαδοποιούνται οι δημοσιεύσεις τους γύρω από ένα συγκεκριμένο θέμα διευκολύνοντας έτσι την αναζήτηση πληροφοριών.
- “@” το σύμβολο at χρησιμοποιείται για να γίνει αναφορά σε κάποιον χρήστη.
- “RT” ή αλλιώς συντομογραφία του retweet, χρησιμοποιείται για την αναμετάδοση ενός tweet κάποιου χρήστη από κάποιον άλλον.
- Emotcon. Το Twitter υποστηρίζει πάνω από 1100 ιδεογράμματα τα οποία είναι συνδυασμός 2 ή περισσότερων ειδικών χαρακτήρων π.χ. 😊 το οποίο αποτελείται από : και). Χρησιμοποιούνται σε μεγάλο βαθμό καθώς οι χρήστες εξοικονομούν λέξεις και αποτυπώνουν σε μεγάλο βαθμό συναισθήματα.
- Ειδικοί χαρακτήρες όπως είναι (*,^,/,\,{,|,-)
- Πολυμεσικό περιεχόμενο. Σε ένα tweet υπάρχει η δυνατότητα να συνοδεύεται από κάποια εικόνα ή βίντεο και χρήση υπερσυνδέσμων.
- Πολυγλωσσικό περιεχόμενο. Ύπαρξη λέξεων που ανήκουν σε περισσότερες από μια γλώσσες.
- Ποικιλία ορθογραφίας. Η πλειοψηφία των χρηστών γράφει σε ανεπίσημο λόγο, χρησιμοποιώντας συντομογραφίες, νεολογισμούς και εκφράσεις της καθημερινότητας.
- Σημεία στίξης και πολλαπλά κενά.

Για την διαχείριση των παραπάνω, έγινε χρήση της βιβλιοθήκης RE της Python, η οποία προσφέρει την δυνατότητα μέσω κανονικών εκφράσεων (regular expression) να γίνει στοχευμένη απαλοιφή στα κομμάτια του μηνύματος που προκαλούν θόρυβο. Για τον λόγο αυτό δημιουργήθηκε μια μέθοδος η οποία εφαρμόζεται σε όλα τα tweets και στην συνέχεια αποθηκεύονται σε ένα νέο dataframe.

Πίνακας 6 Διαδικασία προ-επεξεργασίας

```

def cleanTxt(text):

    text = re.sub(r"b'",",text) # Removing prefix b'

    text = re.sub(r"\n'",",text) # Removing \n

```

```
text = re.sub(r'#', '', text) # Removing hastags

text = re.sub(r'@[A-Za-z0-9]+', '', text) #Removing @mentions

text = re.sub(r'RT[\s]+', '', text) # Removing RT

text = re.sub(r'https?:\V\S+', '', text) # Removing hyperlink

text = re.sub(r'[\^\w\s]', '', text) # Removing punctuation

text = text.lower() # Convert cases to lower

text = re.sub(r'\b\d+\b', '', text) #Removing numbers

text = re.sub(r'[^a-zA-Z0-9]', '', text) #removing special characters

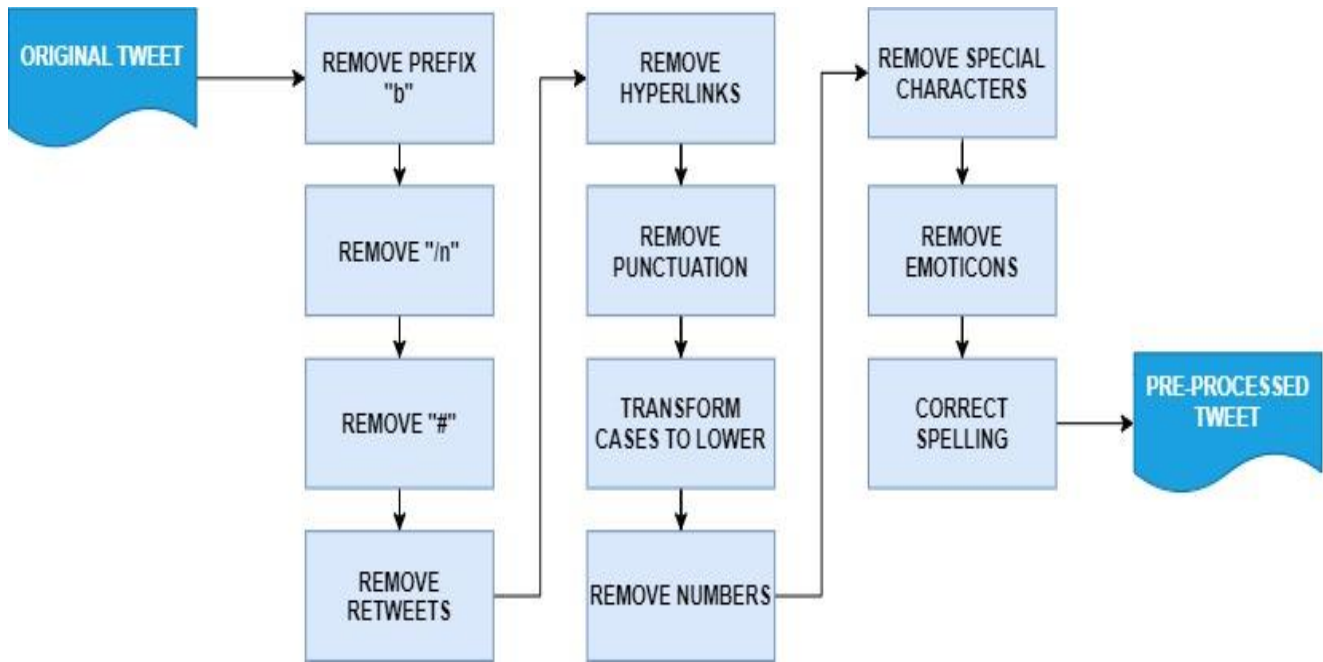
text = re.sub(r"([xX:;][-'\?\.])", '', text) # Removing emoji

text = TextBlob(text).correct() #Correct spelling

return text
```

Κατά την διαδικασία της επιθεώρησης των δεδομένων παρατηρείται ότι χρειάζεται επιπλέον προ-επεξεργασία. Η πρώτη περίπτωση αποτελεί το πρόθεμα “b” μπροστά από κάθε σώμα κειμένου. Αυτό δημιουργείται κατά την αποθήκευση των δεδομένων σε csv, λόγω της κωδικοποίησης σε Utf-8 τα string αποθηκεύονται με το πρόθεμα “b”, όπου αποτελεί την κλάση bytes της python3. Στην συνέχεια αφαιρούνται τα κενά γραμμής προκειμένου να πυκνώσουν οι λέξεις και να αποκτηθεί μια ενιαία διαμόρφωση του κειμένου του tweet.

Τέλος αφαιρούνται όλοι οι αριθμοί, καθώς δεν προσφέρουν κάποιο όφελος στην μέθοδο που έχει επιλεγεί για την ανάλυση του συναισθήματος. Η βιβλιοθήκη TextBlob μας δίνει την δυνατότητα να διορθώσουμε την ορθογραφία του κειμένου. Αυτό το βήμα κρίνεται σημαντικό καθώς όπως θα δούμε στην συνέχεια μια από τις τεχνικές της ταξινόμησης του συναισθήματος που εφαρμόζονται είναι βασισμένη σε λεξικό.



Εικόνα 8 Προεπεξεργασία Tweet

4.3 Ανάλυση συναισθήματος με την χρήση του TextBlob

Για την χρησιμοποίηση του Textblob στην ανάλυση συναισθήματος δημιουργήθηκαν συναρτήσεις ώστε κάθε tweet να αποκτήσει πολικότητα και υποκειμενικότητα. Για τις ανάγκες της μηχανικής μάθησης απαιτείται εκτός από το περιεχόμενο των tweets και μια ετικέτα ή αλλιώς εξαρτημένη μεταβλητή, έτσι δημιουργήθηκε συνάρτηση η οποία δίνει κατηγορική έξοδο με βάση το συναίσθημα (Neutral, Negative, Positive). Τέλος για την καταγραφή των συναισθημάτων και οπτικοποίηση τους με διάγραμμα δημιουργήθηκε συνάρτηση που μετράει το σύνολο των tweets ανά κατηγορία με βάση την πολικότητα.

Πίνακας 7 Διαδικασία TextBlob

```

# Create a function to get the subjectivity
def getSubjectivity(text):
    return TextBlob(text).sentiment.subjectivity

# Create a function to get the polarity
def getPolarity(text):
    return TextBlob(text).sentiment.polarity

# Create two new columns 'Subjectivity' & 'Polarity'
df['Subjectivity'] = df['tweet'].apply(getSubjectivity)
df['Polarity'] = df['tweet'].apply(getPolarity)

# Create a function to get the sentiment
def getSentiment(polarity):
    if polarity < 0 :
        sentiment = 'negative'
    elif polarity == 0:
        sentiment='neutral'

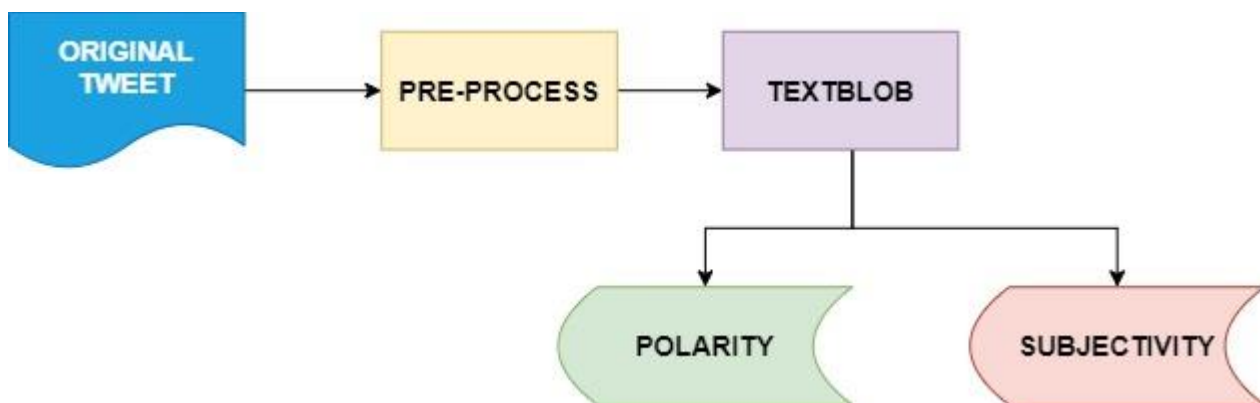
```

```

else:
    sentiment= 'positive'
    return sentiment

def count_sentiment(sentiment):
    count_pos=0
    count_neg=0
    count_neu=0
    for senti in sentiment:
        if senti == 'negative':
            count_neg+=1
        elif senti == 'positive':
            count_pos+=1
        else:
            count_neu+=1
    return print('positive sentiments: {} \nnegative sentiments: {} \nneutral sentiments:
    {}'.format(count_pos,count_neg,count_neu))

```



Εικόνα 9 TEXTBLOB

4.4 Εύρεση ομοιοτήτων με την χρήση του Gensim

Όπως αναλύθηκε το μοντέλο Word2Vec στο κεφάλαιο 3, για να μπορέσουμε να το χρησιμοποιήσουμε θα πρέπει πρώτα να διαιρεθεί το tweet στα επιμέρους συστατικά του δηλαδή σε λέξεις ή αλλιώς tokens. Επιπλέον επαναλαμβανόμενες λέξεις όπως άρθρα και μέρη του λόγου δεν προσφέρουν κάποια πληροφορία, για αυτό και απαλοίζονται με την χρήση έτοιμου λεξικού. Για την βελτίωση των αποτελεσμάτων κρίθηκε σκόπιμη η εφαρμογή ενός μηχανισμού να επιστρέφει η λέξη στο λήμμα της. Για αυτόν το σκοπό επιλέχθηκε το WordnetLemmatizer.

Πίνακας 8 Συνάρτηση δημιουργίας token

```

def get_token(text, stem=False):
    """takes an array of tweets and returns a list of cleaned tokens """

    english_stops = stopwords.words('english') # instantiate
    WL = WordNetLemmatizer() # instantiate
    PS = PorterStemmer()

```

```

sent= []
clean_sent= []

for raw in text:
    tokens = raw.split()
    sent.append(tokens)

for this_sent in sent:
    clear_tokens = [token for token in this_sent if token.isalpha()]
    clear_tokens = [token for token in this_sent if token not in english_stops]
    clear_tokens = [WL.lemmatize(token) for token in this_sent]

    if stem == True:
        clear_tokens = [PS.stem(token) for token in this_sent]
    clean_sent.append(clear_tokens)
    clear_tokens = [token for token in this_sent if len(token)>2]
    #print(clear_tokens)
    return clean_sent

```

Στην συνέχεια δημιουργήθηκαν διαγράμματα ώστε να μην αποκλείσουμε φράσεις από το μοντέλο και αρχικοποιήθηκε το μοντέλο.

Πίνακας 9 Δημιουργία διγραμμάτων

```

word_freq = collections.defaultdict(int)
for sent in sentences:
    for i in sent:
        word_freq[i] += 1
len(word_freq)

w2v_model = Word2Vec(min_count=20,
                      window=2,
                      size=300,
                      sample=6e-5,
                      alpha=0.03,
                      min_alpha=0.0007,
                      negative=20,
                      workers=cores-1)

w2v_model.build_vocab(sentences, progress_per=10000)
w2v_model.train(sentences, total_examples=w2v_model.corpus_count, epochs=30, report_delay=1)

```

4.5 Ταξινόμηση

Για την μηχανική μάθηση είναι απαραίτητο να υπάρχουν κάποια δεδομένα για εκπαίδευση και κάποια για έλεγχο. Τα δεδομένα χωρίστηκαν με την χρήση της μεθόδου split σε δεδομένα εκπαίδευσης 70% και σε δεδομένα ελέγχου 30%. Σε κάθε ένα από τα 4 μοντέλα χαρακτηριστικών Bag of words, TF-IDF, Doc2Vec και Word2Vec εξετάζονται κοινοί αλγόριθμοι μηχανικής μάθησης προκειμένου να μπορέσουν να συγκριθούν τα αποτελέσματα μεταξύ τους. Ως εξαρτημένη μεταβλητή χρησιμοποιήθηκε η έξοδος του συναισθήματος όπως πρόκυπτε με την χρησιμοποίηση του Textblob για κάθε tweet σε τρεις κλάσεις (αρνητικό, ουδέτερο, θετικό).

Χρησιμοποιήθηκε επίσης η μέθοδος για τον διαχωρισμό του κάθε tweet σε tokens, με την διαφοροποίηση αυτή τη φορά ότι επιλέχθηκε η ρίζα της κάθε λέξης (Stemming). Αυτό συνηθίζεται στην ταξινόμηση καθώς βελτιώνει τα αποτελέσματα της. Εφόσον επεξεργάστηκαν τα tokens, αντιστρέφεται η διαδικασία και δημιουργείται το τροποποιημένο tweet προκειμένου να ξεκινήσει η διαδικασία της ταξινόμησης.

Πίνακας 10 Συνάρτηση αντιστροφής token

```
text , sentiment = get_token(dataset['tweet'],stem= True) , list(dataset['sentiment'].values)

def untokenize(text):
    """ This function takes a list of tokenized tweets and returns the untokenized tweet """
    untokenize = []
    for i in range(len(text)):
        this_untoken = ' '.join(text[i])
        untokenize.append(this_untoken)
    return untokenize

#split data
X_train, X_test, y_train, y_test = train_test_split(text, sentiment,
                                                    test_size = 0.3, random_state = 212, shuffle = True)
```

Για την οπτικοποίηση των αποτελεσμάτων των πινάκων σύγχυσης, δημιουργήθηκε συνάρτηση η οποία καλείται σε κάθε αλγόριθμο ξεχωριστά ώστε να γίνει συγκριτική αξιολόγηση στο τέλος.

Πίνακας 11 Συνάρτηση παραγωγής πίνακα σύγχυσης

```
def confusion_matrix(classifier,X_test,y_test):
    """Takes and return confusion matrix plot"""
    plt.style.use('default')
    class_names = ['neg','neu','pos']
    disp = plot_confusion_matrix(classifier,X_test,y_test,
                                display_labels= class_names,
                                cmap=plt.cm.Blues)
    disp.ax_.set_title('Confusion matrix plot')
    plt.show()
```

5 Αποτελέσματα

5.1 Μετρικές αξιολόγησης αλγορίθμων

Για την αξιολόγηση των αλγορίθμων μηχανικής μάθησης χρησιμοποιούνται οι παρακάτω μετρικές:

Πίνακας Σύγχυσης:

Πίνακας 12 Πίνακας Σύγχυσης

	Positive Class	Negative Class
Clasified as Positive	TP	FP
Clasified as Negative	FN	TN

Στον παραπάνω πίνακα εμφανίζεται ο πίνακας σύγχυσης ενός προβλήματος ταξινόμησης δυο κλάσεων που αναπαριστά τα αποτελέσματα της κατηγοριοποίησης ενός αλγορίθμου. Στην κύρια διαγώνιο αντιστοιχούν οι σωστές περιπτώσεις κατηγοριοποίησης TP (True Positive) και TN (True Negative), ενώ οι άλλες δύο περιπτώσεις FN (False Negative) και FP (False Positive) αναφέρονται σε εσφαλμένες ταξινομήσεις.

Ορθότητα:

Ως ορθότητα (Accuracy) ορίζεται η αναλογία όλων των προβλέψεων που ήταν σωστές.

$$Accuracy = \frac{tp + tn}{tp + tn + fp + fn} \quad (5)$$

Ανάκληση:

Ως ανάκληση ή ευαισθησία ορίζεται ο λόγος των σωστών/λανθασμένων προβλέψεων της κλάσης προς τα δεδομένα της κλάσης.

$$Recall_p = \frac{tp}{tp + fn}, Recall_n = \frac{tn}{tn + fp} \quad (6)$$

Ακρίβεια:

Ως ακρίβεια ορίζεται ο λόγος των σωστών/λανθασμένων προβλέψεων της κλάσης προς το σύνολο των προβλέψεων της κλάσης.

$$Precision_p = \frac{tp}{tp + fp}, Precision_n = \frac{tn}{tn + fn} \quad (6)$$

F1-Score:

Ως F1-score ορίζεται ως ο αρμονικός μέσος της ορθότητας ως προς την ανάκληση.

$$F1 - score = 2 \cdot \frac{precision \cdot recall}{precision + recall} = \frac{tp}{tp + \frac{1}{2} \cdot (fp + fn)} \quad (7)$$

5.2 Textblob

Για την εξαγωγή αποτελεσμάτων με το πακέτο του Textblob αρχικά ελέγχθηκε ένα υποσύνολο των συλλεχθέντων δεδομένων και στην συνέχεια ολόκληρο το dataset. Το πρώτο δείγμα επιλέχθηκε τυχαία ώστε να διατηρηθεί η εντροπία του τελικού dataset. Σκοπός ήταν όχι μόνο η εξοικονόμηση υπολογιστικών πόρων, αλλά και σύγκριση των δυο δειγμάτων για αποκλίσεις.

Αυτή η μέθοδος μας δίνει την δυνατότητα αφού προ-επεξεργαστούμε τα δεδομένα μας να τα κατατάξουμε με βάση την πολικότητα και την αντικειμενικότητα τους. Για την παρουσίαση των αποτελεσμάτων έχουν χρησιμοποιηθεί διαγράμματα.

A/A Φάσης	Tweets
1	10.000
2	404.993

Πίνακας 13 Αριθμός tweet κατά φάση

Φάση 1':

Polarity	tweets
Positive	4462
Negative	1863
Neutral	3675
total	10.000

Πίνακας 14 Πολικότητα φάση 1

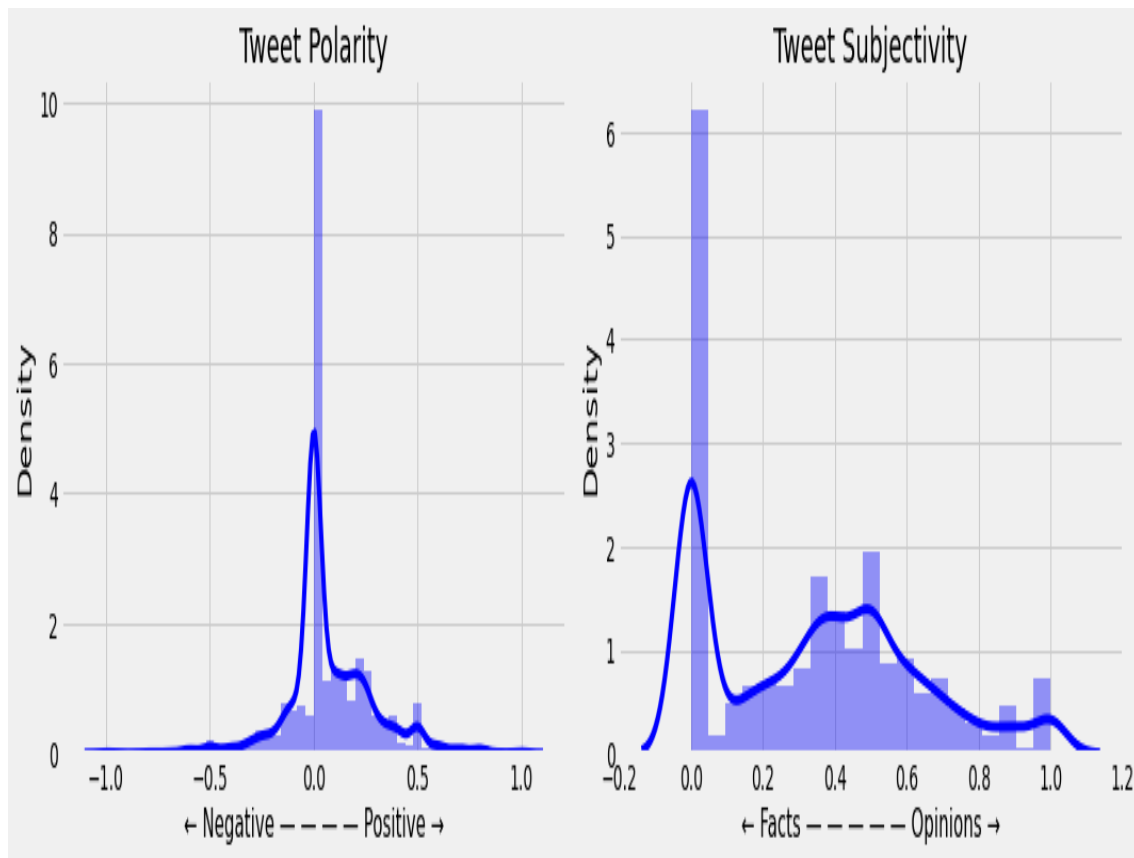


Figure 1 Διάγραμμα πολικότητας και αντικειμενικότητας

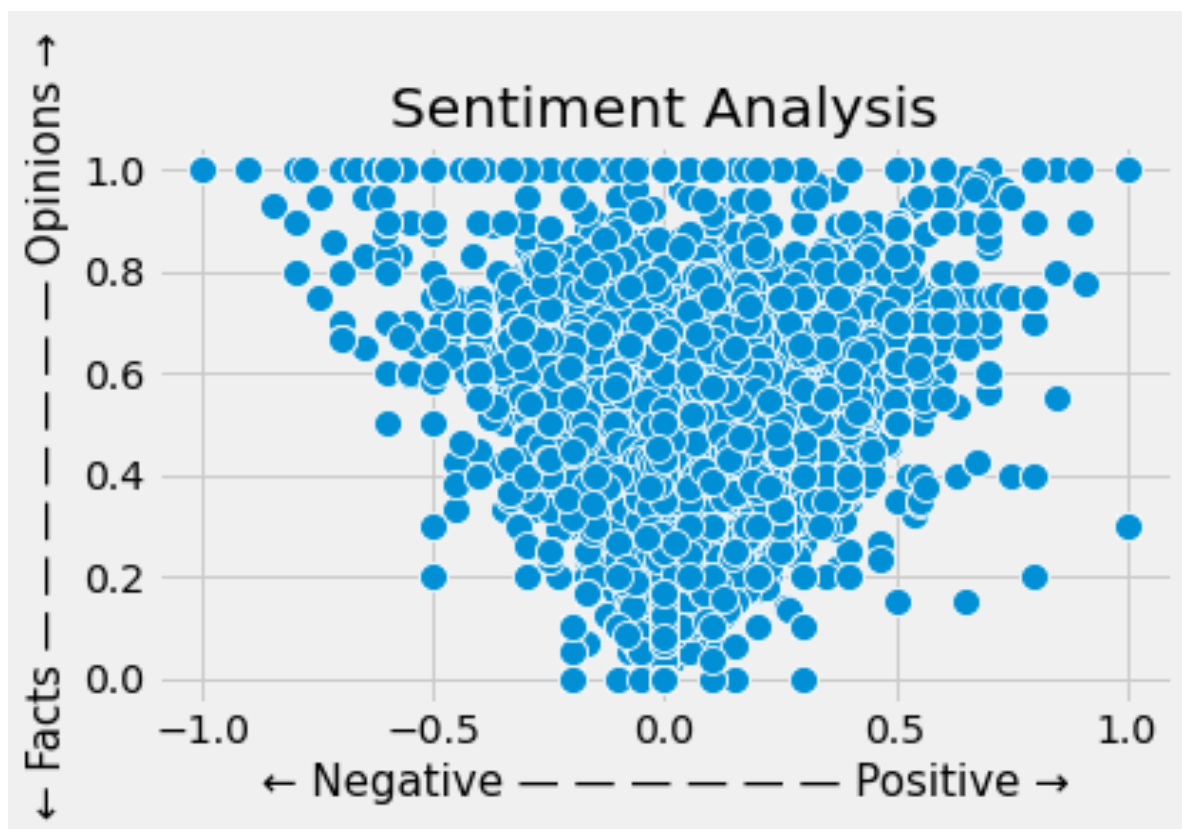


Figure 2 Συνδυαστικό διάγραμμα πολικότητας και αντικειμενικότητας

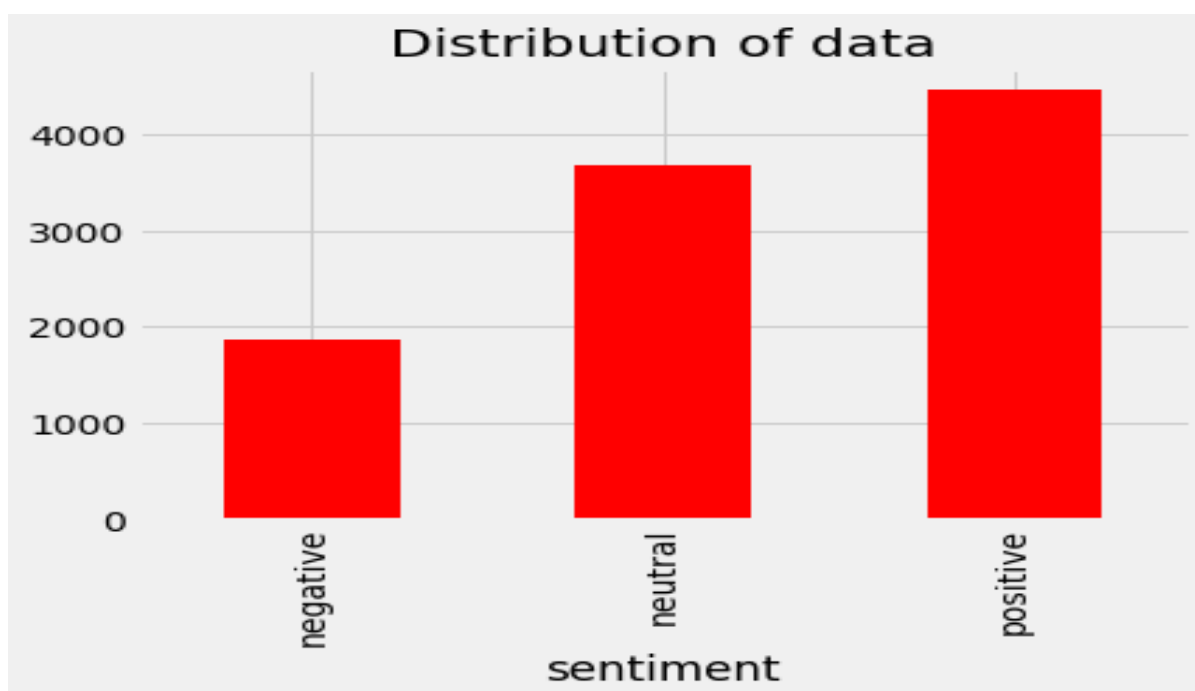
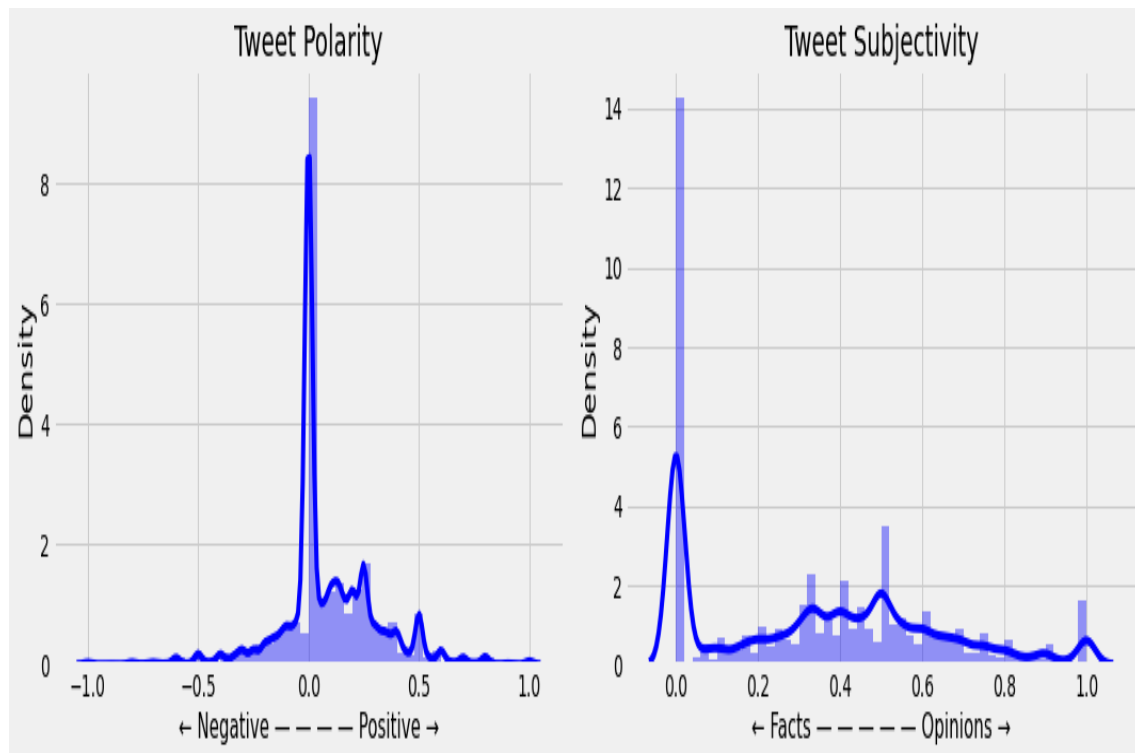


Figure 3 κατανομή συναισθήματος σε 3 κατηγορίες

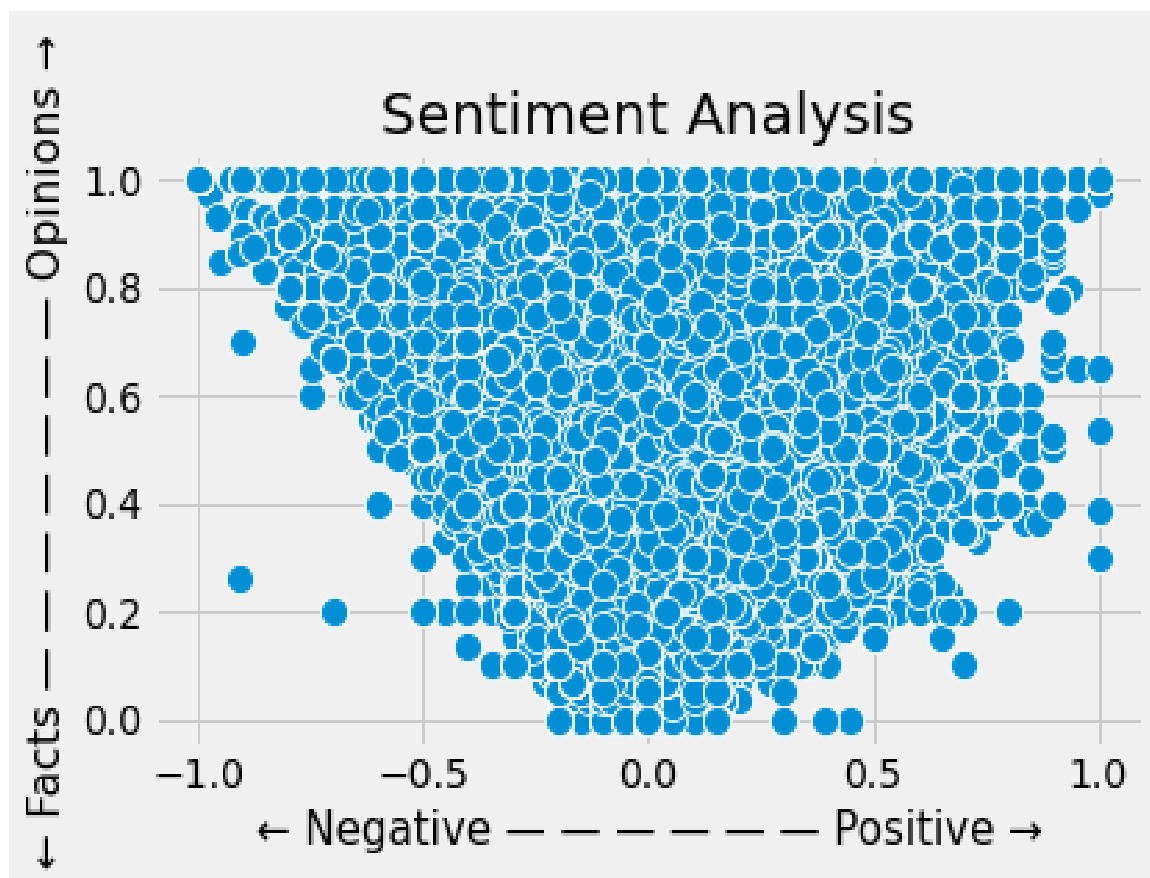
Φάση 2^η:

Polarity	tweets
Positive	192.849
Negative	69.668
Neutral	142.476
total	404.993

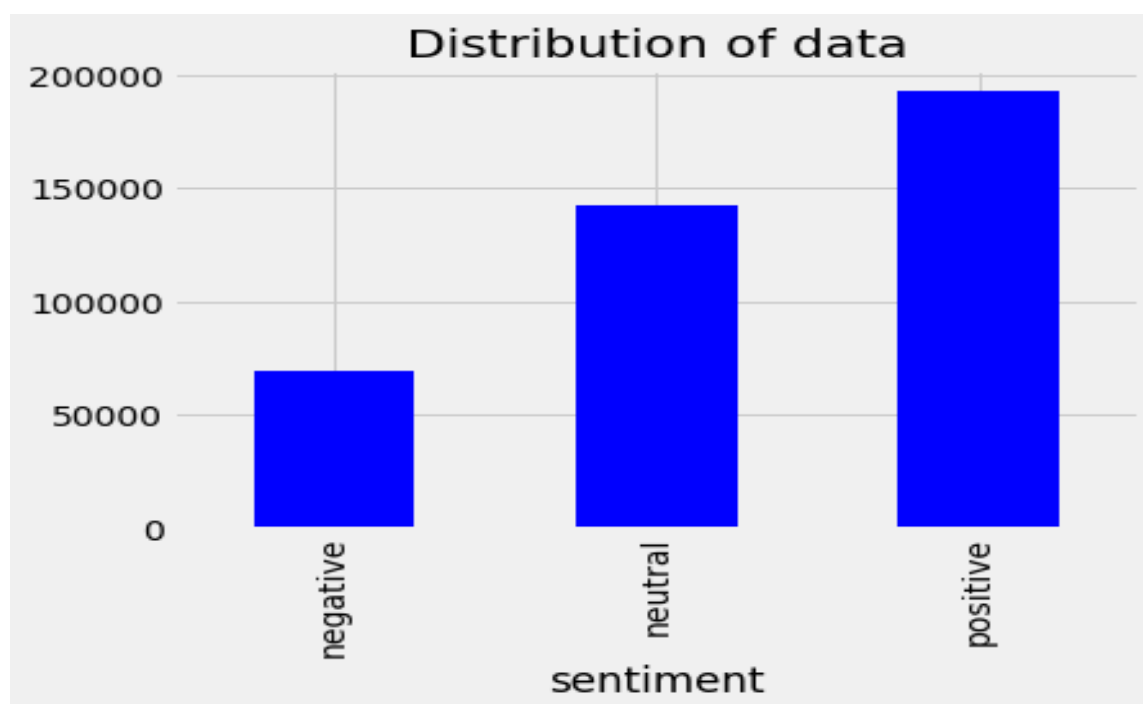
Πίνακας 15 Πολικότητα φάση 2



Εικόνα 10 Διάγραμμα πολικότητας και αντικειμενικότητας



Εικόνα 11 Συνδυαστικό διάγραμμα πολικότητας και αντικειμενικότητας



Εικόνα 12 κατανομή συναισθήματος σε 3 κατηγορίες

Και στις δύο φάσεις παρουσιάζονται κατά αναλογία σχεδόν παρόμοια αποτελέσματα με ανεπαίσθητες διαφοροποιήσεις, πράγμα το οποίο επιβεβαιώνει ότι η κοινή γνώμη, ως προς τα εμβόλια που έχουν αναπτυχθεί για την καταπολέμηση της πανδημίας, κατατάσσεται σε 3 γενικές κατηγορίες:

Πολικότητα	Ποσοστό	Στάση κοινής γνώμης
negative	18-28%	Κατά του εμβολίου
neutral	35-36%	Ουδέτερη
positive	44-47%	Υπέρ του εμβολίου

Πίνακας 16 Στάση κοινής γνώμης

Το πλείστο των tweets ως προς τη πολικότητα είναι θετικά πράγμα το οποίο επαληθεύεται σύμφωνα και με τη δημοσκόπηση του Reuters (11/2020) που αφορά την πρόθεση του κόσμου να εμβολιαστεί. Ενδεικτικά στη Βρετανία, το 73% των ανθρώπων δήλωσαν ότι θα εμβολιαστούν, ενώ στη Δανία ο αριθμός ήταν 70%. Στις Ηνωμένες Πολιτείες, ωστόσο, λιγότεροι από τους μισούς από τους ερωτηθέντες δήλωσαν ότι θα ήταν πρόθυμοι να κάνουν εμβόλιο COVID-19, ένα ποσοστό που παρέμεινε σε γενικές γραμμές σταθερό από τον Ιούλιο. Περισσότερο από το ένα τρίτο των ατόμων που ερωτήθηκαν στην Πολωνία και σχεδόν τα μισά στη Γαλλία - 37% και 48% αντίστοιχα - δήλωσαν ότι θα έλεγαν όχι.

$$density = \frac{\text{number of cases on the bin}}{\text{total cases}} \quad (8)$$

Όσο αφορά την αντικειμενικότητα των παρατηρήσεων υψηλότερη πυκνότητα (8) παρουσιάζουν τα γεγονότα σε σχέση με τις υποκειμενικές γνώμες. Εξερευνώντας τα δεδομένα του δείγματος που παρουσιάζουν την μεγαλύτερη αντικειμενικότητα, διαπιστώνεται ότι το γεγονός αυτό οφείλετε σε συνεχείς δημοσιεύσεις ενημερωτικού κυρίως περιεχομένου από πιστοποιημένες πηγές όπως για παράδειγμα την επιστημονική κοινότητα και διεθνή ειδησεογραφικά πρακτορεία με όλες τις τρέχουσες εξελίξεις που αφορούν τα εμβόλια.

5.3 Word2Vec

Εκμεταλλευόμενοι την δυνατότητα που μας παρέχει η βιβλιοθήκη του Gensim μέσω του Word2Vec, αναζητήσαμε συσχετισμούς λέξεων που σχετίζονται με την πανδημία. Η υπόθεση στηρίχτηκε στην εξής θεώρηση, μια λέξη όπως για παράδειγμα η λέξη “covid” θα πρέπει να έχει κάποια συγγένεια με την λέξη “Pfizer”. Αυτό διαισθητικά προκύπτει από τις πολυπληθείς δημοσιεύσεις των χρηστών του twitter, οι οποίες αναλώνονται στο εμβόλιο κατά της πανδημίας και ειδικότερα σε αυτό που χρησιμοποιεί την «τεχνολογία» mRNA, κάτι το οποίο πιστώνεται από τα αποτελέσματα.

Οι συσχετισμοί αυτοί μας βοηθάνε να δούμε συγκεντρωτικά τη στάση του κόσμου και πως συνδέουν τις διάφορες λέξεις-έννοιες μεταξύ τους. Θεωρητικά έστω ότι κάποιος δεν είχε καμία επίγνωση της υφιστάμενης υγειονομικής κατάστασης, θα μπορούσε από μια τέτοια λεκτική ανασκόπηση να διαμορφώσει εικόνα για το τι συμβαίνει εκεί έξω.

Εξερευνώντας τα αποτελέσματα για την λέξη “mRna” ο πρώτος συσχετισμός με 98% ομοιότητα, είναι η λέξη “breastfeeding”. Ανατρέχοντας στα συλλεχθέντα tweets με κλειδί αναζήτησης “breastfeeding”, από το περιεχόμενο τους εύκολα γίνεται αντιληπτή η ανησυχία του κόσμου σχετικά με το εμβόλιο και αν είναι ασφαλές για όσες γυναίκες θηλάζουν και όσες είναι εγκυμονούσες. Αυτό επιβεβαιώνει και άρθρο δημοσιευμένο στο Harvard Health Publishing (Ilona T. Goldfarb, 2021) με τίτλο “Wondering about COVID-19 vaccines if you’re pregnant or breastfeeding?”

Όσο αφορά την θεραπεία “**treatment**”, το μοντέλο την συσχετίζει άμεσα με την λέξη ηλικιωμένοι με ομοιότητα που αγγίζει το 99%, κάτι το οποίο αποτελεί γενική αλήθεια εφόσον τα συμπτώματα της πανδημίας στα άτομα μεγαλύτερης ηλικίας είναι πολύ έντονα και χρήζουν ιατροφαρμακευτικής περίθαλψης σε αντίθεση με το νεότερο πληθυσμό.



Ένας ενδιαφέρον συσχετισμός είναι της λέξης “**Moderna**” με το δίγραμμα “Warp-speed”. Κάτι το οποίο εκ πρώτης όψης δεν βγάζει κάποιο νόημα. Ύστερα όμως από αναζήτηση ο όρος αποτελεί το Operation Warp Speed και είναι μια συνεργασία δημόσιου-ιδιωτικού τομέα που ξεκίνησε από την κυβέρνηση των ΗΠΑ για τη διευκόλυνση και την επιτάχυνση της ανάπτυξης, κατασκευής και διανομής εμβολίων, θεραπευτικών και διαγνωστικών COVID-19 (Wikipedia, 2020). Αυτή η πληροφορία δείχνει την σοβαρότητα της πανδημίας και πως έχει οργανωθεί ένας ολόκληρος κρατικός οργανισμός για την δημιουργία εμβολίου για την καταπολέμησή της.

Όσο αφορά την προέλευση της πανδημίας η λέξη “Wuhan” έχει συσχετιστεί κατά σειρά με τις λέξεις laboratory, biological, produces και coincidence. Από τον καιρό που εξαπλώθηκε η πανδημία και έγινε γνωστή στον υπόλοιπο κόσμο εκτός Κίνας, μεγάλο μέρος των συζητήσεων αφορούσαν την δημιουργία ενός βιολογικού όπλου σε εργαστήριο στην πόλη Wuhan της Κίνας και μάλιστα ότι αυτό δεν είναι μια απλή σύμπτωση. Στην ίδια συλλογιστική κινείται και η λέξη “China” με τις λέξεις massive, avoid, funded και manufactured έρχεται να συμπληρώσει την θεωρία της κατασκευής ενός μαζικού ιού.

Τέλος αξιοσημείωτα είναι και τα αποτελέσματα της λέξης “Temperature”. Οι θετικές συσχετίσεις κατά σειρά εμφάνισης τους προσδίδουν κατά κάποιο τρόπο τις «τεχνικές προδιαγραφές» του εμβολίου της Pfizer. Δηλαδή αποθήκευσης σε συνθήκες ψύχους και ακριβώς στην θερμοκρασία των -70 C.

Πίνακας 17 Θετικοί συσχετισμοί μοντέλου Word2Vec

Word	Similar positive
Treatment	[('elderly', 0.9954040050506592), (('already', 0.9908822178840637), (('volunteers', 0.9902311563491821), (('health_care', 0.9900848865509033),]
Mrna	[('breastfeeding', 0.9829506278038025), (('pregnant', 0.9774898290634155), (('longterm', 0.9773955345153809), (('tested', 0.9772953391075134), (('symptoms', 0.9763244390487671)]

Covid	[('news', 0.8726587295532227), ('pfizer', 0.8457980155944824), ('document', 0.8406938910484314), ('covidvaccine', 0.8301105499267578), ('corona', 0.8246301412582397),]
pfizer	[('ministers_changed', 0.8202019929885864), ('law_give', 0.8178598284721375), ('pharmaceutical', 0.8162838220596313), ('lol', 0.8128117322921753), ('giving_immunity', 0.8118007183074951), ('event_complications', 0.811510443687439), ('sued', 0.8102775812149048), ('protections', 0.809025764465332), ('well_manufacturers', 0.8079605102539062)]
Vaccine	[('sarscov2', 0.8426002264022827), ('discuss', 0.8029270172119141), ('approves', 0.7895500659942627), ('health', 0.787270188331604)]
Moderna	[('astrazeneca', 0.9700623154640198), ('warp_speed', 0.9663490056991577), ('vaccines', 0.9287846088409424), ('genetic', 0.9253599643707275), ('funding', 0.90350741147995), ('nothing', 0.8945678472518921), ('effectiveness', 0.8933610916137695)]

Wuhan	[('laboratory', 0.9988857507705688), (('biological', 0.9976306557655334), (('produces', 0.9877507090568542), (('coincidence', 0.9781057834625244), (('managed', 0.9753451347351074))]
Sputnik	[('leave', 0.9934360384941101), (('march', 0.9934170246124268), (('production', 0.992794394493103), (('feel', 0.992425262928009), (('working', 0.9918422698974609), (('numbers', 0.9914372563362122))]
III	[('id', 0.9974687695503235), (('really', 0.996260404586792), (('life', 0.9961361885070801), (('anyone', 0.995961606502533))]
China	[('massive', 0.989712119102478), (('avoid', 0.9880062341690063), (('funded', 0.9876366853713989), (('manufactured', 0.98360276222229)]
Approved	[('use', 0.9870569109916687), (('vaccinations', 0.9615805149078369), (('planned', 0.9566890597343445), (('uk', 0.9464610815048218), (('pfizerbiontech', 0.9442018270492554), (('begin', 0.937335729598999)]
temperature	[('low', 0.9960033893585205),

	('required', 0.9950451850891113), ('kept', 0.994460940361023), ('cold', 0.9931341409683228), ('requires', 0.990925669670105), ('chain', 0.9888975620269775), ('needs', 0.988602876663208), ('storage', 0.9882780313491821), ('70c', 0.9860047101974487)]
Us	(['already', 0.9836288094520569), ('speed', 0.9805641174316406), ('phase', 0.9764883518218994), ('treatment', 0.9761033058166504), ('expect', 0.973534107208252)]

5.4 Ταξινόμηση

Σε αυτή την ενότητα παρουσιάζονται οι αλγόριθμοι που χρησιμοποιήθηκαν στα διαφορετικά μοντέλα δημιουργίας χαρακτηριστικών για τις ανάγκες της μηχανικής μάθησης. Η συλλογιστική πορεία που ακολουθήθηκε όσο αφορά το κομμάτι της έρευνας εμπλέκει 4 διαφορετικά μοντέλα από το απλούστερο στο πιο σύνθετο. Η έννοια απλούστερο εμπλέκει όχι μόνο την απλότητα του κώδικα στην υλοποίηση των πειραμάτων αλλά και την δέσμευση λιγότερων υπολογιστικών πόρων. Η ίδια λογική ίσχυσε και τα δοκιμές των αλγορίθμων.

Για την αξιολόγηση των αλγορίθμων έχει χρησιμοποιηθεί η μέθοδος της python "train_test_split", η οποία έχει παραμετροποιηθεί έτσι ώστε να χωριστούν τα δεδομένα εκπαίδευσης στο 70% και τα δεδομένα ελέγχου στο 30%. Συνολικά λόγω περιορισμένων υπολογιστικών πόρων χρησιμοποιήθηκαν 10.000 παραδείγματα.

5.4.1 Bag of Words

Αυτό το μοντέλο παραγωγής χαρακτηριστικών παρουσίασε τις χαμηλότερες αποδόσεις όσο αφορά τις μετρικές αξιολόγησης. Έγινε προσπάθεια βελτίωσης των εξετασθέντων αλγορίθμων με την δημιουργία ισοκατανομής. Κατά αυτόν τον τρόπο, εξισορροπήθηκαν τα δεδομένα στις κλάσεις εισόδου. Εξετάστηκαν 1863 παραδείγματα για κάθε μια από τις κλάσεις positive, negative και neutral

Πίνακας 18 Ισοκατανομή δεδομένων εισόδου

```
# create a dataframe for each value of sentiment
pos = dataset[dataset['sentiment']=='positive']
neg = dataset[dataset['sentiment']=='negative']
neu = dataset[dataset['sentiment']=='neutral']

# find the minimum lenght of dataframes above
min_size = min(len(pos),len(neg),len(neu))
print(min_size)

# sampling the values before slicing
rand_pos = pos.sample(min_size)
rand_neg = neg.sample(min_size)
rand_neu = neu.sample(min_size)
print(pos.shape,neg.shape,neu.shape)

# storing data to new normalized dataset
ED_dataset = pd.concat([rand_pos,rand_neg,rand_neu])

# reset index
ED_dataset.reset_index(drop=True)
```

Ο αλγόριθμος κ-NN διερευνήθηκε για κοντινότερους γείτονες από κ=3 έως κ=6. Εφαρμόστηκε στα πρώτα στάδια της διαδικασίας της ταξινόμησης καθώς το ζητούμενο ήταν η επιβεβαίωση της λειτουργικότητας του μοντέλου των χαρακτηριστικών σε πρώτη φάση με έναν απλοϊκό αλγόριθμο. Λόγω των μετρικών που παρουσιάζει, κρίνεται ως μη κατάλληλος αλγόριθμος για την ταξινόμηση συναισθήματος και στην συνέχεια των δοκιμών απορρίφθηκε. Από τις τιμές του κ που ελέγχθηκαν καλύτερες μετρικές συγκεντρώνει η τιμή κ=5 κοντινότερων γειτόνων.

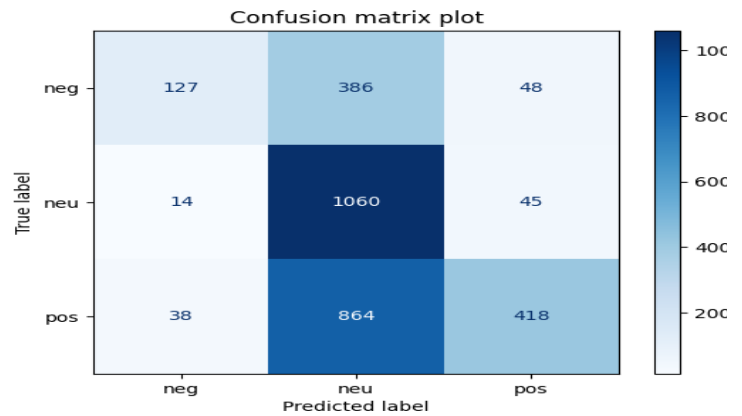
Για να μπορέσουμε να καταλάβουμε την αποτελεσματικότητα του μοντέλου Bag of words συγκριτικά με τα υπόλοιπα μοντέλα που αναλύονται στην παρούσα εργασία, επιλέχθηκε ο αλγόριθμος Naïve Bayes. Ο συγκεκριμένος αλγόριθμος παρουσίασε σχετικά ικανοποιητικά αποτελέσματα με ακρίβεια που φτάνει στο 68% και υψηλή F-αρμονική στα θετικά και ουδέτερα συναισθήματα. Έτσι λοιπόν χρησιμοποιείται ως μέσο σύγκρισης για την επιλογή του καταλληλότερου μοντέλου.

Αλγόριθμος κ-NN:

- knn k: 3

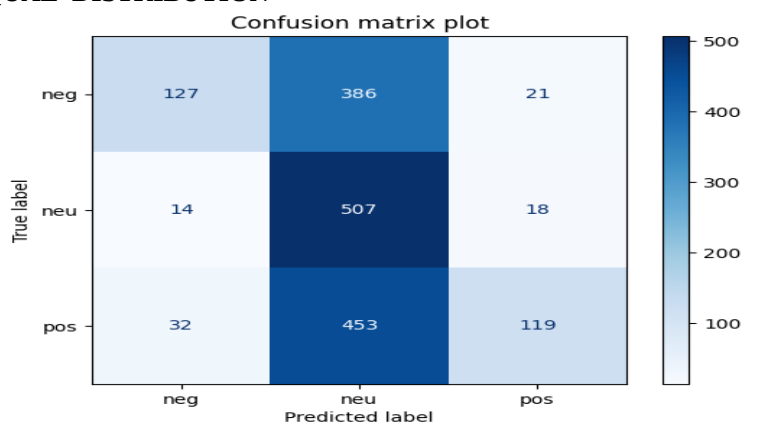
-----WITHOUT EQUAL DISTRIBUTION-----

	Precision	Recall	F1-score
Negative	0.71	0.23	0.34
Neutral	0.46	0.95	0.62
Positive	0.82	0.32	0.46
Accuracy			0.54



-----WITH EQUAL DISTRIBUTION-----

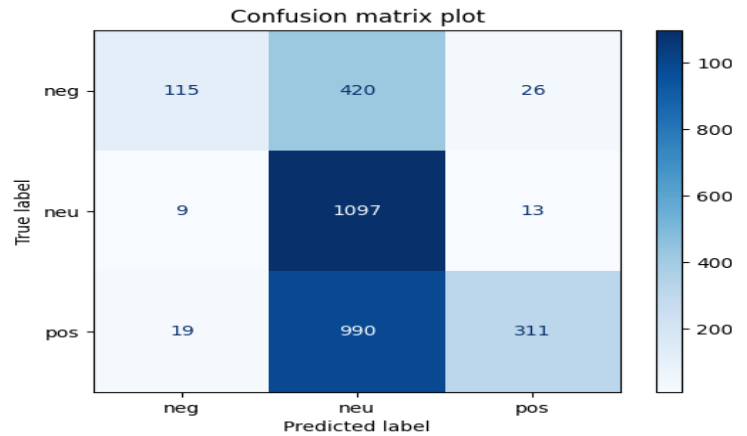
	Precision	Recall	F1-score
Negative	0.73	0.24	0.36
Neutral	0.38	0.94	0.54
Positive	0.75	0.20	0.31
Accuracy			0.45



- knn k: 4

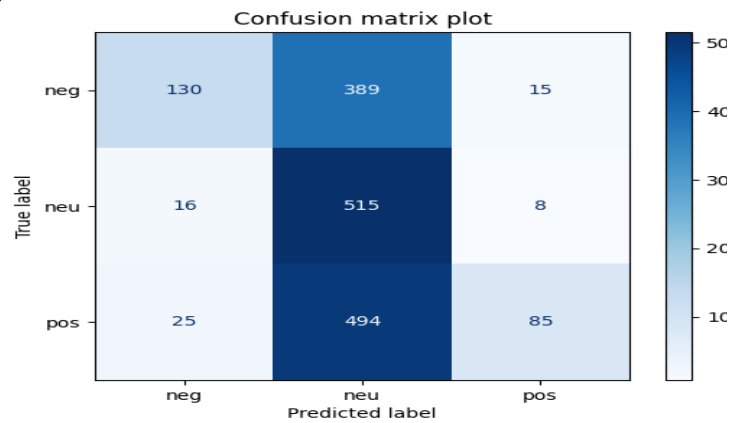
-----WITHOUT EQUAL DISTRIBUTION-----

	Precision	Recall	F1-score
Negative	0.80	0.20	0.33
Neutral	0.44	0.98	0.61
Positive	0.89	0.24	0.37
Accuracy			0.51



-----WITH EQUAL DISTRIBUTION-----

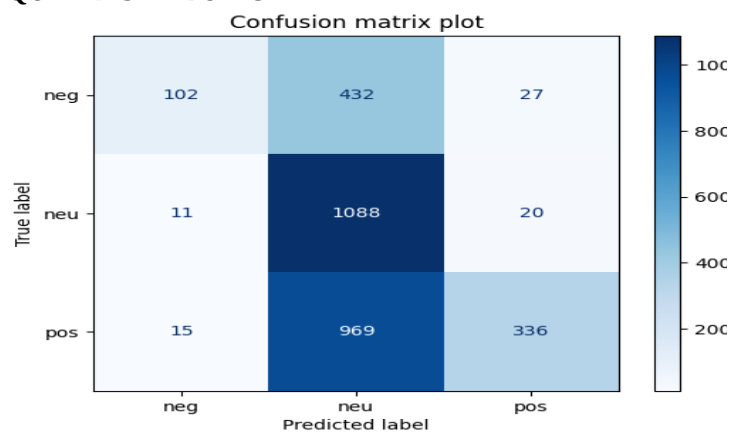
	Precision	Recall	F1-score
Negative	0.76	0.24	0.37
Neutral	0.37	0.96	0.53
Positive	0.79	0.14	0.24
Accuracy			0.44



- knn k: 5

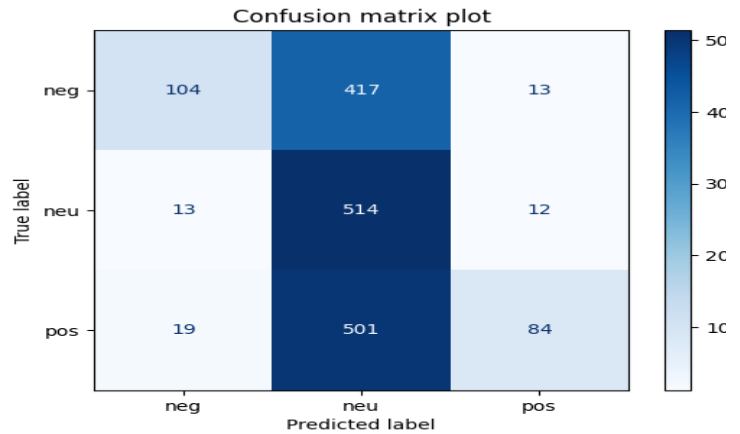
-----WITHOUT EQUAL DISTRIBUTION-----

	Precision	Recall	F1-score
Negative	0.80	0.18	0.30
Neutral	0.44	0.97	0.60
Positive	0.88	0.25	0.39
Accuracy			0.51



-----WITH EQUAL DISTRIBUTION-----

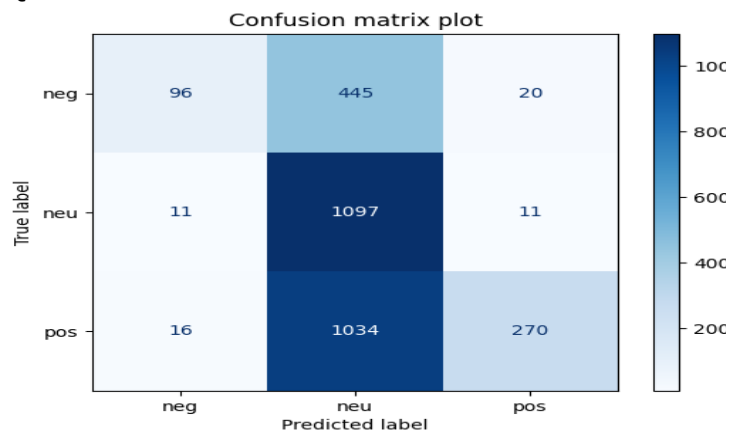
	Precision	Recall	F1-score
Negative	0.76	0.19	0.31
Neutral	0.36	0.95	0.52
Positive	0.77	0.14	0.24
Accuracy			0.42



- knn k: 6

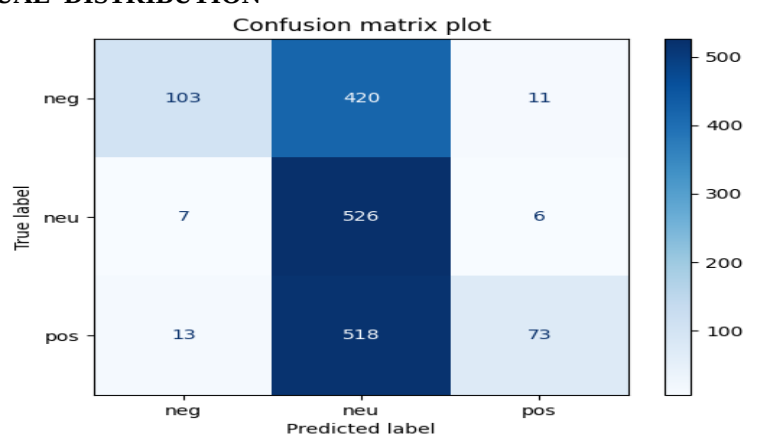
-----WITHOUT EQUAL DISTRIBUTION-----

	Precision	Recall	F1-score
Negative	0.78	0.17	0.28
Neutral	0.43	0.98	0.59
Positive	0.90	0.20	0.33
Accuracy			0.49



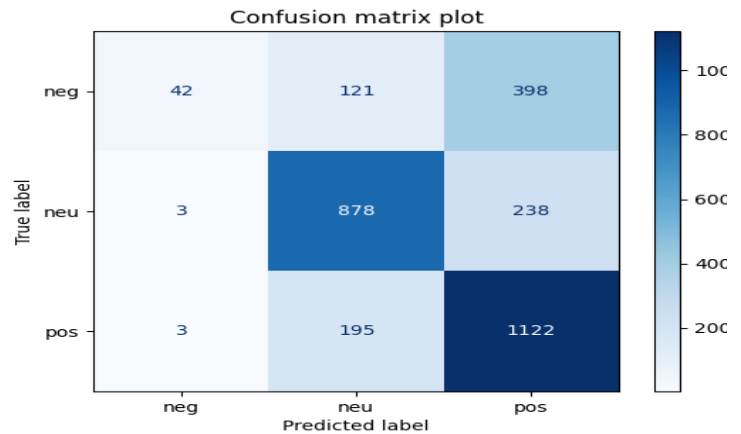
-----WITH EQUAL DISTRIBUTION-----

	Precision	Recall	F1-score
Negative	0.84	0.19	0.31
Neutral	0.36	0.98	0.53
Positive	0.81	0.12	0.21
Accuracy			0.42



Αλγόριθμος Naïve Bayes:

	Precision	Recall	F1-score
Negative	0.88	0.07	0.14
Neutral	0.74	0.78	0.76
Positive	0.64	0.85	0.73
Accuracy			0.68

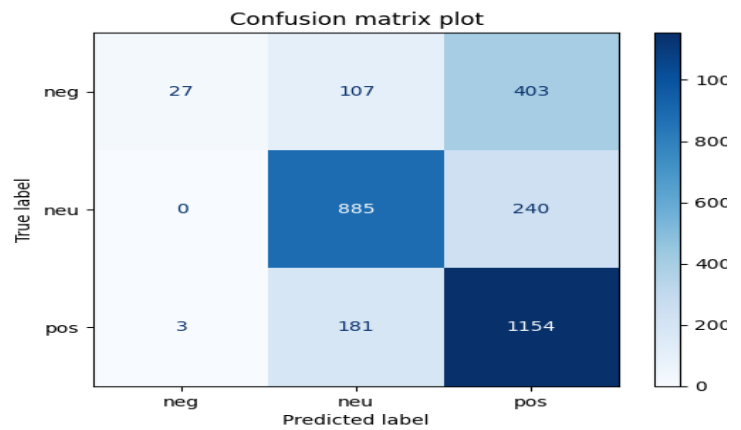


5.4.2 TF-IDF

Σε αυτό το μοντέλο παρουσιάζονται τα καλύτερα αποτελέσματα σε σχέση με τα υπόλοιπα και έτσι γίνεται πειραματισμός με επιπλέον αλγορίθμους. Κατά σειρά εφαρμόστηκαν η λογιστική παλινδρόμηση, οι μηχανές υποστήριξης διανυσμάτων (SVM), το τυχαίο δάσος (random forest) και το νευρωνικό δίκτυο. Την πρώτη θέση στις μετρικές καταλαμβάνει ο αλγόριθμος των μηχανών υποστήριξης διανυσμάτων και στη συνέχεια ακολουθούν με φθίνουσα σειρά το νευρωνικό δίκτυο, η λογιστική παλινδρόμηση και το τυχαίο δάσος.

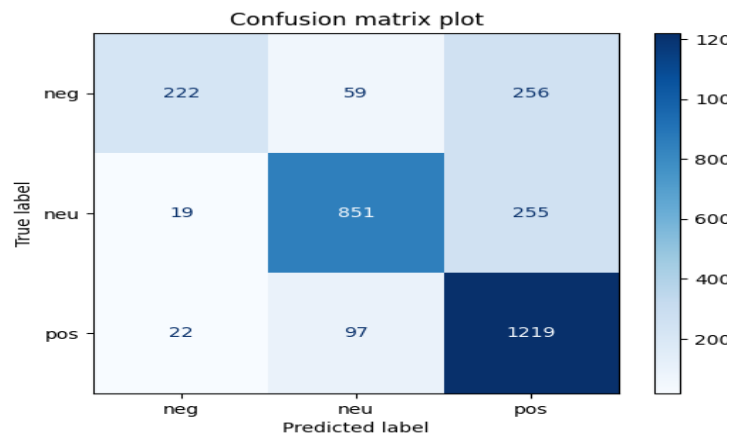
Αλγόριθμος Naïve Bayes:

	Precision	Recall	F1-score
Negative	0.90	0.05	0.10
Neutral	0.75	0.79	0.77
Positive	0.64	0.86	0.74
Accuracy			0.69



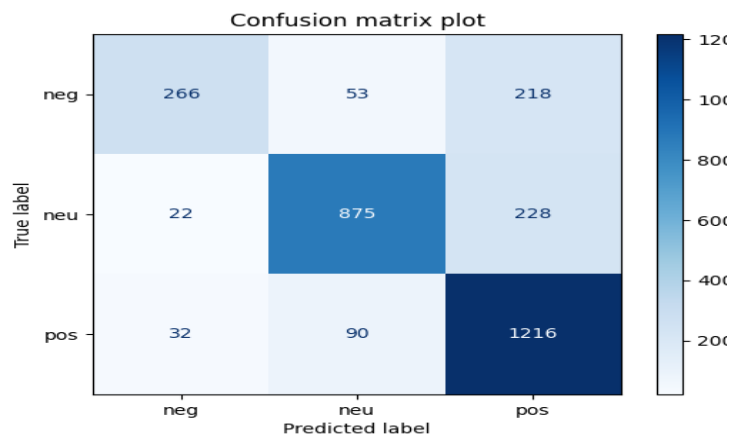
Λογιστική παλινδρόμηση:

	Precision	Recall	F1-score
Negative	0.84	0.41	0.56
Neutral	0.85	0.76	0.80
Positive	0.70	0.91	0.79
Accuracy			0.76



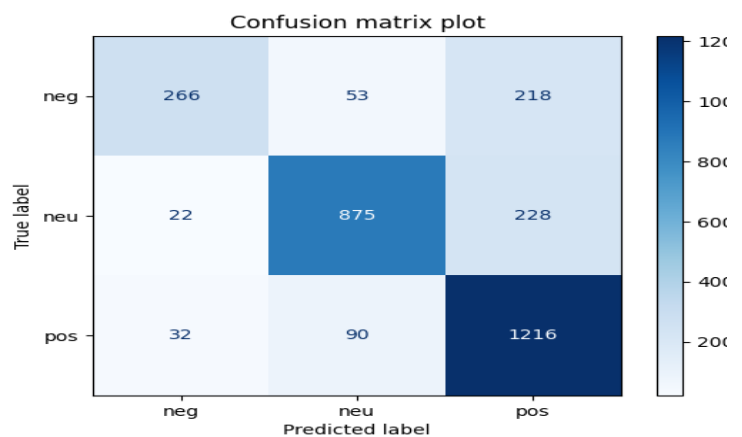
Μηχανές υποστήριξης διανυσμάτων:

	Precision	Recall	F1-score
Negative	0.83	0.53	0.62
Neutral	0.86	0.78	0.82
Positive	0.73	0.91	0.81
Accuracy			0.79



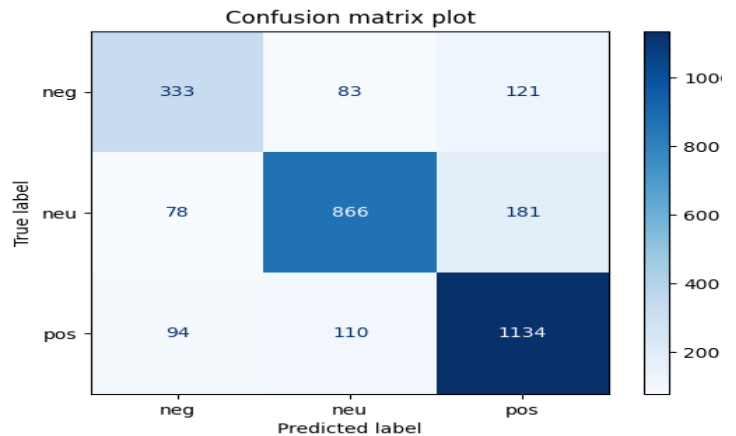
Random Forest:

	Precision	Recall	F1-score
Negative	0.91	0.26	0.41
Neutral	0.76	0.82	0.79
Positive	0.70	0.85	0.77
Accuracy			0.73



Neural Network:

	Precision	Recall	F1-score
Negative	0.66	0.62	0.64
Neutral	0.82	0.77	0.79
Positive	0.79	0.85	0.82
Accuracy			0.78

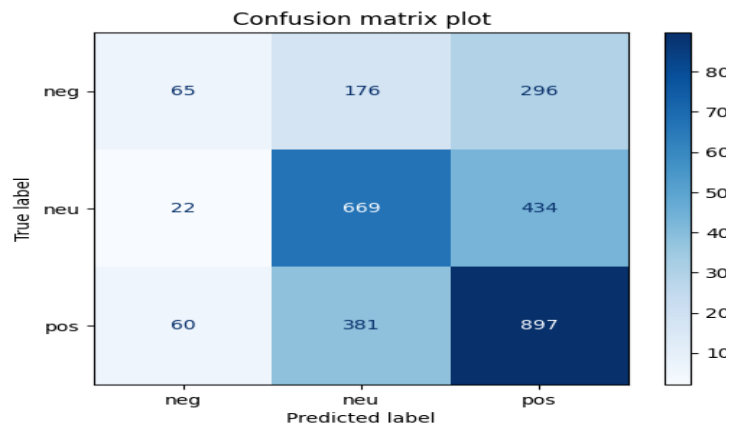


5.4.3 Doc2vec

Σε αυτό το μοντέλο ολόκληρο το κείμενο του tweet παράγει ένα διάνυσμα και αυτός είναι ο λόγος που δεν επιτυγχάνεται σωστά η ταξινόμηση και οι μετρικές συγκεντρώνουν χαμηλά ποσοστά. Έτσι λοιπόν δεν συνεχίστηκαν οι δοκιμές με αυτή την μεθοδολογία και το μοντέλο κρίνεται ως ακατάλληλο για την ταξινόμηση μικρών κειμένων όπως είναι τα εξεταζόμενα tweets.

Αλγόριθμος Naïve Bayes:

	Precision	Recall	F1-score
Negative	0.44	0.12	0.12
Neutral	0.55	0.59	0.57
Positive	0.55	0.67	0.61
Accuracy			0.54

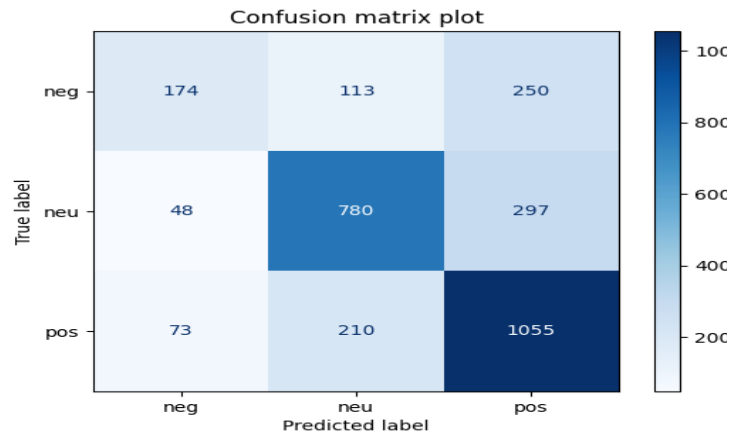


5.4.4 Word2vec

Η βασική ιδέα είναι ότι τα σημασιολογικά διανύσματα (όπως αυτά που παρέχονται από το Word2Vec) θα πρέπει να διατηρούν τις περισσότερες από τις σχετικές πληροφορίες σχετικά με ένα κείμενο ενώ έχουν σχετικά χαμηλή διάσταση που επιτρέπει καλύτερη επεξεργασία μηχανικής μάθησης από την απλή κωδικοποίηση λέξεων. Ωστόσο οι μετρικές που παρουσίασε το συγκεκριμένο μοντέλο είναι ελαφρώς υποδεέστερες από το TF-IDF.

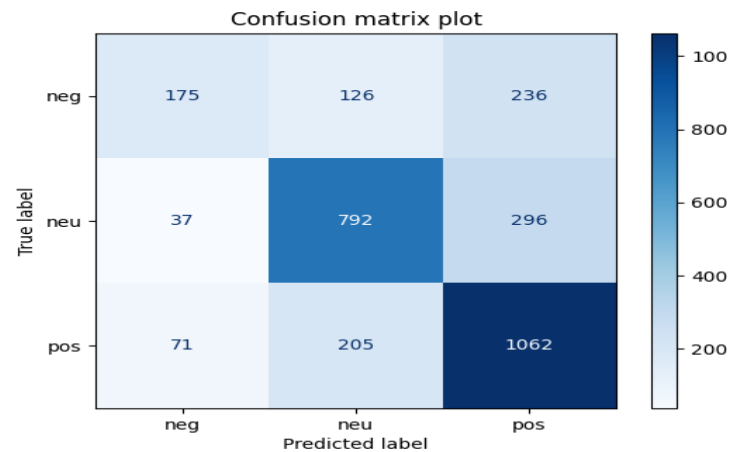
Αλγόριθμος Naïve Bayes:

	Precision	Recall	F1-score
Negative	0.59	0.32	0.42
Neutral	0.71	0.69	0.70
Positive	0.66	0.79	0.72
Accuracy			0.67



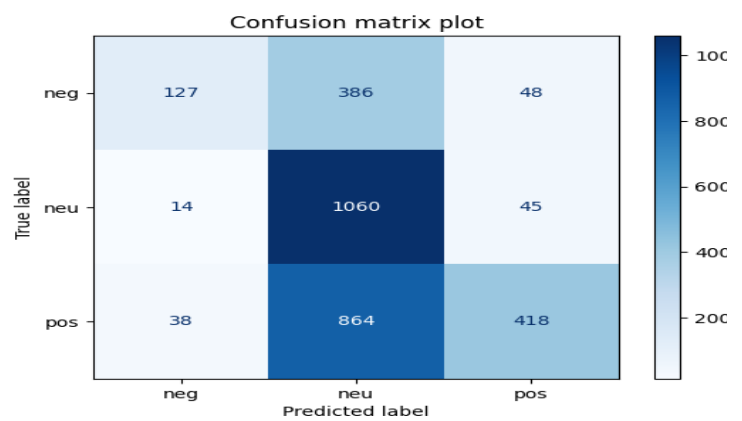
Λογιστική παλινδρόμηση:

	Precision	Recall	F1-score
Negative	0,62	0.33	0.43
Neutral	0.71	0.70	0.70
Positive	0.67	0.79	0.72
Accuracy			0.68



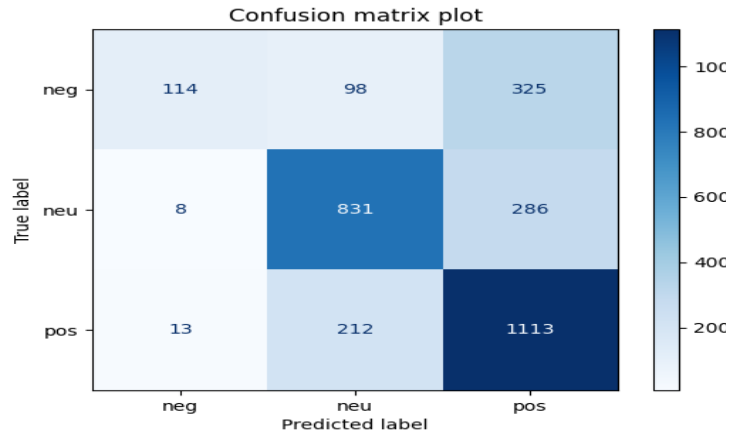
Αλγόριθμος δένδρο απόφασης:

	Precision	Recall	F1-score
Negative	0.76	0.09	0.16
Neutral	0.79	0.28	0.41
Positive	0.50	0.94	0.65
Accuracy			0.54



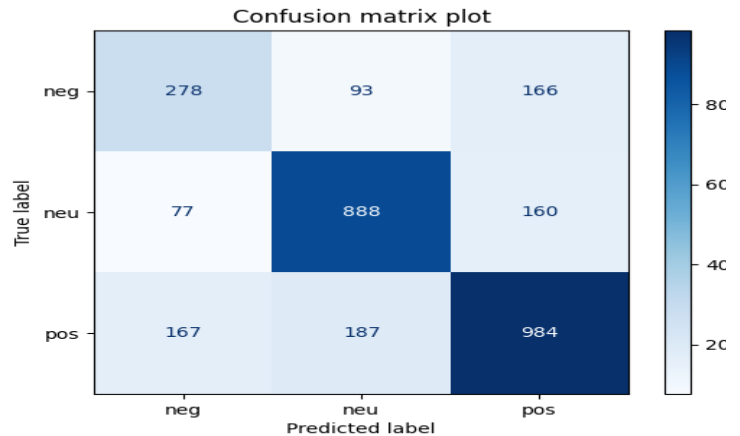
Αλγόριθμος Random Forest:

	Precision	Recall	F1-score
Negative	0.84	0.21	0.34
Neutral	0.73	0.74	0.73
Positive	0.65	0.83	0.73
Accuracy			0.69



Νευρωνικό δίκτυο:

	Precision	Recall	F1-score
Negative	0.53	0.52	0.53
Neutral	0.76	0.79	0.77
Positive	0.75	0.74	0.74
Accuracy			0.72



5.4.4 Δεκαπλή διασταυρωτική μέθοδος

Εφόσον στα προηγούμενα βήματα βρέθηκε το μοντέλο που ευνοεί περισσότερο τους αλγορίθμους στην ταξινόμηση κειμένου για την κατηγοριοποίηση του συναισθήματος, θα πρέπει σε αυτή την φάση να βελτιστοποιηθεί ο τρόπος με τον οποίο αξιολογείται ο αλγόριθμος που παρουσίασε τις καλύτερες μετρικές.

Για τον λόγο αυτό επιλέχθηκε η δεκαπλή διασταυρωτική μέθοδος (10-Fold cross validation). Κατά την μέθοδο αυτή τα συνολικά παραδείγματα χωρίζονται τυχαία σε δέκα ίσα μέρη, εκ των οποίων 9 μέρη χρησιμοποιούνται για την εκπαίδευση του μοντέλου και το υπόλοιπο 1 μέρος χρησιμοποιείται για τον έλεγχο. Το δεύτερο που συμβαίνει σε αυτή την διαδικασία είναι ότι σε κάθε επανάληψη υπολογίζεται και ένα σφάλμα. Το συνολικό σφάλμα του μοντέλου θα είναι ο μέσος όρος του σφάλματος που δημιουργείται σε κάθε επανάληψη.

cross validation

```
In [98]: from sklearn.model_selection import cross_val_score, cross_val_predict

In [94]: clf = svm.SVC(kernel='linear', C=1)
          scores = cross_val_score(clf, X, y, cv=10)

In [95]: print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), scores.std() * 2))

Accuracy: 0.81 (+/- 0.02)
```

Εικόνα 13 αποτέλεσμα διασταυρωτικής μεθόδου

Έτσι λοιπόν ο αλγόριθμος των μηχανών υποστήριξης (SVM) όπως αυτός χρησιμοποιήθηκε στο μοντέλο Tf-Idf παρουσίαζε ακρίβεια 79% ενώ πλέον με την εφαρμογή της δεκαπλής διασταυρωτικής μεθόδου παρουσιάζει $81 \pm 2\%$.

6 Συμπεράσματα-Προτάσεις

Η python αποτελεί μια γλώσσα προγραμματισμού που παρέχει πολλές δυνατότητες και αποτελεί ένα πολύ δυνατό εργαλείο για διαδικασίες όπως η εξόρυξη και ανάλυση δεδομένων. Η διαχείριση των δεδομένων με το πακέτο Pandas παρέχει ένα βελτιστοποιημένο οπτικά περιβάλλον που δεν διαφέρει σε κάτι από ένα υπολογιστικό φύλλο όπως για παράδειγμα ένα αρχείο τύπου excel. Κατά αυτόν τον τρόπο ο χρήστης μπορεί να διαχειρίζεται εύκολα μεγάλα σετ από δεδομένα, να κάνει παρεμβάσεις όπως φιλτράρισμα των δεδομένων σύμφωνα με συγκεκριμένες συνθήκες ή τμηματοποίηση και να ενώνει δεδομένα από διαφορετικές πηγές με απλές εντολές.

Αρχικά έγινε προσπάθεια για συλλογή δεδομένων με την χρήση της Python σε συνδυασμό με το Selenium, ωστόσο αυτή η μέθοδος λειτουργούσε με την ταχύτητα του φυλλομετρητή. Δηλαδή το scrapping «μιμείται» τα βήματα που θα ακολουθούσε ο χρήστης όπως στο manual τρόπο συλλογής δεδομένων αλλά με ταχύτητα μηχανής. Αρχικά βρίσκεις το πεδίο αναζήτησης, πληκτρολογείς τη λέξη κλειδί και όταν έχεις πάρει το κείμενο από όλα τα μηνύματα που εμφανίζονται στην σελίδα μπροστά σου, πρέπει να πατήσεις το κουμπί επόμενο ώστε να συνεχίσει η διαδικασία. Αυτός ο τρόπος λειτούργησε αλλά στερούνταν σε απόδοση και ταχύτητα. Όταν κάποιο element από τα στοιχεία του κώδικα HTML δεν μπορούσε να βρεθεί τότε η διαδικασία κολλούσε και χρειαζόταν εκκίνηση εκ νέου. Έτσι λοιπόν αυτή η μεθοδολογία απορρίφθηκε.

Σε αντίθεση με την προηγούμενη προσέγγιση ιδιαίτερα χρήσιμη και καθοριστική αποδείχτηκε η λειτουργία του Twitter Api για την συλλογή δεδομένων και αποτέλεσε την μέθοδο συλλογής των δεδομένων που χρησιμοποιήθηκαν για τους σκοπούς της παρούσας εργασίας. Το API είναι κατασκευασμένο με τέτοιο τρόπο ώστε να παρέχει τις κατάλληλες εντολές στον χρήστη ώστε να μπορέσει να εκτελέσει με απλές γραμμές κώδικα όλες τις λειτουργίες του Twitter. Από την δημοσίευση tweet, αναζήτηση με βάση οποιοδήποτε φίλτρο (ημερομηνία, γλώσσα, χρήστη, τοποθεσία κτλπ) ακόμα και διαχείριση προφίλ όλα γίνονται εύκολα και αυτοματοποιημένα. Στην πρώτη περίπτωση η συλλογή δεδομένων αποτελούσε θέμα ωρών ενώ πλέον με το Api χιλιάδες δεδομένα αναζητούνται και συλλέγονται σε μόλις λίγα λεπτά.

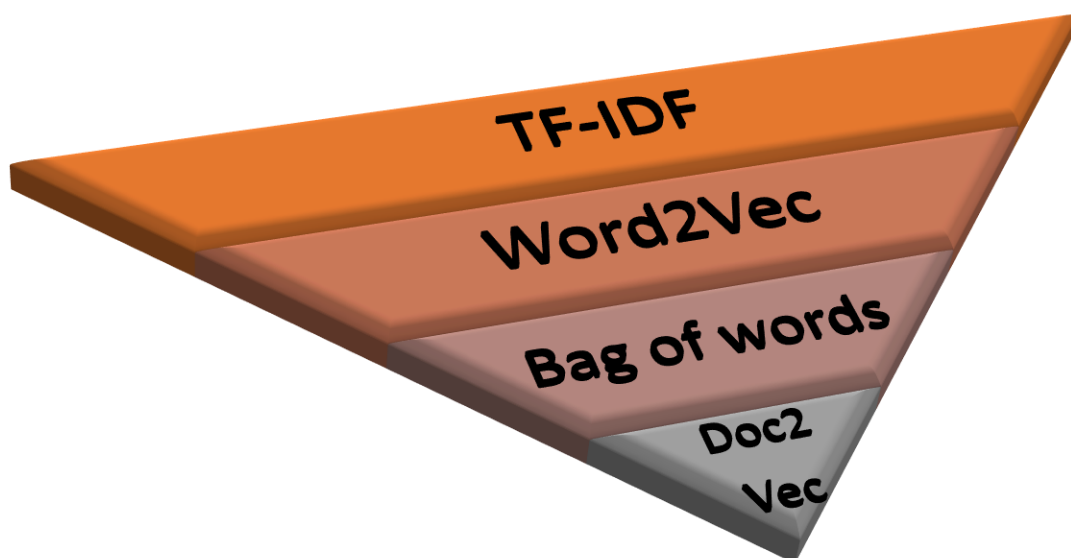
Η προεπεξεργασία των δεδομένων είναι καθοριστικής σημασίας για την σωστή λειτουργία των τεχνικών επεξεργασίας που επιλέχθηκαν. Σκοπός ήταν να μειωθεί ο θόρυβος που προκαλείται στα δεδομένα και να καθαριστούν τα δεδομένα από περιττή πληροφορία. Η φύση όμως των δεδομένων κειμένου αποτελεί από μόνη της μια πρόκληση. Τα tweets που δεν αποτελούν κείμενο επίσημου λόγου, μπορεί να περιλαμβάνουν ασυνταξίες, emoticons, αριθμούς και περιττά σημεία στίξης. Η χρησιμοποίηση των κανονικών εκφράσεων απάλλαξε τα δεδομένα από τα φαινόμενα αυτά ωστόσο υπήρξε δυσκολία στην διαχείριση των τυπογραφικών λαθών. Χρησιμοποιήθηκε η μέθοδος του TextBlob correct_spelling για την διόρθωση των λέξεων αλλά υπήρξαν περιπτώσεις αν και ελάχιστες όπου η αντικατάσταση της λέξης δεν ταίριαζε εννοιολογικά.

Η ανάλυση των συναισθημάτων που αφορούν τα εμβόλια για την πανδημία στηρίχτηκε σε μια υπόθεση ότι θα υπάρχει εμπράκτως ένα μεγάλο ποσοστό του κοινωνικού συνόλου που θα τάσσεται υπέρ των εμβολίων. Αυτό πιστώνεται με την ανάλυση μέσω TextBlob. Οι αρνητές και όσοι δυσφορούν με την ιδέα των εμβολιασμών για την καταπολέμηση της πανδημίας ή ακόμα δεν έχουν πειστεί ότι είναι αληθινή είναι ένα μικρό κομμάτι της κοινής γνώμης που κυμαίνεται από 18% έως 28%. Κάτι το οποίο είναι ιδιαίτερα χρήσιμο σαν συμπέρασμα διότι για να αποκτήσει ανοσία μια κοινωνία αρκεί να εμβολιαστεί το 70% του πληθυσμού της, πράγμα το οποίο επαληθεύεται με βάση την παρούσα ανάλυση.

Για να μπορέσουμε να κατανοήσουμε καλύτερα τα γεγονότα και να να αντιληφθούμε τι έπαιξε καθοριστικό ρόλο στη δημόσια συζήτηση, χρησιμοποιήθηκε το πακέτο Gensim. Το μοντέλο Word2Vec δίνει εξαιρετικά αποτελέσματα με τις συσχετίσεις που παρήχθησαν να αποτυπώνουν τα γεγονότα της περιόδου. Τα σημεία βαρύτητας του δημόσιου διαλόγου αποτελούν οι λέξεις "mRna", "Pfizer", "Wuhan" και "treatment".

Στην συνέχεια τα επεξεργασμένα δεδομένα χρησιμοποιήθηκαν για την παραγωγή ενός μοντέλου ταξινόμησης και για να επιτευχθεί αυτό χρησιμοποιήθηκε η μηχανική μάθηση. Τα δεδομένα χωρίστηκαν σε 70% για εκπαίδευση και 30% για αξιολόγηση. Στα αρχικά στάδια έγινε προσπάθεια με απλούς αλγόριθμους όπως ο K-nn, όμως οι μετρικές αξιολόγησης και ο πίνακας σύγχυσης δεν επέφεραν τα αναμενόμενα αποτελέσματα. Για να βελτιωθεί αυτό έγινε προσπάθεια με ισοκατανομή των παραδειγμάτων στις κλάσεις εισόδου. Η σκέψη αυτή στηρίχτηκε στη θεώρηση να εκπαιδεύονται οι αλγόριθμοι ομοιόμορφα κάτι που στη πράξη δεν αποδείχτηκε. Αντιθέτως αυτό έφερε χειρότερα αποτελέσματα καθώς επηρεάστηκε η εντροπία του δείγματος και έτσι η τεχνική αυτή απορρίφθηκε.

Για να ξεκινήσει η διαδικασία της ταξινόμησης απαραίτητο είναι να μετατραπούν οι λέξεις του κάθε tweet σε χαρακτηριστικά. Αρχικά επιλέχθηκε το μοντέλο Bag of words και στην συνέχεια διερευνήθηκαν και τα υπόλοιπα μοντέλα ώστε να βρεθεί το καταλληλότερο για τις ανάγκες της ταξινόμησης των δεδομένων του data-set. Η σειρά αποτελεσματικότητας απεικονίζεται στο παρακάτω σχήμα με σειρά αποτελεσματικότητας από πάνω προς τα κάτω.



Εικόνα 14 Ιεράρχηση μοντέλων ως προς αποτελεσματικότητα

Από τους αλγόριθμους που δοκιμάστηκαν καταλληλότερος κρίνεται ο αλγόριθμος των μηχανών υποστήριξης διανυσμάτων (SVM) όπου σε συνδυασμό με δεκαπλή διασταυρωτική μέθοδο αξιολόγησης παρουσιάζει ακρίβεια 81% +/-2%.

Περιθώρια βελτίωσης μπορούν να υπάρξουν και προτείνεται στο μέλλον η εύρεση πιο αποτελεσματικού τρόπου για το labeling της πολικότητας των συναισθημάτων (Αρνητικά-Θετικά-Ουδέτρα) πέραν της μεθόδου του TextBlob. Αυτό αποτελούσε και μια από τις προκλήσεις της εργασίας καθώς η κατηγοριοποίηση με βάση το λεξικό δεν επιφέρει πάντοτε τέλεια αποτελέσματα.

Στην διαδικασία της επεξεργασίας της φυσικής γλώσσας μπορεί να ενσωματωθεί και η χρησιμοποίηση των emoticon για την βελτίωση των αποτελεσμάτων, κάτι το οποίο παραλήφθηκε στην παρούσα εργασία και προτείνεται για βελτίωση των παραχθέντων μοντέλων στο μέλλον.

Επιπλέον μπορεί να γίνει έρευνα σε δεδομένα συγκεκριμένης γεωγραφικής περιοχής με την χρησιμοποίηση κριτηρίου γεοεντοπισμού στην μέθοδο αναζήτησης του API. Ενδιαφέρον θα παρουσίαζε να συγκεντρωθούν δεδομένα από διάφορες χώρες του κόσμου και να γίνει συγκριτική αξιολόγηση και ερμηνεία αυτών.

Το πακέτο Gensim παρέχει την δυνατότητα για μη επιβλεπόμενη μάθηση, δηλαδή την ανακάλυψη πιθανής δομής που μπορεί να κρύβεται πίσω από μη χαρακτηρισμένα δεδομένα, κάτι που θα μπορούσε να εφαρμοστεί στο μέλλον στο συγκεκριμένο data-set.

Αναφορές

- Apoorv Agarwal, B. X. (2011, 6 23). Sentiment Analysis of Twitter Data. *Department of Computer Science Columbia University New York, NY 10027 USA*, σσ. 30-38.
- Chhinder Kaur, A. S. (2020). Twitter Sentiment Analysis on Corona virus using Textblob. *Easy Chair*.
- Despoina Chatzakou, N. S. (2015, 04 17). Retweeting Activity on Twitter: Signs of Deception. *Pacific-Asia Conference on Knowledge Discovery and Data Mining*.
- Hung M, L. E. (2020, 8 18). Social Network Analysis of COVID-19 Sentiments: Application of. *JOURNAL OF MEDICAL INTERNET RESEARCH*.
- Ilona T. Goldfarb, M. M. (2021, 1 7). Wondering about COVID-19 vaccines if you're pregnant or breastfeeding? Ανάκτηση από Harvard Medical Publishing: <https://www.health.harvard.edu/blog/wondering-about-covid-19-vaccines-if-youre-pregnant-or-breastfeeding-2021010721722>
- Loria, S. (2012). Ανάκτηση από <https://textblob.readthedocs.io/en/dev/>.
- Nikhil Rajput, B. G. (2020, 04 08). *WORD FREQUENCY AND SENTIMENT ANALYSIS OF TWITTER*. Ανάκτηση από <https://arxiv.org/abs/2004.03925>.
- Roesslein, J. (2009-2021). Ανάκτηση από <https://tweepy.readthedocs.io/en/latest/>.
- Saif Hassan, H. Y.-M. (2012). Semantic Sentiment Analysis of Twitter. *Conference Paper The semantic web – iswc 2012*, σσ. 508–524.
- Statista. (2020). Ανάκτηση από Worldwide digital population as of October 2020: <https://www.statista.com/statistics/617136/digital-population-worldwide/>
- Steven Bird, E. L. (2002). NLTK: The Natural Language Toolkit. *Department of Computer and Information Science University of Pennsylvania, Philadelphia, PA 19104-6389, USA*.
- Tomas Mikolov, K. C. (2013). Efficient Estimation of Word Representations in.
- Wikipedia. (2003). Ανάκτηση από https://en.wikipedia.org/wiki/Logistic_regression
- Wikipedia. (2007). Ανάκτηση από https://en.wikipedia.org/wiki/Random_forest
- Wikipedia. (2020). Ανάκτηση από https://en.wikipedia.org/wiki/Operation_Warp_Speed

Παράρτημα

Στο παράρτημα παρατίθεται ο τελικός κώδικας σε γλώσσα Python που χρησιμοποιήθηκε για τους σκοπούς της εργασίας.

```
# Import Packages
from time import time
import pandas as pd
import numpy as np
pd.options.display.max_colwidth = 100

#gensim
from gensim.models import Word2Vec, KeyedVectors
from gensim.corpora.dictionary import Dictionary
from gensim.models.phrases import Phrases, Phraser
import multiprocessing

#textBlob
from textblob import TextBlob
from wordcloud import WordCloud
import collections
from collections import Counter
#nltk
import nltk
from nltk.tokenize import word_tokenize, TweetTokenizer
from nltk.stem import WordNetLemmatizer, PorterStemmer #, LancasterStemmer # lancaster
associated with rules
from nltk.corpus import stopwords

#matplotlib
import matplotlib.pyplot as plt
plt.style.use('fivethirtyeight')
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")

#regular expression
import re
#emoji
from emoji import demojize
import string
#os
import os

# sklearn
from sklearn.naive_bayes import BernoulliNB
from sklearn.linear_model import LogisticRegression
from sklearn.neighbors import KNeighborsClassifier
from sklearn.neural_network import MLPClassifier
from sklearn.model_selection import train_test_split, cross_val_score
from sklearn.feature_extraction.text import CountVectorizer, TfidfVectorizer
from sklearn.metrics import confusion_matrix, classification_report, mean_squared_error,
plot_confusion_matrix
```

```

#stylish
import sys
from termcolor import cprint

#read data
os.getcwd()
#change directory
os.chdir('/home/ankorf/Documents/capstone/data')
data_names = os.listdir()

# create lists of data
data=[]
for file in data_names:
    text_file = open(file)
    data.append(text_file.readlines())
    text_file.close()

appended_df=[]
for i,file in enumerate(data):
    this_df = pd.DataFrame(file, columns = ['tweet'])
    this_df["query"] = data_names[i]
    appended_df.append(this_df)

total_df = pd.concat(appended_df)
total_df.shape
total_df.head()

# Remove duplicates, missing values
total_df.drop_duplicates()
total_df.dropna()
# Reset index
total_df.reset_index(drop=True)

# Slice data
df = total_df.iloc[:10000]

# Cleaning tweets
def cleanTxt(text):
    text = re.sub(r'"b"',",",text) # Removing prefix b'
    text = re.sub(r'\n',"",text) # Removing \n
    text = re.sub(r'#',"",text) # Removing hastags
    text = re.sub(r'@[A-Za-z0-9]+'',"",text) #Removing @mentions
    text = re.sub(r'RT[\s]+'',"",text) # Removing RT
    text = re.sub(r'https?:\V\S+',",",text) # Removing hyperlink
    text = re.sub(r'[\^w\s]',"",text)# Removing punctuation
    text = text.lower() # Convert cases to lower
    text = re.sub(r'\b\d+\b',"",text) #Removing numbers
    text = re.sub(r'^a-zA-Z0-9]',',',text) #removing special characters
    text = re.sub(r"([xX:;][-'?.])","",text) # Removing emoji
    text = TextBlob(text).correct() #Correct spelling
    text = re.sub(emoji_pattern,"",text)
    text = demojize(text, delimiters=(' ', ' ')) #Remove emoji
    return text

```

```

# Clean the tweets
df['tweet'] = df['tweet'].apply(cleanTxt)

# Inspect cleaned tweets
df.tweet[:9]

# Create a function to get the subjectivity
def getSubjectivity(text):
    return TextBlob(text).sentiment.subjectivity

# Create a function to get the polarity
def getPolarity(text):
    return TextBlob(text).sentiment.polarity

# Create two new columns 'Subjectivity' & 'Polarity'
df['Subjectivity'] = df['tweet'].apply(getSubjectivity)
df['Polarity'] = df['tweet'].apply(getPolarity)

# Plotting the results
sns.scatterplot(df.Polarity, # X-axis
                df.Subjectivity, # Y-axis
                s=100);

plt.title("Sentiment Analysis", fontsize = 20)
plt.xlabel('← Negative — — — — — Positive →', fontsize=15)
plt.ylabel('← Facts — — — — — Opinions →', fontsize=15)
plt.tight_layout()

f, axs = plt.subplots(1, 2, figsize=(15,5))

sns.distplot(df.Polarity, color="b", ax=axs[0])
axs[0].set_title("Tweet Polarity", fontsize = 20)
axs[0].set_xlabel('← Negative — — — — — Positive →', fontsize=15)
sns.distplot(df.Subjectivity, color="b", ax=axs[1])
axs[1].set_title("Tweet Subjectivity", fontsize = 20)
axs[1].set_xlabel('← Facts — — — — — Opinions →', fontsize=15)

plt.tight_layout()

# word cloud visualization
allWords = ' '.join([twts for twts in df['tweet']])
wordCloud = WordCloud(width=500, height=300, random_state=21,
max_font_size=110).generate(allWords)

plt.imshow(wordCloud, interpolation="bilinear")
plt.axis('off')
plt.show()

# Create tokens

```

```

def get_token(text, stem=False):
    """takes an array of tweets and returns a list of cleaned tokens """

    english_stops = stopwords.words('english') # instantiate
    WL = WordNetLemmatizer() # instantiate
    PS = PorterStemmer()
    sent= []
    clean_sent= []

    for raw in text:
        tokens = raw.split()
        sent.append(tokens)

    for this_sent in sent:
        clear_tokens = [token for token in this_sent if token.isalpha()]
        clear_tokens = [token for token in this_sent if token not in english_stops]
        clear_tokens = [WL.lemmatize(token) for token in this_sent]

    if stem == True:
        clear_tokens = [PS.stem(token) for token in this_sent]
        clean_sent.append(clear_tokens)
        clear_tokens = [token for token in this_sent if len(token)>2]
        #print(clear_tokens)
    return clean_sent

clean_sent = get_token(df['tweet'])
# Inspect tokens
clean_sent[:3]
print('Before cleaning: {} \n\n After cleaning: {}'.format(df.tweet[0],clean_sent[0]))

# Creating phrases, bigrams
phrases = Phrases(clean_sent, min_count=30, progress_per=10000)
bigram = Phraser(phrases)
sentences = bigram[clean_sent]
word_freq = collections.defaultdict(int)
for sent in sentences:
    for i in sent:
        word_freq[i] += 1
len(word_freq)

sorted(word_freq, key=word_freq.get, reverse=True)[:10]

# Initializing word2vec model
cores = multiprocessing.cpu_count() # examine computer capabilities
cores # cores are 3 !! this is because i am runing unix @virtual
      # machine and i have engage 3 cores and 1024 mb of ram
w2v_model = Word2Vec(min_count=20,
                      window=2,
                      size=300,
                      sample=6e-5,
                      alpha=0.03,
                      min_alpha=0.0007,
                      negative=20,
                      workers=cores-1)

```

```

t = time()

w2v_model.build_vocab(sentences, progress_per=10000)

print("Time to build vocab: {} mins'.format(round((time() - t) / 60, 2)))
t = time()

w2v_model.train(sentences, total_examples=w2v_model.corpus_count, epochs=30, report_delay=1)

print("Time to train the model: {} mins'.format(round((time() - t) / 60, 2)))

w2v_model.wv.most_similar(negative=["mrna"])
w2v_model.wv.most_similar(positive=["mrna"])
w2v_model.wv.most_similar(negative=["covid"])
w2v_model.wv.most_similar(positive=["covid"])
w2v_model.wv.most_similar(negative=["pfizer"])
w2v_model.wv.most_similar(positive=["pfizer"])
w2v_model.wv.most_similar(negative=["moderna"])
w2v_model.wv.most_similar(positive=["moderna"])

# Classification
# Create a function to get the sentiment
def getSentiment(polarity):
    if polarity <0 :
        sentiment = 'negative'
    elif polarity == 0:
        sentiment='neutral'
    else:
        sentiment= 'positive'
    return sentiment

# Create new column sentiment
df['sentiment'] = df['Polarity'].apply(getSentiment)

def count_sentiment(sentiment):
    count_pos=0
    count_neg=0
    count_neu=0
    for senti in sentiment:
        if senti == 'negative':
            count_neg+=1
        elif senti == 'positive':
            count_pos+=1
        else:
            count_neu+=1
    return print('positive sentiments: {} \nnegative sentiments: {} \nneutral sentiments:
    {}'.format(count_pos,count_neg,count_neu))
count_sentiment(df.sentiment)

dataset = df[['tweet','sentiment']]

```

```

# Equal distribution
# create a dataframe for each value of sentiment
pos = dataset[dataset['sentiment']=='positive']
neg = dataset[dataset['sentiment']=='negative']
neu = dataset[dataset['sentiment']=='neutral']

# find the minimum length of dataframes above
min_size = min(len(pos),len(neg),len(neu))
print(min_size)

# sampling the values before slicing
rand_pos = pos.sample(min_size)
rand_neg = neg.sample(min_size)
rand_neu = neu.sample(min_size)
print(pos.shape,neg.shape,neu.shape)

# storing data to new normalized dataset
ED_dataset = pd.concat([rand_pos,rand_neg,rand_neu])

# sampling new dataset
rand_dataset = normalized_dataset.sample(len(normalized_dataset))

# reset index
ED_dataset.reset_index(drop=True)

# store data
text , sentiment = get_token(dataset['tweet'],stem= True) , list(dataset['sentiment'].values)
ED_text, ED_sentiment = get_token(ED_dataset['tweet'],stem= True) , list(ED_dataset['sentiment'].values)
def untokenize(text):
    """ This function takes a list of tokenized tweets and returns the untokenized tweet """
    untokenize = []
    for i in range(len(text)):
        this_untoken = ' '.join(text[i])
        untokenize.append(this_untoken)
    return untokenize
text = untokenize(text)
ED_text = untokenize(ED_text)

# Split Data
X_train, X_test, y_train, y_test = train_test_split(text, sentiment,
                                                    test_size = 0.3, random_state = 212, shuffle = True)
ED_X_train, ED_X_test, ED_y_train, ED_y_test = train_test_split(ED_text, ED_sentiment,
                                                                test_size = 0.3, random_state = 212, shuffle = True)
len(X_train), len(X_test), len(y_train), len(y_test)

# Confusion matrix function
def confusion_matrix(classifier,X_test,y_test):
    """Takes classifier and return confusion matrix plot"""
    plt.style.use('default')
    class_names = ['neg','neu','pos']
    disp = plot_confusion_matrix(classifier,X_test,y_test,
                                display_labels= class_names,
                                cmap=plt.cm.Blues)
    disp.ax_.set_title('Confusion matrix plot')

```

```

plt.show()

# Bag of words
bow_vectorizer = CountVectorizer(max_features=500000, stop_words='english')
bow = bow_vectorizer.fit_transform(text)
Bow.shape
ED_bow = bow_vectorizer.fit_transform(ED_text)
ED_bow.shape
# splitting data into training and validation set
X_train, X_test, y_train, y_test = train_test_split(bow, sentiment, random_state=42, test_size=0.01)

ED_X_train, ED_X_test, ED_y_train, ED_y_test = train_test_split(ED_bow, ED_sentiment,
random_state=42, test_size=0.3)

#Naive bayes
BNBmodel = BernoulliNB(alpha = 2)
BNBmodel.fit(X_train, y_train)
y_pred = BNBmodel.predict(X_test)

print(classification_report(y_test, y_pred))
confusion_matrix(BNBmodel,X_test,y_test)

# TF-IDF vectorizer
X_train, X_test, y_train, y_test = train_test_split(text, sentiment,
                                                    test_size = 0.01, random_state = 212, shuffle = True)

#initializing vectorizer
vectoriser = TfidfVectorizer(ngram_range=(1,2), max_features=5000)
#apply
vectoriser.fit(X_train)

# Transform
X_train = vectoriser.transform(X_train)
X_test = vectoriser.transform(X_test)

# Neural Network
clf = MLPClassifier(solver='lbfgs', alpha=1e-5,
                  hidden_layer_sizes=(500,), random_state=1)
clf.fit(X_train, y_train)

y_pred = clf.predict(X_test)
print(classification_report(y_test, y_pred))
confusion_matrix(clf,X_test,y_test)

# Support Vector Machine
from sklearn import svm
#Create a svm Classifier
clf = svm.SVC(kernel='linear') # Linear Kernel

#Train the model using the training sets
clf.fit(X_train, y_train)

#Predict the response for test dataset
y_pred = clf.predict(X_test)

```

```

print(classification_report(y_test, y_pred))

# Random forest
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
# instantiate the classifier
rfc = RandomForestClassifier(random_state=0)

# fit the model
rfc.fit(X_train, y_train)
confusion_matrix(rfc,X_test,y_test)

y_pred = rfc.predict(X_test)
print(classification_report(y_test, y_pred))

# Logistic Regression
LRmodel = LogisticRegression(C = 2, max_iter = 1000, n_jobs=-1)
LRmodel.fit(X_train, y_train)
y_pred = LRmodel.predict(X_test)
print(classification_report(y_test, y_pred))
confusion_matrix(LRmodel,X_test,y_test)

# Cross validation
from sklearn.model_selection import cross_val_score, cross_val_predict

clf = svm.SVC(kernel='linear', C=1)

scores = cross_val_score(clf, X, y, cv=10)

print("Accuracy: %0.2f (+/- %0.2f)" % (scores.mean(), scores.std() * 2))

y_pred = cross_val_predict(clf, X_train, y_train, cv=10)

# Doc2vec model
from tqdm import tqdm
tqdm.pandas(desc="progress-bar")
from gensim.models import Doc2Vec
from gensim.models.doc2vec import LabeledSentence
import gensim

def add_label(tweets):
    output = []
    for i, s in enumerate(tweets):
        output.append(LabeledSentence(s, ["tweet_" + str(i)]))
    return output

tokenized_tweet = get_token(text)
labeled_tweets = add_label(tokenized_tweet) # label all the tweets

model_d2v = gensim.models.Doc2Vec(dm=1, # dm = 1 for 'distributed memory' model
                                dm_mean=1, # dm_mean = 1 for using mean of the context word vectors

```

```

        vector_size=500, # no. of desired features
        window=5, # width of the context window
        negative=7, # if > 0 then negative sampling will be used
        min_count=5, # Ignores all words with total frequency lower than
5.
        workers=3, # no. of cores
        alpha=0.1, # learning rate
        seed = 23, # for reproducibility
    )

model_d2v.build_vocab([i for i in tqdm(labeled_tweets)])

model_d2v.train(labeled_tweets, total_examples= len(text), epochs=15)

docvec_arrays = np.zeros((len(tokenized_tweet), 500))
for i in range(len(text)):
    docvec_arrays[i,:] = model_d2v.docvecs[i].reshape((1,500))

docvec_df = pd.DataFrame(docvec_arrays)
docvec_df.shape
X_train, X_test, y_train, y_test = train_test_split(docvec_df, sentiment,
                                                    test_size = 0.3, random_state = 212, shuffle = True)
LRmodel = LogisticRegression(C = 2, max_iter = 1000, n_jobs=-1)
LRmodel.fit(X_train, y_train)
y_pred = LRmodel.predict(X_test)
print(classification_report(y_test, y_pred))
confusion_matrix(LRmodel,X_test,y_test)

# Word2Vec Model
tokenized_tweet = get_token(text, stem=True)
size = 500
model_w2v = gensim.models.Word2Vec(
    tokenized_tweet,
    size=size, # desired no. of features/independent variables
    window=5, # context window size
    min_count= 2, # Ignores all words with total frequency lower than 2.
    sg = 1, # 1 for skip-gram model
    hs = 0,
    negative = 10, # for negative sampling
    workers= 3, # no.of cores
    seed = 34
)

model_w2v.train(tokenized_tweet, total_examples= len(tokenized_tweet), epochs=20)

def word_vector(tokens, size):
    """Preparing Vectors for Tweets. Take mean of all the word vectors present in the tweet """
    vec = np.zeros(size).reshape((1, size))
    count = 0
    for word in tokens:
        try:
            vec += model_w2v[word].reshape((1, size))
            count += 1.
        except KeyError: # handling the case where the token is not in vocabulary

```

```
continue
if count != 0:
    vec /= count
return vec
```

```
wordvec_arrays = np.zeros((len(tokenized_tweet), size))
for i in range(len(tokenized_tweet)):
    wordvec_arrays[i,:] = word_vector(tokenized_tweet[i], size)
wordvec_df = pd.DataFrame(wordvec_arrays)
wordvec_df.shape
X_train, X_test, y_train, y_test = train_test_split(wordvec_df, sentiment,
                                                    test_size = 0.3, random_state = 212, shuffle = True)
```

