## Συμμετέχοντες: 1) Μπατζάνας Νικόλαος (3210130)

## 2)

## Υλοποιήσεις αλγορίθμων:

## Μπατζάνας Νικόλαος (3210130): Naïve Bayes, AdaBoost

## : Random Forest

ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ

2023-2024

## **2Η ΕΡΓΑΣΙΑ**

## Σύντομη περιγραφή

Στην παρούσα εργασία έχουν υλοποιηθεί οι εξής αλγόριθμοι μηχανικής μάθησης: 1) Naive Bayes 2)Random Forest 3) AdaBoost στην γλώσσα Python με στόχο την ταξινόμηση κειμένων σε δύο κλάσεις ( binary classification ). Για την επεξεργασία των δεδομένων του δοσμένου dataset υλοποιήθηκαν καινούριες συναρτήσεις και δεν χρησιμοποιήθηκαν έτοιμες βιβλιοθήκες. Για κάθε αλγόριθμο που υλοποιήθηκε υπάρχει η δυνατότητα να εισαχθούν τιμές στις υπερπαραμέτρους (m,n,k) ώστε τα κείμενα που εξετάζουν οι αλγόριθμοι να παριστάνονται ως διανύσματα μεγέθους (m) στα οποία η τιμή είναι (1) εάν υπάρχει η συγκεκριμένη από τις (m) λέξεις στο κείμενο ή 0 εάν δεν υπάρχει. Το διάνυσμα αποτελείται από τις (m) πιο συχνές λέξεις των δεδομένων εκπαίδευσης παραλείποντας αρχικά τις (n) πιο συχνές και τις (κ) πιο σπάνιες.

Στην υλοποίηση των καμπυλών μάθησης χρησιμοποιήθηκε παραλλαγή του αλγορίθμου custom\_learning\_curve() από το lab4 του μαθήματος. Οι καμπύλες που αναπαριστούν τιμές precision, recall και f1 δημιουργήθηκαν με καινούριο εξ αρχής αλγόριθμο αξιοποιώντας την βιβλιοθήκη matplotlib καθώς επίσης και το classification\_report της scikit-learn.

Όλες οι υλοποιήσεις διαθέτους μεθόδους fit() και predict() για να εκπαιδεύονται στα δεδομένα και να προβλέπουν αντίστοιχα.

Ο αλγόριθμος AdaBoost υπάρχει με δύο υλοποιήσεις. Η πρώτη υλοποίηση του κάνει χρήση ενός DesicionStump με υλοποίηση from scratch. Ενώ η δεύτερη υλοποίηση χρησιμοποιεί την έτοιμη κλάση DecisionTreeClassifier της Scikit-learn με περιορισμό ανάπτυξη των δέντρων μέχρι το πρώτο επίπεδο (max\_depth = 1). **Οι καμπύλες που ακολουθούν στην συνέχεια είναι αποτελέσματα του AdaBoost με την υλοποίηση της κλάσης DecisionStump και όχι με την χρήση του DecisionTreeClassifier της Scikit-Learn.**

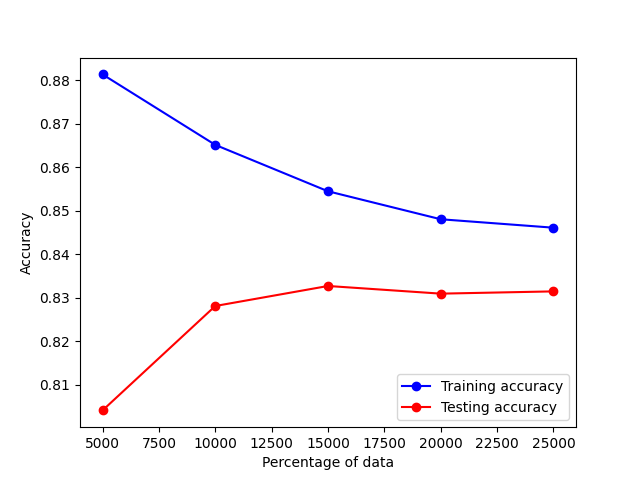
## Πειραματικά αποτελέσματα **Νaive Bayes**

Για τα αποτελέσματα εισήχθησαν οι εξής τιμες στις υπερπαραμέτρους:

m = 6000, n = 300, k = 9000

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, γραμμή, γράφημα

Περιγραφή που δημιουργήθηκε αυτόματα



Για την παρακάτω καμπύλη και τον πίνακα χρησιμοποιήθηκαν τα **test** δεδομένα

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, γραμμή, διάγραμμα

Περιγραφή που δημιουργήθηκε αυτόματα

   precision    recall  f1-score   support

           0       0.84      0.82      0.83     12500

           1       0.82      0.84      0.83     12500

    accuracy                           0.83     25000

   macro avg       0.83      0.83      0.83     25000

weighted avg       0.83      0.83      0.83     25000

## Πειραματικά αποτελέσματα **AdaBoost**

Για τα παρακάτω αποτελέσματα εισήχθησαν οι παρακάτων τιμές στις υπερπαραμέτρους :

**Estimators = 50, m = 700, n = 30, k = 15000**

**\*Χρησιμοποιήθηκαν 50 estimators για οικονομία χρόνου στις δοκιμές**

Καμπύλη μάθησης για τις ανωτέρω παραμέτρους του AdaBoost.

**Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, γραμμή, διάγραμμα

Περιγραφή που δημιουργήθηκε αυτόματα**

Η παρακάτω καμπύλη και ο πίνακας ειναι αποτελέσματα χρήσης των **TEST** δεδομένων

Εικόνα που περιέχει κείμενο, γραμμή, στιγμιότυπο οθόνης, διάγραμμα

Περιγραφή που δημιουργήθηκε αυτόματα

              precision    recall  f1-score   support

           0       0.74      0.75      0.75     12500

           1       0.75      0.74      0.74     12500

    accuracy                           0.75     25000

   macro avg       0.75      0.75      0.75     25000

weighted avg       0.75      0.75      0.75     25000

## Μερος ΄Β

## Συγκρίσεις με έτοιμες υλοποιήσεις από skLearn

## **Naive Bayes**

Learning curve του **GaussianNB**

Εικόνα που περιέχει κείμενο, γραμμή, διάγραμμα, στιγμιότυπο οθόνης

Περιγραφή που δημιουργήθηκε αυτόματα

\*στις συγκρίσεις αυτές έχουν χρησιμοποιηθεί οι ίδιες υπερπαράμετροι

Classification\_report του GaussianNB

              precision    recall  f1-score   support

           0       0.76      0.84      0.80     12500

           1       0.82      0.74      0.78     12500

    accuracy                           0.79     25000

   macro avg       0.79      0.79      0.79     25000

weighted avg       0.79      0.79      0.79     25000

## **AdaBoost**

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, διάγραμμα, γραμμή

Περιγραφή που δημιουργήθηκε αυτόματαLearning curve του AdaBoostClassifier της SkLearn

              precision    recall  f1-score   support

           0       0.79      0.74      0.77     12500

           1       0.76      0.80      0.78     12500

    accuracy                           0.77     25000

   macro avg       0.78      0.77      0.77     25000

weighted avg       0.78      0.77      0.77     25000

## **ΜΕΡΟΣ ΄Γ**

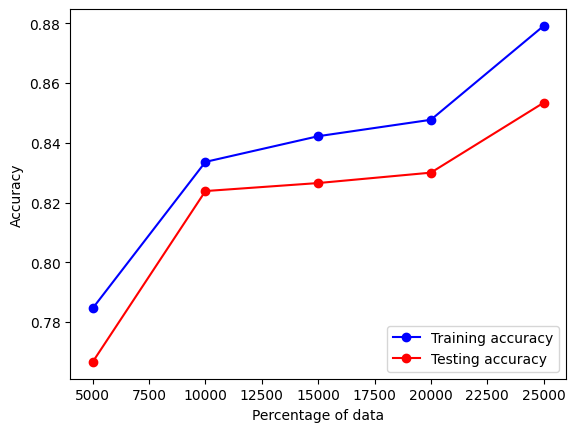
Για το ΄Γ μέρος έγινε μια υλοποίηση ενός RNN με GRU cell ( όπως στον κώδικα του εργαστηρίου). Παραχθήκαν ίδιες καμπύλες όπως και στα προηγούμενα ερωτήματα καθώς επίσης και μια καμπύλη για το loss/epoch για τα train και validation δεδομένα.

**Validation Loss Curve**

Για το παρακάτω γράφημα χρησιμοποίηθηκε η τιμή **« validation\_split = 0.2»**

Εικόνα που περιέχει κείμενο, γράφημα, διάγραμμα, γραμμή

Περιγραφή που δημιουργήθηκε αυτόματα

**Learning Curve**

\*Για την καμπύλη μάθησης αλλα και την καμπύλη των precision, f1-score, recall ο αλγόριθμος έκανε fit με 5 epochs για οικονομία χρόνου

**Classification report curve**

Εικόνα που περιέχει κείμενο, στιγμιότυπο οθόνης, γραμμή, γράφημα

Περιγραφή που δημιουργήθηκε αυτόματα

precision recall f1-score support

0 0.90 0.80 0.84 12500

1 0.82 0.91 0.86 12500

accuracy 0.85 25000

macro avg 0.86 0.85 0.85 25000

weighted avg 0.86 0.85 0.85 25000

## **Συμπεράσματα συγκρίσεων.**

Το Rnn είναι το ακριβέστερο σε σχέση με τα υπόλοιπα μοντέλα μηχανικής μάθησης που υλοποιήθηκαν καθώς έχει υψηλότερη ακρίβεια στις προβλέψεις σε test δεδομένα κατά τουλάχιστον 3-5% σε σχέση με τον Naive Bayes .

Μεταξύ του Naïve Bayes παρατηρούμε ότι υπάρχει καλύτερο accuracy στον custom αλγόριθμο όμως υπάρχει και μεγαλύτερο overfitting στην καμπύλη μάθησης σε σχέση με την υλοποίηση της skLearn.

Ο adaBoost δεν παρουσιάζει υψηλή ακρίβεια σε καμία από τις δύο υλοποιήσεις. Ωστόσο η υλοποίηση της skLearn παρουσίαζει μεγαλύτερη ακρίβεια στις προβλέψεις.