ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ ΙΙ

ΕΡΓΑΣΙΑ 1

Πρόλογος

Σκοπός της εργασίας αυτής είναι να προπονηθεί ένα μοντέλο λογιστικής παλινδρόμησης (**softmax regression**) για ένα πρόβλημα κατηγοριοποίησης πολλαπλών κλάσεων με σκοπό την περαιτέρω ανάλυση του χαρακτήρα (**sentiment analysis**) κάποιων αναρτήσεων του twitter. Στην συνέχεια, θα αναλυθούν λεπτομερώς τόσο τα βήματα που ακολουθήθηκαν όσο και κάποιες παρατηρήσεις αλλά και συγκρίσεις μεταξύ διαφορετικών πρακτικών.

Γενική περιγραφή

Πρώτο βήμα αποτελεί η φόρτωση των δύο datasets που δόθηκαν, **training** και **validation** sets, με την χρήση της βιβλιοθήκης **pandas**. Ύστερα, ακολουθεί η διαδικασία **καθαρισμού** των προηγουμένων, όπου εφαρμόζονται διάφορες τεχνικές για την αφαίρεση αχρείαστης πληροφορίας που αποτελεί κυρίως θόρυβο για το μοντέλο. Σε πρώτη φάση, γίνεται μετατροπή όλων των γραμμάτων σε **πεζά** ώστε να μην διαφοροποιούνται ίδιες λέξεις, ταυτόχρονα γίνεται κανονικοποίηση κάποιων **συντομογραφιών** (contractions) χρησιμοποιώντας μια αντίστοιχη ειδική βιβλιοθήκη και τέλος γίνεται αφαίρεση κάποιων **εξειδικευμένων στοιχείων** του κειμένου όπως των URL's, των HTML elements και tags, των retweets, των emojis και γενικότερα όλων των ειδικών χαρακτήρων και αριθμών. Η διαδικασία αυτή γίνεται ώστε να κρατηθεί η "ουσία" των κειμένων και για να επιτευχθούν αρκετά από τα προαναφερθέντα χρησιμοποιήθηκαν κυρίως **κανονικές εκφράσεις** (regular expressions).

Αφού λοιπόν, τα tweets έχουν καθαριστεί στην συνέχεια αναπαρίστανται κάποια **στατιστικά** στοιχεία που εξάγονται για όλα τα tweets του training set όπως ο αριθμός εμφάνισης κάθε λέξης συνολικά με την μορφή ραβδογράμματος (**bar plot**) και ο συνολικός αριθμός των στοιχείων του training set και τα ποσοστά των διαφορετικών labels. Σε αυτό το σημείο αξίζει να σημειωθεί το γεγονός πως το μέγεθος του training set είναι **15976** το οποίο είναι μικρός αριθμός για εκπαίδευση και το γεγονός πως υπάρχει μια δυσαναλογία ως προς τα διαφορετικά labels όπου **46,7** % καταλαμβάνουν τα ουδέτερα (**neutral**), **13** % τα αρνητικά (**antivax**) και **40,3** % τα θετικά (**provax**) με το πλήθος των αρνητικών να υστερεί των άλλων, πράγμα το

οποίο θα επηρεάσει και στην συνέχεια όπως θα προκύψει την επίδοση του μοντέλου στις αντίστοιχες προβλέψεις.

Πριν αναλυθεί η διαδικασία εκπαίδευσης αξίζει να περιγραφούν κάποιες μέθοδοι για την **βελτίωση** και την **αξιολόγηση** της απόδοσης του μοντέλου. Αρχικά, λοιπόν, υπάρχει μια μέθοδος για την δημιουργία των **learning curves** με βάση το σφάλμα (error) που προκαλούν οι προβλέψεις των διαφορετικών ισόποσων μεγεθών του training set (ξεκινώντας από ένα **5% ως το 100%**) τόσο στο **ίδιο** αυτό κομμάτι του training set όσο και σε **ολόκληρο** το validation set. Επιπλέον, υπάρχει και μια άλλη μέθοδος που αφορά την εύρεση του καλύτερου αριθμού εποχών (**epochs**) για το validation set όπου σκοπός είναι να βρεθεί η εποχή που συνδέεται με το μικρότερο λάθος στο validation set για έναν αρκετά μεγάλο αριθμό επαναλήψεων (επειδή οι καμπύλες που δημιουργούνται δεν είναι τόσο ομαλές με αποτέλεσμα να παρουσιάζονται μερικές φορές τοπικά ελάχιστα) και έτσι γίνεται **early stopping** για την αποφυγή του overfitting. Επιπρόσθετα, και οι δύο συναρτήσεις δημιουργούν learning curves, που τις απεικονίζουν καλώντας την αντίστοιχη μέθοδο, σε ένα διάγραμμα με βάση το σφάλμα (error) ως προς το μέγεθος του training set και ως προς τον αριθμό των επαναλήψεων αντίστοιχα, το οποίο μπορεί να είναι είτε το **cross entropy** (χρησιμοποιώντας τις πιθανότητες για κάθε κλάση) είτε το **misclassification error** που προκύπτει από τις διαφορετικές μετρικές και ισούται προφανώς με (1 – score_μετρικής).

Για την **αυτοματοποίηση** της διαδικασίας τόσο της **εκπαίδευσης** όσο και της **αξιολόγησης** του μοντέλου ορίζονται οι αντίστοιχες μέθοδοι οι οποίες παραμετροποιούνται ως προς το μοντέλο, ως προς τα datasets και ως προς κάποιες άλλες παραμέτρους για βελτιστοποίηση και εμφάνιση στατιστικών στοιχείων τα οποία θα αναλυθούν παρακάτω. Ωστόσο, υπάρχει και μια μέθοδος που αφορά αποκλειστικά την **βελτιστοποίηση** της διαδικασίας του training η οποία χρησιμοποιεί τη **GridSearch** μέθοδο για την ρύθμιση των βέλτιστων υπερπαραμέτρων για το μοντέλο κάνοντας αξιολόγηση με cross validation. Αυτή η διαδικασία μπορεί να μην είναι τέλεια πάντα για την περίπτωση αυτή που έχει δοθεί ένα μικρό training set έχοντας παράλληλα το validation set αλλά τις περισσότερες φορές βρίσκει αρκετά καλές παραμέτρους και καθοδηγεί παράλληλα στην εύρεση τους ακόμα και για το ίδιο το validation set.

Διαφορετικές πρακτικές εκπαίδευσης

- **Εκπαίδευση με Bag of Words vectorizer**

  Σε αυτή την περίπτωση για την απεικόνιση των καθαρισμένων κειμένων σε μορφή η οποία θα είναι αναγνωρίσιμη από το μοντέλο χρησιμοποιήθηκε ο **CountVectorizer** δηλαδή ο vectorizer που παράγει μια **Bag of Words**

απεικόνιση για να **εξάγει** τα **features** των κειμένων (για κάθε **λέξη** των κειμένων στο training set μετράει τον **αριθμό εμφάνισης** της σε κάθε κείμενο του training αλλά και του validation set). Ο vectorizer αυτός παραμετροποιήθηκε αρχικά ως προς την παράμετρο n_gram_range με τιμή (1, 2) που σημαίνει ότι για να φτιάξει το διάνυσμα θα λάβει υπόψη και τις μοναδικές λέξεις (**unigrams**) αλλά και τα ζευγάρια λέξεων (**bigrams**) που εμφανίζονται ώστε να δώσει και την πληροφορία για λέξεις που παρουσιάζονται συχνά μαζί. Αυτό όμως είχε ως αποτέλεσμα την τεράστια **αύξηση** του αριθμού των **features** για κάθε κείμενο οπότε για την μείωση των διαστάσεων και την αποφυγή του **overfitting** (το οποίο αρκετά πιθανό όσο ο αριθμός των features είναι αρκετά μεγάλος) ρυθμίστηκε και ως προς τις παραμέτρους max_features και max_df για να κρατήσει αντίστοιχα τους πιο **συχνούς** όρους (7000 συνολικοί όροι) αποκόπτοντας όμως κάποιους πάρα πολύ συχνούς (όροι που βρίσκονται τουλάχιστον στο 75% των κειμένων). Οπότε ο vectorizer επιστρέφει έναν αραιό πίνακα (**sparse matrix**) με όλες αυτές τις απεικονίσεις για κάθε κείμενο όμως παρατηρήθηκε έντονα πως η εκπαίδευση του μοντέλου **καθυστερούσε** αρκετά να **συγκλίνει**. Αυτό λοιπόν είχε να κάνει με το γεγονός πως οι τιμές δεν ήταν κανονικοποιημένες δηλαδή δεν είχαν την ίδια κλίμακα (**scale**) οπότε (όπως είχε ανεφερθεί και στις διαλέξεις) τα features με τις μικρότερες τιμές καθυστερούσαν αρκετά την **ελαχιστοποίηση** της συνάρτησης κόστους (**loss function**) . Επιπλέον, αυτό επηρέαζε αρκετά στην σύγκλιση όταν γινόταν κανονικοποίηση των **παραμέτρων** του μοντέλου (**regularization**) με την χρήση της **L1** και **L2 νόρμας**. Για την αντιμετώπιση του προβλήματος αυτού χρησιμοποιήθηκε ο **StandardScaler** που κάνει standardization ο οποίος όπως φάνηκε και στη βιβλιογραφία και στην πράξη είναι αρκετά κατάλληλος στην περίπτωση αραιών πινάκων υπό την προϋπόθεση ότι ο **μέσος όρος** (μ) θα είναι **0** και η **τυπική απόκλιση** (σ) **1** (εξού και το with_mean = False). Τέλος, αφού έγιναν κάποιες δοκιμές χρησιμοποιώντας και την μέθοδο εκπαίδευσης με GridSearch και με την βοήθεια των learning curves κάποιες καλές ενδεικτικές τιμές για το μοντέλο ήταν οι εξής:

1. C = 0.1 (σημαίνει πως ο βαθμός του regularization θα είναι μεγάλος και ίσος με 10)
2. Χρησιμοποιείται L1 νόρμα για το regularization
3. Χρησιμοποίεται ο solver saga

Παρατηρήσεις στα αποτελέσματα

Το μοντέλο κατάφερε περίπου **70**% accuracy score στο validation set. Στο **ίδιο** ποσοστό κυμαίνονται και τα precision, recall και f1 scores με **weighted** average και σε **λιγότερο** ποσοστό με **macro** average. Όπως φαίνεται και στα

confusion matrices το μοντέλο τα πηγαίνει **καλά** στα **neutral** tweets και λίγο **λιγότερο καλά** στα **provax** tweets αφού και στις δύο περιπτώσεις κάνει **αρκετές σωστές θετικές** (true) προβλέψεις και καταφέρνει παράλληλα παρόμοιο υψηλό ποσοστό για τα precision, recall και f1 scores. Βέβαια στο **γενικευμένο confusion matrix** μπορεί να δει κανείς πως **μπερδεύεται** αρκετά μεταξύ αυτών των **δύο** κλάσεων βάζοντας αρκετά tweets της μιας κλάσης στην άλλη. Όσον αφορά τα **antivax** tweets, τα πηγαίνει **μέτρια** αφού κάνει **λίγες σωστές θετικές** προβλέψεις με παρόμοιο χαμηλό ποσοστό για τα precision, recall και f1 scores. Τέλος, από τα **learning curves** μπορεί κανείς να διαπιστώσει πως τα δύο curves **απέχουν αρκετά** μεταξύ τους με το validation curve να βρίσκεται αρκετά πάνω με μικρή μείωση και με το training curve να βρίσκεται αρκετά χαμηλά και να αυξάνεται ελάχιστα (γι αυτό και το υψηλό score του training set στο τέλος της εκπαίδευσης) πράγμα το οποίο υποδηλώνει πως το μοντέλο παρά τις τεχνικές που ακολουθήθηκαν για την αντιμετώπιση του overfitting ακόμα πάσχει από αυτό σε έναν μεγάλο βαθμό. Οι τεχνικές αυτές ήταν αρκετά "αυστηρές" αφού οποιαδήποτε άλλη μικρή αλλαγή για να μειωθεί το **variance** αύξανε σε πολύ μεγάλο βαθμό το **bias** του μοντέλου που σημαίνει πως λογικά θα φταίει σε μεγάλο βαθμό η ίδια η είσοδος του vectorizer στο μοντέλο.

- **Εκπαίδευση με HashingVectorizer**

  Η περίπτωση αυτή δεν διαφέρει σε πολύ μεγάλο βαθμό από την προηγούμενη μιας και ο **HashingVectorizer** που χρησιμοποιήθηκε λειτουργεί παρόμοια με τον CountVectorizer. Η μεγάλη διαφορά είναι πως σε αυτή την περίπτωση το κάθε feature στο κάθε διάνυσμα που παράγεται δεν αντιστοιχεί σε μια συγκεκριμένη λέξη αλλά σε κάποιο συγκεκριμένο **hash** που παράγεται από μια **hash function** για κάθε λέξη που παίρνει ως είσοδο. Αυτό αναφέρεται στην βιβλιογραφία και ως hashing trick. Το αρνητικό αυτής της τεχνικής είναι πως ενδέχεται δύο διαφορετικές λέξεις να έχουν το ίδιο hash, ειδικά αν ο αριθμός των features που έχει προκαθοριστεί είναι πολύ μικρός, οπότε σε αυτή την περίπτωση θα δημιουργηθεί κάποια **σύγκρουση** (**conflict**) μεταξύ αυτών και έτσι η θέση τους στα διανύσματα θα ταυτίζεται. Για να αποφευχθούν αυτές οι συγκρούσεις και για να μην μεγαλώσει πολύ η διάσταση των διανυσμάτων (αριθμός features) λαμβάνονται υπόψη μόνο οι μοναδικές λέξεις (**unigrams**) ενώ παράλληλα ο αριθμός των συνολικών όρων του κάθε διανύσματος εξόδου προκαθορίζεται περίπου με βάση τον αριθμό των λέξεων αυτών (n_features = 20000). Τέλος, αφού έγιναν και πάλι κάποιες δοκιμές χρησιμοποιώντας και την μέθοδο εκπαίδευσης με GridSearch και με την βοήθεια των learning curves κάποιες καλές ενδεικτικές τιμές για το μοντέλο ήταν ακριβώς οι ίδιες με αυτές της προηγούμενης περίπτωσης.

Παρατηρήσεις στα αποτελέσματα

Το μοντέλο κατάφερε περίπου **67**% accuracy score στο validation set. Στο **ίδιο** ποσοστό κυμαίνονται και εδώ τα precision, recall και f1 scores με **weighted** average και σε **λιγότερο** ποσοστό με **macro** average. Γενικότερα, παρατηρούνται ακριβώς τα ίδια φαινόμενα με την προηγούμενη περίπτωση, δηλαδή αποδίδει καλύτερα στα **neutral** και στα **provax** tweets χωρίς να μπορεί να τα ξεχωρίσει και εδώ τέλεια και παρομοίως αποδίδει **χειρότερα** στα **antivax** tweets. Ταυτόχρονα διαπιστώνεται και εδώ overfitting χωρίς να μπορούμε να κάνουμε κάτι καλύτερο γι αυτό. Η μόνη διαφορά από πριν είναι πως τα φαινόμενα αυτά παρατηρούνται **εντονότερα** με πολύ **χειρότερα scores**.

- **Εκπαίδευση με TfidfVectorizer**

Και στην περίπτωση αυτή, η διαδικασία είναι παρόμοια με εκείνη του CountVectorizer. Η μόνη διαφορά είναι πως **δεν** χρειάζεται κάποια κανονικοποιήση (**scaling**) των τιμών των features του κάθε διανύσματος μιας και αυτό που ουσιαστικά το κάνει να διαφέρει σημαντικά από πριν είναι πως υπολογίζοντας τους όρους **TF** και **IDF** που προκύπτουν συνδυάζοντας πληροφορία από όλα τα κείμενα αλλά και από το κάθε κείμενο ξεχωριστά και εφαρμόζοντας τους όρους αυτούς σε κάθε τιμή που αντιστοιχεί στον αριθμό εμφάνισης των λέξεων, όπως ακριβώς υπολόγιζε και ο CountVectorizer, οι τιμές όλων των διανυσμάτων προκύπτουν ήδη κανονικοποιημένες ως προς την ίδια κλίμακα δίνοντας **μεγαλύτερο** βάρος παράλληλα στις πιο **σημαντικές** λέξεις οι οποίες δεν είναι και απαραίτητα οι πιο συχνές. Τέλος, αφού έγιναν εξίσου κάποιες δοκιμές χρησιμοποιώντας και την μέθοδο εκπαίδευσης με GridSearch και με την βοήθεια των learning curves αρχικά αποδείχτηκε ότι δούλευε καλύτερα με την αύξηση του αριθμού των max_features σε 12000 σε συνδυασμό όμως με κάποιες καλές ενδεικτικές τιμές για το μοντέλο που τυχαίνει να είναι οι default του μοντέλου και που ήταν οι εξής:

1. C = 1 (σημαίνει πως ο βαθμός του regularization θα είναι ούτε πολύ μεγάλος ούτε πολύ μικρός)
2. Χρησιμοποιείται L2 νόρμα για το regularization
3. Χρησιμοποίεται ο solver l-bfgs

Παρατηρήσεις στα αποτελέσματα

Το μοντέλο έφτασε πάρα πολύ **κοντά** στο **74**% accuracy score στο validation set. Στο **ίδιο** ποσοστό κυμαίνονται και εδώ τα precision, recall και f1 scores

με **weighted** average και σε λιγότερο ποσοστό με **macro** average για το recall κυρίως και συνεπώς για το f1 score. Στην περίπτωση αυτή μπορεί κάποιος να διαπιστώσει πως το μοντέλο τα πηγαίνει **ακόμα καλύτερα** στα **neutral** tweets και στα **provax** tweets αφού και στις δύο κλάσεις κάνει πολύ **περισσότερες σωστές θετικές** (true) προβλέψεις σε σχέση με τις προηγούμενες τεχνικές, με παρόμοιο υψηλό ποσοστό για τα precision, recall και f1 scores. Όσον αφορά τα **antivax** tweets, τα πηγαίνει **λίγο καλύτερα** αφού παρόλο που και εδώ κάνει **λίγες σωστές θετικές** προβλέψεις έχει **βελτιωθεί** με μεγάλη διαφορά το ποσοστό για το **precision**. Τέλος, από τα **learning curves** μπορεί κανείς να διαπιστώσει πως τα δύο curves **δεν απέχουν** τόσο μεταξύ τους αφού το validation curve αρχίζει από αρκετά πάνω και μειώνεται πιο απότομα και το training curve αρχίζει από αρκετά χαμηλά και αυξάνεται εξίσου πιο απότομα (γι αυτό και το πιο χαμηλό score του training set στο τέλος της εκπαίδευσης) πράγμα που σημαίνει πως **βελτιώθηκε** κατά πολύ ο βαθμός του overfitting.

Γενικά Συμπεράσματα

Με βάση τις προηγούμενες παρατηρήσεις αποδείχθηκε **καλύτερη πρακτική** η χρήση του **TfidfVectorizer** εφόσον δεν απαιτεί κάποια επιπλέον τροποποίηση, κάνει το μοντέλο να λειτουργεί αρκετά γρήγορα και επιφέρει καλύτερα scores και λιγότερο overfitting όπου αν δινόταν κιόλας **αρκετά μεγαλύτερο training set** τα training και validation curves θα ήταν ακόμα πιο κοντά. Βεβαία δεν παύει να έχει πρόβλημα η κλάση με των **antivax** tweets όμως και αυτό προκύπτει από θέμα του ίδιου του training set που όπως αναφέρθηκε και παραπάνω τα antivax tweets είναι **αρκετά λιγότερα** σε σχέση με τα tweets των άλλων κλάσεων.

Τεχνικές για επιπλέον βελτίωση της απόδοσης του μοντέλου

Αρχικά προκειμένου να **βελτιωθεί** παραπάνω η **απόδοση** του μοντέλου εφαρμόστηκε **επιπλέον** προεπεξεργασία στα tweets. Αυτή η προεπεξεργασία είναι πιο εξειδικευμένη και θα περιγραφεί στην συνέχεια τι επεξεργασία ακολουθήθηκε επιπλέον. Πρώτα, χρησιμοποιήθηκε **lemmatization** όπου είναι μια τεχνική για να βρεθεί το **λήμμα** της κάθε λέξης πράγμα αρκετά χρήσιμο γιατί όλες οι λέξεις που έχουν κοινό λήμμα έχουν την ίδια σημασία και ύφος και δεν είναι καλό να αντιμετωπίζονται ως διαφορετικές λέξεις. Για πιο αποδοτική εύρεση του λήμματος ανάλογα με τα συμφραζόμενα συνδυάστηκε και με την τεχνική του **POS tagging** που βρίσκει

το **μέρος του λόγου** της κάθε λέξης. Εκτός από lemmatization υπάρχει και η τεχνική του **stemming** που βρίσκει την ρίζα της κάθε λέξης όμως έκανε **χειρότερη** την **απόδοση** του μοντέλου οπότε δεν χρησιμοποιήθηκε. Επίσης, από την **στατιστική ανάλυση** των tweets που έγινε στην αρχή παρατηρήθηκε σε μεγάλο ποσοστό η χρήση κάποιων λέξεων που χρησιμοποιούνται καθημερινώς (**stopwords**) και δεν προσφέρουν κάποιο ιδιαίτερο νόημα (π.χ the, and, of, that…) οπότε είναι καλό να αφαιρεθούν με **εξαίρεση** κάποιες λέξεις που μπορεί να προσφέρουν κάποιο συναίσθημα/ύφος (sentiment) (π.χ no, not, i, we …). Παράλληλα από την στατιστική ανάλυση φαίνεται πως κάποιες λέξεις **μικρού μεγέθους** (π.χ με ένα ή δύο γράμματα) αποτελούν απλά θόρυβο και δεν συνεισφέρουν σε κάτι οπότε πρέπει να αφαιρεθούν και αυτές. Οπότε στο τέλος γίνεται πάλι μια στατιστική ανάλυση και φαίνονται όλα όσα αναφέρθηκαν προηγουμένως πως έχουν εφαρμοστεί επιτυχώς.

Όσον αφορά την **εκπαίδευση** του μοντέλου χρησιμοποιήθηκε ο TfidfVectorizer χωρίς καμία διαφορά στις παραμέτρους του ίδιου του vectorizer και του μοντέλου όμως η μόνη σημαντική διαφορά είναι πως τώρα χρησιμοποιήθηκε η τεχνική του **early stopping**. Όπως φαίνεται στα αποτελέσματα στην αρχή υπάρχει ένα νέο γράφημα με learning curves ως προς τον αριθμό των επαναλήψεων με ένα βέλος στην καλύτερη επανάληψη όπου το **λάθος** στο **validation set** ήταν το **μικρότερο** για έναν μεγάλο αριθμό επαναλήψεων αφού είτε θα υπήρχαν και άλλες πιο μικρές αυξομειώσεις ή ακόμα και μόνο αυξήσεις του λάθους είτε θα σταθεροποιούταν το λάθος όπως στην προκειμένη περίπτωση οπότε δεν υπάρχει λόγος να εκπαιδεύεται παραπάνω το μοντέλο με αποτέλεσμα να προκύψει κάποιο overfitting.

Τα αποτελέσματα έγιναν ακόμα καλύτερα ως προς το accuracy score όπου ξεπέρασε το **74**% (πράγμα το οποίο δεν ήταν και τόσο εύκολο) και παρομοίως υπήρχαν και στα περισσότερα άλλα scores πολύ μεγαλύτερες αυξήσεις με σχεδόν ελάχιστη όμως μείωση κάποιων άλλων λίγων. Το σημαντικότερο είναι πως τα **learning curves** έχουν έρθει **ακόμα πιο κοντά** που σημαίνει ότι έχει βελτιωθεί αρκετά το μοντέλο.

Κλείνοντας με την εκπαίδευση του μοντέλου υπάρχουν κάποιες ενδεικτικές δοκιμές που έγιναν με διαφορετικές παραμέτρους του μοντέλου για να επιλεχθούν οι καλύτερες δηλαδή αυτές που με τις οποίες επιτυγχάνονται τα πιο ικανοποιητικά learning curves. Οι παράμετροι αυτοί συμβαδίζουν με αυτές που επίδειξε και η εκπαίδευση με την χρήση του GridSearch.

Τέλος, από περιέργεια δοκιμάστηκε και ένα άλλο μοντέλο για κατηγοριοποίηση πολλών κλάσεων το **Multinomial Naive Bayes** που είναι και αυτό ένα πιθανοτικό μοντέλο που χρησιμοποιεί τον νόμο του Bayes.

Βέβαια τα αποτελέσματα **δεν** ήταν καθόλου **καλύτερα** οπότε δοκιμάζοντας και άλλα μοντέλα δεν βελτιώνεται παραπάνω η απόδοση.

Οδηγίες για τη δοκιμή με το test set

Επειδή η δομή της εργασίας δεν επιτρέπει την άμεση εκτέλεση των κελιών αφού χρησιμοποιείται το validation set σε αρκετές δοκιμές υπάρχουν σχόλια του τύπου **#RUN THIS** ώστε με την φόρτωση του test set στην θέση του validation να μπορεί να εξεταστεί. **Προσοχή** όμως το όνομα των μεταβλητών δεν έχει αλλάξει σε μεταβλητές τύπου test set οπότε εννοείται πως κατά την εκτέλεση το validation set είναι το test set που έχει φορτωθεί.

Βιβλιογραφία/Πηγές

- Διαφάνειες μαθήματος
- Παλαιότερα παρόμοια projects και βιβλία από άλλα αντίστοιχα μαθήματα της σχολής όπως Data Mining
- Οι αντίστοιχες ενότητες από το βιβλίο του Speech and Language Processing (Jurafsky)
- Sklearn (Tutorials + Examples)
- Stack Overflow
- Stack Exchange
- Πάρα πολλά άρθρα και blogs για ιδέες πάνω στο θέμα όπως
  - https://www.analyticsvidhya.com/blog/2015/10/6-practices-enhance-performance-text-classification-model
  - https://machinelearningmastery.com/learning-curves-for-diagnosing-machine-learning-model-performance
  - https://www.dataquest.io/blog/learning-curves-machine-learning
  - https://www.kaggle.com/pratikbarua/twitter-sentiment-analysis-logistic-regression
  - https://www.stackvidhya.com/plot-confusion-matrix-in-python-and-why
  - …

Ορφέας Τσουράκης

1115201700175