Τεχνητή Νοημοσύνη
2η Σειρά Ασκήσεων

Δημήτριος Αναστασόπουλος - 3180010

Μιχαήλ Δικαιόπουλος - 3180050

Κωνσταντίνος Καρράς - 3180076

Οι αλγόριθμοι που έχουμε υλοποιήσει είναι ο Naive Bayes, ID3 και Ada Boost. Σε όλους τους αλγορίθμους έχουμε χρησιμοποιήσει συνάρτηση κέρδους πληροφορίας για την επιλογή ιδιοτήτων. Όλοι οι αλγόριθμοι έχουν μείνει πιστοί στις οδηγίες της εργασίας και παίρνουν ως όρισμα τις κριτικές σε μορφή διανυσμάτων με 0, 1 χαρακτήρες.

Υπάρχει ένα γενικό αρχείο main.py, από το οποίο μπορούν να κληθούν οι επιμέρους αλγόριθμοι. Επίσης υπάρχει ένα αρχείο functions.py, το οποίο περιέχει γενικές συναρτήσεις για την επεξεργασία των δεδομένων.

Εξαιτίας της έλλειψης δεδομένων ανάπτυξης (development data) αναγκαστήκαμε να χωρίσουμε τα δεδομένα εκπαίδευσης (training data) σε 2 κατηγορίες, ώστε να αποκτήσουμε δεδομένα στα οποία θα μπορούμε να δοκιμάσουμε τις τιμές των υπερπαραμέτρων. Έτσι, το πρώτο 95% των κριτικών του train φακέλου παρέμειναν ως δεδομένα εκπαίδευσης και το τελευταίο 5% χρησιμοποιείται για ανάπτυξη.

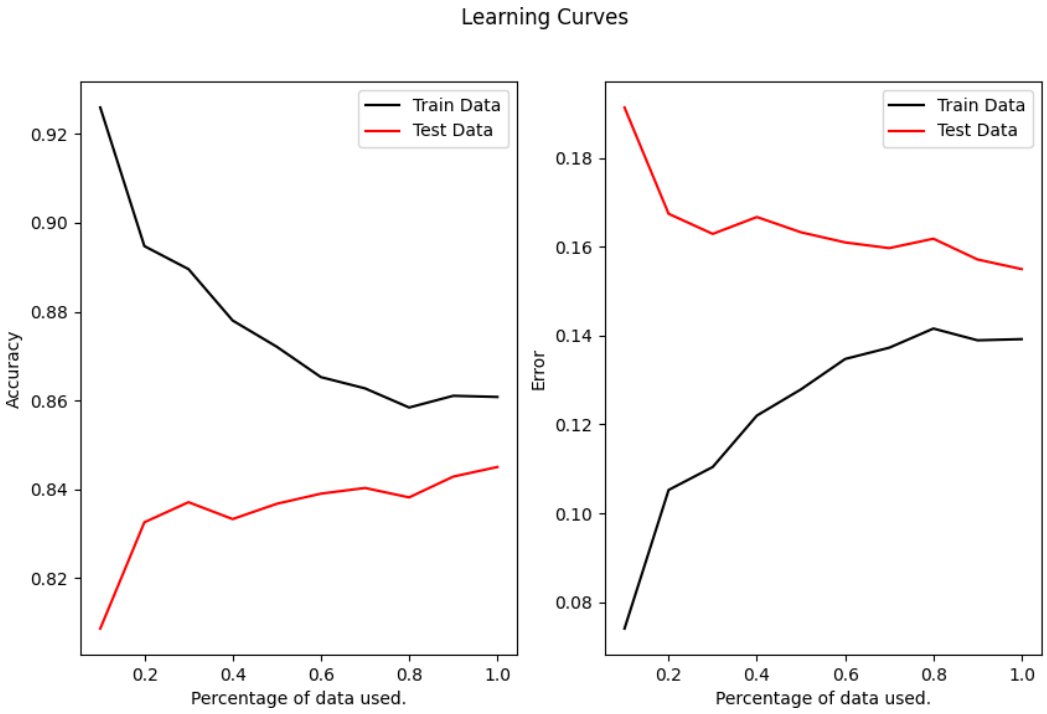
Naive Bayes

Η υλοποίηση του Naive Bayes περιέχεται στο αρχείο naiveBayes.py και εκτελείται από το αρχείο main_naiveBayes.py, το οποίο καλείται από την main.py για να αρχικοποιήσει τα απαραίτητα δεδομένα και να καλέσει τις συναρτήσεις train & evaluate. Επίσης υπάρχει το αρχείο metrics_naiveBayes.py, το οποίο περιέχει συναρτήσεις που μπορούν να υλοποιήσουν την διαδικασία κατασκευής της καμπύλης μάθησης. Οι συναρτήσεις του αρχείου αυτού μπορούν επίσης να καλεστούν από το main_naiveBayes.py.

Η μόνη υπερπαράμετρος που χρησιμοποιεί ο συγκεκριμένος αλγόριθμος είναι το πλήθος των ιδιοτήτων που θέλουμε να έχουν τα διανύσματα κριτικών που θα περνάμε για τις διαδικασίες της εκπαίδευσης και αξιολόγησης. Με πολλαπλές δοκιμές που κάναμε καταλήξαμε ότι η καλύτερη υπερπαράμετρος είναι η τιμή 1200, με την οποία πετύχαμε αρκίβεια (accuracy) 87.72% στα δεδομένα ανάπτυξης. Πέρα από την συγκεκριμένη τιμή δοκιμάσαμε επίσης τις ακόλουθες:

Τιμή υπερπαραμέτρου	**Ακρίβεια στα δεδομένα ανάπτυξης**
100	85.216%
500	86.824%
1000	87.496%
1100	87.624%
1200	87.72%
1250	87.624%
1500	87.4%

Η καμπύλη μάθησης που προέκυψε από την εξειδικευμένη συνάρτησή μας με την βέλτιστη τιμή υπερπαραμέτρου 1200 είναι η ακόλουθη:



Δεν έχουμε υλοποιήσει καμπύλες precision-recall, καθώς οι συγκεκριμένες καμπύλες έχουν ως σκοπό την εύρεση του κατάλληλου threshold και οι αλγόριθμοι που έχουμε επιλέξει δεν κατατάσσουν τις κριτικές με βάση κάποιο threshold. Στην περίπτωση του Naive Bayes γίνεται απλώς μία σύγκριση της πιθανότητας να ανήκει η κριτική στις θετικές και της πιθανότητας να ανήκει η κριτική στις αρνητικές. Συνεπώς, η έξοδος του αλγορίθμου είναι δυαδική και όχι συνεχής τιμή. Μια και η εύρεση του καλύτερου threshold δεν εξυπηρετεί την λειτουργία των αλγορίθμων μας παραθέτουμε τις τιμές των precision, recall και f1 για τον αλγόριθμο με χρήση της βέλτιστης υπερπαραμέτρου.

$$precision = \frac{true\ positives}{true\ positives + false\ positives} = 83.553\%$$

$$recall = \frac{true\ positives}{true\ positives + false\ negatives} = 85.92\%$$

$$f1 = 2 * \frac{precision * recall}{precision + recall} = 84.719\%$$

Το **accuracy** με την βέλτιστη υπερπαράμετρο ισούται με **84.504%**.

ID3

Για τον αλγόριθμο ID3 χρησιμοποιούμε τα εξής αρχεία:

- run_ID3.py
  Είναι ένα μενού αποκλειστικά και μόνο για τον ID3, που δίνει 2 επιλογές στον χρήστη. Είτε να τρέξει τον αλγόριθμο για τα test data, είτε να του εμφανίσει τις καμπύλες μάθησης.
- ID3.py
  Περιέχει 2 κλάσεις (Node και ID3). Στη main φτιάχνουμε ένα αντικείμενο του δέντρου (tree = ID3()) και μετά κάνουμε insert τα στοιχεία που έχουν προέλθει από την train() (ακολουθεί πιο κάτω η σημασία της). Επιστρέφουμε στη μεταβλητή root την ρίζα του δέντρου.
- MetricsForID3.py
  Περιέχει τις εξής συναρτήσεις:
  - train(): Επιστρέφει τα απαραίτητα στοιχεία ώστε να ¨ προπονηθεί ¨ ο αλγόριθμος ID3, ο οποίος επιστρέφει δέντρο.
  - classifier(): Εμφανίζει και επιστρέφει όταν καλείται πόσες σωστές και πόσες λάθος κριτικές έχει κατατάξει ο αλγόριθμος.
  - get_accuracy: Επιστρέφει το accuracy, δηλαδή το ποσοστό επιτυχίας των σωστά κατανεμημένων κριτικών.
  - get_error: Το αντίθετο του get_accuracy.
  - get_learning_curve_points: Ανάλογα με το πλήθος των κριτικών που βρίσκονται στα train δεδομένα και χρησιμοποιούνται την εκάστοτε χρονική στιγμή ($1^η$ φορά -> 10%, $2^η$ φορά -> 20%, …) παίρνουμε τα ποσοστά επιτυχίας για τα train, καθώς και για όλα τα test δεδομένα. Τα αποθηκεύουμε στις αντίστοιχες λίστες και τα χρησιμοποιούμε για να κάνουμε plot το διάγραμμα στην συνάρτηση plot_learning_curve().

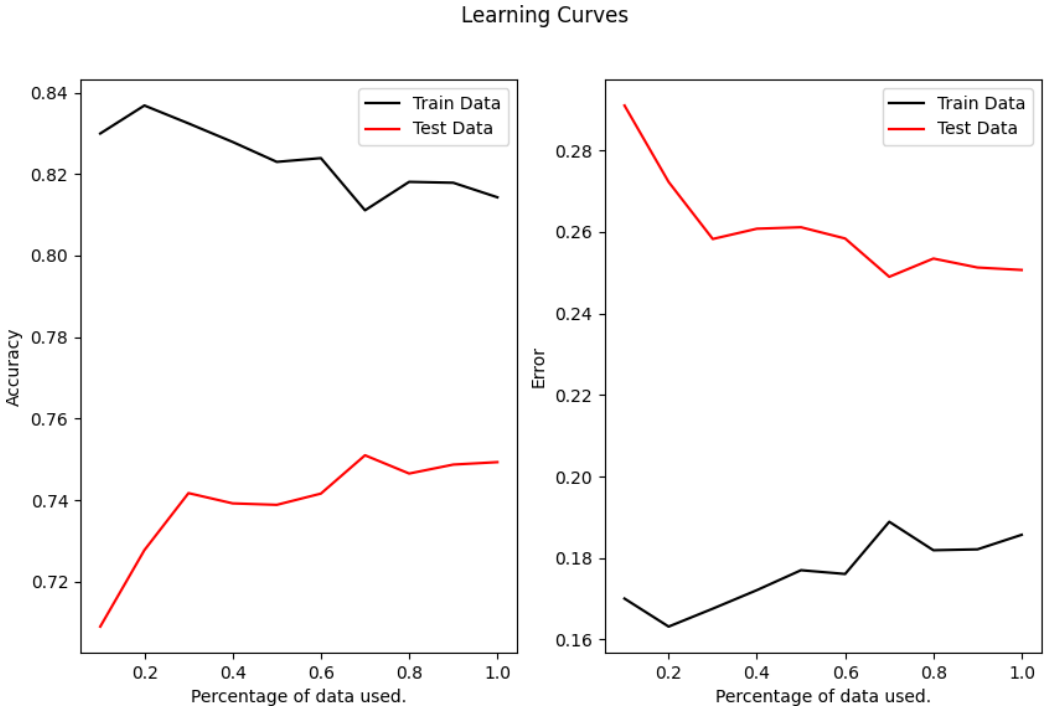
Για την εύρεση της τιμής της υπερπαραμέτρου χωρίζουμε τα train δεδομένα σε train και development με την αναλογία 95% και 5% αντίστοιχα. Η βέλτιστη υπερπαράμετρος είναι το 26 και επιλέχθηκε κατόπιν πολλαπλών δοκιμών. Ακολουθεί πίνακας που εμφανίζει τα ποσοστά επιτυχίας στα development data.

Υπερπαράμετρος	**Ποσοστό Επιτυχίας στα Development Data(%)**
5	69.12
10	71.6
20	75.44
25	75.76
26	**75.84**
27	75.28
30	73.68
50	71.04
100	69.52

Παρατήρηση 1: Αυτή η παράμετρος μεγιστοποιεί το accuracy στα development data. Προφανώς, όσο μεγαλύτερη ήταν η παράμετρος τόσο μεγαλύτερο ήταν και το accuracy στα train δεδομένα, κάτι που δεν συνέβαινε στα development data. Και αυτό, διότι μάλλον γινόταν overfitting πάνω στα train δεδομένα με αποτέλεσμα να μην είναι αντιπροσωπευτικό το δέντρο για ξένα δεδομένα.
Παρατήρηση 2:Παρεμπιπτόντως, στον ID3 δεν σταματάμε όταν ένας κόμβος του δέντρου έχει μόνο θετικές ή αρνητικές κριτικές μέσα, αλλά όταν έχει τουλάχιστον 95% κριτικές που είναι είτε θετικές, είτε αρνητικές (για να μην υπάρχει πάλι πλήρης διαχωρισμός).

Η καμπύλη μάθησης που προέκυψε από την εξειδικευμένη συνάρτησή μας με την βέλτιστη τιμή υπερπαραμέτρου 26 είναι η ακόλουθη:

Learning curves



Precision – Recall – F1

$$precision = \frac{true\ positives}{true\ positives + false\ positives} = 73.956\%$$

$$recall = \frac{true\ positives}{true\ positives + false\ negatives} = 77.522\%$$

$$f1 = 2 * \frac{precision * recall}{precision + recall} = 75.697\%$$

Το **accuracy** με την βέλτιστη υπερπαράμετρο ισούται με **74.932%**.

AdaBoost

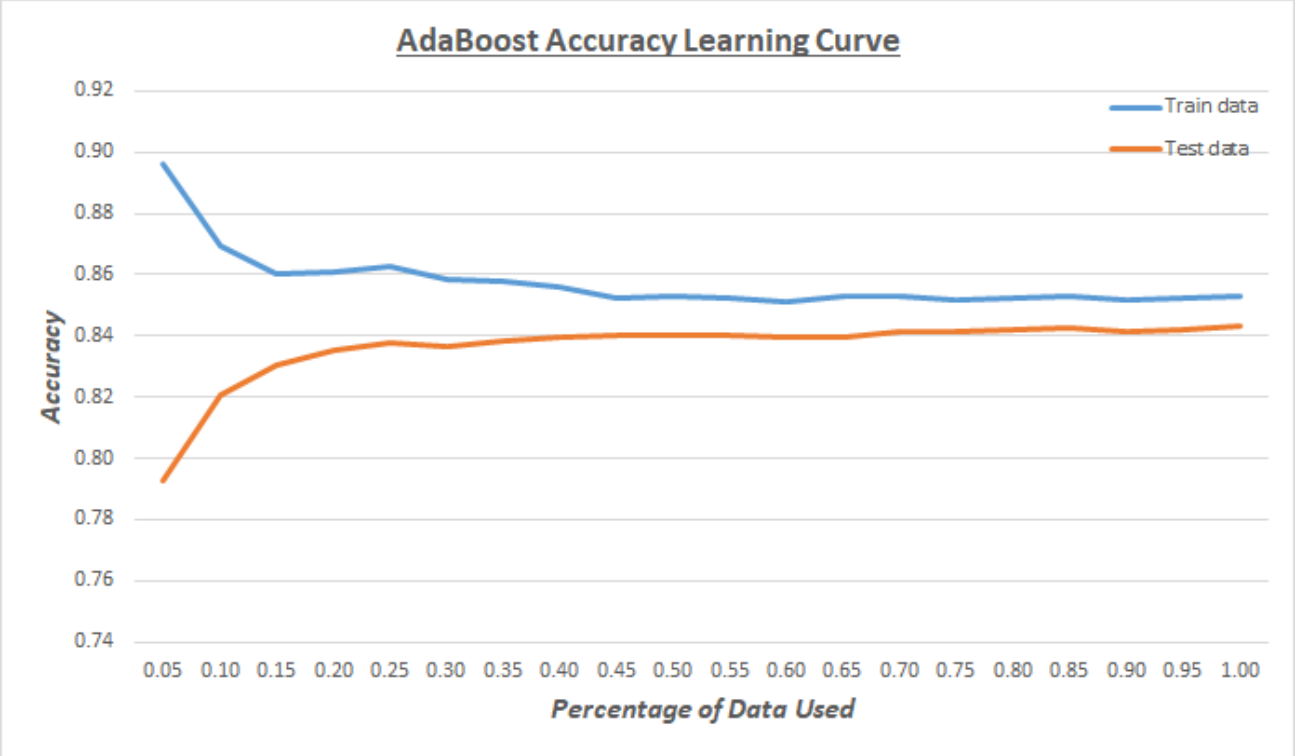
Τα αρχεία που αφορούν τον αλγόριθμο AdaBoost είναι τα run_adaboost.py και adaboost.py. Αφού δοθεί εντολή εκτέλεσης της συνάρτησης run_adaboost(), υλοποιείται η εκπαίδευση με χρήση των train reviews και η αξιολόγηση των test reviews βάσει των παραγόμενων classifiers από την εκπαίδευση. Από το σύνολο των train reviews, χρησιμοποιείται το 95% για την εκπαίδευση και ένα 5% ως development δεδομένα, τα οποία χρησιμοποιήθηκαν κυρίως για την βελτιστοποίηση των υπερπαραμέτρων.

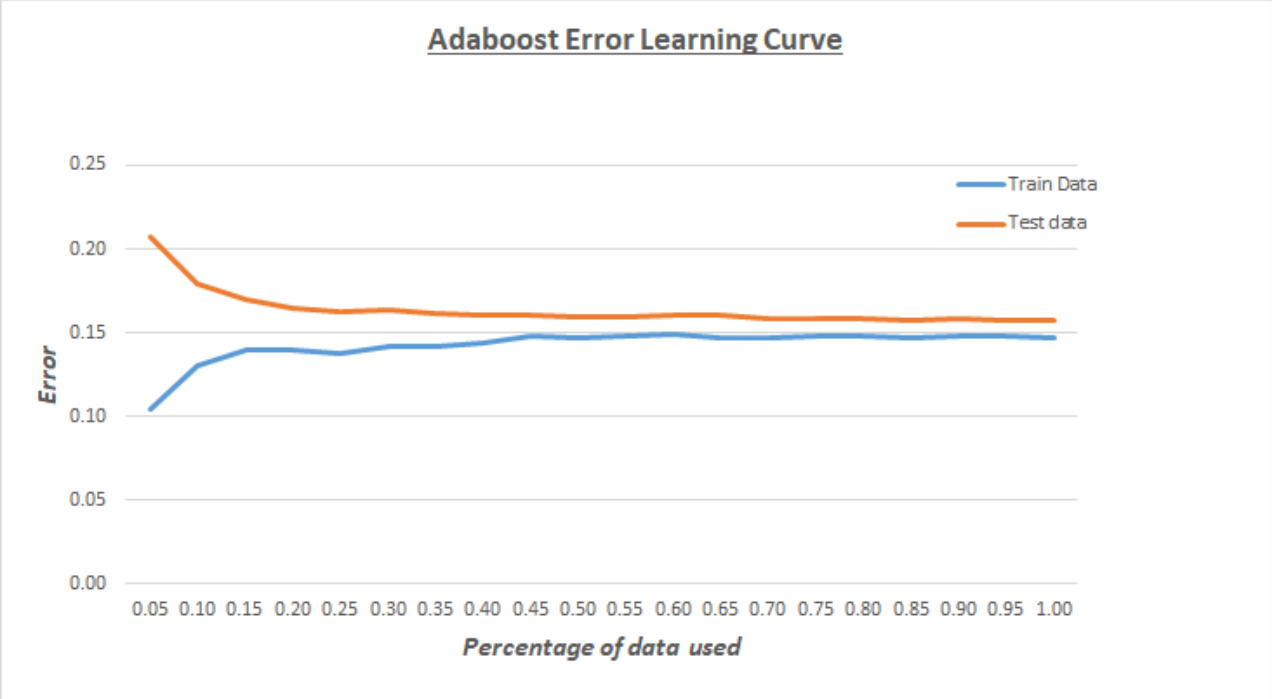
Η εκπαίδευση γίνεται με χρήση της συνάρτησης train στο αρχείο adaboost.py. Βασίζεται σε τρεις υπερπαραμέτρους : **το πλήθος των ιδιοτήτων**(λέξεων) που χρησιμοποιούνται, **το πλήθος των weak classifiers** που υλοποιούνται και **το επιπλέον ποσοστό μείωσης του ύψους του βάρους** που δίνεται στις σωστές προβλέψεις κάθε classifier. Ελέγχθηκε η επίδραση διαφορετικών υπερπαραμέτρων στις προβλέψεις του αλγορίθμου στα development δεδομένα και παρατηρήθηκε βέλτιστη ακρίβεια με χρήση **300 λέξεων, 600 classifiers και 0.982 ποσοστό** αντίστοιχα.

Συνοπτικά, ο αλγόριθμος train λειτουργεί ως εξής : Λαμβάνει ως εισόδους ένα λεξιλόγιο των λέξεων με το μεγαλύτερο information gain (παρατηρήθηκε βελτιστοποίηση της ακρίβειας του με αυτήν την μέθοδο) τις train reviews και τις υπερπαραμέτρους. Αρχικοποιούνται οι πίνακες των βαρών και των weak classifiers και σε κάθε επανάληψη κατασκευάζεται ένα **decision stump** (weak classifier). Για την επιλογή της ιδιότητας που χρησιμοποιείται σε κάθε decision stump, ελέγχεται το σταθμισμένο σφάλμα πρόβλεψης με βάση την τιμή κάθε ιδιότητας και επιλέγεται η ιδιότητα με το μικρότερο σφάλμα. Έπειτα, αναπροσαρμόζονται τα βάρη, ώστε να δοθεί μεγαλύτερη έμφαση στην επόμενη επανάληψη στις κριτικές που αξιολογήθηκαν λανθασμένα από τον επιλεγμένο classifier και υπολογίζεται η βαρύτητα του classifier, βάσει του σφάλματος στα δεδομένα εκπαίδευσης. Επιστρέφεται ο πίνακας με όλους τους classifiers.

Ο αλγόριθμος predict λαμβάνει ένα σύνολο κριτικών και τους classifiers που επέστρεψε ο αλγόριθμος train και με βάση τη βαρύτητα του κάθε classifier αξιολογεί τις κριτικές με βάση την απάντηση της σταθμισμένης πλειοψηφίας.

Learning curves





Precision – Recall – F1

$$precision = \frac{true\ positives}{true\ positives + false\ positives} = 83.15\%$$

$$recall = \frac{true\ positives}{true\ positives + false\ negatives} = 86.07\%$$

$$f1 = 2 * \frac{precision * recall}{precision + recall} = 84.59\%$$

Το **accuracy** με την βέλτιστη υπερπαράμετρο ισούται με **84.316%**.