ΕΘΝΙΚΟ ΜΕΤΣΟΒΙΟ ΠΟΛΥΤΕΧΝΕΙΟ
ΣΧΟΛΗ ΕΦΑΡΜΟΣΜΕΝΩΝ ΜΑΘΗΜΑΤΙΚΩΝ ΚΑΙ ΦΥΣΙΚΩΝ ΕΠΙΣΤΗΜΩΝ

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

ΠΑΤΟΥΧΑ ΕΛΕΝΗ

«Ανίχνευση Ψευδών Ειδήσεων σε Ελληνικό Κείμενο για την περίπτωση του COVID-19 με τη χρήση της γλώσσας προγραμματισμού Python»

ΣΤΗΝ ΕΠΙΣΤΗΜΟΝΙΚΗ ΠΕΡΙΟΧΗ: ΜΑΘΗΜΑΤΙΚΑ

ΕΠΙΒΛΕΠΩΝ: ΣΤΕΦΑΝΕΑΣ ΠΕΤΡΟΣ, Αναπληρωτής Καθηγητής, ΣΕΜΦΕ

ΤΡΙΜΕΛΗΣ ΕΠΙΤΡΟΠΗ

ΣΤΕΦΑΝΕΑΣ ΠΕΤΡΟΣ, Αναπληρωτής Καθηγητής, ΣΕΜΦΕ

ΚΟΛΕΤΣΟΣ ΙΩΑΝΝΗΣ, Αναπληρωτής Καθηγητής, ΣΕΜΦΕ

ΨΑΡΡΑΚΟΣ ΠΑΝΑΓΙΩΤΗΣ, Καθηγητής, ΣΕΜΦΕ

ΑΘΗΝΑ, 10/2021

ΕΥΧΑΡΙΣΤΙΕΣ

Θα ήθελα να ευχαριστήσω την οικογένειά μου και ιδιαίτερα και τον Γιάννη και τον Νικόλα. Επίσης θα ήθελα να ευχαριστήσω την κ. Ολυμπία Βαγγελάτου για τις πολύτιμες συμβουλές της και την βοήθειά της. Τέλος θα ήθελα να ευχαριστήσω τον κ. Πέτρο Στεφανέα για την αμέριστη στήριξη και κατανόηση που έδειξε.

Πατούχα Ελένη

.....

Ονομα Επώνυμο

Περίληψη

Η πανδημία COVID-19 είναι μια τρέχουσα πανδημία με τεράστιες επιπτώσεις στην υγεία του παγκόσμιου πληθυσμού. Ταυτόχρονα το φαινόμενο των ψευδών ειδήσεων που σχετίζονται με τον COVID-19 έχει χαρακτηριστεί σαν μια άλλη μόλυνση, μια επιδημία παραπληροφόρησης και έχει χαρακτηριστεί ως «infodemic». Η εξάπλωση των ψευδών ειδήσεων που αφορούν τον COVID-19 αποτελεί ένα πολύ σοβαρό φαινόμενο με μεγάλες επιπτώσεις στην υγεία.

Σκοπός της εργασίας είναι να συμβάλει στην αυτοματοποίηση της ανίχνευσης των ψευδών ειδήσεων που αφορούν τον COVID-19 σε ελληνικά κείμενα αναλύοντας την αποτελεσματικότητα διαφορετικών μοντέλων επιβλεπόμενης μηχανικής μάθησης με τη χρήση της γλώσσας προγραμματισμού Python.

Για την ανάγκη υλοποίησης της εργασίας συλλέχθηκαν άρθρα από έγκυρη ειδησεογραφική πηγή και άρθρα από πηγές που έχουν χαρακτηριστεί ως πιθανώς αναξιόπιστες από το Greek Hoaxes Detector [1]. Στα άρθρα τα οποία συλλέχθηκαν τοποθετήθηκαν ετικέτες 0 ή 1 ανάλογα με τον αν προέρχονται από αξιόπιστη πηγή ή αν προέρχονται από πηγή που έχει χαρακτηριστεί ως πιθανώς αναξιόπιστη. Το σύνολο δεδομένων που δημιουργήθηκε περιέχει 4715 άρθρα που αφορούν τον COVID-19, εκ των οποίων τα 2664 έχουν ετικέτα με την τιμή 0 και 2051 έχουν ετικέτα με την τιμή 1.

Πραγματοποιήθηκε καθαρισμός του συνόλου δεδομένων και στη συνέχεια υλοποιήθηκε διερευνητική ανάλυση και ανάλυση συναισθήματος με χρήση τεχνικών Εξόρυξης Γνώσης από Κείμενα για την εύρεση ενός κατάλληλου χαρακτηριστικού για την εκπαίδευση των μοντέλων μηχανικής μάθησης. Μετά από την ανάλυση των δεδομένων καταλήξαμε στη συχνότητα εμφάνισης των λέξεων στο κείμενο των άρθρων και στη μέθοδο TF-IDF.

Αναλύθηκε συνδυαστικά η απόδοση των ταξινομητών Binomial Logistic Regression, Naive Bayes Classifier, Support Vector Classifier και Random Forest με βάση μια σειρά από μετρικές και καταλήξαμε στο συμπέρασμα ότι ο Random Forest έχει την καλύτερη απόδοση.

Λέξεις Κλειδιά: Ανάλυση δεδομένων, Κορονοϊός, Εξόρυξη Δεδομένων, Ψευδείς Ειδήσεις, Covid, πανδημία

Abstract

The COVID-19 pandemic is a current pandemic with a huge impact on the health of the world's population. At the same time the phenomenon of false news related to COVID-19 has been characterized as another infection, an epidemic of misinformation and has been characterized as "infodemic". The spread of false news about COVID-19 is a very serious phenomenon with a major impact on health.

The aim of this work is to contribute to the automation of the detection of false news related to COVID-19 in Greek texts by analyzing the effectiveness of different models of supervised machine learning using the Python programming language.

For the need of the work, articles were collected from a valid news source and articles from sources that have been identified as possibly unreliable by the Greek Hoaxes Detector [1]. The collected articles were tagged 0 or 1 depending on whether they come from a reliable source or from a source that has been identified as possibly unreliable. The dataset created contains 4715 articles related to COVID-19, of which 2664 are labeled with the value 0 and 2051 are labeled with the value 1.

The data set was purified and then exploratory and emotion analysis was performed using Text Knowledge Mining techniques to find a suitable feature for the training of machine learning models. After analyzing the data we came to the frequency of words in the text of the articles and the TF-IDF method.

The performance of the Binomial Logistic Regression, Naive Bayes Classifier, Support Vector Classifier and Random Forest classifiers was combined based on a series of metrics and we concluded that Random Forest has the best performance.

Keywords: Data Analysis, Coronavirus, Data Mining, Fake News, Covid, Pandemic

Πίνακας Περιεχομένων Εικόνων

Πίνακας Περιεχομένων

1 Εισαγωγή

1.1 Το φαινόμενο των ψευδών ειδήσεων

1.1.1 Ορισμός των ψευδών ειδήσεων

Ο άνθρωπος μαθαίνει, διαμορφώνει συνείδηση μέσα στο κοινωνικό πλαίσιο. Υιοθετεί απόψεις και συγκροτεί μεγάλο μέρος των αντιλήψεών του με βάση τα δεδομένα που του δίνονται ως αδιαφιλονίκητη γνώση, από ανθρώπους που εμπιστεύεται, όπως οι δάσκαλοι, οι γονείς, οι φίλοι, οι πηγές πληροφόρησης που εμπιστεύεται. Αυτή η κοινωνική μετάδοση της γνώσης βρίσκεται στην καρδιά του ανθρώπινου πολιτισμού. Αλλά πολλές φορές οι πληροφορίες και οι «γνώσεις» είναι λαθεμένες, σκόπιμα ή μη. [2]

Τα ψευδονέα ή ψευδείς ειδήσεις ή πλαστές ειδήσεις (γνωστά και ως fake news), είναι ένα είδος κίτρινου τύπου ή προπαγάνδας που γίνεται με σκόπιμη παραπληροφόρηση ή με φάρσες που διαδίδονται με παραδοσιακά μέσα μαζικής ενημέρωσης ή με τα μέσα κοινωνικής δικτύωσης. [3] Τα ψευδονέα γράφονται και δημοσιεύονται συνήθως με σκοπό να παραπλανήσουν, προκειμένου να βλάψουν έναν οργανισμό, ένα νομικό ή φυσικό πρόσωπο, ή και για οικονομικά ή πολιτικά οφέλη [4] [5] [6], συχνά χρησιμοποιώντας τίτλους με εντυπωσιασμό, ή εξ ολοκλήρου κατασκευασμένους για την αύξηση της αναγνωσιμότητας.

Η σημασία των ψεύτικων ειδήσεων έχει αυξηθεί στον σύγχρονο πολιτικό βίο. Για τα μέσα μαζικής ενημέρωσης, η δυνατότητα προσέλκυσης επισκεπτών στους ιστοτόπους τους είναι απαραίτητη για τη δημιουργία εσόδων από διαφημίσεις μέσω διαδικτύου. Εάν δημοσιεύσουν μια ιστορία με ψευδές περιεχόμενο θα προσελκύσουν χρήστες, προς όφελος των διαφημιζομένων και θα βελτιώσουν τις αξιολογήσεις. Η εύκολη πρόσβαση στα έσοδα από διαφημίσεις μέσω διαδικτύου, η αυξημένη πολιτική πόλωση και η δημοτικότητα των μέσων κοινωνικής δικτύωσης, κυρίως το Facebook News Feed [3], έχουν εμπλακεί στη διάδοση ψευδών ειδήσεων [4] [7] που ανταγωνίζονται με νόμιμες ειδήσεις. Εχθρικοί κρατικοί παράγοντες έχουν επίσης εμπλακεί στη δημιουργία και διάδοση ψευδονέων, ιδίως κατά τη διάρκεια των εκλογών [8].

Τα ψευδονέα υπονομεύουν τη σοβαρή κάλυψη των μέσων ενημέρωσης και δυσκολεύουν τους δημοσιογράφους να καλύπτουν σημαντικές ειδήσεις [9]. Μια ανάλυση από την BuzzFeed διαπίστωσε ότι τα 20 κορυφαία ψευδονέα σχετικά με τις προεδρικές εκλογές στις ΗΠΑ το 2016 διαδόθηκαν

περισσότερο στο Facebook από τις 20 κορυφαίες ιστορίες εκλογών από 19 μεγάλα μέσα μαζικής ενημέρωσης [10] . Οι ανώνυμες ιστοσελίδες που φιλοξενούν ψευδονέα και στερούνται γνωστούς εκδότες έχουν επίσης επικριθεί, επειδή δυσχεραίνουν ακόμα περισσότερο τη δίωξη για τη δυσφήμηση [11].

Ο όρος χρησιμοποιείται επίσης μερικές φορές για να θέσει σε αμφισβήτηση νόμιμες ειδήσεις που εκφράζουν μία αντίθετη πολιτική άποψη [12] [13] . Ο Ντόναλντ Τραμπ κατά τη διάρκεια και μετά την προεκλογική του εκστρατεία και την εκλογή του, διάπλασε τον όρο ψευδονέα «fake news» με αυτή την έννοια όταν τον χρησιμοποίησε για να περιγράψει την αρνητική κάλυψη του ιδίου. Εν μέρει, ως αποτέλεσμα της χρήσης του όρου από τον Τραμπ, ο όρος έχει εξεταστεί ολοένα και περισσότερο και τον Οκτώβριο του 2018 η Βρετανική κυβέρνηση αποφάσισε ότι δεν θα χρησιμοποιεί πλέον τον όρο επειδή είναι ένας "κακώς καθορισμένος και παραπλανητικός όρος που συγκρίνει μια ετερόκλητη κατηγορία ψευδών πληροφοριών, από αυθεντικό σφάλμα έως ξένες παρεμβάσεις στις δημοκρατικές διαδικασίες". [14]

1.1.2 Ο ρόλος των Μέσων Κοινωνικής Δικτύωσης στην διασπορά των ψευδών ειδήσεων

Ο χαρακτήρας και ο ρόλος του διαδικτύου και των Μέσων Κοινωνικής Δικτύωσης καθορίζεται από αυτούς οι οποίοι τα κατέχουν. Ταυτόχρονα το διαδίκτυο και τα Μέσα Κοινωνικής Δικτύωσης έχουν διευκολύνει και επιταχύνει τις διαδικασίες διάχυσης πληροφοριών, όμως την ίδια στιγμή παρέχουν γόνιμο έδαφος για την εξάπλωση παραπληροφόρησης και ψευδών ειδήσεων [15]. Με βάση αυτά τα δύο μπορούμε να συμπεράνουμε ότι το διαδίκτυο και τα Μέσα Κοινωνικής Δικτύωσης μπορούν να χρησιμοποιηθούν ως εργαλείο μαζικής διάδοσης πληροφορίας εξαρτώμενης από τους ιδιοκτήτες τους.

Συνεχώς αναπτύσσονται νέες πιο εκλεπτυσμένες μέθοδοι μαζικής επίδρασης στην κοινωνική συνείδηση. Υπάρχουν πολλές περιπτώσεις που διαπιστώθηκε η λειτουργία διαδικτυακών ρομπότ (bot) στο «Twitter», σε συνδυασμό με πληρωμένα τρολ (internet troll), ακόμη και με ηλεκτρονική πειρατεία(hacking) παραβίασης λογαριασμών «φίλων», ώστε να δημιουργηθεί η εντύπωση ότι μια άποψη είναι εδραιωμένη ευρέως και να ανέβει κάποιο hashtag ψηλά. Όλοι αυτοί στηρίζονται στο ότι οι άνθρωποι αντιδρούν πολλές φορές συναισθηματικά και διαμοιράζονται πρόθυμα παραπληροφόρηση αν αυτή ενισχύει τις προϋπάρχουσες αντιλήψεις ή προκαταλήψεις τους. Η κατανόηση των μεθόδων, των μηχανισμών και της κοινωνικής βάσης πάνω στην οποία στηρίζεται η παραπληροφόρηση είναι ένα πρώτο,

απολύτως απαραίτητο βήμα, ώστε να αποφύγει κανείς να μετατραπεί άθελά του σε αναμεταδότη ψευδών ειδήσεων. [2]

1.2 Η Επικοινωνία της Υγείας

1.2.1 Ορισμός

Η αποτελεσματική επικοινωνία στον τομέα της υγείας είναι ιδιαίτερα σημαντική, καθώς μπορεί να συμβάλλει σχεδόν σε όλες τις πτυχές τις υγειονομικής περίθαλψης. Η επικοινωνία της υγείας ορίζεται σαν οποιουδήποτε τύπου ανθρώπινης επικοινωνίας της οποίας το περιεχόμενο σχετίζεται με την υγεία [16]. Τις τελευταίες δεκαετίες η εφαρμογή και η μελέτη της έχει αναπτυχθεί και εξελιχθεί με ταχείς ρυθμούς. Η αποτελεσματική επικοινωνία της υγείας πλέον αναγνωρίζεται ως βασικό στοιχείο στην αντιμετώπιση των θεμάτων υγείας. [17]

Σημαντική πτυχή της επικοινωνίας της υγείας αποτελεί η επικοινωνία που αφορά την ευρύτερη δημόσια υγεία καθώς μπορεί να οδηγήσει το κοινό σε σωστά ή λάθος συμπεράσματα στο πολύ κρίσιμο ζήτημα της Δημόσιας Υγείας, της προστασίας και της ανάπτυξής της ως κοινωνικό αγαθό. Είναι ευρέως αποδεκτό ότι οι σημαντικότεροι και πιο καθοριστικοί παράγοντες της υγείας είναι κοινωνικοί παράγοντες και η οικονομική κατάσταση και λιγότερο η ατομική συμπεριφορά. Σε αυτό το πλαίσιο γίνεται μεγαλύτερη προσπάθεια εστίασης σε ευρύτερες καμπάνιες δημόσιας υγείας σε σχέση με την επιρροή της ατομικής συμπεριφοράς καθώς θεωρείται πιο αποδοτική προσέγγιση για τη βελτίωση της υγείας. [18] Παρόλα αυτά όταν μια πολιτική δημόσιας υγείας είναι αναποτελεσματική ή επιβλαβής, ολόκληροι πληθυσμοί μπορούν να διατρέξουν σοβαρούς υγειονομικούς κινδύνους. [17]

1.2.2 Παραδείγματα ψευδών ειδήσεων στην επιστήμη της υγείας και οι επιπτώσεις τους

- **Το παράδειγμα του εμβολίου κατά της Ιλαράς- Παρωτίτιδας - Ερυθράς (MMR)**

Το περιοδικό Lancet[1] δώδεκα χρόνια μετά την δημοσίευση της, ανακάλεσε μια μελέτη ορόσημο που έκανε δεκάδες χιλιάδες γονείς στον κόσμο να στραφούν ενάντια στο εμβόλιο κατά της ιλαράς, της παρωτίτιδας και της ερυθράς (MMR), λόγω της συσχέτισης μεταξύ των εμβολιασμών και του αυτισμού. Σε δήλωσή του στις 2 Φεβρουαρίου του 2010 το Βρετανικό ιατρικό περιοδικό ανέφερε ότι πολλά στοιχεία της δημοσίευσης Lancet 1998;351[9103]:637-41 από τον Δρ. Andrew Wakefield και τους συνεργάτες του, είναι λανθασμένα. [19]

Το 1998, ο Δρ Andrew Wakefield, ένας Βρετανός γαστρεντερολόγος, περιέγραψε έναν νέο φαινότυπο αυτισμού που ονομάζεται σύνδρομο παλινδρόμησης αυτισμού-εντεροκολίτιδας υποστηρίζοντας ότι προκλήθηκε από περιβαλλοντικούς παράγοντες όπως ο εμβολιασμός κατά της ιλαράς, της παρωτίτιδας και της ερυθράς (MMR). Η σύνδεση εμβολιασμού-αυτισμού μείωσε την εμπιστοσύνη των γονέων στα προγράμματα εμβολιασμού για τη δημόσια υγεία και δημιούργησε κρίση δημόσιας υγείας στην Αγγλία και ερωτήσεις σχετικά με την ασφάλεια των εμβολίων στη Βόρεια Αμερική. Μετά από 10 χρόνια διαμάχης και διερεύνησης, ο Δρ Wakefield κρίθηκε ένοχος για ηθικά, ιατρικά και επιστημονικά παραπτώματα για τη δημοσίευση της μελέτης που συνέδεε το εμβόλιο με τον αυτισμό. Πρόσθετες μελέτες έδειξαν ότι τα στοιχεία που παρουσιάστηκαν ήταν δόλια. Η υποτιθέμενη σύνδεση αυτισμού-εμβολίου είναι, ίσως, η πιο καταστροφική ιατρική ψευδής είδηση των τελευταίων 100 ετών. [20]

- **Το παράδειγμα της ραγδαίας αύξησης του καπνίσματος στις ΗΠΑ τις δεκαετίες 1950-'60**

Το κάπνισμα αποτελεί έναν από τους πιο σημαντικούς κινδύνους υγείας στον ανεπτυγμένο κόσμο και μια σημαντική αιτία προώρων θανάτων παγκοσμίως. Προκαλεί ένα ευρύ φάσμα ασθενειών, συμπεριλαμβανομένων πολλών τύπων καρκίνου, χρόνιας αποφρακτικής νόσου, εγκεφαλικού επεισοδίου. Επίσης προκαλεί πολύ σοβαρές επιπλοκές στην ανάπτυξη του εμβρύου κατά τη διάρκεια της εγκυμοσύνης. Τις τελευταίες δεκαετίες έχει σημειωθεί μαζική αύξηση της κατανάλωσης καπνού. Σε παγκόσμιο επίπεδο

1 Το Lancet είναι ένα εβδομαδιαίο περιοδικό γενικής ιατρικής. Είναι από τα παλαιότερα και πιο γνωστά γενικά ιατρικά περιοδικά στον κόσμο. (https://en.wikipedia.org/wiki/The_Lancet)

ο δείκτης θνησιμότητας από αιτίες που σχετίζονται με το κάπνισμα αναμένεται να αυξηθεί από 3 εκατομμύρια ετησίως (εκτίμηση 1995) σε 10 εκατομμύρια ετησίως έως το 2030, με το 70% αυτών των θανάτων να συμβαίνουν στις αναπτυσσόμενες χώρες. [21]

Στα μέσα του 20ου αιώνα το κάπνισμα στην Αμερική αυξήθηκε ραγδαία παρόλο που οι έρευνες που αφορούσαν την αύξηση του καρκίνου του πνεύμονα είχαν ξεκινήσει από την δεκαετία του 1920. Οι διαφημιστικές καμπάνιες των Αμερικανικών καπνοβιομηχανιών τις δεκαετίες του 1930 και του 1940 περιελάμβαναν ισχυρισμούς που αφορούσαν την υγεία όπως: «Δε παίρνουν τον αέρα σας» (Camel 1935), «Απαλό στον λαιμό μου» (Lucky Strike, 1937), «Παίξτε με ασφάλεια με το λαιμό σας» (Philip Morris, 1941), «Φρέσκος σαν ορεινός αέρας» (Old Gold, 1946), «βοήθημα για την πέψη» (Camels)[2]. Μέχρι το 1953 η κατανάλωση τσιγάρων αυξάνονταν σταθερά στις ΗΠΑ, όπου το 47% ενήλικων Αμερικανών καταγράφηκαν ως καπνιστές.

Στις αρχές του 1950 άρχισαν να δημοσιεύονται άρθρα στα ιατρικά περιοδικά και στον δημοφιλή τύπο που υποδήλωναν το κάπνισμα ως αιτία του καρκίνου του πνεύμονα. Αυτό είχε σαν αποτέλεσμα την μείωση των πωλήσεων των τσιγάρων το 1953 και στις αρχές του 1954. Τότε οι κατασκευάστριες εταιρίες εισήγαγαν στην αγορά τα «φιλτραρισμένα τσιγάρα» για να αντιμετωπίσουν την ανησυχία για τα προβλήματα υγείας. Τα διαφημιζόμενα οφέλη των φιλτραρισμένων τσιγάρων, ωστόσο, ήταν απατηλά σύμφωνα με έρευνες [22], καθώς οι καπνιστές των φιλτραρισμένων τσιγάρων εισέπνεαν την ίδια ή και περισσότερη πίσσα, νικοτίνη και επιβλαβή αέρια με αυτά που εισέπνεαν οι καπνιστές των μη φιλτραρισμένων τσιγάρων. Τα φίλτρα δεν ήταν φίλτρα με την ουσιαστική έννοια, κάτι που η βιομηχανία το είχε αναγνωρίσει από το 1930, αλλά παρόλα αυτά οι καπνιστές είχαν οδηγηθεί στο συμπέρασμα ότι είναι ασφαλή για την υγεία τους.

Το 1958 τα επιστημονικά δεδομένα που τεκμηρίωναν το κάπνισμα ως αιτιολογικό παράγοντα για τον καρκίνο του πνεύμονα ήταν πλέον αδιάψευστα και οδήγησαν στην πρώτη επίσημη δήλωση από την Υπηρεσία Δημόσιας Υγείας των ΗΠΑ ότι το κάπνισμα αποτελεί αιτία του καρκίνου του πνεύμονα. Τότε οι καπνοβιομηχανία οργάνωσε μαζική εκστρατεία που κράτησε 40 χρόνια για να αμφισβητήσει τα στοιχεία με ισχυρισμούς ότι τα αποδεικτικά στοιχεία ήταν απλώς στατιστικά στοιχεία, ή ότι βασίζονταν σε έρευνες πάνω στα ζώα. [23]

2"They don’t get your wind" (Camel,1935), "gentle on my throat" (Lucky Strike, 1937), "playsafe with your throat" (Philip Morris, 1941), and "Fresh asmountain air" (Old Gold 1946), "aid to digestion" (Camels)

1.3 Η πανδημία COVID-19

Η πανδημία COVID-19 είναι μια τρέχουσα πανδημία που προκλήθηκε από τον κορονοϊό SARS-CoV-2 και αναγνωρίστηκε για πρώτη φορά στην πόλη Ουχάν, πρωτεύουσα της επαρχίας Χουπέι της Κίνας, τον Δεκέμβριο του 2019. Ως και τις 22 Νοεμβρίου 2020 είχαν επιβεβαιωθεί πάνω από 58,5 εκατομμύρια κρούσματα σε 215 χώρες και περιοχές, είχαν σημειωθεί περισσότεροι από 1,38 εκατομμύρια θάνατοι που οφείλονται στη νόσο και είχαν ανακάμψει περισσότεροι από 40,2 εκατομμύρια άνθρωποι.

Η πανδημία έχει κηρυχθεί από τον Παγκόσμιο Οργανισμό Υγείας (Π.Ο.Υ.) ως «Έκτακτη Ανάγκη Δημόσιας Υγείας Διεθνούς Ενδιαφέροντος» (PHEIC), με βάση τις πιθανές επιπτώσεις που θα μπορούσε να έχει ο ιός εάν εξαπλωθεί σε χώρες με ασθενέστερα συστήματα υγειονομικής περίθαλψης.

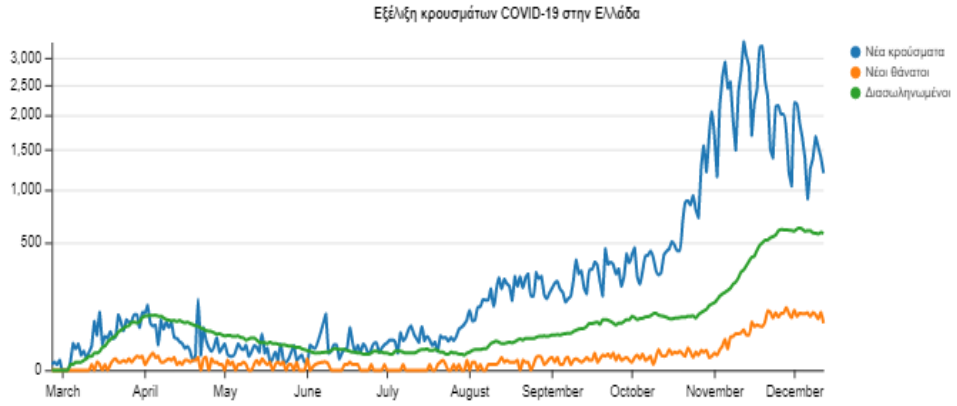
Σε κοινωνικό επίπεδο, έντονη ξενοφοβία και ρατσισμός κατά ανθρώπων κινεζικής και ανατολικής ασιατικής καταγωγής έχουν σημειωθεί λόγω της επιδημίας, τροφοδοτώντας τον φόβο και την εχθρότητα σε διάφορες χώρες. Παραπληροφόρηση που διαδόθηκε κυρίως στο διαδίκτυο σχετικά με τον κορονοϊό, οδήγησε την ΠΟΥ να κηρύξει "πληροφοριοδημία" (αγγλικά: infodemic) στις 2 Φεβρουαρίου. Αντίστοιχα, πολλαπλές θεωρίες συνωμοσίας διαδόθηκαν μεταξύ Μαρτίου και Απριλίου 2020, με αποτέλεσμα κυβερνήσεις ανά τον κόσμο να λάβουν προληπτικά μέτρα για την αντιμετώπισή τους. [24]

1.3.1 Η πανδημία COVID-19 στην Ελλάδα

Ο ιός εμφανίστηκε και εξαπλώθηκε και στην Ελλάδα από τις 26 Φεβρουαρίου 2020 και έπειτα. Η πλειοψηφία των κρουσμάτων που σημειώθηκαν τις πρώτες ημέρες σχετιζόταν με ανθρώπους που ταξίδεψαν στην Ιταλία, μια κύρια επιδημική εστία, και με μια ομάδα προσκυνητών που είχαν ταξιδέψει στο Ισραήλ και την Αίγυπτο, καθώς και επαφές των ατόμων αυτών. Στις 12 Μαρτίου καταγράφηκε ο πρώτος νεκρός στην Ελλάδα και τα πρώτα «ορφανά» κρούσματα, δηλαδή κρούσματα που δεν έγινε δυνατό να ανιχνευθεί ο φορέας από τον οποίο κόλλησαν.

Στα μέσα Σεπτεμβρίου ο αριθμός κρουσμάτων ήταν πολλαπλάσιος του μέγιστου ύψους που είχε στα μέσα Απριλίου, και ο αριθμός νοσηλευμένων σε ΜΕΘ και θανάτων ξεπέρασε τα επίπεδα με τον Απρίλιο, κατά τον οποίο είχαν ήδη επιβληθεί αυστηρά περιοριστικά μέτρα (lockdown). Τον Νοέμβριο ο αριθμός νέος κρουσμάτων έφτασε ως και πάνω από 300 την ημέρα, ο αριθμός νεκρών ως και πάνω 100 την ημέρα, και ο αριθμός διασωληνωμένων πάνω από 600, συνολικά πάνω από τις δυνατότητες

περίθαλψης που διαθέτει το Εθνικό Σύστημα Υγείας, και επιβλήθηκαν νέα περιοριστικά μέτρα (lockdown). [25]. Στην **Εικόνα 1.1** αποτυπώνεται η ημερήσια μεταβολή των κρουσμάτων στην Ελλάδα.



ΕΙΚΟΝΑ 1.1-ΗΜΕΡΗΣΙΑ ΜΕΤΑΒΟΛΗ ΚΡΟΥΣΜΑΤΩΝ COVID-19 ΣΤΗΝ ΕΛΛΑΔΑ

1.3.2 Η διάδοση Ψευδών ειδήσεων για την πανδημία COVID-19 και οι επιπτώσεις τους

Η μεγάλη ζήτηση για πληροφορίες σχετικά με την ασθένεια, τον αντίκτυπό της και τα πολλά αναπάντητα ερωτήματα σχετικά με τον ιό που ανακαλύφθηκε τον Δεκέμβριο του 2020, δημιούργησαν ιδανικό έδαφος για μύθους, ψεύτικες ειδήσεις και θεωρίες συνωμοσίας και παραπληροφόρηση. [26]

Το φαινόμενο των ψευδών ειδήσεων που σχετίζονται με τον COVID-19 έχει χαρακτηριστεί σαν μια άλλη μόλυνση, μια επιδημία παραπληροφόρησης στο διαδίκτυο και έχει ονομαστεί ως “infodemic”. Οι ψευδείς ειδήσεις για τον COVID-19 περιλαμβάνουν θεωρίες συνωμοσίας για την προέλευση του ιού, απάτες, ψευδείς ειδήσεις για κυβερνητικές δραστηριότητες, επικίνδυνες συμβουλές για ψευδείς θεραπείες, αναξιόπιστες αναφορές εμβολίων, φάρσες. Η διάδοση Ψευδών Ειδήσεων για τον Covid-19 έχει προκαλέσει ιδιαίτερη ανησυχία στον Παγκόσμιο Οργανισμό Υγείας. Εκτός από την παραπληροφόρηση που αφορά την προέλευση και τον τρόπο εξάπλωσης του κορονοϊού διαδίδονται ψευδείς θεραπείες κάποιες από τις οποίες περιλαμβάνουν ακόμα και χρήση προϊόντων που είναι επικίνδυνα. [27]

Είναι προφανές ότι η εξάπλωση ψευδών ειδήσεων στη περίπτωση του COVID-19 αποτελεί ένα πολύ σοβαρό φαινόμενο καθώς μπορεί να αποβεί θανατηφόρα. Χαρακτηριστικά στη Νιγηρία, πολλοί άνθρωποι νοσηλεύτηκαν για δηλητηρίαση από χλωροκίνη μετά από δηλώσεις του τότε Αμερικανού

Προέδρου Τραμπ που υποδήλωναν ότι θα μπορούσε να χρησιμοποιηθεί για τη θεραπεία του COVID-19. Στο Ιράν δεκάδες άνθρωποι πέθαναν από κατανάλωση τοξικής μεθανόλης μετά από φήμες ότι «μπορεί να θεραπεύσει το νέο κορωνοϊό» [28] [29].

Ταυτόχρονα αναδεικνύονται και άλλες πλευρές που έχουν να κάνουν με την εμπιστοσύνη στην επιστήμη και στον ρόλο της στην αντιμετώπιση της πανδημίας και των ψευδών ειδήσεων. Υπάρχουν τεράστιες δυνατότητες παγκόσμια, με χιλιάδες επιστήμονες και ερευνητές που μπορούν δυνητικά να οργανώσουν την δουλειά τους στο πλαίσιο σχεδιασμένης συνεργασίας, ώστε να ανταποκριθούν στις έκτακτες ανάγκες της πανδημίας αλλά και στην αντιμετώπιση της παραπληροφόρησης και των ψευδών ειδήσεων. Η αρθρογραφία καταλήγει σε σχέση με τον προβληματισμό που υπάρχει στην επιστημονική κοινότητα στο συμπέρασμα ότι οι επιστήμονες πρέπει να υπερασπιστούν δημόσια την επιστήμη και να συμβάλλουν στην αντιμετώπιση της πανδημίας και στο φαινόμενο αμφισβήτησης της επιστήμης. [26] [30]

2 Θεωρητικά και Τεχνικά Εργαλεία

Στόχος της παρούσας διπλωματικής εργασίας είναι η συμβολή στην αυτοματοποίηση της ανίχνευσης ψευδών ειδήσεων σε ελληνικά κείμενα που αφορούν τον COVID-19, καθώς η διάχυση των ψευδών ειδήσεων που αφορούν την πανδημία αποτελεί ένα εκτεταμένο και σοβαρό φαινόμενο, με μεγάλο υγειονομικό αντίκτυπο στο σύνολο του παγκόσμιου πληθυσμού.

Για την εκπόνηση της εργασίας χρησιμοποιήθηκε η γλώσσα προγραμματισμού Python για την συλλογή, την αποθήκευση, την επεξεργασία, την ανάλυση των δεδομένων και για την δημιουργία των μοντέλων μηχανικής μάθησης. Εφαρμόστηκαν τεχνικές εξόρυξης κειμένου στα ελληνικά κείμενα που συλλέχθηκαν και ανάλυση συναισθήματος. Στη συνέχεια εφαρμόστηκαν τα μοντέλα μηχανικής μάθησης Binomial Logistic Regression, Naive Bayes Classifier, Support Vector Machines και Random Forest ώστε να μελετηθεί η ακρίβεια πρόβλεψης στην ανίχνευση των ψευδών ειδήσεων και να επιλεχθεί το πιο αποδοτικό μοντέλο.

2.1 Η επιστήμη των δεδομένων

Η εξέλιξη της τεχνολογίας έδωσε τη δυνατότητα εξάπλωσης του Internet. Με το πέρασμα του χρόνου, η πρόσβαση στο Internet έγινε προσιτή σε ολοένα και περισσότερους ανθρώπους. Αυτό με τη σειρά του οδήγησε στο να αναπτυχθούν περισσότεροι ιστότοποι και να χρησιμοποιηθούν βάσεις δεδομένων για την αποθήκευση των δεδομένων. Με τη δημιουργία εμπορικών και κοινωνικών ιστοσελίδων υπήρξαν τα πρώτα άλματα στις απαιτήσεις και ανάγκες για αποθήκευση και διαχείριση μεγάλου όγκου δεδομένων. Σήμερα, το πλήθος των διαθέσιμων δεδομένων είναι τεράστιο και αυξάνεται εκθετικά κάθε μέρα. Η μείωση στο κόστος συλλογής και της δυσκολίας στη συλλογή και αποθήκευση των δεδομένων συνετέλεσε σημαντικά στην ανάπτυξη του πεδίου αυτού.

Η Επιστήμη των Δεδομένων είναι ουσιαστικά μία καινούρια επιστήμη, η οποία άρχισε να εμφανίζεται σταδιακά, ξεκινώντας από τα τέλη της δεκαετίας του 1980. Η μεγάλη άνθιση στον τομέα της Επιστήμης των Δεδομένων έλαβε χώρα σταδιακά και ήταν άμεσα εξαρτημένη από τη δυνατότητα που δόθηκε για συλλογή και καταγραφή τεράστιων ποσοτήτων δεδομένων, διαφορετικών μορφών και τύπων, μέσω της ανάπτυξης γρήγορων δικτυακών υποδομών, πάνω στις οποίες μπορούσαν να υποστηριχτούν αξιόπιστες εμπορικές εφαρμογές.

Η άνευ όρων παραγωγή δεδομένων σε εικοσιτετράωρη βάση καλύπτει μια τεράστια γκάμα ανθρώπινων δραστηριοτήτων και όχι μόνο, όπως είναι τα δεδομένα από το καλάθι αγορών, τον ιατρικό

φάκελο του ασθενούς, τις συζητήσεις ή και ανακοινώσεις στα κοινωνικά μέσα δικτύωσης, τις τραπεζικές ή και χρηματιστηριακές συναλλαγές, τα ίχνη κινούμενων οχημάτων, τα δεδομένα αισθητήρων από κινητήρες αεροσκαφών, η καταγραφή συνομιλιών σε κέντρα εξυπηρέτησης πελατών κ.λπ. Τα δεδομένα αυτά διαφέρουν πάρα πολύ μεταξύ τους τόσο σε μορφή (εικόνα, βίντεο, κείμενο, πολυδιάστατα ή πραγματικού χρόνου δεδομένα, ακολουθίες DNA και άλλα πολλά) όσο και στην ταχύτητα συλλογής. Εάν, μάλιστα, δεν υποστούν άμεση ανάλυση, ίσως να είναι ιδιαίτερα δύσκολο να αποθηκευτούν ή να τα επεξεργαστούν οι άνθρωποι, δημιουργώντας έτσι μία καινούρια ερευνητική δράση, γνωστή με τον όρο Μεγάλα Δεδομένα (Big Data). Η Επιστήμη των Δεδομένων στοχεύει σε αυτή τη φάση να καλύψει τις ανάγκες που δημιουργούνται από αυτόν τον νέο τομέα και να προσφέρει λύσεις για την κλιμακούμενη και αποτελεσματική επεξεργασία out-of core (εκτός μνήμης ή εξωτερικής μνήμης) δεδομένων.

Τεχνικές και εργαλεία που χρησιμοποιούνται γι' αυτόν τον σκοπό έχουν ήδη αρχίσει να μορφοποιούνται τόσο στα εργαστήρια όσο και στην αγορά και αποτυπώνονται με όρους όπως Map-Reduce, Hadoop, Hive, MongoDB, GraphPD κ.λπ. Τα δύο τελευταία συστήματα αναφέρονται σε μία ερευνητική περιοχή, που είναι γνωστή και ως NoSQL.

Οι δύο πρωταρχικοί στόχοι στην πρακτική της Επιστήμης των Δεδομένων είναι η δημιουργία μοντέλων, τα οποία να μπορούν να χρησιμοποιηθούν τόσο για την πρόβλεψη, όσο και για την περιγραφή των δεδομένων. Η πρόβλεψη αφορά στη χρήση κάποιων μεταβλητών ή πεδίων μίας βάσης δεδομένων, μέσω των τιμών των οποίων μπορεί να εκτιμηθεί η άγνωστη ή μελλοντική τιμή ενός άλλου γνωρίσματος. Η περιγραφή (σε μορφή σύνοψης ή περιληπτικής παρουσίασης) των δεδομένων εστιάζει στην εύρεση κατανοητών από τον άνθρωπο προτύπων, τα οποία περιγράφουν τα δεδομένα, όπως, δηλαδή, γίνεται κατά την εύρεση συστάδων ή ομάδων αντικειμένων με παρόμοια χαρακτηριστικά. [31]

2.2 Μηχανική Μάθηση

Η Μηχανική Μάθηση (Machine Learning) μπορεί να οριστεί ως: το φαινόμενο κατά το οποίο ένα σύστημα βελτιώνει την απόδοσή του κατά την εκτέλεση μιας συγκεκριμένης εργασίας, χωρίς να υπάρχει ανάγκη να προγραμματιστεί εκ νέου. Βάσει του ορισμού αυτού, η Μηχανική Μάθηση έχει ως σκοπό τη δημιουργία μηχανών ικανών να μαθαίνουν, να βελτιώνουν, δηλαδή, την απόδοσή τους σε κάποιους τομείς μέσω της αξιοποίησης προηγούμενης γνώσης και εμπειρίας.

Ένας σχετικός γενικός ορισμός Μηχανικής Μάθησης δίνεται από τον Mitchell (1997): *«Ένα πρόγραμμα υπολογιστή λέμε ότι μαθαίνει από την εμπειρία Ε ως προς κάποια κλάση εργασιών Τ και μέτρο*

απόδοσης P, αν η απόδοσή του σε εργασίες από το T, όπως μετριέται από το P, βελτιώνεται μέσω της εμπειρίας E.»

Στην Επαγωγική Μάθηση (Inductive Learning), με τη διαδικασία της επαγωγής (induction) ο άνθρωπος μαθαίνει κατανοώντας το περιβάλλον του μέσω παρατηρήσεων και δημιουργεί μια απλοποιημένη (αφαιρετική) εκδοχή του που ονομάζεται νοητικό μοντέλο (mental model). Επιπλέον, ο άνθρωπος έχει τη δυνατότητα να οργανώνει και να συσχετίζει τις εμπειρίες και τις παρατηρήσεις του δημιουργώντας νέες δομές που ονομάζονται νοητικά πρότυπα (mental patterns), με αξιοποίηση και του επαγωγικού και του απαγωγικού συλλογισμού. Στη δημιουργία νέων προτύπων από παλαιά βασίζονται οι τρόποι μάθησης που εξαρτώνται σε μεγαλύτερο ή μικρότερο βαθμό από την προϋπάρχουσα γνώση για ένα πρόβλημα, όπως είναι η μάθηση από επεξηγήσεις και η μάθηση από περιπτώσεις.

Ένας εναλλακτικός ορισμός για τη Μηχανική Μάθηση θα μπορούσε να είναι: Μηχανική Μάθηση ονομάζεται η ικανότητα ενός υπολογιστικού συστήματος να δημιουργεί μοντέλα ή πρότυπα από ένα σύνολο δεδομένων.

Η Μηχανική Μάθηση ασχολείται με τη μελέτη αλγορίθμων που βελτιώνουν τη συμπεριφορά τους σε κάποια εργασία που τους έχει ανατεθεί χρησιμοποιώντας την εμπειρία τους. Αν και απέχουμε πάρα πολύ από τη δημιουργία μηχανών που μαθαίνουν τόσο καλά όσο ο άνθρωπος, για συγκεκριμένες περιοχές μάθησης έχουν αναπτυχθεί αλγόριθμοι οι οποίοι έχουν επιτρέψει την εμφάνιση σύγχρονων εμπορικών εφαρμογών με σημαντική επιτυχία. Επιπλέον, τα αποτελέσματα από τις εφαρμογές της Τεχνητής Νοημοσύνης αρχίζουν ήδη να είναι ορατά και να δίνουν απαντήσεις σε αναπάντητα, έως τώρα, ερωτήματα των άλλων κλάδων που διερευνούν την ικανότητα του ανθρώπου να μαθαίνει.

Εκτός της ίδιας της Τεχνητής Νοημοσύνης, μεταξύ των επιστημονικών κλάδων που επωφελούνται από τα επιτεύγματα στον τομέα της Μηχανικής Μάθησης συγκαταλέγονται οι: Εξόρυξη Δεδομένων, Πιθανότητες και Στατιστική, Θεωρία της Πληροφορίας, Αριθμητική Βελτιστοποίηση, Θεωρία της Πολυπλοκότητας, Θεωρία Ελέγχου (προσαρμοστική), Ψυχολογία (εξελικτική, γνωστική), Νευροβιολογία και Γλωσσολογία. [32]

Ο τομέας της Μηχανικής Μάθησης αναπτύσσει τρεις τρόπους μάθησης, ανάλογους με τους τρόπους με τους οποίους μαθαίνει ο άνθρωπος: επιβλεπόμενη μάθηση, μη επιβλεπόμενη μάθηση και ενισχυτική μάθηση. Πιο αναλυτικά:

- Επιβλεπόμενη Μάθηση (Supervised Learning) είναι η διαδικασία όπου ο αλγόριθμος κατασκευάζει μια συνάρτηση που απεικονίζει δεδομένες εισόδους (σύνολο εκπαίδευσης) σε γνωστές επιθυμητές εξόδους, με απώτερο στόχο τη γενίκευση της συνάρτησης αυτής και για εισόδους με

άγνωστη έξοδο. Χρησιμοποιείται σε προβλήματα Ταξινόμησης (Classification), Πρόγνωσης (Prediction), Διερμηνείας (Interpretation)

-
- Μη Επιβλεπόμενη Μάθηση (Unsupervised Learning), όπου ο αλγόριθμος κατασκευάζει ένα μοντέλο για κάποιο σύνολο εισόδων υπό μορφή παρατηρήσεων χωρίς να γνωρίζει τις επιθυμητές εξόδους. Χρησιμοποιείται σε προβλήματα Ανάλυσης Συσχετισμών (Association Analysis), Ομαδοποίησης (Clustering)

- Ενισχυτική Μάθηση (Reinforcement Learning), όπου ο αλγόριθμος μαθαίνει μια στρατηγική ενεργειών μέσα από άμεση αλληλεπίδραση με το περιβάλλον. Χρησιμοποιείται κυρίως σε προβλήματα Σχεδιασμού (Planning), όπως για παράδειγμα ο έλεγχος κίνησης ρομπότ και η βελτιστοποίηση εργασιών σε εργοστασιακούς χώρους.

Για κάθε πρόβλημα προς επίλυση στο χώρο της Μηχανικής Μάθησης υπάρχει ένας κατάλληλος τρόπος μάθησης και για κάθε τρόπο μάθησης υπάρχει τουλάχιστον ένας κατάλληλος αλγόριθμος που μπορεί να χρησιμοποιηθεί. Ορισμένοι αλγόριθμοι δέχονται ως είσοδο μόνο παρατηρήσεις και άλλοι λαμβάνουν υπόψη τους λίγο ή περισσότερο την προϋπάρχουσα γνώση.

Οι αλγόριθμοι μηχανικής μάθησης που αφορούν την αυτοματοποίηση των διαδικασιών λήψης απόφασης μέσω της διαδικασίας γενίκευσης από ήδη γνωστά παραδείγματα είναι οι πιο διαδεδομένοι. Αυτή η διαδικασία είναι γνωστή ως επιβλεπόμενη μάθηση. Στην επιβλεπόμενη μάθηση ο χρήστης παρέχει στον αλγόριθμο ζεύγη με εισόδους και επιθυμητές εξόδους, ο αλγόριθμος μάθησης εκπαιδεύεται με την αντιστοίχιση των ζευγαριών εισόδου εξόδου και βρίσκει έναν τρόπο να παράγει την επιθυμητή έξοδο σε μια δοσμένη είσοδο που του είναι άγνωστη χωρίς την ανθρώπινη βοήθεια.

Παραδείγματα επιβλεπόμενης μηχανικής μάθησης:

- Προσδιορισμός του ταχυδρομικού κώδικα από χειρόγραφα ψηφία σε φάκελο αλληλογραφίας
- Προσδιορισμός εάν ένας όγκος είναι καλοήθης βάσει μιας ιατρικής εικόνας
- Εντοπισμός δόλιας δραστηριότητας σε συναλλαγές με πιστωτική κάρτα

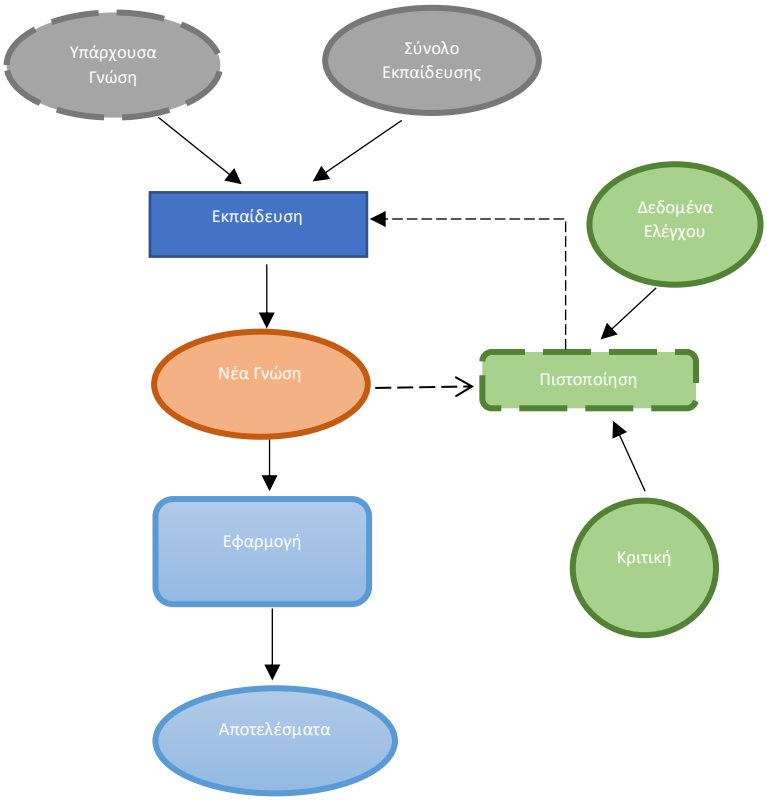
Στη μη επιβλεπόμενη μάθηση είναι γνωστά μόνο τα δεδομένα εισόδου και δε δίνονται γνωστά αποτελέσματα στον αλγόριθμο όπως γίνεται στην εποπτευόμενη μάθηση.

Παραδείγματα μη επιβλεπόμενης μηχανική μάθησης:

- Προσδιορισμός θεμάτων σε ένα σύνολο αναρτήσεων ιστολογίου
- Κατηγοριοποίηση πελατών σε ομάδες σύμφωνα με τις προτιμήσεις τους
- Εντοπισμός μη φυσιολογικών προτύπων πρόσβασης σε έναν ιστότοπο

[33]

Στην **Εικόνα 2.1** αποτυπώνεται ο γενικός τρόπος λειτουργίας των αλγορίθμων Μηχανικής Μάθησης. Η βασικότερη φάση κάθε αλγόριθμου είναι η εκπαίδευση, όπου ο αλγόριθμος χρησιμοποιεί ως είσοδο ένα σύνολο δεδομένων εκπαίδευσης (training set) προς επίτευξη του σκοπού του, τη δημιουργία νέας γνώσης. Επιπλέον, μπορεί είτε να χρησιμοποιήσει λιγότερο ή περισσότερο την υπάρχουσα γνώση είτε να μην τη χρησιμοποιήσει καθόλου. Την εκπαίδευση ακολουθεί η φάση της πιστοποίησης της παραγόμενης νέας γνώσης. Συνήθως, η πιστοποίηση πραγματοποιείται καταρχάς από τον ίδιο τον αλγόριθμο μέσω διαδικασιών ανάκλησης (recall) με τη βοήθεια δεδομένων ελέγχου (test data) και, στη συνέχεια, μέσω κριτικής που κάνει ο χρήστης βάσει των γνώσεων που διαθέτει για το πρόβλημα που επιχειρεί να λύσει ο αλγόριθμος. Τέλος, η νέα γνώση δίνεται προς χρήση σε εφαρμογές στις οποίες είναι απαραίτητη, για να λυθούν πραγματικά προβλήματα. [32]



Εικονα 2.1-Ο Γενικος Τροπος Λειτουργιας των Αλγοριθμων Μηχανικης Μαθησης

2.2.1 Υπερπροσαρμογή(Overfitting) και Υποπροσαρμογή(Underfitting)

Στη Μηχανική μάθηση ένας κίνδυνος είναι η μηχανή να μάθει πολύ καλά τα δεδομένα του δείγματος και στη συνέχεια να είναι λιγότερο ακριβής στον πραγματικό έλεγχο. Αυτό ονομάζεται υπερπροσαρμογή(overfitting). Μια λύση για να το αντιμετωπίσουμε είναι να χωρίσουμε τα δεδομένα σε δύο κομμάτια. Το πρώτο, το κομμάτι της εκπαίδευσης να δίνεται στο μηχάνημα και το δεύτερο, το κομμάτι του ελέγχου, να χρησιμοποιηθεί αργότερα για να ελέγξουμε πόσο καλά μπορεί να λειτουργήσει το μοντέλο σε αυτά τα άγνωστα σε αυτό δεδομένα. Οι πιο διαδεδομένες αναλογίες που χρησιμοποιούνται είναι 80/20, 75/25, 70/30 όπου το μεγαλύτερο μέρος των δεδομένων είναι το δείγμα εκπαίδευσης και το μικρότερο είναι το δείγμα ελέγχου. [34]

Για την εκτίμηση της ακρίβειας ενός μοντέλου πρέπει να χρησιμοποιηθούν παρατηρήσεις διαφορετικές από αυτές που χρησιμοποιήθηκαν για την εκπαίδευση. Με τον όρο υπερπροσαρμογή στα δεδομένα εκπαίδευσης (data overfitting) ορίζουμε το φαινόμενο όπου το μοντέλο «απομνημονεύει» τις περιπτώσεις οι οποίες υπάρχουν στο σύνολο εκπαίδευσης, αντί να εκπαιδεύεται ουσιαστικά, ενσωματώνοντας «κανόνες» γενικότερης ισχύος. Ένα υπερβολικά προσαρμοσμένο μοντέλο ενσωματώνει και τον θόρυβο των δεδομένων. Ακόμα όμως και όταν δεν υπάρχει θόρυβος, η υπερβολική προσαρμογή του μοντέλου στα συγκεκριμένα δεδομένα θα το εμποδίσει να προβλέψει σωστά την κλάση νέων παρατηρήσεων. Η υπερπροσαρμογή παρουσιάζεται όταν ένα μοντέλο είναι υπερβολικά περίπλοκο. Το μοντέλο αυτό είναι ικανό να αφομοιώσει τις ιδιαιτερότητες των δεδομένων εκπαίδευσης, αντί να καταγράψει σχέσεις γενικότερης ισχύος.

Ένα υπερπροσαρμοσμένο μοντέλο επιτυγχάνει εξαιρετικά υψηλές επιδόσεις έναντι των δεδομένων εκπαίδευσης, οι επιδόσεις του όμως έναντι άγνωστων παρατηρήσεων δεν είναι ικανοποιητικές. Για τον λόγο αυτό, εξαιρετικά υψηλός ρυθμός ακρίβειας έναντι του συνόλου εκπαίδευσης, όχι μόνον δεν είναι ασφαλές μέτρο της επιτυχίας του μοντέλου, αλλά αποτελεί ένδειξη πιθανής υπερπροσαρμογής του.

Αντίστροφο πρόβλημα της υπερπροσαρμογής είναι η υποπροσαρμογή. Στην περίπτωση της υποπροσαρμογής, το μοντέλο είναι υπερβολικά απλό για να ενσωματώσει τις ουσιαστικές σχέσεις, οι οποίες υπάρχουν στα δεδομένα εκπαίδευσης. Αποτέλεσμα της υποπροσαρμογής είναι η χαμηλή ακρίβεια έναντι και των δεδομένων εκπαίδευσης και των άγνωστων παρατηρήσεων. [35]

2.2.2 Επιβλεπόμενη Μηχανική Μάθηση

Η επιβλεπόμενη Μηχανική Μάθηση είναι ένας από τους πιο συχνά χρησιμοποιημένους και επιτυχημένους τύπους Μηχανικής Μάθησης. Υπάρχουν δύο βασικοί τύποι επιβλεπόμενης μηχανικής μάθησης, η ταξινόμηση και η παλινδρόμηση.

Στην ταξινόμηση τα δεδομένα εισόδου χωρίζονται σε δύο ή περισσότερες κλάσεις, και η μηχανή πρέπει να κατασκευάσει ένα μοντέλο, το οποίο θα αντιστοιχίζει τα δεδομένα σε μία ή περισσότερες (multi-label) κλάσεις. Στη δυαδική(binary) ταξινόμηση η ετικέτα μπορεί να επιλεχθεί από ένα σύνολο δύο πιθανών τιμών. Ένα παράδειγμα δυαδικής ταξινόμησης είναι η ταξινόμηση ενός email ως είτε ανεπιθύμητο είτε όχι. Στην πολυκλασσική ταξινόμηση η ετικέτα μπορεί να επιλεχθεί από ένα σύνολο με περισσότερες από δύο τιμές. Ένα παράδειγμα πολυκλασσικής ταξινόμησης είναι η ταξινόμηση ενός φρούτου σε ένα από τα είδη μήλο, λεμόνι, μπανάνα, καρπούζι βάσει των χαρακτηριστικών του(σχήμα, μέγεθος, χρώμα κλπ)

Στην παλινδρόμηση, τα αποτελέσματα είναι συνεχή και όχι διακριτά, το ζητούμενο είναι να προβλεφθεί ένας συνεχής αριθμός ή αλλιώς ένας πραγματικός αριθμός. Ένα παράδειγμα παλινδρόμησης είναι η πρόβλεψη του ετησίου εισοδήματος ενός φυσικού προσώπου βάσει κάποιων χαρακτηριστικών του(εκπαίδευση, ηλικία, τόπος διαμονής κλπ). Η προβλεπόμενη τιμή μπορεί να είναι οποιοσδήποτε αριθμός ανήκει σε ένα δεδομένο εύρος τιμών. [33]

Στην επιστήμη των δεδομένων το πρώτο βήμα για την επίλυση ενός συγκεκριμένου προβλήματος είναι ο προσδιορισμός της ερώτησης που πρέπει να απαντηθεί. Ο τύπος της απάντησης που αναζητούμε μπορεί να μας οδηγήσει σε ένα συγκεκριμένο σύνολο τεχνικών που θα ακολουθήσουμε για την επίλυση του προβλήματος.

- Αν η ερώτηση μας απαντηθεί από το σύνολο (NAI,OXI) τότε αντιμετωπίζουμε πρόβλημα ταξινόμησης. Επίσης, το σύνολο απαντήσεων θα μπορούσε να είναι ένα οποιοδήποτε σύνολο με πεπερασμένο αριθμό επιλογών.
- Αν η ερώτησή μας απαντηθεί με ένα πραγματικό αριθμό τότε αντιμετωπίζουμε ένα πρόβλημα παλινδρόμησης. [36]

Όπως έχει προαναφερθεί, στην επιβλεπόμενη μάθηση θέλουμε να εκπαιδεύσουμε ένα μοντέλο με δεδομένα ώστε να το κάνουμε ικανό να προβλέψει το αποτέλεσμα από άγνωστα δεδομένα εισόδου. Όταν καταφέρνουμε το μοντέλο να κάνει ακριβείς προβλέψεις στα άγνωστα δεδομένα λέμε ότι είναι ικανό να γενικεύσει από τα δεδομένα εκπαίδευσης στα δεδομένα ελέγχου. Στόχος είναι να φτιάξουμε ένα μοντέλο με όσο το δυνατόν μεγαλύτερη ικανότητα γενίκευσης. [33]

2.2.3 Γραμμική Παλινδρόμηση (Linear Regression)

Η απλή γραμμική παλινδρόμηση είναι μια γραμμική προσέγγιση για την πρόβλεψη μιας εξαρτημένης μεταβλητής Y δεδομένης μιας ανεξάρτητης μεταβλητής X. Η γραμμική συσχέτιση εκφράζεται μαθηματικά από τη σχέση: $Y \approx \beta_0 + \beta_1 X$, όπου το σύμβολο «≈» δείχνει ότι υπάρχει περίπου γραμμική σχέση, Y η εξαρτημένη μεταβλητή, X η ανεξάρτητη μεταβλητή, β_0 η τεταγμένη, δηλαδή η τιμή της εξαρτημένης για X = 0 και β_1 η κλίση της ευθείας. Τα β_0 και β_1 μαζί είναι γνωστά ως συντελεστές ή ως παράμετροι . Η σχέση $Y = \beta_0 + \beta_1 X + \varepsilon$ μας δίνει την πιο αποτελεσματική γραμμική προσέγγιση της σχέσης μεταξύ X και Y, με ε το τυχαίο σφάλμα και $E(\varepsilon_i) = 0$, που είναι η απόκλιση της Y από την ευθεία γραμμικής παλινδρόμησης $E(y_i) = \beta_0 + \beta_1 x$

Στη διαδικασία της Μηχανικής Μάθησης, μόλις χρησιμοποιηθούν τα δεδομένα εκπαίδευσης για τις εκτιμήσεις $\hat{\beta}_0$ και $\hat{\beta}_1$ για τους συντελεστές του μοντέλου, μπορούμε να προβλέψουμε τις μελλοντικές τιμές της εξαρτημένης μεταβλητής δοσμένης μιας άγνωστης έως τώρα ανεξάρτητης μεταβλητής, χρησιμοποιώντας τη σχέση : $\hat{y} = \hat{\beta}_0 + \hat{\beta}_1 x$. Όπου $\hat{y}$ η πρόβλεψη της μεταβλητής Y δοσμένης μιας X= x. Το σύμβολο «^» το χρησιμοποιούμε για να δηλώσουμε είτε μια εκτιμώμενη τιμή για άγνωστη παράμετρο ή συντελεστή, ή να δηλώσουμε την προβλεπόμενη τιμή απόκρισης.

Ο σκοπός μας είναι να βρούμε τους κατάλληλους συντελεστές β_0, β_1 ώστε να πετύχουμε μια καλή προσαρμογή της ευθείας στα δεδομένα, δηλαδή να ελαχιστοποιηθούν όσο το δυνατόν οι αποκλίσεις των δεδομένων από τις εκτιμώμενες τιμές.

Η πολλαπλή γραμμική παλινδρόμηση είναι επέκταση της απλής γραμμικής παλινδρόμησης και εκφράζεται από τη σχέση: $Y = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p + \varepsilon$. Αντίστοιχα με την απλή γραμμική παλινδρόμηση, στην πολλαπλή γραμμική παλινδρόμηση στόχος είναι να βρούμε τους κατάλληλους συντελεστές β_0, β_1, β_2, ….β_p ώστε να ελαχιστοποιηθούν όσο το δυνατόν οι αποκλίσεις των δεδομένων από τις εκτιμώμενες τιμές.

Μια διαδεδομένη μέθοδος για την εκτίμηση των συντελεστών είναι η μέθοδος των ελαχίστων τετραγώνων όπου όσο πιο αποδοτικό είναι το μοντέλο τόσο πιο μικρό είναι το μέσο τετραγωνικό σφάλμα(mean squared error). Έστω $\hat{y}_i = \hat{\beta}_0 + \hat{\beta}_1 x_i$ η πρόβλεψη του Y για την i τιμή του X. Τότε η τιμή $e_i = y_i - \hat{y}_i$ αντιπροσωπεύει το κατάλοιπο(residual), δηλαδή την διαφορά μεταξύ της πραγματικής τιμής από την πρόβλεψη του γραμμικού μοντέλου. Αν ο αριθμός των παρατηρήσεων μας είναι n, τότε ορίζουμε ως υπόλοιπο άθροισμα τετραγώνων(residual sum of squares) την τιμή:

$$\text{RSS} = \sum_{i=1}^{n} {e_i}^2 = \sum_{i=1}^{n} (y_i - \hat{y}_i)^2 = \sum_{i=1}^{n} [y_i - (\hat{\beta}_0 + \hat{\beta}_1 x_i)]^2$$

Από τη σχέση αυτή λαμβάνουμε τις πρώτες μερικές παραγώγους, τις οποίες θέτουμε ίσες με το μηδέν και από το 2x2 γραμμικό σύστημα εκτιμάμε τις παραμέτρους:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^{n}(x_i - \overline{x})\,(y_i - \overline{y})}{\sum_{i=1}^{n}(x_i - \overline{x})^2}$$

$$\hat{\beta}_0 = \overline{y} - \hat{\beta}_1 \overline{x}$$

Με $\overline{y} = \frac{1}{n}\sum_{i=1}^{n} y_i$ και $\overline{x} = \frac{1}{n}\sum_{i=1}^{n} x_i$

Σημαντικό εργαλείο στην εκτίμηση της ακρίβειας του μοντέλου είναι το τυπικό σφάλμα(S.E. Mean) το οποίο ορίζεται ως η τυπική απόκλιση του μέσου όρου του δείγματος:

$$SE = \frac{\sigma}{\sqrt{n}}$$

όπου σ η τυπική απόκλιση:

$$\sigma^2 = \frac{\sum_{i=1}^{n}(x_i - \overline{x})^2}{n}$$

Αυτό που εξετάζουμε στα γραμμικά μοντέλα, είναι η ύπαρξη ή μη γραμμικής σχέσης μεταξύ της εξαρτημένης (dependent) μεταβλητής Y και της ανεξάρτητης (independent) μεταβλητής X.
Οπότε ορίζουμε ως μηδενική υπόθεση: «Δεν υπάρχει γραμμική σχέση μεταξύ Y και X», δηλαδή:

$$\text{H}_0: \hat{\beta}_1 = 0,$$

Έναντι της εναλλακτικής υπόθεσης: «Υπάρχει γραμμική σχέση μεταξύ Y και X», δηλαδή:

$$\text{H}_1: \hat{\beta}_1 \neq 0$$

Ο έλεγχος γίνεται σε (1-α)% επίπεδο σημαντικότητας και η στατιστική συνάρτηση υπό την H_0 είναι:

$$T = \frac{\hat{\beta}_1}{SE(\hat{\beta}_1)}$$

Ακολουθεί την t-κατανομή με n-2 βαθμούς ελευθερίας (T-test). Περιοχή απόρριψης: απορρίπτουμε την H_0 αν $|T| > t_{n=2;a/2}$;

Άλλα σημαντικά εργαλεία μέτρησης της ακρίβειας του μοντέλου είναι:

Το συνολικό άθροισμα των τετραγώνων:

$$TSS = \sum_{i=1}^{n} (y_i - \bar{y})^2$$

Το υπολειμματικό τυπικό σφάλμα:

$$RSE = \sqrt{\frac{1}{n-p-1}RSS}$$

Ο συντελεστής προσδιορισμού:

$$R^2 = \frac{TSS - RSS}{TSS} = 1 - \frac{RSS}{TSS}$$

Ο F-έλεγχος:

$$F = \frac{(TSS - RSS)/p}{RSS/(n-p-1)}$$

Οι τιμές που μπορεί να πάρει ο R^2 είναι από 0 έως 1, όσο πιο κοντά είναι στο 1 τόσο πιο ικανοποιητικά προσαρμόζεται η ευθεία παλινδρόμησης στα δεδομένα. Αντίθετα, όσο πιο μικρός είναι ο RSE τόσο πιο ικανοποιητική είναι η ακρίβεια του μοντέλου.

[37] [38]

2.2.4 Λογιστική Παλινδρόμηση (Logistic Regression)

Στη λογιστική παλινδρόμηση η εξαρτημένη μεταβλητή είναι κατηγορική και δίτιμη. Στη λογιστική παλινδρόμηση εξετάζουμε την πιθανότητα (τα ποσοστά) εμφάνισης των δύο κατηγοριών σε σχέση με τις ανεξάρτητες μεταβλητές-παράγοντες. Επειδή σκοπός είναι να εκτιμηθεί η πιθανότητα εμφάνισης ενός συμβάντος, συνεπάγεται ότι οι τιμές που θα πρέπει να προκύπτουν από το γραμμικό υπόδειγμα περιέχονται στο διάστημα [0,1].

Στη λογιστική παλινδρόμηση χρησιμοποιούμε την λογιστική συνάρτηση:

$$p(X) = \frac{e^{\beta_0+\beta_1 X}}{1 + e^{\beta_0+\beta_1 X}}$$

ή

$$p(X) = \frac{e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_p X_p}}{1 + e^{\beta_0+\beta_1 X_1+\beta_2 X_2+\cdots+\beta_p X_p}}$$

Από την οποία καταλήγουμε στην:

$$\log(\frac{p(X)}{1-p(X)}) = \beta_0 + \beta_1 X$$

ή

$$\log(\frac{p(X)}{1-p(X)}) = \beta_0 + \beta_1 X_1 + \beta_2 X_2 + \cdots + \beta_p X_p$$

Όπου β_0 είναι το ύψος της κλίσης της γραμμής παλινδρόμησης και β_i είναι οι συντελεστές παλινδρόμησης καθένας των οποίων εκφράζει το μέγεθος συνεισφοράς της αντίστοιχης μεταβλητής. Θετική τιμή του συντελεστή δηλώνει ότι η επεξηγηματική μεταβλητή αυξάνει την πιθανότητα της επιτυχημένης έκβασης (να συμβεί δηλαδή το γεγονός), αρνητική τιμή σημαίνει ότι η μεταβλητή μειώνει την πιθανότητα αυτής της έκβασης. Υψηλή τιμή του συντελεστή σημαίνει ότι η ανεξάρτητη μεταβλητή επηρεάζει πολύ ισχυρά την πιθανότητα να συμβεί το γεγονός ή μη, ενώ χαμηλή τιμή δηλώνει μικρή επίδραση της ανεξάρτητης μεταβλητής στην πιθανότητα εμφάνισης της ανάλογης έκβασης.

Οι πιθανότητες που συγκλίνουν υπέρ της εμφάνισης ενός γεγονότος ή πρόθεσης εκφράζονται ως λόγος ζεύγους ακέραιων τιμών (odds) όπου ο αριθμητής προσδιορίζει την πιθανότητα που έχει το προσδοκώμενο γεγονός να συμβεί και ο παρονομαστής την πιθανότητα να μη συμβεί. Έτσι, αν p είναι η πιθανότητα να εμφανιστεί το γεγονός και 1-p η πιθανότητα να μη συμβεί τότε ο λόγος των πιθανοτήτων θα είναι p/(1-p).

Στόχος μας είναι να βρούμε τις τιμές β_0, β_i έτσι ώστε η εφαρμογή αυτών των εκτιμήσεων στο μοντέλο για την p(X), να δίνει έναν αριθμό κοντά στο 1 για όλους τους παράγοντες πρόβλεψης που έχουν απόκριση Y = 1 και έναν αριθμό κοντά στο 0 για όλους τους παράγοντες πρόβλεψης που έχουν απόκριση Y = 0. Στο μοντέλο της λογιστικής παλινδρόμησης η εκτίμηση των συντελεστών πραγματοποιείται με τη μέθοδο μέγιστης πιθανοφάνειας (maximum likelihood method).

Η συνάρτηση ορίζεται ως εξής:

$$L = \prod_{i=1}^{n} p(x_i \theta)$$

ή

$$L = \sum_{i=1}^{n} \log_e p(x_i \theta)$$

όπου θ είναι μια παράμετρος της μεταβλητής η οποία μπορεί να μεταβάλλεται ελεύθερα.

Η προβλεπόμενη τιμή για κάθε παρατήρηση θα ισούται με:

$$\hat{l} = \frac{1}{n} \log_e L$$

[39] [38] [37]

2.2.5 Μπαϋεσιανοί Κατηγοριοποιητές

Τα Μπαϋεσιανά Δίκτυα (Bayesian Networks) είναι ισχυρά εργαλεία για αναπαράσταση σύνθετων σχέσεων μεταξύ μεταβλητών και για εξαγωγή συμπερασμάτων σε συνθήκες αβεβαιότητας. Ανήκουν στην κατηγορία των γραφικών πιθανοτικών μοντέλων, τα οποία αναπαριστούν σχέσεις με μορφή γράφων. Κάθε κόμβος του γράφου συμβολίζει μια στοχαστική μεταβλητή και κάθε βέλος συμβολίζει μια σχέση εξάρτησης ανάμεσα σε δύο μεταβλητές. Τα Μπαϋεσιανά Δίκτυα αρχικά δεν θεωρήθηκαν εργαλεία κατηγοριοποίησης, αργότερα όμως ανακαλύφθηκε ότι οι Αφελείς Μπαϋεσιανοί κατηγοριοποιητές (Naive Bayesian Classifiers), μια απλουστευμένη εκδοχή των Μπαϋεσιανών Δικτύων, έχουν αυξημένες δυνατότητες κατηγοριοποίησης, συγκρίσιμες με αυτές των Νευρωνικών Δικτύων και των Δένδρων Αποφάσεων. Σήμερα τα Μπαϋεσιανά Δίκτυα αποτελούν μια καταξιωμένη μέθοδο Εξόρυξης Δεδομένων, λόγω της στιβαρής θεωρητικής τους θεμελίωσης, της ικανότητας τους να καταγράφουν περίπλοκες σχέσεις αλληλεξάρτησης, του συμβολικού φορμαλισμού τους και της δυνατότητας τους να εφαρμόζονται σε προβλήματα κατηγοριοποίησης(Heckerman, 1997).

Τα Μπαϋεσιανά Δίκτυα έλκουν το θεωρητικό τους υπόβαθρο από τη στατιστική και πιο συγκεκριμένα από το θεώρημα του Bayes, που υπολογίζει την υπό συνθήκη πιθανότητα P(H|X), δηλαδή την πιθανότητα να επαληθευτεί η υπόθεση H με δεδομένο ότι ισχύει το γεγονός X. Σύμφωνα με το θεώρημα του Bayes, η πιθανότητα P(H|X)δίνεται από την εξίσωση:

$$P(H|X) = \frac{P(H) * P(X|H)}{P(X)}$$

Όπου P(H) είναι η εκ των προτέρων πιθανότητα να ισχύει η υπόθεση H, P(X) είναι η εκ των προτέρων πιθανότητα να συμβεί το γεγονός X και P(X|H) είναι η πιθανότητα να συμβεί το γεγονός X με δεδομένο ότι ισχύει η υπόθεση H.

2.2.5.1 Αφελείς Μπαϋεσιανοί Κατηγοριοποιητές(Naïve Bayes)

Ο Αφελής Μπαϋεσιανός κατηγοριοποιητής αποτελεί ευθεία εφαρμογή του θεωρήματος Bayes. Υποθέτουμε ότι X είναι μια παρατήρηση του συνόλου δεδομένων και H είναι η υπόθεση ότι παρατήρηση αυτή ανήκει στην κλάση C_i. Πιο συγκεκριμένα, το X θεωρείται ως ένα άνυσμα n τιμών X=(x1,x2,...,xn). Υποθέτουμε ότι υπάρχουν m κλάσεις C_1, C_2, ..., C_m. Σύμφωνα με το θεώρημα του Bayes, η πιθανότητα να ανήκει η παρατήρηση X στην κλάση C_i υπολογίζεται από την εξίσωση:

$$P(C_i|X) = \frac{P(X|C_i) * P(C_i)}{P(X)}$$

Για να προβλέψει την κλάση μιας άγνωστης παρατήρησης, ο Αφελής Μπαϋεσιανός κατηγοριοποιητής υπολογίζει τις πιθανότητες για την κάθε κλάση και εκχωρεί την παρατήρηση στην κλάση με τη μεγαλύτερη πιθανότητα. Εφόσον το P(X) είναι ίδιο για όλες τις κλάσεις και το $P(C_i)$ μπορεί εύκολα να υπολογιστεί (ως το πλήθος των παρατηρήσεων που ανήκουν στην κλάση C_i προς το πλήθος όλων των παρατηρήσεων), το ζητούμενο είναι ο υπολογισμός του $P(X|C_i)$. Ο υπολογισμός του $P(X|C_i)$ μπορεί να αποδειχθεί ιδιαίτερα περίπλοκος εάν θεωρηθεί ότι υπάρχει σχέση εξάρτησης μεταξύ των διαστάσεων του ανύσματος X, δηλαδή μεταξύ των μεταβλητών εισόδου. Αντιθέτως, αν θεωρηθεί ότι, δοθείσης της κλάσης, οι μεταβλητές εισόδου είναι μεταξύ τους ανεξάρτητες, τότε ο υπολογισμός του $P(X|C_i)$ απλοποιείται και δίνεται από την εξίσωση:

$$P(X|C_i) = \prod_{k=1}^{n} P(x_k|C_i)$$

όπου x_k είναι η τιμή της διάστασης k του ανύσματος X.

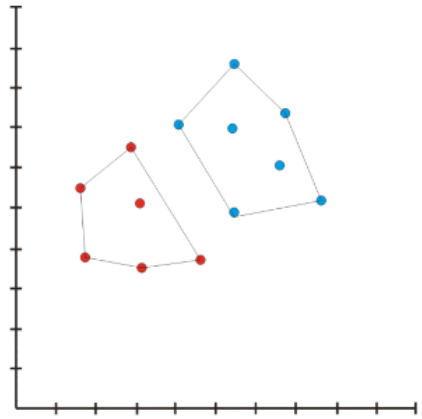
Ο κατηγοριοποιητής, αφού υπολογίσει τις πιθανότητες $P(C_i|X)$ για όλες τις κλάσεις C_i, εκχωρεί την παρατήρηση στην κλάση με τη μεγαλύτερη πιθανότητα. Εάν ισχύει η υπόθεση ότι δεδομένης της κλάσης είναι ανεξάρτητες οι μεταβλητές εισόδου, ο Αφελής Μπαϋεσιανός κατηγοριοποιητής επιτυγχάνει τους υψηλότερους ρυθμούς ακρίβειας. Ωστόσο, στην πράξη τις περισσότερες φορές η υπόθεση αυτή δεν ισχύει.
[40]

2.2.6 Μηχανές Διανυσμάτων Υποστήριξη (Support Vector Machines)

Οι Μηχανές Διανυσμάτων Υποστήριξης (Support Vector Machines(SVM)) προτάθηκαν από τον Vapnik(1995) και γρήγορα γνώρισαν μεγάλη διάδοση λόγω της στιβαρής θεωρητικής θεμελίωσης τους και των υψηλών επιδόσεων τους. Οι Μηχανές Διανυσμάτων Υποστήριξης αποτέλεσαν αντικείμενο ενδιαφέροντος πολλών ερευνητών και εφαρμόστηκαν για την ανάπτυξη μοντέλων σε πλήθος προβλημάτων κατηγοριοποίησης. Προορίζονται για προβλήματα δυαδικής ταξινόμησης στην οποία υπάρχουν δύο τάξεις. Οι Μηχανές Διανυσμάτων Υποστήριξης αποτελούν μια γενίκευση ενός απλού ταξινομητή που ονομάζεται ταξινομητής μεγίστου περιθωρίου.

Βασική τους ιδέα είναι η κατασκευή ενός υπερεπιπέδου (hyperplane),το οποίο διαχωρίζει τις κλάσεις και λειτουργεί ως συνάρτηση απόφασης. Οι νέες παρατηρήσεις κατηγοριοποιούνται ανάλογα

με την πλευρά του υπερεπιπέδου στην οποία βρίσκονται. Ας θεωρήσουμε μια απλή περίπτωση όπου η κλάση είναι δυαδική και οι παρατηρήσεις είναι γραμμικά διαχωρίσιμες. Το κυρτό περίβλημα(convex hull) ενός συνόλου σημείων είναι το μικρότερο κυρτό πολύγωνο, το οποίο περικλείει όλα τα σημεία του συνόλου. Οι δύο κλάσεις είναι γραμμικά διαχωρίσιμες, όταν τα κυρτά περιβλήματα τους δεν επικαλύπτονται. Παράδειγμα παρατηρήσεων δυαδικής κλάσης, οι οποίες είναι γραμμικά διαχωρίσιμες, απεικονίζεται στην **Εικόνα 2.2**. Οι παρατηρήσεις συμβολίζονται ως μικροί κύκλοι, ενώ το διαφορετικό χρώμα συμβολίζει τις διαφορετικές κλάσεις. Η μια τιμή κλάσης μπορεί να οριστεί ως θετική και να συμβολιστεί με την τιμή +1, ενώ η άλλη τιμή να οριστεί ως αρνητική και να συμβολιστεί με την τιμή -1.



ΕΙΚΟΝΑ 2.2-ΠΑΡΑΔΕΙΓΜΑ ΠΑΡΑΤΗΡΗΣΕΩΝ ΔΥΑΔΙΚΗΣ ΚΛΑΣΗΣ, ΟΙ ΟΠΟΙΕΣ ΕΙΝΑΙ ΓΡΑΜΜΙΚΑ ΔΙΑΧΩΡΙΣΙΜΕΣ

Το γενικό υπερεπιπέδου διαχωρισμού ορίζεται από την εξίσωση:

$$w^T x + b = 0$$

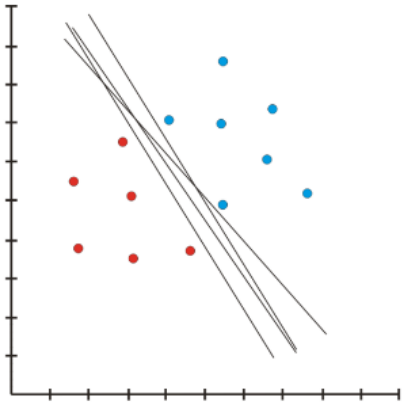
όπου w είναι ένα διάνυσμα βαρών, το οποίο είναι κάθετο στο επίπεδο και ορίζει τον προσανατολισμό του και b είναι το κατώφλι. Η μεταβολή της τιμής του b έχει σαν αποτέλεσμα την παράλληλη μετατόπιση του επιπέδου. Για μια παρατήρηση x_1 θετικής κλάσης ισχύει ότι:

$$w^T x_1 + b > 0$$

ενώ για μια παρατήρηση x_2 αρνητικής κλάσης ισχύει ότι:

$$w^T x_2 + b < 0$$

Πλέον το πρόβλημα της κατηγοριοποίησης ανάγεται σε πρόβλημα καθορισμού του υπερεπιπέδου διαχωρισμού. Όπως φαίνεται στην **Εικόνα 2.3** υπάρχουν πολλά υπερεπίπεδα, τα οποία θα μπορούσαν να χρησιμοποιηθούν, και το ερώτημα είναι ποιο από αυτά είναι το καλύτερο.

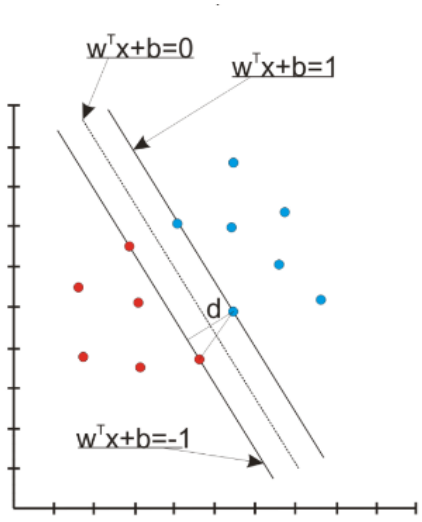


ΕΙΚΟΝΑ 2.3-ΚΑΘΟΡΙΣΜΟΣ ΤΟΥ ΥΠΕΡΕΠΙΠΕΔΟΥ ΔΙΑΧΩΡΙΣΜΟΥ

Για τον υπολογισμό του βέλτιστου επιπέδου εισάγεται η έννοια του περιθωρίου (margin). Ως περιθώριο ορίζεται η μικρότερη απόσταση ενός σημείου από το υπερεπίπεδο διαχωρισμού. Η κλίμακα του περιθωρίου επηρεάζεται από το διάνυσμα βαρών w. Θεωρούμε τα σημεία x_i, τα οποία είναι πλησιέστερα στο υπερεπίπεδο. Μπορούμε να ρυθμίσουμε τις τιμές των w και b έτσι ώστε η απόσταση των σημείων αυτών από το υπερεπίπεδο να είναι ίση με 1:

$$|(w^T x_i) + b| = 1$$

Θεωρούμε δύο σημεία x_1 και x_2 τα οποία είναι πλησιέστερα στο υπερεπίπεδο, δηλαδή η απόσταση τους από αυτό είναι ίση με 1 και τα οποία βρίσκονται εκατέρωθεν του υπερεπιπέδου, δηλαδή η τιμή κλάσης του ενός είναι +1 και του άλλου -1. Από τα σημεία αυτά μπορούμε να ορίσουμε το περιθώριο ως την απόσταση τους d, μετρημένη κάθετα στο υπερεπίπεδο, όπως φαίνεται στην **Εικόνα 2.4**



ΕΙΚΟΝΑ 2.4-ΥΠΟΛΟΓΙΣΜΟΣ ΠΕΡΙΘΩΡΙΟΥ ΥΠΕΡΕΠΙΠΕΔΟΥ ΔΙΑΧΩΡΙΣΜΟΥ

Το περιθώριο υπολογίζεται σύμφωνα με την:

$$\left(\frac{w}{\|w\|}(x_1 - x_2)\right) = \frac{2}{\|w\|}$$

Το βέλτιστο υπερεπίπεδο διαχωρισμού των κλάσεων είναι αυτό που εξασφαλίζει το μέγιστο περιθώριο. Τα σημεία, τα οποία βρίσκονται στο όριο του περιθωρίου, ονομάζονται διανύσματα υποστήριξης. Προφανώς η κάθετη απόσταση από το υπερεπίπεδο των σημείων x1και x2είναι ίση με το μισό του περιθωρίου, δηλαδή $1/\|w\|$. Το πρόβλημα μετατρέπεται σε ένα πρόβλημα βελτιστοποίησης. Η ποσότητα $1/\|w\|$ πρέπει να μεγιστοποιηθεί για κάθε σημείο, με τον περιορισμό ότι η απόσταση του πλησιέστερου σημείου θα είναι ίση με 1. Για n σημεία xi το παραπάνω πρόβλημα διατυπώνεται ως εξής:

$$Maximize \frac{1}{\|w\|}$$

με τον περιορισμό ότι:

$$\min_{i=1,2,..,n} |(w^T x_i) + b| = 1$$

Με δεδομένο ότι η κλάση y_i μιας παρατήρησης x_i μπορεί να πάρει τιμές +1 ή -1, καθώς και ότι το $w^T x_i+b$ θα έχει τιμή >=1 για παρατηρήσεις θετικής κλάσης και <=-1 για παρατηρήσεις αρνητική κλάσης, προκύπτει ότι το γινόμενο του ($w^T x_i+b$) με την τιμή της κλάσης θα δίνει αποτέλεσμα μεγαλύτερο ή ίσο του 1.

$$y_i * (w^T x_i + b) \geq 1$$

Το πρόβλημα επαναδιατυπώνεται ως εξής:

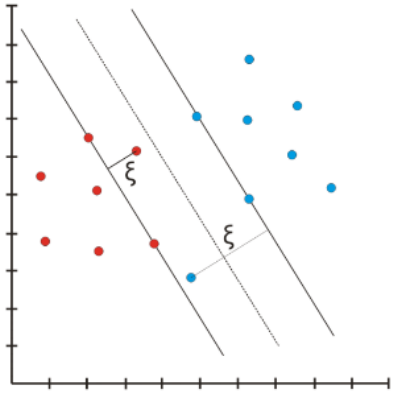
$$Minimize \frac{1}{2}\|w^2\|$$

Το πρόβλημα μπορεί να λυθεί με τη χρήση του τετραγωνικού προγραμματισμού. Σε προβλήματα του πραγματικού κόσμου μπορεί να μην είναι όλες οι παρατηρήσεις γραμμικά διαχωρίσιμες. Για να ξεπεράσει το πρόβλημα του απόλυτου γραμμικού διαχωρισμού, ο Vapnik εισήγαγε τις μεταβλητές χαλαρότητας ξ_i. Με τη συμμετοχή των μεταβλητών χαλαρότητας προκύπτει η σχέση:

$$y_i * (w^T x_i + b) \geq 1 - \xi_i$$

με ξ_i >=0

Αν για ένα σημείο x_i η μεταβλητή ξ_i είναι μεγαλύτερη από 1, τότε το σημείο κατηγοριοποιείται εσφαλμένα, όπως φαίνεται στην **Εικόνα 2.5**

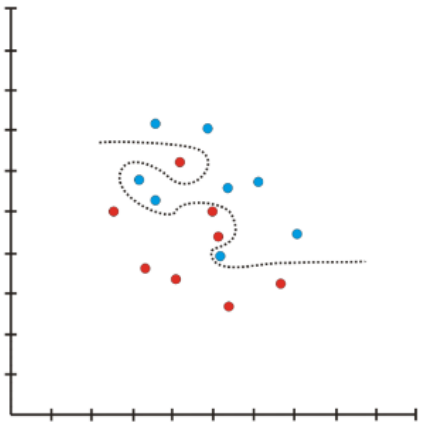


ΕΙΚΟΝΑ 2.5-ΣΦΑΛΜΑΤΑ ΚΑΤΗΓΟΡΙΟΠΟΙΗΣΗΣ

Το άθροισμα των ξ_i μπορεί να θεωρηθεί το πλήθος των σφαλμάτων κατηγοριοποίησης. Έτσι προκύπτει η σχέση:

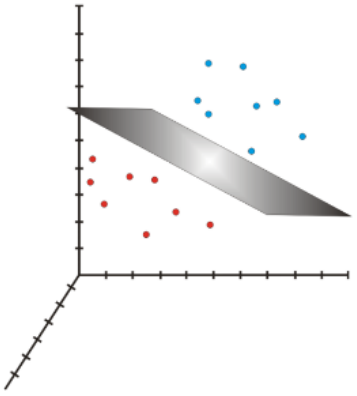
$$Minimize \frac{1}{2}\|w^2\| + C\sum_{i=1}^{n}\xi_i$$

Η σταθερά C είναι μια παράμετρος, που ορίζει το ισοζύγιο μεταξύ πολυπλοκότητας και εμπειρικού σφάλματος. Οι περιπτώσεις δυνατότητας γραμμικού διαχωρισμού των κλάσεων είναι μάλλον σπάνιες σε πραγματικά προβλήματα. Εάν όμως τα σημεία x_i προβληθούν με μία μη γραμμική διανυσματική συνάρτηση $\phi(x_i)$ σε έναν χώρο περισσότερων διαστάσεων, τότε είναι πιθανό οι απεικονίσεις τους στον νέο χώρο να είναι γραμμικώς διαχωρίσιμες. Στην **Εικόνα 2.6** απεικονίζονται τα σημεία στον αρχικό δισδιάστατο χώρο. Τα σημεία δεν είναι γραμμικώς διαχωρίσιμα.



ΕΙΚΟΝΑ 2.6-ΑΠΕΙΚΟΝΙΣΗ ΜΗ ΓΡΑΜΜΙΚΩΣ ΔΙΑΧΩΡΙΣΙΜΩΝ ΣΗΜΕΙΩΝ ΣΤΟΝ ΔΙΣΔΙΑΣΤΑΤΟ ΧΩΡΟ

Στην **Εικόνα 2.8** τα σημεία προβάλλονται σε έναν τρισδιάστατο χώρο, και εκεί είναι γραμμικώς διαχωρίσιμα. Εφόσον στον χώρο αυτόν ισχύει ο γραμμικός διαχωρισμός, μπορεί να εφαρμοστεί η μέθοδος των διανυσμάτων υποστήριξης που παρουσιάστηκε προηγουμένως.



ΕΙΚΟΝΑ 2.7- ΑΠΕΙΚΟΝΙΣΗ ΓΡΑΜΜΙΚΩΣ ΔΙΑΧΩΡΙΣΙΜΩΝ ΣΗΜΕΙΩΝ ΣΤΟΝ ΤΡΙΣΔΙΑΣΤΑΤΟ ΧΩΡΟ

Η συνάρτηση απόφασης επαναδιατυπώνεται ως εξής:

$$f(x) = w^T \varphi(x) + b$$

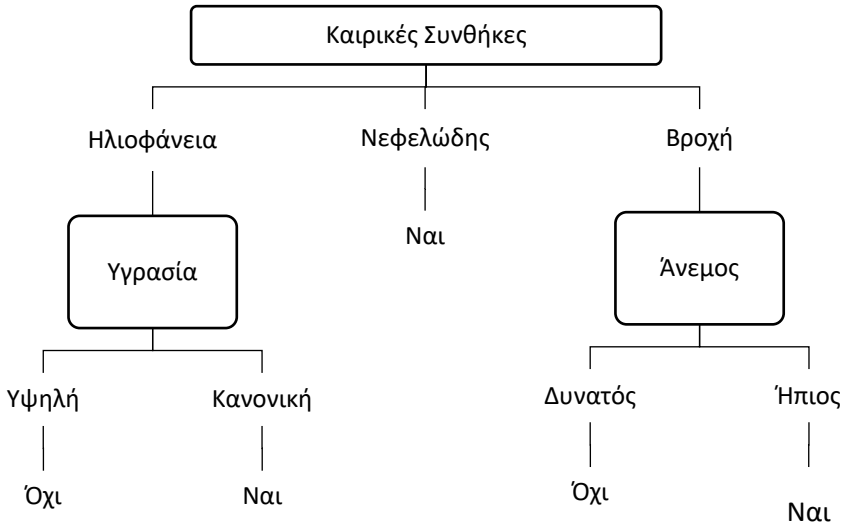
Ο προσδιορισμός της συνάρτησης φ μπορεί να είναι εξαιρετικά δύσκολος και ο χώρος προβολής μπορεί να έχει πάρα πολλές διαστάσεις. Όμως για τον υπολογισμό της συνάρτησης απόφασης f, απαιτείται μόνο ο ορισμός του εσωτερικού γινόμενου $\phi(x_i)*\phi(x_j)$. Ορίζουμε μια συνάρτηση K(xi, x_j) η οποία υπολογίζει το εσωτερικό γινόμενο των απεικονίσεων $\phi(x_i)$ και $\phi(x_j)$. Η συνάρτηση K καλείται συνάρτηση πυρήνα(kernel function)

$$K(x_i, x_j) = \varphi(x_i) * \varphi(x_j)$$

Διάφορες συναρτήσεις μπορούν να χρησιμοποιηθούν ως συναρτήσεις πυρήνα. Σε αυτές περιλαμβάνονται η Συνάρτηση Ακτινωτής Βάσης (Radial Base Function – RBF), η Σιγμοειδής, η πολυωνυμική και η αντίστροφη πολυτετραγωνική συνάρτηση. Ο πυρήνας καθορίζει τη μορφή του υπερεπιπέδου διαχωρισμού και συνεπώς επηρεάζει την απόδοση του κατηγοριοποιητή. Η επιλογή της καλύτερης συνάρτησης πυρήνα είναι θέμα το οποίο διερευνάται (Steinwart, 2003)
[40] [37]

2.2.7 Τα Δέντρα Αποφάσεων(Decision Trees)

Τα Δένδρα Αποφάσεων είναι μια από τις βασικότερες και πιο δημοφιλείς μεθόδους κατηγοριοποίησης. Βασική λογική της κατασκευής τους είναι η διαδοχική διάσπαση του συνόλου των παρατηρήσεων σε υποσύνολα. Κριτήριο για τη διάσπαση είναι οι τιμές των μεταβλητών. Η διαδικασία των διαδοχικών διασπάσεων αναπαρίσταται με μια ανεστραμμένη δενδρική δομή. Στην κορυφή βρίσκεται ο κόμβος-ρίζα του δένδρου. Σε κατώτερα επίπεδα βρίσκονται επιπλέον κόμβοι, οι οποίοι συνδέονται με ακμές με άλλα στοιχεία του δένδρου. Στο κατώτερο επίπεδο κάθε κλάδου βρίσκονται τα φύλλα του δένδρου. Ο κόμβος - ρίζα έχει μόνο εξερχόμενες ακμές που τον συνδέουν με στοιχεία του κατώτερου επιπέδου. Οι υπόλοιποι κόμβοι έχουν εισερχόμενες ακμές που τους συνδέουν με τους κόμβους του ανώτερου επιπέδου και εξερχόμενες ακμές που τους συνδέουν με στοιχεία του κατώτερου επιπέδου. Τέλος, τα φύλα έχουν μόνο εισερχόμενες ακμές, οι οποίες τα συνδέουν με τους κόμβους του ανώτερου επιπέδου. Κάθε κόμβος αντιπροσωπεύει έναν έλεγχο στα δεδομένα και αντίστοιχη διάσπαση τους σε δύο ή περισσότερα υποσύνολα, ανάλογα με το αποτέλεσμα του ελέγχου. Η συνηθέστερη εκδοχή είναι ο έλεγχος να περιλαμβάνει μία μόνο μεταβλητή, έχουν προταθεί ωστόσο αλγόριθμοι όπου σε έναν κόμβο ελέγχονται περισσότερες μεταβλητές. Κάθε ακμή αντιπροσωπεύει ένα αποτέλεσμα του ελέγχου και το αντίστοιχο υποσύνολο των δεδομένων. Τέλος, κάθε φύλο αντιπροσωπεύει μια απόφαση κατηγοριοποίησης. [40]



ΕΙΚΟΝΑ 2.8-ΑΠΕΙΚΟΝΙΣΗ ΔΕΝΤΡΟΥ ΑΠΟΦΑΣΗΣ

Στην Εικόνα 2.8 απεικονίζεται ένα Δένδρο Απόφασης για την απόφαση για έναν περίπατο. Ο κόμβος-ρίζα αναφέρεται στο σύνολο των δεδομένων. Στο επίπεδο ο Καιρός χωρίζεται σε τρία υποσύνολα ανάλογα με τις καιρικές συνθήκες. Στο πρώτο υποσύνολο ανήκει η ηλιοφάνεια, στο δεύτερο ο νεφελώδης καιρός, ενώ στο τρίτο υποσύνολο ανήκει ο άνεμος. Τα τρία υποσύνολα συμβολίζονται με αντίστοιχους κλάδους. Ο πρώτος κλάδος, ο οποίος αντιστοιχεί στην ηλιοφάνεια, καταλήγει σε έναν εσωτερικό κόμβο. Στον κόμβο αυτόν γίνεται ένας δεύτερος έλεγχος που αφορά την υγρασία. Αν η υγρασία είναι υψηλή τότε η απόφαση για περίπατο είναι όχι, αν η υγρασία είναι κανονική τότε η απόφαση είναι ναι. Ο δεύτερος κλάδος καταλήγει ένα φύλο, δηλαδή σε μια απόφαση κατηγοριοποίησης. Η απόφαση είναι θετική, και αυτό σημαίνει ότι γι' αυτήν την κατηγορία η απόφαση για περίπατο είναι θετική. Με τον ίδιο τρόπο ελέγχεται η περίπτωση της βροχής στον τρίτο κλάδο της ρίζας για το αν η απόφαση για περίπατο θα είναι θετική ή όχι.

Το μοντέλο κατασκευάζεται από έναν αλγόριθμο με επεξεργασία ενός συνόλου δεδομένων εκπαίδευσης. Το μοντέλο, αφού κατασκευαστεί, μπορεί να χρησιμοποιηθεί για την κατηγοριοποίηση νέων παρατηρήσεων. Για κάθε νέα παρατήρηση πραγματοποιούνται έλεγχοι τιμών των μεταβλητών της, σύμφωνα με τους κόμβους του δένδρου, και ακολουθείται μια διαδρομή από τη ρίζα μέχρι κάποιο φύλο, όπου λαμβάνεται και η απόφαση κατηγοριοποίησης. Στο παράδειγμα της Εικόνας ο καιρός θα ελεγχθεί πρώτα ως προς την ηλιοφάνεια, τα σύννεφα και τη βροχή. Εάν έχει ηλιοφάνεια, θα ελεγχθεί η υγρασία. Αν η υγρασία είναι χαμηλή τότε η απόφαση για περίπατο θα είναι θετική.

2.2.8 Τυχαία Δάση(Random Forests)

Ένα τυχαίο δάσος είναι ένας ταξινομητής που αποτελείται από μια συλλογή ταξινομητών δένδρων $\{h(x,\Theta_k), k = 1, \ldots\}$ όπου $\{\Theta_k\}$ είναι ανεξάρτητα κατανεμημένα πανομοιότυπα τυχαία διανύσματα και κάθε δέντρο δίνει μια ψήφο μονάδας για την πιο δημοφιλή τάξη στην είσοδο x. [41]

Ένα τυχαίο δάσος είναι ουσιαστικά μια συλλογή δέντρων αποφάσεων, όπου κάθε δέντρο είναι ελαφρώς διαφορετικό από τα άλλα. Η ιδέα πίσω από τα τυχαία δάση είναι ότι κάθε δέντρο μπορεί να κάνει μια ακριβή πρόβλεψη, αλλά υπάρχει πιθανότητα να υπερπροσαρμοστεί σε ένα κομμάτι των δεδομένων. Δημιουργώντας πολλά δέντρα, τα οποία λειτουργούν αποδοτικά και υπερπροσαρμόζονται με διαφορετικούς τρόπους, μπορούμε να μειώσουμε την ποσότητα της υπερπροσαρμογής στον μέσο όρο των αποτελεσμάτων. Για την εφαρμογή αυτής της μεθόδου θα πρέπει να φτιάξουμε πολλά δέντρα. Κάθε δέντρο πρέπει να κάνει μια εργασία για την πρόβλεψη του στόχου και θα πρέπει επίσης να είναι διαφορετική από τα υπόλοιπα δέντρα. Τα τυχαία δάση δημιουργούν την ακολουθία των μοντέλων τους

εκπαιδεύοντάς τα σε υποσύνολα των δεδομένων. Τα υποσύνολα επιλέγονται τυχαία από τα δεδομένα εκπαίδευσης. [33]

2.3 Εξόρυξη Γνώσης από Κείμενο(Text Mining)

Η ταχεία πρόοδος στην απόκτηση ψηφιακών δεδομένων οδήγησε στην ταχέως αναπτυσσόμενη ποσότητα αποθηκευμένων δεδομένων σε βάσεις δεδομένων, αποθήκες δεδομένων ή άλλα είδη αποθετηρίων δεδομένων. Αν και πολύτιμη η πληροφορία που μπορεί να κρύβεται πίσω από τα δεδομένα, ο τεράστιος όγκος τους καθιστά δύσκολο, αν όχι αδύνατο, να μπορούν οι άνθρωποι τα την εξάγουν χωρίς ισχυρά εργαλεία. Προκειμένου να λυθεί αυτό το πρόβλημα στα τέλη της δεκαετίας του 1980, προέκυψε μια νέα ορολογία που ονομάζεται Εξόρυξη Δεδομένων(Data Mining), η οποία αφορά την εξαγωγή γνώσης από τεράστιους όγκους δεδομένων. Λόγω του διεπιστημονικού χαρακτήρα της, η εξόρυξη δεδομένων σχετίζεται με πολλούς επιστημονικούς κλάδους όπως βάσεις δεδομένων, μηχανική μάθηση, στατιστική, ανάκτηση πληροφοριών, οπτικοποίηση δεδομένων.

Η Εξόρυξη Κειμένου(Text Mining) είναι ένα πεδίο εφαρμογής της εξόρυξης δεδομένων και ένας τομέας επιστημονικής έρευνας που βρίσκεται υπό σημαντική ανάπτυξη. Ο στόχος της εξόρυξης δεδομένων είναι η εκμετάλλευση πληροφοριών που περιέχονται σε έγγραφα κειμένου με διάφορους τρόπους, όπως ανακάλυψη προτύπων και τάσεων στα δεδομένα, συσχετίσεων μεταξύ οντοτήτων, κανόνων πρόβλεψης κ.λπ. [42] . Ένας ορισμός για την εξόρυξη κειμένων είναι: «ένας άλλος τρόπος για να δείτε την εξόρυξη δεδομένων κειμένου είναι ως διαδικασία διερευνητικής ανάλυσης δεδομένων που οδηγεί σε έως τώρα άγνωστες πληροφορίες ή απαντήσεις για ερωτήσεις για τις οποίες η απάντηση δεν είναι γνωστή επί του παρόντος». [43]

Η εξόρυξη κειμένων χρησιμοποιεί τεχνικές από καθιερωμένα επιστημονικά πεδία όπως εξόρυξη δεδομένων, μηχανική εκμάθηση, ανάκτηση πληροφοριών, επεξεργασία φυσικής γλώσσας, στατιστική και διαχείριση γνώσεων, ώστε να μπορέσουν οι άνθρωποι να κερδίσουν διορατικότητα, κατανόηση και ερμηνεία μεγάλων ποσοτήτων δεδομένων. Συνήθως, η εξόρυξη κειμένων περιλαμβάνει προεπεξεργασία εγγράφων, αποθήκευση και ευρετηρίαση ενδιάμεσων αποτελεσμάτων, ανάλυση και οπτικοποίηση των αποτελεσμάτων.

2.3.1 Διαδικασία Εξόρυξης Γνώσης από Κείμενο

Για την εξόρυξη ενός κειμένου, πρέπει πρώτα να το επεξεργαστούμε σε μια μορφή που οι διαδικασίες εξόρυξης δεδομένων μπορεί να χρησιμοποιηθεί.

2.3.1.1 Συλλογή Δεδομένων

Το πρώτο βήμα στην εξόρυξη κειμένου είναι η συλλογή των δεδομένων(εγγράφων). Σε πολλά σενάρια εξόρυξης κειμένων, τα σχετικά έγγραφα ενδέχεται να έχουν ήδη δοθεί ή να είναι εύκολο να ληφθούν. Τότε το κύριο ζήτημα είναι ο καθαρισμός των δεδομένων και η διασφάλιση της ποιότητας τους. Άλλες φορές τα έγγραφα μπορούν να ληφθούν από αποθήκες εγγράφων ή βάσεις δεδομένων. Σε αυτή τη περίπτωση είναι λογικό να αναμένεται ότι ο καθαρισμός δεδομένων έγινε πριν από την κατάθεση και μπορούμε να είμαστε σίγουροι για την ποιότητα των εγγράφων.

Σε ορισμένες εφαρμογές, μπορεί να χρειαστεί να αναπτυχθεί μια διαδικασία συλλογής δεδομένων. Για παράδειγμα, για μια εφαρμογή Ιστού που περιλαμβάνει έναν αριθμό αυτόνομων τοποθεσιών Web, μπορεί κάποιος αναπτύξει ένα εργαλείο λογισμικού, όπως ένα πρόγραμμα ανίχνευσης Ιστού που να συλλέγει τα έγγραφα. Σε άλλες εφαρμογές, μπορεί να αναπτυχθεί μια διαδικασία καταγραφής συνδεδεμένη σε μια ροή δεδομένων εισόδου για μια χρονική διάρκεια. Για παράδειγμα, μια εφαρμογή ελέγχου ηλεκτρονικού ταχυδρομείου που να καταγράφει όλα τα εισερχόμενα και εξερχόμενα μηνύματα σε διακομιστή αλληλογραφίας για μια χρονική περίοδο.

2.3.1.2 Αφαίρεση δομής του κειμένου

Μόλις τα έγγραφα συλλεχθούν το πιο πιθανό είναι να βρίσκονται σε μια ποικιλία μορφών ανάλογα με τον τρόπο δημιουργίας τους. Ορισμένα έγγραφα μπορεί να έχουν δημιουργηθεί από έναν επεξεργαστή κειμένου με τη δικιά του μορφή, άλλα μπορεί να έχουν δημιουργηθεί χρησιμοποιώντας ένα απλό πρόγραμμα επεξεργασίας κειμένου και να έχουν αποθηκευτεί ως κείμενο ASCII και ορισμένα μπορεί να έχουν σαρωθεί και αποθηκευτεί ως εικόνες. Για να μπορέσουν επεξεργαστούν όλα τα έγγραφα θα ήταν χρήσιμο να τα μετατραπούν σε μια τυπική μορφή. Αυτή η διαδικασία ονομάζεται τυποποίηση εγγράφου. Με λίγα λόγια τα κείμενα επεξεργάζονται με τέτοιο τρόπο ώστε να αφαιρούνται τα δομικά τους χαρακτηριστικά.

2.3.1.3 Κατακερματισμός(Tokenization)

Το πρώτο βήμα στην επεξεργασία του κειμένου είναι ο διαχωρισμός της ροής των χαρακτήρων σε λέξεις ή αλλιώς σε κέρματα(tokens). Χωρίς προσδιορισμό των κερμάτων(tokens) η

εξαγωγή πληροφοριών υψηλού επιπέδου από το έγγραφο είναι δύσκολη. Κάθε κέρμα αναφέρεται σε έναν τύπο, έτσι ο αριθμός των κερμάτων είναι υψηλότερος από τον αριθμό των τύπων.

Ο διαχωρισμός μιας ροής χαρακτήρων σε κέρματα για ένα πρόγραμμα υπολογιστή μπορεί να είναι περίπλοκη διαδικασία. Ο λόγος είναι ότι ορισμένοι χαρακτήρες μερικές φορές είναι οριοθέτες συμβόλων και μερικές φορές όχι, ανάλογα με την εφαρμογή. Υποθέτουμε πάντα ότι οι χαρακτήρες “space”, “tab”, “newline” είναι οριοθέτες και δεν υπολογίζονται ως κέρματα. Συχνά ονομάζονται «λευκά κενά»(white spaces). Οι χαρακτήρες () <> !? " είναι πάντα οριοθέτες και μπορεί επίσης να είναι διακριτικά. Οι χαρακτήρες . ,: - «μπορεί να είναι ή όχι οριοθέτες, ανάλογα με το περιβάλλον τους. Μια τελεία, κόμμα ή άνω και κάτω τελεία μεταξύ αριθμών δεν θεωρείται κανονικά οριοθέτης αλλά μάλλον μέρος ενός αριθμού. Οποιοδήποτε άλλο κόμμα ή άνω και κάτω τελεία είναι οριοθέτης αλλά μπορεί να είναι και ένα σύμβολο. Μια τελεία μπορεί να είναι μέρος μιας συντομογραφίας ή το τέλος μια πρότασης. Η απόστροφος έχει επίσης πολλές χρήσεις μπορεί να αποτελεί μέρος του τρέχοντος κέρματος (D'angelo) ή μπορεί να υποδηλώνει μια κτητική (Tess’). Η παύλα είναι τερματιστής κέρμα, σύμβολο αφαίρεσης ή ένα διαχωριστικό (555-1212 ως αριθμός τηλεφώνου).

Για το καλύτερο δυνατό αποτέλεσμα, θα πρέπει πάντα ο κατακερματιστής(tokenizer) να προσαρμόζεται για το διαθέσιμο κείμενο και να λαμβάνεται υπόψιν ότι διαδικασία διαμόρφωσης εξαρτάται από τη γλώσσα. Οι γενικές αρχές ισχύουν για όλες τις γλώσσες αλλά οι λεπτομέρειες διαφέρουν από γλώσσα σε γλώσσα.

2.3.1.4 Λημματοποίηση(Lemmatization)

Μόλις μια ροή χαρακτήρων τμηματοποιηθεί σε μια ακολουθία διακριτικών, το επόμενο βήμα συνήθως είναι η μετατροπή του κάθε κέρματος μια τυπική φόρμα, μια διαδικασία που ονομάζεται ανακοπή(stemming) ή λημματοποίηση(lemmatization). Η αναγκαιότητα του βήματος αυτού εξαρτάται από την περίσταση.

Μια μέθοδος λημματοποίησης είναι η διαδικασία «inflectional stemming» μια διαδικασία αντίστοιχη της μορφολογικής ανάλυσης. Αυτή η διαδικασία περιορίζεται στην κανονικοποίηση γραμματικών παραλλαγών όπως ενικός / πληθυντικός και το παρόν / παρελθόν. Παρόλο που αυτή η διαδικασία δεν αναμένεται να είναι τέλεια αναμένεται εντοπίσει σωστά αρκετά σημαντικό αριθμό στελεχών. Μια άλλη μέθοδος είναι η αποκοπή στη ρίζα. Σκοπός είναι να μετατραπούν τα κέρματα σε μια ριζική μορφή χωρίς παραμορφωτικά ή παράγωγα προθέματα και επιθήματα.

2.3.1.5 Διανυσματικοποίηση

Στη διαδικασία κατηγοριοποίησης εγγράφων το βασικό χαρακτηριστικό(feature) τους είναι τα κέρματα ή αλλιώς οι λέξεις που περιέχουν. Εύκολα μπορούμε να καταλήξουμε στο συμπέρασμα ότι για την περιγραφή ενός κειμένου μπορούμε να χρησιμοποιήσουμε τη συχνότητα εμφάνισης των κερμάτων ή λέξεων σε αυτό.

Η συλλογή όλου του συνόλου των χαρακτηριστικών συνήθως ονομάζεται συνήθως λεξικό. Τα κέρματα ή λέξεις του λεξικού αποτελούν τη βάση για τη δημιουργία ενός υπολογιστικού φύλλου με αριθμητικά δεδομένα που αντιστοιχούν στη συλλογή εγγράφων. Κάθε σειρά είναι ένα έγγραφο και κάθε στήλη αντιπροσωπεύει ένα χαρακτηριστικό(feature). Έτσι, ένα κελί στο υπολογιστικό φύλλο είναι μια μέτρηση ενός χαρακτηριστικού(που αντιστοιχεί σε μια στήλη) για ένα έγγραφο (που αντιστοιχεί σε μια σειρά). Στο πιο βασικό μοντέλο τέτοιων δεδομένων, ελέγχουμε απλώς την παρουσία ή την απουσία των λέξεων. Στα κελιά του υπολογιστικού φύλλου αντιστοιχούν οι αριθμοί 0 ή 1 που υποδηλώνουν την απουσία ή την παρουσία της λέξης που αντιστοιχεί στη στήλη του κελιού στο έγγραφο που αντιστοιχεί στη γραμμή του κελιού.

Σε κάποιες περιπτώσεις η μείωση του λεξικού μπορεί να είναι αναγκαία. Τότε μειώνουμε το μέγεθος του λεξικού με διάφορους μετασχηματισμούς του λεξικού και των λέξεων του. Ανάλογα με το μοντέλο μάθησης, αυτή η διαδικασία μπορεί να βελτιώσει την απόδοση της πρόγνωσης. Μια περίπτωση είναι η δημιουργία τοπικού λεξικού. Αν έχουμε ένα πρόβλημα δυαδικής ταξινόμησης σε επιβλεπόμενη μάθηση μπορούμε να δημιουργήσουμε ένα λεξικό μόνο με τις λέξεις της μιας κλάσης. Μια άλλη μέθοδος μείωσης του μεγέθους του λεξικού είναι η αφαίρεση των stopwords. Stopwords είναι λέξεις οι οποίες δεν προσφέρουν τίποτα στην προγνωστική ικανότητα, όπως τα άρθρα, αντωνυμίες κλπ. Η συχνότητα εμφάνισης των λέξεων μπορεί να αποτελέσει μια αποτελεσματική μέθοδος μείωσης του όγκου του λεξικού. Οι πιο συχνές λέξεις είναι συνήθως οι stopwords η οποίες και αφαιρούνται. Οι εναπομείνασες λέξεις με τη μεγαλύτερη συχνότητα εμφάνισης είναι και οι πιο σημαντικές, και θα μπορούσαν να αποτελέσουν το τοπικό λεξικό. Η διαδικασία της λημματοποίησης επίσης μπορεί να συνεισφέρει σημαντικά στην μείωση του λεξικού.

Σε προηγούμενη παράγραφο περιγράψαμε τη δημιουργία υπολογιστικού φύλλου που υποδηλώνει την απουσία ή την παρουσία της λέξης στο κείμενο, τοποθετώντας στο κελί τις τιμές 0 και 1 αντίστοιχα. Μια άλλη μέθοδος είναι η αντιστοίχιση στο κελί της συχνότητας εμφάνισης της λέξης, η οποία γενικά είναι χρήσιμη στην πρόβλεψη αλλά προσθέτει πολυπλοκότητα στις προτεινόμενες λύσεις. Μια ακόμα μέθοδος, η οποία λειτουργεί αρκετά καλά, είναι η αντιστοίχιση στο κελί της τιμής 0 για απουσία

της λέξης από το κείμενο, 1 για παρουσία της λέξης μόνο μια φορά, 2 για παρουσία της λέξης τουλάχιστον δύο φορές.

Το επόμενο βήμα μετά την μέτρηση της συχνότητας εμφάνισης μια λέξης στο κείμενο είναι ο μετασχηματισμός αυτής της καταμέτρησης με βάση την σημαντικότητα αυτής της λέξης. Η διαδικασία TF-IDF(Term Frequency-Inverse Document Frequency) είναι μια μέθοδος για να υπολογιστεί η βαρύτητα ή η βαθμολογία των λέξεων. Ο τρόπος υπολογισμού της εκφράζεται στη σχέση:

$$tf - idf(j) = tf(j) * idf(j)$$

$$idf(j) = \log \frac{N}{df(j)}$$

Όπου j ή λέξη, tf(j) το πλήθος των εμφανίσεων της λέξης στο μήνυμα, N το συνολικό πλήθος μηνυμάτων στο σύνολο των δεδομένων και df(j) το συνολικό πλήθος των μηνυμάτων που περιέχουν την λέξη j.
[44] [45]

2.4 Η γλώσσα προγραμματισμού Python

Η Python είναι μια γλώσσα προγραμματισμού υψηλού επιπέδου, ανοικτού πηγαίου κώδικα (open source) και γενικής χρήσης. Είναι εύκολη στην εκμάθηση και παρέχει ισχυρές δυνατότητες σε αρχάριους και έμπειρους προγραμματιστές. Ένα από τα κύρια χαρακτηριστικά της είναι η αντικειμενοστρέφεια. Αναπτύχθηκε από τον Guido van Rossum, στις αρχές της δεκαετίας ’90 ως διάδοχος της γλώσσας ABC και το όνομά της προέρχεται από την ομάδα κωμικών Monty Python. Πρόκειται για μία υψηλού επιπέδου αντικειμενοστρεφή γλώσσα προγραμματισμού με σκοπό την εύκολη και γρήγορη δημιουργία κώδικα, αφού δεν κάνει χρήση ειδικών συμβόλων και σημείων στίξης αλλά κενών διαστημάτων(whitespaces) για την σύνταξη των εντολών της. [46]

2.4.1 Πλεονεκτήματα της Python

- Είναι αντικειμενοστρεφής

Υποστηρίζει προηγμένες έννοιες όπως πολυμορφισμό, υπερφόρτωση χειριστή, πολλαπλή κληρονομικότητα που σε συνδυασμό με την απλή της σύνταξη διευκολύνει την εφαρμογή του αντικειμενοστρεφή προγραμματισμού.

- Είναι δωρεάν

Όπως και με άλλα λογισμικά ανοιχτού κώδικα, όπως Tcl, Perl, Linux και Apache ολόκληρος ο πηγαίος κώδικας του συστήματος Python μπορεί να ανακτηθεί δωρεάν από το Διαδίκτυο. Δεν υπάρχουν περιορισμοί στην αντιγραφή του, την ενσωμάτωσή του στα διάφορα συστήματά κλπ.

- Είναι φορητή

Λόγω του ανοικτού της κώδικα, η Python έχει υλοποιηθεί (δηλαδή αλλάχθηκε για να λειτουργεί) σε πολλές πλατφόρμες. Όλα τα Python προγράμματά μπορούν να δουλέψουν σε κάθε σημαντική πλατφόρμα που χρησιμοποιείται αυτήν τη στιγμή. Τα προγράμματα Python εκτελούνται σήμερα παντού, από PDA έως υπερυπολογιστές. Μπορεί να χρησιμοποιηθεί σε Συστήματα Linux και Unix, Microsoft Windows και DOS, Mac OS (και OS X και Classic), BeOS, OS / 2, VMS και QNX,VxWorks, Cray supercomputers και IBM mainframes, PDAs που λειτουργούν με Palm OS, PocketPC, κινητά τηλέφωνα που χρησιμοποιούν Symbian OS και Windows Mobile, Κονσόλες παιχνιδιών και iPod.

- Είναι ισχυρή

Από την σκοπιά των χαρακτηριστικών η python μπορεί να θεωρηθεί σαν κάτι υβριδικό. Το εύρος των εργαλείων της τοποθετείται μεταξύ των παραδοσιακών scripting γλωσσών προγραμματισμού όπως η Tcl, η Scheme, και η Perl, και γλωσσών ανάπτυξης συστημάτων όπως η C, C++ και η Java. Η Python παρέχει όλη την απλότητα και την ευκολία χρήσης των γλωσσών scripting μαζί με προηγμένα εργαλεία μηχανικής λογισμικού

- Είναι ενσωματώσιμη

Τα προγράμματα της Python μπορούν εύκολα να ενσωματωθούν σε άλλες γλώσσες όπως η C/C++

- Είναι εύκολη στη χρήση

Η Python εκτελεί άμεσα τα προγράμματα, κάτι που δημιουργεί μια διαδραστική εμπειρία προγραμματισμού. Το αποτέλεσμα μιας αλλαγής στο πρόγραμμα εμφανίζεται άμεσα για τον χρήστη. Έχει απλή σύνταξη και ισχυρά ενσωματωμένα εργαλεία. Θεωρείται από κάποιους «εκτελέσιμος ψευδοκώδικας» επειδή εξαλείφει μεγάλο μέρος της πολυπλοκότητας, τα προγράμματα της είναι απλούστερα, μικρότερα και πιο ευέλικτα από τα αντίστοιχα προγράμματα σε γλώσσες όπως C, C ++ και Java.

- Είναι εύκολη στη εκμάθηση

Σε σύγκριση με άλλες γλώσσες προγραμματισμού, η βασική γλώσσα Python είναι εξαιρετικά εύκολη στην εκμάθηση. [47]

2.4.2 Η Python για την Ανάλυση Δεδομένων και τη Μηχανική Μάθηση

Η Python είναι μια γλώσσα προγραμματισμού που συνδυάζει τη δυναμική των γλωσσών προγραμματισμού γενικής χρήσης με τις δυνατότητες γλωσσών προγραμματισμού ειδικού πεδίου (domain-specific programming language) όπως η Matlab ή η R. Η Python διαθέτει βιβλιοθήκες για την είσοδο δεδομένων, για την οπτικοποίηση, για στατιστικά στοιχεία, για την επεξεργασία της φυσικής γλώσσας, για την επεξεργασία εικόνων κ.α. Αυτή η τεράστια γκάμα εργαλείων που διαθέτει παρέχει στους επιστήμονες των δεδομένων σημαντικές δυνατότητες. Ένα από τα κύρια πλεονεκτήματα της Python είναι η δυνατότητα που δίνει στον χρήστη για άμεση αλληλεπίδραση με τον κώδικα με τη χρήση ενός τερματικού ή άλλων εργαλείων όπως το Jupyter Notebook . Για την μηχανική μάθηση και την ανάλυση των δεδομένων είναι σημαντικό να χρησιμοποιούνται εργαλεία που επιτρέπουν γρήγορη επανάληψη και εύκολη αλληλεπίδραση με τον χρήστη, όπως αυτά που διαθέτει η Python.

2.4.3 Βασικές βιβλιοθήκες και εργαλεία της Python

- Jupyter Notebook

Το Jupyter Notebook είναι ένα διαδραστικό περιβάλλον για την εκτέλεση κώδικα. Είναι ένα πολύ χρήσιμο εργαλείο για διερευνητική ανάλυση και χρησιμοποιείται ευρέως από επιστήμονες των δεδομένων.

- Scikit-Learn

Η Scikit-Learn είναι μια βιβλιοθήκη μηχανικής μάθησης, ανοιχτού κώδικα, που υποστηρίζει την εποπτευόμενη και μη εποπτευόμενη μάθηση. Περιέχει διάφορα χρήσιμα εργαλεία μηχανικής μάθησης, για την επεξεργασία και την προετοιμασία δεδομένων, για την επιλογή και αξιολόγηση του μοντέλου όπως και πολλά άλλα βοηθητικά προγράμματα [48]. Αποτελεί το πιο δημοφιλές και διακεκριμένο εργαλείο της Python για τη μηχανική μάθηση.

- NumPy

Η NumPy είναι μια βιβλιοθήκη ανοιχτού κώδικα με στόχο την εφαρμογή αριθμητικών υπολογισμών μέσω της Python. Δημιουργήθηκε το 2005 βασιζόμενη στις προγενέστερες βιβλιοθήκες Numercial και Numarray [49] Η NumPy είναι ένα από τα θεμελιώδη πακέτα για επιστημονικούς υπολογισμούς στην Python. Εμπεριέχει λειτουργίες για πολυδιάστατους πίνακες, μαθηματικές συναρτήσεις υψηλού επιπέδου, εφαρμογές στη γραμμική άλγεβρα κ.α. Η Scikit-Learn λαμβάνει σαν είσοδο μόνο τα δεδομένα που είναι στη μορφή πίνακα NumPy.

- SciPy

Η SciPy είναι μια συλλογή μαθηματικών αλγορίθμων και συναρτήσεων βασισμένη στην βιβλιοθήκη NumPy της Python. Προσθέτει σημαντικές δυνατότητες στη διαδραστικότητα της Python, καθώς παρέχει στον χρήστη εντολές υψηλού επιπέδου και κλάσεις για την διαχείριση και την οπτικοποίηση των δεδομένων. Ένα ακόμα πλεονέκτημα που προσθέτει η SciPy στην Python είναι ότι ενισχύει τη δυνατότητα για χρήση της στην ανάπτυξή επιστημονικών προγραμμάτων και εξειδικευμένων εφαρμογών. [50]

- Matplotlib

Η matplotlib είναι η κύρια βιβλιοθήκη σχεδίασης γραφημάτων στην Python. Παρέχει πολλαπλές λειτουργίες για την οπτικοποίηση δεδομένων, όπως γραμμικά γραφήματα, ιστογράμματα, γραφήματα διασποράς κ.α. Στην Ανάλυση Δεδομένων η οπτικοποίηση των δεδομένων είναι μια πολύ κρίσιμη διαδικασία καθώς μπορεί να αποκαλύψει στον χρήστη σημαντικές πληροφορίες.

- Seaborn

Η Seaborn είναι μια βιβλιοθήκη για την δημιουργία στατιστικών γραφικών στην Python. Έχει δημιουργηθεί στην κορυφή της matplotlib και ενσωματώνεται εύκολα με την βιβλιοθήκη Pandas.

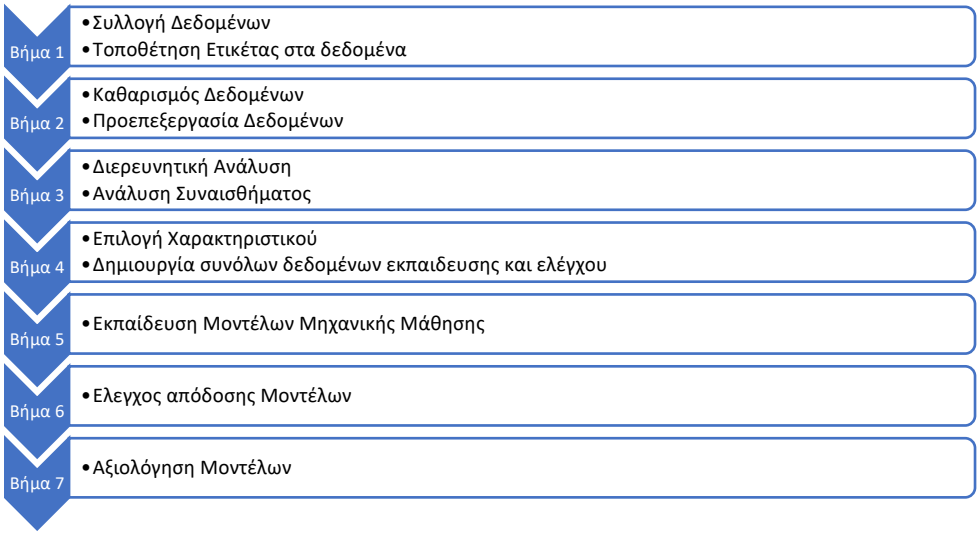
- Pandas

Η Pandas είναι μια βιβλιοθήκη ανοιχτού κώδικα ιδανική για επεξεργασία και ανάλυση δεδομένων. Η δημιουργία και ανάπτυξη της Pandas έχει ως στόχο να αποτελέσει το θεμελιώδες και υψηλού επιπέδου στοιχείο για την πρακτική και πραγματική ανάλυση δεδομένων στην Python [51]. Βασικό της στοιχείο είναι μια δομή δεδομένων που ονομάζεται DataFrame το οποίο επί της ουσίας είναι ένας πίνακας αντίστοιχος με το spreadsheet του Excel, για το οποίο η Pandas παρέχει μια ποικιλία από μεθόδους για επεξεργασία και ανάλυση. Σε αντίθεση με την NumPy, η οποία προϋποθέτει ότι όλα τα στοιχεία του πίνακα να είναι του ίδιου τύπου, η Pandas παρέχει την δυνατότητα κάθε στήλη να έχει διαφορετικό τύπο δεδομένων. Μια ακόμη πολύτιμη δυνατότητα που παρέχει η Pandas είναι η ικανότητα να ανακτά δεδομένα από μια μεγάλη ποικιλία αρχείων και βάσεων δεδομένων όπως csv files, Excel files, SQL.

3 Μεθοδολογική Προσέγγιση

Για την υλοποίηση του πειράματος συλλέχθηκαν δεδομένα στα οποία τοποθετήθηκε ετικέτα 0 ή 1. Ακολούθησε καθαρισμός και προεπεξεργασία των δεδομένων, διερευνητική ανάλυση και ανάλυση συναισθήματος. Στη συνέχεια έγινε διανυσματικοποίηση των δεδομένων και επιλογή ενός κατάλληλου χαρακτηριστικού για την εκπαίδευση των μοντέλων. Ακολούθησε η διαδικασία Μηχανικής Μάθησης, εκπαίδευση των μοντέλων και έλεγχός της απόδοσης τους με χρήση διάφορων μετρικών. Σε όλα τα βήματα της διαδικασίας χρησιμοποιήθηκε η γλώσσα προγραμματισμού Python.

Στο διάγραμμα που ακολουθεί απεικονίζεται η διαδικασία που ακολουθήθηκε για την υλοποίηση του πειράματος.



3.1 Συλλογή Δεδομένων

Τα άρθρα(δεδομένα) που χρησιμοποιήθηκαν για το πείραμα προέρχονται από το αρχείο έγκυρης ηλεκτρονικής ειδησεογραφικής ιστοσελίδας και από το αρχείο ιστοσελίδων που έχουν χαρακτηριστεί από το Greek Hoaxes Detector [1] ως ιστοσελίδες με αναξιόπιστο περιεχόμενο. Για τη διαδικασία συλλογής των άρθρων χρειάστηκε να αναπτυχθεί, με τη χρήση της γλώσσας προγραμματισμού Python, ένας κώδικας ανίχνευσης ιστού που να συλλέγει τα έγγραφα, ξεχωριστός για την κάθε ιστοσελίδα ανάλογα με το τρόπο αρχειοθέτησης των άρθρων της. Το διάστημα δημοσίευσης των άρθρων που ανακτήθηκαν είναι από 1/1/2020 έως 18/11/2020.

Για την αποθήκευση των άρθρων δημιουργήθηκε μια βάση δεδομένων με τη διασύνδεση Python και SQLite. Στα άρθρα που προέρχονταν από την έγκυρη ειδησεογραφική σελίδα τοποθετήθηκε η ετικέτα 0 και στα άρθρα που προέρχονταν από τις χαρακτηρισμένες ως αναξιόπιστες πηγές τοποθετήθηκε η ετικέτα 1.

3.2 Καθαρισμός και προεπεξεργασία Δεδομένων

Για τον καθαρισμό των δεδομένων ακολουθήθηκαν οι εξής ενέργειες:

- Μετατροπή ημερομηνιών σε συμβατή μορφή
- Απαλοιφή διπλών εγγραφών
- Απαλοιφή URL
- Μετατροπή κεφαλαίων σε μικρά
- Απαλοιφή ειδικών χαρακτήρων και σημείων στίξης
- Απαλοιφή αριθμών
- Απαλοιφή τόνων
- Φιλτράρισμα δεδομένων που περιέχουν λέξεις κλειδιά για τον COVID-19

Στη συνέχεια για την προεπεξεργασία των δεδομένων ακολουθήθηκαν τεχνικές Εξόρυξης Γνώσης Κειμένων(Text Mining):

- Κατακερματισμός
- Αφαίρεση Stopwords
- Λημματοποίηση

3.3 Διερευνητική Ανάλυση

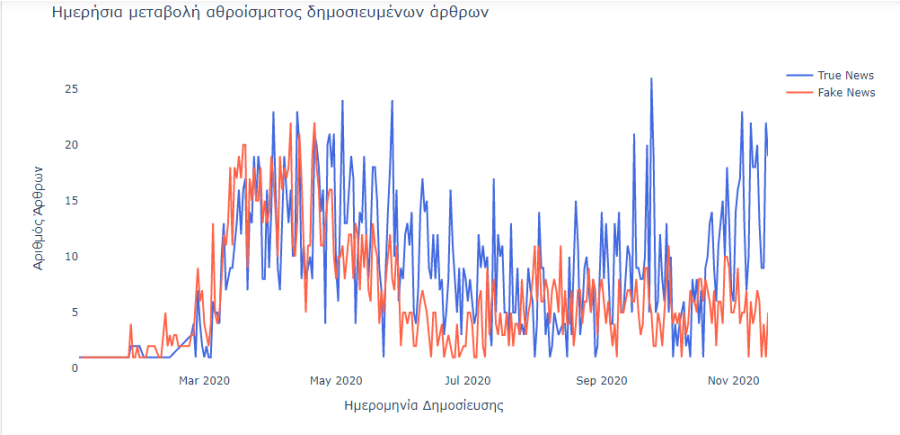
Μετά τον καθαρισμό και την προεπεξεργασία των δεδομένων ακολούθησε Διερευνητική Ανάλυση στα δεδομένα. Παρακάτω παρουσιάζονται τα στοιχεία και οι μετρικές που αναλύθηκαν παρουσιάζονται.

Καταμέτρηση Δεδομένων

Το σύνολο των δεδομένων περιέχει 4715 άρθρα που αφορούν τον COVID-19 εκ των οποίων 2664 με ετικέτα 0(True News) και 2051 με ετικέτα 1(Fake News).

Καταμέτρηση ημερήσιας μεταβολής άρθρων

Στην Εικόνα 3.1 αναπαρίσταται η ημερήσια μεταβολή του αθροίσματος των δημοσιευμένων άρθρων για τον COVID-19 ανάλογα με την ετικέτα τους.



ΕΙΚΟΝΑ 3.1-ΗΜΕΡΗΣΙΑ ΜΕΤΑΒΟΛΗ ΑΡΙΘΜΟΥ ΆΡΘΡΩΝ

Στατιστικά που αφορούν το άθροισμα των λέξεων (χωρίς την αφαίρεση των stopwords) για τους Τίτλους και τα Κείμενα των άρθρων

Στην Εικόνα 3.2 αποτυπώνεται ο πίνακας των στατιστικών και στην Εικόνα 3.3 αναπαρίστανται οι Κατανομές των Αθροισμάτων των λέξεων των Τίτλων και των Άρθρων ανάλογα με την ετικέτα τους. Από τον πίνακα και την γραφική παράσταση παρατηρούμε ότι τα Fake News έχουν αρκετά μεγαλύτερο αριθμό λέξεων ανά άρθρο, όπως επίσης και μεγαλύτερη διασπορά. Κάποιοι δείκτες μέτρησης που παρουσιάζονται είναι:

Τίτλοι

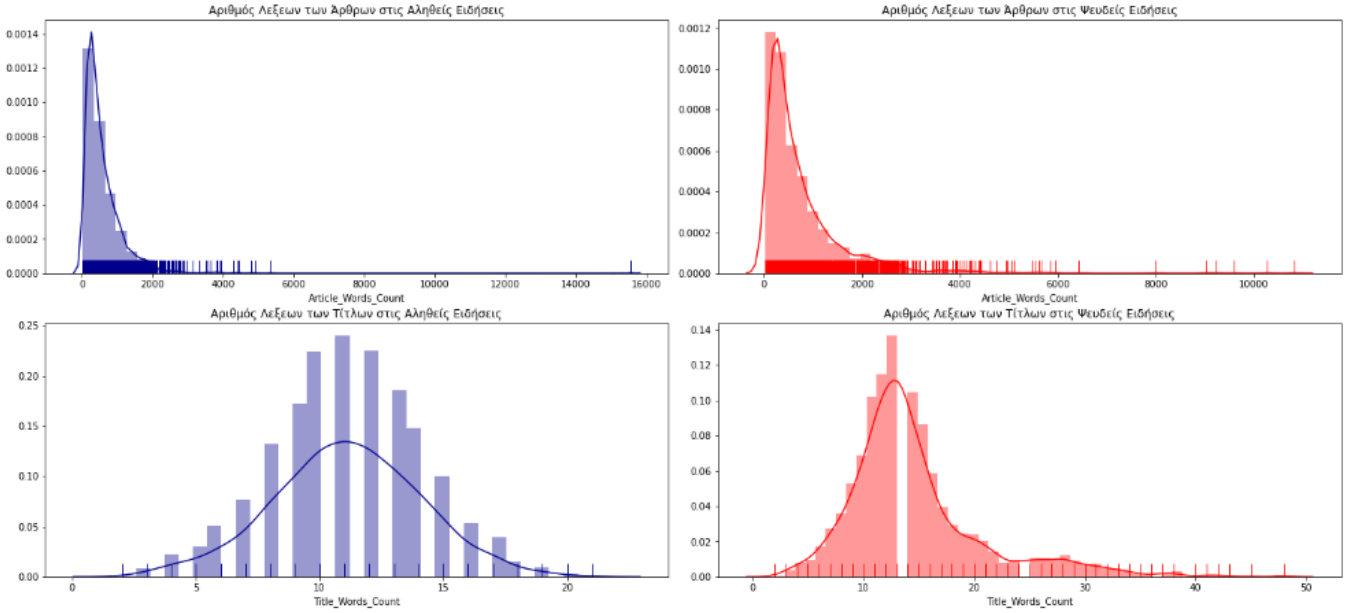
- Μ.Ο λέξεων ανά άρθρο στα True News: 11.13
- Μ.Ο λέξεων ανά άρθρο στα Fake News: 14.59

Κείμενο Άρθρων

- Μ.Ο λέξεων ανά άρθρο στα True News: 596.83
- Μ.Ο λέξεων ανά άρθρο στα Fake News: 790.04

	Article_Words_Count_True	Article_Words_Count_Fake	Title_Words_Count_True	Title_Words_Count_Fake
count	2664.00	2051.00	2664.00	2051.00
mean	596.83	790.04	11.13	14.59
std	617.19	946.36	3.00	6.15
min	31.00	27.00	2.00	2.00
25%	232.00	236.00	9.00	11.00
50%	422.00	473.00	11.00	13.00
75%	777.25	981.00	13.00	16.00
max	15559.00	10815.00	21.00	48.00

ΕΙΚΟΝΑ 3.2-ΠΙΝΑΚΑΣ ΣΤΑΤΙΣΤΙΚΩΝ ΑΘΡΟΙΣΜΑΤΟΣ ΤΩΝ ΛΕΞΕΩΝ



ΕΙΚΟΝΑ 3.3-ΑΠΕΙΚΟΝΙΣΗ ΚΑΤΑΝΟΜΗΣ ΑΘΡΟΙΣΜΑΤΟΣ ΛΕΞΕΩΝ

Απεικόνιση n-grams

Με τον όρο n-gram εννοούμε μια ακολουθία n μονάδων. Μπορεί να έχουμε n-gram όπου η μονάδα είναι ένας χαρακτήρας, μια συλλαβή, μια λέξη, τα σημεία στίξης κλπ. Οι διάφορες τιμές του n δημιουργούν n-gram διαφορετικού μήκους.Έτσι για n=1 έχουμε unigram, για n=2 bigram, για n=3 tirgram. Στην **Εικόνα 3.4** απεικονίζονται τα 15 πιο συχνά εμφανιζόμενα bigram και trigram για κάθε ετικέτα(True-0 ή Fake-1) με μονάδα τις λέξεις(χωρίς τις stopwords). Από το αποτέλεσμα παρατηρούμε αρκετές διαφορές στα n-grams με βάση την ετικέτα των άρθρων.

True News

	bidgram	frequency
0	(κυριακος, μητσοτακης)	1030
1	(κυβερνητικος, εκπροσωπος)	566
2	(συστημα, υγειας)	494
3	(αλεξης, τσιπρας)	492
4	(δημοσιας, υγειας)	470
5	(ανατολικη, μεσογειο)	435
6	(πρωθυπουργος, κυριακος)	403
7	(στελιος, πετσας)	382
8	(αξιωματικης, αντιπολιτευσης)	358
9	(αλεξη, τσιπρα)	317
10	(φωφη, γεννηματα)	309
11	(οσον, αφορα)	307
12	(υπουργος, εξωτερικων)	303
13	(συστηματος, υγειας)	301
14	(κυριακου, μητσοτακη)	278

True News

	tridgram	frequency
0	(πρωθυπουργος, κυριακος, μητσοτακης)	402
1	(κυβερνητικος, εκπροσωπος, στελιος)	221
2	(εκπροσωπος, στελιος, πετσας)	221
3	(εθνικο, συστημα, υγειας)	161
4	(δημοσιο, συστημα, υγειας)	127
5	(πρωθυπουργο, κυριακο, μητσοτακη)	125
6	(συριζα, προοδευτικη, συμμαχια)	121
7	(προστασια, δημοσιας, υγειας)	119
8	(αρχηγος, αξιωματικης, αντιπολιτευσης)	105
9	(δημοσιου, συστηματος, υγειας)	103
10	(πρωθυπουργου, κυριακου, μητσοτακη)	102
11	(υγειας, βασιλης, κικιλιας)	99
12	(εθνικου, συστηματος, υγειας)	97
13	(υπουργος, υγειας, βασιλης)	95
14	(ενημερωση, πολιτικων, συντακτων)	90

Fake News

	bidgram	frequency
0	(χρηση, μασκας)	484
1	(δημοσιας, υγειας)	474
2	(οσον, αφορα)	242
3	(λιανικο, εμποριο)	232
4	(bill, gates)	220
5	(υπουργειου, υγειας)	219
6	(μπιλ, γκειτς)	209
7	(σωτηρης, τσιοδρας)	205
8	(σουπερ, μαρκετ)	201
9	(πρωτη, φορα)	199
10	(δημοσια, υγεια)	187
11	(ηνωμενες, πολιτειες)	180
12	(πολιτικης, προστασιας)	179
13	(χρονικο, διαστημα)	175
14	(κυριακος, μητσοτακης)	172

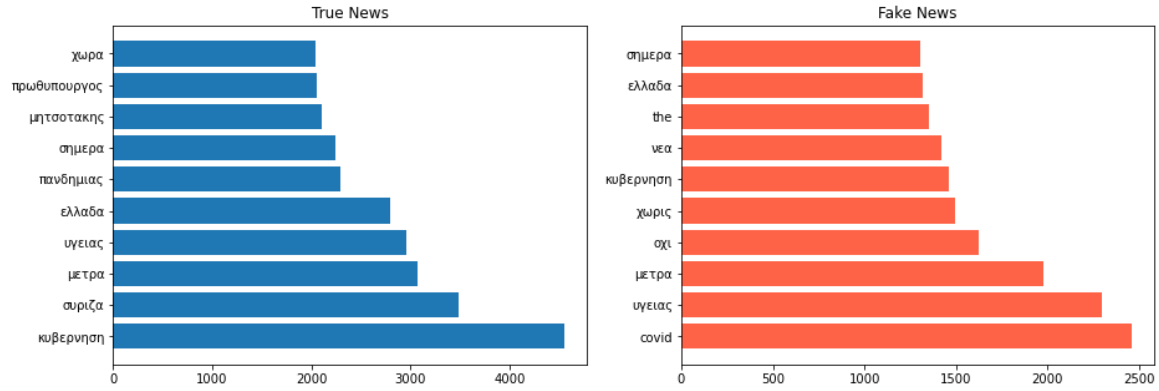
Fake News

	tridgram	frequency
0	(υποχρεωτικη, χρηση, μασκας)	114
1	(παγκοσμιος, οργανισμος, υγειας)	77
2	(ηλεκτρονικου, τηλεφωνικου, εμποριου)	69
3	(τηλεφωνικου, εμποριου, παραδοση)	69
4	(εμποριου, παραδοση, οικον)	69
5	(παραδοση, οικον, eshop)	69
6	(οικον, eshop, κτλ)	69
7	(υπηρεσιες, ηλεκτρονικου, τηλεφωνικου)	65
8	(eshop, κτλ, λιανικο)	62
9	(κτλ, λιανικο, εμποριο)	62
10	(υφυπουργος, πολιτικης, προστασιας)	61
11	(οσων, εργαζομενων, αφμ)	60
12	(εργαζομενων, αφμ, ληγει)	60
13	(εξαιρεση, υπηρεσιες, ηλεκτρονικου)	57
14	(παγκοσμιου, οργανισμου, υγειας)	56

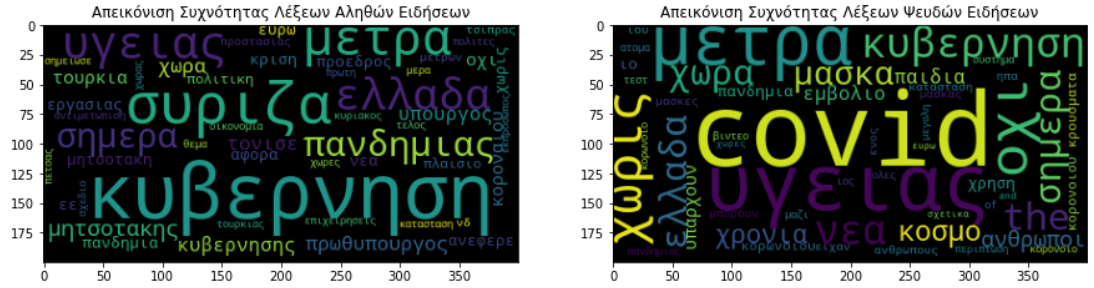
ΕΙΚΟΝΑ 3.4-N-GRAMS

Συχνότητα εμφάνισης λέξεων

Στην **Εικόνα 3.5** απεικονίζονται οι 10 λέξεις που εμφανίζονται πιο συχνά στα άρθρα για κάθε ετικέτα ξεχωριστά. Στην **Εικόνα 3.6** απεικονίζονται τα συννεφόλεξα με τις 50 λέξεις που εμφανίζονται πιο συχνά στα άρθρα για κάθε ετικέτα ξεχωριστά. Από τα συννεφόλεξα και το γράφημα με τις 10 πιο συχνές λέξεις παρατηρούμε ότι υπάρχουν σημαντικές διαφορές στις λέξεις που χρησιμοποιούνται σε True και Fake άρθρα.



ΕΙΚΌΝΑ 3.5-ΔΙΑΓΡΑΜΜΑ ΣΥΧΝΟΤΗΤΑΣ ΛΕΞΕΩΝ



ΕΙΚΌΝΑ 3.6-ΣΥΝΝΕΦΟΛΕΞΟ ΣΥΧΝΟΤΗΤΑΣ ΛΕΞΕΩΝ

3.4 Ανάλυση Συναισθήματος

Για την υλοποίηση Ανάλυσης Συναισθήματος αξιοποιήθηκε η μεθοδολογία που περιγράφεται στην εργασία «*Συναισθηματική Ανάλυση Ελληνικών Tweets και Hashtags με χρήση λεξικού Συναισθημάτων*» [52].

Έγινε ανάλυση συναισθήματος με τη χρήση του λεξικού του Adam Tsakalidis, "Greek Sentiment Lexicon". Το λεξικό είναι διαθέσιμο στο https://github.com/MKLab-ITI/greek-sentiment-lexicon. Το λεξικό περιλαμβάνει συναισθηματική αξιολόγηση των λημμάτων από τέσσερεις ανεξάρτητους βαθμολογητές. Στην εργασία αυτή επιλέχθηκε ο μέσος όρος των τεσσάρων βαθμολογιών ώστε να προκύψουν οι τελικές βαθμολογίες κάθε λήμματος. Τα πεδία που χρησιμοποιήθηκαν είναι Polarity, Anger, Disgust, Fear, Happiness, Sadness, Surprise.

Κάποια χαρακτηριστικά του λεξικού:

- Το λεξικό περιέχει 2,315 λήμματα
- Πολικότητα(Polarity): Έχει τις τιμές Θετικό(Positive) -> («POS»), Αρνητικό(Negative) -> («NEG»), Θετικό και Αρνητικό(Both positive and negative) -> («BOTH») και Ουδέτερο(Neutral) -> («N/A»)
- Θυμός(Anger), Αηδία(Disgust), Φόβος(Fear), Χαρά(Happiness), Λύπη(Sadness), Έκπληξη(Surprise): Έχουν τιμές 1-5 και («N/A»)
- Στις περιπτώσεις των επίθετων, και τα τρία φύλα (αρσενικό, θηλυκό, ουδέτερο) υπονοούνται με την παροχή των επιθημάτων (-ος -η -ο) πχ αλογικός -η -ο.

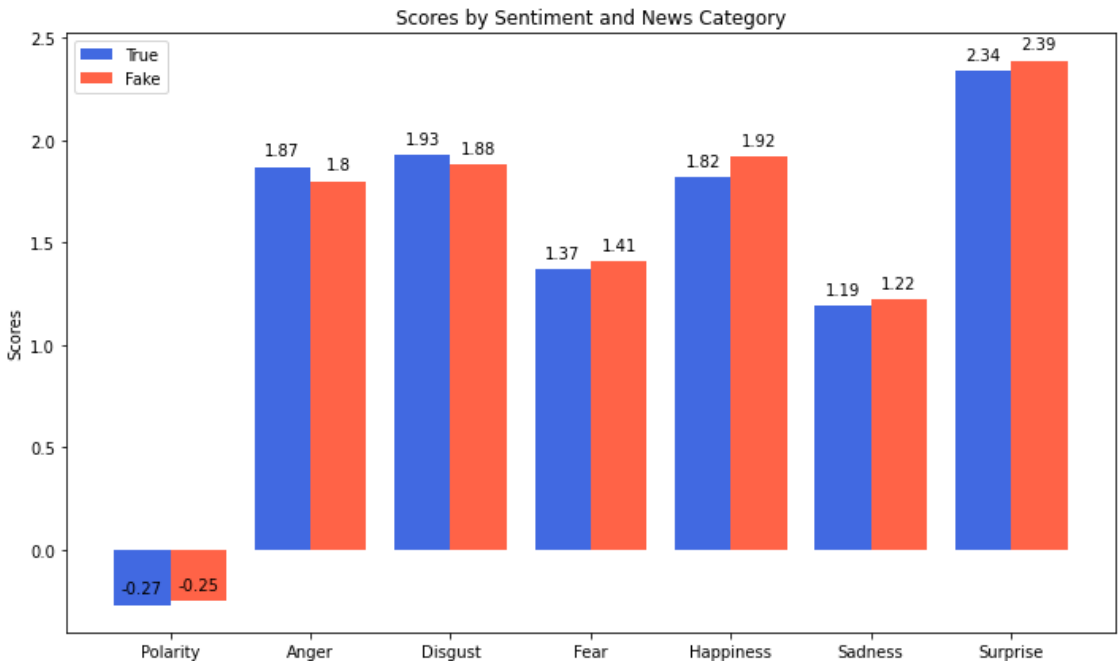
Επεξεργασία λεξικού:

- Αντικατάσταση των τιμών Polarity ως εξής:
  - POS->1
  - NEG-> -1
  - BOTH->0
  - Neutral -> None
- Μετατροπή των κεφαλαίων γραμμάτων σε πεζά
- Αφαίρεση των τόνων
- Αφαίρεση των καταλήξεων -ος, -η, -ο

Για την καλύτερη αντιστοίχιση σε ομόριζες λέξεις ακολουθήθηκε η διαδικασία Λημματοποίησης στα άρθρα και στα λήμματα του λεξικού. Ο stemmer που χρησιμοποιήθηκε είναι διαθέσιμος στο: https://pypi.org/project/greek-stemmer/ . Για τις ανάγκες λειτουργίας του stemmer μετατρέψαμε όλα τα γράμματα σε κεφαλαία. Ο τρόπος που χρησιμοποιείται είναι ως εξής:

```
from greek_stemmer import GreekStemmer
stemmer = GreekStemmer()
stemmer.stem('ΘΑΛΑΣΣΑ')
Output: 'ΘΑΛΑΣΣ'
```

Στις **Εικόνα 3.7**, **Εικόνα 3.8**, **Εικόνα 3.9**, **Εικόνα 3.10**, **Εικόνα 3.11**, **Εικόνα 3.12** και **Εικόνα 3.13** παρουσιάζονται τα γραφήματα τις ανάλυσης συναισθήματος..



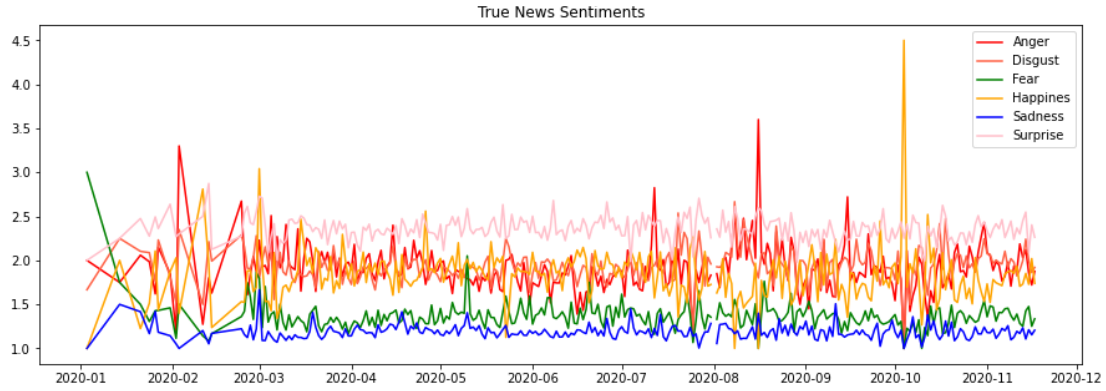
ΕΙΚΌΝΑ 3.7-ΑΠΟΔΟΣΗ ΑΝΑ ΣΥΝΑΙΣΘΗΜΑ

	Polarity	Anger	Disgust	Fear	Happiness	Sadness	Surprise
count	278.000000	278.000000	278.000000	278.000000	278.000000	278.000000	278.000000
mean	-0.252225	1.903998	1.935142	1.371023	1.834166	1.189370	2.358372
std	0.189341	0.272092	0.186682	0.164316	0.299570	0.083903	0.136296
min	-1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.981481
25%	-0.346921	1.750552	1.833750	1.288119	1.689565	1.139471	2.272225
50%	-0.245787	1.880994	1.921011	1.356404	1.848949	1.180627	2.353779
75%	-0.155032	2.021587	2.014782	1.428526	1.971567	1.222587	2.444958
max	1.000000	3.600000	2.666667	3.000000	4.500000	1.666667	2.873656

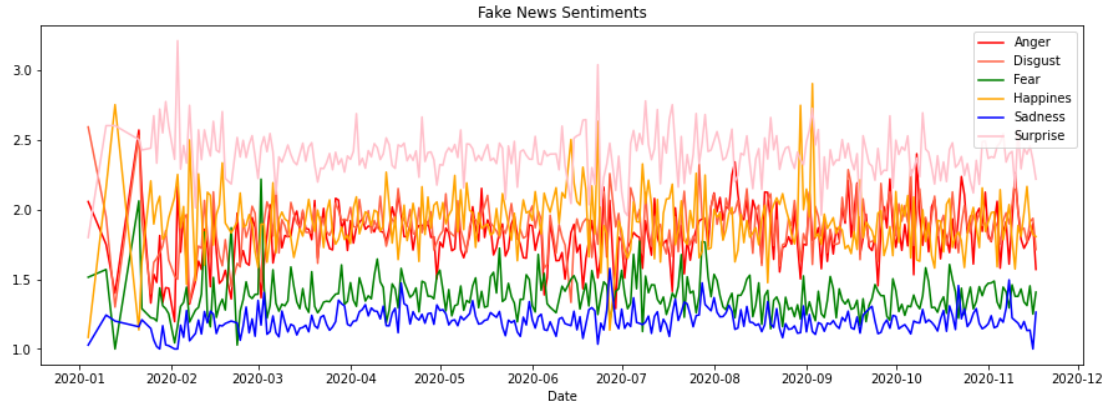
ΕΙΚΟΝΑ 3.8-ΣΤΑΤΙΣΤΙΚΑ ΣΥΝΑΙΣΘΗΜΑΤΩΝ ΓΙΑ TRUE NEWS

	Polarity	Anger	Disgust	Fear	Happiness	Sadness	Surprise
count	296.000000	296.000000	296.000000	296.000000	296.000000	296.000000	296.000000
mean	-0.240309	1.813696	1.881179	1.380486	1.911956	1.202350	2.394040
std	0.163599	0.199284	0.171853	0.139814	0.214272	0.082295	0.161464
min	-0.958333	1.195652	1.300000	1.000000	1.083333	1.000000	1.798611
25%	-0.332706	1.705643	1.786872	1.296085	1.792697	1.151399	2.320570
50%	-0.244243	1.812639	1.891531	1.362892	1.899692	1.191452	2.394491
75%	-0.155504	1.924137	1.973952	1.448390	2.029778	1.251800	2.469850
max	0.351256	2.568182	2.590278	2.216519	2.901093	1.576923	3.208333

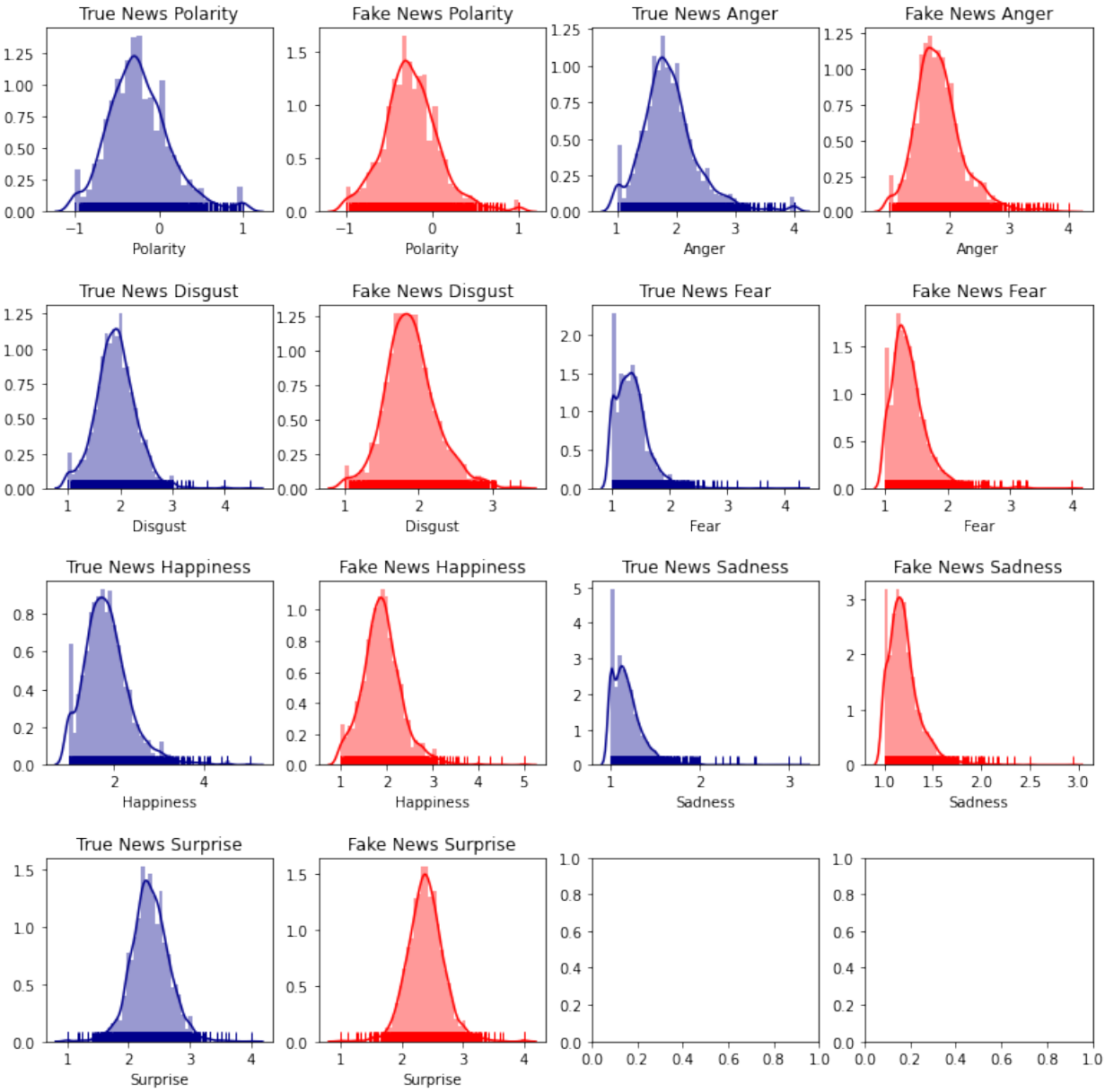
ΕΙΚΟΝΑ 3.9-ΣΤΑΤΙΣΤΙΚΑ ΣΥΝΑΙΣΘΗΜΑΤΩΝ ΓΙΑ FAKE NEWS



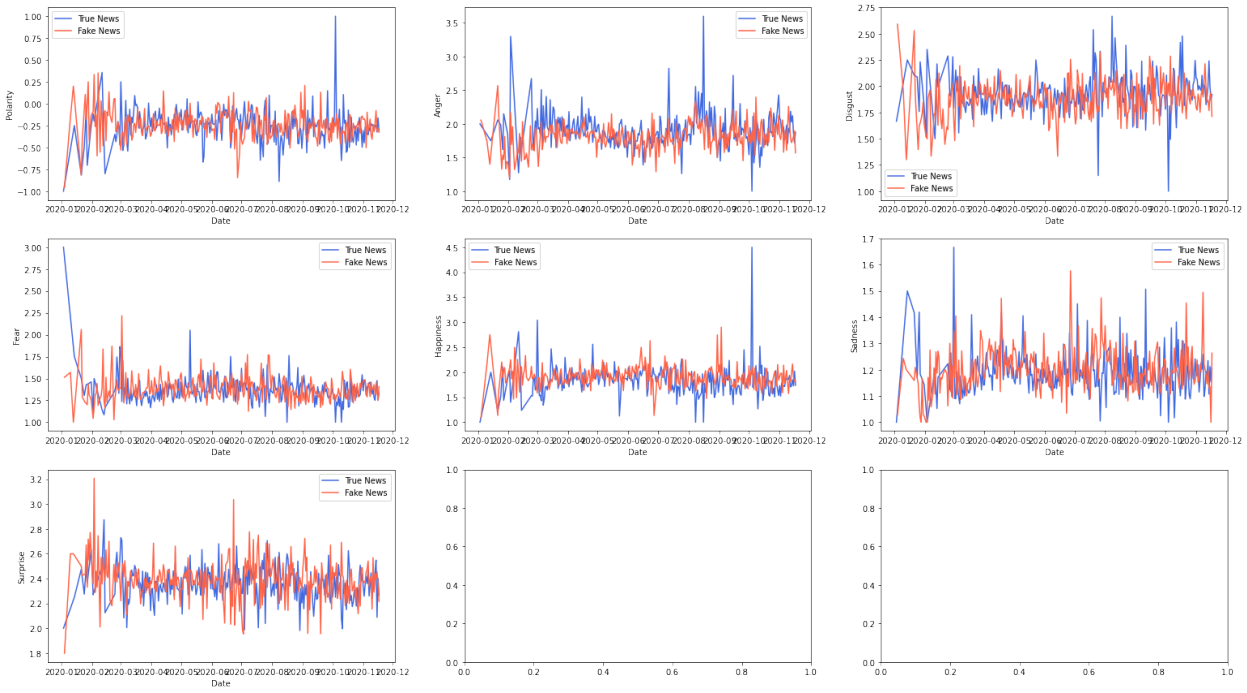
ΕΙΚΟΝΑ 3.10-ΗΜΕΡΗΣΙΑ ΜΕΤΑΒΟΛΗ ΣΥΝΑΙΣΘΗΜΑΤΩΝ TRUE NEWS



ΕΙΚΟΝΑ 3.11- ΗΜΕΡΗΣΙΑ ΜΕΤΑΒΟΛΗ ΣΥΝΑΙΣΘΗΜΑΤΩΝ FAKE NEWS



ΕΙΚΟΝΑ 3.12-ΓΡΑΦΗΜΑΤΑ ΚΑΤΑΝΟΜΗΣ ΣΥΝΑΙΣΘΗΜΑΤΩΝ



ΕΙΚΟΝΑ 3.13-ΓΡΑΦΗΜΑΤΑ ΗΜΕΡΗΣΙΑΣ ΜΕΤΑΒΟΛΗΣ ΑΝΑ ΣΥΝΑΙΣΘΗΜΑ

3.5 Επιλογή κατάλληλου χαρακτηριστικού και δημιουργία συνόλου δεδομένων εκπαίδευσης και δεδομένου ελέγχου

Μετά την ολοκλήρωση της Διερευνητικής Ανάλυσης και της Ανάλυσης Συναισθήματος επιλέχθηκε ως ένα κατάλληλο χαρακτηριστικό για την εκπαίδευση των μοντέλων μηχανικής μάθησης η συχνότητα εμφάνισης μιας λέξης και η μέθοδος TF-IDF(Term Frequency-Inverse Document Frequency).

Τα δεδομένα χωρίστηκαν με αναλογία 80/20 όπου 80% είναι το σύνολο των δεδομένων εκπαίδευσης και 20% είναι το σύνολο των δεδομένων ελέγχου. Για καλύτερη απόδοση χρησιμοποιήθηκαν οι λέξεις που εμφανίζονται σε ποσοστό κειμένων πάνω από 10%.

Στις **Εικόνα 3.14** και **Εικόνα 3.15** παρουσιάζονται οι πρώτες εγγραφές από τοι σύνολο δεδομένων εκπαίδευσης και το σύνολο δεδομένων ελέγχου. Τα δεδομένα αποθηκεύτηκαν σε αρχεία csv για την ανάκτηση τους στη διαδικασία εφαρμογής των Μοντέλων Μηχανικής Μάθησης.

Αναλυτικά οι διαστάσεις των πινάκων είναι:

	Documents	Features
Tain Set	3825	436
Test Set	957	436

	covid	lockdown	αγορ	αγων	αθην	ακολουθ	ακριβως	αλεξ	αλλ	αλλαγ	...	χθες	χιλιαδ	χρειαζ	χρης	χρησιμοποι	χρον	χωρ
0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.000000	0.000000	...	0.0	0.000000	0.0	0.0	0.0	0.000000	0.000000
1	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.000000	0.000000	...	0.0	0.124367	0.0	0.0	0.0	0.149283	0.000000
2	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.000000	0.000000	...	0.0	0.000000	0.0	0.0	0.0	0.000000	0.262195
3	0.0	0.0	0.0	0.0	0.041292	0.0	0.0	0.0	0.074052	0.487732	...	0.0	0.000000	0.0	0.0	0.0	0.000000	0.083818
4	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.000000	0.000000	...	0.0	0.000000	0.0	0.0	0.0	0.102702	0.083062

5 rows × 436 columns

ΕΙΚΟΝΑ 3.14-TRAIN SET

	covid	lockdown	αγορ	αγων	αθην	ακολουθ	ακριβως	αλεξ	αλλ	αλλαγ	...	χθες	χιλιαδ	χρειαζ	χρης	χρησιμοποι	χρ
0	0.000000	0.0	0.0	0.000000	0.000000	0.000000	0.000000	0.0	0.025007	0.000000	...	0.000000	0.0	0.118056	0.000000	0.000000	0.0000
1	0.082626	0.0	0.0	0.030643	0.062859	0.000000	0.028436	0.0	0.038365	0.000000	...	0.000000	0.0	0.012075	0.061859	0.025321	0.0160
2	0.000000	0.0	0.0	0.000000	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	...	0.000000	0.0	0.000000	0.000000	0.000000	0.0000
3	0.000000	0.0	0.0	0.000000	0.290090	0.000000	0.000000	0.0	0.177053	0.000000	...	0.000000	0.0	0.000000	0.000000	0.000000	0.1853
4	0.028949	0.0	0.0	0.000000	0.070475	0.031884	0.000000	0.0	0.043014	0.037576	...	0.039616	0.0	0.000000	0.000000	0.000000	0.0900

5 rows × 436 columns

ΕΙΚΟΝΑ 3.15-TEST SET

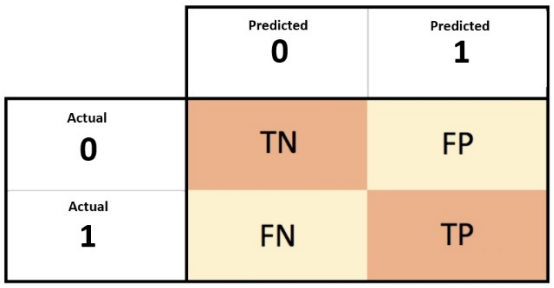
3.6 Εκπαίδευση και έλεγχος απόδοσης Μοντέλων Μηχανικής Μάθησης

Για την υλοποίηση του πειράματος εκπαιδεύτηκαν και ελέγχθηκαν τα μοντέλα Binomial Logistic Regression, Multinomial Naive Bayes Classifier, Support Vector Classifier και Random Forest.

Παρακάτω αναφέρονται οι μετρικές που χρησιμοποιήθηκαν για την αξιολόγηση της απόδοσης των μοντέλων.

Πίνακας Σύγχυσης(Confusion Matrix)

Ο πίνακας σύγχυσης είναι ένας NxN πίνακας που χρησιμοποιείται για την αξιολόγηση της απόδοσης ενός μοντέλου ταξινόμησης με N τον αριθμό των κλάσεων. Στο πείραμα μας το N ισούται με 2. Ο πίνακας συγκρίνει τις πραγματικές τιμές με αυτές που προβλέφθηκαν από το μοντέλο μηχανικής μάθησης, οπότε ο πίνακας δίνει μια συνολική εικόνα για το πόσο καλά αποδίδει το μοντέλο. Στην εικόνα απεικονίζεται ο πίνακας σύγχυσης για ένα πρόβλημα δυαδικής ταξινόμησης όπως το δικό μας.



ΕΙΚΌΝΑ 3.16-ΠΙΝΑΚΑΣ ΣΥΓΧΥΣΗΣ

- TP(True Positive): Ο αριθμός των ψευδών ειδήσεων που αναγνωρίστηκαν
- TN(True Negative): Ο αριθμός των αληθινών ειδήσεων που αναγνωρίστηκαν
- FP(False Positive): Ο αριθμός των ψευδών ειδήσεων που δεν αναγνωρίστηκαν
- FN(False Negative): Ο αριθμός των αληθινών ειδήσεων που αναγνωρίστηκαν εσφαλμένα ως ψευδείς

Ακρίβεια(Accuracy)

Η ακρίβεια είναι μια μετρική για την αξιολόγηση των μοντέλων ταξινόμησης. Άτυπα θα μπορούσαμε να πούμε ότι υπολογίζεται από τον τύπο:

$$Accuracy = \frac{Number\ of\ correct\ predictions}{Total\ Number\ of\ predictions}$$

Για την δυαδική ταξινόμηση μπορεί να υπολογιστεί από τον τύπο:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

Precision

Η μετρική Precision μας δείχνει πόσες από τις σωστά προβλεπόμενες περιπτώσεις αποδείχθηκαν θετικές(1).

$$Percision = \frac{TP}{TP + FP}$$

Recall

Η μετρική Recall μας δείχνει πόσες από τις πραγματικά θετικές περιπτώσεις(1) μπόρεσε να προβλέψει σωστά το μοντέλο.

$$Recall = \frac{TP}{TP + FN}$$

F1-Score

Η βαθμολογία F1 είναι ένας αρμονικός μέσος όρος Precision and Recall και έτσι δίνει μια συνδυασμένη ιδέα για αυτές τις δύο μετρήσεις. Είναι μέγιστο όταν το Precision είναι ίσο με το Recall.

$$F1 - Score = \frac{2}{\frac{1}{Recall} + \frac{1}{Precision}}$$

ROC και AUC

Μια καμπύλη ROC (χαρακτηριστική καμπύλη λειτουργίας δέκτη) είναι μια μέτρηση αξιολόγησης για δυαδικά προβλήματα ταξινόμησης. Είναι μια καμπύλη πιθανότητας που σχεδιάζει το TPR έναντι FPR, με:

$$TPR = \frac{TP}{TP + FN}$$

$$FPR = \frac{FP}{FP + TN}$$

Η περιοχή κάτω από την καμπύλη (AUC) είναι το μέτρο της ικανότητας ενός ταξινομητή να διακρίνει μεταξύ τάξεων και χρησιμοποιείται ως σύνοψη της καμπύλης ROC. Όσο υψηλότερη είναι η AUC, τόσο καλύτερη είναι η απόδοση του μοντέλου στη διάκριση μεταξύ θετικών και αρνητικών τάξεων. Η τιμή AUC κυμαίνεται από 0 έως 1. Ένα μοντέλο του οποίου οι προβλέψεις είναι 100% λάθος έχει AUC 0,0. κάποιος του οποίου οι προβλέψεις είναι 100% σωστές έχει AUC 1,0.

Η μετρική **FPR(False Positive Rate)** δείχνει το ποσοστό των αληθινών ειδήσεων που διαγνώστηκαν ως ψευδείς. Αντίστοιχα η μετρική **FNR(False Negative Rate)** δείχνει το ποσοστό των ψευδών ειδήσεων που δε κατάφεραν να αναγνωριστούν.

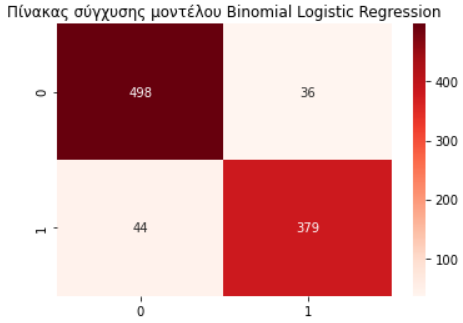
$$FNR = \frac{FN}{FN + TP}$$

3.6.1 Μετρικές απόδοσης των Μοντέλων Ταξινόμησης που εκπαιδεύτηκαν για το πείραμα

Binomial Logistic Regression

Accuracy: 91.64
Mean Squared Error: 0.29
AUC: 0.97
Στην Εικόνα 3.17 παρουσιάζονται ο πίνακας σύγχυσης και μετρικές.

	precision	recall	f1-score	suppor
0	0.92	0.93	0.93	53
1	0.91	0.90	0.90	42
accuracy			0.92	95
macro avg	0.92	0.91	0.92	95
weighted avg	0.92	0.92	0.92	95

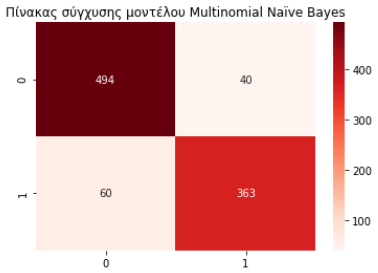


ΕΙΚΌΝΑ 3.17-ΑΠΟΤΕΛΕΣΜΑΤΑ BINOMIAL LOGISTIC REGRESSION

Multinomial Naïve Bayes

Accuracy: 89.55
Mean Squared Error: 0.32
AUC: 0.95
Στην Εικόνα 3.18 παρουσιάζονται ο πίνακας σύγχυσης και μετρικές.

	precision	recall	f1-score	support
0	0.89	0.93	0.91	534
1	0.90	0.86	0.88	423
accuracy			0.90	957
macro avg	0.90	0.89	0.89	957
weighted avg	0.90	0.90	0.90	957



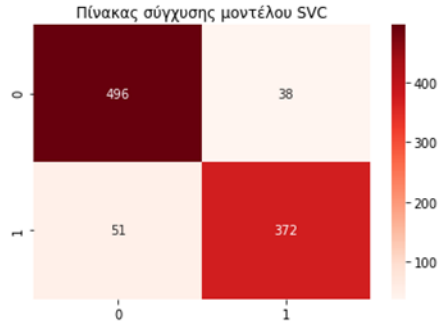
ΕΙΚΌΝΑ 3.18-ΑΠΟΤΕΛΕΣΜΑΤΑ NAIVE BAYES

SVC

Accuracy: 90.7
Mean Squared Error: 0.3
AUC: 0.9

Στην Εικόνα 3.19 παρουσιάζονται ο πίνακας σύγχυσης και μετρικές.

	precision	recall	f1-score	support
0	0.91	0.93	0.92	534
1	0.91	0.88	0.89	423
accuracy			0.91	957
macro avg	0.91	0.90	0.91	957
weighted avg	0.91	0.91	0.91	957



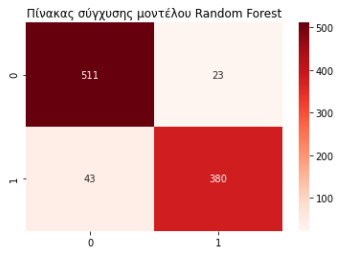
ΕΙΚΌΝΑ 3.19-ΑΠΟΤΕΛΕΣΜΑΤΑ SVC

Random Forest

Accuracy: 93.10
Mean Squared Error: 0.26
AUC: 0.98

Στην Εικόνα 3.20 παρουσιάζονται ο πίνακας σύγχυσης και μετρικές.

	precision	recall	f1-score	support
0	0.92	0.96	0.94	534
1	0.94	0.90	0.92	423
accuracy			0.93	957
macro avg	0.93	0.93	0.93	957
weighted avg	0.93	0.93	0.93	957



ΕΙΚΌΝΑ 3.20-ΑΠΟΤΕΛΕΣΜΑΤΑ RANDOM FOREST

4 Αποτελέσματα Πειράματος

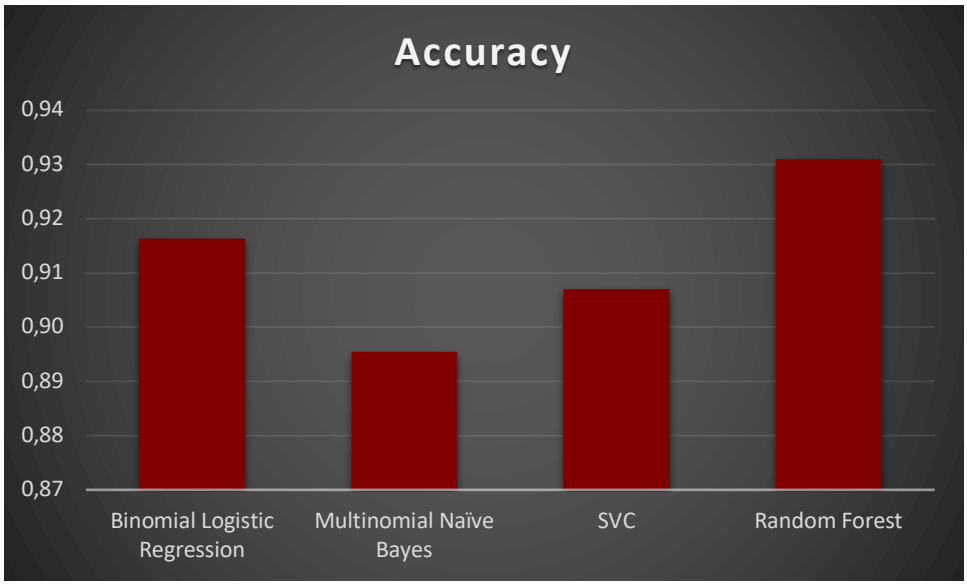
Στον πίνακα που απεικονίζεται στην **Εικόνα 4.1** παρουσιάζονται τα συγκεντρωτικά αποτελέσματα των μετρήσεων απόδοσης που έγιναν στα μοντέλα μηχανικής μάθησης.

	Binomial Logistic Regression	Multinomial Naïve Bayes	SVC	Random Forest
Accuracy	91.64%	89.55%	90.70%	93.10%
Mean Squared Error	0.29	0.32	0.30	0.26
TP	379	363	372	380
TN	498	494	496	511
FP	36	40	38	23
FN	44	60	51	43
Precision	0.91	0.90	0.91	0.94
Recall	0.90	0.86	0.88	0.90
F1-Score	0.90	0.88	0.89	0.92
FNR	0.10	0.14	0.12	0.10
FPR	0.07	0.07	0.07	0.04
AUC	0.97	0.95	0.90	0.98

ΕΙΚΟΝΑ 4.1-ΣΥΓΚΕΝΤΡΩΤΙΚΟΣ ΠΙΝΑΚΑΣ

4.1 Αποτελέσματα Accuracy

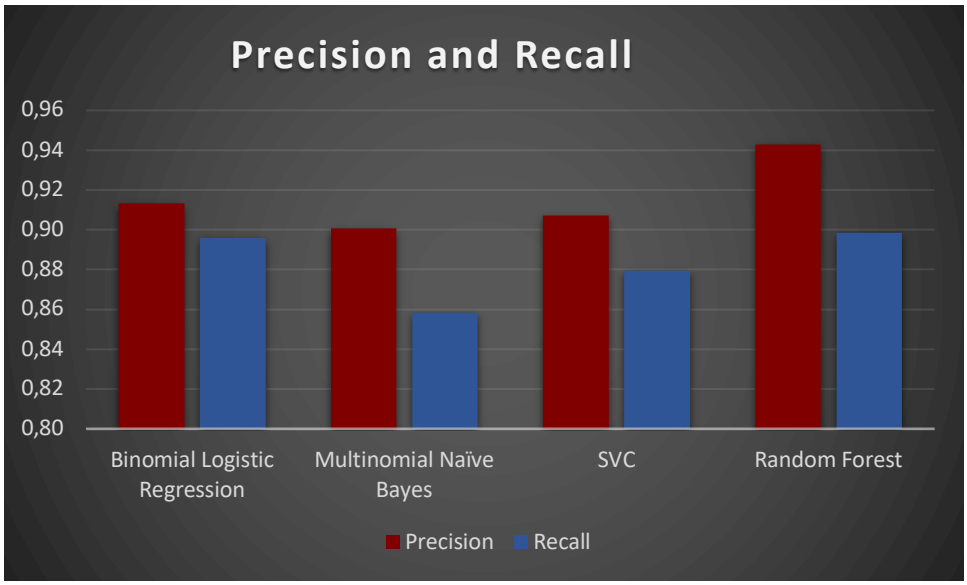
Στο διάγραμμα που απεικονίζεται στην **Εικόνα 4.2** συγκρίνεται η μετρική Accuracy των ταξινομητών. Παρατηρούμε ότι τη καλύτερη απόδοση στη γενικότερη μετρική Accuracy την έχει το μοντέλο Random Forest με ποσοστό σωστής αναγνώρισης 93.10%. Ακολουθεί το μοντέλο Binomial Logistic Regression με ποσοστό 91.64%. Αξίζει να σημειωθεί ότι όλοι οι ταξινομητές είχαν απόδοση πάνω από 80%.



ΕΙΚΟΝΑ 4.2-ACCURACY

4.2 Αποτελέσματα Precision και Recall

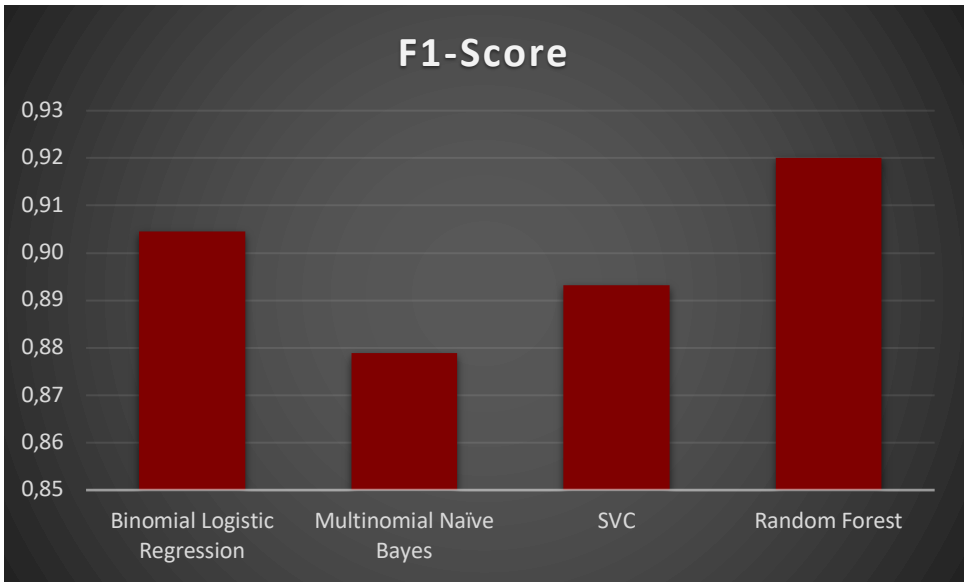
Στο διάγραμμα που απεικονίζεται στην **Εικόνα 4.3** συγκρίνονται οι μετρικές Precision και Recall της απόδοσης των ταξινομητών. Παρατηρούμε ότι με βάση τα αποτελέσματα της Precision και της Recall ότι ο Random Forest είναι πιο αξιόπιστος στην διάγνωση των ψευδών ειδήσεων. Ακολουθεί το μοντέλο Binomial Logistic Regression που έχει ίδια Precision με τον SVC, αλλά υψηλότερο Recall. Η απόδοση όλων των ταξινομητών με βάση τις μετρικές Precision και Recall είναι υψηλή.



ΕΙΚΌΝΑ 4.3-PRECISION AND RECALL

4.3 Αποτελέσματα F1 Score

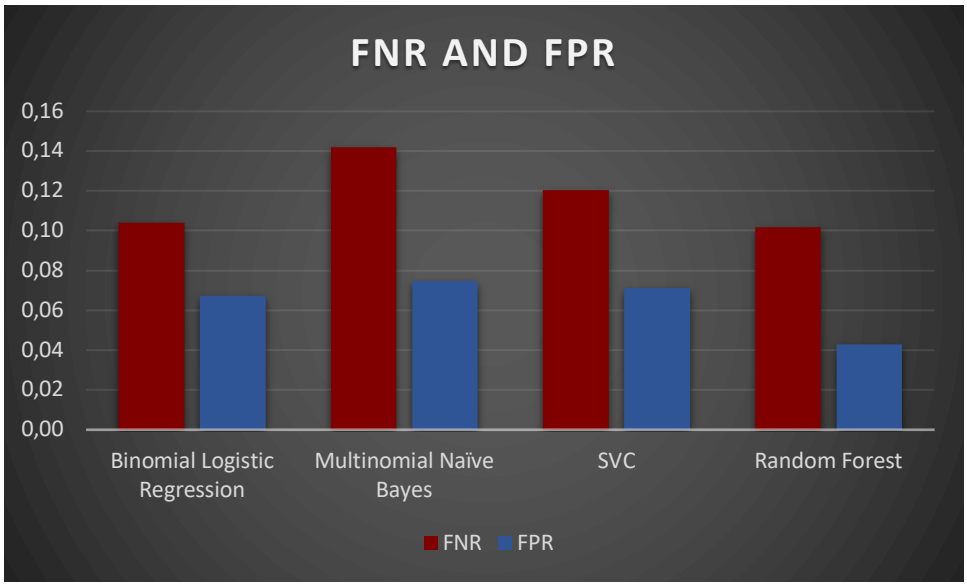
Στο διάγραμμα που απεικονίζεται στην **Εικόνα 4.4** συγκρίνεται η μετρική F1-Score της απόδοσης των ταξινομητών. Παρατηρούμε ότι με βάση τα αποτελέσματα της F1-Score Random Forest είναι πιο αξιόπιστος στην διάγνωση των ψευδών ειδήσεων.



ΕΙΚΟΝΑ 4.4-F1 SCORE

4.4 Αποτελέσματα FNR και FPR

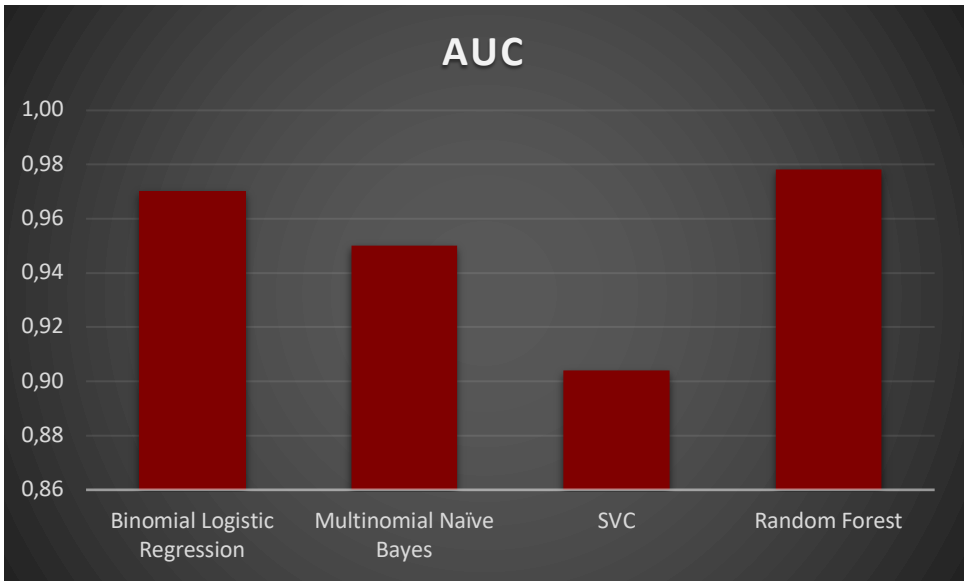
Στο διάγραμμα που απεικονίζεται στην **Εικόνα 4.5** συγκρίνονται οι μετρικές FNR και FPR της απόδοσης των ταξινομητών. Όσο πιο μικρές είναι οι τιμές τόσο καλύτερη απόδοση έχει ο αλγόριθμος. Με βάση τα αποτελέσματα της FNR ο Random Forest και το μοντέλο Binomial Logistic Regression έχουν την ίδια απόδοση, όμως ό Random Forest έχει καλύτερη απόδοση με βάση την FPR.



ΕΙΚΌΝΑ 4.5-FNR AND FPR

4.5 Αποτελέσματα AUC

Στο διάγραμμα που απεικονίζεται στην **Εικόνα 4.6** συγκρίνεται η μετρική AUC της απόδοσης των ταξινομητών. Με βάση τα αποτελέσματά της, ο Random Forest έχει καλύτερη απόδοση, ενώ ακολουθεί το μοντέλο Binomial Logistic Regression



ΕΙΚΌΝΑ 4.6-AUC

5 Συμπεράσματα

5.1 Σύνοψη και Συμπεράσματα

Στη παρούσα διπλωματική εργασία προτείνεται μια προσέγγιση μηχανικής μάθησης για τη αυτόματη ανίχνευση ψευδών ειδήσεων που αφορούν τον COVID-19. Τα δεδομένα συλλέχθηκαν, μέσω αλγορίθμων που αναπτύχθηκαν με τη γλώσσα προγραμματισμού Python, από έγκυρη ειδησεογραφική πηγή και από πηγές που έχουν χαρακτηριστεί ως αναξιόπιστες. Εφαρμόστηκε επιβλεπόμενη μάθηση καθώς η πληροφορία για την κατηγοριοποίηση των δεδομένων υπήρχε από πριν.

Στα δεδομένα εφαρμόστηκαν τεχνικές Εξόρυξης Γνώσης από Κείμενα και ακολούθησε Διερευνητική Ανάλυση και Ανάλυση Συναισθήματος. Μια δυσκολία που αντιμετωπίσαμε σε αυτό το σημείο ήταν η εύρεση ενός λεξικού συναισθημάτων της ελληνικής γλώσσας ώστε να προχωρήσουμε στην Ανάλυση Συναισθήματος. Το λεξικό που χρησιμοποιήθηκε αποτελεί μια καλή βάση, αλλά τα λήμματα που περιέχει είναι αρκετά περιορισμένα ώστε να μπορούμε να καταλήξουμε σε ολοκληρωμένα και ασφαλή συμπεράσματα. Ολοκληρώνοντας τη διαδικασία της ανάλυσης επιλέχθηκε η συχνότητα εμφάνισης των λέξεων και η μέθοδος TF-IDF ως κατάλληλο χαρακτηριστικό για την υλοποίηση των μοντέλων Μηχανικής Μάθησης.

Υλοποιήθηκαν τέσσερα μοντέλα επιβλεπόμενης μηχανικής μάθησης. Το μοντέλο Binomial Logistic Regression, το μοντέλο Multinomial Naïve Bayes, το μοντέλο Support Vector Classifier(SVC) και το μοντέλο Random Forest. Η απόδοση των μοντέλων ελέγχθηκε συνδυαστικά από μια σειρά μετρικών. Το μοντέλο που μας έδωσε τα καλύτερα αποτελέσματα είναι το μοντέλο Random Forest.

Τα αποτελέσματα του πειράματος μας δείχνουν ότι υπάρχει δυνατότητα ανίχνευσης των ψευδών ειδήσεων που αφορούν τον COVID-19.

5.2 Προοπτικές για μελλοντική έρευνα

Στην εργασία αυτή η υλοποίηση των μοντέλων έγινε με βάση ένα χαρακτηριστικό. Θα ήταν ενδιαφέρον να γίνει υλοποίηση των μοντέλων και μέτρηση της απόδοσης τους και με άλλα χαρακτηριστικά, όπως το περιεχόμενο των τίτλων των άρθρων. Μια άλλη πλευρά είναι η εξέταση της απόδοσης των αλγορίθμων με βάση νέα δεδομένα, τα οποία να προέρχονται από διαφορετικές πηγές από

αυτές που έχουν χρησιμοποιηθεί. Τέλος θα μπορούσε να γίνει υλοποίηση και άλλων μοντέλων μηχανικής μάθησης ώστε να αναζητήσουμε ένα ακόμα πιο αποδοτικό μοντέλο.

ΠΑΡΑΡΤΗΜΑ Ι
ΚΩΔΙΚΑΣ

1. Συλλογή Δεδομένων

Για τη συλλογή δεδομένων ανατρέξαμε στο αρχείο της κάθε ιστοσελίδας από την οποία κατεβάσαμε τα άρθρα. Τα δεδομένα αποθηκεύτηκαν σε Βάση Δεδομένων με τη χρήση της SQLite. Στις Εικόνες παρατίθενται ενδεικτικά παραδείγματα των αλγόριθμων που χρησιμοποιήθηκαν

```
import requests
from requests import get
from bs4 import BeautifulSoup
import pandas as pd
import numpy as np
import sqlite3
from newspaper import Article
import feedparser as fp
from datetime import datetime
```

```
def Connect_DB():
    global cur,conn
    try:
        conn = sqlite3.connect('News.db')
        cur = conn.cursor()
    except sqlite3.Error as error:
        print("Error while connecting to sqlite", error)
```

```
def Close_DB(conn,cur):
    try:
        conn.commit()
        cur.close()
    except sqlite3.Error as error:
        print("Error with sqlite", error)
```

```
Connect_DB()

sql_query="""CREATE TABLE IF NOT EXISTS NewsTable (
            NewsId INTEGER PRIMARY KEY AUTOINCREMENT,
            Site VARCHAR,
            URL VARCHAR,
            Title VARCHAR,
            Article vARCHAR,
            Date DATE,
            Fake INTEGER)
            """
cur.execute(sql_query)


Close_DB(conn,cur)
```

```
#Insert Articles into NewsTable
def Insert_articles(cur,site, url,Fake):
    try:
        article = Article(url)
        article.download()
        article.parse()
        if article.publish_date is not None:
            cur.execute("""INSERT INTO NewsTable (Site, Url, Title, Article, Date, Fake)
                        VALUES(?,?,?,?,?,?)""",(site, url, article.title, article.text, article.publish_date, Fake))
    except  Exception as e:
        print(e)
        print("continuing...")
        pass
```

```
def getURL(page, find):
    start_link = page.find(find)
    if start_link == -1:
        return None, 0
    start_quote = page.find('"', start_link)
    end_quote = page.find('"', start_quote + 1)
    url = page[start_quote + 1: end_quote]
    return url, end_quote
```

```
#Παράδειγμα αναζήτησης άρθρων
Connect_DB()

mlist=[]
site,Fake="mysite",0 #Fake=0 για True News, Fake=1 για Fake News
for i in range(1,32):# Διαφοροποιείται ανάλογα τον τρόπο που αποθηκεύει κάθε σελίδα τα άρθρα
    for j in range (1,12):# Διαφοροποιείται ανάλογα τον τρόπο που αποθηκεύει κάθε σελίδα τα άρθρα
        try:
            url = "mysite/2020/%s/%s/politics" % (str(j).zfill(2),str(i).zfill(2))
            results = requests.get(url)
            page = str(BeautifulSoup(results.text, "html.parser"))# Διαφοροποιείται ανάλογα τον τρόπο που αποθηκεύει κάθε σελίδα
            while True:
                url2, n = getURL(page,'href')
                page = page[n:]
                if url2:
                    if url2.startswith(url):# Διαφοροποιείται ανάλογα τον τρόπο που αποθηκεύει κάθε σελίδα τα άρθρα
                        if url2 in mlist:
                            break
                        else:
                            mlist.extend([url2])
                            Insert_articles(cur,site,url2,Fake)
                else:
                    break
        except Exception as e:
            print(e)
            print("Error in page :",str(j), " ", url, " continuing...")
            pass
Close_DB(conn,cur)
```

2. Καθαρισμός και προεπεξεργασία δεδομένων

```
import pandas as pd
import numpy as np
import sqlite3
import re
import string
```

```
try:
    conn = sqlite3.connect('News.db')
    cur = conn.cursor()
except sqlite3.Error as error:
    print("Error while connecting to sqlite", error)
```

```
SQL_Query = pd.read_sql_query("""Select Site,Title,Article,Date,Fake
                                 From NewsTable

                              """,conn)

df = pd.DataFrame(SQL_Query)

conn.commit()
cur.close()
```

```
#Ημερομηνία σε συμβατή μορφή
News_DF["Date"]= pd.to_datetime(News_DF["Date"],utc=True,errors = 'coerce').dt.date
```

```
News_DF['Fake']=News_DF['Fake'].astype(int)
```

```
#Απαλοιφή διπλοεγγραφών
News_DF.drop_duplicates(subset ="Title",
                     keep = False, inplace = True)
```

```
for column in ['Title','Article']:
    News_DF[column] = News_DF[column].replace(to_replace=r'https?:\/\/.*[\r\n]*',value='',regex=True)#Απαλοιφή url
    News_DF[column] = News_DF[column].replace(to_replace=r'www.*[\r\n]*',value='',regex=True)
    News_DF[column]=News_DF[column].str.lower() # Μετατροπή κεφαλαίων σε μικρά
    News_DF[column].replace(r'\s+|\\n', ' ', regex=True, inplace=True) # Απαλοιφή ειδικών χαρακτήρων
    News_DF[column] = News_DF[column].apply(lambda x:''.join([i for i in x
                                            if i not in string.punctuation+"«»"]))# Απαλοιφή σημείων στίξης
    News_DF[column] = News_DF[column].str.rstrip(string.digits) #Απαλοιφή αριθμών
    News_DF[column] = News_DF[column].str.replace('\d+', '')
    News_DF[column]=News_DF[column].str.replace("ά","α") #Απαλοιφή τόνων
    News_DF[column]=News_DF[column].str.replace("έ","ε")
    News_DF[column]=News_DF[column].str.replace("ή","η")
    News_DF[column]=News_DF[column].str.replace("ί","ι")
    News_DF[column]=News_DF[column].str.replace("ό","ο")
    News_DF[column]=News_DF[column].str.replace("ύ","υ")
    News_DF[column]=News_DF[column].str.replace("ώ","ω")
    News_DF[column]=News_DF[column].str.replace("ΐ","ι")
```

```
#Δημιουργία dataset με ειδήσεις για τον covid
Covid_News=News_DF[News_DF["Article"].str.contains("""lockdown|cov|sars|καραντινα|κορονοιο|
                                                    κορωνοιο|κοροναιο|κορωναιο|τηλεκπαιδευσ|
                                                    pfizer|κρουσμ|εοδυ|τηλεργασ|13033|πανδημια|
                                                    υδροχλωροκινη|PCR|Rapid|λοκνταουν|πφαιζερ""")]
```

```
#Δημιουργία dataset με ειδήσεις για τον covid
Covid_News=News_DF[News_DF["Article"].str.contains("""lockdown|cov|sars|καραντινα|κορονοιο|
                                                     κορωνοιο|κοροναιο|κορωναιο|τηλεκπαιδευσ|
                                                     pfizer|κρουσμ|εοδυ|τηλεργασ|13033|πανδημια|
                                                     υδροχλωροκινη|PCR|Rapid|λοκνταουν|πφαιζερ""")]
```

```
Covid_News=Covid_News[(Covid_News['Date'] > pd.to_datetime('2019-12-31')) &(Covid_News['Date'] < pd.to_datetime('2020-11-18'))]
Covid_News.head()
```

	Site	Title	Article	Date	Fake
38	T1	αναβαλλεται η συνδιασκεψη του κιναλ εξαιτιας τ...	την αναβολη των προγραμματισμενων εκδηλωσεων γ...	2020-03-01	0
47	T1	οι απειθαρχοι του κοροναιου και τα προστιμα	σημερα θα αναφερθω στους απειθαρχους που μολις...	2020-04-01	0
48	T1	μητσοτακης στο cnn για κοροναιο δωσαμε τον λογ...	η ταση στην ελλαδα ειναι παρα πολυ καλη σχετικ...	2020-04-01	0
50	T1	κοροναιος επιχειρησιακο σχεδιο για την αποφυγη...	την αναγκη εκπονησης ολοκληρωμενου επιχειρησια...	2020-04-01	0
51	T1	μητσοτακης παρακολουθουμε την υγεια μας παραμε...	παρακολουθουμε την υγεια μας παραμενουμε ενημε...	2020-04-01	0

```
Covid_News.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 4715 entries, 38 to 19336
Data columns (total 5 columns):
 #   Column   Non-Null Count  Dtype 
---  ------   --------------  ----- 
 0   Site     4715 non-null   object
 1   Title    4715 non-null   object
 2   Article  4715 non-null   object
 3   Date     4715 non-null   object
 4   Fake     4715 non-null   int32 
dtypes: int32(1), object(4)
memory usage: 202.6+ KB
```

```
#Καταμέτρηση Fake και True News για τον covid για το έτος 2020(έως 18/11)
Covid_News.groupby('Fake').count()['Title']
```

```
Fake
0    2664
1    2051
Name: Title, dtype: int64
```

```
#Αποθήκευση Dataframe με άρθρα για τον covid στη βάση δεδομένων
try:
    conn = sqlite3.connect('News.db')
    cur = conn.cursor()
except sqlite3.Error as error:
    print("Error while connecting to sqlite", error)
cur.execute("DROP TABLE IF EXISTS Covid_News")
Covid_News.to_sql(name='Covid_News', con=conn)
conn.commit()
cur.close()
```

3. Διερευνητική Ανάλυση

```
import pandas as pd
import numpy as np
import sqlite3
import re
import string
import nltk
from nltk.stem import WordNetLemmatizer
from nltk.tokenize import word_tokenize
from nltk.corpus import stopwords
from nltk.util import ngrams
from nltk.stem import WordNetLemmatizer
lemmatizer = WordNetLemmatizer()
```

```
def Connect_DB():
    global conn, cur
    try:
        conn = sqlite3.connect('News.db')
        cur = conn.cursor()
    except sqlite3.Error as error:
        print("Error while connecting to sqlite", error)
```

```
Connect_DB()
```

```
SQL_Query = pd.read_sql_query("""Select *
                                 From Covid_News

                              """,conn)

Covid_News = pd.DataFrame(SQL_Query)

conn.commit()
cur.close()
Covid_News=Covid_News[['Site','Title','Article','Date','Fake']]
```

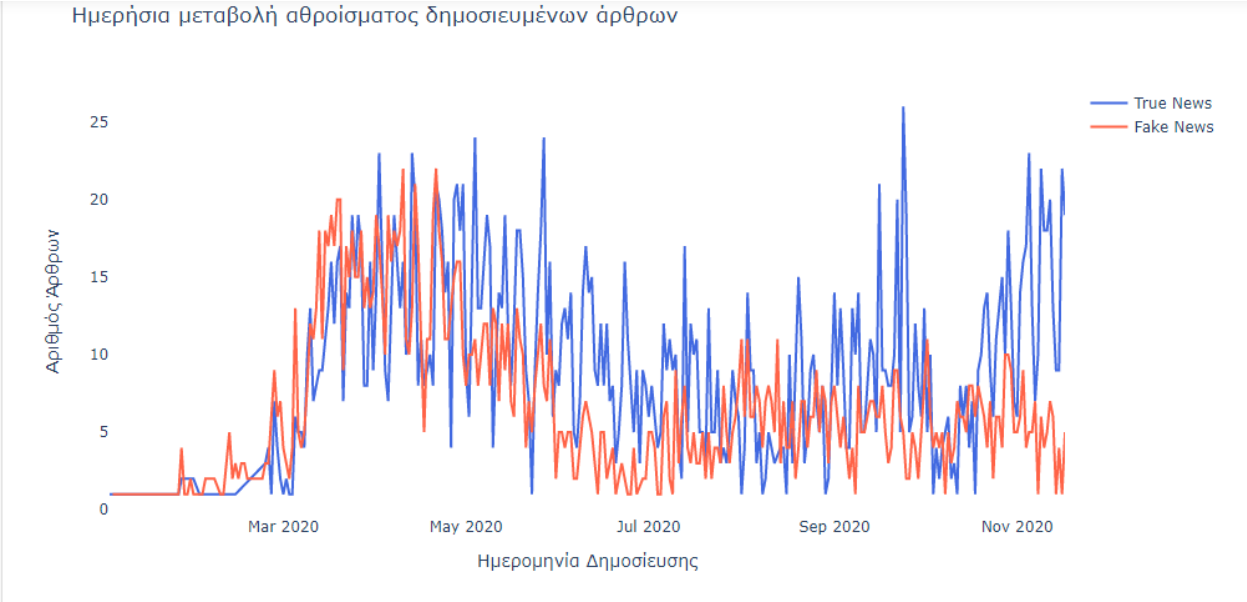
```
#Μορφοποίηση ημερομηνίας σε συμβατή μορφή
Covid_News["Date"]= pd.to_datetime(Covid_News["Date"],utc=True,errors = 'coerce').dt.date
```

```
#Δημιουργία στηλών με των αριθμό των λέξεων, ngrams, λίστα με λέξεις
nltk.download('punkt')
nltk.download('wordnet')
nltk.download('stopwords')
stop_words = stopwords.words('greek')
stop_words.extend(['γι','απ','στα','ολο','στους','μονο','μεχρι','ακομα','αυτος','αυτες','κανουν','υπαρχει','ηδη',
                   'ολους','συμφωνα','επισης','οποιος','στιγμη','ειτε','εχεις','γινει','παντα','εως','δηλαδη',
                   'ακομη','πανω','μαλιστα','εδω','μεταξυ','λοιπον','αλλη','τοσο','κατι','λογω','οποιο','τωρα',
                   'επειδη','πολυ','ολα','εν','αυτον','δυο','καθε','ακομα','καθως','σαν','ενας','κανει','αφου','πλεον',
                   'μεσω','εις','εναν','δια','οποια','ουτε','προς','ετσι','ολοι','πριν','μπορει','μεσα','σου','σας','μου',
                   'ομως','οπου','πρεπει','οταν','http','ενα','οπως','γιατι','ειπε','ωστε','ηταν','από','της','τους','τη',
                   'του','μας','έχει','έχουν','μια','εχει','στις','ομως','ειχε','εχουν','ως','εχουμε','πιο'])

for c in ['Article','Title']:
    Covid_News[c+'_Words_Count']=Covid_News[c].apply(lambda x: len(x.split()))
    Covid_News['Words_in_'+c]=Covid_News[c].apply(lambda x: word_tokenize(x))
    Covid_News['Words_in_'+c]=Covid_News['Words_in_'+c].apply(lambda x: [item for item in x if item not in stop_words])
    Covid_News['Words_in_'+c]=Covid_News['Words_in_'+c].apply(lambda x: [item for item in x if item not in string.punctuation])
    Covid_News['Words_in_'+c]=Covid_News['Words_in_'+c].apply(lambda x: [item for item in x if item.isalpha()])
    Covid_News[c+'_bigrams'] = Covid_News['Words_in_'+c].apply(lambda row: list(nltk.ngrams(row, 2)))
    Covid_News[c+'_trigrams'] = Covid_News['Words_in_'+c].apply(lambda row: list(nltk.ngrams(row, 3)))
```

Δημιουργία δύο στηλών “Article_Words_Count” και “ Title_Words_Count ” για την καταμέτρηση όλων των λέξεων που εμπεριέχονται στο κάθε άρθρο και στον κάθε τίτλο αντίστοιχα. Δημιουργήθηκαν δύο στήλες “Words_in_Article” και “Words_in_Title” οι οποίες περιέχουν για το κάθε άρθρο ή τίτλο αντίστοιχα μια λίστα με το σύνολο των λέξεων από τις οποίες έχουμε αφαιρέσει stopwords (λέξεις που χρησιμοποιούνται ευρέως αλλά δε συνεισφέρουν στον προσδιορισμό του κειμένου ως αληθή ή ψευδή είδηση). Τέλος δημιουργήθηκαν στήλες για τα bigrams και τα trigrams των τίτλων και των άρθρων.

```
#Δημιουργία Dataframe με το άθροισμα των true news ανά ημέρα το έτος 2020
Covid_true_count = Covid_News[Covid_News['Fake']==0].groupby("Date").count().reset_index()
Covid_true_count=Covid_true_count[['Date','Title']].rename(columns={'Title': 'Articles_Count'})
mask = ((Covid_true_count['Date'] > pd.to_datetime('2019-12-31')) & (Covid_true_count['Date'] < pd.to_datetime('2020-11-18')))
Covid_true_count=Covid_true_count.loc[mask]

#Δημιουργία Dataframe με το άθροισμα των fake news ανά ημέρα το έτος 2020
Covid_fake_count = Covid_News[Covid_News['Fake']==1].groupby("Date").count().reset_index()
Covid_fake_count=Covid_fake_count[['Date','Title']].rename(columns={'Title': 'Articles_Count'})
mask = ((Covid_fake_count['Date'] > pd.to_datetime('2019-12-31')) & (Covid_fake_count['Date'] < pd.to_datetime('2020-11-18')))
Covid_fake_count=Covid_fake_count.loc[mask]
```

```
#Απεικόνιση Ημερήσιας μεταβολής αρθοίσματος δημοσιευμένων άρθρων για τον Covid
from plotly.graph_objs import *
import plotly.graph_objects as go

layout = Layout(
    title='Ημερήσια μεταβολή αθροίσματος δημοσιευμένων άρθρων',
    xaxis = go.layout.XAxis(title = go.layout.xaxis.Title(text='Ημερομηνία Δημοσίευσης')),
    yaxis = go.layout.YAxis(title = go.layout.yaxis.Title(text='Αριθμός Άρθρων')),
    paper_bgcolor='rgba(0,0,0,0)',
    plot_bgcolor='rgba(0,0,0,0)')

fig = go.Figure(layout=layout)
fig.add_scatter(x=Covid_true_count["Date"], y=Covid_true_count['Articles_Count'],mode='lines', line_color='royalblue',
                name = "True News")
fig.add_scatter(x=Covid_fake_count["Date"], y=Covid_fake_count['Articles_Count'],mode='lines', line_color='tomato',
                name = "Fake News")
fig.show()
```



```
#Δημιουργία dataframe για τα true news και dataframe για τα fake news
Covid_true = Covid_News[(Covid_News['Fake']==0)&(Covid_News['Date'] > pd.to_datetime('2019-12-31')) &
                        (Covid_News['Date'] < pd.to_datetime('2020-11-18'))]
Covid_fake = Covid_News[(Covid_News['Fake']==1)&(Covid_News['Date'] > pd.to_datetime('2019-12-31')) &
                        (Covid_News['Date'] < pd.to_datetime('2020-11-18'))]
Covid_fake.reset_index(drop=True, inplace=True)
Covid_true.reset_index(drop=True, inplace=True)
```

```
print('Αριθμός άρθρων από αξιόπιστη πηγή: ', Covid_true.shape[0],
      '\nΑριθμός άρθρων από πιθανώς αξιόπιστες πηγές: ',Covid_fake.shape[0])
```

```
Αριθμός άρθρων από αξιόπιστη πηγή:  2664
Αριθμός άρθρων από πιθανώς αξιόπιστες πηγές:  2051
```

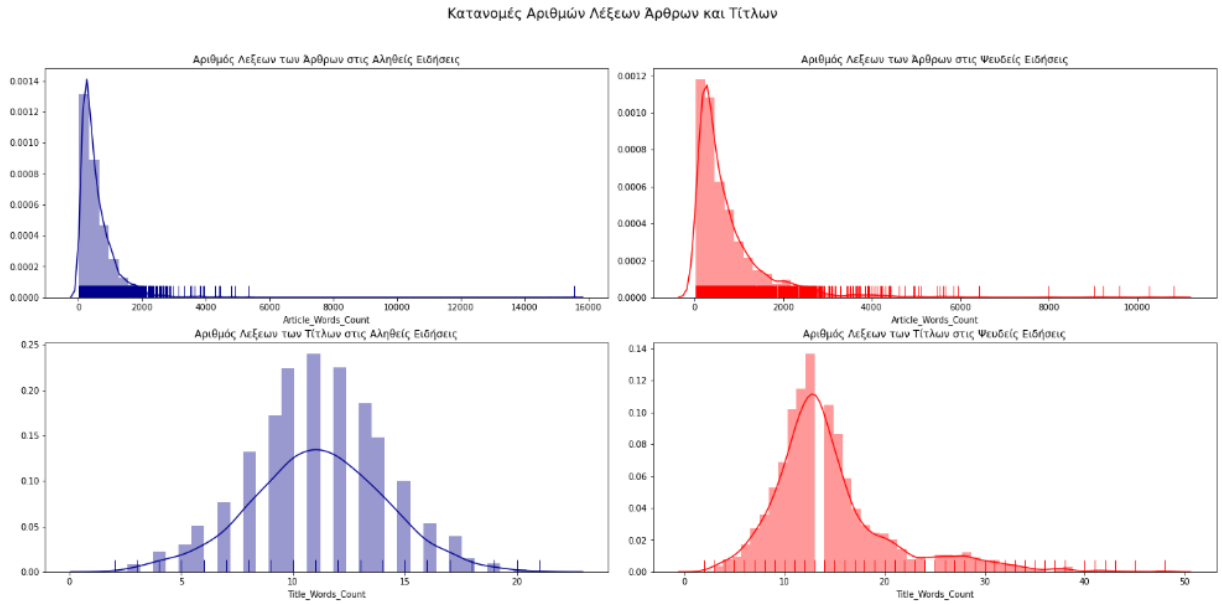
```
print('Fake_News/True_News=',Covid_fake.shape[0]/Covid_true.shape[0])
```

```
Fake_News/True_News= 0.7698948948948949
```

```
#Δημιουργία dataframe με τα στατιστικά των λέξεων ανα τίτλο και άρθρο
Word_Statistics=Covid_true.describe()[['Article_Words_Count',
                                       'Title_Words_Count']].join(Covid_fake.describe()[['Article_Words_Count',
                                                                                         'Title_Words_Count']],
                                                                                        lsuffix='_True', rsuffix='_Fake')
Word_Statistics=Word_Statistics[['Article_Words_Count_True','Article_Words_Count_Fake',
                                 'Title_Words_Count_True','Title_Words_Count_Fake']]
Word_Statistics.round(2)
```

	Article_Words_Count_True	Article_Words_Count_Fake	Title_Words_Count_True	Title_Words_Count_Fake
count	2664.00	2051.00	2664.00	2051.00
mean	596.83	790.04	11.13	14.59
std	617.19	946.36	3.00	6.15
min	31.00	27.00	2.00	2.00
25%	232.00	236.00	9.00	11.00
50%	422.00	473.00	11.00	13.00
75%	777.25	981.00	13.00	16.00
max	15559.00	10815.00	21.00	48.00

```
import seaborn as sns
import matplotlib.pyplot as plt
%matplotlib inline

fig, ax = plt.subplots(2, 2,figsize=(20, 10))
fig.suptitle('Κατανομές Αριθμών Λέξεων Άρθρων και Τίτλων', fontsize=16)
sns.distplot(Covid_true['Article_Words_Count'], ax = ax[0,0],
             color = "darkblue", rug = True).set_title("Αριθμός Λεξεων των Άρθρων στις Αληθείς Ειδήσεις")
sns.distplot(Covid_fake['Article_Words_Count'], ax = ax[0,1],
             color = "red", rug = True).set_title("Αριθμός Λεξεων των Άρθρων στις Ψευδείς Ειδήσεις")
sns.distplot(Covid_true['Title_Words_Count'], ax = ax[1,0],
             color = "darkblue", rug = True).set_title("Αριθμός Λεξεων των Τίτλων στις Αληθείς Ειδήσεις")
sns.distplot(Covid_fake['Title_Words_Count'], ax = ax[1,1],
             color = "red", rug = True).set_title("Αριθμός Λεξεων των Τίτλων στις Ψευδείς Ειδήσεις")
fig.tight_layout()
fig.subplots_adjust(top=0.88)
```




```
# Δημιουργία ngramms
from itertools import chain
from collections import Counter
from pandas import DataFrame
Covid_true.name='True News'
Covid_fake.name='Fake News'

for df in [Covid_true,Covid_fake]:
    bigrams = df['Article_bigrams'].tolist()
    bigrams = list(chain(*bigrams))
    bigrams = [(x.lower(), y.lower()) for x,y in bigrams]

    bigram_counts = Counter(bigrams)
    print('\n',df.name,'\n',DataFrame(bigram_counts.most_common(15), columns=['bidgram','frequency']))
```

```
for df in [Covid_true,Covid_fake]:
    trigrams = df['Article_trigrams'].tolist()
    trigrams = list(chain(*trigrams))
    trigrams = [(x.lower(), y.lower(),z.lower()) for x,y,z in trigrams]

    trigram_counts = Counter(trigrams)
    print('\n',df.name,'\n',DataFrame(trigram_counts.most_common(15), columns=['tridgram','frequency']))
```

True News

	bidgram	frequency
0	(κυριακος, μητσοτακης)	1030
1	(κυβερνητικος, εκπροσωπος)	566
2	(συστημα, υγειας)	494
3	(αλεξης, τσιπρας)	492
4	(δημοσιας, υγειας)	470
5	(ανατολικη, μεσογειο)	435
6	(πρωθυπουργος, κυριακος)	403
7	(στελιος, πετσας)	382
8	(αξιωματικης, αντιπολιτευσης)	358
9	(αλεξη, τσιπρα)	317
10	(φωφη, γεννηματα)	309
11	(οσον, αφορα)	307
12	(υπουργος, εξωτερικων)	303
13	(συστηματος, υγειας)	301
14	(κυριακου, μητσοτακη)	278

True News

	tridgram	frequency
0	(πρωθυπουργος, κυριακος, μητσοτακης)	402
1	(κυβερνητικος, εκπροσωπος, στελιος)	221
2	(εκπροσωπος, στελιος, πετσας)	221
3	(εθνικο, συστημα, υγειας)	161
4	(δημοσιο, συστημα, υγειας)	127
5	(πρωθυπουργο, κυριακο, μητσοτακη)	125
6	(συριζα, προοδευτικη, συμμαχια)	121
7	(προστασια, δημοσιας, υγειας)	119
8	(αρχηγος, αξιωματικης, αντιπολιτευσης)	105
9	(δημοσιου, συστηματος, υγειας)	103
10	(πρωθυπουργου, κυριακου, μητσοτακη)	102
11	(υγειας, βασιλης, κικιλιας)	99
12	(εθνικου, συστηματος, υγειας)	97
13	(υπουργος, υγειας, βασιλης)	95
14	(ενημερωση, πολιτικων, συντακτων)	90

Fake News

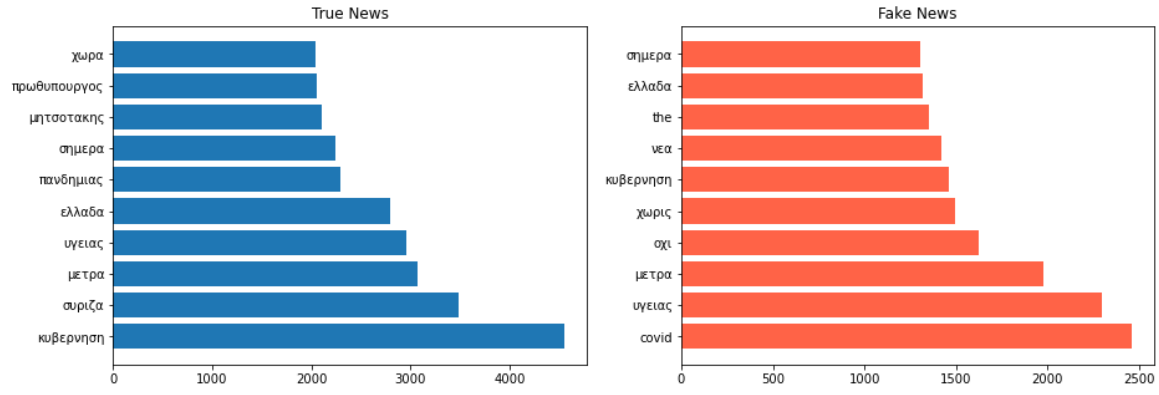
	bidgram	frequency
0	(χρηση, μασκας)	484
1	(δημοσιας, υγειας)	474
2	(οσον, αφορα)	242
3	(λιανικο, εμποριο)	232
4	(bill, gates)	220
5	(υπουργειου, υγειας)	219
6	(μπιλ, γκειτς)	209
7	(σωτηρης, τσιοδρας)	205
8	(σουπερ, μαρκετ)	201
9	(πρωτη, φορα)	199
10	(δημοσια, υγεια)	187
11	(ηνωμενες, πολιτειες)	180
12	(πολιτικης, προστασιας)	179
13	(χρονικο, διαστημα)	175
14	(κυριακος, μητσοτακης)	172

Fake News

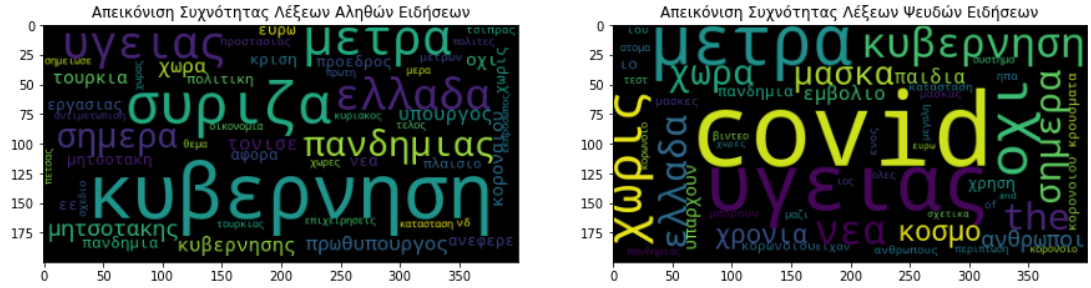
	tridgram	frequency
0	(υποχρεωτικη, χρηση, μασκας)	114
1	(παγκοσμιος, οργανισμος, υγειας)	77
2	(ηλεκτρονικου, τηλεφωνικου, εμπορίου)	69
3	(τηλεφωνικου, εμπορίου, παραδοση)	69
4	(εμπορίου, παραδοση, οικον)	69
5	(παραδοση, οικον, eshop)	69
6	(οικον, eshop, κτλ)	69
7	(υπηρεσιες, ηλεκτρονικου, τηλεφωνικου)	65
8	(eshop, κτλ, λιανικο)	62
9	(κτλ, λιανικο, εμποριο)	62
10	(υφυπουργος, πολιτικης, προστασιας)	61
11	(οσων, εργαζομενων, αφμ)	60
12	(εργαζομενων, αφμ, ληγει)	60
13	(εξαιρεση, υπηρεσιες, ηλεκτρονικου)	57
14	(παγκοσμιου, οργανισμου, υγειας)	56

```
#Απεικόνιση των 10 συχνότερων λέξεων για τα True και τα Fake News
True_Article_Word_Frequency = pd.DataFrame(Counter(chain.from_iterable(Covid_true['Words_in_Article'])).most_common(50),
                                           columns=['Word','Frequency'])
Fake_Article_Word_Frequency = pd.DataFrame(Counter(chain.from_iterable(Covid_fake['Words_in_Article'])).most_common(50),
                                           columns=['Word','Frequency'])
%matplotlib inline
fig, axes = plt.subplots(ncols=2, figsize=(15, 5))
axes[0].set_title('True News')
axes[1].set_title('Fake News')
axes[0].barh(True_Article_Word_Frequency.set_index('Word').reset_index()[:10]['Word'],
             True_Article_Word_Frequency.set_index('Word').reset_index()[:10]['Frequency'])
axes[1].barh(Fake_Article_Word_Frequency.set_index('Word').reset_index()[:10]['Word'],
             Fake_Article_Word_Frequency.set_index('Word').reset_index()[:10]['Frequency'], color = 'Tomato')
```

```
<BarContainer object of 10 artists>
```



```
# Συννεφόλεξο των 50 συχνότερων λέξεων για τα True και τα Fake News
import wordcloud
from wordcloud import WordCloud
t_wc = WordCloud(max_words=50,relative_scaling=1,background_color ='black',
                 normalize_plurals=False).generate_from_frequencies(True_Article_Word_Frequency.set_index('Word')['Frequency'])

f_wc = WordCloud(max_words=50,relative_scaling=1,background_color ='black',
                 normalize_plurals=False).generate_from_frequencies(Fake_Article_Word_Frequency.set_index('Word')['Frequency'])

fig, axs = plt.subplots(1,2, figsize=(15, 30))
axs[0].set_title('Απεικόνιση Συχνότητας Λέξεων Αληθών Ειδήσεων')
axs[0].imshow(t_wc)

axs[1].set_title('Απεικόνιση Συχνότητας Λέξεων Ψευδών Ειδήσεων')
axs[1].imshow(f_wc)
plt.show()
```




```
#Εύρεση του αριθμού των κοινών λέξεων για τις 50 συχνότερες λέξειω σε True και Fake News
print('Η τομή των 50 συχνότερων λέξεων σε Fake και True News είναι: ' ,
      len(set(Fake_Article_Word_Frequency['Word'].tolist()) & set(True_Article_Word_Frequency['Word'].tolist())), ' λέξεις')
```

```
Η τομή των 50 συχνότερων λέξεων σε Fake και True News είναι:  14  λέξεις
```

4. Ανάλυση Συναισθήματος

```
# Επεξεργασία λεξικού για να είναι συμβατό με τα άρθρα
Lex_DF = pd.read_csv("greek_sentiment_lexicon.tsv",sep='\t')

Lex_DF=Lex_DF[['Term',
               'Polarity1','Polarity2', 'Polarity3', 'Polarity4',
               'Anger1', 'Anger2', 'Anger3','Anger4',
               'Disgust1', 'Disgust2', 'Disgust3', 'Disgust4',
               'Fear1','Fear2', 'Fear3', 'Fear4',
               'Happiness1', 'Happiness2', 'Happiness3','Happiness4',
               'Sadness1', 'Sadness2', 'Sadness3', 'Sadness4',
               'Surprise1', 'Surprise2', 'Surprise3', 'Surprise4']]
Lex_DF['Term']=Lex_DF['Term'].str.lower() # Μετατροπή κεφαλαίων σε μικρά
Lex_DF['Term']=Lex_DF['Term'].str.replace("ά","α") #Απαλοιφή τόνων
Lex_DF['Term']=Lex_DF['Term'].str.replace("έ","ε")
Lex_DF['Term']=Lex_DF['Term'].str.replace("ή","η")
Lex_DF['Term']=Lex_DF['Term'].str.replace("ί","ι")
Lex_DF['Term']=Lex_DF['Term'].str.replace("ό","ο")
Lex_DF['Term']=Lex_DF['Term'].str.replace("ύ","υ")
Lex_DF['Term']=Lex_DF['Term'].str.replace("ώ","ω")
Lex_DF['Term']=Lex_DF['Term'].str.replace("ΐ","ι")

Lex_DF.head()
```

	Term	Polarity1	Polarity2	Polarity3	Polarity4	Anger1	Anger2	Anger3	Anger4	Disgust1	...	Happiness3	Happiness4	Sadness1	Sadness2	Sadne
0	αβαφτιστος	BOTH	NaN	BOTH	NaN	3.0	NaN	5.0	NaN	4.0	...	1.0	NaN	4.0	NaN	
1	χριστος	BOTH	BOTH	BOTH	NEG	5.0	5.0	5.0	3.0	4.0	...	5.0	1.0	5.0	5.0	
2	α	BOTH	BOTH	BOTH	BOTH	4.0	5.0	5.0	1.0	5.0	...	5.0	1.0	4.0	5.0	
3	αβαπτιστος	BOTH	NaN	BOTH	NaN	3.0	NaN	5.0	NaN	4.0	...	1.0	NaN	4.0	NaN	
4	αβεβαιοτητα	NaN	NEG	NaN	NEG	NaN	1.0	NaN	1.0	NaN	...	NaN	1.0	NaN	1.0	

```
# Απόδοση των τιμών -1,0,1 στη θέση των NEG, BOTH,POS
for polarity in ['Polarity1','Polarity2','Polarity3','Polarity4']:
    Lex_DF[polarity]=Lex_DF[polarity].str.replace("BOTH","0")
    Lex_DF[polarity]=Lex_DF[polarity].str.replace("NEG","-1")
    Lex_DF[polarity]=Lex_DF[polarity].str.replace("POS","1")
    Lex_DF[polarity]=Lex_DF[polarity].apply(lambda x: x if pd.isnull(x) else int(x))
Lex_DF.head()
```

	Term	Polarity1	Polarity2	Polarity3	Polarity4	Anger1	Anger2	Anger3	Anger4	Disgust1	...	Happiness3	Happiness4	Sadness1	Sadness2	Sadne
0	αβαφτιστος	0.0	NaN	0.0	NaN	3.0	NaN	5.0	NaN	4.0	...	1.0	NaN	4.0	NaN	
1	χριστος	0.0	0.0	0.0	-1.0	5.0	5.0	5.0	3.0	4.0	...	5.0	1.0	5.0	5.0	
2	α	0.0	0.0	0.0	0.0	4.0	5.0	5.0	1.0	5.0	...	5.0	1.0	4.0	5.0	
3	αβαπτιστος	0.0	NaN	0.0	NaN	3.0	NaN	5.0	NaN	4.0	...	1.0	NaN	4.0	NaN	
4	αβεβαιοτητα	NaN	-1.0	NaN	-1.0	NaN	1.0	NaN	1.0	NaN	...	NaN	1.0	NaN	1.0	

5 rows × 29 columns

```
sentiments=['Polarity', 'Anger','Disgust', 'Fear', 'Happiness', 'Sadness','Surprise']
```

```
# Εύρεση του Μ.Ο. της πολικότητας και των συναισθημάτων για την κάθε λέξη ΄
for column in sentiments:
    Lex_DF[column] = Lex_DF[[column+'1',column+'2',column+'3',column+'4']].mean(axis=1)
Lex_DF=Lex_DF[['Term','Polarity', 'Anger','Disgust', 'Fear', 'Happiness', 'Sadness','Surprise']]
Lex_DF.head()
```

```
# Εύρεση του Μ.Ο. της πολικότητας και των συναισθημάτων για την κάθε λέξη ΄
for column in sentiments:
    Lex_DF[column] = Lex_DF[[column+'1',column+'2',column+'3',column+'4']].mean(axis=1)
Lex_DF=Lex_DF[['Term','Polarity', 'Anger','Disgust', 'Fear', 'Happiness', 'Sadness','Surprise']]
Lex_DF.head()
```

	Term	Polarity	Anger	Disgust	Fear	Happiness	Sadness	Surprise
0	αβαφτιστος	0.00	4.00	4.50	1.00	1.0	2.50	4.50
1	χριστος	-0.25	4.50	3.75	4.25	4.0	4.00	4.50
2	α	0.00	3.75	4.00	4.00	4.0	3.75	4.75
3	αβαπτιστος	0.00	4.00	4.50	1.00	1.0	2.50	4.50
4	αβεβαιοτητα	-1.00	1.00	1.00	2.50	1.0	1.50	1.00

```
#Αφαίρεση καταλήξεων -ος - η -ο
Lex_DF['Term']=Lex_DF['Term'].apply(lambda x: x.split("-")[0])
```

```
#Ευρεση ρίζας των λέξεων του λεξικού
from greek_stemmer import GreekStemmer
stemmer = GreekStemmer()

Lex_DF['Term']=Lex_DF['Term'].str.upper()
Lex_DF['Term']=Lex_DF['Term'].apply(lambda x: stemmer.stem(x))
Lex_DF.head()
```

```
C:\Users\eleni\anaconda3s\lib\site-packages\greek_stemmer\__init__.py:341: YAMLLoadWarning:

calling yaml.load() without Loader=... is deprecated, as the default Loader is unsafe. Please read https://msg.pyyaml.org/load
for full details.
```

	Term	Polarity	Anger	Disgust	Fear	Happiness	Sadness	Surprise
0	ΑΒΑΦΤΙΣΤ	0.00	4.00	4.50	1.00	1.0	2.50	4.50
1	ΧΡΙΣΤ	-0.25	4.50	3.75	4.25	4.0	4.00	4.50
2	Α	0.00	3.75	4.00	4.00	4.0	3.75	4.75
3	ΑΒΑΠΤΙΣΤ	0.00	4.00	4.50	1.00	1.0	2.50	4.50
4	ΑΒΕΒΑΙΟΤΗΤ	-1.00	1.00	1.00	2.50	1.0	1.50	1.00

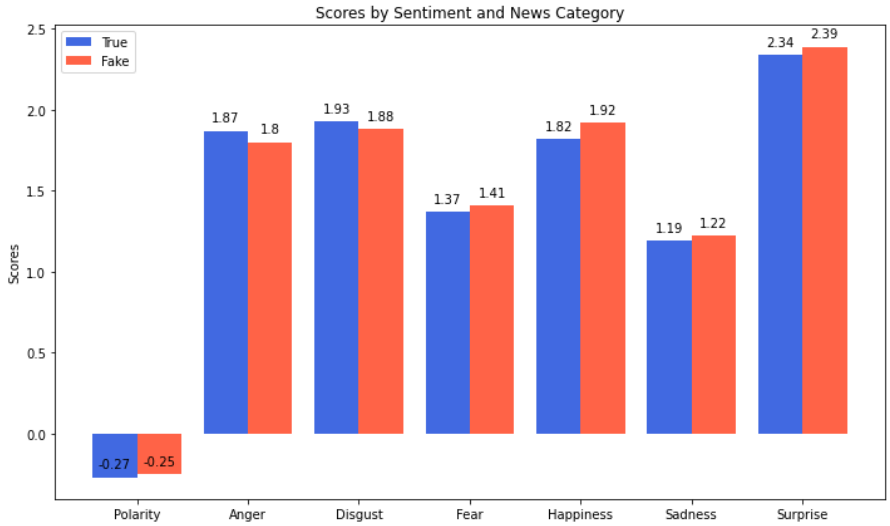
Εφαρμόσαμε ανάλυση συναισθήματος στο σύνολο των λέξεων των άρθρων που έχουν ετικέτα 0(True News) και ετικέτα 1(Fake News).

```
# Δημιουργία δύο dataframe με το σύνολο των λέξεων που εμπεριέχονται στα True και τα Fake News και την συχνότητά τους
Words_True_Article=pd.DataFrame.from_dict(Counter(chain.from_iterable(Covid_true['Words_in_Article'])),orient='index',
                                          columns=['Count']).reset_index().rename(columns= {'index':'Term'}, inplace=False)
Words_Fake_Article=pd.DataFrame.from_dict(Counter(chain.from_iterable(Covid_fake['Words_in_Article'])),orient='index',
                                          columns=['Count']).reset_index().rename(columns= {'index':'Term'}, inplace=False)
```

```
# Εύρεση της βαθμολογίας της πολικότητας και των συναισθημάτων για το σύνολο των λέξεων για τα fake και true news
df_Sent = pd.DataFrame()
i=0
Sent = pd.DataFrame(columns=sentiments)
for df in [Words_True_Article, Words_Fake_Article]:
    df['Term']=df['Term'].str.upper()# Μεταρτροπή σε κεφαλαία για να είναι συμβατό με τον stemmer
    df['Term']=df['Term'].apply(lambda x: stemmer.stem(x))# Εφαρμογή του stemmer
    df2= pd.merge(Lex_DF, df, left_on='Term', right_on='Term')
    News_Sent=[]
    for c in sentiments:
        df_Sent=df2[['Term',c,'Count']].dropna(subset = [c])
        News_Sent.extend([(df_Sent[c]*df_Sent['Count']).sum()/df_Sent['Count'].sum()])
    Sent.loc[i] = News_Sent
    i=i+1
Sent.round(2)
#0 -> True News, 1 ->Fake News
```

	Polarity	Anger	Disgust	Fear	Happiness	Sadness	Surprise
0	-0.27	1.87	1.93	1.37	1.82	1.19	2.34
1	-0.25	1.80	1.88	1.41	1.92	1.22	2.39

```
# Απεικόνιση συναισθημάτων και πολικότητας
labels = sentiments

x = np.arange(len(labels))  # the label locations
width = 0.4  # the width of the bars

fig, ax = plt.subplots(figsize=(10,6))
rects1 = ax.bar(x - width/2, Sent.loc[0].round(2), width, label='True', color='Royalblue')
rects2 = ax.bar(x + width/2, Sent.loc[1].round(2), width, label='Fake',color='Tomato')


ax.set_ylabel('Scores')
ax.set_title('Scores by Sentiment and News Category')
ax.set_xticks(x)
ax.set_xticklabels(labels)
ax.legend()


def autolabel(rects):
    """Attach a text label above each bar in *rects*, displaying its height."""
    for rect in rects:
        height = rect.get_height()
        ax.annotate('{}'.format(height),
                    xy=(rect.get_x() + rect.get_width() / 2, height),
                    xytext=(0, 5),  # 3 points vertical offset
                    textcoords="offset points",
                    ha='center', va='bottom')


autolabel(rects1)
autolabel(rects2)

fig.tight_layout()

plt.show()
```




```
# # Εύρεση της βαθμολογίας της πολικότητας και των συναισθημάτων για κάθε άρθρο στο σύνολο των άρθρων(Fake + True)
Sent_for_each_Article= pd.DataFrame(columns=sentiments)

for index, row in Covid_News.iterrows():
    lst=row['Words_in_Article']
    df=pd.DataFrame(lst,columns=['Term'])
    df['Freq']=0
    df['Term']=df['Term'].str.upper()
    df['Term']=df['Term'].apply(lambda x: stemmer.stem(x))
    df=df.groupby('Term').count().reset_index()
    df2= pd.merge(Lex_DF, df.groupby('Term').sum(), left_on='Term', right_on='Term')
    News_Sent=[]
    for c in sentiments:
        df_Sent=df2[['Term',c,'Freq']].dropna(subset = [c])
        News_Sent.extend([(df_Sent[c]*df_Sent['Freq']).sum()/df_Sent['Freq'].sum()])
    Sent_for_each_Article = Sent_for_each_Article.append(pd.DataFrame([News_Sent], columns=sentiments),ignore_index=True)

Covid_News=pd.merge(Covid_News.reset_index(), Sent_for_each_Article.reset_index(), left_on='index',
                    right_on='index').set_index('index')
```

```
Covid_News.info()
```

```
<class 'pandas.core.frame.DataFrame'>
Int64Index: 4782 entries, 0 to 4781
Data columns (total 20 columns):
 #   Column               Non-Null Count  Dtype
---  ------               --------------  -----
 0   Site                 4782 non-null   object
 1   Title                4782 non-null   object
 2   Article              4782 non-null   object
 3   Date                 4777 non-null   object
 4   Fake                 4782 non-null   int64
 5   Article_Words_Count  4782 non-null   int64
 6   Words_in_Article     4782 non-null   object
 7   Article_bigrams      4782 non-null   object
 8   Article_trigrams     4782 non-null   object
 9   Title_Words_Count    4782 non-null   int64
 10  Words_in_Title       4782 non-null   object
 11  Title_bigrams        4782 non-null   object
 12  Title_trigrams       4782 non-null   object
 13  Polarity             4745 non-null   float64
 14  Anger                4745 non-null   float64
 15  Disgust              4745 non-null   float64
 16  Fear                 4745 non-null   float64
 17  Happiness            4745 non-null   float64
 18  Sadness              4745 non-null   float64
 19  Surprise             4745 non-null   float64
dtypes: float64(7), int64(3), object(10)
memory usage: 784.5+ KB
```

```
#Διαχωρισμός dataset με συναισθήματα σε fake & true news
Covid_true = Covid_News[(Covid_News['Fake']==0)&(Covid_News['Date'] > pd.to_datetime('2019-12-31'))
                        & (Covid_News['Date'] < pd.to_datetime('2020-11-18'))]
Covid_fake = Covid_News[(Covid_News['Fake']==1)&(Covid_News['Date'] > pd.to_datetime('2019-12-31'))
                        & (Covid_News['Date'] < pd.to_datetime('2020-11-18'))]
Covid_fake.reset_index(drop=True, inplace=True)
Covid_true.reset_index(drop=True, inplace=True)
```

```
# Ημερήσια στατιστικά συναισθήματος για τα true news
Covid_true.groupby('Date').mean()[sentiments].reset_index().describe()
```

	Polarity	Anger	Disgust	Fear	Happiness	Sadness	Surprise
count	278.000000	278.000000	278.000000	278.000000	278.000000	278.000000	278.000000
mean	-0.252225	1.903998	1.935142	1.371023	1.834166	1.189370	2.358372
std	0.189341	0.272092	0.186682	0.164316	0.299570	0.083903	0.136296
min	-1.000000	1.000000	1.000000	1.000000	1.000000	1.000000	1.981481
25%	-0.346921	1.750552	1.833750	1.288119	1.689565	1.139471	2.272225
50%	-0.245787	1.880994	1.921011	1.356404	1.848949	1.180627	2.353779
75%	-0.155032	2.021587	2.014782	1.428526	1.971567	1.222587	2.444958
max	1.000000	3.600000	2.666667	3.000000	4.500000	1.666667	2.873656

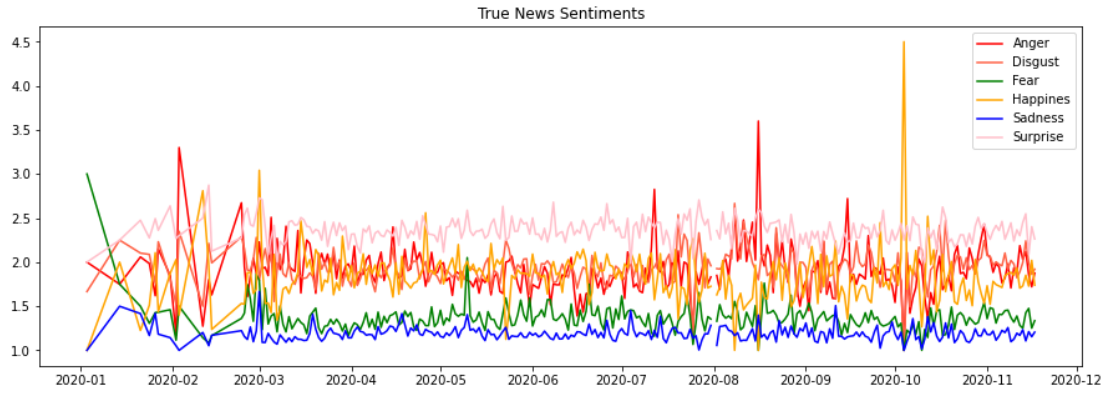
```
# Ημερήσια στατιστικά συναισθήματος για τα fake news
Covid_fake.groupby('Date').mean()[sentiments].reset_index().describe()
```

	Polarity	Anger	Disgust	Fear	Happiness	Sadness	Surprise
count	296.000000	296.000000	296.000000	296.000000	296.000000	296.000000	296.000000
mean	-0.240309	1.813696	1.881179	1.380486	1.911956	1.202350	2.394040
std	0.163599	0.199284	0.171853	0.139814	0.214272	0.082295	0.161464
min	-0.958333	1.195652	1.300000	1.000000	1.083333	1.000000	1.798611
25%	-0.332706	1.705643	1.786872	1.296085	1.792697	1.151399	2.320570
50%	-0.244243	1.812639	1.891531	1.362892	1.899692	1.191452	2.394491
75%	-0.155504	1.924137	1.973952	1.448390	2.029778	1.251800	2.469850
max	0.351256	2.568182	2.590278	2.216519	2.901093	1.576923	3.208333

Γραφήματα Συναισθημάτων

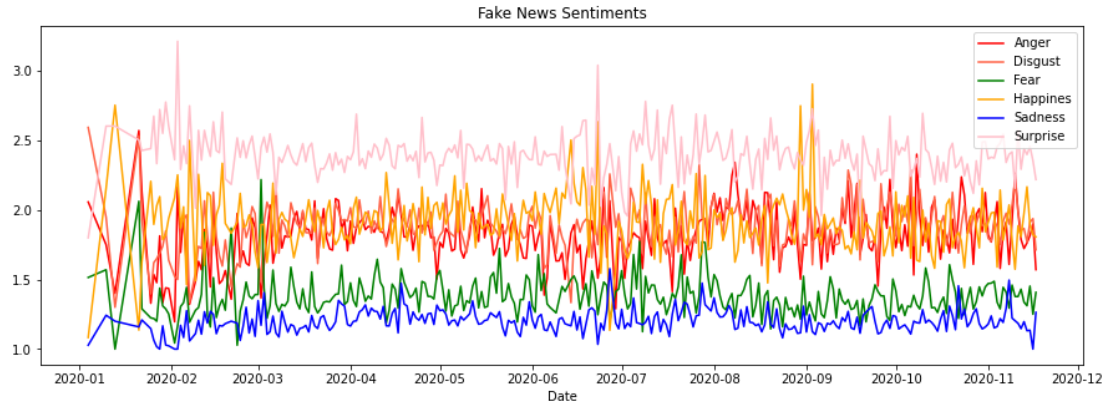
```
ax = Covid_true.groupby('Date').mean().reset_index().plot(kind='line', x='Date', y='Anger', color='Red', label='Anger',figsize=(1
Covid_true.groupby('Date').mean().reset_index().plot(kind='line', x='Date', y='Disgust', color='Tomato', label='Disgust', ax=ax)
Covid_true.groupby('Date').mean().reset_index().plot(kind='line', x='Date', y='Fear', color='Green', label='Fear', ax=ax)
Covid_true.groupby('Date').mean().reset_index().plot(kind='line', x='Date', y='Happiness', color='Orange', label='Happines', ax=a
Covid_true.groupby('Date').mean().reset_index().plot(kind='line', x='Date', y='Sadness', color='Blue', label='Sadness', ax=ax)
Covid_true.groupby('Date').mean().reset_index().plot(kind='line', x='Date', y='Surprise', color='Pink', label='Surprise', ax=ax)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x2c3222d85b0>
```



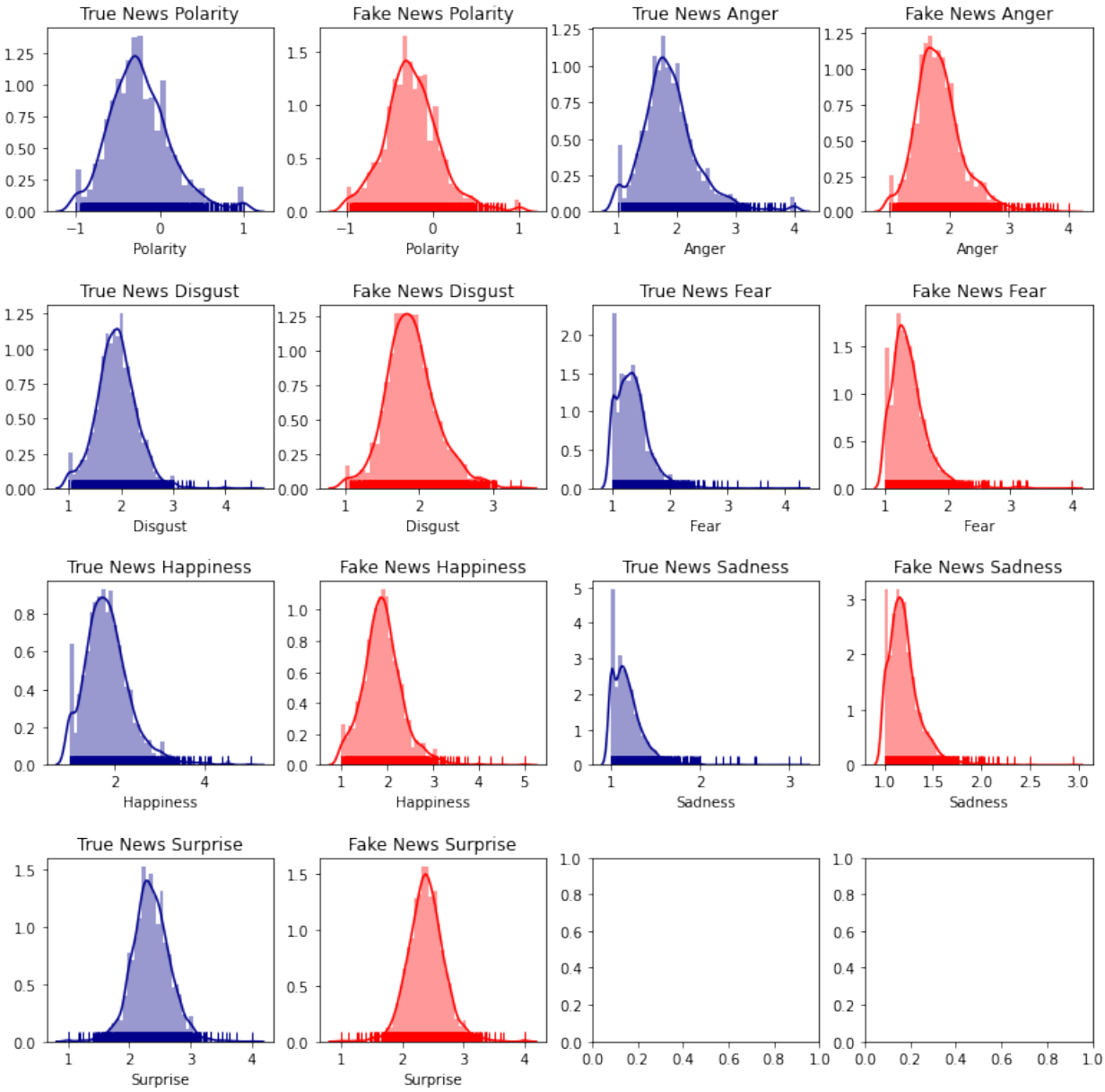
```
ax = Covid_fake.groupby('Date').mean().reset_index().plot(kind='line', x='Date', y='Anger', color='Red', label='Anger',figsize=(1
Covid_fake.groupby('Date').mean().reset_index().plot(kind='line', x='Date', y='Disgust', color='Tomato', label='Disgust', ax=ax)
Covid_fake.groupby('Date').mean().reset_index().plot(kind='line', x='Date', y='Fear', color='Green', label='Fear', ax=ax)
Covid_fake.groupby('Date').mean().reset_index().plot(kind='line', x='Date', y='Happiness', color='Orange', label='Happines', ax=a
Covid_fake.groupby('Date').mean().reset_index().plot(kind='line', x='Date', y='Sadness', color='Blue', label='Sadness', ax=ax)
Covid_fake.groupby('Date').mean().reset_index().plot(kind='line', x='Date', y='Surprise', color='Pink', label='Surprise', ax=ax)
```

```
<matplotlib.axes._subplots.AxesSubplot at 0x2c31ec7e340>
```



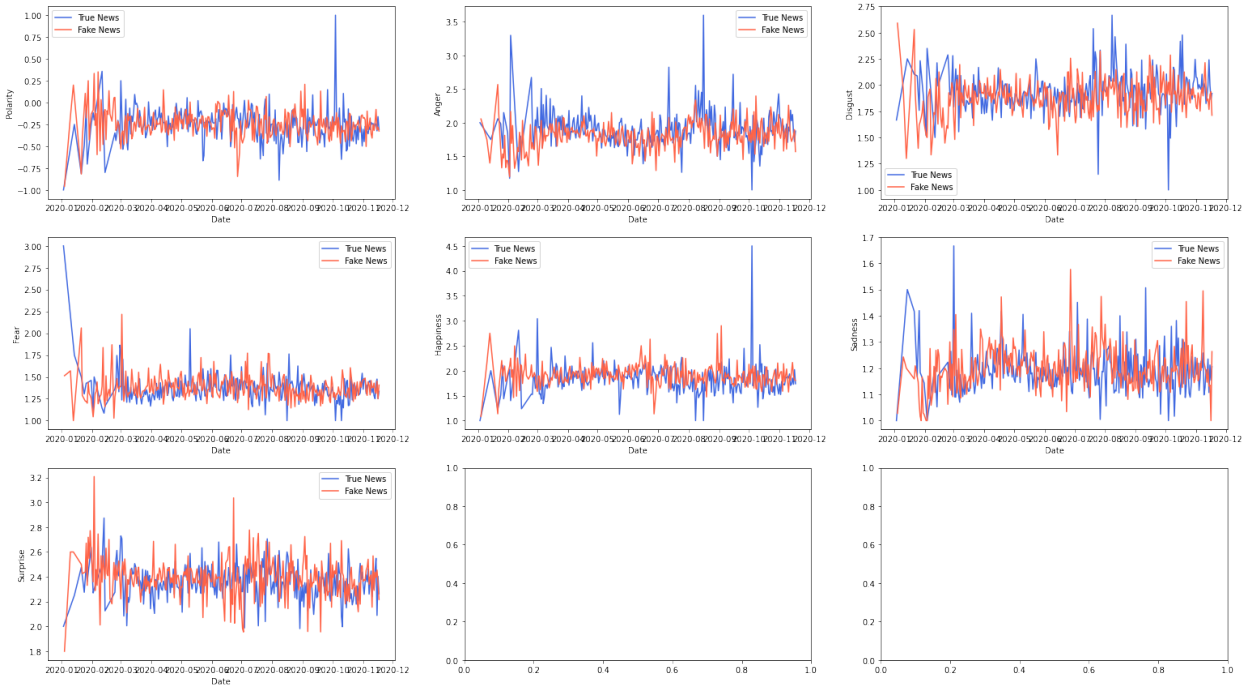
Γραφήματα αναπαράστασης κατανομής συναισθημάτων

```
%matplotlib inline
fig, ax = plt.subplots(4, 4, figsize=(15, 15))
i=0
plt.subplots_adjust(bottom=0.1, right=0.8, top=1.5, hspace = 0.25)
for c in sentiments:
    sns.distplot(Covid_true[c], ax = ax[i//2,(i*2)%4], color = "darkblue", rug = True).set_title("True News " + c)
    sns.distplot(Covid_fake[c], ax = ax[i//2,(i*2+1)%4], color = "red", rug = True).set_title("Fake News " + c)
    i=(i+1)
```



Γραφήματα Ημερήσιας Μεταβολής Συναισθημάτων

```
%matplotlib inline
fig, ax = plt.subplots(3,3, figsize=(30, 8))

plt.subplots_adjust(bottom=0.1, right=0.8, top=1.5, hspace = 0.2)
sns.lineplot(Covid_true.groupby('Date').mean().reset_index()['Date'],Covid_true.groupby('Date').mean().reset_index()['Polarity'],
             ax=ax[0,0], color='Royalblue',label='True News')
sns.lineplot(Covid_fake.groupby('Date').mean().reset_index()['Date'],Covid_fake.groupby('Date').mean().reset_index()['Polarity'],
             ax=ax[0,0], color='Tomato', label='Fake News')
sns.lineplot(Covid_true.groupby('Date').mean().reset_index()['Date'],Covid_true.groupby('Date').mean().reset_index()['Anger'],
             ax=ax[0,1], color='Royalblue',label='True News')
sns.lineplot(Covid_fake.groupby('Date').mean().reset_index()['Date'],Covid_fake.groupby('Date').mean().reset_index()['Anger'],
             ax=ax[0,1], color='Tomato',label='Fake News')
sns.lineplot(Covid_true.groupby('Date').mean().reset_index()['Date'],Covid_true.groupby('Date').mean().reset_index()['Disgust'],
             ax=ax[0,2], color='Royalblue',label='True News')
sns.lineplot(Covid_fake.groupby('Date').mean().reset_index()['Date'],Covid_fake.groupby('Date').mean().reset_index()['Disgust'],
             ax=ax[0,2], color='Tomato',label='Fake News')
sns.lineplot(Covid_true.groupby('Date').mean().reset_index()['Date'],Covid_true.groupby('Date').mean().reset_index()['Fear'],
             ax=ax[1,0], color='Royalblue',label='True News')
sns.lineplot(Covid_fake.groupby('Date').mean().reset_index()['Date'],Covid_fake.groupby('Date').mean().reset_index()['Fear'],
             ax=ax[1,0], color='Tomato',label='Fake News')
sns.lineplot(Covid_true.groupby('Date').mean().reset_index()['Date'],Covid_true.groupby('Date').mean().reset_index()['Happiness'],
             ax=ax[1,1], color='Royalblue',label='True News')
sns.lineplot(Covid_fake.groupby('Date').mean().reset_index()['Date'],Covid_fake.groupby('Date').mean().reset_index()['Happiness'],
             ax=ax[1,1], color='Tomato',label='Fake News')
sns.lineplot(Covid_true.groupby('Date').mean().reset_index()['Date'],Covid_true.groupby('Date').mean().reset_index()['Sadness'],
             ax=ax[1,2], color='Royalblue',label='True News')
sns.lineplot(Covid_fake.groupby('Date').mean().reset_index()['Date'],Covid_fake.groupby('Date').mean().reset_index()['Sadness'],
             ax=ax[1,2], color='Tomato',label='Fake News')
sns.lineplot(Covid_true.groupby('Date').mean().reset_index()['Date'],Covid_true.groupby('Date').mean().reset_index()['Surprise'],
             ax=ax[2,0], color='Royalblue',label='True News')
sns.lineplot(Covid_fake.groupby('Date').mean().reset_index()['Date'],Covid_fake.groupby('Date').mean().reset_index()['Surprise'],
             ax=ax[2,0], color='Tomato',label='Fake News')
```



5. Δημιουργία train και test dataset, εφαρμογή tf-idf

```
#Εφαρμογή του stemmer στις λέξεις των άρθρων για να έχουμε μεγαλύτερη ακρίβεια στο αποτέλεσμα
X=Covid_News['Words_in_Article'].apply(lambda x: [stemmer.stem(item.upper()) for item in x])
X=X.apply(lambda x:[item.lower() for item in x]).tolist()
pd.DataFrame(X).reset_index(drop = True).to_csv('Words_in_Article')
y=pd.DataFrame(Covid_News['Fake']).reset_index(drop=True).rename({'Fake':'label'}, axis=1)
y.to_csv('label')
```

```
import sklearn.model_selection as ms

#Create 80-20 train test split
X_train, X_test, y_train, y_test = ms.train_test_split(X, y['label'], test_size = 0.2, random_state=1)
```

```
# Εφαφρμόγή TF- IDF για να βρούμε τις "σημαντικότερς λέξεις", αυτές που εμφανίζονται σε ποσοστό κεμένων πάνω από 10%
from sklearn.feature_extraction.text import TfidfVectorizer

tfidf = TfidfVectorizer(min_df = 0.1, preprocessor = ' '.join)

#Αποθήκευσή τους σε train dataset
response_train = tfidf.fit_transform(X_train)
feature_names_train = tfidf.get_feature_names()
dense_train = response_train.todense()
denselist_train = dense_train.tolist()
df_train = pd.DataFrame(denselist_train, columns=feature_names_train)
df_train.head()
```

	covid	lockdown	αγορ	αγων	αθην	ακολουθ	ακριβως	αλεξ	αλλ	αλλαγ	...	χθες	χιλιαδ	χρειαζ	χρης	χρησιμοποι	χρον	χωρ
0	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.000000	0.000000	...	0.0	0.000000	0.0	0.0	0.0	0.000000	0.000000
1	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.000000	0.000000	...	0.0	0.124367	0.0	0.0	0.0	0.149283	0.000000
2	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.000000	0.000000	...	0.0	0.000000	0.0	0.0	0.0	0.000000	0.262195
3	0.0	0.0	0.0	0.0	0.041292	0.0	0.0	0.0	0.074052	0.487732	...	0.0	0.000000	0.0	0.0	0.0	0.000000	0.083818
4	0.0	0.0	0.0	0.0	0.000000	0.0	0.0	0.0	0.000000	0.000000	...	0.0	0.000000	0.0	0.0	0.0	0.102702	0.083062

5 rows × 436 columns

```
from sklearn.feature_extraction.text import TfidfVectorizer

tfidf = TfidfVectorizer(preprocessor = ' '.join, vocabulary = feature_names_train)

#Αποθήκευσή τους σε test dataset
response_test = tfidf.fit_transform(X_test) #This is the Sparse Document-Term Matrix
feature_names_test = tfidf.get_feature_names()
dense_test = response_test.todense() #This is the Dense Document-Term Matrix
denselist_test = dense_test.tolist()

df_test = pd.DataFrame(denselist_test, columns=feature_names_test)
df_test.head()
```

	covid	lockdown	αγορ	αγων	αθην	ακολουθ	ακριβως	αλεξ	αλλ	αλλαγ	...	χθες	χιλιαδ	χρειαζ	χρης	χρησιμοποι	χρ
0	0.000000	0.0	0.0	0.000000	0.000000	0.000000	0.000000	0.0	0.025007	0.000000	...	0.000000	0.0	0.118056	0.000000	0.000000	0.0000
1	0.082626	0.0	0.0	0.030643	0.062859	0.000000	0.028436	0.0	0.038365	0.000000	...	0.000000	0.0	0.012075	0.061859	0.025321	0.0160
2	0.000000	0.0	0.0	0.000000	0.000000	0.000000	0.000000	0.0	0.000000	0.000000	...	0.000000	0.0	0.000000	0.000000	0.000000	0.0000
3	0.000000	0.0	0.0	0.000000	0.290090	0.000000	0.000000	0.0	0.177053	0.000000	...	0.000000	0.0	0.000000	0.000000	0.000000	0.1853
4	0.028949	0.0	0.0	0.000000	0.070475	0.031884	0.000000	0.0	0.043014	0.037576	...	0.039616	0.0	0.000000	0.000000	0.000000	0.0900

5 rows × 436 columns

```
print('Διαστάσεις πίνακα εκπαίδευσης :', df_train.shape,
      '\nΔιαστάσεις πίνακα ελέγχου :', df_test.shape)
```

```
Διαστάσεις πίνακα εκπαίδευσης : (3825, 436)
Διαστάσεις πίνακα ελέγχου : (957, 436)
```

```
#Αποθήκευση
df_train.to_csv('training_data.csv')
df_test.to_csv('testing_data.csv')
y_train = pd.DataFrame(y_train, columns=["label"])
y_test = pd.DataFrame(y_test, columns=["label"])
y_train.to_csv('train_labels.csv')
y_test.to_csv('test_labels.csv')
Covid_fake.to_csv('fakeForModeling.csv')
Covid_true.to_csv('trueForModeling.csv')
```

6. Εφαρμογή μοντέλων μηχανικής μάθησης

Ανάκτηση των δεδομένων

```
import pandas as pd
import numpy as np
import seaborn as sns
import matplotlib.pyplot as plt
import nltk
import sklearn
from sklearn import metrics
import seaborn as sns
from sklearn.metrics import mean_squared_error
```

```
x_train = pd.read_csv('training_data.csv')
x_test = pd.read_csv('testing_data.csv')
y_train = pd.read_csv('train_labels.csv')
y_test = pd.read_csv('test_labels.csv')
```

```
x_train.drop(columns=['Unnamed: 0'], inplace = True)
x_test.drop(columns=['Unnamed: 0'], inplace = True)
y_train.drop(columns=['Unnamed: 0'], inplace = True)
y_test.drop(columns=['Unnamed: 0'], inplace = True)
```

```
fake = pd.read_csv('fakeForModeling.csv')
true = pd.read_csv('trueForModeling.csv')
```

```
fake.drop(columns=['Unnamed: 0'], inplace = True)
true.drop(columns=['Unnamed: 0'], inplace = True)
```

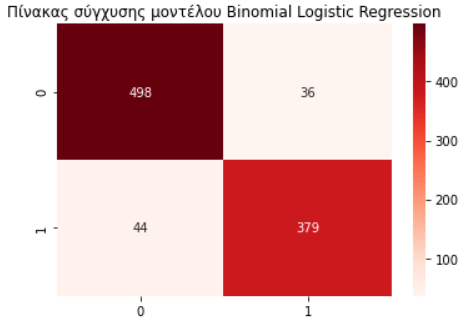
Binomial Logistic Regression

```
from sklearn.linear_model import LogisticRegression

lr = LogisticRegression()
lr.fit(x_train, np.ravel(y_train))
y_pred_lr = lr.predict(x_test)
print ("Accuracy is: ", metrics.accuracy_score(y_test, y_pred_lr))
print("Mean Squared Error is:", np.sqrt(mean_squared_error(y_test, y_pred_lr)))
```

```
Accuracy is:  0.9164054336468129
Mean Squared Error is: 0.28912724941310364
```

```
lr_cm = metrics.confusion_matrix(y_test, y_pred_lr)
print(metrics.classification_report(y_test, y_pred_lr))
labels = np.array([lr_cm[0],lr_cm[1]])
sns.heatmap(lr_cm, annot=labels, fmt = '',cmap='Reds')
plt.title('Πίνακας σύγχυσης μοντέλου Binomial Logistic Regression')
plt.show()
```

```
              precision    recall  f1-score   support

           0       0.92      0.93      0.93       534
           1       0.91      0.90      0.90       423

    accuracy                           0.92       957
   macro avg       0.92      0.91      0.92       957
weighted avg       0.92      0.92      0.92       957
```




```
#Auc
y_pred_prob_lr = lr.predict_proba(x_test)[:, 1]
metrics.roc_auc_score(y_test, y_pred_prob_lr)
```

```
0.9696080254292063
```

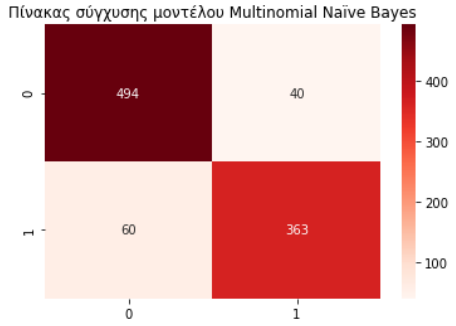
Multinomial Naïve Bayes

```
from nltk import classify
from nltk import NaiveBayesClassifier
from sklearn.naive_bayes import MultinomialNB

nb = MultinomialNB()
nb.fit(x_train, np.ravel(y_train))
y_pred_class = nb.predict(x_test)
print("Accuracy is:", metrics.accuracy_score(y_test, y_pred_class))
print("Mean Squared Error is:", np.sqrt(mean_squared_error(y_test, y_pred_class)))
```

```
Accuracy is: 0.8955067920585162
Mean Squared Error is: 0.32325409191761795
```

```
nb_cm = metrics.confusion_matrix(y_test, y_pred_class)
print(metrics.classification_report(y_test, y_pred_class))
labels =  np.array([nb_cm[0],nb_cm[1]])
sns.heatmap(nb_cm, annot=labels, fmt = '',cmap='Reds')
plt.title('Πίνακας σύγχυσης μοντέλου Multinomial Naïve Bayes')
plt.show()
```

```
              precision    recall  f1-score   support

           0       0.89      0.93      0.91       534
           1       0.90      0.86      0.88       423

    accuracy                           0.90       957
   macro avg       0.90      0.89      0.89       957
weighted avg       0.90      0.90      0.90       957
```




```
#Auc
y_pred_prob_nb = nb.predict_proba(x_test)[:, 1]
metrics.roc_auc_score(y_test, y_pred_prob_nb)
```

```
0.9486634614533251
```

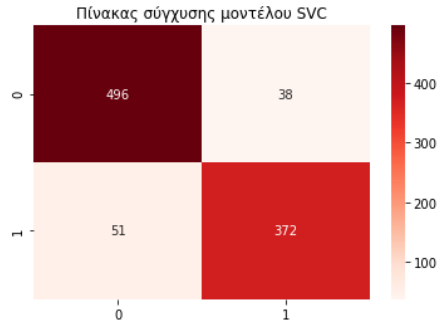
SVC

```
from sklearn.svm import SVC

svc = SVC(kernel='linear', random_state=1)
svc.fit(x_train, np.ravel(y_train))
y_pred_svm = svc.predict(x_test)
print("Accuracy is:", metrics.accuracy_score(y_test, y_pred_svm))
print("Mean Squared Error is:", np.sqrt(mean_squared_error(y_test, y_pred_svm)))
```

```
Accuracy is: 0.9070010449320794
Mean Squared Error is: 0.30495730040108987
```

```
svm_cm = metrics.confusion_matrix(y_test, y_pred_svm)
print(metrics.classification_report(y_test, y_pred_svm))
labels = np.array([svm_cm[0],svm_cm[1]])
sns.heatmap(svm_cm, annot=labels, fmt = '',cmap='Reds')
plt.title('Πίνακας σύγχυσης μοντέλου SVC')
plt.show()
```

```
              precision    recall  f1-score   support

           0       0.91      0.93      0.92       534
           1       0.91      0.88      0.89       423

    accuracy                           0.91       957
   macro avg       0.91      0.90      0.91       957
weighted avg       0.91      0.91      0.91       957
```




```
#Auc
print (metrics.roc_auc_score(y_test, y_pred_svm))
```

```
0.9041357877121683
```

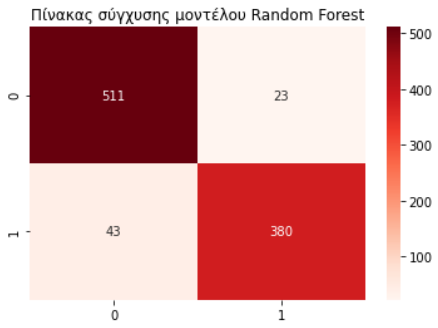
Random Forest

```
from sklearn.ensemble import RandomForestClassifier
from sklearn.model_selection import GridSearchCV

rf = RandomForestClassifier(random_state = 1)

param_grid = {
    'n_estimators': [200],
    'max_depth': [50, 60, 70]
    }

grid_search_rf = GridSearchCV(estimator = rf, param_grid = param_grid, cv = 5)
grid_search_rf.fit(x_train, np.ravel(y_train))
grid_search_rf.best_params_
y_pred_rf = grid_search_rf.predict(x_test)
print("Accuracy is:", metrics.accuracy_score(y_test, y_pred_rf))
print("Mean Squared Error is:", np.sqrt(mean_squared_error(y_test, y_pred_rf)))
```

```
Accuracy is: 0.9310344827586207
Mean Squared Error is: 0.2626128657194451
```

```
rf_cm = metrics.confusion_matrix(y_test, y_pred_rf)
labels = np.array([rf_cm[0],rf_cm[1]])
print(metrics.classification_report(y_test, y_pred_rf))
sns.heatmap(rf_cm, annot=labels, fmt = '',cmap='Reds')
plt.title('Πίνακας σύγχυσης μοντέλου Random Forest')
plt.show()
```

```
              precision    recall  f1-score   support

           0       0.92      0.96      0.94       534
           1       0.94      0.90      0.92       423

    accuracy                           0.93       957
   macro avg       0.93      0.93      0.93       957
weighted avg       0.93      0.93      0.93       957
```




```
# auc
y_pred_prob_rf = grid_search_rf.predict_proba(x_test)[:, 1]
metrics.roc_auc_score(y_test, y_pred_prob_rf)
```

```
0.9782297836923703
```

Βιβλιογραφία

[1] E. Hoaxes, «Greek Hoaxes Detector Version 1.5,» Ελληνικά Hoaxes, 06 09 2019. [Ηλεκτρονικό]. Available: http://bit.ly/2r0PLzo. [Πρόσβαση 15 11 2020].

[2] Σ. ΞΕΝΙΚΟΥΔΑΚΗΣ, «Αποπροσανατολισμός, σύγχυση, παραπληροφόρηση μέσω διαδικτύου,» *Rizospastis, https://www.rizospastis.gr/page.do?publDate=26%2F10%2F2019&pageNo=35,* τόμ. αριθ. φύλλου: 13366, p. 35, 2019.

[3] T. Zeynep, ««It's the (Democracy-Poisoning) Golden Age of Free Speech»,» *Wired,* p. 1, 2018. https://www.wired.com/story/free-speech-issue-tech-turmoil-new-censorship/?CNDID=50121752.

[4] E. Hunt, «What is fake news? How to spot it and what you can do to stop it,» *The Guardian,* αρ. https://www.theguardian.com/media/2016/dec/18/what-is-fake-news-pizzagate, 2016.

[5] R. Schlesinger, «Fake News in Reality,» 17 04 2017. [Ηλεκτρονικό]. Available: https://www.usnews.com/opinion/thomas-jefferson-street/articles/2017-04-14/what-is-fake-news-maybe-not-what-you-think. [Πρόσβαση 12 19 2020].

[6] M. Webster, «The Real Story of 'Fake News': The term seems to have emerged around the end of the 19th century,» [Ηλεκτρονικό]. Available: https://www.merriam-webster.com/words-at-play/the-real-story-of-fake-news. [Πρόσβαση 19 12 2020].

[7] N. Woolf, «How to solve Facebook's fake news problem: experts pitch their ideas,» *The Guardian,* αρ. https://www.theguardian.com/technology/2016/nov/29/facebook-fake-news-problem-experts-pitch-ideas-algorithms, 2016.

[8] POLITICO, «Fake news busters,» 14 09 2017. [Ηλεκτρονικό]. Available: http://www.politico.eu/article/fake-news-busters-germany-ben-scott/. [Πρόσβαση 19 12 2020].

[9] C. Merlo, «www.univision.com,» 04 04 2017. [Ηλεκτρονικό]. Available: https://www.univision.com/noticias/america-latina/el-millonario-negocio-detras-de-los-sitios-de-fake-news-en-mexico. [Πρόσβαση 19 12 2020].

[10] JUJU CHANG, JAKE LEFFERMAN, CLAIRE PEDERSEN and GEOFF MARTZ, «When Fake News Stories Make Real News Headlines,» 29 09 2016. [Ηλεκτρονικό]. Available: https://abcnews.go.com/Technology/fake-news-stories-make-real-news-headlines/story?id=43845383. [Πρόσβαση 19 12 2020].

[11] P. Callan, «Sue over fake news? Not so fast,» 06 12 2016. [Ηλεκτρονικό]. Available: https://edition.cnn.com/2016/12/05/opinions/suing-fake-news-not-so-fast-callan/index.html. [Πρόσβαση 12 12 2020].

[12] Mihailidis, Paul; Viotty, Samantha , ««Spreadable Spectacle in Digital Culture: Civic Expression, Fake News, and the Role of Media Literacies in "Post-Fact" Society»,» *SAGE Journals,* τόμ. 61, αρ. 4, pp. 441- 454, 2017.

[13] J. Habgood-Coote, «Stop talking about fake news!,» *Inquiry,* Τόμ. %1 από %262,2019, αρ. 9-10, 2018.

[14] ΒΙΚΙΠΑΙΔΕΙΑ, «Ψευδείς ειδήσεις,» 22 07 2020. [Ηλεκτρονικό]. Available: https://el.wikipedia.org/wiki/%CE%A8%CE%B5%CF%85%CE%B4%CE%B5%CE%AF%CF%82_%CE%B5%CE%B9%CE%B4%CE%AE%CF%83%CE%B5%CE%B9%CF%82. [Πρόσβαση 19 12 2020].

[15] Marcella Tambuscio, Giancarlo Ruffo, Alessandro Flammini, Filippo Menczer, «Fact-checking Effect on Viral Hoaxes: A Model of Misinformation Spread in,» 05 2015. [Ηλεκτρονικό]. Available: https://www.researchgate.net/publication/283009320. [Πρόσβαση 20 12 2020].

[16] E. Rogers, «The field of health communication today: an up-to-date report,» *Journal of Health Communication,* pp. 1, 15–23, 1996.

[17] D. Berry, Health communication Theory and practise, Milton Keynes, United Kingdom: Open University Press, 2007.

[18] Bennett, P. and Murphy, S., Psychology and Health Promotion, Buckingham: Open University Press, 1997.

[19] E. Laura, «Lancet retracts 12-year-old article linking autism to MMR vaccines,» *Canadian Medical Assosiation,* τόμ. 182, αρ. 4, 2010.

[20] D. K. Flaherty, «The Vaccine-Autism Connection: A Public Health Crisis Caused by Unethical Medical Practices and Fraudulent Science,» 13 09 2011. [Ηλεκτρονικό]. Available: https://journals.sagepub.com/doi/10.1345/aph.1Q318. [Πρόσβαση 19 12 2020].

[21] K. Fagerström, «The Epidemiology of Smoking,» 15 10 2012. [Ηλεκτρονικό]. Available: https://link.springer.com/article/10.2165/00003495-200262002-00001. [Πρόσβαση 20 12 2020].

[22] Unknown, «HUMAN SMOKING BEHAVIOR,» 1983. [Ηλεκτρονικό]. Available: https://www.industrydocuments.ucsf.edu/tobacco/docs/#id=ymfx0045. [Πρόσβαση 20 12 2020].

[23] K. Michael Cummings and Robert N. Proctor, «The Changing Public Image of Smoking in the United States: 1964–2014,» *Cancer Epidemiol Biomarkers Prev,* τόμ. 23, αρ. 1, pp. 32-36, 2014.

[24] ΒΙΚΙΠΑΙΔΕΙΑ, «Πανδημία κορονοϊού 2019–20,» ΒΙΚΙΠΑΙΔΕΙΑ, 10 12 2020. [Ηλεκτρονικό]. Available: https://el.wikipedia.org/wiki/%CE%A0%CE%B1%CE%BD%CE%B4%CE%B7%CE%BC%CE%AF%CE%B1_%CE%BA%CE%BF%CF%81%CE%BF%CE%BD%CE%BF%CF%8A%CE%BF%CF%8D_2019%E2%80%9320. [Πρόσβαση 20 12 2020].

[25] ΒΙΚΙΠΑΙΔΕΙΑ, «Πανδημία του κορονοϊού στην Ελλάδα το 2020,» ΒΙΚΙΠΑΙΔΕΙΑ, 15 12 2020. [Ηλεκτρονικό]. Available:

https://el.wikipedia.org/wiki/%CE%A0%CE%B1%CE%BD%CE%B4%CE%B7%CE%BC%CE%AF%CE%B1_%CF%84%CE%BF%CF%85_%CE%BA%CE%BF%CF%81%CE%BF%CE%BD%CE%BF%CF%8A%CE%BF%CF%8D_%CF%83%CF%84%CE%B7%CE%BD_%CE%95%CE%BB%CE%BB%CE%AC%CE%B4%CE%B1_%CF%84%CE%BF_2020. [Πρόσβαση 20 12 2020].

[26] Nature, «Coronavirus misinformation, and how scientists can help to fight it,» 17 06 2020. [Ηλεκτρονικό]. Available: https://www.nature.com/articles/d41586-020-01834-3. [Πρόσβαση 20 12 2020].

[27] H. Gold, «Inside the WHO's fight to stop false information about coronavirus from spreading,» 05 03 2020. [Ηλεκτρονικό]. Available: https://edition.cnn.com/2020/03/05/tech/facebook-google-who-coronavirus-misinformation/index.html. [Πρόσβαση 20 12 2020].

[28] iefimerida.gr, «Εκατοντάδες νεκροί στο Ιράν -Ηπιαν μεθανόλη ως «θεραπεία για τον κορωνοϊό»,» 30 3 2020. [Ηλεκτρονικό]. Available: https://www.iefimerida.gr/kosmos/iran-ekatontades-nekroi-ipian-methanoli-therapeia-koronoioy. [Πρόσβαση 20 12 2020].

[29] Molly Offer-Westort, Leah R. Rosenzweig, Susan Athey, «Optimal Policies to Battle the Coronavirus “Infodemic” Among Social Media Users in Sub-Saharan Africa: Pre-analysis Plan,» 19 10 2020. [Ηλεκτρονικό]. Available: https://www.gsb.stanford.edu/faculty-research/working-papers/optimal-policies-battle-coronavirus-infodemic-among-social-media. [Πρόσβαση 20 12 2020].

[30] J. Woolford, «Science Communication: how can it help against fake news?,» 10 03 2020. [Ηλεκτρονικό]. Available: https://www.hindawi.com/post/science-communication-how-can-it-help-against-fake-news/. [Πρόσβαση 20 12 2020].

[31] ΒΑΣΙΛΕΙΟΣ Σ. ΒΕΡΥΚΙΟΣ, ΒΑΣΊΛΕΙΟΣ ΚΑΓΚΛΉΣ, ΗΛΊΑΣ Κ. ΣΤΑΥΡΌΠΟΥΛΟΣ, Η επιστήμη των δεδομένων μέσα από τη γλώσσα R, Εκδόσεις Κάλλιπος, 2015.

[32] Κ. Γεωργούλη, Τεχνητή Νοημοσύνη, ΣΥΝΔΕΣΜΟΣ ΕΛΛΗΝΙΚΩΝ ΑΚΑΔΗΜΑΪΚΩΝ ΒΙΒΛΙΟΘΗΚΩΝ www.kallipos.gr, 2015.

[33] Andreas C. Müller and Sarah Guido, Introduction to Machine Learning with Python A Guide for Data Scientists, United States of America: O’Reilly Media, Inc., 2017.

[34] Matt Wiley, Joshua F. Wiley, Advanced R Statistical Programming and Data Models: Analysis, Machine Learning, and Visualization, Columbia City, IN, USA: Apress, 2019.

[35] Κ. Ευστάθιος, Επιχειρηματική ευφυΐα και εξόρυξη δεδομένων, https://repository.kallipos.gr/handle/11419/1236: Εκδόσεις Κάλλιπος, 2015.

[36] Laura Igual , Santi Seguí, Introduction to Data Science:; A Python Approach to Concepts, Techniques and Applications, Springer International Publishing Switzerland: Springer, 2017.

[37] Gareth James, Daniela Witten, Trevor Hastie, Robert Tibshirani, An Introduction toStatistical Learning: with Applications in R, New York: Springer, 2017.

[38] Χαλικιάς Μιλτιάδης, Λάλου Παναγιώτα, Μανωλέσου Αλεξάνδρα, Μεθοδολογία έρευνας και εισαγωγή στη Στατιστική Ανάλυση Δεδομένων με το IBM SPSS STATISTICS, Εκδόσεις Κάλλιπος, 2015.

[39] Π. Δημήτριος, Ανάλυση πολυμεταβλητών τεχνικών, https://repository.kallipos.gr/handle/11419/2128: Εκδόσεις Κάλλιπος, 2015.

[40] Κ. Ευστάθιος, Επιχειρηματική ευφυΐα και εξόρυξη δεδομένων, https://repository.kallipos.gr/handle/11419/1226: Εκδόσεις Κάλλιπος, 2015.

[41] L. BREIMAN, «Random Forests,» σε *Machine Learning*, Kluwer Academic Publishers, 2001, pp. 45, 5–32.

[42] Grobelnik, M., Mladenic, D., Milic-Frayling,, «Text Mining as Integration of Several Related Research Areas,» KDD'2000 Workshop on Text Mining, 2000.

[43] M. Hearst, «Untangling Text Data Mining,,» Proceedings of the 3th Annual Meeting of the Association for Computational Linguistics, 1999.

[44] D. S. Sirmakessis, Text Mining and its Applications: Results of the NEMIS Launch Conference, New York: Springer-Verlag Berlin Heidelberg, 2004.

[45] T. Z. N. I. Sholom M. Weiss, Fundamentals of Predictive Text Mining 2nd edition, London: Springer-Verlag London, 2015.

[46] Κωνσταντίνος Μαγκούτης, Χρήστος Νικολάου, "Εισαγωγή στον αντικειμενοστραφή προγραμματισμό με Python", www.kalipos.gr, 2015.

[47] M. Lutz, Learning Python, Fourth Edition, Gravenstein Highway North, Sebastopol: O'Reilly Media, Inc, 2009.

[48] scikit-learn, «https://scikit-learn.org/,» scikit-learn, [Ηλεκτρονικό]. Available: https://scikit-learn.org/. [Πρόσβαση 30 12 2020].

[49] numPy, «https://numpy.org/about/,» [Ηλεκτρονικό]. Available: https://numpy.org/about/. [Πρόσβαση 30 12 2020].

[50] Scipy, «https://docs.scipy.org/doc/scipy/reference/tutorial/general.html,» [Ηλεκτρονικό]. Available: https://docs.scipy.org/doc/scipy/reference/tutorial/general.html. [Πρόσβαση 30 12 2020].

[51] «https://pandas.pydata.org/about/,» [Ηλεκτρονικό].

[52] Δημήτριος Μάλλης, Γεώργιος Καλαματιανός, Δημήτριος Νικολαράς, Αυγερινός Αραμπατζής, «Τμήμα Ηλεκτρολόγων Μηχανικών και Μηχανικών Υπολογιστών, Πολυτεχνική Σχολή,

Δημοκρίτειο Πανεπιστήμιο Θράκης,» 2015. [Ηλεκτρονικό]. Available: http://www.aviarampatzis.com/publications/sfhmmy2015b.pdf. [Πρόσβαση 15 12 2020].

[53] Sarah Guido, Andreas Müller, IIntroduction to Machine Learning with Python, United States of America: O'Reilly Media, Inc., 2017.

[54] M. Himma-Kadakas, « Alternative facts and fake news entering journalistic content production cycle,» *Cosmopolitan Civil Societies: An Interdisciplinary Journal,* τόμ. 9 (2), pp. 25-41, 2017.

[55] S. Mukhopadhyay, Advanced Data Analytics Using Python, Kolkata, West Bengal, India: Apress, 2018.