ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΜΕ ΕΦΑΡΜΟΓΕΣ ΣΤΗ ΒΙΟΙΑΤΡΙΚΗ

ΑΝΑΓΝΩΡΙΣΗ ΟΝΟΜΑΤΙΚΩΝ ΟΝΤΟΤΗΤΩΝ ΓΙΑ ΤΗΝ ΑΝΙΧΝΕΥΣΗ ΨΕΥΔΩΝ ΕΙΔΗΣΕΩΝ ΜΕΣΩ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ

Όλγα Κόγιου

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Υπεύθυνος
Σωτήρης Τασουλής
Επίκουρος καθηγητής

Λαμία, 2021

ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ
ΣΧΟΛΗ ΘΕΤΙΚΩΝ ΕΠΙΣΤΗΜΩΝ
ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ ΜΕ ΕΦΑΡΜΟΓΕΣ ΣΤΗ ΒΙΟΙΑΤΡΙΚΗ

ΑΝΑΓΝΩΡΙΣΗ ΟΝΟΜΑΤΙΚΩΝ ΟΝΤΟΤΗΤΩΝ ΓΙΑ ΤΗΝ ΑΝΙΧΝΕΥΣΗ ΨΕΥΔΩΝ ΕΙΔΗΣΕΩΝ ΜΕΣΩ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ

Όλγα Κόγιου

ΠΤΥΧΙΑΚΗ ΕΡΓΑΣΙΑ

Επιβλέπων
Σωτήρης Τασουλής
Επίκουρος καθηγητής

Λαμία, 2021

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

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

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

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

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

Ημερομηνία: 31/05/2021

Η Δηλούσα
Όλγα Κόγιου

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

ΑΝΑΓΝΩΡΙΣΗ ΟΝΟΜΑΤΙΚΩΝ ΟΝΤΟΤΗΤΩΝ ΓΙΑ ΤΗΝ ΑΝΙΧΝΕΥΣΗ ΨΕΥΔΩΝ ΕΙΔΗΣΕΩΝ ΜΕΣΩ ΜΗΧΑΝΙΚΗΣ ΜΑΘΗΣΗΣ

Όλγα Κόγιου

Τριμελής Επιτροπή:

Σωτήρης Τασουλής, Επίκουρος Καθηγητής

Αθανάσιος Καακαρούντας, Αναπληρωτής Καθηγητής

Χαράλαμπος Καρανίκας, Λέκτορας

ΠΕΡΙΛΗΨΗ

Το διαδίκτυο αποτελεί σήμερα ένα από τα κύρια μέσα ενημέρωσης καθώς καταφέρνει να καθιστά τις ειδήσεις ευρέως προσβάσιμες σε ελάχιστο χρονικό διάστημα. Ο όγκος των πληροφοριών που παράγονται καθημερινά ολοένα και αυξάνονται. Παράλληλα όμως αυξάνονται και οι ανησυχίες που εγείρονται γύρω από την αξιοπιστία και την ποιότητα των ειδήσεων που διαδίδονται. Επιτακτικής ανάγκης συνεπώς, είναι η εύρεση ενός αυτόματου τρόπου διαχωρισμού των ψευδών ειδήσεων από τις αληθείς. Ο στόχος της αυτόματης αυτής ανίχνευσης των ψευδών ειδήσεων, είναι να μειωθεί στο ελάχιστο η ανθρώπινη παρέμβαση, εξοικονομώντας έτσι χρόνο. Στην παρούσα εργασία, παρουσιάζεται ένας τρόπο ανάλυσης της φυσικής γλώσσας από κείμενο και κατηγοριοποίησης των ειδήσεων σε αληθείς και ψευδείς. Πιο συγκεκριμένα, το σύνολο δεδομένων που χρησιμοποιήθηκε προέρχεται από τον ιστότοπο Kaggle και περιέχει τόσο ψευδείς όσο και αληθείς ειδήσεις. Αρχικά, από τα δεδομένα αυτά χρειάζεται να εξαχθεί η πληροφορία που θα μας βοηθήσει στην κατηγοριοποίησή τους. Η διαδικασία αυτή πραγματοποιείται με την αναγνώριση ονοματικών οντοτήτων. Κατόπιν, γίνεται διαχωρισμός των ειδήσεων αυτών σε ψευδείς ή αληθείς μέσω της επιβλεπόμενης μάθησης. Τα αποτελέσματα που παρουσιάζονται αποδεικνύουν ότι η αυτοματοποιημένη μέθοδος που προτείνεται αποτελεί κατάλληλο εργαλείο ανάλυσης της φυσικής γλώσσας για την εξαγωγή συμπερασμάτων και για την αναγνώριση των ψευδών ειδήσεων.

ABSTRACT

Nowadays, the internet is one of the main sources of information since it can contribute to wide accessibility in very little time. Today, the amount of information is forever increasing and so are the concerns regarding the reliability and the quality of the news being spread. Therefore, the construction of an automatic way to detect fake news is an urgent need. The main goal is to reduce human effort to its minimum. In this work, a way of natural language analysis from text and classification of fake and real news is proposed. More specifically, the dataset used comes from the website Kaggle and contains fake as well as real news. Firstly, feature extraction is needed in order to simplify the data. Named entity recognition works towards this direction. Afterwards, supervised learning is introduced to classify the data. Presented results prove that the suggested automatic model is a fit tool for natural language processing deductions and for fake news detection.

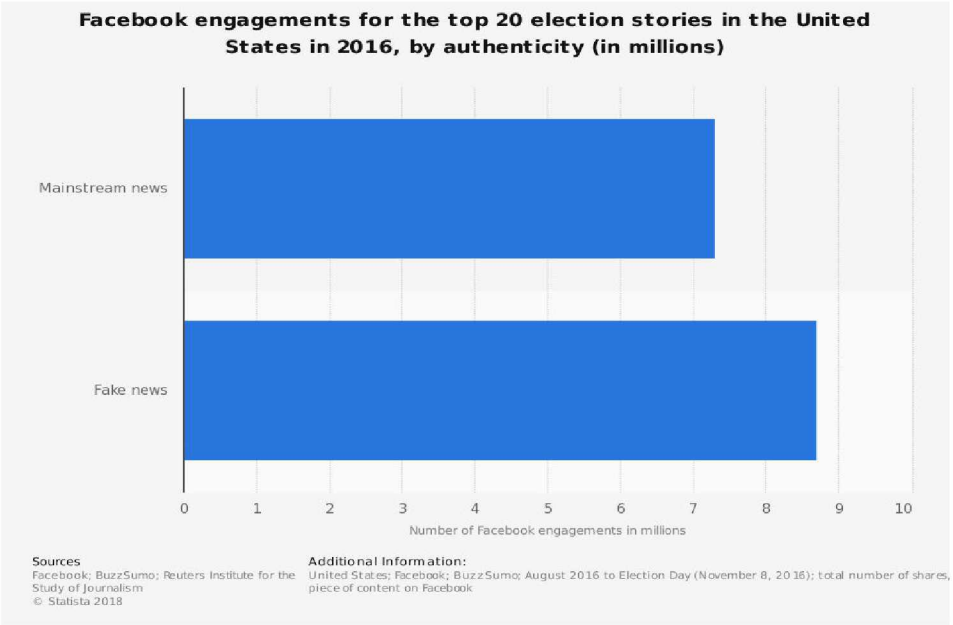
Keywords: Machine Learning, Text Classification, Fake News, Naive Bayes, spaCy, Entity recognition

ΠΕΡΙΕΧΟΜΕΝΑ

1 ΕΙΣΑΓΩΓΗ

Ο όρος «Ψευδείς ειδήσεις» χρησιμοποιείται ευρέως για την περιγραφή σκανδαλοθηρικών και ανακριβών ειδήσεων, ιδίως για να περιγράψει αντικειμενικά λανθασμένα και παραπλανητικά άρθρα που δημοσιεύτηκαν κυρίως με σκοπό το χρηματικό κέρδος μέσω προβολών σελίδας και διαφημίσεων. Το 2016 μάλιστα, η παραπληροφόρηση εντός του αμερικανικού πολιτικού σκηνικού, αποτέλεσε αντικείμενο ιδιαίτερης προσοχής, ιδίως μετά την εκλογή του Προέδρου Τραμπ [1].

Παράλληλα, τα μέσα κοινωνικής δικτύωσης, λόγω της μεγάλης επιρροής τους στην καθημερινή σύγχρονη ζωή, φαίνεται να παίζουν και μεγάλο ρόλο στην τάχιστη εξάπλωση των ψευδών ειδήσεων. Πιο συγκεκριμένα, ενδιαφέρον είναι το πόρισμα της ανάλυσης των αποτελεσμάτων στην κινητικότητα των Αμερικανών χρηστών του Facebook, στις εκλογές του 2016, όπου φαίνεται ότι οι ψευδείς ειδήσεις κέρδισαν σε δημοτικότητα (likes, shares κτλ) σε σχέση με τις πραγματικά αληθείς από μεγάλα διαδικτυακά ειδησεογραφικά πρακτορεία (*Εικόνα 1.1)*



Εικόνα 1.1: Δημοτικότητα των κορυφαίων 20 Ειδήσεων στις ΗΠΑ κατά την προεκλογική εκστρατεία του 2016 (κατά εκατομμύριο). Πηγή: BuzzFeed

Καθίσταται σαφές λοιπόν, πως οι ειδήσεις και η ποιότητά τους κατέχουν καίριο ρόλο στην διαμόρφωση της κοινής γνώμης και μπορούν να οδηγήσουν σε ποικίλες κοινωνικοπολιτικές προεκτάσεις. Δεδομένης της κρισιμότητας του προβλήματος, η αντιμετώπισή του έχει αρχίσει να απασχολεί τόσο την επιστημονική κοινότητα όσο και το κοινό που αναζητά έναν άμεσο τρόπο επαλήθευσης των ειδήσεων που βρίσκουν στο διαδίκτυο. Πρόκειται όμως για ένα σύνθετο πρόβλημα που μπλέκει πολλούς τομείς και γνωστικά αντικείμενα και σίγουρα υπάρχουν πολλά ακόμα που πρέπει να γίνουν μέχρι να έχει εξαλειφθεί πλήρως το πρόβλημα της προπαγάνδας και της παραπληροφόρησης. Σε κάθε περίπτωση, απαιτείται ανάλυση τόσο της φυσικής γλώσσας όσο και του συναισθηματικού τόνου που χρησιμοποιείται στο άρθρο καθώς και του πλήθους των τεκμηρίων που υπάρχουν όπως πολιτικά ονόματα, ημερομηνίες, στατιστικά στοιχεία, γεωγραφικές αναφορές κτλ.

1.1 Σχετικές εργασίες- υλοποιήσεις.

Σε αυτή την ενότητα θα αναφερθούμε στις υπάρχουσες υλοποιήσεις που στοχεύουν στον εντοπισμό μη έγκυρων ειδήσεων στο διαδίκτυο. Θα εστιάσουμε σε αυτές που προέρχονται από την επιστήμη της Πληροφορικής και βασίζονται σε τεχνικές μηχανικής μάθησης και εξόρυξης γνώσης από κείμενο για την ανάλυση των δεδομένων και την εξαγωγή συμπερασμάτων.

Ιστορικά, μία από τις πρώτες εφαρμογές ήταν η ανίχνευση ψευδών ειδήσεων (hoaxes[1]) σε μηνύματα ηλεκτρονικό ταχυδρομείο και ιστοσελίδες. Στο πλαίσιο της ανίχνευσης ηλεκτρονικών μηνυμάτων (email) απάτης, το "Spamassassin"[2] χρησιμοποιεί μεθόδους βασισμένες σε λέξεις-κλειδιά με λογιστική παλινδρόμηση (logistic regression) [2]. Έτσι, μετράται η συχνότητα εμφάνισης της κάθε λέξεις τόσο στα επιθυμητά, όσο και στα ανεπιθύμητα emails και με βάση αυτή γίνεται η ταξινόμηση.

Μια ακόμη σημαντική και πρόσφατη προσέγγιση έγινε από τους Rohit Kumar Kaliyar κ.ά. [4] όπου προτείνουν το "FNDNet" για τον εντοπισμό των ψεύτικων ειδήσεων. Το μοντέλο "FNDNet" έχει σχεδιαστεί για να μαθαίνει αυτόματα τις διαφορετικές δυνατότητες για την ταξινόμηση επισφαλών ειδήσεων μέσω πολλαπλών κρυφών επιπέδων ενσωματωμένων στο βαθύ νευρωνικό δίκτυο, αντί να βασίζεται σε εξαγόμενα από τον δημιουργό χαρακτηριστικά. Αντίθετα, δημιουργείται ένα βαθύ συνελικτικό νευρωνικό δίκτυο (CNN) για την εξαγωγή διαφόρων χαρακτηριστικών σε κάθε επίπεδο.

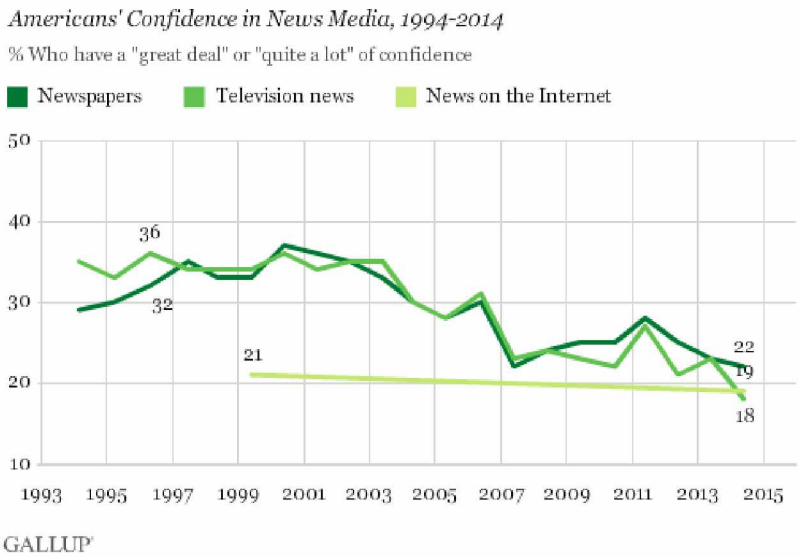
Τέλος, μια καινοτόμα ιδέα ξεκίνησε το 2018 από τους Marco L. Della Vedova κ.ά. [5] οι οποίοι συνεχίζοντας το έργο [6] παρουσίασαν μια νέα προσέγγιση ανίχνευσης ψεύτικων ειδήσεων όπου, συνδυάζοντας αλγοριθμικές μεθόδους φιλτραρίσματος κειμένου ή άρθρων (method based) και μεθόδους αξιολόγησης των άρθρων από το κοινό (social based), υπερτερεί σε υπάρχουσες προσεγγίσεις στη βιβλιογραφία. Οι υβριδικές μέθοδοι όπως αυτή, έχουν χρησιμοποιηθεί και σε άλλες παρόμοιες περιπτώσεις ως μια προσπάθεια κατάργησης της χρήσης αμιγών αξιολογήσεων από το κοινό (social based) όταν ένα αντικείμενο π.χ. άρθρο έχει μηδενικές κριτικές [7].

1.2 Περιγραφή προβλήματος

Όσο περισσότερο η τεχνολογία φαίνεται να λαμβάνει πρωταγωνιστικό ρόλο στην ενημέρωση, τόσο εντείνεται η αμφισβήτηση των μέχρι τώρα ευρέως χρησιμοποιούμενων μέσων ενημέρωσης [8]. Από δημοσκόπηση που έγινε το 2016, μετά την εκλογή του πρώην Προέδρου των ΗΠΑ Ντόναλντ Τραμπ, φαίνεται μια καθαρά πτωτική τάση στην εμπιστοσύνη των Αμερικανών στα Μαζικά Μέσα Ενημέρωσης [9] *(Εικόνα 1.2)*.

[1] Η έννοια hoax αναφέρεται σε ηλεκτρονικά σκουπίδια που μπορεί δίνουν παραπλανητικές πληροφορίες στους χρήστες ή στους αναγνώστες μιας ηλεκτρονικής διεύθυνσης.

[2] Σύμφωνα με τους Blanzieri και Bryl [3], το spam είναι ανεπιθύμητο ταχυδρομείο.



Εικόνα 1.2: Η εμπιστοσύνη του Αμερικανικού κοινού στα μέσα μαζικής ενημέρωσης κατά τα έτη 1993-2015 σε ποσοστά (%). Πηγή: news.gallup

Ως αποτέλεσμα, αυξάνεται η καταφυγή του κοινού σε εναλλακτικές μορφές ενημέρωσης που προσφέρει το διαδίκτυο όπως οι ενημερωτικές ιστοσελίδες και τα μέσα κοινωνικής δικτύωσης. Επομένως, οι αναγνώστες σε μια προσπάθειά τους να αποφύγουν την υποκειμενικότητα μεγάλων σταθμών, εκτίθενται σε ακόμα μεγαλύτερο όγκο ειδήσεων κρυμμένο πίσω από την ανωνυμία του διαδικτύου. Επιπλέον, ο αλγόριθμος για την προβολή περιεχομένου στα κοινωνικά δίκτυα οδηγεί στην μονόπλευρη και ομοιόμορφη ενημέρωση των χρηστών καθώς περιεχόμενο προτείνεται στους χρήστες με βάση τα ενδιαφέροντά τους και τις σελίδες και δημοσιεύσεις που έχουν ήδη επισκεφτεί. Έτσι, δημιουργείται ένα μόνιμο μοτίβο καλλιέργειας τέτοιων ειδήσεων οι οποίες κερδίζουν το αναγνωστικό κοινό και οξύνουν το πρόβλημα. Το κοινό, ως αποτέλεσμα, δυσκολεύεται να ξεχωρίσει άμεσα ποια άρθρα έχουν σκοπό να παραπλανήσουν και ποια όχι.

Ιδιαίτερα σε νεαρές ηλικίες όπου το διαδίκτυο αποτελεί το κύριο μέσο ενημέρωσης [10] φαίνεται ότι σύμφωνα με έρευνα που διεξήχθει από το Πανεπιστήμιο Stanford [11] η ανικανότητα διαχωρισμού των ψευδών ειδήσεων είναι πρωτοφανής καθώς πάνω από το 80% των παιδιών που πήραν μέρος στην έρευνα πίστευαν ότι το διαφημιστικό περιεχόμενο αποτελεί έμπιστη είδηση.

Το γεγονός αυτό αποτελεί μια τουλάχιστον ανησυχητική διαπίστωση που μπορεί να οφείλεται σε πληθώρα παραγόντων. Ένας βασικός παράγοντας είναι χωρίς αμφισβήτηση οι πολλές διαφορετικές κατηγορίες αναληθών ειδήσεων. Μερικές από αυτές είναι:

- Η λάθος ταύτιση εικόνων και τίτλων με το περιεχόμενο (clickbait)
- Το λανθασμένο περιεχόμενο άρθρων (π.χ. παλιές ειδήσεις)
- Η σάτιρα ή κωμωδία
- Η χρήση προσωπικών απόψεων
- Τα κατασκευασμένα άρθρα

Η ανίχνευση των διαφορετικών ειδών κρίνεται εξαιρετικά δύσκολη και πολλές φορές αδύνατη καθώς τα είδη δεν έχουν, πολλές φορές, ειδοποιά χαρακτηριστικά και μάλιστα μπορεί να υπάρχει επικάλυψη μεταξύ τους. Λόγω των διαφορετικών κατηγοριών που παρατηρούνται, η εργασία αυτή εστιάζει στην κύρια μορφή των ψευδών ειδήσεων, δηλαδή σε λάθος ισχυρισμούς στο περιεχόμενο του άρθρου.

Βάσει λοιπόν των παραπάνω, καταλήγουμε στο συμπέρασμα ότι ένα τέτοιου είδους πρόβλημα θα μπορούσε αρχικά να αντιμετωπιστεί σε δύο σκέλη. Το ένα σκέλος περικλείει τις μεθόδους και τεχνικές που μπορούν να εφαρμοστούν για τον εντοπισμό ψευδών ειδήσεων και το δεύτερο σκέλος αφορά στην υλοποίηση ενός εύχρηστου συστήματος που να αφορά στο κοινό και να μπορεί να διανεμηθεί.

Αρχικά, τα άρθρα που επιλέχθηκαν χρειάζεται να απλοποιηθούν, δηλαδή η φυσική γλώσσα με την οποία γράφτηκαν πρέπει να στερηθεί από την περίσσεια της με την μέθοδο της λεξικολογικής ανάλυσης του κειμένου [12]. Στην συνέχεια, ενσωματώνεται το κομμάτι της κατηγοριοποίησης με την χρήση τεχνικών επιβλεπόμενης μηχανικής μάθησης [13]. Υψίστης σημασίας κρίνεται η εξαγωγή και διαλογή των χαρακτηριστικών εκείνων που θα αντιπροσωπεύουν κατάλληλο το σύνολο δεδομένων που θα χρησιμοποιηθεί.

Στόχος αυτής της εργασίας είναι η κατασκευή ενός γρήγορου, έμπιστου και ακριβούς εργαλείου για την ανίχνευση των ψευδών ειδήσεων αλλά και την επεξεργασία της φυσικής γλώσσας, φιλικό προς τον χρήστη που να διανέμεται σε πολλές πλατφόρμες.

1.3 Συνεισφορά

Καθώς αναφερόμαστε σε επεξεργασία φυσικής γλώσσας και άρθρων και την εξόρυξης γνώσης από πηγές στο διαδίκτυο, καθίσταται αναγκαία η ανίχνευση ονοματικών οντοτήτων, όπως γίνεται και στις περισσότερες περιπτώσεις εκπαίδευσης ταξινομητών με δεδομένα κειμένου. Οι **οντότητες**[3] στο κείμενο μπορούν να προσφέρουν σημαντική γνώση και διαφέρουν από άρθρο σε άρθρο γι αυτό και η διερεύνησή τους αποτελεί ενδιαφέρον κομμάτι.

Βάσει της ήδη υπάρχουσας βιβλιογραφίας γύρω από την έρευνα που αφορά την εξόρυξη γνώσης από κείμενα, πραγματοποιήθηκε εξαγωγή γνωρισμάτων από ένα σύνολο κειμένων κάνοντας χρήση διαφόρων βιβλιοθηκών της Python.

Στη συνέχεια, τα γνωρίσματα που συλλέχθηκαν χρησιμοποιήθηκαν για την εκπαίδευση αλγορίθμων μηχανικής μάθησης όπως είναι ο **Random Forest**, ο **Naive Bayes** και η **Support Vector Machine**. Παράλληλα, αποτυπώθηκε μια σύντομη συγκριτική μελέτη που αναδεικνύει τις διαφορές στον τρόπο λειτουργίας αλλά και κάποιες μετρήσεις απόδοσης που έχουν γίνει έως σήμερα μεταξύ του **spaCy** και των πιο δημοφιλών εργαλείων NLP (Natural Language Processing) όπως είναι το **NLTK** και το **coreNLP.**

Κατά αυτόν τον τρόπο αναδεικνύεται η αναγνώριση ονοματικών οντοτήτων στον τομέα των ψευδών ειδήσεων. Συνεχίζοντας το έργο [14] λοιπόν, σε αυτή την εργασία δίνεται έμφαση στην ανίχνευση των ψευδών ειδήσεων και στην σημαντικότητα της χρήσης της spaCy σε αντίθεση με άλλα εργαλεία καθώς και σύγκριση των αποτελεσμάτων χωρίς την εμφάνιση των ονοματικών οντοτήτων στα χαρακτηριστικά των δεδομένων.

Στον *πίνακα 1.1* φαίνονται η σύγκριση της συνεισφοράς της παρούσας εργασίας σε σύγκριση με το πιο εμπεριστατωμένο έργο ανάδειξης της αναγνώρισης ονοματικών οντοτήτων [15].

---

[3] Στην εξαγωγή πληροφορίας, ονοματική οντότητα αποτελεί ένα αντικείμενο του πραγματικού κόσμου, όπως άτομα, τοποθεσίες, οργανισμοί, προιόντα, κτλ. στα οποία μπορεί να αποδοθούν κανονικά ονόματα. Μπορούν είτε να είναι αφηρημένα είτε να έχουν φυσική ύπαρξη. Παραδείγματα ονοματικών οντοτήτων περιλαμβάνουν τα ονόματα Μπάρακ Ομπάμα, Νέα Υόρκη, Volkswagen Golf,ή οτιδήποτε άλλο μπορεί να έχει όνομα.

	NER Survey[15]	*Our Solution*
Fake News		✓
NLTK	✓	✓
spaCy	✓	✓
coreNLP		✓
No entity solution		✓
Classification		✓

Πίνακας 1.1: Σύγκριση βασικών στοιχείων της παρούσας εργασίας με την Ερευνητική δουλειά των Schmitt, Xavier για την ανίχνευση ψευδών ειδήσεων.

Επιπλέον, σύμφωνα με τα αποτελέσματα της ταξινόμησης και χρησιμοποιωντας το καταλληλότερο χαρακτηριστικό, έγινε κατασκευή μιας **Διαδικτυακής Εφαρμογής** (Web App) με την χρήση του Streamlit[4].

Τέλος, για την ευρεία διάθεση της εφαρμογής αναδείχθηκε η αρχή λειτουργίας των **Docker containers**, που αποτελεί ένα μέσο ανεξαρτητοποίησης της εφαρμογής από το λειτουργικό περιβάλλον, καθώς και του **Docker Swarm,** παρέχοντας έτσι ένα κατάλληλο γραφικό περιβάλλον για τον χρήστη.

1.4 Δομή εργασίας

Σε αυτή την ενότητα παρουσιάζεται συνοπτικά η δομή της εργασίας και το αντικείμενο διερεύνησης του κάθε κεφαλαίου.

Στο *Κεφάλαιο 2* αποτυπώνονται και επεξηγούνται οι βασικές έννοιες γύρω από τη Μηχανική Μάθηση και την εξόρυξη γνώσης από κείμενο. Σκοπός του κεφαλαίου αυτού είναι να μεταδώσει στον αναγνώστη την απαραίτητη γνώση για την κατανόηση των τεχνικών που έχουν χρησιμοποιηθεί σε αυτήν την εργασία και αφορούν την κατηγοριοποίηση κειμένου.

Στο *Κεφάλαιο 3* αναλύεται η μεθοδολογία που ακολουθήθηκε για την συλλογή των δεδομένων που αποτέλεσαν το σύνολο εκπαίδευσης των αλγορίθμων ταξινόμησης. Ακόμη, παρουσιάζονται τα γνωρίσματα που επιλέχθηκαν να χρησιμοποιηθούν κατά την διαδικασία χαρακτηρισμού μιας είδησης ως ψευδούς.

Οι μετρήσεις της απόδοσης των αλγορίθμων που χρησιμοποιήθηκαν για την επίδειξη της κατηγοριοποίησης παρουσιάζονται στα αποτελέσματα του *4ου Κεφαλαίου.*

Η περιγραφή και η ανάλυση της υλοποίησης του συστήματος πραγματοποιείται στο *Κεφάλαιο 5*. Γίνεται αναφορά στα κύρια εργαλεία που χρησιμοποιήθηκαν, επεξηγείται ο τρόπος λειτουργίας τους και αναδεικνύεται ο ρόλος τους.

[4] Το Streamlit αποτελεί μια βιβλιοθήκη ανοικτού λογισμικού της Python όπου εξυπηρετεί την δημιουργία Διαδικτυακών Εφαρμογών Μηχανικής Μάθησης και Επιστήμης των Δεδομένων (Data Science Web Apps).

Τέλος, στο *Κεφάλαιο 6* γίνεται η εξαγωγή των συμπερασμάτων της έρευνας και για την αρχιτεκτονική που επιλέχθηκε για την υλοποίηση του συστήματος ενώ προτείνονται ιδέες για μελλοντικές επεκτάσεις του.

2 ΘΕΩΡΗΤΙΚΟ ΥΠΟΒΑΘΡΟ

Σε αυτό το Κεφάλαιο γίνεται επεξήγηση των θεμελιωδών εννοιών που θα συναντηθούν στην συνέχεια σε αυτή την εργασία. Έτσι, ο αναγνώστης καλύπτεται θεωρητικά και καταρτάται επιστημονικά για να μπορέσει να ολοκληρώσει την ανάγνωση της δουλειάς μας. Αρχικά, η *Ενότητα 2.1* πραγματεύεται την χρήση της Μηχανικής Μάθησης για τον εντοπισμό των Ψευδών ειδήσεων. Πιο συγκεκριμένα, αναλύεται η διαδικασία εξαγωγής χαρακτηριστικών και οι βιβλιοθήκες που τίθενται προς σύγκριση. Στην συνέχεια, επεξηγείται η διαδικασία της κατηγοριοποίησης και περιγράφονται οι τρόποι λειτουργίας των ταξινομητών. Τέλος, γίνεται αναφορά στις μετρικές εκτίμησης της απόδοσής τους.

2.1 Εντοπισμός ψευδών ειδήσεων με μεθόδους Μηχανικής Μάθησης

Η Μάθηση (Learning) είναι μία από τις θεμελιώδεις ιδιότητες της νοήμονος συμπεριφοράς του ανθρώπου. Αντίστοιχα, εξελίχθηκε η Μηχανική Μάθηση (Machine Learning) που χαρακτηρίζει τον τρόπο λειτουργίας ενός υπολογιστικού συστήματος. Αναλυτικότερα, 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 ονομάζεται Μηχανική Μάθηση. 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 E ως προς κάποια κλάση εργασιών T και μέτρο απόδοσης P, αν η απόδοσή του σε εργασίες από το T, όπως μετριέται από το P, βελτιώνεται μέσω της εμπειρίας E.*"

Η μηχανική μάθηση είναι στενά συνδεδεμένη και συχνά συγχέεται με υπολογιστική στατιστική [16], ένας κλάδος, που επίσης επικεντρώνεται στην πρόβλεψη μέσω της χρήσης των υπολογιστών. Έχει ισχυρούς δεσμούς με την μαθηματική βελτιστοποίηση, η οποία παρέχει μεθόδους, τη θεωρία και τομείς εφαρμογής. Η Μηχανική μάθηση εφαρμόζεται σε μια σειρά από υπολογιστικές εργασίες, όπου τόσο ο σχεδιασμός όσο και ο ρητός προγραμματισμός των αλγορίθμων είναι ανέφικτος. Παραδείγματα εφαρμογών αποτελούν τα φίλτρα Ανεπιθύμητων Μηνυμάτων (spam filtering) [18] και η Οπτική Αναγνώριση χαρακτήρων (OCR) [17].

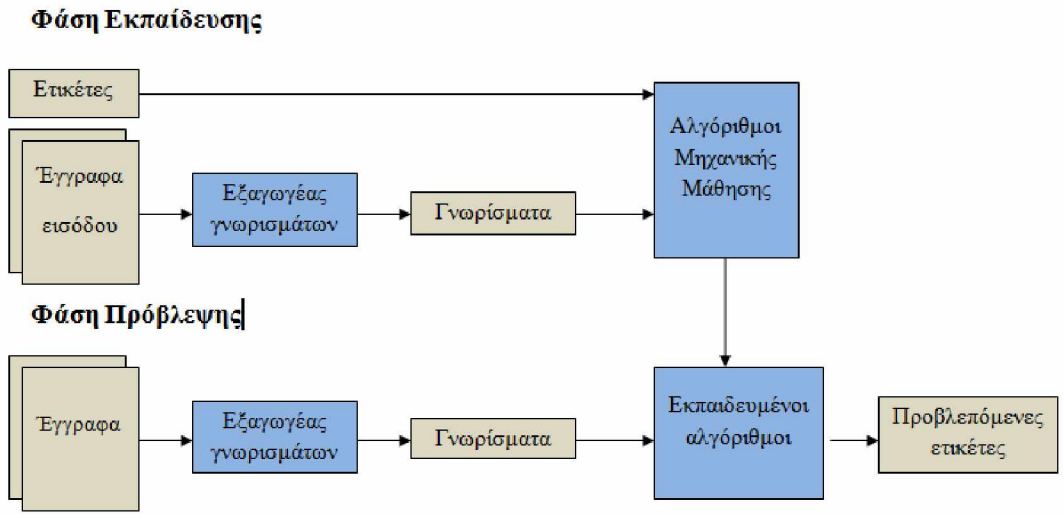
Σε αναλογία με την ανθρώπινη μάθηση, η Μηχανική Μάθηση χωρίζεται σε τρεις (3) βασικές κατηγορίες: *την επιβλεπόμενη μάθηση, την μη επιβλεπόμενη μάθηση και την ενισχυτική μάθηση.* Αναλυτικότερα:

- **Επιβλεπόμενη μάθηση** (Supervised Learning): στόχος αυτού του είδους μάθησης είναι ο χαρακτηρισμός των δεδομένων σύμφωνα με κάποια επιθυμητή τιμή από τα δεδομένα εκπαίδευσης. Κάθε αντικείμενα προς μελέτη αποτελειται από ένα σύνολο εισόδου (συνήθως διάνυσμα από χαρακτηριστικά) και μια τιμή εξόδου. Χρησιμοποιείται σε προβλήματα:
    1. Ταξινόμησης (Classification)
    2. Πρόγνωσης (Prediction)
    3. Διερμηνείας (Interpretation)
- **Μη επιβλεπόμενης μάθησης** (Unsupervised Learning): στόχος της είναι η ανακάλυψη πιθανής δομής που μπορεί να κρύβεται πίσω από μη

χαρακτηρισμένα δεδομενα. Εφόσον τα παραδείγματα τα οποία χρησιμοποιούνται δεν είναι χαρακτηρισμένα, δεν υπάρχει σφάλμα ή σήμα ανταμοιβής για να αξιολογηθούν οι πιθανές λύσεις. Αυτό είναι που διακρίνει την μη-επιβλεπόμενη μάθηση από την επιβλεπόμενη μάθηση και την ενισχυτική (ημι-επιβλεπόμενη) μάθηση. Ένα παράδειγμα μη επιτηρούμενης μάθησης είναι η συσταδοποίηση (clustering) η οποία χρησιμοποιείται για την ομαδοποίηση παρόμοιων δεδομένων σε συστάδες (clusters).

- **Ενισχυτική μάθηση** (Reinforcement Learning): είναι μία τεχνική μηχανικής μάθησης όπου το σύστημα μάθησης προσπαθεί να μάθει μέσα από την άμεση αλληλεπίδραση με το περιβάλλον. Το σύστημα έχει ως στόχο τη μεγιστοποίηση της "ανταμοιβής" που λαμβάνει.Η τεχνική αυτή είναι πολύ σημαντική για τα σημερινά παιχνίδια καθώς και την ρομποτική.

Για κάθε πρόβλημα της Μηχανικής Μάθησης υπάρχει ένας κατάλληλος τρόπος μάθησης του συστήματος ώστε να επιτευχθούν τα κατάλληλα αποτελέσματα. Όλοι οι αλγόριθμοι Μηχανικής Μάθησης αναπαράγουν την γνώση. Άλλοι αλγόριθμοι αναπαράγουν προϋπάρχουσα γνώση και άλλοι δέχονται σαν είσοδο μόνο παρατηρήσεις. Στην *Εικόνα 2.1* αποτυπώνεται ο γενικός τρόπος λειτουργίας ενός αλγορίθμου Μηχανικης Μάθησης. Η διαδικασία της "μάθησης" περιλαμβάνει μια την εκπαίδευση από ένα σύνολο δεδομένων εκπαίδευσης (training set), όπου εκεί εφαρμόζονται οι νοητικές διεργασίες του συστήματος. Έτσι επιτυγχάνεται η δημιουργία γνώσης. Στην συνέχεια, ο αλγόριθμος αξιολογείται ελέγχοντας την απόδοσή του σε ένα σύνολο δεδομένων ελέγχου (test set).



Εικόνα 2.1: Φάσεις Μηχανικής Μάθησης

Στην παρούσα εργασία θα ασχοληθούμε με την πρώτη (1η) κατηγορία Μηχανικής Μάθησης αυτή της επιτηρούμενης μάθησης, και συγκεκριμένα με το πρόβλημα της ταξινόμησης (classification) και της εξόρυξης πληροφορίας από κείμενο (text mining).

2.1.1 Εξόρυξη Γνώσης από Κείμενο

Η Εξόρυξη Γνώσης από Κείμενα μπορεί να ορισθεί ως μία διαδικασία εξαγωγής νέας πληροφορίας μέσω της οποίας ο χρήστης αλληλεπιδρά σε μία συλλογή κειμένων χρησιμοποιώντας ένα σύνολο εργαλείων ανάλυσης [19]. Στόχος του Text

Mining είναι η εξαγωγή χρήσιμης πληροφορίας από πηγές δεδομένων μέσω της αναγνώρισης και της διερεύνησης σημαντικών προτύπων. Στην περίπτωση βέβαια του Text Mining οι πηγές δεδομένων είναι συλλογές κειμένων και τα πρότυπα (patterns) αναζητούνται σε μη δομημένα κείμενα (unstructured texts)[20].

Η εξόρυξη γνώσης από κείμενο είναι μια υποκατηγορία της εξόρυξης δεδομένων και έχει αρκετά κοινά στοιχεία με την ανάλυση κειμένου (text analytics). Στόχος είναι η μετατροπή των κειμένων, που είναι εκφρασμένα σε φυσική γλώσσα και πολλές φορές μπορεί να ακολουθούν μια χαλαρή δομή, σε μια αναπαράσταση δεδομένων εύκολα αξιοποιήσιμη από έναν αλγόριθμο.

Η ανάλυση κειμένου είναι πολύ χρήσιμη προς διάφορες κατευθύνσεις. Εντούτοις, υπάρχει ένας τομέας που ονομάζεται Επεξεργασία Φυσικής Γλώσσας ή διαφορετικά Υπολογιστική Γλωσσολογία (Natural Language Processing) που σημειώνει σημαντική πρόοδο στην ανάλυση κειμένου. Στόχος του NLP είναι η δημιουργία προγραμμάτων σε ηλεκτρονικούς υπολογιστές που θα μπορούν να διαβάζουν, να κατανοούν ή και να δημιουργούν κείμενα σε φυσική γλώσσα. Επιπλέον, μέσω του NLP δίνεται η δυνατότητα απλοποίησης και προεπεξεργασίας κειμένων γραμμένων σε φυσική γλώσσα. Ως αποτέλεσμα, η εξαγωγή της γνώσης εξυπηρετείται αφού τα δεδομένα αναπαριστώνται σε μια πιο εύκολα αξιοποιήσιμη από τον αλγόριθμο μορφή.

2.1.2 Προεπεξεργασία Κειμένου

Η προεπεξεργασία κειμένου είναι ένα βασικό βήμα για την εξόρυξη της γνώσης. Το επεξεργασμένο κείμενο που προκύπτει χρησιμοποιείται στην συνέχεια για την εξαγωγή των χαρακτηριστικών που θα χρησιμοποιηθούν στην ταξινόμηση (feature extraction). Τα κύρια στάδια που ακολουθήθηκαν για την προεπεξεργασία είναι:

- **Διαχωρισμός λέξεων (Tokenization)**: η διαδικασία αυτή αναφέρεται στην γενικότερη τμηματοποίηση μιας ακολουθίας λέξεων, κατακερματιζοντας την σε κομμάτια που συνήθως είναι λέξεις ή φράσεις, τα λεγόμενα "tokens". Ο διαχωρισμός λέξεων μπορεί να γίνει απευθείας είτε να συνοδεύεται από την αφαίρεση των σημείων στίξης ανάμεσα στις λέξεις.
- **Φιλτράρισμα λέξεων (Filtering):** Το φιλτράρισμα αναφέρεται στην αφαίρεση λέξεων από το κείμενο. Συχνά τη διαδικασία αυτή την συναντάμε και με τον όρο "αφαίρεση λέξεων παύσης" ή αλλιώς "stopword removal". Οι λέξεις παύσης (stopwords) είναι λίστες των πιο κοινών λέξεων και άρθρων που συναντάμε σε κείμενα τα οποία δεν προσφέρουν εννοιολογικά στην κατηγοριοποίηση.
- **Λημματοποίηση λέξεων (Lemmatization):** που χρησιμοποιείται για τον εντοπισμό της ρίζας κάθε λέξης [21]. Θυμίζει αρκετά την τεχνική της **Αποκατάληξης (Stemming)**, με την μόνη διαφορά ότι στην δεύτερη η ρίζα που προκύπτει δεν είναι πάντα η πραγματική. Για παράδειγμα, οι λέξεις "agree", "agreed" και "agreeing" κατά την Λημματοποίηση θα δώσουν σαν ρίζα την λέξη "agree" ενώ κατά την Αποκατάληξη θα δώσουν "agr".

2.1.3 Μορφολογική Ανάλυση

Όπως είναι εμφανές, μορφολογική ανάλυση είναι η μελέτη στην μορφολογία των διαφορετικών λέξεων. Ήδη ο Leonard Bloomfield (1933: 207) επισημαίνει ότι

"οι γλώσσες διαφέρουν περισσότερο μορφολογικά παρά συντακτικά". Η μεγαλύτερη διαφορά στην μορφή των λέξεων είναι κυριώς κατά την κλίση τους, δηλαδή διαφέρει ανάμεσα στα μέρη του λόγου (part-of-speech). Κατά την διαδικασία της ετικετοποίησης σε μέρη του λόγου (Part-of-speech tagging) μια προκαθορισμένη από τους κανόνες της φυσικής γλώσσας ετικέτα τοποθετείται σε κάθε λέξη του κειμένου. Η κάθε ετικέτα αφορά σε ένα μέρος του λόγου π.χ. ρήμα, ουσιαστικό, αντωνυμία. Η πληροφορία αυτή μας είναι πολύ σημαντική, αρχικά γιατί κατατάσσει τις λέξεις σε διαφορετικές ομάδες με κοινά χαρακτηριστικά και χρήση και επιπλέον, επειδή συνεισφέρει στον γρήγορο εντοπισμό λαθών με βάση το συντακτικό και την διαπίστωση της ορθής χρήσης της γλώσσας σύμφωνα με τα συμφραζόμενα και τις περιβάλλουσες λέξεις. Για παράδειγμα, εάν διαπιστωθεί ότι μια λέξη είναι ουσιαστικό τότε η επόμενη θα πρέπει να είναι κατά πάσα πιθανότητα ρήμα, σύμφωνα με το συντακτικό. Ο Διαχωρισμός των λέξεων (Tokenization) ειναι πολλες φορές το πρώτο βήμα αυτής της διαδικασίας γιατί μπορεί να διαχωρίσει τις λέξεις αλλά και τα σημεία στίξης. Η στίξη μπορεί να πάρει τις δικές της ετικέτες για κάθε κατηγορία σημείων ή να αφαιρεθεί τελείως.

Tag	Description
CC	Coordinating conjunction
CD	Cardinal number
DT	Determiner
EX	Existential there
FW	Foreign word
IN	Preposition or subordinating conjunction
JJ	Adjective
JJR	Adjective, comparative
JJS	Adjective, superlative
LS	List item marker
MD	Modal
NN	Noun, singular or mass
NNS	Noun, plural
NNP	Proper noun, singular
NNPS	Proper noun, plural
PDT	Predeterminer
POS	Possessive ending
PRP	Personal pronoun

Tag	Description
PRP$	Possessive pronoun
RB	Adverb
RBR	Adverb, comparative
RBS	Adverb, superlative
RP	Particle
SYM	Symbol
TO	to
UH	Interjection
VB	Verb, base form
VBD	Verb, past tense
VBG	Verb, gerund or present participle
VBN	Verb, past participle
VBP	Verb, non3rd person singular present
VBZ	Verb, 3rd person singular present
WDT	Whdeterminer
WP	Whpronoun
WP$	Possessive whpronoun
WRB	Whadverb

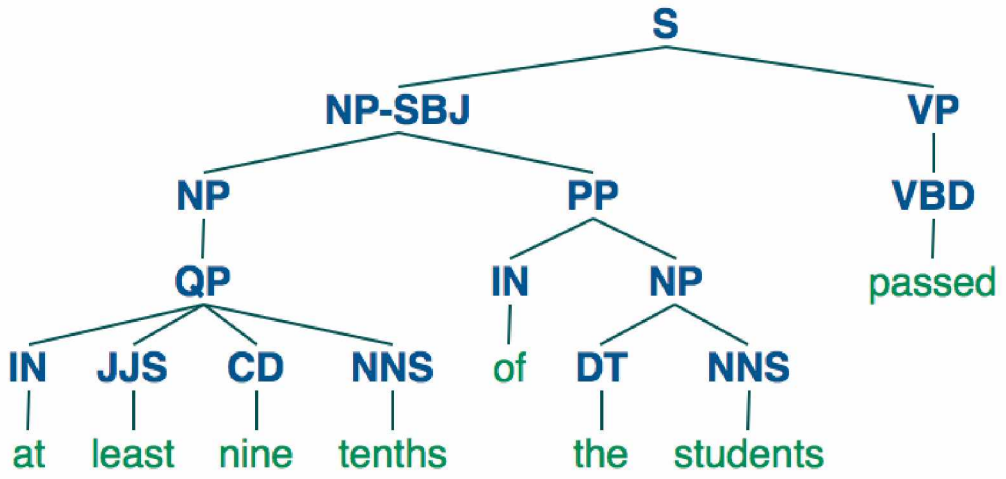
Εικόνα 2.2: Το σύνολο ετικετών του NLTK

2.1.4 Συντακτική ανάλυση

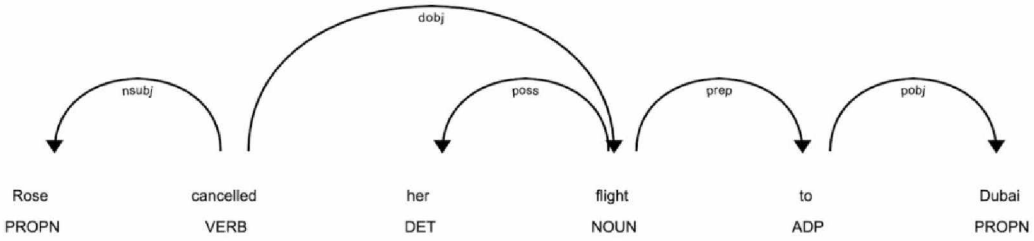
Συντακτική ανάλυση είναι η διαδικασία κατά την οποία υπολογίζουμε την συντακτική δομή συμβόλων π.χ. λέξεων. Οι συντακτικοί κανόνες της πρότασης αναπαρίστανται με την χρήση ιεραρχικών δενδροειδών δομών (parse trees). Στην *Εικόνα 2.3* δίνεται ένα παράδειγμα συντακτικής ανάλυσης της φράσης "At least nine tenths of the students passed" ("Τουλάχιστον εννέα δέκατα των μαθητών πέρασαν") με την χρήση δενδροειδούς μορφής. Παρατηρείται, ότι γίνεται χρήση των μερών του λόγου για την ταξινόμηση των επιμέρους λέξεων με συντακτικούς χαρακτηρισμούς.

Μια υποκατηγορία της συντακτικη ανάλυσης είναι η **ανάλυση των εξαρτήσεων** (dependency parsing), η οποία στοχεύει στην ανακάλυψη των νοηματικών εξαρτήσεων που διέπουν τις λέξεις της πρότασης. πως και στην κλασική συντακτική ανάλυση, οι εξαρτήσεις δομούνται με την χρήση δεντροειδών διαγραμμάτων που ονομάζονται "δέντρα εξαρτήσεων" (tree dependencies) και

εξυπηρετούν την οπτικοποίηση των εξαρτήσεων (*Εικόνα 2.4)*. Ακολουθώντας αυτή την λογική, οι νοηματικές εξαρτήσεις θα μπορούσαν να αναπαρασταθούν με τρεις οντότητες, μια να είναι η σχέση εξάρτησης (είδος), η δεύτερη να είναι η πηγή της εξάρτησης και η τρίτη το εξαρτώμενο μέλος.



Εικόνα 2.3: Ιεραρχική δενδροειδής δομή.



Εικόνα 2.4: Αναπαράσταση ενός δέντρου εξαρτήσεων.

Μια πιο απλοϊκή προσέγγιση του διαχωρισμού της πρότασης σε κομμάτια είναι και η χρήση ονοματικών οντοτήτων, όπως έχει προαναφερθεί. Η ιδέα είναι ότι από κάθε πρόταση δεν κρατάμε όλες τις συντακτικές λέξεις καθώς δεν αποτελούν όλες χρήσιμη πληροφορία. Προς αυτή την κατεύθυνση, γίνεται και πάλι χρήση της ετικετοποίησης, με την διαφορά ότι αυτή την φορά οι ετικέτες που χρησιμοποιούνται είναι διαφορετικές, και αφορούν σε οντότητες της φυσικής μας γλώσσας, π.χ. άνθρωπος, αριθμός, ημερομηνία κτλ. Όπως και στην κατηγοριοποίηση των μερών του λόγου, η ομαδοποίηση και εδώ είναι πολύ σημαντική καθώς όχι μόνο δίνονται πληροφορίες για τις ίδιες τις λέξεις αλλά και τις γειτονικές λέξεις της μελετούμενης. Στην *Εικόνα 2.5* φαίνεται ένα παράδειγμα ετικετοποίησης της φράσης "Hi, my name is Aman Kharwal. I am from India. I want to work with Google. Steve Jobs is my inspiration" ("Γεια, το όνομά μου είναι Αμάν Χαρβάλ. Είμαι από την Ινδία. Θέλω να δουλέψω στην Google. Ο Στιβ Τζομπς είναι η έμπνευσή μου").

Εικόνα 2.5: Ετικετοποίηση για την Αναγνώριση Ονοματικών Οντοτήτων μέσα σε πρόταση.

2.1.5 Ταξινόμηση Κειμένου

Όπως αναφέρθηκε και παραπάνω η Ταξινόμηση(classification) αφορά στη δημιουργία μοντέλων πρόβλεψης διακριτών τάξεων. Τα αντικείμενα που ταξινομούνται οργανώνονται σε σύνολα δεδομένων και μπορεί να είναι οποιασδήποτε μορφής π.χ. έγγραφα όπου το πρόβλημα ονομάζεται Κατηγοριοποίηση εγγράφων. Η παρούσα εργασία εστιάζει στην κατηγοριοποίηση ειδήσεων ως αληθείς ή ψευδείς υπό τη μορφή γραπτού κειμένου σε άρθρα.

Γενικά, η ταξινόμηση είναι ένα πρόβλημα που μπορεί να αντιμετωπιστεί ποικιλοτρόπως. Πριν την εξέλιξη και διάδοση των μεθόδων της Μηχανικής Μάθησης η διαδικασία απαιτούσε την ανθρώπινη ενέργεια καθώς γινόταν χειροκίνητα. Για παράδειγμα, η οργάνωση αρχείων σε έναν υπολογιστή γινόταν από τον ίδιο τον χρήστη. Στόχος της επιβλεπόμενης Ταξινόμησης λοιπόν, είναι η μείωση της ανθρώπινης προσπάθειας και της αυτοματοποίησης της διαδικασίας που πλέον γίνεται σε πολύ μικρότερο χρόνο και με λιγότερη ταλαιπωρία.

Για την λήψη αποφάσεων το σύστημα στηρίζεται σε Υποθετικούς Κανόνες (if-then rules) [22] της μορφής *if <hypothesis> then <action>*. Εάν αυτοί οι κανόνες οριστούν σωστά η κατηγοριοποίηση μπορεί να πραγματοποιηθεί με επιτυχία και συνήθως σημειώνοντας μεγάλη ορθότητα. Παράλληλα, ο αλγόριθμος ταξινόμησης που προκύπτει είναι αρκετά γενικός ώστε να μπορεί να επαναχρησιμοποιηθεί σε διαφορετικές περιπτώσεις ακόμη και αν τα δεδομένα εισόδου μεταβληθούν.

Η κατηγοριοποίηση αποτελεί την πρόβλεψη κλάσεων αντικειμένων. Μία **Κλάση** (**Class**) είναι ένα σύνολο δηλώσεων που αφορούν στην περιγραφή μιας συγκεκριμένης κατηγορίας αντικειμένων (object). Για τον διαχωρισμό αυτών συνεπώς πρέπει να υπάρχει τουλάχιστον μία κλάση. Όταν το πρόβλημα περιλαμβάνει μόνο δύο κλάσεις ονομάζεται δυαδικό ή διωνυμική Ταξινόμηση όπως για παράδειγμα το πρόβλημα των ψευδών και αληθών ειδήσεων. Κάθε κλάση είναι η κατηγορία που συγκεντρώνει κοινά χαρακτηριστικά αντικειμένων που διαφέρουν ειδοποιά σε σύγκριση με τα αντικείμενα που ανήκουν στην άλλη κλάση. Οι δύο κλάσεις λειτουργούν ανταγωνιστικά και ένας ταξινομητής που έχει εκπαιδευτεί σωστά μέσω των Υποθετικών κανόνων είναι σε θέση να ξεχωρίζει τις δύο κλάσεις με ευκρίνεια καθώς και αν παίρνει αποφάσεις για το ποιο αντικείμενο ανήκει στην κάθε κλάση.

Εάν κάθε έγγραφο που θέλουμε να κατηγοριοποιήσουμε μπορεί να ανήκει (με βάση την ομοιότητα των χαρακτηριστικών) σε παραπάνω από δύο κατηγορίες τότε το πρόβλημα ονομάζεται πρόβλημα Πολλαπλών ετικετών (Multi-label problem). Οι κατηγορίες μπορούν σε αυτή την περίπτωση να αλληλοεπικαλύπτονται γαυτό και η ταξινόμηση δεν παρουσιάζει πάντα τα επιθυμητά αποτελέσματα.

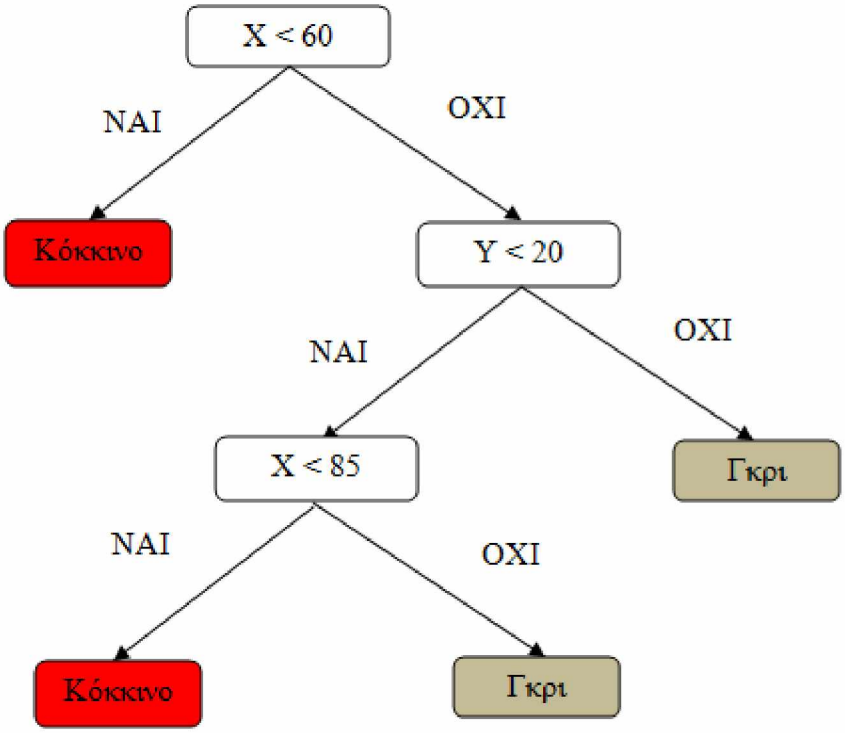
Η παρούσα εργασία εστιάζει στην πρώτη κατηγορία δηλαδή κατατάσσεται στην κατηγορία των διωνυμικών προβλημάτων. Στις παρακάτω υποενότητες θα

αναλυθούν οι τρεις ταξινομητές που χρησιμοποιήθηκαν και θα περιγραφεί η λειτουργία αλλά και οι μετρικές ανάλυσης της απόδοσής τους.

2.1.6 Ταξινομητής Random Forest

Τα Τυχαία Δάση Αποφάσεων (Random forests or random decision forests) [23] είναι μια μέθοδος εκμάθησης συνόλου για ταξινόμηση και παλινδρόμηση που λειτουργεί κατασκευάζοντας ένα πλήθος δέντρων αποφάσεων κατά το χρόνο εκπαίδευσης και εξάγοντας την τάξη της μέσης πρόβλεψης των μεμονωμένων δέντρων. Τα τυχαία δάση συνήθως ξεπερνούν σε απόδοση τα δέντρα αποφάσεων (decision trees).

Καθώς ο ταξινομητής αποτελείται από επιμέρους **Δέντρα Αποφάσεων**[5] ας εμβαθυνουμε πρώτα στην στην λειτουργία αυτών με την χρήση ενός παραδείγματος *(Εικόνα 2.6)*.



Εικόνα 2.6: Λειτουργία ενός Δέντρου αποφάσεων

[5] Στην εξόρυξη δεδομένων, ένα δέντρο απόφασης είναι ένα προγνωστικό μοντέλο το οποίο μπορεί να χρησιμοποιηθεί για να αναπαραστήσει τους ταξινομητές καθώς και τα μοντέλα παλινδρόμησης. Όταν ένα δέντρο απόφασης χρησιμοποιείται για διαδικασίες κατάταξης, είναι πιο κατάλληλο να αναφέρεται ως ένα δέντρο ταξινόμησης.Τα δέντρα ταξινόμησης χρησιμοποιούνται για την ταξινόμηση ενός αντικειμένου ή ενός γεγονότος σε ένα προκαθορισμένο σύνολο κατηγοριών με βάση τις αξίες των χαρακτηριστικών του.

Ο στόχος του Δέντρου είναι ο διαχωρισμός των διαφορετικών κλάσεων, χωρίς αυτές να είναι εμφανείς με μια απλή ματιά. Γι αυτό και τα Δέντρα υπακούουν σε κανόνες σύμφωνα με τους οποίους παίρνουν αποφάσεις για το ποιά είναι η κατηγορία στην οποία ανήκει το κάθε στοιχείο προς ταξινόμηση. Οι **κανόνες** παίρνουν την μορφή ερωτήσεων οι οποίες απαντώνται με "Ναι" ή "Όχι" (εάν το πρόβλημα είναι δυαδικό). Για παράδειγμα, το Δέντρο ξεκινά με έναν αρχικό κόμβο που ονομάζεται Ρίζα (Root node) όπου γίνεται η πρώτη ερώτηση, δηλαδή εδώ "Είναι το Χ μικρότερο του 60;". Εάν ναι τότε όλα τα στοιχεία που εκπληρώνουν αυτό τον περιορισμό κατατάσσονται στην Κλάση κόκκινο. Τα υπόλοιπα πρέπει να διερευνηθούν σε επόμενη ερώτηση για να ταξινομηθούν σε κλάση. Με επόμενη ερώτηση μερικά στοιχεία τοποθετούνται στην Κλάση γκρι και στην συνέχεια έχουμε την τελευταία ερώτηση που παίρνει απόφαση και για τα εναπομείναντα στοιχεία. Ως αποτέλεσμα, οι αλλεπάλληλες ερωτήσεις του Δέντρου ταξινομούν όλα τα στοιχεία σε δύο κλάσεις κόκκινο και γκρι. Οι κλάσεις βρίσκονται στα φύλλα (Leaf nodes) του Δέντρου.

Τα Τυχαία Δάση Αποφάσεων (Random forest classifier) στηρίζονται λοιπόν στα αποτελέσματα των Δέντρων Αποφάσεων καθώς λειτουργούν σαν μέσος όρος των δέντρων. Πιο συγκεκριμένα, κάθε μεμονωμένο δέντρο εξάγει ένα συμπέρασμα για τις κλάσεις και το πλειοψηφικά επικρατέστερο πόρισμα κερδίζει σύμφωνα με τα Τυχαία Δάση (*Εικόνα 2.7).*

Το μυστικό της επιτυχίας του αλγορίθμου αυτού είναι ο πλειοψηφικός τρόπος λειτουργίας τους, καθώς διατηρείται η αυτονομία των επιμέρους στοιχείων του ταξινομητή. Σε αντίθεση με άλλα συστήματα όπου γίνεται άθροισμα των αποτελεσμάτων, η τεχνική που λαμβάνει υπόψη μόνο την επικρατέστερη άποψη και όχι το σύνολο των απόψεων παρεχει καλύτερη ακρίβεια στην πρόβλεψη όπως ερευνήθηκε [24].

Εικόνα 2.7: Επιμέρους Δέντρα Αποφάσεων που λειτουργούν μέσα στα Τυχαία Δάση αποφάσεων.

Για να διασφαλιστεί λοιπόν αυτή η αρχή της ανεξαρτησίας κάθε μεμονωμένης μονάδας που απαρτίζει το Τυχαίο Δάσος ακολουθούνται δύο κανόνες:

1. **Bagging (Bootstrap Aggregation):** Τα Δέντρα αποφάσεων είναι ιδιαίτερα ευαίσθητα σε αλλαγές στο σύνολο εκπαίδευσης που έχουν μάθει, χαρακτηριστικό που εκμεταλλεύεται ο ταξινομητής. Έτσι, αντί να ανατεθεί το ίδιο σύνολο εκπαίδευσης σε κάθε Δέντρο ή να χωρίσουμε το σύνολο εκπαίδευσης σε κομμάτια όσα και ο αρθμός των Δέντρων και να δοθεί σε καθένα από αυτά διαφορετικό κομμάτι, ο αλγόριθμος βασίζεται στην τυχαιότητα. Συγκεκριμένα, δίνεται σε κάθε Δέντρο ίδιο σε μέγεθος υποσύνολο του ίδιου συνόλου εκπαίδευσης επιλέγοντας τυχαία στοιχεία και επανατοποθετώντας τα ύστερα στο κοινό σύνολο. Για παράδειγμα, έχοντας το σύνολο εκπαίδευσης [1,2,3,4,5,6] θα μπορούσε να δοθεί σε ένα Δέντρο το κομμάτι [1,2,2,3,4,4] και σε ένα άλλο [3,3,3,3,3,3]. Έτσι, διασφαλίζεται η αυτονομία και η ποικιλομορφία των Δέντρων που δίνει επιθυμητά αποτελέσματα στον αλγόριθμο.
2. **Feature Randomness:** Σε ένα τυπικό Δέντρο απόφασης, κατά τον διαχωρισμό των κόμβων, δηλαδή την ώρα της απόφασης, το Δέντρο μπορεί να συμβουλευτεί για να ταξινομήσει σωστά όποιο χαρακτηριστικό από τον χώρο των χαρακτηριστικών θεωρεί ότι εξυπηρετεί καλύτερο στο ερώτημα που τέθηκε. Αντίθετα, στα Δέντρα που χρησιμοποιούνται στα Τυχαία Δάση, δεν υπάρχει δυνατότητα επιλογής του χαρακτηριστικού πάνω στο οποίο θα στηριχθεί η απόφαση καθώς αυτό καθορίζεται τυχαία με επανατοποθέτηση όπως στην προηγούμενη περίπτωση.

2.1.7 Ταξινομητής Naive Bayes

Στη στατιστική, οι Αφελείς ταξινομητές Bayes (Naive Bayes Classifiers) είναι μια οικογένεια απλών "πιθανολογικών ταξινομητών" που βασίζονται στην εφαρμογή του θεωρήματος του Bayes με ισχυρές (αφελείς-naive) παραδοχές ανεξαρτησίας μεταξύ των χαρακτηριστικών. Είναι από τα πιο απλά μοντέλα δικτύου Bayesian, αλλά σε συνδυασμό με την εκτίμηση της πυκνότητας του πυρήνα, μπορούν να επιτύχουν υψηλότερα επίπεδα ορθότητας.

Υπάρχουν δύο βασικά μοντέλα κατηγοριοποίησης που χρησιμοποιούνται στον Naive Bayes αλγόριθμο, τα Multi-variate Bernouli και Multinomial. Και τα δύο στηρίζονται στην κατανομή των λέξεων στο κείμενο προς κατηγοριοποίηση. Στην περίπτωση του Multivariate Bernouli το έγγραφο αποτυπώνεται ως ένα διάνυσμα από γνωρίσματα που φέρουν δυαδικές τιμές αναπαριστώντας έτσι την ύπαρξη ή όχι της κάθε λέξης μέσα στο έγγραφο ενώ δεν λαμβάνεται υπόψη η συχνότητα εμφάνισης της κάθε λέξης. Στο Multinomial μοντέλο χρησιμοποιούνται οι συχνότητες εμφάνισης των λέξεων.

Περιληπτικά, ο Naive Bayes είναι ένα μοντέλο πιθανότητας. Δεδομένου ότι υπάρχει ένα στοιχείο προς ταξινόμηση, που αντιπροσωπεύεται από ένα διάνυσμα $x= (x_1, x_2, .., x_n)$ το οποίο αναπαριστά n χαρακτηριστικά, παραχωρεί δεσμευμένες πιθανότητες ως εξής:

$$p(Ck|xi) = \frac{p(Ck)\cdot p(xi|Ck}{p(xi)} \qquad (2.1)$$

που εκφράζουν την πιθανότητα το έγγραφο με διάνυσμα γνωρισμάτων x_i να ανήκει σε μια κατηγορία γνωρισμάτων C_k.

- C_k η κατηγορία από ένα σύνολο κατηγοριών $\{C_1, C_2 ... C_n\}$
- x_i το διάνυσμα γνωρισμάτων από ένα έγγραφο $i \in \{1,2..,n\}$
- Η πρότερη πιθανότητα $P(C_k)$ της κατηγορίας k, (η πιθανότητα να συναντήσουμε αυτή την κατηγορία).

- Και η $P(x_i|C_k)$ η πιθανότητα του διανύσματος γνωρισμάτων x_i δεδομένης της κατηγορίας C_k .

Δηλαδή, σε μια απλοποιημένη μορφή θα μπορούσε να γραφεί ως εξής:

$$\text{Μεταγενέστερο} = \frac{\text{προηγούμενο} \cdot \text{πιθανότητα}}{\text{απόδειξη}}$$

Ενδιαφέρον παρουσιάζει μόνο ο αριθμητής αυτού του κλάσματος αφού ο παρονομαστής δεν εξαρτάται από το C και τα x_i δίνονται. Άρα πρακτικά ο παρονομαστής μένει σταθερός. Ο αριθμητής είναι ισοδύναμος με το κοινό μοντέλο πιθανότητας: $p(C_k, x_1, \ldots, x_n)$, που μπορεί να γραφτεί και ως:

$$p(C_k,x_1, \ldots, x_n) = p(x_1|x_2, \ldots, x_n, C_k)\cdot p(x_2|x_3, .., x_n, C_k)\cdot p(x_n|C_k)\cdot p(C_k) \quad (2.2)$$

χρησιμοποιώντας τον κανόνα της αλυσίδας.

Η πιθανότητα για κάθε ένα από τα γνωρίσματα υπολογίζεται βάση του κανόνα *maximum-likelihood* και ορίζεται ως:

$$\hat{p}(x_i|C_k)=\frac{Nx_i,Ck}{NC_k} \quad (2.3)$$

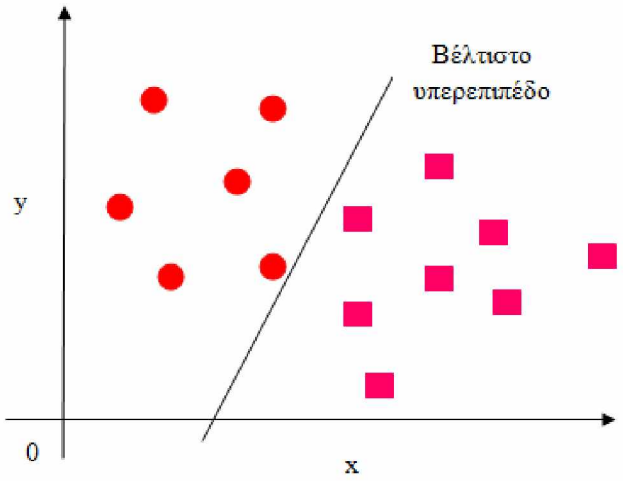
- $N_{xi,Ck}$ η συχνότητα εμφάνισης του γνωρίσματος x_i σε έγγραφα που ανήκουν στην κατηγορία C_k
- N_{Ck} ο συνολικός αριθμός γνωρισμάτων που συναντώνται στην κατηγορία C_k .

Στόχος είναι η μεγιστοποίηση της πιθανότητας $P(C_k|x_i)$ μέσω των δεδομένων εκπαίδευσης ώστε να προβλεφθεί η σωστή κατηγορία.

2.1.8 Ταξινομητής Support Vector Machine

Στη μηχανική μάθηση, οι Μηχανές Διανυσμάτων Υποστήριξης-Φορέα (Support Vector Machines) είναι εποπτευόμενα μοντέλα μάθησης με συναφείς αλγόριθμους μάθησης που αναλύουν δεδομένα για ταξινόμηση και ανάλυση παλινδρόμησης. Όπως και οι προαναφερθέντες, είναι αλγόριθμος εποπτευόμενης μηχανικής μάθησης.

Λαμβάνοντας υπόψη ένα σύνολο παραδειγμάτων εκπαίδευσης, όπου το καθένα επισημαίνεται ότι ανήκει σε μία από τις δύο κατηγορίες, ο αλγόριθμος εκπαίδευσης SVM δημιουργεί ένα μοντέλο που εκχωρεί νέα παραδείγματα στην κάθε περίπτωση, καθιστώντας τον έναν μη πιθανοτικό δυαδικό γραμμικό ταξινομητή.Το SVM αντιστοιχεί παραδείγματα εκπαίδευσης σε σημεία στον χώρο έτσι ώστε να μεγιστοποιείται το κενό μεταξύ των δύο κατηγοριών. Στην συνέχεια, όσα νέα παραδείγματα εισέρχονται προς χαρτογράφηση, προβλέπεται ότι ανήκουν στην αντίστοιχη κατηγορία ανάλογα με το ποια πλευρά του χώρου εκατέρωθεν του κενού, ανήκουν. Η περιοχή που διαχωρίζει επαρκώς τις δύο κατηγορίες στις δύο διαστάσεις ονομάζεται **υπερεπίπεδο** (hyperspace) και συμβολίζεται με μια ευθεία γραμμή. Το πλήθος των ευθειών που χρειάζεται για να διαχωριστούν τα σημεία εξαρτάται από το πλήθος των κλάσεων που έχουμε στα δεδομένα μας. Για παράδειγμα, εάν έχουμε δύο κατηγορίες τότε ο ελάχιστος αριθμός τέτοιων ευθειών που χρειάζεται για τον διαχωρισμό είναι μία (*Εικόνα 2.8).* Το SVM παίρνει σαν είσοδο την διανυσματική αναπαράσταση των γνωρισμάτων ενός εγγράφου και προσπαθεί να προβλέψει την κατηγορία στην οποία ανήκει.



Εικόνα 2.8: : Αναπαράσταση υπερεπιπέδων διαχωρισμού.

Έστω το παρακάτω σύνολο διανυσμάτων εκπαίδευσης δύο κατηγοριών:

$$D=\{(x_1, y_1)\ (x_2, y_2)\ ,\ldots\ ,(x_l, y_l)\},\ x\in\mathbb{R} \qquad (2.4)$$

$$\text{όπου } \{ y_k=1\ \alpha\nu\ x_k \in class\ A$$

$$y_k=-1\alpha\nu\ x_k \in class\ B\} \qquad (2.5)$$

Το υπερεπίπεδο διαχωρισμού ορίζεται από την παρακάτω εξίσωση:

$$w^T \cdot x+b=0 \qquad (2.6)$$

όπου:

- w είναι ένα διάνυσμα βαρών κάθετο στο υπερεπίπεδο και ορίζει τον προσανατολισμό του.
- b η τιμή κατωφλίου που ορίζει την παράλληλη μετατόπιση του υπερεπιπέδου.

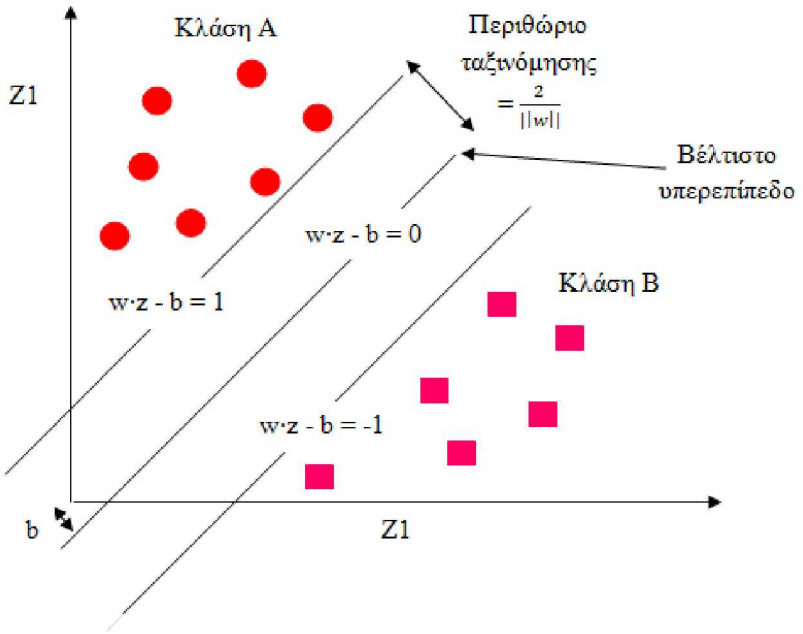
Επομένως από τα παραπάνω προκύπτει ότι για δείγμα που ανήκει στην κλάση Α ισχύει:

$$w^T \cdot x\ 1+b>0 \qquad (2.7)$$

Αναπαράσταση υπερεπιπέδων διαχωρισμού. και για ένα δείγμα που ανήκει στην κλάση Β:

$$w^T \cdot x\ 2+b<0 \qquad (2.8)$$

Τα στοιχεία δύο κλάσεων συνεπώς μπορούν να διαχωρίζονται από ένα μεγάλο πλήθος υπερεπιπέδων. Για την εύρεση του του καταλληλότερου χρησιμοποιείται η έννοια του **Περιθωρίου ταξινόμησης** (margin). Ως περιθώριο ταξινόμησης ορίζεται η ελάχιστη κάθετη απόσταση ενός σημείου μιας κλάσης από το υπερεπίπεδο διαχωρισμού. Το βέλτιστο υπερεπίπεδο είναι αυτό που μεγιστοποιεί το περιθώριο διαχωρισμού (*Εικόνα 2.9).*



Εικόνα 2.9: Έννοια του περιθωρίου ταξινόμησης και βέλτιστου υπερεπιπέδου.

2.2 Μετρικές Εκτίμησης Απόδοσης

Για να μετρηθεί η απόδοση ενός αλγορίθμου κατηγοριοποίησης, δηλαδή η ορθότητα των προβλέψεων που κάνει κατά την κατηγοριοποίηση ενός άγνωστου συνόλου εγγράφων, χρησιμοποιούνται κατάλληλες μετρικές εκτίμησης. Οι μετρικές αυτές διαφέρουν ανάλογα με τον τύπο κατηγοριοποίησης και τον αλγόριθμο που έχει επιλεγεί. Στην περίπτωση της κατηγοριοποίησης κειμένου και συγκεκριμένα στην διωνυμική κατηγοριοποίηση οι ευρέως χρησιμοποιούμενες μετρικές είναι οι:

- **Ορθότητα** (Accuracy).
- **Ανάκληση** (Recall).
- **Ακρίβεια** (Precision).
- **F1-Score**

Ο *πίνακας σύγχυσης (Confusion matrix)* είναι ένας τετραγωνικός *(m×m, m=2)* πίνακας, όπου το *(i, j)* στοιχείο του ισούται με το πλήθος των σημείων που, ενώ προέρχονται από την κλάση *i*, καταχωρούνται στην κλάση *j*. Πρόκειται για έναν πίνακα που παρέχει πληροφορίες σχετικά με το αν κάποιες κλάσεις έχουν τη τάση να συγχέονται με άλλες κλάσεις. Οι στήλες του πίνακα αντιστοιχούν στις κατηγορίες που είναι ήδη γνωστές ενώ οι γραμμές σε αυτές που ο ταξινομητής έχει προβλέψει. Σε κάθε κελί του πίνακα ανήκει ένα ποσοστό:

- **Αληθώς θετικά** (True Positives- TP): σε αυτή την κατηγορία ανήκουν τα αντικείμενα που προβλέφθηκαν από την διαδικασία ταξινόμησης σαν θετικά και η πραγματική τάξη στην οποία ανήκουν είναι πράγματι η θετική.
- **Αληθώς αρνητικά** (True Negatives- TN): σε αυτή την κατηγορία ανήκουν τα αντικείμενα που προβλέφθηκαν από την διαδικασία ταξινόμησης σαν αρνητικά και η πραγματική τάξη στην οποία ανήκουν είναι πράγματι η αρνητική.
- **Ψευδώς θετικα** (False Positives- FP): σε αυτή την κατηγορία ανήκουν τα αντικείμενα που προβλέφθηκαν από την διαδικασία ταξινόμησης σαν θετικά αλλά αντιθέτως η πραγματική τάξη στην οποία ανήκουν είναι η αρνητική.

- **Ψευδώς αρνητικά** (False Negatives- FN): σε αυτή την κατηγορία ανήκουν τα αντικείμενα που προβλέφθηκαν από την διαδικασία ταξινόμησης σαν αρνητικά αλλά αντιθέτως η πραγματική τάξη στην οποία ανήκουν είναι η θετική.

Στο πρόβλημα που μελετάται σε αυτή την μελέτη έχουμε αληθείς και αναληθείς ειδήσεις οι οποίες μπορούν να κωδικοποιηθούν με ετικέτες ως εξής:

1. Αληθής είδηση (0): Θετική ή Positive
2. Ψευδής ειδηση (1): Αρνητική ή Negative

Άρα για το συγκεκριμένο πρόβλημα υπό μελέτη οι προβλέψεις διαμορφώνονται ως εξής:

- **Αληθώς θετικά** (True Positives- TP): σε αυτή την κατηγορία ανήκουν τα άρθρα που προβλέφθηκαν από την διαδικασία ταξινόμησης σαν αληθή και στην πραγματικότητα είναι πράγματι θετικά.
- **Αληθώς αρνητικά** (True Negatives- TN): σε αυτή την κατηγορία ανήκουν τα άρθρα που προβλέφθηκαν από την διαδικασία ταξινόμησης σαν ψευδή και στην πραγματικότητα είναι πράγματι ψευδή.
- **Ψευδώς θετικα** (False Positives- FP): σε αυτή την κατηγορία ανήκουν τα άρθρα που προβλέφθηκαν από την διαδικασία ταξινόμησης σαν αληθή αλλά αντιθέτως στην πραγματικότητα είναι ψευδή.
- **Ψευδώς αρνητικά** (False Negatives- FN): σε αυτή την κατηγορία ανήκουν τα άρθρα που προβλέφθηκαν από την διαδικασία ταξινόμησης σαν ψευδή αλλά αντιθέτως στην πραγματικότητα είναι αληθή.

Άρα ο πίνακας σύγχυσης διαμορφώνεται ως εξής:

		Πραγματικές τιμές	
		Θετικά (0)	**Αρνητικά (1)**
Προβλεπόμενες τιμές	**Θετικά (0)**	**TP**	**FP**
	Αρνητικά (1)	**FN**	**TN**

Εικόνα 2.10: Ο πίνακας σύγχυσης

2.2.1 Ορθότητα

Ως Ορθότητα(Accuracy) ορίζεται ο συνολικός αριθμός των σωστών προβλέψεων που έκανε ο αλγόριθμος έναντι του συνόλου των προβλέψεων.

$$Accuracy = \frac{TN + TP}{TP+FP+FN+TN} \qquad (2.9)$$

2.2.2 Ανάκληση

Με την Ανάκληση(Recall) μπορούμε να βρούμε το ποσοστό των αρθρων που είναι πραγματικά Ψευδή:

$$Recall = \frac{TN}{TN + FP} \qquad (2.10)$$

Σημειώνεται ότι στον παρονομαστή του κλάσματος τοποθετείται τόσο ο αριθμός των Αληθώς Αρνητικών αλλά και ο αριθμός των Ψευδώς Θετικών περιπτώσεων αφού έτσι αθροίζεται ο συνολικός αριθμός όλων των ψευδών άρθρων.

2.2.3 Ακρίβεια

Με τον όρο Ακρίβεια (Precision) περιγράφουμε τον αριθμό των άρθρων που ο ταξινομητής κατηγοριοποίησε ως Ψευδή και είναι πραγματικά Ψευδή.

$$Precision = \frac{TP}{TP + FP} \qquad (2.11)$$

Ο αριθμητής του κλάσματος αναφέρεται στα άρθρα που ορθώς προβλέφθηκαν ως Ψευδή (TP) και στον παρονομαστή υπάρχουν όλα τα άρθρα που είτε σωστά (TP) είτε λανθασμένα (FP), προβλέφθηκαν ως Ψευδή.

2.2.4 F1 Score

Η μετρική F1 ουσιαστικά αποτελεί τον αρμονικό μέσο των Precision και Recall και δίνεται από τον παρακάτω τύπο:

$$F1 = \frac{2 \cdot Precision \cdot Recall}{Precision + Recall} \qquad (2.12)$$

Οι μετρικές που περιγράφηκαν είναι πολύ σημαντικές καθώς βοηθούν στην διεξαγωγή έγκυρων συμπερασμάτων για την επίδοση του μοντέλου. Το ιδανικό σενάριο για ένα τέλεια αποδοτικό σύστημα θα ήταν το πλήθος των Ψευδώς Θετικών (FP) και Ψευδώς Αρνητικών (FN) να ηταν μηδενικό. Αυτό πρακτικά θα σήμαινε ότι η Ορθότητα (Accuracy) θα ήταν 100%, πράγμα που είναι πολύ δύσκολο έως και αδύνατο να επιτευχθεί.

Ωστόσο, σκοπός των κατασκευαστών τέτοιων μοντέλων είναι να λαμβάνουν υπόψη τους αυτές τις μετρικές και να τις βελτιώνουν όσο το δυνατόν γίνεται γιατί έτσι βελτιώνουν την ποιότητα του συστήματος και την ικανότητα του στην λήψη αποφάσεων. Σε κάθε περίπτωση όμως η υπερμετρη εστίαση στην βελτίωση των μετρικών για ένα συγκεκριμένο σύνολο δεδομένων, δηλαδή για ένα συγκεκριμένο πρόβλημα μπορεί να φέρει τα αντίθετα αποτελέσματα, δηλαδή να προκύψει ***Υπερμοντελοποίηση.*** Η Υπερμοντελοποίηση (Overfitting) στα στατιστικά στοιχεία, είναι η υπερβολική προσαρμογή και συνεπάγεται την παραγωγή μιας ανάλυσης που αντιστοιχεί πολύ στενά ή ακριβώς σε ένα συγκεκριμένο σύνολο δεδομένων. Μπορεί συνεπώς να μην χωρέσει πρόσθετα δεδομένα ή να προβλέψει αξιόπιστα τις

μελλοντικές παρατηρήσεις. Η Υπερμοντελοποίηση είναι ένα λάθος που εμπίπτει στην παρατήρηση των μετρικών χωρίς την επαληθευση τους με άλλα παραδείγματα. Αντίθετα, η ***Γενίκευση*** είναι η ικανοτητα του συστήματός να λαμβάνει αποφάσεις για μελλοντικά προβλήματα έχοντας "μάθει" σωστά από την εκπαίδευση που προηγήθηκε. Όσο καλύτερη ικανότητα γενίκευσης έχει το μοντέλο, τόσο ορθότερα εκπαιδευμένο είναι.

3 ΔΕΔΟΜΕΝΑ ΚΑΙ ΕΞΑΓΩΓΗ ΓΝΩΡΙΣΜΑΤΩΝ

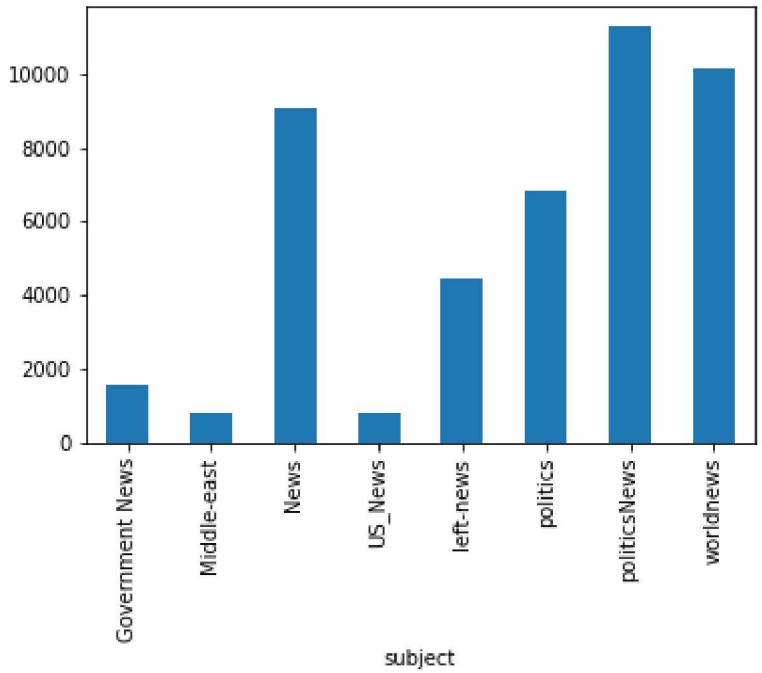
Σε αυτό το Κεφάλαιο γίνεται εστιασμένη μελέτη σε δύο βασικά βήματα της Μηχανικής Μάθησης, την επιλογή και διαμόρφωση του συνόλου των δεδομένων και στην επιλογή των κατάλληλων γνωρισμάτων που χαρακτηρίζουν τα έγγραφα προς μελέτη.

3.1 Σύνολο δεδομένων

Σε πολλές μελέτες της Μηχανικής Μάθησης η εύρεση ενός κατάλληλου συνόλου δεδομένων είναι μια ιδιαίτερα χρονοβόρα και δύσκολη διαδικασία. Ωστόσο, με την αυξανόμενη άνθιση του ενδιαφέροντος προς αυτόν τον κλάδο τα τελευταία χρόνια, είναι πλέον προσβάσιμα σύνολα δεδομένων που είναι έγκυρα και αξιόπιστα από δημόσιους ιστούς όπως το *Kaggle*. Η Kaggle[6], θυγατρική της Google LLC, είναι μια διαδικτυακή κοινότητα επιστημόνων δεδομένων και επαγγελματιών μηχανικής μάθησης. Παρόλο που αποτελεί αναμφισβήτητα μια εύχρηστη πηγή δεδομένων, το μειονέκτημά της είναι ότι σε πολλά σύνολα δεν αναφέρεται η πηγή προέλευσης του κάθε συνόλου.

Για την ταξινόμηση άρθρων, θα πρέπει να είναι διαθέσιμο σύνολο από έγγραφα με ειδησεογραφικό περιεχόμενα, που να έχουν προηγουμένως χαρακτηριστεί ως αληθή ή ψευδή. Τέτοιου τύπου σύνολα δεδομένων παρέχει το Kaggle, χωρίς ωστόσο να δίνει πληροφορίες για το ειδησεογραφικό σάιτ από το οποίο προέρχονται. Έτσι, καθίσταται δύσκολο να εξαχθεί κάποιο συμπέρασμα για τις πηγές των άρθρων όσον αφορά την ποιότητα και την αλήθεια των ειδήσεων που παράγουν. Οι πηγές που θεωρούνται αξιόπιστες είναι η *Forbes*[7] και η *Washington* [8] μεταξύ άλλων. **Η ανάλυση της εγκυρότητας σε αυτή την εργασία γίνεται σε επίπεδο άρθρου και όχι σε επίπεδο ιστοτόπου.** Για πληρότητα στην εξέταση του ζητήματος της εγκυρότητας ενός άρθρου, τα άρθρα του συνόλου δεδομένων είναι από ποικίλες θεματολογίες όπως η πολιτική και η παγκόσμια επικαιρότητα *(Διάγραμμα 3.1)*.

[6] *www.kaggle.com*
[7] *www.forbes.com*
[8] *www.washigtonpost.com*



Διάγραμμα 3.1: Θεματολογίες των άρθρων που επιλέχθηκαν για την εργασία.

Το σύνολο δεδομένων πάνω στο οποίο στηρίζεται η παρούσα μελέτη, κατασκευάστηκε ύστερα από ένωση δύο επιμέρους, ένα που αποτελείται μόνο από Ψευδείς και ένα μόνο με Αληθείς ειδήσεις. Οι ειδήσεις των υποσυνόλων έχουν αναμειχθεί για την δημιουργία ενός κοινού συνόλου με ετικέτες που δηλώνουν σε ποια κατηγορία ανήκει το κάθε άρθρο. Το σύνολο παρέχει πληροφορίες για την ημερομηνία έκδοσης κάθε άρθρου και το είδος στο οποίο ανήκει, χαρακτηριστικά τα οποία δεν χρειάζονται για την μελέτη παρά μόνο για την διασφάλιση της διαφορετικότητας στα άρθρα. Γι' αυτό και αφαιρέθηκαν από το σύνολο κατά την ανάλυση. Οι τίτλοι των άρθρων έχουν ενσωματωθεί στο κύριο περιεχόμενο και αντιμετωπίζονται όπως και το υπόλοιπο κείμενο. Τέλος, για την προεργασία των δεδομένων ακολουθήθηκαν όλες οι τεχνικές που αναφέρθηκαν στην *υποενότητα 2.1.2* με την παράλειψη της Λημματοποίησης και της Αποκατάληξης καθώς ύστερα από δοκιμές με τους ταξινομητές, παρατηρήθηκε ότι τα κείμενα που έχουν υποστεί επεξεργασία με αυτές τις τεχνικές δεν κατηγοριοποιούνται με την ίδια ακρίβεια με αυτά που δεν έχουν.

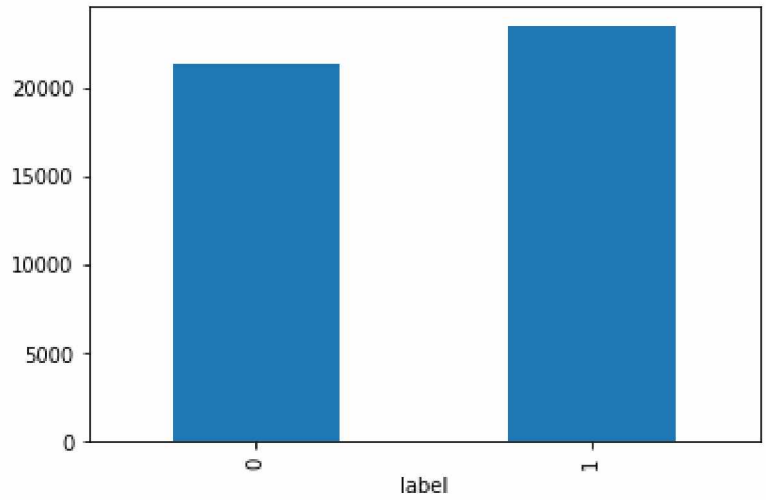
3.2 Επιλογή χαρακτηριστικών

Η γραπτή φυσική γλώσσα σε μορφή κειμένου δεν μπορεί να χρησιμοποιηθεί αυτούσια για την εκπαίδευση ενός αλγορίθμου ταξινόμησης, καθώς ο αλγόριθμος είναι προγραμματισμένος να αναγνωρίζει αριθμούς και χώρους διανυσμάτων. Τα κείμενα θα πρέπει να μετασχηματιστούν σε κατάλληλη μορφή που να είναι φιλική προς τον αλγόριθμο να αντιληφθει και να μάθει. Ο πιο διαδεδομένος τρόπος είναι η αναπαράσταση ενός κειμένου ως **διάνυσμα βαρών.** Ανάλογα με την φύση του κάθε προβλήματος τα βάρη μπορούν να πάρουν διάφορες μορφές και να αντιπροσωπεύουν τα διάφορα χαρακτηριστικά. Για παράδειγμα, τα βάρη θα μπορούσαν να πάρουν την μορφή [0,1] ανάλογα με την ύπαρξη ή όχι μιας λέξης μέσα στο κείμενο. Σε μια άλλη περίπτωση το διάνυσμα θα μπορούσε να αναπαριστά την συχνότητα εμφάνισης των διαφόρων λέξεων.

Από διάφορες μελέτες που έχουν γίνει, φαίνεται ότι ο παραπλανητικός λόγος διαφέρει από τον αληθή σε διάφορα μορφολογικά και συντακτικά χαρακτηριστικά. Πιο συγκεκριμένα, οι τίτλοι των ψευδών άρθρων φαίνεται να είναι μεγαλύτεροι σε μέγεθος αλλά και με περισσότερα ονόματα ενώ η χρήση αντωνυμιών είναι συχνότερη στις έγκυρες ειδήσεις. Λαμβάνοντας υπόψη όλα τα παραπάνω, έχουν εξαχθεί συνολικά δέκα τέσσερα (14) χαρακτηριστικά που ομαδοποιούνται σε τρεις (3) βασικές κατηγορίες, με τελικό στόχο την εύρεση του καλύτερου και αντιπροσωπευτικότερου χαρακτηριστικού για τον χαρακτηρισμό μιας είδησης σε Ψευδής ή Αληθής . Ένας δεύτερος λόγος σύγκρισης των γνωρισμάτων είναι η ανάδειξη της σημαντικότητας ή όχι των συντακτικών χαρακτηριστικών και συγκεκριμένα των **ονοματικών οντοτήτων** για την αξιοπιστία ενός άρθρου. Για παράδειγμα, σκοπός είναι να διαπιστωθεί εάν η χρήση οντοτήτων (από κάθε είδος οντοτήτων) μπορεί να μας δώσει ένα ακριβές συμπέρασμα για την ποιότητα της είδησης, σε δεύτερη φάση εάν η διαφορετικότητα στην χρήση μερών του λόγου μπορεί να δώσει κάποιο ασφαλές συμπέρασμα και τέλος, ποιο είναι το καλύτερο χαρακτηριστικό για την εξαγωγή αυτού του συμπεράσματος.

Σε αυτό το σημείο αξίζει να αναφερθεί ότι για την εξαγωγή του χαρακτηριστικού που αφορά στα μέρη του λόγου χρησιμοποιήθηκε η χρήση των βιβλιοθηκών της Python spaCy, NLTK και coreNLP όπως αναφέρθηκε και στην *ενότητα 1.3*. Ύστερα από σύγκριση των αποτελεσμάτων η spaCy χρησιμοποιήθηκε για το χαρακτηριστικό των ονοματικών οντοτήτων. Ακόμη, όσα γνωρίσματα αναφέρονται σε stopwords (π.χ. ποσοστό των λέξεων που είναι stopwords) υπολογίστηκαν με βάση το λεξικό που διαθέτει ενσωματωμένο το spaCy και το NLTK. Το γλωσσολογικό μοντέλο για την spaCy που χρησιμοποιήθηκε είναι το "core_web_sm". Πρόκειται για ένα CNN (Convolutional Neural Network) εκπαιδευμένο με σύνολα δεδομένων τα οποία περιλαμβάνουν διαδικτυακό περιεχόμενο όπως blogs, σχόλια και ειδησεογραφικά άρθρα.

Το σύνολο δεδομένων που χρησιμοποιήθηκε αποτελείται συνολικά από 44898 άρθρα εκ των οποίων τα 21578 είναι αληθή ενώ τα υπόλοιπα ψευδή *(Διάγραμμα 3.2)*.



Διάγραμμα 3.2: Πλήθος αληθών και ψευδών ειδήσεων του συνόλου δεδομένων

3.2.1 Υφολογική Ανάλυση

Σε αυτή την κατηγορία εντάσσονται όλα τα χαρακτηριστικά που σχετίζονται με το ύφος του κειμένου καθώς και τα γραμματικά στοιχεία. Επτά (7) από τα γνωρίσματα των δεδομένων εμπίπτουν σε αυτή την κατηγορία:

1. *Αριθμός λέξεων (Number of words)*
2. *Αριθμός δυαδικών ή τριαδικών φράσεων (Number of 2-grams, 3-grams)*
3. *Σχετική Συχνότητα εμφάνισης λέξεων (TF IDF for words)*
4. *Σχετική Συχνότητας εμφάνισης δυαδικών ή τριαδικών φράσεων (TF IDF for 2-grams, 3-grams)*
5. *Αριθμός stopwords με χρήση της spaCy και αφαίρεσή τους (Number of stopwords spaCy+removed)*
6. *Αριθμός stopwords με χρήση της NLTK και αφαίρεσή τους (Number of stopwords NLTK+removed)*
7. *Συναισθηματική πολικότητα (VADER polarity)*

Για την εξαγωγή των τεσσάρων πρώτων χαρακτηριστικών χρησιμοποιήθηκε η ιδέα του *bag-of-words*. Το κείμενο αναπαρίσταται από έναν πίνακα αριθμών συγκεκριμένων διαστάσεων. Ένας συγκεκριμένος αριθμός δίνεται σε κάθε λέξη ο οποίος αφορά στην συχνότητα εμφάνισης της κάθε λέξης. Έτσι, κωδικοποιείται ένα κείμενο σε γλώσσα κατανοητή από μοντέλα Μηχανικής Μάθησης και δημιουργείται ένα λεξικό γνωστών λέξεων. Για τα δύο πρώτα γνωρίσματα χρησιμοποιήθηκε ο **Countvectorizer.**

Το *TF-IDF* δηλώνει τον βαθμό σχετικότητας μιας λέξης για ένα κείμενο συγκρίνοντας έναν αριθμό κειμένων. Ο υπολογισμός γίνεται με τον πολλαπλασιασμό της συχνότητας της λέξης (ή φράσης) στο κείμενο με την αντίστροφη συχνότητα της λέξης (ή φράσης) σε ένα σύνολο εγγράφων. Για τα γνωρίσματα αυτά χρησιμοποιήθηκε ο **TfidfVectorizer.**

Τέλος η συναισθηματική ανάλυση ενός εγγράφου αφορά στην ανάλυση της συναισθηματικής χροιάς και του ύφους που χρησιμοποιήθηκε κατά την συγγραφή του άρθρου. Η πολικότητα ορίζεται στο [-1, 1] ενώ ύστερα από κανονικοποίηση περιορίζεται στο [0, 1], με αρνητικές να χαρακτηρίζονται απόψεις με σχόλια αποδοκιμασίας και αρνητισμού. Για το χαρακτηριστικό αυτό έγινε χρήση του **VADER** (Valence Aware Dictionary and sEntiment Reasoner) που είναι ένα εργαλείο βασισμένο σε λεξικό και κανόνες, κατάλληλο για την ανάλυση συναισθήματος σε γραπτό λόγο [25].

3.2.2 Μορφολογική Ανάλυση

Όπως αναφέρθηκε και στην *υποενότητα 2.1.3* σε αυτή την κατηγορία ανήκει το χαρακτηριστικό των μερών του λόγου. Η επεξεργασία της φυσικής γλώσσας είναι ένα ευρύ πεδίο που μελετά τους τρόπους όσο το δυνατόν πιστότερης αναπαράστασης της γλώσσας χωρίς την απώλεια της ιδιαιτεροτητας της. Η πολυπλοκότητα του συγκεκριμένου προβλήματος έχει οδηγήσει σε ανάπτυξη διαφόρων εργαλείων για την επεξεργασία με πιο γνωστά να είναι τα *coreNLP* και *NLTK*. Σε αυτή την υποενότητα παρουσιάζονται οι διαφορές μεταξύ των βιβλιοθηκών αυτών σε σύγκριση με την *spaCy*, ένα πιο σύγχρονο μέσο, και το συγκριτικό της απόδοσής τους. Άρα στην κατηγορία αυτή υπάρχει μόνο ένα γνώρισμα που στην πραγματικότητα έχει μελετηθεί σαν τρία (3) ξεχωριστά.

3.2.2.1 Natural Language Toolkit

Το Εργαλείο Φυσικής Γλώσσας, ή πιο συχνά NLTK, είναι μια σειρά από βιβλιοθήκες και προγράμματα για συμβολική και στατιστική επεξεργασία φυσικής γλώσσας για Αγγλικά γραμμένα στη γλώσσα προγραμματισμού Python. Είναι ένα πρόγραμμα ανοιχτού λογισμικού το οποίο αναπτύχθηκε από τους Steven Berd και Edward Loper στο Τμήμα Επιστήμης των Υπολογιστών και Πληροφοριών στο Πανεπιστήμιο της Πενσυλβανίας. Το NLTK περιλαμβάνει γραφικές επιδείξεις και αναπαραστάσεις των δεδομένων. Συνοδεύεται από ένα εγχειρίδιο [26] που εξηγεί τις βασικές έννοιες πίσω από το τα στάδια επεξεργασίας της φυσικής γλώσσας.

Είναι ένα εργαλείο με στόχο την χρήση του σαν μέσο διδασκαλίας, μεμονωμένο εργαλείο μελέτης και πλατφόρμα για την δημιουργία πρωτοτύπων και και την δημιουργία ερευνητικών συστημάτων. Το NLTK υποστηρίζει λειτουργίες ταξινόμησης, διακριτικοποίησης, δημιουργίας ετικετών, ετικετών, ανάλυσης και σημασιολογικής μελέτης.

3.2.2.2 Stanford coreNLP

Το coreNLP[27] είναι ένα εργαλείο επεξεργασίας Φυσικής Γλώσσας που αναπτύχθηκε από το Stanford NLP group και κυκλοφόρησε ως ανοιχτό λογισμικό. Για εμπορική χρήση η βιβλιοθήκη διανέμεται μόνο με χρήση ειδικής άδειας. Είναι γραμμένο σε γλώσσα προγραμματισμού Java και δίνει τις βασικές δυνατότητες του *Tokenization, Part-of-speech tagging* και *Named entity recognition.* Για την αρχή της χρήσης του πρέπει πρώτα να εκκινήσει ο server από όπου παίρνονται όλες οι πληροφορίες του εκπαιδευμένου εργαλείο για την εξαγωγή συμπερασμάτων. Εκτός από την γραμμή εντολών παρέχει και ένα API[9] για την καλύτερη εξυπηρέτηση του χρήστη. Έτσι, το εργαλείο προτίθεται για τον πειραματισμό. Τέλος, παρόλο που υποστηρίζεται από Java υπάρχουν και παραλλαγές από άλλες γλώσσες προγραμματισμού όπως η Python.

3.2.2.3 spaCy

Το spaCy είναι μια βιβλιοθήκη λογισμικού ανοιχτού κώδικα για προηγμένη επεξεργασία φυσικής γλώσσας, γραμμένη στις γλώσσες προγραμματισμού Python και Cython. Είναι ένα από τα νεότερα εργαλεία καθώς κατασκευάστηκε από τους Matthew Honnibal και Ines Montani με πρώτη έκδοση το 2015 και χρήση της άδειας του MIT[10]. Η βιβλιοθήκη αυτή δίνει τις ίδιες δυνατότητες *(Εικόνα 3.1)*[28] με τις προαναφερθέντες αλλά με καλύτερη απόδοση καθώς φτιάχτηκε με γνώμονα το βέλτιστο κάθε φορά αλγόριθμο.

---

[9] Το API είναι ένας διαμεσολαβητής που περαλμβάνει το αίτημα του χρήστη και το μεταβιβάζει είτε σε ένα server μέσω πρωτοκόλλου είτε σε μια τρίτη εφαρμογή.

[10] *https://en.wikipedia.org/wiki/MIT_License*

	SPACY	NLTK	CORENLP
Programming language	Python	Python	Java / Python
Neural network models	✓	✗	✓
Integrated word vectors	✓	✗	✗
Multi-language support	✓	✓	✓
Tokenization	✓	✓	✓
Part-of-speech tagging	✓	✓	✓
Sentence segmentation	✓	✓	✓
Dependency parsing	✓	✗	✓
Entity recognition	✓	✓	✓
Entity linking	✓	✗	✗
Coreference resolution	✗	✗	✓

Εικόνα 3.1: Σύγκριση των δυνατοτήτων των τριών εργαλείων επεξεργασίας φυσικής γλώσσας. Πηγή: researchgate.net

3.2.2.4 Σύγκριση των τεχνολογιών

Είναι σαφές λοιπόν ότι τα τρία εργαλεία αυτά έχουν πολλές ομοιότητες μεταξύ τους και μπορούν να χρησιμοποιηθούν τόσο για την μορφολογική όσο και για την συντακτική μελέτη ενός εγγράφου. Ένας τρόπος ανάδειξης κάποιου είναι με την προτίμηση των χρηστών και των ερευνητών, καθώς όπως είναι λογικό, προτιμάται μια βιβλιοθήκη με πληρέστερη και πιο τεκμηριωμένη βιβλιογραφία και ενεργή κοινότητα προγραμματιστών. Ωστόσο αυτός ο τρόπος δεν είναι πάντα αξιόπιστος καθώς η προτίμηση εξαρτάται και από τα χρόνια κυκλοφορίας μιας τεχνολογίας και όπως αναφέρθηκε παραπάνω το spaCy είναι μια βιβλιοθήκη αρκετά νεότερη από τις υπόλοιπες. Για τον λόγο αυτό, η σύγκριση των τριών έγινε σύμφωνα με τον χρόνο την απόδοσή τους. Αναλυτικότερα, ο χρόνος επεξεργασίας των δεδομένων παίζει μεγάλο ρόλο στην απόφαση του βέλτιστου πακέτου. Όπως φαίνεται στον *Πίνακα 3.1* η spaCy παρουσιάζει **μεγαλύτερη ορθότητα** και καλύτερους χρόνους, γι αυτό και κατατάσσεται στους πιο γρήγορους αναλυτές του κόσμου.

ABSOLUT (ms/doc)				***RELATIVE (to spaCy)***		
SYSTEM	*Tokenizer*	*Tagging*	*Parsing*	*Tokenizer*	*Tagging*	*Parsing*
spaCy	*0.2ms*	*1ms*	*19ms*	*1x*	*1x*	*1x*
coreNLP	*2ms*	*10ms*	*49ms*	*10x*	*10x*	*2.6x*
NLTK	*4ms*	*443ms*	*n/a*	*20x*	*443x*	*n/a*

Πίνακας 3.1: Σύγκριση των χρονικών ρεκόρ των spaCy, coreNLP, NLTK. Πηγή: spacy.io

Ο *Πίνακας 3.1* μας δίνει μια πρώτη εκτίμηση της επίδοσης των τριών αναλυτών αλλά δεν είναι προσαρμοσμένος στο σύνολο δεδομένων που εστιάζει η παρούσα εργασία. Ως εκ τούτου, κρίθηκε απαραίτητη η εξαγωγή των μερών του λόγου για τα δεδομένα των άρθρων με την χρήση και των τριών τεχνολογιών σαν ξεχωριστό χαρακτηριστικό και στην συνέχεια εκπαίδευση των τριών ταξινομητών που περιγράφονται στις *υποενότητες 2.1.6, 2.1.7* και *2.1.8*. Τα αποτελέσματα παρουσιάζονται αναλυτικά στον *Πίνακα 3.2*.

Classifier	***PoS NLTK***	***PoS coreNLP***	***PoS spaCy***
Random Forest	*95,033407*	*52,910170*	*95,404602*
Naive Bayes	*79,821826*	*52,910170*	*85,389755*
Support Vector Machine	*86,926503*	*52,620638*	*92,331106*

Πίνακας 3.2: Συγκριτικό της ορθότητας ταξινόμησης κάθε ταξινομητή (Random Forest, Naive Bayes, Support Vector Machine) για την κατηγοριοποίηση των μερών του λόγου για κάθε ένα εργαλείο (spaCy, coreNLP, NLTK)

Η ορθότητα ταξινόμησης δεν υποδηλώνει κάτι για την βέλτιστη απόδοση ανάμεσα στα spaCy, coreNLP και NLTK ωστόσο μας δίνει μια γενικότερη εικόνα για τους τρεις αναλυτές στα δεδομένα μας. Λαμβάνοντας υπόψη ότι ένας από τους **στόχους της εργασίας είναι η όσο το δυνατόν σωστότερη κατηγοριοποίηση των ειδήσεων σε αληθείς και ψευδείς,** κρίνεται καίριο να χρησιμοποιηθεί αυτή η βιβλιοθήκη που αντιπροσωπεύει πιστότερα τα συγκεκριμένα δεδομένα και μας δίνει καλύτερα αποτελέσματα. Αυτός ήταν και ο λόγος που έγινε η σύγκριση με την χρήση των ταξινομητών, ωστέ να γίνει εμφανές ποια από τις τρεις υπερτερεί βγάζοντας ένα μέσο σκορ της ορθότητας με βάση και τους τρεις αλγορίθμους εκπαίδευσης. Η spaCy, φαίνεται να αντιπροσωπεύει καλύτερα τα δεδομένα και να προσφέρει ασφαλέστερα και γρηγορότερα αποτελέσματα.

3.2.3 Συντακτική Ανάλυση

Ένας ακόμη στόχος της παρούσας μελέτης **είναι η διαπίστωση της σημαντικότητας των ονοματικών οντοτήτων στην κατηγοριοποίηση εγγράφων και συγκεκριμένα ειδήσεων**. Η συντακτική ανάλυση όπως αναφέρθηκε και στην *υποενότητα 2.1.4* αφορά στο γνώρισμα αυτό. Για τον υπολογισμό του χαρακτηριστικού αυτού οι τεχνολογίες που πρωτοστατούν είναι και πάλι οι spaCy, coreNLP και NLTK. Καθώς το πόρισμα από την σύγκριση αυτών των τριών ανέδειξε την **spaCy** ως την καταλληλότερη τόσο σε θεωρητικό όσο και σε πρακτικό επίπεδο, επιλέχθηκε αυτή για την εξαγωγή του γνωρίσματος. Επιπλέον, για την διεξοδικότερη εξέταση της επίδρασης των ονοματικών οντοτήτων στην κατηγοριοποίηση εξήχθησαν συγκεκριμένα σαν διαφορετικό γνώρισμα οι ονοματικές οντότητες PERSON (άνθρωπος), ORG (οργανισμός), GPE (τοποθεσίες π.χ. χώρες, πόλεις, πολιτείες), DATE (ημερομηνίες) και PERCENT (ποσοστό %). Άρα συνολικά σε αυτή την κατηγορία γνωρισμάτων εμπίπτουν τέσσερα (4) χαρακτηριστικά εκ των οποίων όλα αφορούν στην αναγνώριση ονοματικών οντοτήτων.

4 ΑΠΟΤΕΛΕΣΜΑΤΑ ΑΛΓΟΡΙΘΜΩΝ

Παρακάτω παρατίθενται τα αποτελέσματα της ορθότητας των αλγορίθμων εκπαίδευσης για κάθε μια από τις κατηγορίες (Fake, Real) ειδήσεων με βάση την ορθότητα ταξινόμησης καθώς και αναλύεται η απόδοση του χαρακτηριστικού των ονοματικών οντοτήτων σύμφωνα με τις μετρικές εκτιμώμενης απόδοσης που περιγράφονται στην *Ενότητα 2.2*.

Όπως αναφέρθηκε και στο *Κεφάλαιο 3* για την εξαγωγή των συμπερασμάτων ορθότητας το αρχικό σύνολο δεδομένων χωρίστηκε σε training και testing υποσύνολα εκ των οποίων το testing αποτελεί το 30% του αρχικού και το training το 70%. Το training χρησιμοποιείται για την εκμάθηση του αλγορίθμου. Ο διαχωρισμός είναι ένα κρίσιμο βήμα που χρειάζεται να διατηρεί τις ισορροπίες. Το 30% είναι ένα σχετικά μικρό ποσοστό του αρχικού συνόλου, ωστόσο μας διασφαλίζει ένα αρκετά μεγάλο υποσύνολο εκπαίδευσης και έτσι καλύτερη εκπαίδευση του ταξινομητή. Ένα μεγάλο υποσύνολο εκπάιδευσης επίσης, ελοχεύει τον κίνδυνο της *Υπερμοντελοποίησης* (βλέπε *υποενότητα 2.2.4)*. Υπερμοντελοποίση μπορεί να παρατηρηθεί ακόμη, στην περίπτωση εκ των προτέρων γνώσης του συνόλου δεδομένων διότι έτσι παρατηρείται μια εμμονή στην διατήρησή τους και προσαρμογή των παραμέτρων στα χαρακτηριστικά αυτα. Από την άλλη, ένα μεγαλύτερο testing dataset θα εξυπηρετούσε την αποφυγή εσφαλμένων αποτελεσμάτων αφού τα δείγματα θα διατηρούσαν την ποικιλομορφία τους αλλά παράλληλα θα μείωνε το training dataset με πιθανό αποτέλεσμα την λιγότερο σφαιρική εκπαίδευση.

4.1 Μελέτη ρύθμισης παραμέτρων

Για την αποφυγή των παραπάνω σκοπέλων χρησιμοποιήθηκαν οι τεχνικές του ***k-fold Cross Validation*** καθώς και του ***Grid search.*** Οι τεχνικές αυτές μας επιτρέπουν να εντοπίσουμε αρχικά τις βέλτιστες παραμέτρους για τους αλγόριθμος εκπαίδευσης ύστερα από επαναληπτικές δοκιμές εκπαίδευσης και να τις εφαρμόσουμε στα αρχικά δεδομένα εκπαίδευσης για επαλήθευση των αποτελεσμάτων ορθότητας.

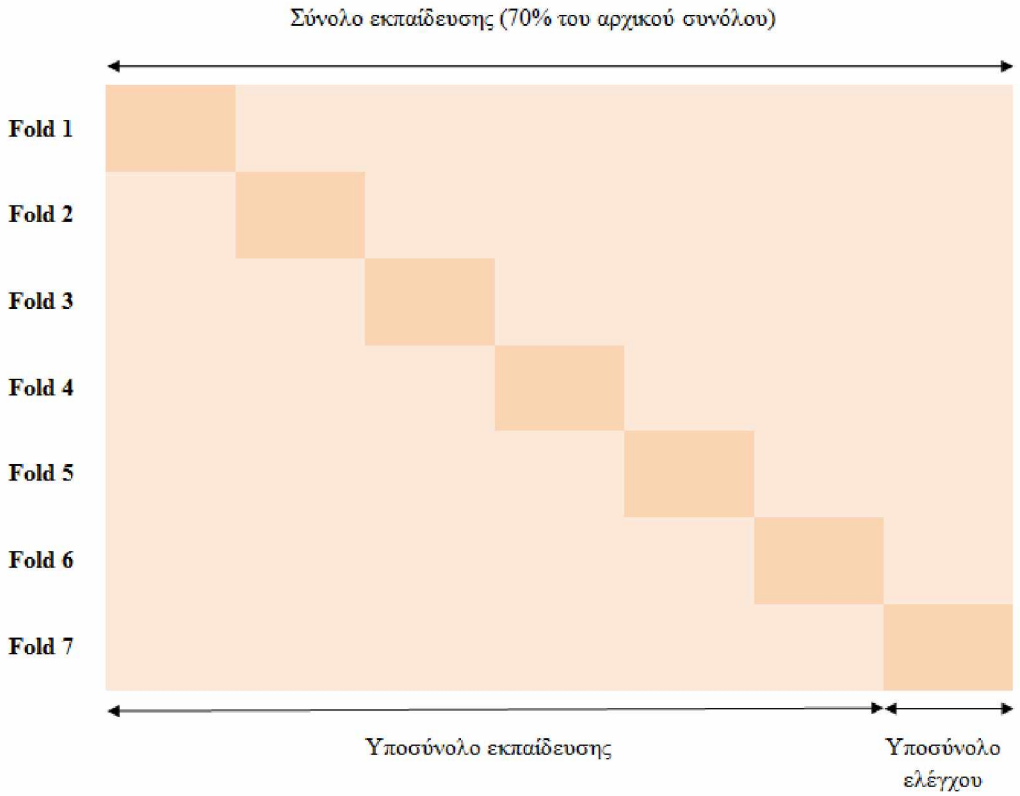
Παρακάτω θα περιοριστούμε στους κατηγοριοποιητές Support Vector Machine και Random Forest καθώς αυτοί δέχονται ορίσματα. Βέβαια, η ίδια διαδικασία ακολουθήθηκε και στην περίπτωση του Naive Bayes που όμως μας επιβεβαίωσε τα αρχικά αποτελέσματα.

4.1.1 Random Forest

Για τον Random Forest η παράμετρος που ρυθμίστηκε είναι μία και αφορά στο *n_estimators* που είναι ο αριθμός των Δέντρων Απόφασης. Η προκαθορισμένη τιμή για την παράμετρο είναι 100. Μια πολύ μεγάλη τιμή της παραμέτρου μπορεί να συνεπάγεται την αργή εκτέλεση του αλγορίθμου και την αύξηση της πολυπλοκότητας ενω αντίθετα μια μικρή αξία μπορεί να οδηγήσει σε σύγχυση της πλειοψηφικής εκλογής του αποτελέσματος. Αναλυτικότερα, σε μια πολύ μικρή τιμή χάνεται πλήρως η ιδέα των περισσότερων σε πλήθος ψήφων ενώ όσο αυξάνεται το πλήθος των Δέντρων τόσο αποφεύγεται η πιθανότητα ισοψηφίας. Για να εντοπιστεί η

καταλληλότερη τιμή εφαρμόστηκε *7-fold Cross Validation* και *Grid search* στο αρχικό σύνολο εκπαίδευσης που αποτελεί το 70% των αρχικών άρθρων.

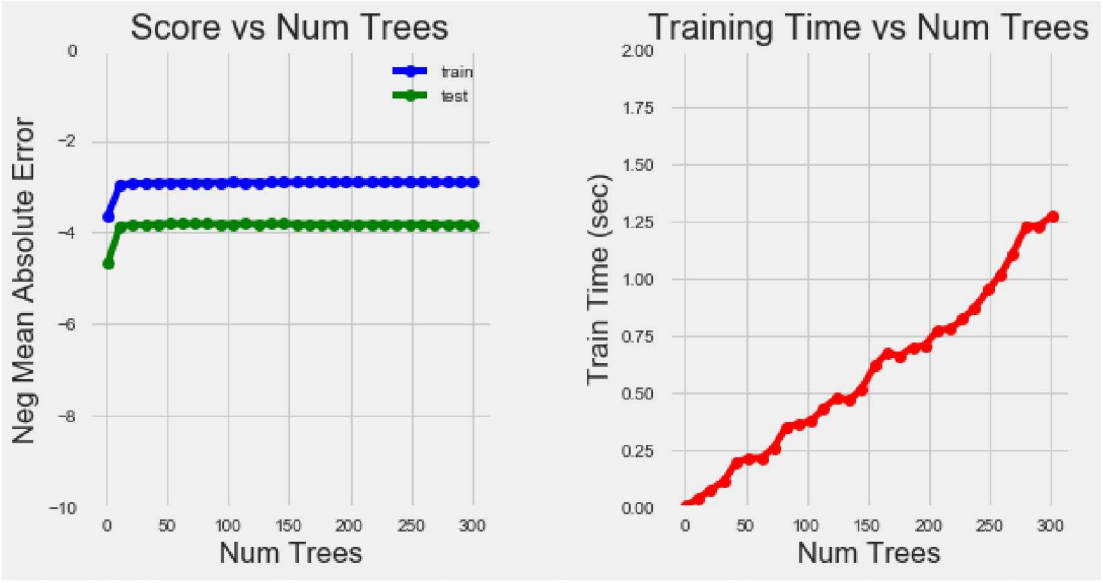
Η διαδικασία περιλαμβάνει τον διαχωρισμό του συνόλου σε δύο τμήματα training και testing τα οποία μεταβάλλονται ανάλογα με τον αριθμό των folds που έχουμε επιλέξει *(Εικόνα 4.1)*.



Εικόνα 4.1: Παράδειγμα 7-fold Cross Validation.

Για κάθε τιμή n_estimators από το σύνολο τιμών που μας ενδιαφέρει επαναλαμβάνεται η διαδικασία για όλα τα folds. Οι τιμές που επιλέχθηκαν για τη διερεύνηση είναι οι παρακάτω:

$$n\_estimators \in \{0, 50, 100, 150, 200, 250, 300\} \qquad (4.1)$$



Εικόνα 4.2: Σύγκριση ορθότητας και χρόνου εκπαίδευσης για τις διάφορες τιμές της παραμέτρου του Random Forest και για 10-fold Cross Validation.

Από την *Εικόνα 4.2* παρατηρούμε ότι ο χρόνος εκπαίδευσης αυξάνεται παράλληλα με την αύξηση στο πλήθος των Δέντρων. Από τα μηδέν (0) μέχρι τα τριακόσια (300) Δέντρα έχουμε αύξηση 150% στον χρόνο που είναι στατιστικά σημαντική. Αντίθετα, το τετραγωνικό σφάλμα παραμένει σταθερό τόσο στο υποσύνολο εκπαίδευσης όσο και στον υποσύνολο ελέγχου ύστερα τα τριάντα (30) περίπου Δέντρα. Ωστόσο, επειδή η τιμή σταθεροποίησης του σφάλματος δεν είναι ξεκάθαρη, για την ρύθμιση αυτού του ορίσματος επιλέχθηκε η τιμή πενήντα (50) από όπου και υπάρχει σαφής σύγκλιση της τιμής μέσης ορθότητας σφάλματος. Για να εντοπιστεί η καταλληλότερη τιμή εφαρμόστηκε *10-fold Cross Validation* και *Grid search* στο αρχικό σύνολο εκπαίδευσης που αποτελεί το 70% των αρχικών άρθρων.

4.2.2 Support Vector Machine

Για τον SVM λοιπόν, τα ορίσματα που χρησιμοποιήσαμε είναι δύο και αφορούν:

- Τον πυρήνα ταξινόμησης (kernel).
- Την τιμή C.

Για την τιμή του πυρήνα ταξινόμησης επιλέχθηκε η τιμή του *"linear"* (γραμμικός πυρήνας) καθώς ύστερα από δοκιμές έδειχνε μεγαλύτερη ανοχή στα αποτελέσματα. Η παράμετρος C δηλώνει την ανοχή σε σφάλματα κατά την ταξινόμηση. Δηλαδή μια μεγάλη τιμή C μειώνει την περίπτωση εντοπισμού σφαλμάτων αυξάνοντας την ορθότητα. Ενώ αντίθετα μια μικρή τιμή C σημαίνει και αυστηρότητα στον υπολογισμό της ορθότητας άρα λιγότερη ανοχή σε σφάλματα.

Η ποεπιλεγμένη τιμή της παραμέτρου C είναι 100. Για κάθε τιμή C από το σύνολο τιμών που μας ενδιαφέρει επαναλαμβάνεται η διαδικασία για όλα τα folds. Οι τιμές που επιλέχθηκαν για τη διερεύνηση είναι οι παρακάτω:

$$C \in \{1, 10, 100, 1000, 10000, 20000, 30000\} \qquad (4.2)$$

Στον *Πίνακα 4.1* παρατίθενται τα αποτελέσματα της ορθότητας ταξινόμησης για το χαρακτηριστικό των ονοματικών οντοτήτων (παραδειγματικά) για τις διάφορες τιμές του C.

C	*1*	*10*	*100*	*1000*	*10000*	*20000*	*30000*
training	*76,1772*	*79.1072*	*80.5772*	*81.2472*	*81.5872*	*81.9072*	*81.9972*
testing	*75,7535*	*78.4635*	*79.5135*	*80.5335*	*81.0435*	*80.8835*	*80.8835*

Πίνακας 4.1: Συγκριτικό της απόδοσης του ταξινομητη SVM για τις διάφορες τιμές του C.

Όπως φαίνεται από τον πίνακα το καλύτερο σκορ ταξινόμησης είναι για την τιμή του *C= 10.000* ενώ η χειρότερη είναι για την *C=1.*

4.2 Απόδοση κατηγοριοποίησης

Αφού έχει γίνει συλλογή και προεπεξεργασία των δεδομένων, εξαγωγή των χαρακτηριστικών τους, διερεύνηση του καλύτερου εργαλείου για την εξαγωγή αυτή και ρύθμιση των παραμέτρων για τους αλγορίθμους εκπαίδευσης, το τελευταίο στάδιο είναι η εκπαίδευση του αλγορίθμου και η παρουσίαση των αποτελεσμάτων για την επιλογή του καλύτερου χαρακτηριστικού και την εκτίμηση της επίδρασης των ονοματικών οντοτήτων.

4.2.1 Υφολογικά χαρακτηριστικά

	Count of ngram	*Count of words*	*TF IDF ngram*	*TF IDF words*	*NLTK removed count*	*SpaCy removed count*	*VADER polarity*
Random Forest	*98.040*	*98.841*	*98.757*	*99.688*	*98.285*	*98.530*	*70.200*
Naive Bayes	*98.262*	*95.545*	*95.877*	*93.385*	*98.374*	*98.262*	*52.160*
Support Vector Machine	*98.819*	*99.576*	*99.532*	*98.953*	*97.706*	*97.661*	*60.957*

Πίνακας 4.2: Σύγκριση της ορθότητας ταξινόμησης των υφολογικών χαρακτηριστικών.

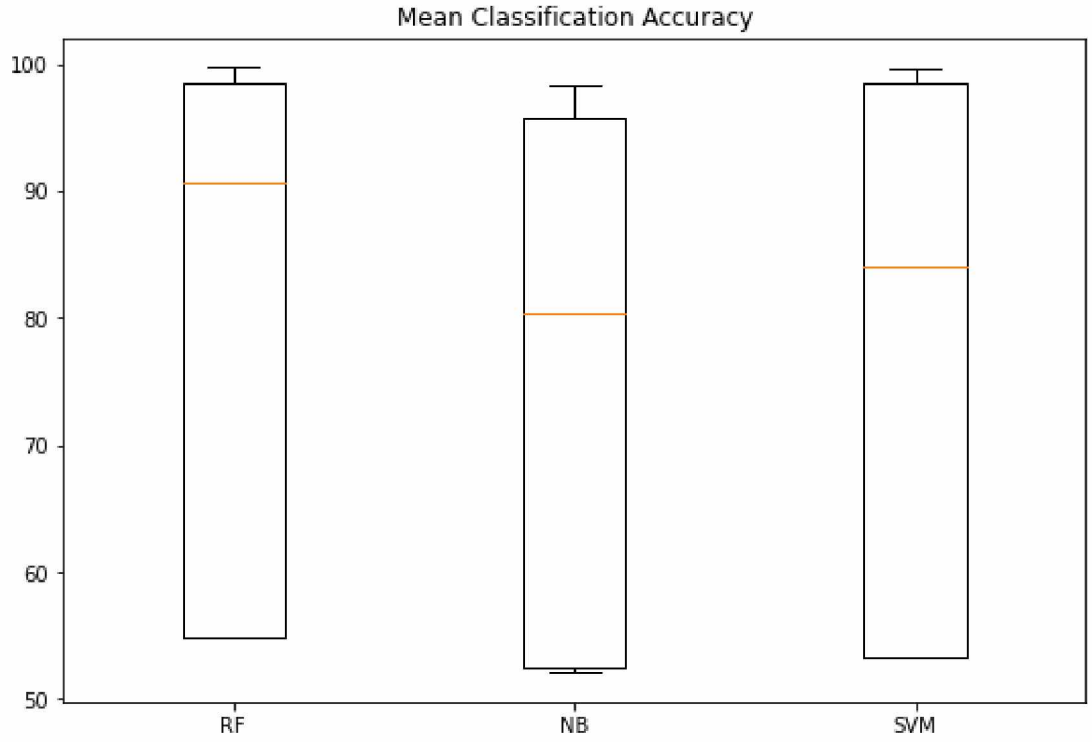
4.2.2 Μορφολογικά και συντακτικά χαρακτηριστικά

	PoS	*NER*	*PER. ER*	*GPE ER*	*ORG ER*	*DATE ER*	*PERC. ER*
Random Forest	*95.404*	*85.998*	*54.936*	*54.936*	*54.936*	*54.951*	*54.951*
Naive Bayes	*85.389*	*74.677*	*52.487*	*54.936*	*52.487*	*52.487*	*52.487*
Support Vector Machine	*92.331*	*75.753*	*53.251*	*53.251*	*53.251*	*53.251*	*53.251*

Πίνακας 4.3: Σύγκριση της ορθότητας ταξινόμησης των μορφολογικών και συντακτικών χαρακτηριστικών.

Με βάση τους *Πίνακες 4.2* και *4.3* το ακριβέστερο χαρακτηριστικό είναι **ο αριθμός των δυαδικών και τριαδικών φράσεων** διότι είναι το μόνο γνώρισμα με υψηλό σκορ ορθότητας και για τους τρεις ταξινομητές που χρησιμοποιήθηκαν που παράλληλα εμφανίζει συνοχή στα αποτελέσματα. Εδώ χρησιμοποιείται σαν μέτρο ανάδειξης καλύτερου χαρακτηριστικού η ορθότητα ταξινόμησης διότι είναι αρκετή αφού κατά την σύγκριση τους χρησιμοποιήθηκαν οι ίδιοι ταξινομητές στο ίδιο υποσύνολο εκπαίδευσης και ελέγχου. Η ίδια καθολική μετρική λοιπόν για όλα τα γνωρίσματα δίνει μια εικόνα της κατηγοριοποίησης.

Στο *Διάγραμμα 4.1* παρουσιάζεται η γραφική αναπαράσταση της μέσης ορθότητας για τους τρεις ταξινομητές για το σύνολο των χαρακτηριστικών. Όπως είναι εμφανές οι μέσες ακρίβειες στους ταξινομητές Random Forest και Support Vector Machine είναι υψηλότερες από αυτή του Naive Bayes κάτι που είναι αναμενόμενο από την στιγμή που και οι δύο ταξινομητές υπέστησαν *Grid search.*



Διάγραμμα 4.1: Μέσες Ορθότητες ταξινόμησης για τους τρεις αλγορίθμους.

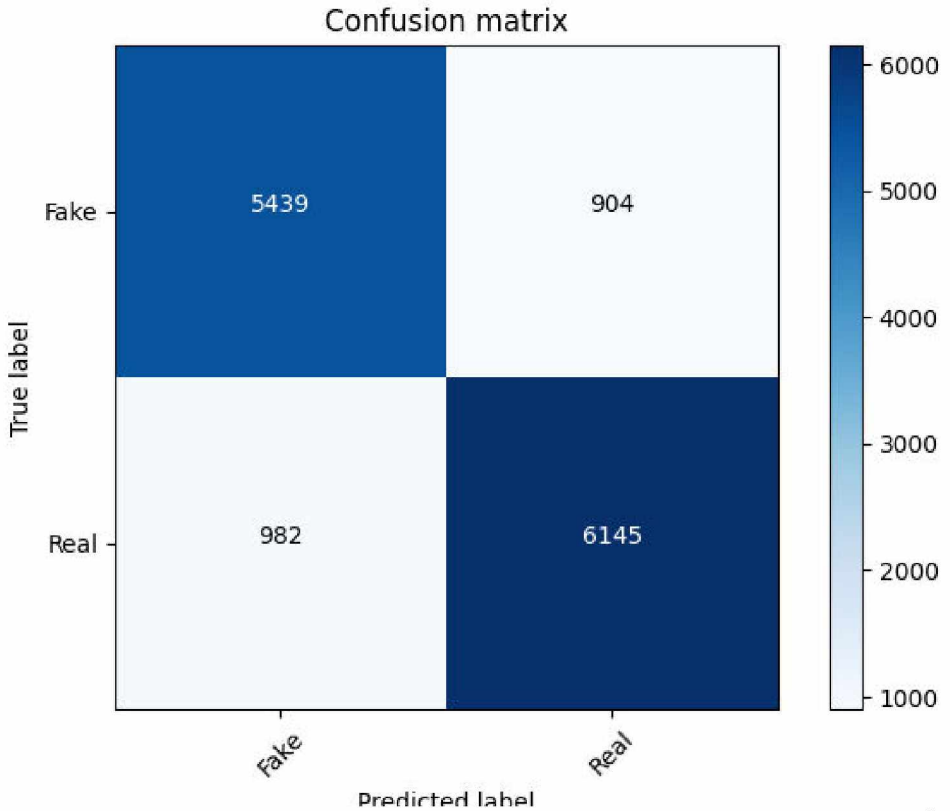
4.3 Απόδοση της αναγνώρισης ονοματικών οντοτήτων

Το τελευταίο βήμα διερεύνησης είναι η διαπίστωση της σημαντικότητας των ονοματικών οντοτήτων στην κατηγοριοποίηση των ειδήσεων σε δύο κλάσεις (Fake, Real). Για την διαδικασία αυτή θα γίνει επίδειξη των πινάκων σύγχυσης αλλά και των μετρικών αξιολόγησης όπως αυτές αναφέρθηκαν στην *Ενότητα 2.2*.

4.3.1 Αναγνώριση ονοματικών οντοτήτων για τον ταξινομητή Random Forest

Από την *Εικόνα 4.3* γίνεται ο εξής υπολογισμός των μετρικών σύμφωνα με όσα αναφέρθηκαν στο αντίστοιχο Κεφάλαιο, με αρνητική πρόβλεψη να θεωρείται η ψεύτικη είδηση (Fake) και θετική πρόβλεψη η είδηση να είναι αληθής (Real):

- *Accuracy:* 85.99851
- *Recall:* 46.95269
- *Precision:* 87.17548
- *F1:* 61.03301

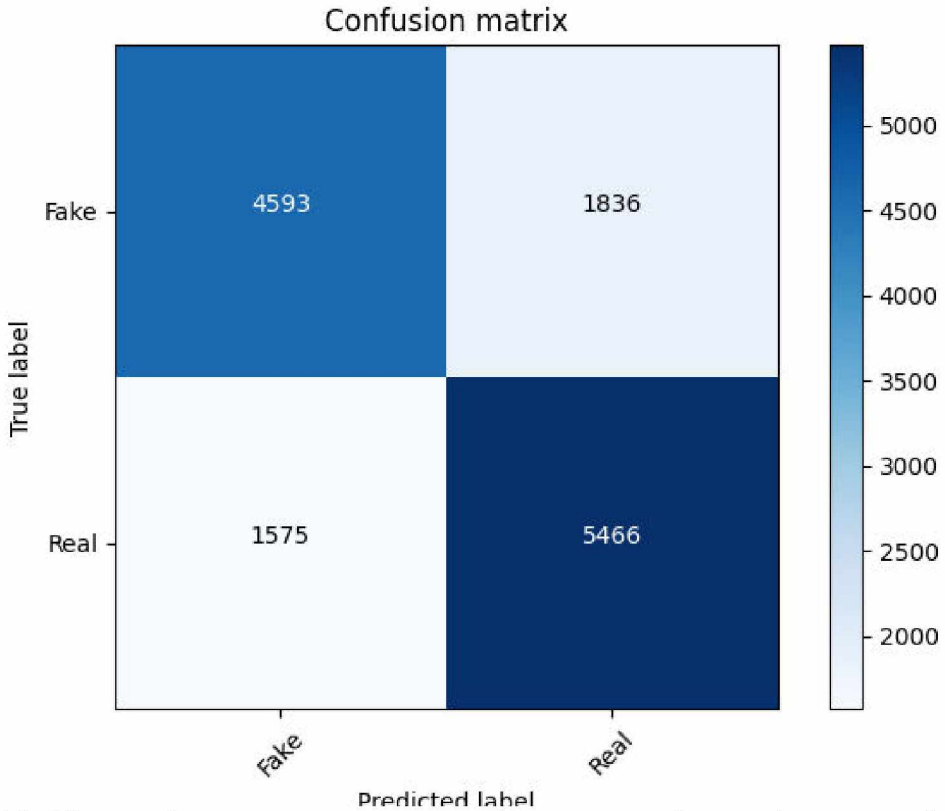


Εικόνα 4.3: Πίνακας σύγχυσης για το χαρακτηριστικό των ονοματικών οντοτήτων στον ταξινομητή Random Forest

4.3.2 Αναγνώριση ονοματικών οντοτήτων για τον ταξινομητή Naive Bayes

Από την *Εικόνα 4.4* απορρέει ο εξής υπολογισμός των μετρικών, με θετική και αρνητική πρόβλεψη όπως αναφέρθηκε στην περίπτωση του Random Forest:

- *Accuracy:* 74.67706
- *Recall:* 45.66060
- *Precision:* 74.85620
- *F1:* 56.72203

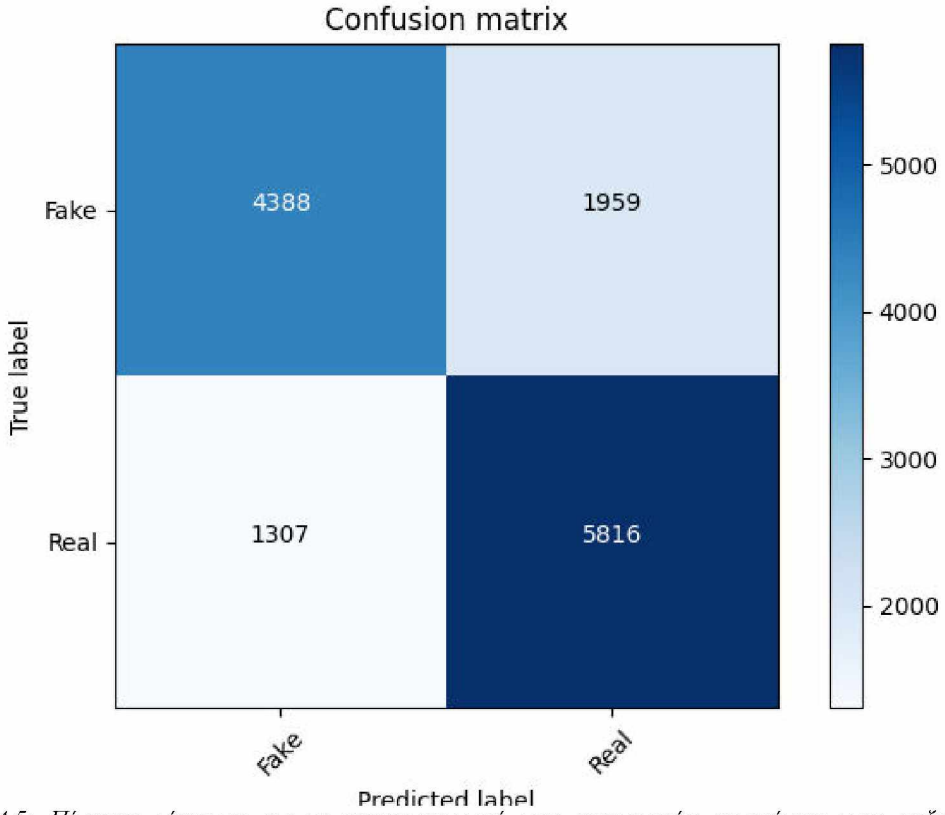


Εικόνα 4.4: Πίνακας σύγχυσης για το χαρακτηριστικό των ονοματικών οντοτήτων στον ταξινομητή Naive Bayes

4.3.3 Αναγνώριση ονοματικών οντοτήτων για τον ταξινομητή Support Vector Machine

Από την *Εικόνα 4.5* απορρέει ο εξής υπολογισμός των μετρικών, με θετική και αρνητική πρόβλεψη όπως αναφέρθηκε στις ανωτέρω περιπτώσεις:

- *Accuracy:* 75.75352
- *Recall:* 43.00274
- *Precision:* 74.80385
- *F1:* 54.61104

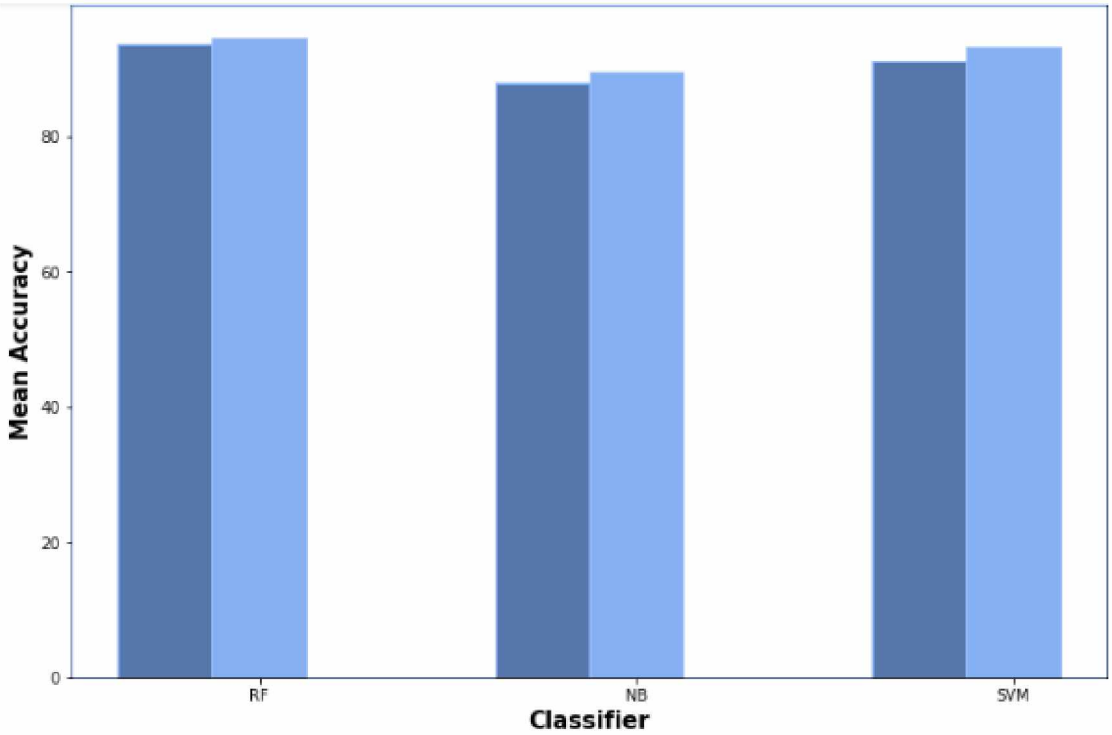


Εικόνα 4.5: Πίνακας σύγχυσης για το χαρακτηριστικό των ονοματικών οντοτήτων στον ταξινομητή Support Vector Machine

Συμπερασματικά, τα χαρακτηριστικά του *Πίνακα 4.3* φαίνεται να έχουν μέτρια προς ικανοποιητική επίδοση, με ικανοποιητική επίδοση να θεωρείται η άνω του 70%. Η κάθε μια κατηγορία οντοτήτων ωστόσο, σαν ξεχωριστό χαρακτηριστικό, δεν αποτελεί δείκτη διαχωρισμού των ειδήσεων στις δύο κατηγορίες, αληθείς και ψευδείς. Το σύνολο των ονοματικών οντοτήτων σαν χαρακτηριστικό, όμως μπορεί να λειτουργήσει σαν διαφορά ανάμεσα στις δύο κατηγορίες, χωρίς αυτό να συνεπάγεται ότι είναι απόλυτο κριτήριο. Όπως φαίνεται μάλιστα, τα περισσότερα από τα χαρακτηριστικά του *Πίνακα 4.2* έχουν στατιστικά καλύτερη επίδοση σε σχέση με αυτά του *Πίνακα 4.3*. Επιπλέον, όπως ήταν αναμενόμενο, και στο χαρακτηριστικό των ονοματικών οντοτήτων, ο Random Forest με τον Support Vector Machine έχουν μεγαλύτερη ορθότητα σε σχέση με τον Naive Bayes αφού έχουν περάσει την διαδικασία του *Grid search* όπως αποδείχθηκε και στην *υποενότητα 4.2.2*.

Επιπρόσθετα με όσα αναφέρθηκαν, για την εξαγωγή ασφαλούς συμπεράσματος σχετικά με την σημαντικότητα του συγκεκριμένου γνωρίσματος έγινε απεικόνιση στο *Διάγραμμα 4.2* της συνολικής ορθότητας μαζί με το χαρακτηριστικό των ονοματικών οντοτήτων και της ορθότητας χωρίς να λάβουμε υπόψη το χαρακτηριστικό αυτό καθώς και τα τέσσερα (4) επιμέρους. Και για τους τρεις ταξινομητές παρατηρείται μια ελαφρά αύξηση στην ορθότητα ύστερα από τον συνυπολογισμό των ονοματικών οντοτήτων στην μέση ορθότητα. Έτσι, το συμπέρασμα που απορρέει είναι ότι **η αναγνώριση ονοματικών οντοτήτων για την ταξινόμηση των ειδήσεων σε δύο κατηγορίες (Fake, Real) δεν συνεισφέρει στατιστικά σημαντικά, ωστόσο κάθε μικρή συνεισφορά βελτιώνει την συνολική απόδοση του ταξινομητή και πλησιάζει τον στόχο του πλήρη διαχωρισμού.** Η μικρή συνεισφορά δεν είναι αποθαρρυντική για την εξαγωγή του χαρακτηριστικού, αφού εκτός της βελτίωσης, αποτελεί μια ενδιαφέρουσα πληροφορία για την ανάλυση

δεδομένων κειμένου και έτσι συχνά γίνεται απόσπαση του χαρακτηριστικού για λόγους μελέτης.



Διάγραμμα 4.2: Η σύγκριση συνολικής ορθότητας συνυπολογίζοντας το NER (κόκκινο) και χωρίς το NER (πράσινο) για τους τρεις ταξινομητές.

5 ΥΛΟΠΟΙΗΣΗ ΕΦΑΡΜΟΓΗΣ - ΠΕΡΙΓΡΑΦΗ ΤΕΧΝΟΛΟΓΙΩΝ

Υπάρχουν τρία (3) διαφορετικά είδη εφαρμογών κινητού τηλεφώνου (mobile apps). Το πρώτο είδος είναι οι *Φυσικές εφαρμογές (native apps)* όπου η εφαρμογή έχει κατασκευαστεί κάθε φορά για το συγκεκριμένο λογισμικό όπως π.χ. κινητά με λογισμικό iOS της Apple ή κινητά με λογισμικό Android της Google. Τέτοιου είδους εφαρμογές προσφέρουν την βέλτιστη δυνατή εμπειρία στον χρήστη καθώς είναι προσαρμοσμένες καλά στο περιβάλλον του κινητού και τρέχουν ομαλότερα και γρηγορότερα. Υπάρχουν επίσης *υβριδικές εφαρμογές (hybrid apps)* οι οποίες μπορούν να εγκατασταθούν σε συσκευές αλλά τρέχουν μέσω του φυλλομετρητή (web browser). Όλες οι υβριδικές εφαρμογές έχουν κατασκευαστεί μέσω της γλώσσας προγραμματισμού *HTLM5*. Η τρίτη και τελευταία ομάδα εφαρμογών είναι οι ***διαδικτυακές εφαρμογές (web apps)*** όπου αλλάζουν τον σχεδιασμό τους ανάλογα εάν χρησιμοποιούνται από κινητό ή όχι. Είναι προσαρμοστικές στο μέγεθος της οθόνης και κατασκευάζονται με την χρήση γλωσσών προγραμματισμού. Λόγω των πολλών πλεονεκτημάτων των διαδικτυακών εφαρμογών, όπως η ανεξαρτησία από το υλικό (hardware), σε αυτή την εργασία επιλέχθηκε αυτό το είδος για την εφαρμογή ανίχνευσης ψευδών ειδήσεων και την επεξεργασία άρθρων από το διαδίκτυο. Η περιγραφή των τεχνολογιών που χρησιμοποιήθηκαν για την ανοικοδόμηση της διαδικτυακής εφαρμογής εφαρμογής περιγράφεται σε αυτό το κεφάλαιο.

5.1 Streamlit

Το *Streamlit* (Treuille, Teixeira, & Kelly, 2020) είναι ένα εργαλείο ανοιχτού λογισμικού της Python για την κατασκευή εφαρμογών με διαδραστικό περιεχόμενο και παρουσίαση αποτελεσμάτων. Χρησιμοποιείται συχνά σε εφαρμογές Μηχανικής Μάθησης και παρέχει δυνατότητες όπως την επίδειξη σχημάτων, την ανάλυση κειμένου και την εμφάνιση στατιστικών στοιχείων. Η ενσωμάτωση του Streamlit μετακινεί τις εφαρμογές της Python έξω από το παράθυρο της κονσόλας και σε έναν διαδικτυακού περιηγητή. Έτσι, μέσα σε ελάχιστο χρόνο γίνεται ευρύτερα η κοινή χρήση της διαδικτυακής εφαρμογής που μας ενδιαφέρει. Το Streamlit διαθέτει επίσης ένα API το οποίο προσφέρει μια απλουστευμένη μορφή της εφαρμογής στο χρήστη, που δεν γνωρίζει τι υπάρχει στο πίσω κομμάτι της εφαρμογής (back-end). Ο κώδικας που εκτελείται μοιάζει με έναν τυπικό κώδικα σε Python που έχει βασικές συναρτήσεις (functions) με ορίσματα. Επίσης, υπάρχουν εντολές του API για να διαμορφωθεί η σελίδα που βλέπει ο χρήστης (front-end). Κάθε εφαρμογή μεταφράζεται σαν script όπου τρέχουν από την κορυφή στο τέλος και κάθε φορά που ένας χρήστης κάνει χρήση της εφαρμογής (δηλαδή ανοίγει ένας καινούργιος φυλλομετρητής) το script επανα εκτελείται. Όσο το script εκτελείται το Streamlit εμφανίζει το αιτούμενο αποτέλεσμα ζωντανά στην οθόνη. Το script αποθηκεύει σε μια προσωρινή μνήμη (cache) για την αποφυγή επανυπολογισμού δύσκολων συναρτήσεων, ώστε να γίνονται σε εύλογο χρόνο οι ενημερώσεις. Κάθε φορά που ένας χρήστης αλληλεπιδρά με ένα στοιχείο της διεπαφής η τιμή εξόδου ανατίθενται πλέον στο στοιχείο ως η νέα του τιμή.

Ολόκληρη η αρχιτεκτονική είναι τέτοια που κάθε συνδεδεμένος χρήστης έχει τη δική του περίοδο σύνδεσης στο διακομιστή και το δικό του ξεχωριστό *νήμα(thread)* όπου εκτελείται το *source file* της εφαρμογής. Ενώ εκτελείται το αρχείο, η βιβλιοθήκη Streamlit σε αυτό το νήμα γράφει μπορεί να γράψει μόνο σε αυτό το συγκεκριμένο αντικείμενο περιόδου σύνδεσης (επειδή αυτή είναι η μόνη περίοδος

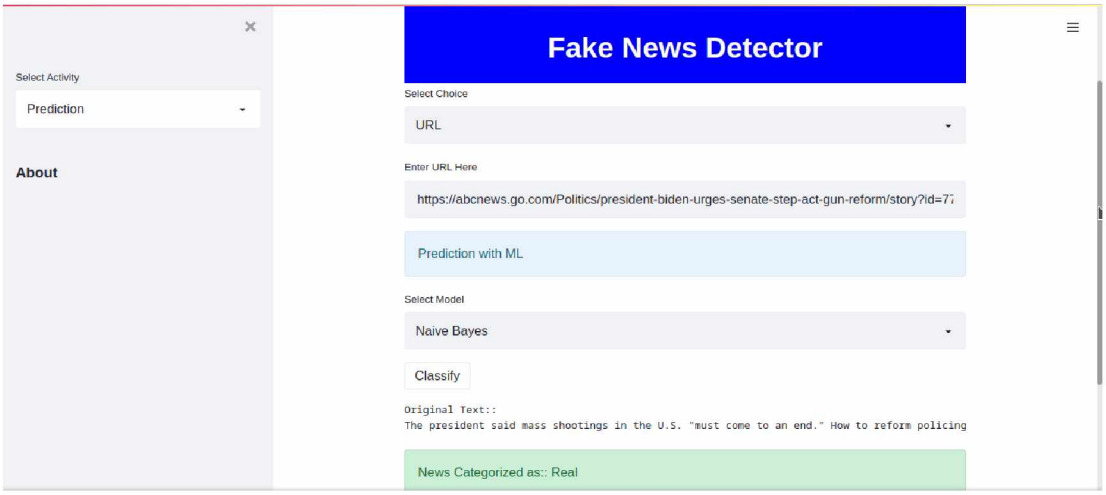
σύνδεσης που έχει ακόμη αναφορά). Στη συνέχεια, ο διακομιστής Streamlit περνάει μέσω του websocket της κάθε περιόδου, υπαγορεύει και γράφει τυχόν εκκρεμή μηνύματα από κάθε συνεδρία στο αντίστοιχο websocket. Για τον λόγο αυτό, η ίδια η αρχιτεκτονική του εργαλείου περιορίζει το πλήθος των νημάτων (και κατ' επέκταση των χρηστών που μπορούν να βρίσκονται ταυτόχρονα συνδεδεμένοι) στα 256.

Για την εφαρμογή που κατασκευάστηκε, έγινε χρήση του Streamlit σε συνδυασμό με πολλές διαφορετικές βιβλιοθήκες της Python. Η διαδικτυακή εφαρμογή δίνει την δυνατότητα για τέσσερις (4) βασικές επιλογές που αφορούν την κατηγοριοποίηση οποιονδήποτε άρθρων από οποιαδήποτε πηγή, και την εξαγωγή ορισμένων γλωσσικών χαρακτηριστικών από αυτά, που θα περιγραφούν στις ακόλουθες υποενότητες.

5.1.1 Κατηγοριοποίηση είδησης

Η κατηγοριοποίηση γίνεται με την χρήση των αλγορίθμων *Random Forest, Naive Bayes* και *Support Vector Machine* εκ των οποίων ο χρήστης μπορεί να διαλέξει. Οι τρεις κατηγοριοποιητές είναι ήδη εκπαιδευμένοι από το σύνολο δεδομένων που αναφέρθηκε στο *Κεφάλαιο 3* και σαν κριτήριο ταξινόμησης χρησιμοποιήθηκε το πιο ακριβές χαρακτηριστικό από τα δεδομένα που είναι ο αριθμός δυαδικών και τριαδικών φράσεων όπως διαπιστώθηκε στην *υποενότητα 4.2.2*. Η ανάλυση του συνόλου δεδομένων, της εξαγωγής χαρακτηριστικών και της σύγκρισης στα *Κεφάλαια 2, 3* και *4* έγιναν για τον σκοπό αυτό, δηλαδή την κατασκευή της εφαρμογής με τον καλύτερο τρόπο. Για να πακεταριστεί και να μεταφερθεί αυτή η γνώση, έχουν χρησιμοποιηθεί αρχεία της Python συγκεκριμένης μορφής, τα οποία μετατρέπουν αντικείμενα της Python (π.χ. λίστες, λεξικά κτλ) σε μορφή χαρακτήρων. Το άρθρο που δίνεται σαν είσοδος, μετατρέπεται σε μορφή πίνακα (Vectorizer) όπου μετράται η συχνότητα εμφάνισης των φράσεων, όπως ακριβώς και στην εξαγωγή του χαρακτηριστικού από το σύνολο δεδομένων. Οι ήδη εκπαιδευμένοι αλγόριθμοι, παίρνουν την απόφαση της ποιότητας της εισερχόμενης πληροφορίας.Τέλος, στον χρήστη δίνεται και η δυνατότητα επιλογής επικόλλησης του ενιαίου εντοπιστή πόρου (URL[11]) και εντοπισμός του άρθρου από την ίδια την εφαρμογή ή επικόλλησης του άρθρου χειροκίνητα από τον ίδιο τον χρήστη. Η επιλογή αυτή δίνεται σε όλες τις επιλογές (features) που προσφέρει η εφαρμογή. Στην *Εικόνα 5.1* δίνεται ένα στιγμιότυπο της εφαρμογής αφού έχουμε επικολλήσει ένα διαδικτυακό άρθρο που αφορά στον Πρόεδρο της Αμερικής Joe Biden, με χρήση συνδέσμου και με ταξινομητή τον Naive Bayes.

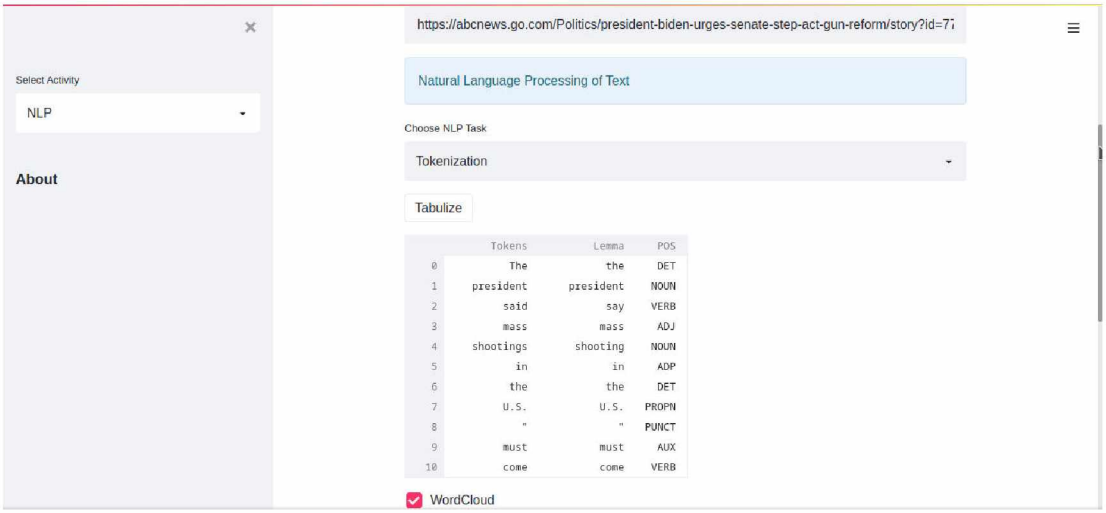
[11] δηλώνει μια διεύθυνση ενός πόρου του Παγκόσμιου Ιστού.



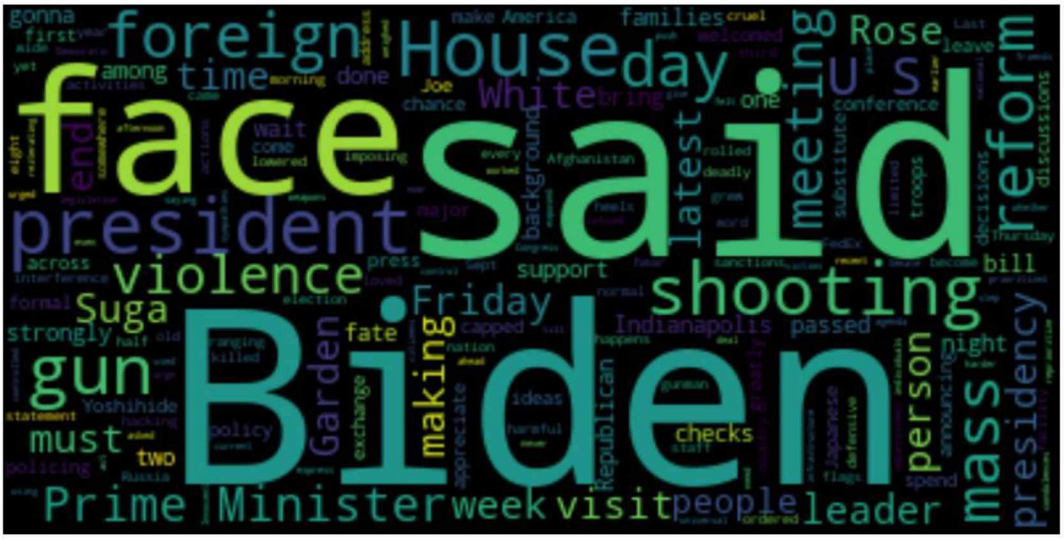
Εικόνα 5.1: Στιγμιότυπο της εφαρμογής ύστερα από κατηγοριοποίηση της είδησης σε Αληθής (Real).

5.1.2 Γλωσσικά χαρακτηριστικά

Μια ακόμη επιλογή αφορά στην εξαγωγή χαρακτηριστικών με τις μεθόδους εξαγωγής και προεπεξεργασίας γλώσσας που αναφέρθηκαν στις *υποενότητες 4.2.2* και *2.1.2*. Πιο συγκεκριμένα, η επιλογή αυτή εμφανίζει σε μορφή πίνακα την κάθε λέξη σαν ένα διαφορετικό *Token*, το Λήμμα της κάθε λέξης (*Lemma*) καθώς και το μέρος του λόγου της (*POS*). Επιπροσθέτως, δίνεται η δυνατότητα δημιουργία ενός *σύννεφου λέξεων (word cloud)*, ενός οπτικού γνωρίσματος δηλαδή που παρουσιάζει αναλογία μεγέθους λέξης με συχνότητα στο κείμενο. Η *Εικόνα 5.2* αποτελεί στιγμιότυπο της εφαρμογής του ίδιου συνδέσμου και της εμφάνισης του πίνακα ενώ η *Εικόνα 5.3* είναι το σύννεφο λέξεων. Για τα γνωρίσματα αυτά έγινε χρήση των *NLTK* και *wordcloud*.



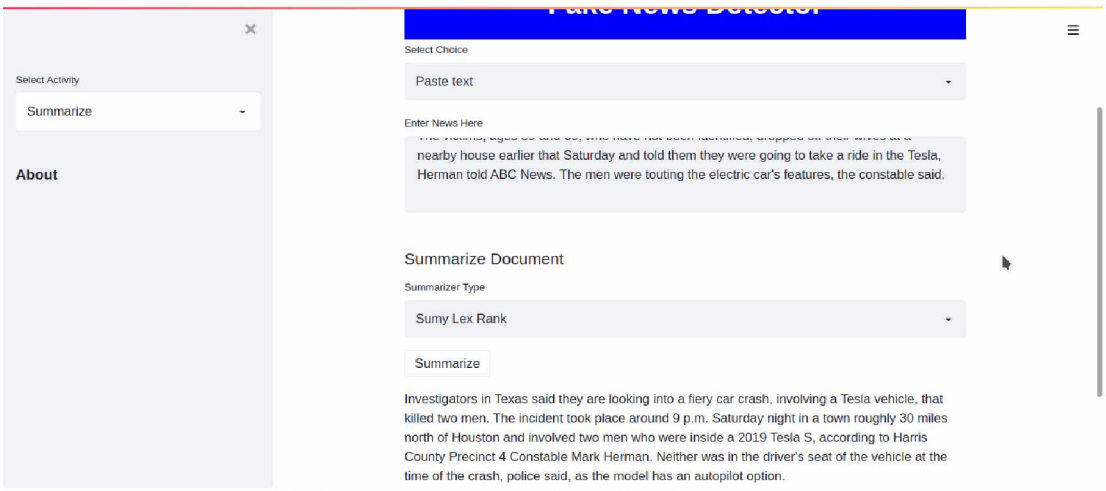
Εικόνα 5.2: Στιγμιότυπο της εφαρμογής ύστερα από την επιλογή των γλωσσικών χαρακτηριστικών.



Εικόνα 5.3: Σύννεφο λέξεων για το τυχαίο άρθρο.

5.1.3 Περίληψη κειμένου

Για την περίληψη του κειμένου χρησιμοποιούνται τεχνικές εύρεσης των σημαντικότερων λέξεων ή φράσεων για το κείμενο, οι οποίες και κρατιούνται. Η περίληψη ακολουθεί κάποιους κανόνες έκτασης καθώς το κειμενο που προκύπτει πρέπει να είναι περίπου το 30% του αρχικού κειμένου. Η επιλογή αυτή υπάρχει για τους αναγνώστες με έλλειψη χρόνου που θέλουν να αποκτήσουν την πληροφορία με ελάχιστο διάβασμα. Για την επίτευξη της περίληψης χρησιμοποιούνται τα πακέτα της Python *Gensim* και *Sumy Lex Rank*. Για την επίδειξη του συγκεκριμένου γνωρίσματος έγινε χειροκίνητη επικόλληση ενός άρθρου που αφορά σε ένα αυτοκίνητο της Tesla με χρήση του *Sumy Lex Rank*. Η *Εικόνα 4.4* μας δείχνει την περίληψη.

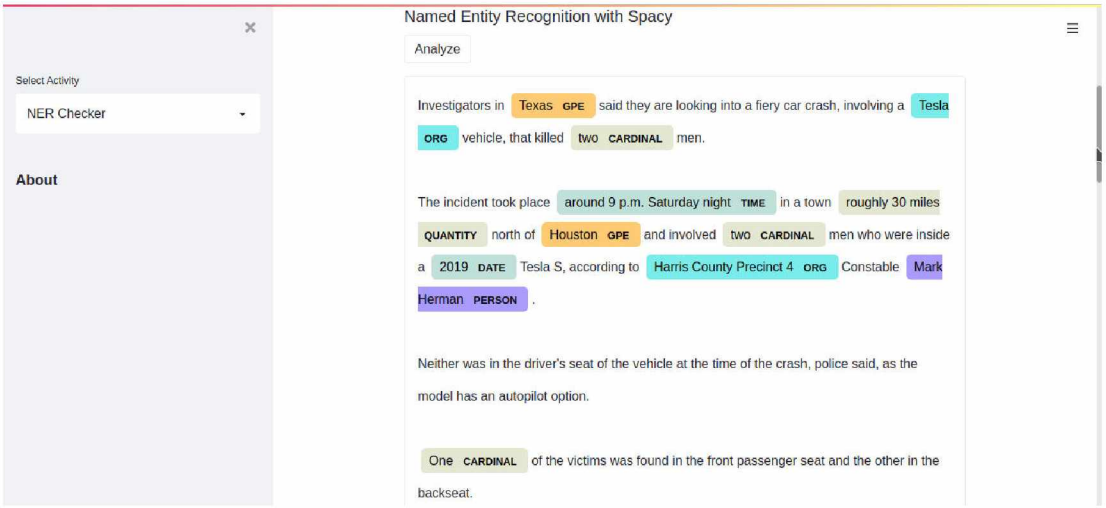


Εικόνα 5.4: Περίληψη του άρθρου με την χρήση του Sumy Lex Rank.

5.1.4 Αναγνώριση ονοματικών οντοτήτων

Σε αυτή την τελευταία δυνατότητα που δίνεται από την εφαρμογή υπάρχει οπτική αναγνώριση των ονοματικών οντοτήτων μέσα από το κείμενο, με τρόπο

εμφανή στον αναγνώστη. Η διαδικασία αυτή γίνεται με την χρήση της *spaCy* για λόγους που αναφέρθηκαν στην *υποενότητα 3.2.2.4* και *3.2.3*. Στην *Εικόνα 5.5* υπάρχουν όλες οι ονοματικές οντότητες που βρέθηκαν στο άρθρο της Tesla με οπτικό τρόπο ενώ στην *Εικόνα 5.6* είναι οι ίδιες οντότητες σε μορφή λίστας.



Εικόνα 5.5: Αναγνώριση ονοματικών οντοτήτων με οπτικό τρόπο

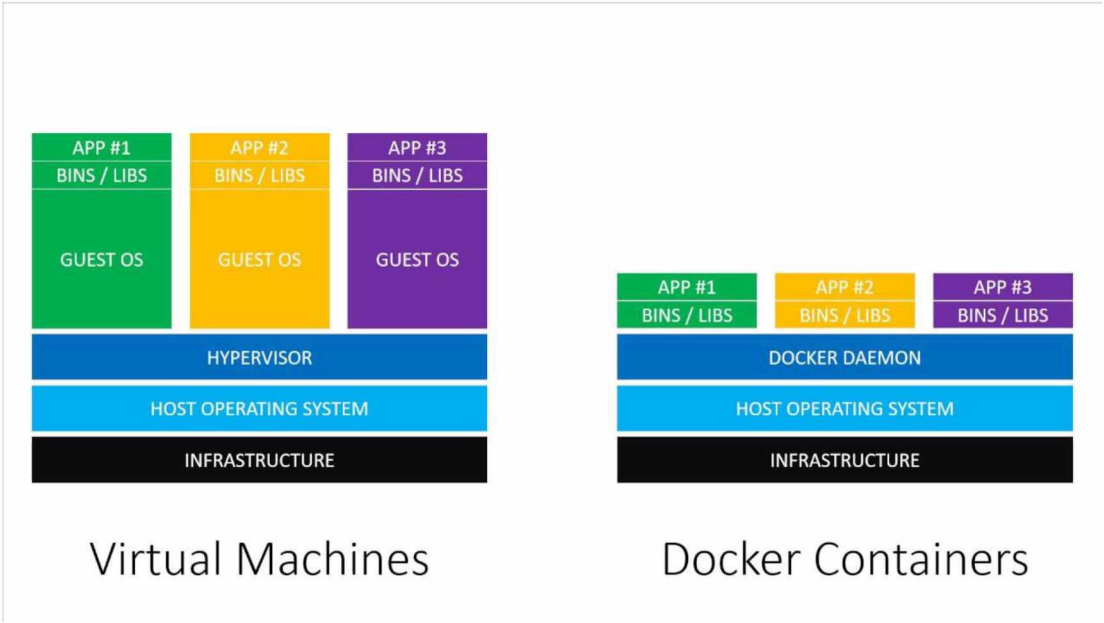
Εικόνα 5.6: Αναγνώριση ονοματικών οντοτήτων σε μορφή λίστας

5.2 Docker containers

Τα *Docker Containers* είναι βασικά δομικά στοιχεία του συστήματος. Το Docker είναι μια πλατφόρμα λογισμικού ανοιχτού κώδικα που υλοποιεί Εικονικοποίηση (Virtualization) σε επίπεδο Λειτουργικού Συστήματος. Ουσιαστικά το Docker προσφέρει αυτοματοποιημένες διαδικασίες για την ανάπτυξη εφαρμογών σε απομονωμένες Περιοχές Χρήστη (User Spaces) που ονομάζονται Software Containers. Το λογισμικό χρησιμοποιεί τεχνολογίες του πυρήνα του Linux όπως τα cgroups και οι χώροι ονομάτων πυρήνα (kernel namespaces), για να επιτρέπει σε ανεξάρτητα software containers να εκτελούνται στο ίδιο λειτουργικό σύστημα. Συνοπτικά, μας επιτρέπει να τρέξουμε και να μεταφέρουμε μια εφαρμογή χωρίς να υπάρχει εξάρτηση από τον εξυπηρετητή. Μοιάζουν πολύ με τις εικονικές μηχανές (virtual machines) ωστόσο διαφέρουν στην διαχείριση των πόρων. Αναλυτικότερα, οι

εικονικές μηχανές αποτελούν αυτόνομοι διακομιστές που τρέχουν το δικό τους λειτουργικό σύστημα, μέσα σε έναν κεντρικό διακομιστή (hypervisor) του οποίου τους πόρους (π.χ. cpu, μνήμη κτλ) μοιράζονται οι εικονικοί διαχειριστές. Από την άλλη μεριά, τα containers εκτελούνται και αυτά σε έναν διακομιστή (φυσικό ή εικονικό) όμως πέρα από τους πόρους μοιράζονται και τον πυρήνα του λειτουργικού συστήματός του. Έτσι δεν απαιτείται διαφορετικό λειτουργικό για κάθε ξεχωριστό container. Κατά αυτόν τον τρόπο τα containers δεν έχουν μεγάλες απαιτήσεις σε πόρους του συστήματος και έτσι εκκινούν γρηγορότερα από μια εικονική μηχανή. Επιπλέον, το μέγεθος των στιγμιοτύπων (*images*[12]) αποτελεί ένα ακόμα πλεονέκτημα καθώς μπορεί να είναι της τάξης των μόλις μερικών MBs γεγονός που οφείλεται στην απουσία πυρήνα και στην έλλειψη χρήσης επιπρόσθετων βιβλιοθηκών.

Εφόσον ο σκοπός χρήσης των δύο αυτών τεχνολογιών είναι παρόμοιος, η ανεξαρτητοποιημένη εκτέλεσης της εφαρμογής, συχνά οι δύο τεχνολογίες συγχέονται ή και ταυτίζονται. Ωστόσο, έχουν διαφορά στην λειτουργία τους και έτσι πρέπει να χρησιμοποιούνται σε διαφορετικές περιπτώσεις. Όσον αφορά την κατασκευή μιας εφαρμογής, είναι σημαντικό να αναφερθεί ότι στην περίπτωση των εικονικών μηχανών πρέπει να κατέβει ο κώδικας της εφαρμογής, τα εκτελέσιμα, οι βιβλιοθήκες της Python, τα δεδομένα που χρειάζονται για το λειτουργικό και οτιδήποτε χρειάζεται για το συστημα (π.χ. οδηγοί συσκευών κτλ) ενώ στην περίπτωση των containers το μόνο που χρειάζεται για την λειτουργία της εφαρμογής είναι αρχεία που αφορούν στην ίδια την εφαρμογή (*Εικόνα 5.7*).



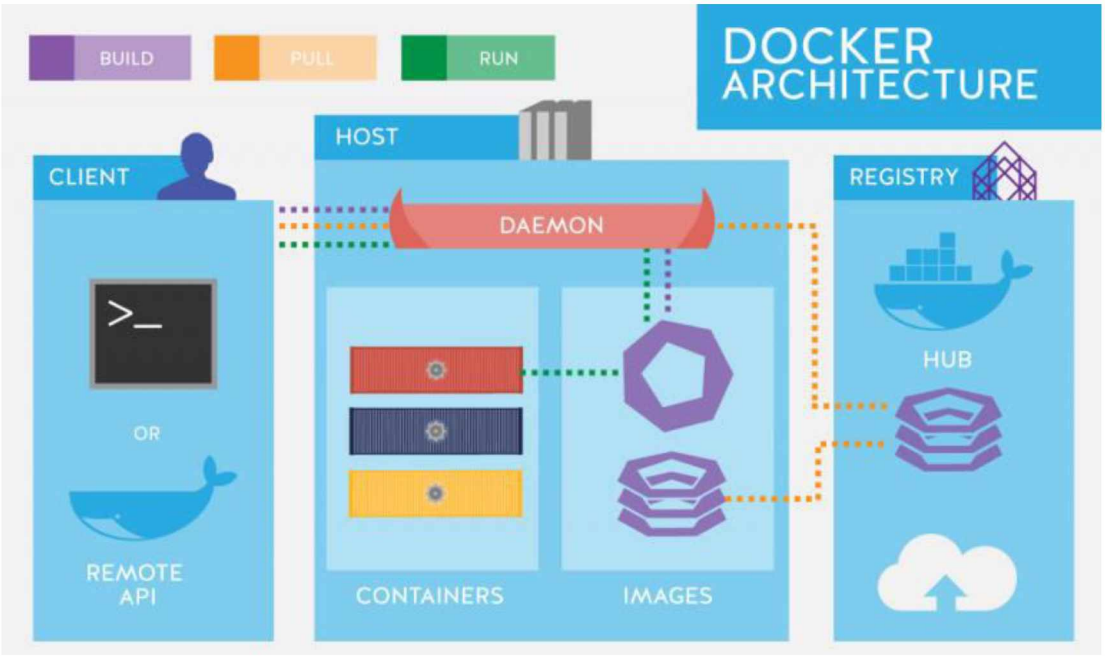
Εικόνα 5.7: Σύγκριση Εικονικών μηχανών και Docker containers. Πηγή:docker.com

Τα containers πρέπει να λαμβάνονται υπόψη ως κάτι εφήμερο που μπορεί να καταστραφεί ανά πάσα στιγμή, γι αυτό και δεν θεωρείται καλή πρακτική η αποθήκευση δεδομένων σε αυτά όπως για παράδειγμα μια *Βάση δεδομένων*.

[12] όπως και στις εικονικές μηχανές έτσι και στα containers υπάρχει ένα αρχικό στιγμιότυπο από όπου εκκινεί το container.

5.2.1 Τα βασικά στοιχεία του λογισμικού Docker

Η αρχιτεκτονική του Docker χρησιμοποιεί ένα μοντέλο πελάτη- εξυπηρετητή (client- server) που επικοινωνούν μέσω ενός REST API, με την διαμεσολάβηση διεπαφής χρήστη από την γραμμή εντολών. Ο πελάτης Docker επικοινωνεί με τον Docker δαίμονα που αναλαμβάνει όλες τις διεργασίες όπως είναι το χτίσιμο, τρέξιμο και η διανομή της εφαρμογής. Ο πελάτης και ο εξυπηρετητής Docker μπορούν να βρίσκονται στο ίδιο μηχάνημα ή να έχουν απομακρισυνδεση σύνδεση (*Εικόνα 5.8)*

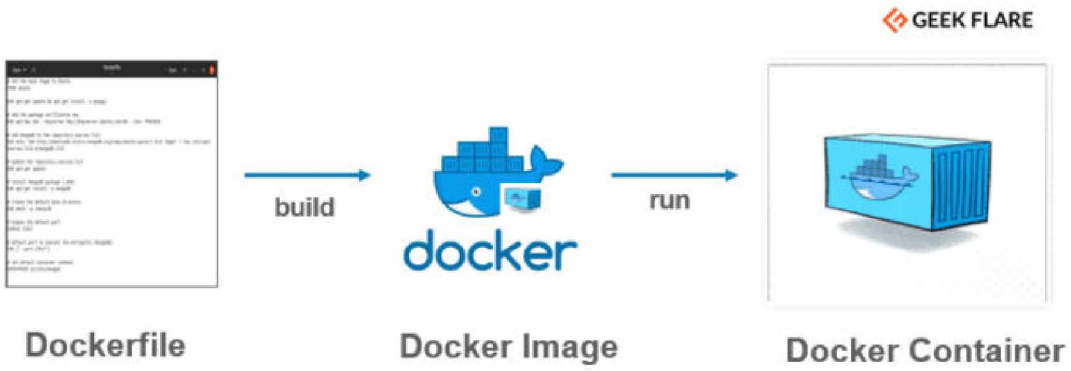


Εικόνα 5.8: Το Docker με τα βασικά του δομικά στοιχεία (client, daemon, registry στο Hub)

Το Docker Hub είναι μια δωρεάν δημόσια cloud υπηρεσία που μας παρέχει το Docker για τον διαμοιρασμό και την κοινοποίηση Docker container στιγμιοτύπων. Κάθε φορά που αιτούμαστε ένα συγκεκριμένο στιγμιότυπο ο διαχειριστής δαίμονας το "τραβάει" από το κοινό αποθετήριο. Ομοίως εάν θέλουμε να μοιραστούμε ένα σε όλη την κοινότητα ο δαίμονας μοιράζεται τις αλλαγές μας.

5.2.2 Τρόπος λειτουργίας των στιγμιοτύπων και των συστημάτων αρχείων τύπου UnionFS

Όπως αναφέρθηκε ήδη, κάθε container για να εκκινήσει χρειάζεται μια αρχική εικόνα (base image). Το στιγμιότυπο αυτό μπορεί να σε μια διανομή Linux π.χ. Ubuntu και να περιέχει ορισμένα βασικά εργαλεία και βιβλιοθήκες. Λόγω των ιδιαίτερων χαρακτηριστικών του συστήματος αρχείων του Docker το οποίο θα περιγραφεί στην συνέχεια, τα στιγμιότυπα δίνουν μόνο την δυνατότητα ανάγνωσης και όχι αποθήκευσης δεδομένων (read-only). Επομένως, συχνά χρειάζεται η κατασκευή νέων στιγμιότυπων για την εκτέλεση της εφαρμογής. Αυτό γίνεται με την χρήση ενός **Dockerfile.** Το Dockerfile είναι ουσιαστικά ένα αρχείο που περιέχει τυποποιημένες εντολές με συγκεκριμένη σύνταξη για την εκτέλεση της εφαρμογής. Όπως και στην περίπτωση του Streamlit, το Dockerfile αποτελεί ένα είδους script όπου η εκτέλεση των οδηγιών γίνεται από πάνω προς τα κάτω *(Εικόνα 5.9).*

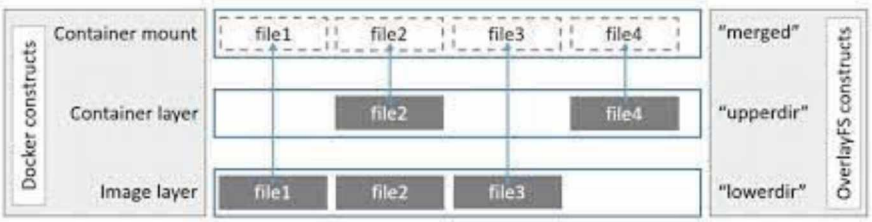


Εικόνα 5.9: Τρόπος λειτουργίας του συστήματος αρχείων Docker.

Κάθε μια εντολή του Dockerfile ανοικοδομεί και ένα ακόμα επίπεδο στην εφαρμογή μας. Στο πρώτο επίπεδο, στην βάση, βρίσκεται ο πυρήνας. Στην συνέχεια ακολουθεί το αρχικό στιγμιότυπο το οποίο αποτελείται από το σύστημα αρχείων ρίζας (rootfs) και οποιοδήποτε άλλο επίπεδο. Αν και κατά την εκκίνηση μιας συνηθισμένης μηχανής Linux το rootfs θα έμπαινε αρχικά σε κατάσταση read-only, και στην συνέχεια θα ήταν έτοιμο προς εγγραφή, κάτι τέτοιο δεν συμβαίνει στα Docker containers. Το rootfs μαζί με τα υπόλοιπα επίπεδα παραμένει σε κατάσταση read-only. Στην κορυφή της λίστας βρίσκεται το μοναδικό επίπεδο στο οποίο επιτρέπεται ανάγνωση αλλά και εγγραφή ακι είναι αυτό στο οποίο αντιστοιχεί η κατάσταση εκτέλεσης του εκάστοτε container.

Για να γίνει καλύτερα η κατανόηση της αρχής λειτουργίας των στιγμιοτύπων θα πρέπει να εξεταστούν το βασικά χαρακτηριστικα του συστήματος αρχείων τύπου *UnionFS*. Στόχος του συστήματος αυτού είναι η διαχείριση αρχείων και καταλόγων που μπορεί να είναι αποθηκευμένα σε διαφορετικά συστήματα αρχείων σαν ένα ενιαίο σύστημα. Το Docker υποστηρίζει διάφορους μηχανισμούς που υλοποιούν την λειτουργικότητα του *UnionFS* όπως οι AUFS, OverlayFS, Btrfs, Device Mapper και άλλοι. Οι πιο διαδεδομένοι και ευρέως χρησιμοποιούμενοι είναι οι AUFS και OverlayFS με τον δεύτερο να υπάρχει σε δύο εκδόσεις. Αν και βασίζονται στις ίδιες αρχές λειτουργίας υπάρχουν διαφορές μεταξύ τους ανάλογα το λειτουργικό σύστημα που πρόκειται να χρησιμοποιηθεί, το είδος της εφαρμογής καθώς και το σύστημα αρχείων που έχει ο εξυπηρετητής που φιλοξενεί την υπηρεσία Docker. Για την εργασία αυτή χρησιμοποιήθηκε το OverlayFS2 που είναι η δεύτερη έκδοση του OverlayFS, γρηγορότερο από το AUFS. Για το λειτουργικό σύστημα και το σύστημα αρχείων στην συσκευή όπου φιλοξενείται το Docker, επιλέχθηκαν τα Linux- kubuntu και Ext4 αντίστοιχα.

Το OverlayFS είναι απλούστερο στην υλοποίηση από το AUFS. Στην *Εικόνα 5.10* δύο κατάλογοι αναπαριστώνται ως στρώματα (layers), με το κατώτερο αν είναι το αρχικό στιγμιότυπο (*"lowerdir"*) ενώ το επίπεδο του container να είναι το ανώτερο (*"lowerdir"*). Το επίπεδο *"merged"* δίνει την συνολική εικόνα των δεδομένων που υπάρχουν στα κατώτερα επίπεδα.



Εικόνα 5.10: : Σχηματική αναπαράσταση του τρόπου λειτουργίας του OverlayFS. Πηγή: docker.com

Ενώ το επίπεδο του στιγμιοτύπου επιτρέπει μόνο την ανάγνωση δεδομένων το επίπεδο του container επιτρέπει και την εγγραφή. Με αυτόν τον τρόπο μπορούμε να αλλάξουμε το περιεχόμενο ενός container και να αποθηκεύσουμε τις αλλαγές μας σαν ένα νέο στιγμιότυπο. Για να πραγματοποιηθεί αυτή η διαδικασία το OverlayFS ενεργοποιεί την επιλογή του "*copy-on-write*", όπου ένα στιγμιότυπο που μπορεί να βρίσκεται στα κατώτερα επίπεδα ενός στιγμιότυπου, που όπως αναφέρθηκε είναι read-only, αντιγράφεται στα ανώτερα επίπεδα ενός καινούργιου στιγμιοτύπου. Σε περιπτώσεις που υπάρχει τροποποίηση μεγάλων αρχείων, υπάρχει καθυστέρηση και και πτώση της απόδοσης. Ωστόσο, η διαδικασία αυτή θα λάβει χώρα μόνο την πρώτη φορά που θα προσπελαστεί το εκάστοτε αρχείο. Εάν το αρχείο τροποποιηθεί για δεύτερη φορά, τότε θα προσπελαστεί απευθείας από το επίπεδο του container

Ένα βασικό πλεονέκτημα που προσφέρει το OverlayFS, ως ένα σύστημα αρχείων τύπου Union, είναι ότι επιτρέπει την επαναχρησιμοποίηση κάθε επιπέδου από διαφορετικά στιγμιότυπα. Έτσι εξοικονομείται αποθηκευτικός χώρος ενώ παράλληλα μειώνεται και ο χρόνος που απαιτείται για την δημιουργία ενός στιγμιότυπου που χρησιμοποιουνται τα υπάρχοντα επίπεδα. Ακόμη ένα επίπεδο μπορεί να χρησιμοποιείται ταυτόχρονα από περισσότερα του ενός στιγμιότυπα. Αν στην πορεία χρειαστεί να γίνει οποιαδήποτε αλλαγή στα αρχεία τότε εκκινούν νέα στιγμιότυπα τα οποία αντικαθιστούν σταδιακά τα παλιά.

5.3 Docker Swarm και ενορχήστρωση containers

Ένα από τα προβλήματα που προκύπτουν σχετικά με τα Docker containers είναι η διαχείρισή τους. Πώς δηλαδή θα διεκπαιρεώνονται εργασίες που αφορούν στην εκκίνηση και στο χτίσιμο της εφαρμογής, στην εκτέλεση των containers, στην εξισορρόπηση του φορτίου, στην ανοχή σε σφάλματα, στην αυτόματη επιδιόρθωση κτλ. Ιδανικά αυτές οι εργασίες θα πρέπει να γίνονται με όσο το δυνατόν λιγότερη παρέμβαση από τον χρήστη. Γι' αυτό τον λόγο το λογισμικό Docker προσφέρει την υπηρεσία του Docker Swarm, ένα εργαλείο ενορχήστρωσης και ομαδοποιησης των containers. Συνοπτικά, το Docker Swarm χωρίζει τα containers σε σμήνη ή clusters με σκοπό την εκτέλεση εργασιών σε μορφή containers. Η λειτουργία του βασίζεται στην χρήση της βιβλιοθήκες SmartKit. Το SmartKit είναι ένα ξεχωριστό λογισμικό ανοικτού κώδικα που περιλαμβάνει ένα σύνολο από εργαλεία ενορχήστρωσης για κατανεμημένα συστήματα. Ουσιαστικά το Docker Swarm αξιοποιεί πλήρως το Docker API κάνοντας την χρήση του πολύ απλή και σύντομη και επιλέχθηκε έναντι του Kubernetes, το οποίο ανήκει στην ίδια κατηγορία, λόγω της απλότητας στην εγκατάσταση και την παραμετροποίησή του καθώς και της εγγενούς υποστήριξης που παρέχει για το Docker. Οι βασικές εντολές είναι τρεις:

- **Docker swarm:** για την αρχικοποίηση της συστοιχίας καθώς επίσης και την είσοδο ή έξοδο κόμβων σε/από αυτή.
- **Docker node**: για τον έλεγχο των κόμβων που συμμετέχουν σε μία συστοιχία.
- **Docker service**: για την εποπτεία των υπηρεσιών που τρέχουν στη συστοιχία.

Σε μια συστοιχία Docker Swarm οι κόμβοι χωρίζονται σε δύο κατηγορίες, τους *managers* και τους *workers*. Οι managers αναλαμβάνουν τον προγραμματισμό της εκτέλεσης των διεργασιών (tasks) από τους workers, εξυπηρετούν τις εισερχόμενες κλίσεις του API ενώ παράλληλα συγκρατούν όλη την πληροφορία των μεταδεδομένων για την κατάσταση της συστοιχίας. Οι workers είναι οι κόμβοι που

είναι επιφορτισμένοι με την εκτέλεση των containers. Οι managers μπορεί να λειτουργούν σαν workers, οι workers δεν εχουν συντονιστικό ρόλο, παρά μόνο εάν χειροκίνητα αλλάξει θέση από worker σε manager.

Η υψηλή διαθεσιμότητα και η ανοχή σε σφάλματα είναι ανάλογη του αριθμού των κόμβων που συμμετέχουν σε ένα σμήνος. Πιο συγκεκριμένα εξαρτάται από τον αριθμό των managers που υπάρχουν. Για παράδειγμα, ένα σύστημα με πέντε (5) manager κόμβους, μπορεί να λειτουργήσει απρόσκοπτα ακόμα και εάν χάσει δύο (2) από τους managers. Γενικά ο τύπος *(N-1)/2* όπου *N* είναι ο αριθμός των διαθέσιμων managers μάς δίνει το μέγιστο αριθμό κόμβων που μπορεί να αποτύχουν χωρίς να επηρεαστεί η ομαλή λειτουργία της συστοιχίας. Η ύπαρξη πολλών managers συνδράμει στην πρόληψη σφαλμάτων στο σύστημα, όπως προαναφέρθηκε, όμως δεν είναι πάντα αναγκαία και μάλιστα μπορεί να αποβεί και επιζήμια σε ορισμένες περιπτώσεις. Οι καθυστερήσεις που υπεισέρχονται στο σύστημα για την επικοινωνία των κόμβων αποτελεί έναν τέτοιον κινδυνο. Γι' αυτό και πριν γίνει επιλογή του αριθμού των κόμβων της συστοιχίας πρέπει να εξεταστούν οι απαιτήσεις της εφαρμογής και του περιβάλλοντος εκτέλεσής της. Η επιλογή αυτή γίνεται σύμφωνα με τον αλγόριθμο *Raft*. Με βάση αυτόν, η συστοιχία αποτελείται από μια ομάδα κόμβων/διεργασιών (quorum) όπου υπάρχει ένας manager. Αυτός είναι υπεύθυνος για την ροή δεδομένων από ανάμεσα στους κόμβους όσο είναι ενεργός. Σε περίπτωση σφάλματος, εκλέγεται νέος αρχηγός [29]. Είναι ένας αλγόριθμος που ξεκίνησε να χρησιμοποιείται στο *Blockchain*[13] αλλά συνέχισε σε διάφορα πεδία.

Στην περίπτωση της παρούσας εργασίας, επιλέχθηκαν δύο κόμβοι σαν σμήνος, ένας που είναι ταυτόχρονα manager και worker, και ένας που είναι μόνο worker. Συνολικά δηλαδή δύο workers και ένας manager. Δεν δόθηκε βαρύτητα στην υψηλή διαθεσιμότητα της συστοιχίας καθότι προορίζεται για την επίδειξη της πλατφόρμας στο σύνολό της. Οι κόμβοι είναι φυσικές μηχανές, για λόγους εξοικονόμησης πόρων αλλά και διότι χρησιμοποιήθηκαν containers, άρα δεν υπήρξε λόγος χρήσης των εικονικών μηχανών. Τα χαρακτηριστικά των μηχανών περιγράφονται παρακάτω. Για την μηχανή manager/worker:

- **OS (Operating System)**: Linux- kubuntu
- **CPU (Processor)**: 4 cores
- **RAM**: 8GB
- **HDD**: 116.9GB
- **SSD**: 41GB

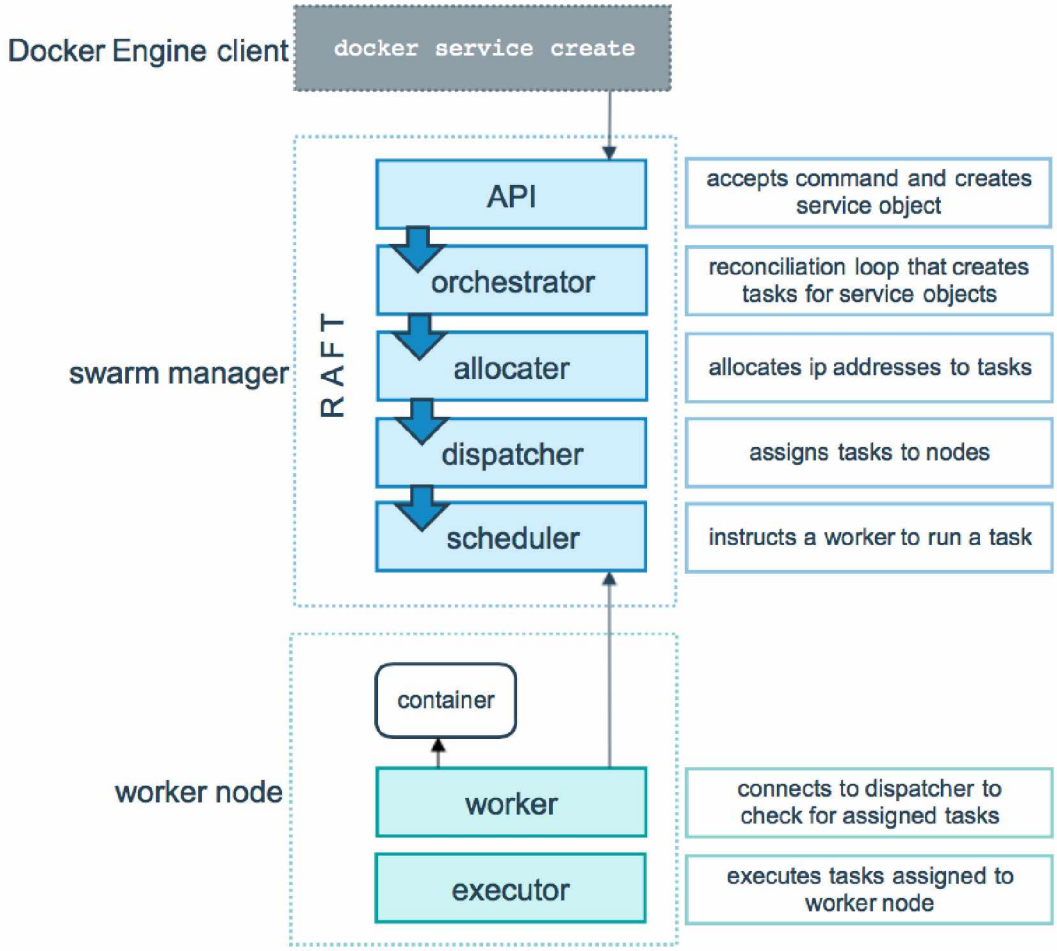
Ενώ για την μηχανή worker:

- **OS (Operating System)**: Linux- kubuntu
- **CPU (Processor)**: 4 cores
- **RAM**: 6GB
- **HDD**: 230GB

Για την εκτέλεση των containers σε μια συστοιχία Docker Swarm χρησιμοποιούνται τα ***services***. Τα services δεν είναι τίποτα άλλο παρά οι οδηγίες που χρειάζονται όπου βρίσκονται δηλωμένες οι απαραίτητες παράμετροι όπως το στιγμιότυπο για την δημιουργία της εφαρμογής, οι περιβαλλοντικές μεταβλητές, τυχόν περιορισμοί σε RAM, CPU κτλ. Αυτό μπορεί να πραγματοποιηθεί με την σύνταξη ενός αρχείου παραμέτρων τύπου ***"YAML"*** το οποίο συνήθως χωρίζεται σε δύο τμήματα. Το πρώτο τμήμα αφορά στα βασικά στοιχεία της πλατφόρμας όπως το όνομα και την διεύθυνση AP Gateway ενώ το δεύτερο αφορά στα functions. Έτσι

[13] Η πρώτη ιστορικά, εφαρμογή της τεχνολογίας πραγματοποιήθηκε στον χώρο των ψηφιακών νομισμάτων, και ήταν η περίπτωση του bitcoin.

περιέχονται το όνομα του αρχείου που βρίσκεται ο κώδικας για το εκάστοτε function καθώς και το στιγμιότυπο πάνω στο οποίο θέλουμε να εκτελεστεί το contaner που θα δημιουργηθεί από το function. Με την δημιουργία των services η δημιουργία της εφαρμογής δρομολογείται από τον manager δημιουργώντας ένα task (*Εικόνα 5.11).*



Εικόνα 5.11: Ο τρόπος εκτέλεσης ενός Docker service. Πηγή: docker.com

Για κάθε ένα task ο manager εκκινεί και ένα container. Εάν στην πορεία κάποιο από αυτά τα task αποτύχει, τότε ο manager το διαγράφει μαζί με το container και εκκινεί ένα νέο.

5.4 Περιγραφή ροής εκτέλεσης

Ο κώδικας αναπτύχθηκε σε Python 3 και χρησιμοποιήθηκε το περιβάλλον PyCharm CE. Αποτελείται από τρία (3) επιμέρους κομμάτια. Το πρώτο περιλαμβάνει την *Διαγνωστική Ανάλυση Δεδομένων (Explanatory Data Analysis)* όπου γίνεται μια διερεύνηση των δεδομένων με γραφικά και στατιστικά στοιχεία για την εξοικείωση με αυτά και την καλύτερη κατανόησή τους. Το δεύτερο κομμάτι αποτελεί τον κώδικα της διαδικασία που περιγράφηκε στο *Κεφάλαιο 3*, για την εξαγωγη χαρακτηριστικών και την εκπαίδευση των ταξινομητών αξιοποιώντας τις βιβλιοθήκες sklearn και scikit-learn της Python. Αποτελείται από τρία (3) αρθρώματα κώδικα. Τέλος, το τρίτο κομμάτι περιλαμβάνει τον κώδικα της εφαρμογής μέσα σε Docker containers που αποτελείται από ένα κεντρικό άρθρωμα.

Η φάση της εκπαίδευσης πραγματοποιείται τοπικά. Για τα δεδομένα εκπαίδευσης χρησιμοποιείται το άρθρωμα εξαγωγής γνωρισμάτων για την επεξεργασία τους. Να σημειωθεί εδώ τα γνωρίσματα κανονικοποιούνται ώστε να φέρουν τιμές στο εύρος [0, 1]. Τα μοντέλα που προκύπτουν από την εκπαίδευση των αλγορίθμων, αποθηκεύονται ως **αντικείμενα** με χρήση της βιβλιοθήκης cPickle ώστε να φορτωθούν αργότερα κατά την εκτέλεση της εφαρμογής.Για τον λόγο αυτό, ο αριθμός των συνδεδεμένων χρηστών δεν επηρεάζει την ταχύτητα της εξυπηρέτησής τους, παρά μόνο ο όγκος του άρθρου καθώς η εξαγωγή χαρακτηριστικών σε ένα μεγαλύτερο κείμενο μπορεί να αποβεί πιο χρονοβόρα με μικρές μεταβολές. Η εφαρμογή μπορεί να θεωρηθεί και ένα (1) function από τα containers.
Τα επιμέρους αρθρωτά που το απαρτίζουν είναι:

- το άρθρωμα για τον καθαρισμό και την προεπεξεργασία των δεδομένων
- το άρθρωμα για την εξαγωγή των γνωρισμάτων
- το άρθρωμα για τον έλεγχο της ορθότητας
- Το λεξικό όρων stanford-core NLP

και παράγει, εκτός από τα αποτελέσματα ορθότητας για την ταξινόμηση:

- το μοντέλο πρόβλεψης για το χαρακτηριστικό που επιλέχθηκε (σε μορφή *.pickle)*
- τα τρια (3) μοντέλα πρόβλεψης για τους τρεις ταξινομητές (σε μορφή *.pickle)*
- τους τρεις (3) πίνακες σύγχυσης που είδαμε στις *υποενότητες 4.3.1, 4.3.2* και *4.3.3*

Εδώ επειδή χρησιμοποιείται ένα μόνο μοντέλο εκπαίδευσης, που αναπαρίσταται από ένα μόνο function δηλαδή ένα Dockerfile μέσω του οποίου δημιουργείται το ανάλογο στιγμιότυπο. Το στιγμιότυπο αυτό περιέχει:

- το μοντέλο πρόβλεψης για το χαρακτηριστικό που επιλέχθηκε (σε μορφή *.pickle)*
- τα τρια (3) μοντέλα πρόβλεψης για τους τρεις ταξινομητές (σε μορφή *.pickle)*
- το άρθρωμα για το API της εφαρμογής
- Το αρχείο requirements μέσα στο οποίο ορίζονται οι επιπλέον Python εξαρτήσεις που απαιτούνται.

Αφού γίνει η μεταφορά των παραπάνω στοιχείων στον κόμβο όπου έχει εγκατασταθεί το Streamlit, πραγματοποιείται το "χτίσιμο" των στιγμιοτύπων και η εκκίνηση των containers τα οποία πλέον σερβίρουν τους αλγορίθμους ταξινόμησης ως ένα function. Να σημειώσουμε εδώ ότι το μοντέλο πρόβλεψης θα μπορούσε να βρίσκεται αποθηκευμένο σε κάποιο αποθετήριο και να κατεβαίνει από εκεί κάθε φορά που δημιουργείται ένα νέο στιγμιότυπο. Αρχικά πρέπει να γίνει διανομή της εφαρμογής τοπικά με την εντολή *"docker run"* ώστε να "τρέξει" το περιεχόμενο του στιγμιοτύπου στο συγκεκριμένο container *(Παράδειγμα 5.1).*

```
$ docker build -t <USERNAME>/<YOUR_IMAGE_NAME> .
$ docker run -p 8501:8501 <USERNAME>/<YOUR_IMAGE_NAME>
```

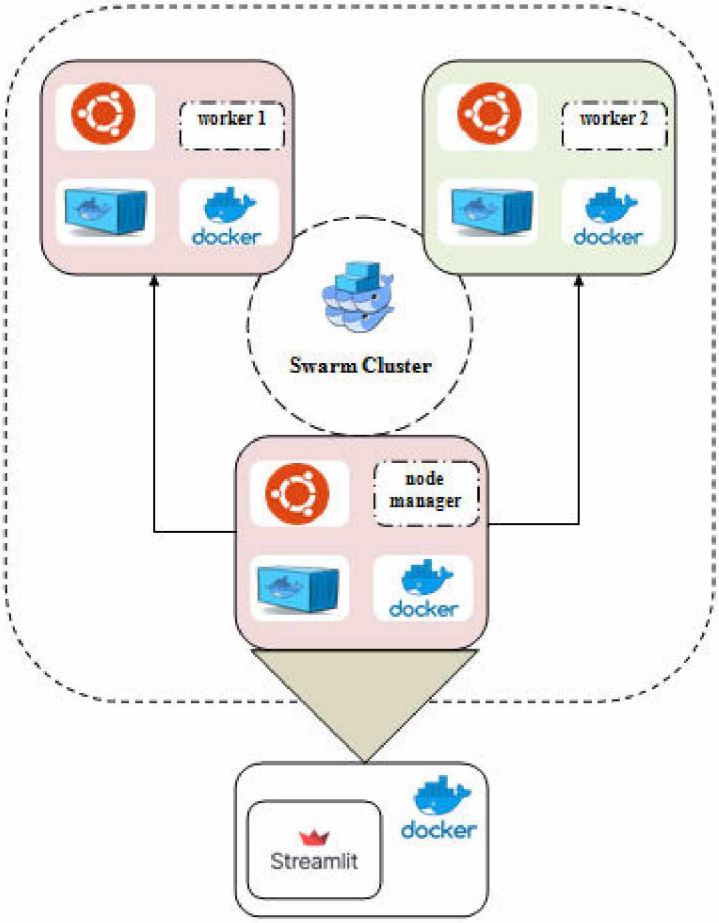
Παράδειγμα 5.1: Χτίσιμο και εγκατάσταση του στιγμιοτύπου σε container.

Το χτίσιμο και η εγκατάσταση των functions στο Docker Swarm γίνεται όπως αναφέραμε και στην *ενότητα 5.3* με την χρήση ενός αρχείου παραμέτρων σε σύνταξη YAML και της εντολής "*stack deploy*" (*Παράδειγμα 5.2*). Έτσι, δημιουργείται μια υπηρεσία (service) όπου αποτελείται από επεκτάσιμες ομάδες containers με πρόσθετες δυνατότητες δικτύωσης, διατηρουμενες από το Swarm.

```
$ docker stack deploy -c <STACK_NAME>.yml <DEMO_NAME>
```

Παράδειγμα 5.2: Χτίσιμο και εγκατάσταση του service που ορίζονται στο αρχείο .yml

Έτσι, η εφαρμογή είναι πλέον διαθέσιμη από την θύρα (port) 8501 για τον παραλήπτη στο τοπικό δίκτυο και με την χρήση του IP που κατέχει ο manager κόμβος και με άνοιγμα της θύρας στο router για εξωτερικές συνδέσεις. Το πρωτόκολλο επικοινωνίας είναι το *HTTP* άρα ο διακομιστής εξυπηρετεί κάθε κλίση του function για την χρήση της εφαρμογής. Αυτό παρέχεται αυτόματα μέσω του Streamlit API. Στην *Εικόνα 5.12* φαίνεται η ροή εκτέλεσης για την δημιουργία όλου του συστήματος για την διανομή της τελικής εφαρμογής.



Εικόνα 5.12: Ροή εκτέλεσης.

6 ΕΠΙΛΟΓΟΣ

Σε αυτό το τελευταίο κεφάλαιο βρίσκεται το συμπέρασμα όλης της εργασίας καθώς και κάποιες προτάσεις για μελλοντικές επεκτάσεις.

6.1 Συμπέρασμα

Σε αυτή την εργασία παρουσιάστηκε ένας τρόπος ανίχνευσης των Ψευδών ειδήσεων τόσο σε θεωρητικό όσο και σε πρακτικό επίπεδο. Επιπλέον, το κέντρο βάρους πλαισιώνει την χρησιμότητα και την σημασία των ονοματικών οντοτήτων για τον σκοπό αυτό. Σε αντίθεση με τις μέχρι τώρα υλοποιήσεις, η εργασία αυτή ανέλυσε τις μεθόδους και τα εργαλεία εξαγωγής χαρακτηριστικών από το κείμενο με καινούργια εργαλεία όπως το *spaCy* έναντι του *NLTK* καθώς και ανέδειξε έναν εύκολο και γρήγορο τρόπο δημιουργίας *Διαδικτυακων εφαρμογών* με χρήση των βιβλιοθηκών της Python ώστε να εφαρμοστούν πρακτικά όλα όσα αποδείχτηκαν με την μελέτη αυτή.

Επιπροσθέτως, αναπτύχθηκε κώδικας σε Python και για την εκπαίδευση των αλγορίθμων εκπαίδευσης που χρησιμοποιήθηκαν (*Random Forest, Naive Bayes και Support Vector Machine).* Έγινε χρήση και σύγκριση των μοντέλων πρόβλεψης για τα οποία έγινε λεξιλογική και υφολογική ανάλυση της φυσικής γλώσσας.

Συνοψίζοντας καλό θα ήταν να αναφερθεί ότι η χρήση των Διαδικτυακών εφαρμογών είναι μια καλή προσέγγιση για την αξιοποίηση της γνώσης που παράγεται μέσω Μηχανικής Μάθησης. Σε συνδυασμό με τα *Docker containers* που απλοποιούν την διαδικασία της διανομής της εφαρμογής καθώς και το *Docker Swarm* για την γρηγορότερη εξυπηρέτηση των χρηστών αλλά και την αποφυγή σφαλμάτων στο μηχάνημα που λειτουργεί σαν διανομέας της εφαρμογής, προσφέρουν τόσο ένα φιλικό περιβάλλον για τον χρήστη όσο και μια γρήγορη πηγή γνώσης που να χαρίζει ανεξαρτησία από το λειτουργικό και το λογισμικό. Πολλές φορές ωστόσο, ο συνδυασμός των δύο αυτών τεχνολογιών μπορεί να μην είναι εφικτός, γι αυτό και ο έλεγχος του χρόνου εκτέλεσης της εφαρμογής είναι απαραίτητος σε τέτοια συστήματα.

6.2 Μελλοντικές Υλοποιήσεις

Αρχικά, στο κομμάτι της ανίχνευσης των Ψευδών Ειδήσεων πιο στοχευμένες τεχνικές και εξειδικευμένοι αλγόριθμοι μπορούν να χρησιμοποιηθούν. Η παρούσα εργασία περιορίστηκε στην εξαγωγή χαρακτηριστικών από τα ίδια τα άρθρα. Μια συνέχισή της θα μπορούσε να χρησιμοποιεί πληροφορίες που προέρχονται από κοινωνικούς ιστούς όπως τον αριθμό των θεάσεων μιας είδησης, τα σχόλια των χρηστών, τις κοινοποιήσεις της κ.ά. για την παράλληλη εξαγωγή στατιστικών πορισμάτων. Επίσης, θα μπορούσαν να αξιοποιηθούν και τα ήδη υπάρχοντα χαρακτηριστικά που σε αυτή την ανάλυση παραλήφθηκαν π.χ. το θέμα, την ημερομηνία έκδοσης του άρθρου κτλ.

Επιπλέον, μια μελλοντική βελτίωση θα μπορούσε να αφορά στην αλλαγή του τρόπου κατασκευής της εφαρμογής με την χρήση *serverless* αρχιτεκτονικής. Με τον

όρο *Serverless* αρχιτεκτονική περιγράφεται ένα PoC (*Proof-of-concept*[14]) σύστημα το οποίο παραπέμπει στην έλλειψη εξυπηρετητών. Ωστόσο, στην πραγματικότητα είναι μια αρχιτεκτονική που επιτρέπει στην εστίαση στην λειτουργικότητα της εφαρμογής παρά στην διαχείριση υποδομής (π.χ. εγκατάσταση και παραμετροποίηση εξυπηρετητών κτλ). Η ιδέα είναι ότι δεν ακολουθείται το μονολιθικό μοντέλο όπου ένα μεγάλο κομμάτι κώδικα περιέχει όλες τις πληροφορίες για την δόμηση της εφαρμογής αλλά με την ύπαρξη πολλών μικρών υπηρεσιών (*services-functions)* που επιτελούν ξεχωριστές διεργασίες. Ουσιαστικά, η αρχιτεκτονική αυτή αποτελεί μετεξέλιξη των microservices όπου η εφαρμογή έσπαγε σε πολλά επιμέρους κατανεμημένα κομμάτια. ένα τέτοιο σύστημα θα μπορούσε να επιτευχθεί με την πλατφόρμα OpenFaaS όπου κάθε function περικλείεται μέσα σε ένα Docker container. Επίσης υπάρχουν κατάλληλα εργαλεία για την ενορχήστρωση του *Docker swarm* και ένα API που καλεί το ανάλογο function κάθε φορά μέσω *HTTP* πρωτοκόλλου[15].

Τέλος, η χρήση των Docker containers ανεξαρτητοποιεί την εφαρμογή όπως έχει αναφερθεί και παραπάνω. Για τον λόγο αυτό, η πλατφόρμα μπορεί να αξιοποιηθεί και σε άλλες περιπτώσεις μηχανικής μάθησης όπως για παράδειγμα η αναγνώριση εικόνων ή και εφαρμογές με εντελώς διαφορετικό αντικείμενο.

[14] Η απόδειξη της έννοιας, γνωστή και ως απόδειξη της αρχής, είναι η πραγματοποίηση μιας συγκεκριμένης μεθόδου ή ιδέας προκειμένου να αποδειχθεί η σκοπιμότητά της ή μια επίδειξη κατ 'αρχήν με σκοπό να επαληθευτεί ότι κάποια έννοια ή θεωρία έχει πρακτικό δυναμικό.

[15] Το Πρωτόκολλο Μεταφοράς Υπερκειμένου είναι ένα πρωτόκολλο επικοινωνίας. Αποτελεί το κύριο πρωτόκολλο που χρησιμοποιείται στους φυλλομετρητές του Παγκοσμίου Ιστού για να μεταφέρει δεδομένα ανάμεσα σε έναν διακομιστή και έναν πελάτη.

ΒΙΒΛΙΟΓΡΑΦΙΚΕΣ ΑΝΑΦΟΡΕΣ

[1] S. Maheshwari, How fake news goes viral: A case study, Nov. 2016. [Online]. Available: https://www.nytimes. com / 2016 / 11 / 20 / business / media / how - fake - news - spreads.html (visited on 11/08/2017).
[6] E. Tacchini, G. Ballarin, M. L. Della Vedova, S. Moret, and L. de Alfaro, "Some Like it Hoax: Automated Fake News Detection in Social Networks," in Proceedings of the Second Workshop on Data Science for Social Good, vol. 1960. Skopje, Macedonia: CEUR-WS, 2017
[7]F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, Recommender Systems Handbook. Springer, 2015.
[8]Rubin, Victoria & Conroy, Niall & Chen, Yimin & Cornwell, Sarah. (2016). Fake News or Truth? Using Satirical Cues to Detect Potentially Misleading News. . 10.18653/v1/W16-0802.
[9]Allcott, Hunt & Gentzkow, Matthew. (2017). Social Media and Fake News in the 2016 Election. Journal of Economic Perspectives. 31. 211-236. 10.1257/jep.31.2.211.
[10]Kuiper, Els, and Monique Volman. "The Web as a source of information for students in K–12 education." *Handbook of research on new literacies* 5 (2008): 241-266.
[11]Domonoske, Camila. "Students have 'dismaying'inability to tell fake news from real, study finds." *National Public Radio* 23 (2016).
[12]Chowdhary, K. R. "Natural language processing." *Fundamentals of Artificial Intelligence*. Springer, New Delhi, 2020. 603-649.
[13]Kotsiantis, Sotiris B., I. Zaharakis, and P. Pintelas. "Supervised machine learning: A review of classification [2] Justin Mason. Filtering spam with spamassassin. In HEANet Annual Conference, page 103, 2002.
[3]Blanzieri, E. and A. Bryl, A survey of learning-based techniques of email spam filtering. Artif. Intell. Rev., 2008. 29(1): p. 63-92.
[4]Kaliyar, Rohit Kumar, et al. "FNDNet–A deep convolutional neural network for fake news detection." *Cognitive Systems Research* 61 (2020): 32-44.
[5]Della Vedova, Marco L., et al. "Automatic online fake news detection combining content and social signals." *2018 22nd Conference of Open Innovations Association (FRUCT)*. IEEE, 2018.
techniques." *Emerging artificial intelligence applications in computer engineering* 160.1 (2007): 3-24.
[14]Nadeau, David, and Satoshi Sekine. "A survey of named entity recognition and classification." Lingvisticae Investigationes 30.1 (2007): 3-26.
[15]Schmitt, Xavier, et al. "A Replicable Comparison Study of NER Software: StanfordNLP, NLTK, OpenNLP, SpaCy, Gate." *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)*. IEEE, 2019.
[16]Givens, Geof H., and Jennifer A. Hoeting. *Computational statistics*. Vol. 703. John Wiley & Sons, 2012.
[17]Wernick, Yang, Brankov, Yourganov and Strother, Machine Learning in Medical Imaging, *IEEE Signal Processing Magazine*, vol. 27, no. 4, July 2010, pp. 25-38.
[18]Guzella, Thiago S., and Walmir M. Caminhas. "A review of machine learning approaches to spam filtering." *Expert Systems with Applications* 36.7 (2009): 10206-10222.
[19]Sehral, A. K. (2008). Text Mining: The search for novelty in text. A report submitted in partial fulfillment of the requirements of the Ph.D Comprehensive Examination in the Department of Computer Science
[20]Nahm, U. Y & Raymond J. M. (2002). Text Mining with Information Extraction. Technical Report SS-02-06. Department of Computer Sciences, University of Texas.
[21]Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. Introduction to Information Retrieval. Cambridge University Press, New York, NY, USA.
[22]Fidelis, Marcos Vinicius, Heitor S. Lopes, and Alex A. Freitas. "Discovering comprehensible classification rules with a genetic algorithm." *Proceedings of the 2000 Congress on Evolutionary Computation. CEC00 (Cat. No. 00TH8512)*. Vol. 1. IEEE, 2000.
[23]Belgiu, M., & Drăguţ, L. (2016). Random forest in remote sensing: A review of applications and future directions. *ISPRS journal of photogrammetry and remote sensing*, *114*, 24-31.
[24] Hanley, J. A., Negassa, A., Edwardes, M. D. D., & Forrester, J. E. (2003). Statistical analysis of correlated data using generalized estimating equations: an orientation. *American journal of epidemiology*, *157*(4), 364-375.
[25]Hutto, C., & Gilbert, E. (2014, May). Vader: A parsimonious rule-based model for sentiment analysis of social media text. In *Proceedings of the International AAAI Conference on Web and Social Media* (Vol. 8, No. 1).

[26] Bird, S., Klein, E., & Loper, E. (2009). *Natural language processing with Python: analyzing text with the natural language toolkit*. " O'Reilly Media, Inc.".
[27] Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J. R., Bethard, S., & McClosky, D. (2014, June). The Stanford CoreNLP natural language processing toolkit. In *Proceedings of 52nd annual meeting of the association for computational linguistics: system demonstrations* (pp. 55-60).
[28] Schmitt, X., Kubler, S., Robert, J., Papadakis, M., & LeTraon, Y. (2019, October). A replicable comparison study of NER software: StanfordNLP, NLTK, OpenNLP, SpaCy, Gate. In *2019 Sixth International Conference on Social Networks Analysis, Management and Security (SNAMS)* (pp. 338-343). IEEE.
[29]Huang, D., Ma, X., & Zhang, S. (2019). Performance analysis of the raft consensus algorithm for private blockchains. *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, *50*(1), 172-181.