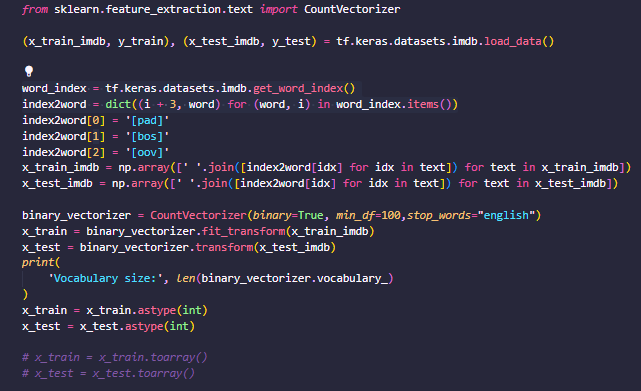
|  |  |  |  |
| --- | --- | --- | --- |
|  | Τεχνητή Νοημοσύνη 2η Εργασία | |  |
|  | |
|  | |
|  | Ταξινόμηση κριτικών IMDB | |  |
| * Σοφία-Ζωή Σωτηρίου : 3210192 * Ευάγγελος Λευτάκης 3200093 * Ρέα Σκλήκα 3210181 |  |  |
|  | |  |

Αρχικοποίηση Δεδομένων

Αφού λάβουμε το imdb dataset μαζί το λεξικό word index του μέσω του Keras, προσθέτουμε στο λεξικό τις λέξεις [pad], [bos], [oov] που είναι tokens με ειδική σημασία για την αναπαράσταση κειμένου. Έπειτα με την χρήση του countVectorizer φιλτράρουμε τις λέξεις και κρατάμε μόνο αυτές που εμφανίζονται τουλάχιστον 100 φορές (min\_df) και παράλληλα μετατρέπουμε τις προτάσεις σε δυαδικές αναπαραστάσεις από 0 για απουσία και 1 για εμφάνιση μιας λέξης σε κάθε πρόταση. Αλλάζουμε τον τύπο τον δεδομένων από float64 που είναι το default σε integer για βελτιωμένη απόδοση. Τέλος, μετατρέπουμε τα δεδομένα μάθησης και ελέγχου σε dense arrays με την χρήση της toarray() καθώς by default λαμβάνουμε ένα sparse array από τον count vectorizer η οποία μορφή δεν είναι συμβατή με πολλές από τις διαδικασίες που απαιτούνται για τις διάφορες μεθόδους τις εργασίας.



1ο Μέρος

Για να κάνουμε την κατηγοριοποίηση των κριτικών από το dataset μας στο πρώτο κομμάτι της εργασίας αναπτύξαμε APIs για τους αλγορίθμους μηχανικής μάθησης Naive Bayes, Logisitic Regression και AdaBoost και να αξιολογήσουμε τις επιδόσεις τους.

Logistic regression:

Ο αλγόριθμος που υλοποιήσαμε είναι λογιστικής παλινδρόμησης με L2 κανονικοποίηση και κανόνα ανανέωσης βαρών: 

Η διεπαφή αποτελείται από τις παρακάτω μεθόδους:

* \_\_init\_\_(*epochs* ,*learning\_rate* ,*threshold* ,*regularization\_factor*): Παίρνει ως ορίσματα τις διάφορες υπερπαραμέτρους όπως αριθμός επαναλήψεων, ρυθμός μάθησης παράγοντας κανονικοποίησης και threshold. Κατασκευάζει και επιστρέφει τον ταξινομητή με τα ορίσματα που προσδιορίσαμε.
* fit(x\_train\_input,y\_train\_input): Παίρνει ως ορίσματα τα δεδομένα εκπαίδευσης. Αρχικά διαιρούμε τα δεδομένα σε εκπαίδευσης και επικύρωσης 80-20 αντίστοιχα. Για κάθε εποχή τα παραδείγματα αναδιατάσσονται τυχαία προς αποφυγή overfitting και να μην δημιουργηθούν γνώσεις που έχουν να κάνουν με την διάταξη των παραδειγμάτων εκπαίδευσης. Για την ενημέρωση των βαρών χρησιμοποιούμε Stochastic Gradient Ascent χωρίς mini batches. Επιστρέφει διάφορες μετρικές που παράγονται από την evaluate() (εμφανίζεται παρακάτω)
* predict(x\_test): Παίρνει ως όρισμα τα δεδομένα ελέγχου. Με τον τύπο τις σιγμοειδούς καμπύλης υπολογίζουμε την πιθανότητα κάθε παράδειγμα ελέγχου να είναι στην θετική κατηγορία. Έπειτα αν η πιθανότητα αυτή είναι μεγαλύτερη από το threshold θωρούμε ότι η κριτική είναι θετική. Επιστρέφει έναν πίνακα y\_predicted με μήκος ίσο με τα παραδείγματα ελέγχου ο οποίος αποτελείται από 0 και 1 για εκτιμόμενα θετική ή αρνητική κριτική.
* predict\_prob(x\_test): Το ίδιο με την predict απλά επιστρέφει τις εκτιμώμενες πιθανότητες για τα δεδομένα ελέγχου χωρίς να τα κατατάσσει.
* initialize\_weights(): Αρχικοποιούμε τα βάρη σε τυχαίους αριθμούς πολύ κοντά στο μηδέν.
* update\_weights(x\_train, y\_train): Χρησιμοποιώντας sga με τον τύπο ανανεώνουμε τα βάρη με τον τύπο ανανέωσης.
* evaluate(y\_test, y\_predicted): Συγκρίνουμε τα αποτελέσματα που εκτιμήθηκαν με τα πραγματικά και επιστρέφουμε ένα πίνακα με μετρικές όπως [precision,recall,f1]
* Άλλες Βοηθητικές μέθοδοι όπως set\_threshold() για αλλαγή του threshold ενός ταξινομητή, sigmoid() για υπολογισμό του τύπου .

Αποτελέσματα:

A graph of a logistic regression results

Description automatically generated

A screenshot of a graph

Description automatically generated

A graph with a line and a line

Description automatically generated with medium confidence

A black and white chart with white text

Description automatically generated

Naïve Bayes:

Ο αλγόριθμος που χρησιμοποιείται είναι αυτός του Αφελή ταξινομητή Bayes (πολυμεταβλητή μορφή Bernoulli)

Η διεπαφή αποτελείται από τις παρακάτω μεθόδους:

* \_\_init\_\_(alpha):

Παίρνει ως όρισμα την υπερπαράμετρο alpha, η οποία είναι η εκτιμήτρια Laplace και κατασκευάζει τον ταξινομητή αρχικοποιώντας τη πιθανότητα ένα παράδειγμα να είναι θετικό (και από εκεί προκύπτει και η πιθανότητα να είναι αρνητικό ως 1-θετικήΠιθανότητα) και οι δεσμευμένες πιθανότητες αν μία λέξη δεν υπάρχει | το παράδειγμα είναι θετικό και αν μια λέξη δεν υπάρχει | το παράδειγμα είναι αρνητικό. Από αυτές τις δύο προκύπτουν και οι δεσμευμένες πιθανότητες αν υπάρχει μια λέξη το παράδειγμα να είναι θετικό/αρνητικό.

* fit(x\_train\_input,y\_train\_input):

Παίρνει ως ορίσματα τα δεδομένα εκπαίδευσης και αρχικά τα χωρίζει (80% για εκπαίδευση και 20% για επικύρωση). Έπειτα υπολογίζει τη πιθανότητα ένα παράδειγμα να είναι θετικό με βάση τα δεδομένα εκπαίδευσης. Μετά για κάθε λέξη υπολογίζει τη δεσμευμένη πιθανότητα αν αυτή η λέξη δεν υπάρχει | το αποτέλεσμα είναι θετικό/αρνητικό χρησιμοποιώντας και εκτιμήτρια Laplace.

* getParams();

Επιστρέφει τη παράμετρο alpha

* predict(x\_test):

Παίρνει ως όρισμα τα δεδομένα ελέγχου. Για κάθε παράδειγμα υπολογίζονται τα αθροίσματα των λογαρίθμων όλων των δεσμευμένων πιθανοτήτων κάθε λέξης του παραδείγματος και για θετικό και για αρνητικό αποτέλεσμα. Προστίθενται και οι λογάριθμοι των πιθανοτήτων ένα παράδειγμα να είναι θετικό/αρνητικό και συγκρίνονται τα δύο αθροίσματα. Επιστρέφει έναν πίνακα με μήκος ίσο με τα παραδείγματα ελέγχου ο οποίος αποτελείται από 0 και 1 για εκτιμόμενα θετική ή αρνητική κριτική ανάλογα με το ποιο άθροισμα ήταν μεγαλύτερο.

* predict\_proba(x\_test):

Κάνει την ίδια δουλειά με το predict απλά επιστρέφει μόνο τις πιθανότητες που υπολόγισε χωρίς να κατατάσσει τα παραδείγματα.

* evaluate(y\_true, y\_predicted):

Συγκρίνει τα αποτελέσματα της εκτίμησης με τα σωστά αποτελέσματα και επιστρέφει έναν πίνακα με μετρικά (precision, recall, f1 score).

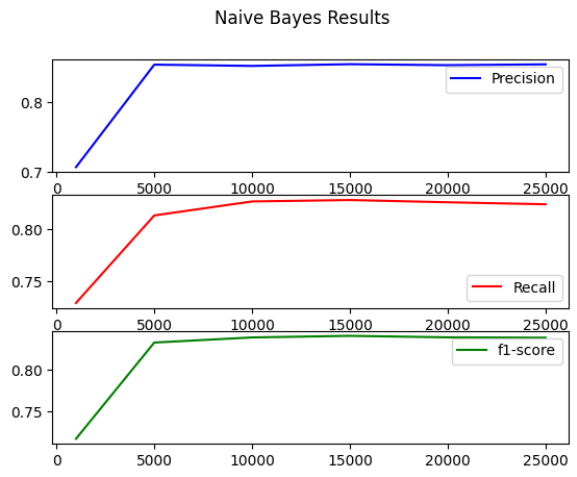
* classification\_diagrams(x\_train, y\_train,x\_test, y\_test, train\_sizes):

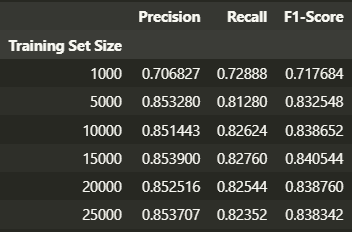
Δέχεται όλα τα δεδομένα εκπαίδευσης και ελέγχου και παράγει διαγράμματα precision, recall, f1 πίνακα με τα σκορ αυτών των μετρικών.

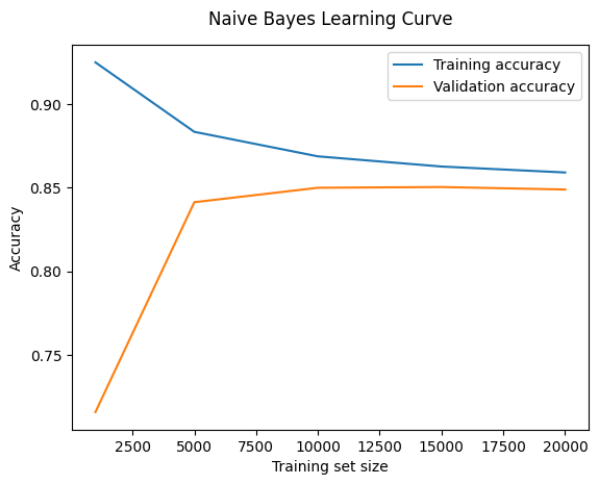
* custom\_classification\_report(x\_train, y\_train,x\_test, y\_test, train\_sizes):

Δέχεται όλα τα δεδομένα εκπαίδευσης και ελέγχου και παράγει καμπύλη εκπαίδευσης.

Αποτελέσματα:







A black and white screen with numbers

Description automatically generated

AdaBoost:

Ο αλγόριθμος που χρησιμοποιείται είναι

2ο Μέρος

Σε αυτό το σημείο θα συγκρίνουμε τις υλοποιήσεις των αλγορίθμων μας με τις υλοποιήσεις του sci-kit learn. Για αυτή την σύγκριση πέρα τον μετρικών ακρίβειας όπως accuracy, precision, recall και f1 score έχουμε υλοποιήσει διάφορα διαγράμματα όπως precision-recall καμπύλη, ROC καμπύλη και καμπύλη μάθησης.