



ΠΑΝΕΠΙΣΤΗΜΙΟ ΘΕΣΣΑΛΙΑΣ

Προγραμματιστική εργασία με θέμα:

'Sentiment Analysis of Covid-19 Tweets'

COVID-19
Coronavirus

Εξόρυξη Δεδομένων Μεγάλου Όγκου
Χειμερινό Εξάμηνο 2021-2022

Παναγιώτου Δήμητρα_2595
Καραμούστου Βασιλική_2424



Sentiment Analysis of Covid-19 Tweets

Καραμούστου Βασιλική και Παναγιώτου Δήμητρα

Εξόρυξη Δεδομένων 2021-22

Τμήμα Ηλεκτρολόγων Μηχανικών & Μηχανικών Υπολογιστών

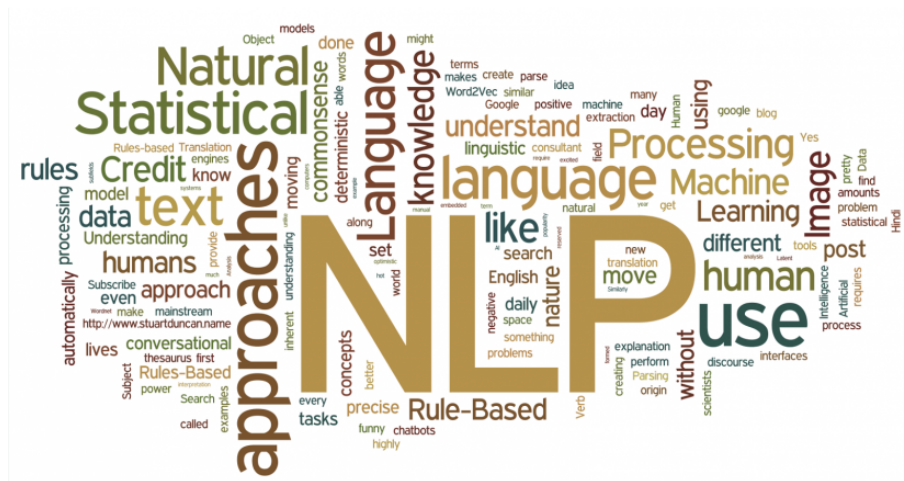
Πανεπιστήμιο Θεσσαλίας, Βόλος

{vkaramoustou,dimpanagiotou}@e-ce.uth.gr

Περίληψη Στην παρακάτω αναφορά παρουσιάζεται μια υλοποίηση Natural Language Processing πάνω σε δεδομένα αναφορικά με την πανδημία COVID-19. Για την παραπάνω υλοποίηση χρησιμοποιήθηκαν τεχνικές προεπεξεργασίας δεδομένων, αλγόριθμοι Μηχανικής Μάθησης και τεχνικές Deep Learning, έτσι ώστε να επιτευχθούν ακριβή μοντέλα πρόβλεψης. Το πρόβλημα που λύνει η υλοποίηση μας, είναι ένα πρόβλημα ταξινόμησης (Θετικών ή Αρνητικών σχολίων).

1 Natural Language Processing

Η **επεξεργασία φυσικής γλώσσας (NLP)** αναφέρεται στον κλάδο της επιστήμης των υπολογιστών - και πιο συγκεκριμένα, στον κλάδο της τεχνητής νοημοσύνης που ασχολείται με την ικανότητα των υπολογιστών να κατανοούν κείμενο και εκφράσεις του προφορικού λόγου με τον ίδιο σχεδόν τρόπο που μπορούν οι άνθρωποι.



Το NLP συνδυάζει την υπολογιστική γλωσσική μοντελοποίηση της ανθρώπινης γλώσσας βασισμένη σε κανόνες και μοντέλα στατιστικής, μηχανικής μάθησης και βαθιάς

μάθησης. Αυτές οι τεχνολογίες επιτρέπουν στους υπολογιστές να επεξεργάζονται την ανθρώπινη γλώσσα, που προέρχεται είτε από κείμενα είτε από φωνητικά δεδομένα, και να «κατανοούν» το πλήρες νόημά της, με την πρόθεση και το συναίσθημα του ομιλητή ή του συγγραφέα.

Η επεξεργασία φυσικής γλώσσας συναντάται σε προγράμματα υπολογιστών που μεταφράζουν κείμενο από μια γλώσσα σε μία άλλη, ανταποκρίνονται σε προφορικές εντολές και συνοψίζουν μεγάλους όγκους κειμένου γρήγορα. Υπάρχει μια καλή πιθανότητα να έχετε αλληλεπιδράσει με τεχνικές NLP εν έχετε χρησιμοποιήσει συστήματα GPS που λειτουργούν με φωνή, ψηφιακούς βοηθούς, λογισμικό υπαγόρευσης ομιλίας σε κείμενο, chatbot εξυπηρέτησης πελατών και άλλες ανέσεις για τους καταναλωτές. Παράλληλα οι τεχνικές NLP υποβοηθούν και υποστηρίζουν σε μεγάλο βαθμό τις επιχειρηματικές αναλύσεις που βοηθούν στον εξορθολογισμό των επιχειρηματικών λειτουργιών, και οδηγούν στην αύξηση της παραγωγικότητας των εργαζομένων και στην απλοποίηση των κρίσιμων για την επιχειρηματικών διαδικασιών.

Η **ανάλυση συναισθήματος**, γνωστή και ως sentiment analysis, είναι μια υλοποίηση της επεξεργασία φυσικής γλώσσας (NLP) και χρησιμοποιείται για να προσδιορίσει τον συναισθηματικό τόνο πίσω από ένα σώμα κειμένου. Αυτός είναι ένας αρκετά δημοφιλής τρόπος για τους οργανισμούς να προσδιορίζουν και να κατηγοριοποιούν απόψεις σχετικά με ένα προϊόν, μια υπηρεσία ή μια ιδέα. Περιλαμβάνει τεχνικές εξόρυξης δεδομένων, μηχανικής μάθησης (ML) κα τεχνητής νοημοσύνης (AI) για την εξόρυξη κειμένου για συναίσθημα και υποκειμενικές πληροφορίες.

Τα συστήματα ανάλυσης συναισθήματος βοηθούν τους οργανισμούς να συλλέγουν πληροφορίες από μη οργανωμένο και μη δομημένο κείμενο που προέρχεται από διαδικτυακές πηγές, όπως μηνύματα ηλεκτρονικού ταχυδρομείου, αναρτήσεις ιστολογίου, εισιτήρια υποστήριξης, συνομιλίες ιστού, κανάλια κοινωνικών μέσων, φόρουμ και σχόλια. Οι αλγόριθμοι αντικαθιστούν τη χειροκίνητη επεξεργασία δεδομένων με την εφαρμογή μεθόδων που βασίζονται σε κανόνες. Τα συστήματα αυτά βασίζονται σε κανόνες που εκτελούν ανάλυση συναισθήματος με βάση προκαθορισμένους κανόνες (όπως ένα λεξικό), ενώ άλλα συστήματα έχουν αυτοματοποίηση αυτή την διαδικασία και μαθαίνουν κατευθείαν από τα δεδομένα με τεχνικές μηχανικής εκμάθησης.

2 Τα δεδομένα μας

Για τους σκοπούς της εργασίας χρησιμοποιήθηκε ένα σύνολο δεδομένων από το twitter. Το συγκεκριμένο σύνολο δεδομένων περιέχει 44,955 εγγραφές οι οποίες πρακτικά είναι 44,955 διαφορετικά σχόλια χρηστών του twitter τα οποία διακρίνονται σε 'Πολύ Αρνητικά', 'Αρνητικά', 'Ουδέτερα', 'Θετικά', 'Πολύ Θετικά'. Στα δεδομένα συμπεριλαμβάνονται και άλλες πληροφορίες σχετικά με τους χρήστες όπως το όνομα χρήστη, το διακριτικό της οθόνης, την ημερομηνία που πραγματοποιήθηκε το σχόλιο καθώς και την τοποθεσία του χρήστη. Η πλειοψηφία των σχολίων στο συγκεκριμένο σύνολο δεδομένων αφορούν θετικά επηρεασμένα σχόλια.

Για την μελέτη των δεδομένων πραγματοποιήθηκε αρχικά μια ανάλυση για να εντοπισθούν μοτίβα και λέξεις που επαναλαμβάνονται σε σχόλια θετικού ή αρνητικού περιεχομένου.

3 Προεπεξεργασία Δεδομένων

Πριν τα δεδομένα τροφοδοτηθούν στα μοντέλα εκπαίδευσης, υπόκεινται σε διαδικασίες προεπεξεργασίας έτσι ώστε το κείμενο να μετατραπεί σε γλώσσα κατανοητή τόσο για τον υπολογιστή όσο και για τους αλγορίθμους εκπαίδευσης. Τα βήματα της

προεπεξεργασίας είναι αρκετά απλά και παρουσιάζονται παρακάτω. Πριν προχωρήσουμε με τεχνικές προ-επεξεργασίας συγχωνεύσαμε τα 'Θετικά' και 'Πολύ Θετικά' σχόλια καθώς και τα 'Αρνητικά' και 'Πολύ Αρνητικά' σχόλια ενώ εξαίρεσαμε από την μλέτη μας σχόλια που είχαν αξιολογηθεί και ταξινομηθεί ως 'Ουδέτερα'.

3.1 Αφαίρεση σημείων στίξης

Για να επιτευχθεί η αφαίρεση των σημείων στίξης, χρησιμοποιείται η `sub()` (της python βιβλιοθήκης RE). Μπορεί να αντικαταστήσει τα σημεία στίξης των συγκεκριμένων χαρακτήρων, σε αυτήν την περίπτωση με κάποιο άλλο χαρακτήρα ή να τα αφαιρέσει τελείως. Ο κώδικας αντικαθιστά τα σημεία στίξης με κενά (" ").

3.2 Μετατροπή κεφαλαίων σε μικρά

Υπάρχει μια κοινή προσέγγιση για τη μετατροπή των γραμμάτων από κεφαλαία σε μικρά για λόγους απλότητας. Βοηθά στη διατήρηση της ροής συνέπειας κατά τη διάρκεια των εργασιών NLP και της εξόρυξης κειμένου. Η συνάρτηση `low()` κάνει την όλη διαδικασία αρκετά απλή.

3.3 Αφαίρεση tags και URLs

Για να εκτελέσουμε αντιστοιχίσεις με κανονικές εκφράσεις, χρησιμοποιήθηκε η `re.compile()` για να μετατρέψει τα tags και URLs σε αντικείμενα, ώστε η αναζήτηση μοτίβων να γίνει ευκολότερη κατά την εκπαίδευση των μοντέλων.

3.4 NLTK Techniques

Στην συνέχεια χρησιμοποιήθηκαν τεχνικές NLTK για περαιτέρω επεξεργασία των δεδομένων. Το NLTK σημαίνει Natural Language Toolkit. Είναι ένα ισχυρό εργαλείο, πλήρες με διαφορετικές ενότητες και βιβλιοθήκες Python για την εκτέλεση απλής έως πολύπλοκης επεξεργασίας φυσικής γλώσσας (NLP). Αυτές οι βιβλιοθήκες NLP λειτουργούν ως μεταφραστές μεταξύ μηχανών (όπως το Alexa, το Siri ή το Google Assistant) και των ανθρώπων, έτσι ώστε τα μηχανήματα να έχουν την κατάλληλη απόκριση. Το NLTK έχει ένα μεγάλο, δομημένο κείμενο γνωστό ως `corpus` που περιέχει αρχεία κειμένου αναγνώσιμα από μηχανή σε έναν κατάλογο που δημιουργήθηκε για εργασίες NLP.

Αφαίρεση των stopwords Τα stopwords είναι οι λέξεις που χρησιμοποιούνται συνήθως και αφαιρούνται από το κείμενο καθώς δεν προσθέτουν αξία στην ανάλυση. Αυτές οι λέξεις έχουν λιγότερο ή καθόλου νόημα. Η βιβλιοθήκη NLTK αποτελείται από μια λίστα λέξεων που θεωρούνται stopwords για την αγγλική γλώσσα. Μερικά από αυτά είναι: [i, me, my, myself, we, our, ours, ourselves, you, you're, you've, you'll, you'd, your, yours, yourself, yourselves, he, most, other, some, such, no, nor, not, only, own, same, so, then, too, very, s, t, can, will, just, don, don't, should, should've, now, d, ll, m, o, re, ve, y, ain, aren't, could, couldn't, didn't, didn't]

Δεν είναι απαραίτητο να χρησιμοποιήθε σε όλες τις περιπτώσεις το παραπάνω λεξικό με stopwords, γιατί κάθε μοντέλο είναι διαφορετικό. Για παράδειγμα, η λέξη «How» μπορεί να είναι stopwords για ένα μοντέλο, αλλά για κάποιο άλλο πρόβλημα μπορεί να είναι σημαντική. Έχει λοιπόν νόημα να δημιουργήσουμε μια προσαρμοσμένη λίστα με λέξεις τερματισμού για διαφορετικά προβλήματα.

Αποκοπή καταλήξεων Επόμενο βήμα στην επεξεργασία του κειμένου είναι να επανέλθουν οι λέξεις στη ρίζα/βασική τους μορφή. Για παράδειγμα, λέξεις όπως "προγραμματιστής", "προγραμματισμός", "πρόγραμμα" προέρχονται από την λέξη "πρόγραμμα".

Λημματοποίηση Η λημματοποίηση μετασχηματίζει τη λέξη στην αρχική της ρίζα και παράλληλα φροντίζει να μην χάσει η λέξη το νόημά της. Η λημματοποίηση έχει ένα προκαθορισμένο λεξικό που με το οποίο ελέγχει την ύπαρξη της ρίζας στο λεξικό μετά τον μετασχηματισμό.

Tokenization Το tokenizing πρακτικά χωρίζει μια ολόκληρη πρόταση σε λέξεις. Αντί αυτής της μεθόδου, μπορεί να χρησιμοποιηθεί ένα απλό διαχωριστικό για αυτό το σκοπό. Αλλά ένα διαχωριστικό θα αποτύχει να χωρίσει τις συντομογραφίες που χωρίζονται με "." ή ειδικούς χαρακτήρες, όπως το H.D.M.I., για παράδειγμα. Τα περισσότερα από αυτά τα προβλήματα μπορούν να λυθούν χρησιμοποιώντας τη βιβλιοθήκη nltk. Η ενότητα wordtokenize χωρίζει τις λέξεις σε διακριτικά και αυτές οι λέξεις λειτουργούν ως είσοδος για τη διαδικασία κανονικοποίησης και καθαρισμού. Ακόμα η μέθοδος αυτή μπορεί να χρησιμοποιηθεί για τη μετατροπή μιας συμβολοσειράς (κείμενο) σε αριθμητικά δεδομένα, έτσι ώστε τα μοντέλα μηχανικής εκμάθησης να μπορούν να την αφομοιώσουν.

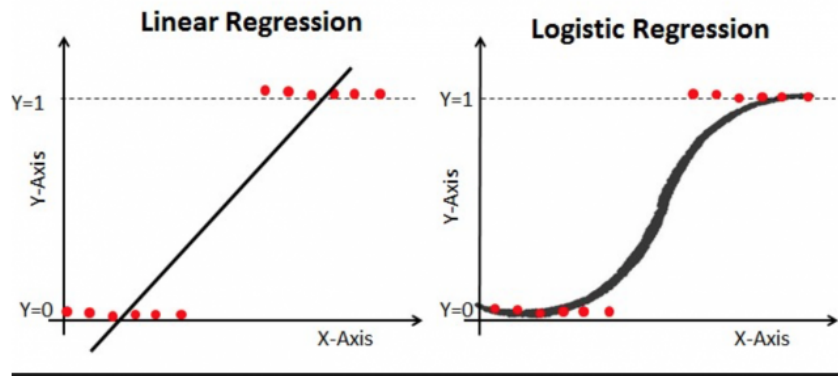
4 Μοντέλα Εκπαίδευσης

ο πρόβλημα της αξιολόγησης των σχολίων σε θετικά ή αρνητικά ανήκει στην κατηγορία των προβλημάτων ταξινόμησης, με την μεταβλητή που θέλουμε να προλέψουμε να παίρνει τιμές 0(αρνητικό) και 1(θετικό). Για τον σκοπό αυτό χρησιμοποιήθηκαν οι παρακάτω αλγόριθμοι:

4.1 Logistic Regression

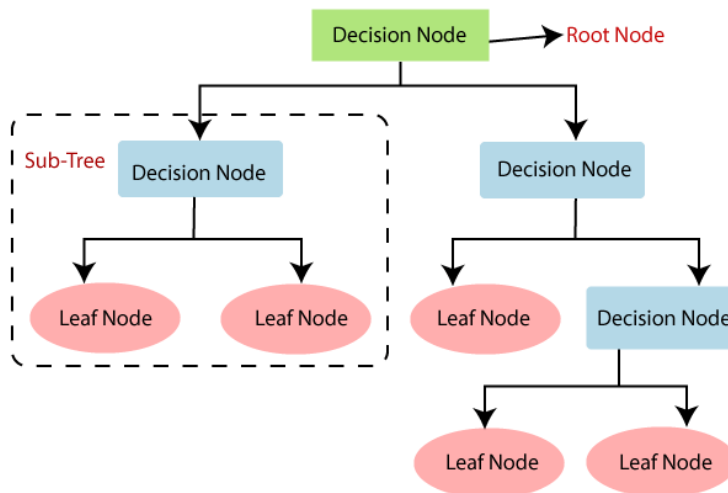
Ο αλγόριθμος Logistic Regression μοντελοποιεί την πιθανότητα ενός διακριτού αποτελέσματος δεδομένης μιας μεταβλητής εισόδου. Τα πιο κοινά μοντέλα Logistic Regression διαχειρίζονται ένα πρόβλημα δυαδικής ταξινόμησης, κάτι που μπορεί να πάρει δύο τιμές όπως true/false, yes/no, και ούτω καθεξής. Γενικά ο αλγόριθμος Logistic Regression αποτελεί μία χρήσιμη μέθοδος ανάλυσης για προβλήματα ταξινόμησης, όπου στόχος είναι να προσδιοριστεί εάν ένα νέο δείγμα και να ενταχθεί στην κατηγορία που ταιριάζει καλύτερα.

Ο καλύτερος τρόπος να σκεφτούμε το Logistic Regression είναι ότι πρόκειται για γραμμική παλινδρόμηση (Linear Regression) αλλά για προβλήματα ταξινόμησης. Η Logistic Regression χρησιμοποιεί ουσιαστικά τη λογιστική συνάρτηση που ορίζεται παρακάτω



για να μοντελοποιήσει μια δυαδική μεταβλητή εξόδου. Η κύρια διαφορά μεταξύ Linear Regression και Logistic Regression είναι ότι το εύρος του Logistic Regression οριοθετείται μεταξύ 0 και 1. Επιπλέον, σε αντίθεση με την Linear Regression, το Logistic Regression δεν απαιτεί γραμμική σχέση μεταξύ των μεταβλητών εισόδου και εξόδου.

4.2 Decision Tree



Το Decision Tree είναι μια supervised learning που μπορεί να χρησιμοποιηθεί τόσο για προβλήματα classification όσο και για προβλήματα regression, αλλά κυρίως προτιμάται για την επίλυση προβλημάτων classification. Είναι ένας ταξινομητής με δομή

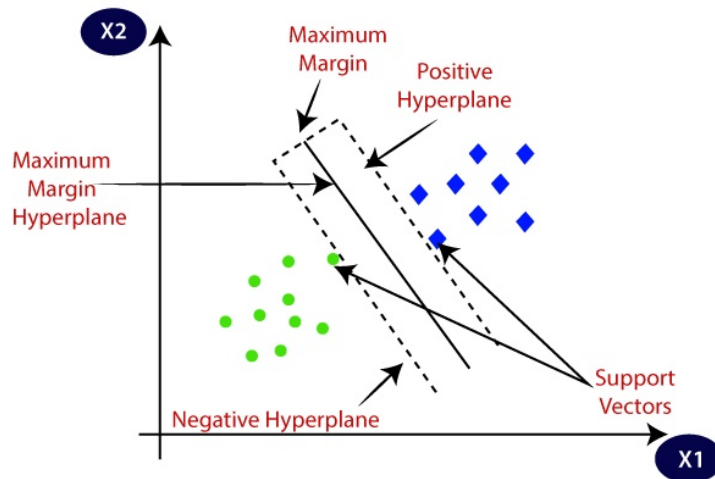
δέντρου, όπου οι εσωτερικοί κόμβοι αντιπροσωπεύουν τα χαρακτηριστικά ενός συνόλου δεδομένων, οι κλάδοι αντιπροσωπεύουν τους κανόνες απόφασης και κάθε κόμβος φύλλου αντιπροσωπεύει το αποτέλεσμα.

Κατά την εφαρμογή ενός δέντρου απόφασης, προκύπτει το κύριο ζήτημα τους πώς να επιλέξει κάποιος το καλύτερο χαρακτηριστικό για τον ρίζα και για τους υποκόμβους. Έτσι, για την επίλυση τέτοιων προβλημάτων υπάρχει μια τεχνική που ονομάζεται μέτρο επιλογής χαρακτηριστικών ή ASM. Με αυτή τη μέτρηση, μπορούμε εύκολα να επιλέξουμε το καλύτερο χαρακτηριστικό για τους κόμβους του δέντρου. Στο δικό μας μοντέλο χρησιμοποιήθηκε το **Gini Index** το οποίο μετρά τον βαθμό ή την πιθανότητα μίας συγκεκριμένης μεταβλητής να ταξινομηθεί εσφαλμένα όταν επιλέγεται τυχαία. Υπολογίζεται αφαιρώντας το άθροισμα των τετραγώνων πιθανοτήτων κάθε τάξης ξεκινώντας από την πρώτη.

$$\text{Gini Index} = 1 - \sum (P(x = k))^2$$

4.3 Support Vector Machines

Ο Support Vector Machine ή SVM είναι ένας από τους πιο δημοφιλείς αλγόριθμους supervised learning, ο οποίος χρησιμοποιείται για προβλήματα classification και regression. Ωστόσο, κατά κύριο λόγο, χρησιμοποιείται για προβλήματα classification στη Μηχανική Μάθηση. Ο στόχος του αλγόριθμου SVM είναι να δημιουργήσει την κα-



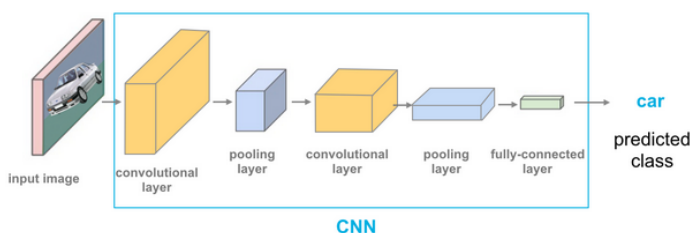
λύτερη γραμμή ή όριο απόφασης που μπορεί να διαχωρίσει το χώρο n-διαστάσεων σε κλάσεις, ώστε να μπορούμε εύκολα να βάλουμε το νέο σημείο δεδομένων στη σωστή κατηγορία. Αυτό το όριο καλύτερης απόφασης ονομάζεται υπερεπίπεδο(hyperplane).

Το SVM επιλέγει τα ακραία σημεία/διανύσματα που βοηθούν στη δημιουργία του hyperplane. Αυτές οι ακραίες περιπτώσεις ονομάζονται support vectors και ως εκ τούτου ο αλγόριθμος ονομάζεται Support Vector Machine.

4.4 Convolutional Neural Network

Στο Deep Learning, το Convolutional Neural Network (CNN) είναι ένας τύπος Τεχνητού Νευρωνικού Δικτύου. Το CNN αποτελείται από μία ακολουθία επιπέδων και κάθε στρώμα μετατρέπει μία πληθώρα activations σε μία άλλη μέσω μιας διαφοροποιησιμης συνάρτησης. Χρησιμοποιούμε τρεις κύριους τύπους επιπέδων για τη δημιουργία αρχιτεκτονικών CNN: Συνελκτικό Επίπεδο (Convolutional Layer), Επίπεδο συγκέντρωσης (Pooling Layer) και Πλήρως Συνδεδεμένο Επίπεδο (Fully-Connected Layer). Κάθε στρώμα αποτελείται από ένα συγκεκριμένο σύνολο νευρώνων, σε κάθε στρώμα οι νευρώνες που υπάρχουν συνδέονται μεταξύ τους.

Πιο συγκεκριμένα το συνελκτικό επίπεδο φιλτράρει την είσοδο και οδηγεί σε ένα activation. Η επαναλαμβανόμενη εφαρμογή του ίδιου φίλτρου σε μια είσοδο έχει ως αποτέλεσμα έναν χάρτη activation που ονομάζεται feature map, υποδεικνύοντας τις τοποθεσίες και την ισχύ ενός χαρακτηριστικού εισόδου. Το pooling layer είναι υπεύθυνο για την λήψη δειγμάτων από τα feature maps με το να συνοψίζουν την πληροφορία των χαρακτηριστικών με την συνένωση των feature maps. Τέλος το fully connected layer αποτελεί ένα απλό feed forward νευρωνικό υπεύθυνο να συνδέει τα επίπεδα των δύο προηγούμενων επιπέδων.



Συγκεκριμένα για το νευρωνικό μας χρησιμοποιήσαμε τον tokenizer από την βιβλιοθήκη του keras, ο οποίος μετέτρεψε την κάθε εγγραφή-σχόλιο σε ακολουθία αριθμών. Επειδή κάθε σχόλιο είχε διαφορετικό πλήθος λέξεων χρησιμοποιήσαμε την μέθοδο padding έτσι ώστε όλες οι παραπάνω ακολουθίες, που δημιουργήσαμε με τον tokenizer, να έχουν το ίδιο μέγεθος. Τέλος χρησιμοποιήσαμε ένα Glove embedding pre-trained λεξικό σαν lookup για να δημιουργήσουμε τον πίνακα των χαρακτηριστικών μας ο οποίος τελικά δόθηκε σαν είσοδος στο νευρωνικό που περιγράψαμε παραπάνω. Το νευρωνικό χρησιμοποιεί δύο επίπεδα συνέλιξης (filters = 16, kernel size = 3) και σαν συνάρτηση ενεργοποίησης relu. Ακόμα χρησιμοποιεί max pooling, dropout και τελικά ένα hidden layer 250 μονάδων-νευρώνων και ένα επίπεδο εξόδου με την συνάρτηση ενεργοποίησης την sigmoid, δεδομένου ότι αντιμετωπίζουμε ένα πρόβλημα ταξινόμησης.

5 Αποτελέσματα

Για τους αλγορίθμους που παρουσιάστηκαν στην προηγούμενη ενότητα πραγματοποιήθηκε έλεγχος ακρίβειας για το σύνολο δεδομένων της εργασίας και αξιολογήθηκαν

οι παρακάτω μετρικές:

Accuracy: Η ακρίβεια μετρά απλώς πόσο συχνά ο ταξινομητής προβλέπει σωστά. Μπορούμε να ορίσουμε την ακρίβεια ως την αναλογία του αριθμού των σωστών προβλέψεων και του συνολικού αριθμού προβλέψεων.

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

Precision: Το precision εξηγεί πόσες από τις σωστά προβλεπόμενες περιπτώσεις ήταν στην πραγματικότητα θετικές. Μπορούμε να ορίσουμε το precision ως τον αριθμό των αληθινών θετικών διαιρεμένων με τον αριθμό των προβλεπόμενων θετικών.

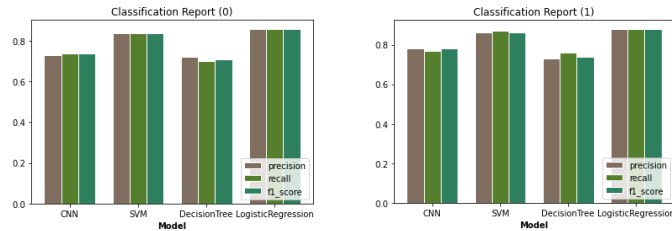
$$\frac{TP}{TP+FP}$$

Recall: Το Recall εξηγεί πόσες από τις πραγματικές θετικές περιπτώσεις μπορέσαμε να προβλέψουμε σωστά με το μοντέλο μας. Μπορούμε να ορίσουμε την ανάκληση ως τον αριθμό των αληθινών θετικών διαιρεμένων με τον συνολικό αριθμό των πραγματικών θετικών.

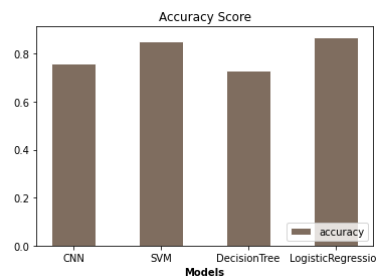
$$\frac{TP}{TP+FN}$$

f1score: Δίνει μια συνδυαστική ιδέα για τις μετρήσεις precision και recall. Παίρνει την μέγιστη τιμή του όταν το precision είναι ίσο με το recall. Το F1 Score είναι το αρμονικό μέσο precision και recall.

Για τα δικά μας δεδομένα μεγαλύτερη ακρίβεια συγκέντρωσε ο αλγόριθμος του logistic regression με ποσοστό 87% και ακολουθεό ο SVM με ποσοστό 85% για το test set. Οι δύο αυτοί αλγόριθμοι προηγούνται και στις μετρικές precision, recall και $f1_score$. *DecisionTree* 73%



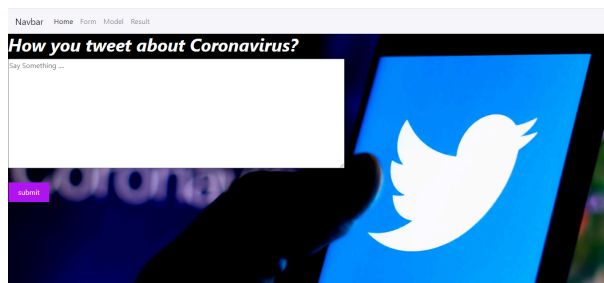
Εικ. 1. 2 Bars from classification report

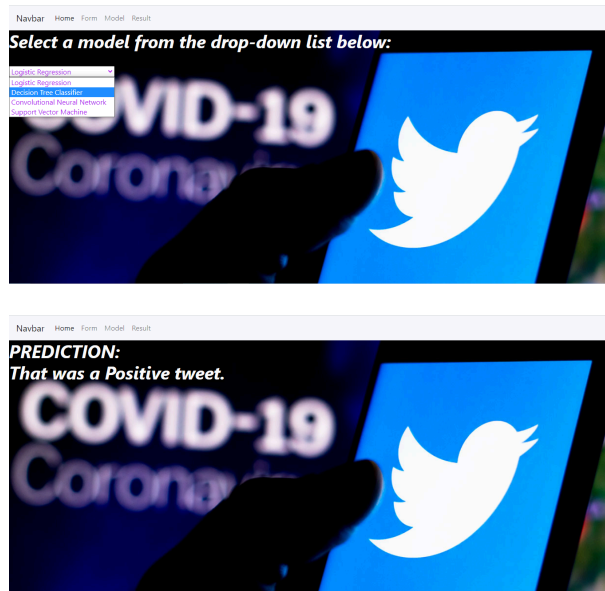


Εικ. 2. Accuracy score

6 User Interface

Τελικά υλοποιήθηκε ένα φιλικό προς το χρήστη interface, όπου ο χρήστης δίνει σαν είσοδο ένα σχόλιο-κείμενο καθώς και το μοντέλο με το οποίο επιλέγει να γίνει η πρόβλεψη. Σαν αποτέλεσμα εμφανίζεται η πρόβλεψη του κάθε επιλεγμένου μοντέλου για τα συγκεκριμένο σχόλιο που δόθηκε σαν είσοδος.





7 Οδηγίες χρήσης και εγκατάστασης της εφαρμογής

1. Κατεβάστε το συμπιεσμένο αρχείο
2. Αποσυμπιέστε το
3. Σε περίπτωση που δεν υπάρχουν τα απαραίτητα requirements για το προγράμμα μας τρέξτε το παρακάτω script για την εγκατάσταση των requirements με την παρακάτω εντολή
pip3 install -r requirements.txt
4. Σε περιβάλλον anaconda ανοίξτε στον φάκελο και εκτελέστε την εντολή
python gui.py
5. Ανακατευθυνθείτε στη διεύθυνση <http://127.0.0.1:5000/> όπου θα ανοίξει το home page της εφαρμογής.
6. Επιλέξτε την καρτέλα "Form" και συμπληρώστε το κενό κελί. Πατήστε "Submit"
7. Αυτομάτως θα ανακατευθυνθείτε στην καρτέλα "Model" όπου και θα επιλέξετε τον αλγόριθμο, από την "drop-down" λίστα, με τον οποίο θέλετε να γίνει η πρόβλεψη και στην συνέχεια πατήστε "Submit".
8. Τέλος, αυτόματα γίνεται θα ανακατευθυνθείτε στην καρτέλα "Result" όπου και θα παρουσιαστεί το αποτέλεσμα της πρόβλεψης.

8 Βιβλιογραφία

<https://towardsdatascience.com/logistic-regression-detailed-overview-46c4da4303bc>

<https://scikit-learn.org/stable/modules/svm.html>

<https://towardsdatascience.com/support-vector-machine-introduction-to-machine-learning-algorithms-934a444fca47>

<https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53>

<https://www.ibm.com/topics/convolutional-neural-networks>

<https://www.analyticsvidhya.com/blog/2021/05/convolutional-neural-networks-cnn/>

<https://scikit-learn.org/stable/modules/tree.html>