Sentiment Analysis of Twitter Data

ΘΕΩΡΙΑ ΑΠΟΦΑΣΕΩΝ

Αλεξίου Σταύρος 1059680

Αφεντάκη Φλωρεντία 1059576

Μηνάς Δημοσθένης 1059602

Φεβρουάριος 2022

Περιεχόμενα

Εισαγωγή

Στα πλαίσια της εξαμηνιαίας εργασίας του μαθήματος *Θεωρίας Αποφάσεων* επιλέξαμε να υλοποιήσουμε την άσκηση με τίτλο *Sentiment Analysis of Twitter Data.* Καθένας από εμάς έφτασε στην λύση της άσκησης αυτής με έναν διαφορετικό τρόπο τον οποίο θα παρουσιάσει στην συνέχεια της παρούσας αναφοράς. Ο τρόπος που εργαστήκαμε ήταν να επιλέξουμε, ο καθένας, και να υλοποιήσουμε από ένα (διαφορετικό) είδος νευρωνικού δικτύου για την επίλυση του ίδιου προβλήματος αλλά και έναν αλγόριθμο πρόβλεψης αποτελέσματος από αυτούς που αναφέρονται ενδεικτικά στην εκφώνηση της άσκησης (*Linear Regression, Logistic Regression, Decision Tree, XBoost Classification, Naive Bayes Classifier*). Η σειρά με την οποία θα γίνει η παρουσίαση των απαντήσεων είναι ενδεικτική και δεν αποτελεί κριτήριο για την αξιολόγηση βέλτιστης ή μη απάντησης.

Περιγραφή του προβλήματος

Σκοπός της παρούσας εργασίας είναι να αναπτυχθεί κατάλληλος κώδικας, ο οποίος θα αναλύει το συναίσθημα ενός χρήστη, βασισμένος στο *tweet* που δημοσίευσε σε μια δεδομένη περίοδο. Το πρόβλημα που κληθήκαμε να επιλύσουμε αφορά στην ανάλυση – πρόβλεψη του συναισθήματος που εκφράζει το κείμενο, *tweet*, του χρήστη χρησιμοποιώντας προχωρημένες τεχνικές εξόρυξης κειμένου με στόχο την ανάλυση του προβλεπόμενου συναισθήματος το οποίο διακρίνεται σε θετικό, αρνητικό και ουδέτερο.

Το σύνολο δεδομένων που μας διατέθηκε απαρτίζεται από 1,6 εκατομμύρια προεπεξεργασμένα *tweets* για τα οποία παρέχεται επίσης η πληροφορία του συναισθήματος που εκφράζουν (0 = αρνητικό και 4 = θετικό), η απαρίθμηση αυτών, η ημερομηνία, ένα πεδίο σημαίας, το όνομα του χρήστη που δημοσίευσε το *tweet* και τέλος το περιεχόμενο του *tweet.* Το σύνολο αυτό στην αρχική του μορφή είναι απόλυτα γραμμικό (περιέχει μόνο 2 τιμές συναισθήματος, *0 = αρνητικό* και *4 = θετικό*), αλλά στην συνέχεια προσπαθήσαμε να το τροποποιήσουμε κατάλληλα, προσθέτοντας μια ακόμα τιμή συναισθήματος, (2 = ουδέτερο) και εφαρμόζοντας αυτή τη φορά τις τεχνικές μας για το νέο σύνολο δεδομένων που προέκυψε (βλ. Επίλυση προβλήματος – Αφεντάκη Φλωρεντία).

Βιβλιογραφικά υπάρχουν κυρίως 2 τεχνικές με τις οποίες προσεγγίζεται η ανάλυση συναισθήματος σε dataset όπως το *Tweeter, IMDB* κλπ. Προσεγγίσεις με βάση το λεξικό (*Lexicon-Based*) και με βάση τη μηχανική μάθηση. Οι *Lexicon-Based* προσεγγίσεις, χρησιμοποιούν αντιστοιχήσεις ανάμεσα στο κείμενο που θέλουν να σχολιάσουν και σε ένα λεξικό με προκαθορισμένη ετικέτα συναισθήματος. Οι προσεγγίσεις που βασίζεται στη Μηχανική Μάθηση χρησιμοποιούν τεχνικές ταξινόμησης του κειμένου. Υπάρχουν κυρίως δύο τύποι τεχνικών μηχανικής μάθησης. Μη – Εποπτευόμενη μάθηση: Στο μοντέλο δεν παρέχεται καμία a priori πληροφορία για τα δεδομένα και την κλάση στην οποία ανήκουν, για αυτό τον λόγο βασίζονται στη μέθοδο της ομαδοποίηση (*clustering*). Εποπτευόμενη μάθηση: Στο μοντέλο παρέχονται οι ετικέτες – στόχοι για τα δεδομένα κατά την διάρκεια της εκπαίδευσης του μοντέλου. Στην εργασία αυτή, ασχοληθήκαμε με τεχνικές επιβλεπόμενες μάθησης και για μερικά από τα αποτελέσματα εφαρμόσαμε, υβριδικό μοντέλο ανάμεσα σε τεχνικές του λεξικού και της Μηχανικής Μάθησης.

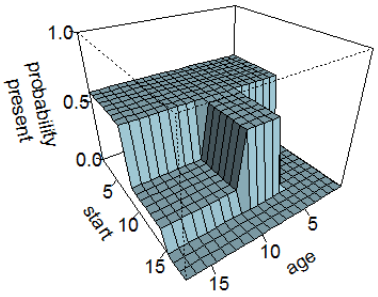
Προεπεξεργασία του συνόλου δεδομένων

Το πρώτο στάδιο για την υλοποίηση του ζητουμένου της εργασίας αυτής, είναι τόσο η ανάγνωση των δεδομένων από το πρόγραμμα μας όσο και η κατάλληλη προεπεξεργασία τους. Η διαδικασία αυτή, περιέχει αφαίρεση των περίσσιων στηλών αλλά και τεχνικές φιλτραρίσματος του κειμένου, δηλαδή τεχνικές με τις οποίες απομακρύνουμε στοιχεία των tweets όπως *emojis, user's mentions, URLs* κ.λπ. Ακόμη, σε αυτό το στάδιο εμπίπτει και ο επιπλέον συναισθηματικός χαρακτηρισμός των δεδομένων μας. Πιο συγκεκριμένα, όπως αναφέρθηκε προηγουμένως, στο δοθέν *dataset* υπάρχουν μόνο δύο, προκαθορισμένοι, τύποι συναισθήματος. Αποφασίσαμε λοιπόν, να δημιουργήσουμε ένα επιπλέον *dataset* στο οποίο, ο συναισθηματικός χαρακτηρισμός των δεδομένων μας, θα γίνεται εκ νέου από την Python, συμπεριλαμβάνοντας με αυτόν τον τρόπο, τον χαρακτηρισμό για το ουδέτερο συναίσθημα (*neutral = 0*). Ορίσαμε έτσι, δύο διαδικασίες την *preprocess()* και την *preprocessSentiment()*, οι οποίες δημιουργούν, και επιστρέφουν τα αντίστοιχα *dataset*. Θα θέλαμε να υπογραμμίσουμε πως για την καλύτερη απόδοση του κώδικα, εξ όσον οι παραπάνω διαδικασίες αποθηκεύουν το επεξεργασμένο *dataset* στην μνήμη, το στάδιο της προεπεξεργασίας πραγματοποιείται αν και μόνο αν διαγράψουμε το επεξεργασμένο αρχείο.

Ένα ακόμα στάδιο που θα θέλαμε να αναφέρουμε, είναι το στάδιο της διανύματοποίησης. Το σύνολο των δεδομένων που πρέπει να επεξεργαστούμε αφορά ένα σύνολο κειμένων. Για τον λόγο αυτό, είναι αναγκαίο η εφαρμογή ενός ακόμα σταδίου επεξεργασίας, αυτού της διανύματοποίησης. Η αντιστοίχιση των tweets με τα κατάλληλα διανύσματα, είναι μία διαδικασία η οποία επηρεάζει την απόδοση του εκάστοτε αλγορίθμου, για τον λόγο αυτό, την διαδικασία αυτήν τη σχολιάζουμε παρακάτω.

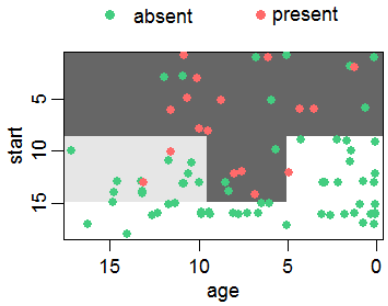
Επίλυση του προβλήματος – Αλεξίου Σταύρος

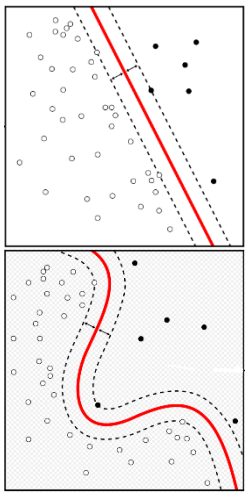
Η προσέγγιση της λύσης του δεδομένου προβλήματος πραγματοποιήθηκε με διάφορους τρόπους. Αρχικά, έγινε προσπάθεια επίλυσης με χρήση ενός αλγόριθμου βαθιάς μάθησης αλλά εκ των υστέρων δημιουργήθηκε και νευρωνικό δίκτυο για την σύγκριση των επιμέρους αποτελεσμάτων.



Decision Tree Classifier

Ο αλγόριθμος Ταξινόμησης Δένδρου Αποφάσεων, Decision Tree Classifier, είναι ένα μοντέλο πρόβλεψης που χρησιμοποιείται για την εξόρυξη δεδομένων και δίνει λύση σε γραμμικά προβλήματα ή αλλιώς προβλήματα στα οποία γίνεται διαμέριση του χώρου σε δυο νέους υποχώρους. Ο τρόπος λειτουργίας του είναι αρκετά απλός σε σχέση με παρόμοιους αλγορίθμους. Στους εσωτερικούς κόμβους αποθηκεύεται η πληροφορία για ένα αντικείμενο (στη συγκεκριμένη περίπτωση είναι τα διανύσματα των *tweets* του *dataset*) ενώ στα *φύλλα* εμπεριέχονται τα 'συμπεράσματα' για την μεταβλητή στόχος, στην περίπτωσή μας είναι το συναίσθημα του αντίστοιχου *tweet*. Με άλλα λόγια στους εσωτερικούς κόμβους αποθηκεύεται πληροφορία για την ένωση (*conjuction*) χαρακτηριστικών των *tweets* τα οποία οδηγούν στη μεταβλητή στόχο που βρίσκεται στα φύλλα. Ο αλγόριθμος Δένδρου Ταξινόμησης βρίσκεται ανάμεσα στους πιο γνωστούς αλγόριθμους μηχανικής μάθησης, δεδομένης της ευφυΐας και της απλότητάς του.





Naive Bayes Classifier

Στη συνέχεια, χρησιμοποιήθηκε ένας ακόμα αλγόριθμος μηχανικής μάθησης για την σύγκριση των αποτελεσμάτων. Ο αλγόριθμος του Bayes, Naive Bayes Classifier, είναι ένας ταξινομητής ο οποίος βασίζεται στην πιθανοτική θεωρία του Bayes ($P(A|B) = \frac{P(B|A)P(A)}{P(B)}$) με ισχυρές υποθέσεις μεταξύ των χαρακτηριστικών τα οποία είναι μεταξύ τους στατιστικά ανεξάρτητα. Είναι ένας αλγόριθμος με εξαιρετικά μεγάλη κλιμάκωση ο οποίος δίνει λύση σε γραμμικά προβλήματα. Σε αυτό το σημείο, επισημαίνεται ότι λόγω της φύσης του προβλήματος ο αλγόριθμος του Bayes απέφερε σημαντικά καλύτερα αποτελέσματα από τον αλγόριθμο του Δένδρου Αποφάσεων.

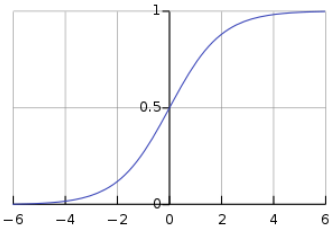
Recurrent Neural Network LSTM - RNNLSTM

Τέλος, υλοποιήθηκε νευρωνικό δίκτυο (Recurrent Neural Network - RNN) με σκοπό την σύγκριση των αποτελεσμάτων, το οποίο είναι βασισμένο στην εφαρμογή των ίδιων βαρών επαναλαμβανόμενα για να δημιουργήσει μια δομημένη πρόβλεψη από τα διανύσματα εισόδου διαπερνώντας τα με τυπολογική σειρά. Τα επίπεδα που χρησιμοποιήθηκαν είναι το *Embedding, CuDNNLSTM, Dropout, Dense* με τα τρία τελευταία να επαναλαμβάνονται δυο φορές. Η σημαντική παρατήρηση εδώ είναι ότι τα αποτελέσματα, όπως ήταν αναμενόμενο, ήταν σημαντικά καλύτερα.

Επίλυση του προβλήματος – Αφεντάκη Φλωρεντία

Στην παρούσα προσέγγιση, χρησιμοποιήσαμε αλγορίθμους επιβλεπόμενης μάθησης όπως η Λογιστική Παλινδρόμηση, ο XGBoost, ο SVM καθώς επίσης και χρήση Ανατροφοδοτούμενου Νευρωνικού Δικτύου (*Recurrent Neural Network – RNN)*. Παρακάτω περιγράφουμε συνοπτικά τις προαναφερθείσες μεθόδους.

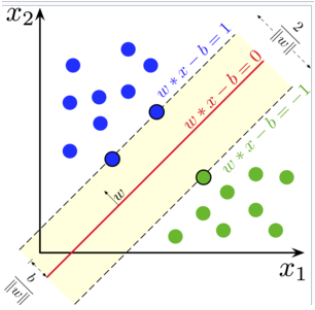
Λογιστική Παλινδρόμηση: Οι προβλέψεις αυτής της τεχνικής μοντελοποιούνται σε γραμμικές εξισώσεις, χρησιμοποιώντας ως συνάρτηση κόστους την μέγιστη πιθανοφάνεια εκτίμησης. Δηλαδή,



$$\ln \frac{P(\omega_i|x)}{P(\omega_M|x)} = w_i^T x + w_{i,0} = 0, για\ i = 1,2, \dots M-1$$

Είναι γνωστή για την γραφική της αναπαράσταση, δηλαδή για την προσαρμογή των δεδομένων μελέτης στην εξίσωση της Σιγμοειδής καμπύλης όπως αυτή παρουσιάζεται στο διπλανό σχήμα. Είναι από τις πιο κλασσικές μεθόδους επιβλεπόμενης μάθησης ενώ μπορεί να χρησιμοποιηθεί για classification και επεξεργασία εικόνας.

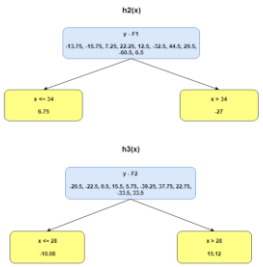
SVM (**S**upport **V**ectors **M**achines): Οι Μηχανές Διανυσματικής Στήριξης χρησιμοποιούν έναν εναλλακτικό σχεδιασμό γραμμικών ταξινομητών οι οποίοι βασίζονται στα υπερεπίπεδα. Πιο συγκεκριμένα στόχος του αλγορίθμου είναι ο σχεδιασμός – εύρεση ενός υπερεπιπέδου το οποίο, να έχει το μέγιστο περιθώριο(*margin*) ανάμεσα στις κλάσεις. Ένα υπερεπίπεδο απόφασης μπορεί να περιγραφεί ως:



$$g(x) = w^T x + w_0 = 0$$

Όπου το *w* το διάνυσμα βαρών το w_0 το threshold ενώ το *x* το σύνολο των σημείων στο υπερεπίπεδο απόφασης. Ένα παράδειγμα γραμμικού SVM με δύο διαχωρίσιμες κλάσεις παρουσιάζεται στο διπλανό σχήμα. Σε προβλήματα που τα δεδομένα το επιβάλουν , μπορούν να χρησιμοποιηθούν και μη – γραμμικά kernel.

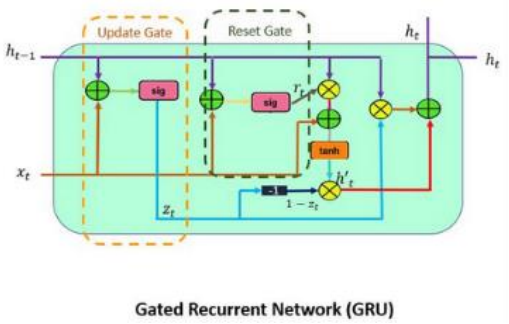
XGBoost (e**X**treme **G**radient **B**oosting): Ο *XGBoost* αλγόριθμος είναι βασισμένος στο συνδυασμό των αλγορίθμων *Decision Tree* και του *Gradient Descent.* Συγκεκριμένα χρησιμοποιεί προσωρινά μικρότερα ανεπτυγμένα δέντρα (*Decision Tree)* στα οποία εφαρμόζει τον αλγόριθμο Gradient Descent για την ελαχιστοποίηση του σφάλματος . Θεωρείται βέλτιστος αφού επιστρέφει καταφέρνει υψηλά αποτελέσματα απόδοσης σε μικρό χρόνο εκπαίδευσης.



RNN (**R**ecurrent **N**eural **N**etwork): Το Ανατροφοδοτούμενο νευρωνικό δίκτυο είναι ένα δίκτυο κόμβων τους λεγόμενους νευρώνες, ο καθένας με μία κατευθυνόμενη (μονόδρομο) ακμή σε κάθε άλλο κόμβο. Στο RNN, το κύριο χαρακτηριστικό του είναι η κρυφή κατάσταση(hidden state), η οποία συμβολίζεται με ht, βάση της οποία υπολογίζεται, μάλιστα, η έξοδος σε κάθε βήμα. Η κρυφή κατάσταση λειτουργεί ως μνήμη στο δίκτυο, η οποία είναι χρήσιμη σε εφαρμογές όπως η ταξινόμηση

της φυσικής γλώσσας, αφού συγκροτεί τη διαδοχική εξάρτηση στις πληροφορίες. Τα επίπεδα που χρησιμοποιήθηκαν είναι το *Embedding, CuDNNGRU, Dropout, CuDNNGRU, Dropout, Dense.*

CudNNGRU (**G**ated **R**ecurrent **U**nit): Το GRU layer αποτελείται από 2 επίπεδα πυλών, την πύλη της ενημέρωσης και την πύλη της λήθης. Η πύλη της ενημέρωσης καθορίζει το ποσό της πληροφορίας που θα χρησιμοποιηθεί ο μοντέλο, από το προηγούμενο επίπεδο ενώ, στον αντίποδα, η πύλη της λήθης το ποσό της πληροφορίας θα που θα «ξεχάσει» ο μοντέλο, από το προηγούμενο επίπεδο. Όπως βλέπουμε και στο σχήμα στα δεξιά, το GRU βασίζεται στο Hadamard product και χρησιμοποιεί τόσο την υπερβολική εφαπτομένη όσο και την σιγμοειδή συνάρτηση. Το CudNNGRU layer δεν είναι τίποτα άλλο, παρά μία εξειδικευμένη υλοποίηση του GRU για να εκτελείται βέλτιστα, αποκλειστικά από την κάρτα γραφικών, τόσο σε απόδοση όσο και σε χρόνο. Οι προδιαγραφές που πρέπει να ακολουθεί το μοντέλο μας μπορούν να βρεθούν εδώ. Τέλος, θεωρείται σκόπιμο, να συγκρίνουμε τις διαφορές ανάμεσα στον GRU και τον LSTM. Ο LSTM χρησιμοποιεί ένα επιπλέον επίπεδο στην έξοδο το οποίο επιφέρει στο μοντέλο περισσότερη πολυπλοκότητα. Όπως πειραματικά ελέγξαμε, ο GRU έχει παρόμοια αποτελέσματα με τον LSTM, απαιτώντας, λιγότερο χρόνο εκπαίδευσης στα ίδια δεδομένα.



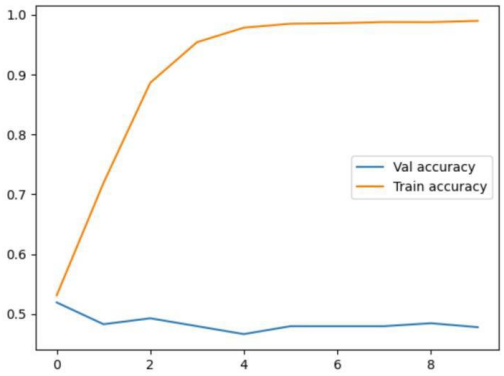
Gated Recurrent Network (GRU)

Ανάλυση συναισθήματος

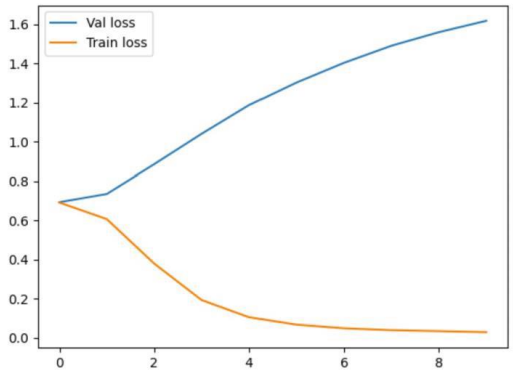
Θέλοντας να δοκιμάσουμε τις παραπάνω τεχνικές συμπεριλαμβάνοντας και το ουδέτερο συναίσθημα (neutral = 0). Αποφασίσαμε να χρησιμοποιήσουμε μία υβριδική προσέγγιση στο πρόβλημα της συναισθηματικής ανάλυσης. Πιο συγκεκριμένα να χρησιμοποιήσουμε επιβλεπόμενους ταξινομητές οι οποίοι χρησιμοποιούν ένα καινούριο σύνολο δεδομένων στο οποίο ο χαρακτηρισμός συναισθήματος γίνεται βασισμένος σε λεξικό. Έτσι για την δημιουργία του καινούργιου dataset χρησιμοποιήσαμε την ανοιχτή βιβλιοθήκη αναλυτή συναισθήματος *VADER (**V**alence **A**ware **D**ictionary and s**E**ntiment **R**easoner)* η οποία είναι rule/lexicon-based. Πρέπει να σημειώσουμε πως τεχνικές όπως εκείνες, που βασίζονται στα λεξικά για τον χαρακτηρισμό του συναισθήματος, εμπίπτουν, συχνά, σε λάθος προβλέψεις, αφού μερικές από τις δυσκολίες που αντιμετωπίζουν είναι στην ανίχνευση σαρκασμού καθώς και σε εξαρτήσεις τομέα (Domain). Εξαρτήσεις δηλαδή, που αφορούν τον διαφορετικό συναισθηματικό χαρακτήρα της ίδιας λέξης σε διαφορετικά θεματικά επίπεδα. Παραδείγματος χάριν, η λέξη απρόβλεπτος όταν θέλουμε να χαρακτηρίσουμε μία ταινία θεωρείται θετικά φορτισμένη, ενώ αν χρησιμοποιήσουμε την ίδια λέξη για να χαρακτηρίσουμε έναν οδηγό αρνητική.

Επίλυση του προβλήματος – Μηνάς Δημοσθένης

Dense/Dropout Network



Έγινε υλοποίηση νευρωνικού δικτύου με χρήση του *Sequential Model* της βιβλιοθήκης *keras* με κυρίαρχα επίπεδα τα *Dense* και *Dropout*. Δοκιμάστηκαν διάφορα μοντέλα κρατώντας πάντα ίδια την συνάρτηση σφάλματος και τον optimizer. Το μεταβλητό μέρος αποτελούσαν τα επίπεδα του νευρωνικού και ο αριθμός των νευρώνων σε κάθε επίπεδο. Από τα αποτελέσματα των διάφορων predictions του νευρωνικού και από την μετρική *accuracy* της βιβλιοθήκης *keras*, που υπολογίζει, πρακτικά, το πόσο συχνά οι προβλέψεις ισούνται με τα *labels* αποδείχθηκε ότι είναι μια καλή μετρική για να αξιολογήσουμε την ποιότητα του νευρωνικού μας δικτύου. Επιπλέον, μια άλλη παράμετρος που θεωρείται μεταβλητή κατά τη διάρκεια των πειραμάτων ήταν η *batch_size*. Η παράμετρος αυτή, ουσιαστικά, χωρίζει την είσοδο σε ισομήκη κομμάτια κατά τη διάρκεια εκτέλεσης του κώδικα και τα τροφοδοτεί μέσα στο πρώτο επίπεδο του νευρωνικού μας δικτύου. Από τα πειράματα εξάχθηκαν τα παρακάτω συμπεράσματα:



1. Όσους περισσότερους νευρώνες περιέχει το δίκτυο στο επίπεδο εισόδου τόσο υψηλότερο *accuracy* πετυχαίνεις. Το κρίσιμο σημείο βρίσκεται στον ορισμό του νευρωνικού δικτύου, καθώς, στην αρχή είναι σημαντικό οι συνδέσεις να αρχικοποιούνται σωστά διότι στην συνέχεια, και λόγω της μετρικής *loss* που έχουμε επιλέξει οι αλλαγές δεν φέρουν δραστική επίδραση στο νευρωνικό μας δίκτυο.
2. Δεν παίζει μεγάλο ρόλο ο αριθμός των κρυφών επιπέδων. Αυτό το συμπέρασμα εξάχθηκε έπειτα από διάφορα πειράματα που πραγματοποιήθηκαν. Αυτό θα μπορούσε να θεωρηθεί ίσως ως αποτέλεσμα της καλής προεπεξεργασίας που είχε γίνει προγενέστερα.
3. Η συνάρτηση ενεργοποίησης είναι η *relu* (*rectified linear activation function*) στο επίπεδο εισόδου και η *sigmoid* στο επίπεδο εξόδου. Οι λόγοι πίσω από τις συγκεκριμένες αποφάσεις είναι ότι η πρώτη έχει την δυνατότητα να μηδενίζει τις αρνητικές τιμές, τις οποίες σπάνια η συνάρτηση διανυματοποίησης που χρησιμοποιήθηκε (*tf-idf)* μπορεί να αποδώσει, γεγονός το οποίο θέλουμε να αποφύγουμε, αφού προσπαθούμε να απαλύνουμε την καμπύλη στις ακραίες περιπτώσεις. Όσον αφορά την επιλογή της *sigmoid* ως συνάρτηση ενεργοποίησης στο επίπεδο εξόδου, έγινε διότι υπάρχουν μόνο δυο πιθανές τιμές εξόδου (0 και 1). Η συνάρτηση αυτή παρέχει την δυνατότητα εύκολου καθορισμού του τελικού αποτελέσματος. Όσες τιμές είναι πάνω από 0.5 τις θεωρούμε 1 και κάτω από 0.5 τις θεωρούμε 0.

Προεπεξεργασία των δεδομένων

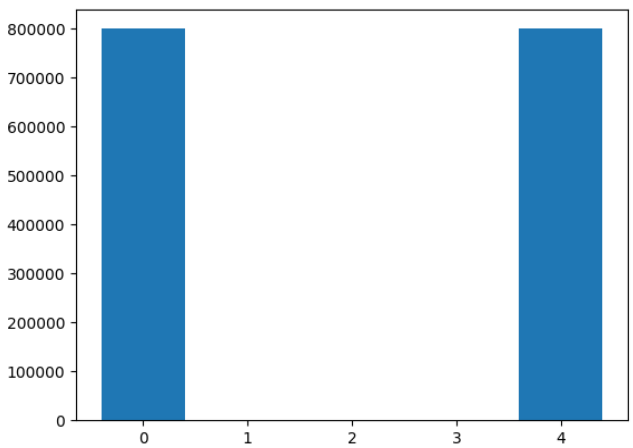
Σε πρώτη φάση, θα πρέπει να περαστούν σε κατάλληλη μορφή εισόδου για νευρωνικό δίκτυο, όλα τα δεδομένα κειμένου που διαθέταμε από το dataset. Πριν από αυτή τη διαδικασία όμως, όπως και σε κάθε text dataset, θα πρέπει τα δεδομένα να περάσουν από την φάση της προεπεξεργασίας, έτσι ώστε να έχουμε όσο το δυνατόν βέλτιστα αποτελέσματα. Αφαιρούνται, έτσι, τα:

1. Hashtags
2. Mentions
3. Links
4. Punctuations
5. Stopwords

Υπάρχουν προφανείς λόγοι κατά τους οποίους η παραπάνω διαδικασία είναι απαραίτητη:

1. Τα hashtag και τα mention στα tweets είναι κάτι το σύνηθες και αυτό ίσως να αποπροσανατόλιζε το δίκτυο μας από την σωστή πρόβλεψη του συναισθήματος.
2. Τα links συνήθως δεν είναι απαραίτητα για την επεξεργασία κειμένου και για αυτό από την βιβλιογραφία προτείνετε να τα αφαιρούμε.
3. Ανάλογα βέβαια την περίπτωση δεν χρειάζονται πάντα τα *punctuations*. Στην συγκεκριμένη περίπτωση έχει γίνει έτσι η προεπεξεργασία που η πρόταση που παράγεται χάνει το νόημα της. Επιπλέον, όταν τα παραπάνω αφαιρέθηκαν παρατηρήθηκαν καλύτερα αποτελέσματα στην μετρική της ακρίβειας.
4. Ως *stopwords* θεωρούνται όλες οι λέξεις που το νόημα του είναι ουδέτερο στην κατανόηση μιας πρότασης, ή ενός κειμένου. Για αυτό το λόγο, αφαιρούνται με την βοήθεια της βιβλιοθήκης nltk.
5. Στα δεδομένα πρόβλεψης, μέσα από το visualization του dataset παρατηρήθηκε ότι υπάρχουν 800000 θετικά και αντίστοιχα, τόσα αρνητικά *tweets,* πράγμα που σημαίνει ότι το dataset μας είναι άρτια χωρισμό. Λόγω του ότι υπάρχουν μόνο θετικά και αρνητικά *tweets,* το πρόβλημα θα μπορούσε να θεωρηθεί και ως ένα πρόβλημα δυαδικής ταξινόμησης. Για το λόγο αυτό, τις αξιολογήσεις που σχολιάστηκαν θετικές (συμβολισμένες με 4) μετατράπηκαν σε 1. Πλέον, δηλαδή, αναφερόμαστε σε ένα γραμμικό πρόβλημα που 0 σημαίνει αρνητικό και 1 σημαίνει θετικό. Επιπλέον με αυτή την αλλαγή μπορεί να χρησιμοποιηθεί στο επίπεδο εξόδου του νευρωνικού δικτύου η συνάρτηση ενεργοποίησης *relu* (*rectidfied linear activation function*) που είναι μια συνάρτηση η οποία περιέχει τιμές στο ανοικτό διάστημα (0, 1) και έτσι θα μπορούν να επιτευχθεί δόμηση της αρχιτεκτονικής του νευρωνικό δικτύου.

.



Συμπεράσματα - Αποτελέσματα

Recurrent LSTM	Dataset Separation		
Metrics	90/10	80/20	60/40
Training Loss	0.2954	0.2964	0.2971
Testing Loss	0.3017	0.3154	0.3161
Training Accuracy	0.8977	0.8855	0.8879
Testing Accuracy	0.8777	0.8662	0.8641

Recurrent GRU	Dataset Separation		
Metrics	90/10	80/20	60/40
Training Loss	0.3185	0.3287	0.3150
Testing Loss	0.3851	0.3917	0.3974
Training Accuracy	0.8659	0.8562	0.8641
Testing Accuracy	0.8282	0.8237	0.8212

Recurrent GRU Sentiment	Dataset Separation		
Metrics	90/10	80/20	60/40
Training Loss	0.1136	0.1123	0.1103
Testing Loss	0.1266	0.1269	0.1293
Training Accuracy	0.9493	0.9501	0.9502
Testing Accuracy	0.9452	0.9446	0.9437

Dense	Dataset Separation		
Metrics	90/10	80/20	60/40
Training Loss	0.1469	0.1509	0.1446
Testing Loss	1.143	1.1538	1.152
Training Accuracy	0.9328	0.9286	0.9346
Testing Accuracy	0.76	0.7577	0.7557

	Dataset Separation		
Algorithm	90/10	80/20	60/40
Logistic Regression	0.7835	0.7833	0.78
Naive Bayes	0.8569	0.8552	0.8543
Decision Tree	0.6185	0.6136	0.6112
XGBoost	0.7598	0.7598	0.7609

	Dataset Separation		
Algorithm - S	90/10	80/20	60/40
Logistic Regression	0.8982	0.8978	0.8951
XGBoost	0.8321	0.8325	0.83189