ΤΜΗΜΑ ΠΛΗΡΟΦΟΡΙΚΗΣ

ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ
ΧΕΙΜΕΡΙΝΟ ΕΞΑΜΗΝΟ 2023-2024

Ελένη Κεχριώτη
Μαρία Σχοινάκη
Χρήστος Σταμούλος

Μηχανική Μάθηση

Εισαγωγή

Η παρούσα εργασία με θέμα «**Ανάλυση συναισθήματος πάνω σε κριτικές ταινιών**» έχει σκοπό να κατατάσσει κριτικές ταινιών σε αρνητικές ή θετικές χρησιμοποιώντας αλγορίθμους μηχανικής μάθησης και νευρωνικών δικτύων.

Τα δεδομένα για τις κριτικές τα αντλήσαμε με την βοήθεια του **IMDB dataset** του Keras.

Οι αλγόριθμοι **μηχανικής μάθησης** που επιλέξαμε να υλοποιήσουμε είναι οι:

- **Random Forest**
- **Naive Bayes**
- **Logistic Regression**

Το **νευρωνικό δίκτυο** που επιλέξαμε να υλοποιήσουμε είναι το:

- **MLP**

Μέρος A&B

<u>Προεπεξεργασία δεδομένων</u>
Με την χρήση του βοηθητικού κώδικα του φροντιστηρίου, και την βοήθεια της συνάρτησης *tf.keras.datasets.imdb.load_data,* αντλήσαμε τα δεδομένα των κριτικών και τα χωρίσαμε σε **δεδομένα εκπαίδευσης**(*train*) και **δεδομένα ελέγχου**(*test*). Τα δεδομένα αυτά βέβαια, περιείχαν μόνο τις **m** συχνότερες λέξεις, πλην τις **n** πιο συχνές και **k** πιο σπάνιες λέξεις από αυτές. Αυτό επετεύχθη, βάζοντας σαν ορίσματα στην συνάρτηση τα, m-k για το num_words και το n για το skip_top. Τα **m, n, k** είναι **υπερπαράμετροι** , τις οποίες ορίσαμε εμείς για κάθε διαφορετικό αλγόριθμο, έτσι ώστε να επιτύχουμε το βέλτιστο αποτέλεσμα. Οι κριτικές των ταινιών, αποθηκεύονται στους πίνακες **x_train, x_test** και η κατηγορία στην οποία ανήκει κάθε κριτική ταινίας στους πίνακες **y_train, y_test**, με καθέναν από αυτούς τους πίνακες να είναι μεγέθους **25.000**. Οι πίνακες **y_train, y_test** περιέχουν **0** και **1** με το **1** να αναπαριστά την **θετική κριτική** και το **0** την **αρνητική κριτική**. Έπειτα, τροποποιήσαμε τους πίνακες **x_train, x_test**, έτσι ώστε αντί να περιέχουν μία λίστα με αριθμούς για κάθε κριτική, (*εκ των οποίων κάθε αριθμός αντιστοιχεί σε μία λέξη*), να περιέχουν συμβολοσειρές λέξεων.

Τέλος, για να είναι τα δεδομένα μας πιο μορφοποιημένα σύμφωνα με την προγραμματιστική λογική, τα μετατρέψαμε σε δυαδικά δεδομένα. Δηλαδή οι πίνακες **x_train, x_test**, πέρασαν από έναν *binary CountVectorizer*, ώστε να μετατραπούν σε **δυαδικά διανύσματα**. Κάθε κριτική αντιστοιχεί σε ένα διάνυσμα(*25.000 διανύσματα για κάθε πίνακα*). Κάθε διάνυσμα έχει πλήθος τιμών, όσο το μέγεθος του λεξιλογίου μας και κάθε τιμή του διανύσματος, αν είναι 1 σημαίνει ότι η συγκεκριμένη λέξη του λεξιλογίου εμφανίζεται στην τρέχουσα κριτική, ενώ αν είναι 0, σημαίνει, ότι δεν εμφανίζεται. Καταλήγουμε στα **(x_train_binary, y_train)** και **(x_test_binary, y_test)**.

$$X = \begin{bmatrix} \vec{x_1} \\ \vdots \\ \vec{x_t} \end{bmatrix} \quad y = \begin{bmatrix} y_1 \\ \vdots \\ y_t \end{bmatrix}$$

Information Gain

Για τους αλγορίθμους **Naive Bayes** και **Logistic Regression**, υλοποιήθηκε, με την χρήση των *βοηθημάτων των φροντιστηρίων* και αλγόριθμος **Information Gain**, ο οποίος εξετάζει το κέρδος πληροφορίας, αλλά αποφασίσαμε να μην τον χρησιμοποιήσουμε καθώς κατανάλωνε πολύ υπολογιστικό κόστος, ενώ δεν προσέφερε ικανοποιητικά αποτελέσματα. Συγκεκριμένα, η διαφορά με την χρήση του και μη χρήση του, ήταν απειροελάχιστη, οπότε δεν συνέφερε εν τέλη να χρησιμοποιηθεί.

Σημείωση: Σε όλη την έκταση της εργασίας, χρησιμοποιήθηκε ως επί το πλείστων, η βιβλιοθήκη **numpy** της python, η οποία επιτάχυνε την διαδικασία καθώς μείωνε τον υπολογιστικό φόρτο των προγραμμάτων μας.

Naive Bayes

Υλοποιούμε την **Bernoulli** μορφή του αλγορίθμου **Naive Bayes** με εκτιμήτρια **Laplace** για την αποφυγή μηδενικών πιθανοτήτων και χρησιμοποιούμε λογαριθμικές πιθανότητες έτσι ώστε να εξαλείψουμε το σενάριο σφαλμάτων λόγω υπερβολικά μικρών πολλαπλασιασμών.

Μέθοδοι της κλάσης BernoulliNaiveBayes

Fit

Είναι η μέθοδος με την οποία εκπαιδεύουμε τον αλγόριθμο πάνω σε ένα σύνολο δεδομένων εκπαίδευσης (x) με τις ορθές αποκρίσεις τους (y).
Ας αναλύσουμε περαιτέρω.
Αρχικά, υπολογίζουμε τις a-priori πιθανότητες κάθε κλάσης και τις λογαριθμίζουμε. Έπειτα, υπολογίζουμε την πιθανότητα να είναι παρών ένα χαρακτηριστικό (δηλαδή το διάνυσμα χαρακτηριστικών να έχει τιμή 1 για το συγκεκριμένο) δεδομένου ότι μια κριτική ανήκει σε μία κατηγορία. Η συγκεκριμένη πληροφορία αποθηκεύεται σε δύο μεταβλητές την positive_1 και την negative_1, με τις πιθανότητες να είναι φυσικά λογαριθμισμένες. Αντίστοιχα, σε δύο άλλες μεταβλητές, positive_0 και negative_0, υπολογίζουμε την λογαριθμισμένη πιθανότητα **μη** εμφάνισης ενός χαρακτηριστικού (δηλαδή το διάνυσμα χαρακτηριστικών να έχει τιμή 0 για το συγκεκριμένο), δεδομένου ότι ένα παράδειγμα χωρίς αυτό το χαρακτηριστικό ανήκει στην κάθε κλάση.
Οι πιθανότητες που αναφέρθηκαν υπολογίζονται ως εξής:

1. Αθροίζουμε κάθε στήλη του πίνακα με τα παραδείγματα εκπαίδευσης και καταλήγουμε με ένα διάνυσμα που περιέχει το πόσες φορές εμφανίστηκε κάθε χαρακτηριστικό σε μία συγκεκριμένη κατηγορία.
2. Μετατρέπουμε κάθε τιμή του διανύσματος από άθροισμα σε πιθανότητα εμφάνισης του κάθε χαρακτηριστικού δεδομένης μίας κλάσης, διαιρώντας το αποτέλεσμα του βήματος 1 με τη συνολική ποσότητα της κλάσης εφαρμόζοντας Laplace.

Predict

Είναι η μέθοδος που προβλέπει/ κατηγοριοποιεί ένα σύνολο δεδομένων ανάπτυξης που δίνονται (x).

Σε δύο μεταβλητές positive και negative, υπολογίζουμε για κάθε κριτική την πιθανότητα να είναι θετική και αρνητική, δεδομένου των χαρακτηριστικών της.

Για τον υπολογισμό αυτό, χρειαζόμαστε έναν βοηθητικό πίνακα x_reverse, ο οποίος είναι αντίθετος του πίνακα x, δηλαδή έχει 1 όπου ο x έχει 0 και το αντίστροφο. Θα μας χρειαστεί για το εσωτερικό γινόμενο.
Πιο συγκεκριμένα, για την θετική κατηγορία υπολογίζουμε το dot product (εσωτερικό γινόμενο) με τον πίνακα x_reverse και τον πίνακα positive_0, καθώς και του πίνακα x με τον positive_1 και τους προσθέτουμε, μαζί με την a-priori πιθανότητα της θετικής κατηγορίας. Αυτό το κάνουμε γιατί, η πιθανότητα ένα κείμενο να ανήκει στα θετικά, δεδομένου των χαρακτηριστικών θα είναι το άθροισμα των πιθανοτήτων να ανήκει στη θετικά με βάση την ύπαρξη ή όχι κάθε χαρακτηριστικού, καθώς και η γενικότερη πιθανότητα να ανήκει η κριτική στα θετικά χωρίς την σκέψη των χαρακτηριστικών. Αντίστοιχα και για την αρνητική κατηγορία.
Τέλος, αποφασίζουμε αν το κείμενο ανήκει στη θετική ή αρνητική κατηγορία ανάλογα με το ποιά πιθανότητα είναι μεγαλύτερη.

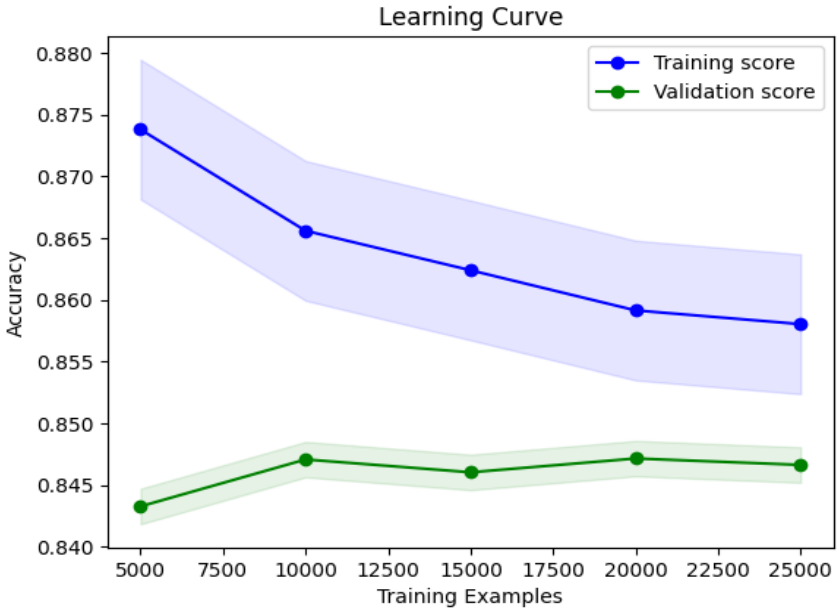
Αποτελέσματα, Συγκρίσεις και Συμπεράσματα

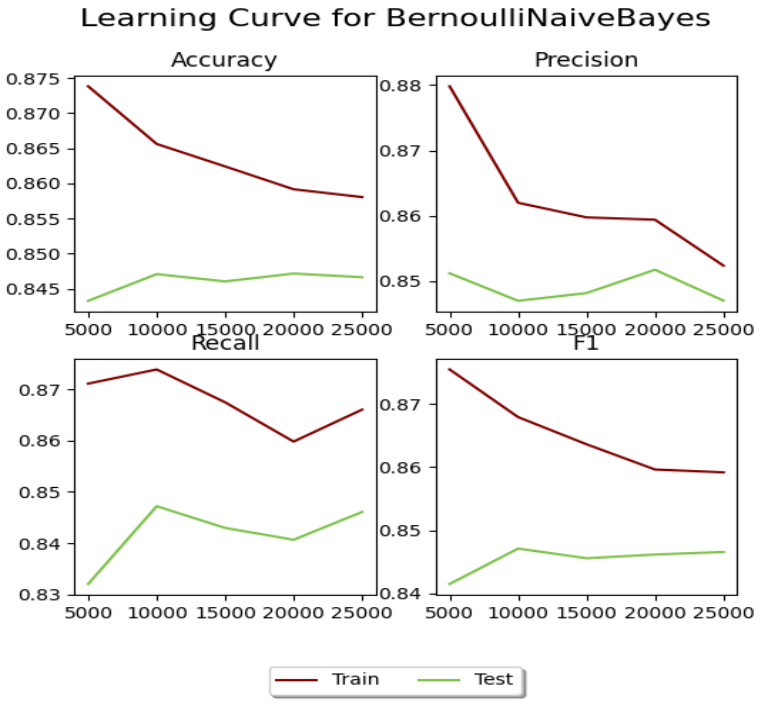
Τα αποτελέσματα που αναλύουμε έγιναν από πειράματα που είχαν τις εξής υπερπαραμέτρους **m=3000, n=50, k=80, IG=false**. Οι υπερπαράμετροι αυτοί βρέθηκαν, μετά απο πειραματισμό με διάφορους συνδυασμούς τιμών και επιλογή αυτού του συνδυασμού με τη μεγαλύτερη ακρίβεια.
Όπως φαίνεται και στα διαγράμματα με τα αποτελέσματα του αλγορίθμου **Naive Bayes**, η ακρίβεια στα train δεδομένα και στα test δεδομένα είναι πολύ κοντά. Όσο αυξάνουμε το σύνολο δεδομένων εκπαίδευσης, τόσο μειώνεται η επίδοση του αλγορίθμου στα train δεδομένα, ενώ αυξάνεται για τα test. Άρα ο αλγόριθμός μας όντως **μαθαίνει και εκπαιδεύεται**, εφόσον γίνεται αισθητή πλέον η έννοια της "εμπειρίας" και μάλιστα **ανταποκρίνεται** ιδιαίτερα καλά στα δεδομένα αξιολόγησης. Εκπαιδεύοντας και αξιολογώντας τον αλγόριθμο της βιβλιοθήκης SkLearn, και αφού τον συγκρίναμε με τον δικό μας αλγόριθμο, παρατηρήσαμε(*όπως φαίνεται και παρακάτω στα διαγράμματα*) ότι τα ποσοστά βρίσκονται αρκετά κοντά. Συγκεκριμένα, και από τα διαγράμματα καθώς και από τους πίνακες(*ή/και καλύτερα τον πίνακα διαφοράς*), φαίνεται ότι οι προσεγγίσεις είναι σχεδόν πανομοιότυπες.

Πίνακας Διαφοράς για τον BernoulliNaiveBayes και τον SKLearn's BernoulliNB

	Train Accuracy	Test Accuracy	Precision Train	Precision Test	Recall Train	Recall Test	F1 Train	F1 Test
5000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
10000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
15000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
20000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000

Learning, F1, Precision, Recall, F1 Curves for Naive Bayes

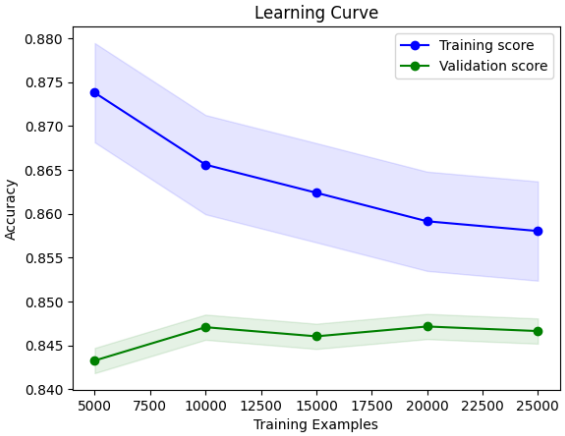


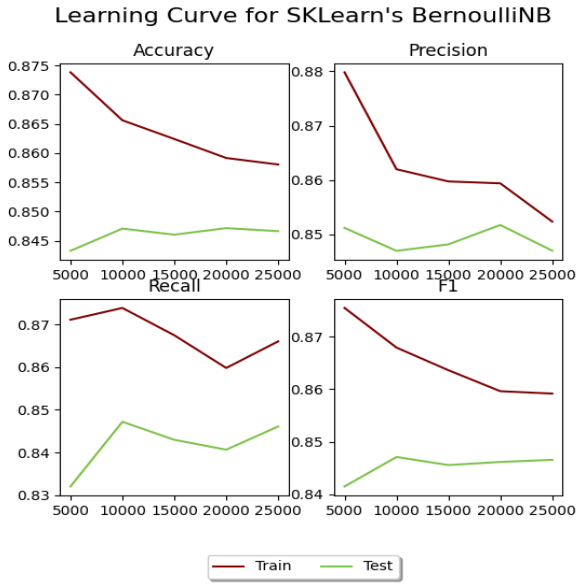


	Train Accuracy	Test Accuracy	Precision Train	Precision Test	Recall Train	Recall Test	F1 Train	F1 Test
5000	0.87	0.84	0.88	0.85	0.87	0.83	0.88	0.84
10000	0.87	0.85	0.86	0.85	0.87	0.85	0.87	0.85
15000	0.86	0.85	0.86	0.85	0.87	0.84	0.86	0.85
20000	0.86	0.85	0.86	0.85	0.86	0.84	0.86	0.85
25000	0.86	0.85	0.85	0.85	0.87	0.85	0.86	0.85

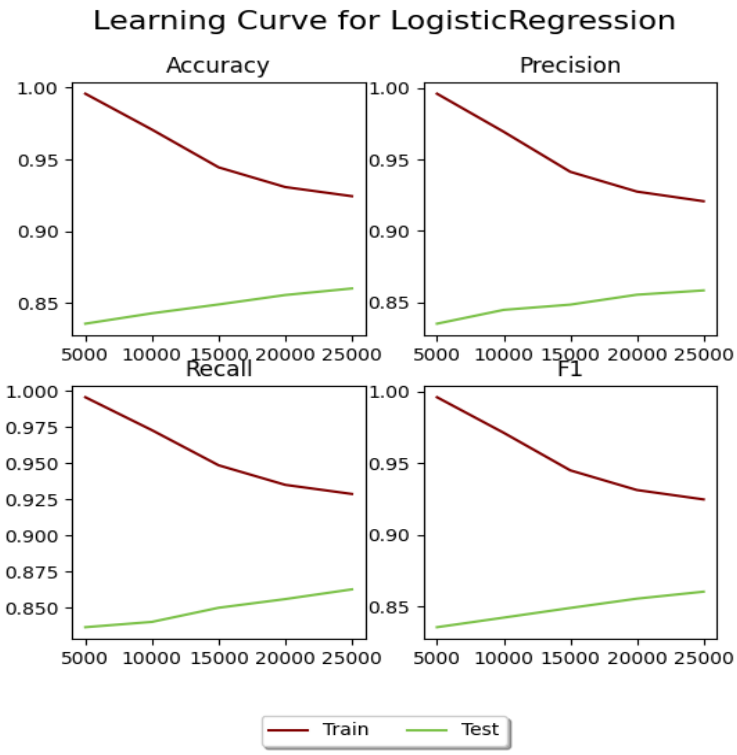
[6]

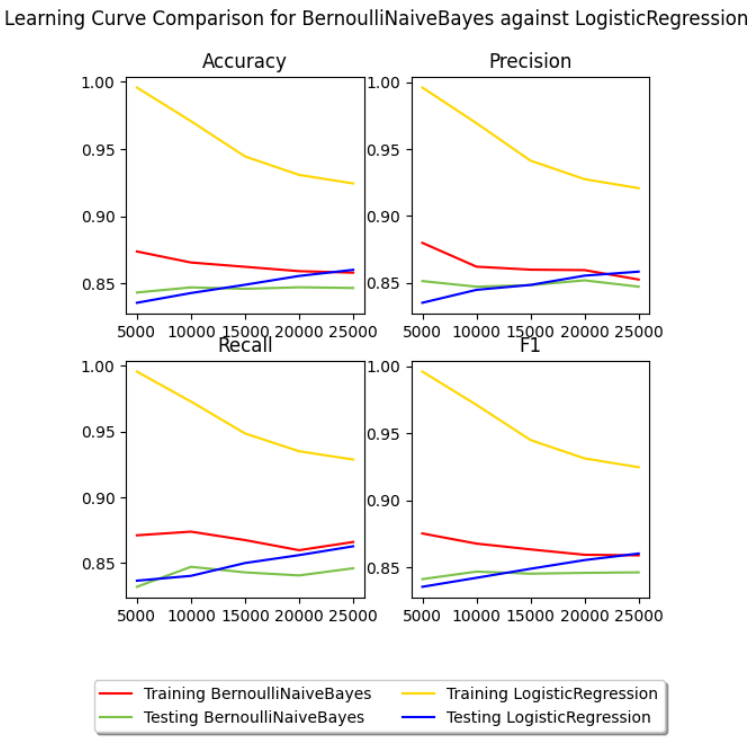
Learning, F1, Precision, Recall Curves for SkLearn's Naive Bayes





Learning, F1, Precision, Recall Curves for SkLearn's Logistic Regression





Πίνακας Διαφοράς για τον BernoulliNaiveBayes και τον LogisticRegression

	Train Accuracy	Test Accuracy	Precision Train	Precision Test	Recall Train	Recall Test	F1 Train	F1 Test
5000	0.130000	0.000000	0.120000	0.020000	0.130000	0.010000	0.120000	0.000000
10000	0.100000	0.010000	0.110000	0.010000	0.100000	0.010000	0.100000	0.010000
15000	0.080000	0.000000	0.080000	0.000000	0.080000	0.010000	0.080000	0.000000
20000	0.070000	0.010000	0.070000	0.010000	0.080000	0.020000	0.070000	0.010000
25000	0.060000	0.010000	0.070000	0.010000	0.060000	0.010000	0.060000	0.010000

Logistic Regression

Υλοποιούμε τον Logistic Regression με lasso ομαλοποίηση για την αποφυγή της υπερπροσαρμογής στα δεδομένα εκπαίδευσης και στοχαστική ανάβαση κλίσης. Παρακάτω αναλύουμε τις μεθόδους της κλάσης LogisticRegression

fit:

Είναι η μέθοδος με την οποία εκπαιδεύουμε τον αλγόριθμο πάνω σε ένα σύνολο δεδομένων εκπαίδευσης (x) με τις ορθές αποκρίσεις τους (y).
Αρχικοποιούμε τα βάρη των χαρακτηριστικών με 0 και ξεκινάμε την εκπαίδευση του αλγορίθμου η οποία πραγματοποιείται σε 3 επίπεδα τα οποία διατρέχονται επαναληπτικά για το δοσμένο μέγιστο αριθμό επαναλήψεων(max_iter).
Συγκεκριμένα τα εξής επίπεδα:
Υπολογίζουμε το εσωτερικό γινόμενο των βαρών και του πίνακα X και με βάση αυτό υπολογίζουμε τις προβλεπόμενες αποκρίσεις y_pred μέσω της σιγμοειδούς συνάρτηση.
Υλοποιούμε τη στοχαστική ανάβαση κλίσης, έχοντας στο νού έναν νοητό λόφο, με σκοπό ανεβαίνοντας τον να φτασουμε σε ολικά μέγιστα.
Βρίσκουμε το μέσο όρο κάθε διανύσματος του εσωτερικού γινομένου του ανάστροφου πίνακα X και της διαφοράς των αποκρίσεων (y_pred-y).
Ομαλοποιούμε την ανάβαση αφαιρώντας από το (a) το γινόμενο του λ με το άθροισμα των τετραγώνων των βαρών (L2 ομαλοποίηση).
Αθροίζουμε στα βάρη τη στοχαστική ανάβαση κλίσης πολλαπλασιασμένη με το η, έτσι ώστε να προσεγγίσουμε τη σύγκλιση με ομοιόμορφο ρυθμό.

predict:

Είναι η μέθοδος που προβλέπει/ κατηγοριοποιεί ένα σύνολο δεδομένων ανάπτυξης που δίνονται (x).
Η πρόβλεψη υλοποιείται ως εξής
Υπολογίζουμε το εσωτερικό γινόμενο των δεδομένων που δόθηκαν για πρόβλεψη με τα βάρη των χαρακτηριστικών που αποθηκεύσαμε μετά την εκπαίδευση. Ύστερα, βρίσκουμε τις προβλεπόμενες αποκρίσεις μέσω της σιγμοειδούς συνάρτησης και αποφασίζουμε αν η τελική πρόβλεψη ανήκει στη θετική ή αρνητική κατηγορία με βάση το αν η σιγμοειδής τιμή είναι μεγαλύτερη ή μικρότερη το σταθερού κατωφλιού 0.5.
Οι παράμετροι lambda_value , eta και max_iter επιλέχτηκαν κατάλληλα μέσω ενός αλγορίθμου πειραματισμών. Ο αλγόριθμος υλοποιήθηκε με γνώμονα το μέγεθος accuracy πάνω σε δεδομένα ανάπτυξης με διάφορες υποψήφιες τιμές και επιλέχτηκε ο συνδυασμός lambda value: 0.01 , max_iterr: 1000 , eta: 0.001.

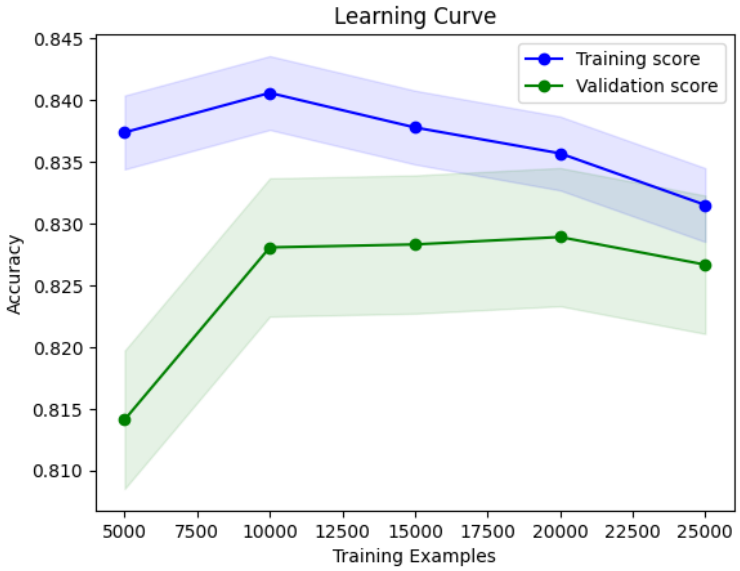
Αποτελέσματα, Συγκρίσεις και Συμπεράσματα

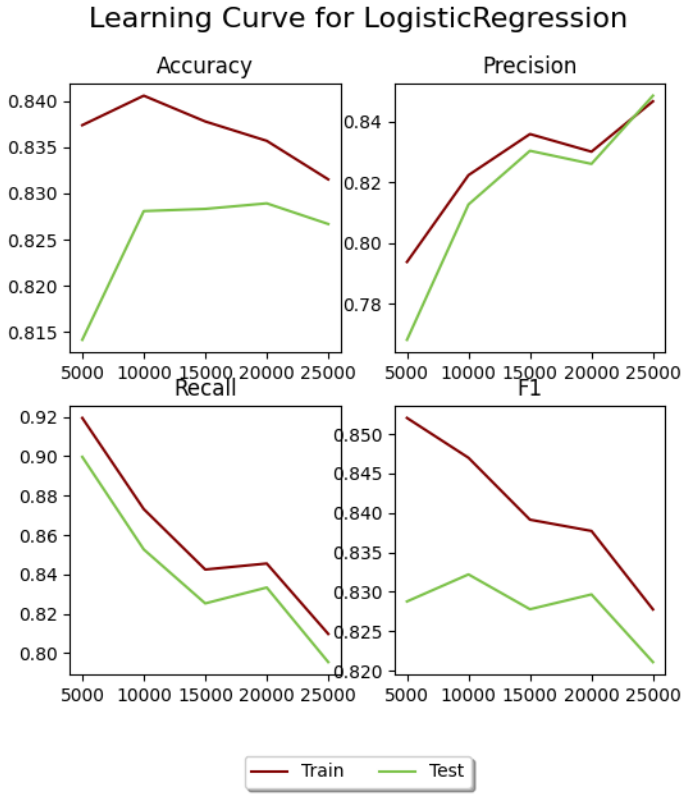
Τα αποτελέσματα που αναλύουμε έγιναν από πειράματα που είχαν τις εξής υπερπαραμέτρους **m=2500, n=200, k=20, IG=false**. Οι υπερπαράμετροι αυτοί βρέθηκαν, μετά απο πειραματισμό με διάφορους συνδυασμούς τιμών και επιλογή αυτού του συνδυασμού με τη μεγαλύτερη ακρίβεια. Όπως φαίνεται και στα διαγράμματα με τα αποτελέσματα του αλγορίθμου **Logistic Regration**, η ακρίβεια στα train δεδομένα και στα test δεδομένα είναι πολύ κοντά. Όσο αυξάνουμε το σύνολο δεδομένων εκπαίδευσης, τόσο αυξάνεται η επίδοση του αλγορίθμου στα test δεδομένα. Άρα ο αλγόριθμός μας όντως μαθαίνει και εκπαιδεύεται, εφόσον γίνεται αισθητή πλέον η έννοια της “εμπειρίας” και μάλιστα ανταποκρίνεται ιδιαίτερα καλά στα δεδομένα αξιολόγησης. Εκπαιδεύοντας και αξιολογώντας τον αλγόριθμο της βιβλιοθήκης SkLearn, και αφού τον συγκρίναμε με τον δικό μας αλγόριθμο, παρατηρήσαμε(όπως φαίνεται και παρακάτω στα διαγράμματα) ότι τα ποσοστά βρίσκονται αρκετά κοντά. Συγκεκριμένα, και από τα διαγράμματα καθώς και από τους πίνακες(ή/και καλύτερα τον πίνακα διαφοράς), φαίνεται ότι οι προσεγγίσεις είναι σχεδόν πανομοιότυπες.

Classification Table Difference for LogisticRegression against SKLogisticRegression

	Train Accuracy	Test Accuracy	Precision Train	Precision Test	Recall Train	Recall Test	F1 Train	F1 Test
5000	0.170000	0.030000	0.230000	0.090000	0.040000	0.100000	0.150000	0.000000
10000	0.120000	0.010000	0.150000	0.030000	0.070000	0.030000	0.110000	0.000000
15000	0.090000	0.010000	0.100000	0.020000	0.090000	0.000000	0.090000	0.010000
20000	0.080000	0.020000	0.100000	0.030000	0.070000	0.000000	0.080000	0.020000
25000	0.080000	0.020000	0.060000	0.010000	0.110000	0.060000	0.080000	0.030000

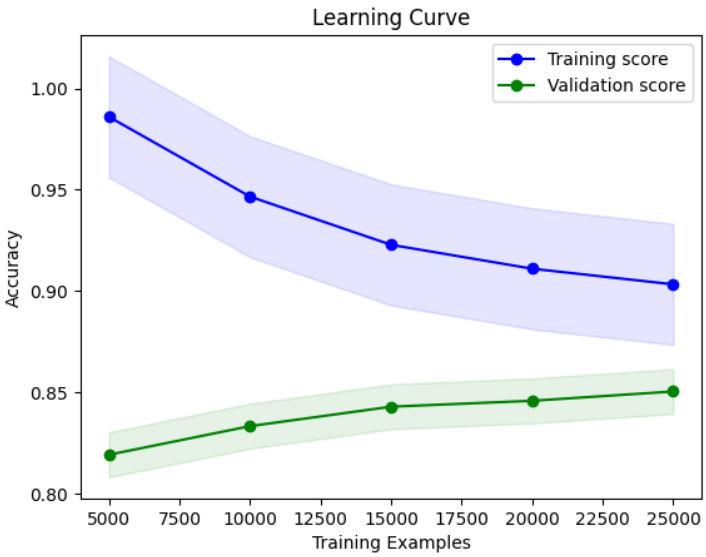
Learning, F1, Precision, Recall, F1 Curves for Logistic Regression

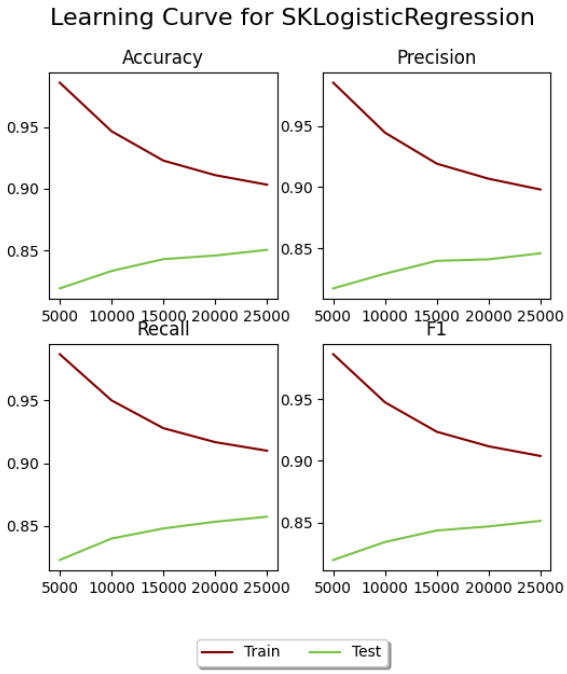


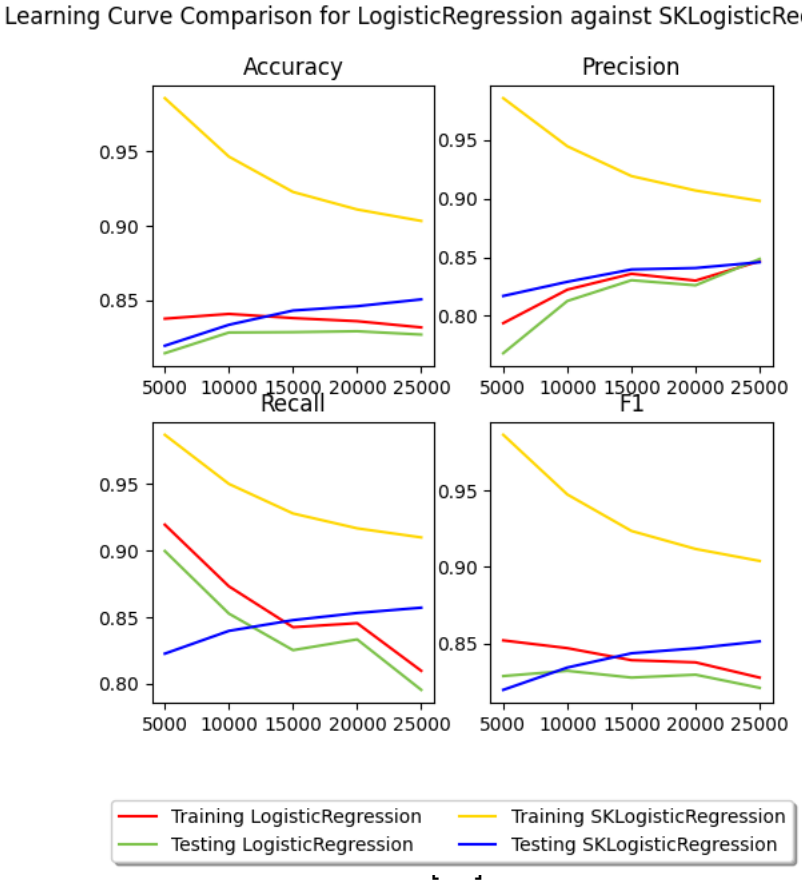


	Train Accuracy	Test Accuracy	Precision Train	Precision Test	Recall Train	Recall Test	F1 Train	F1 Test
5000	0.84	0.81	0.79	0.77	0.92	0.90	0.85	0.83
10000	0.84	0.83	0.82	0.81	0.87	0.85	0.85	0.83
15000	0.84	0.83	0.84	0.83	0.84	0.83	0.84	0.83
20000	0.84	0.83	0.83	0.83	0.85	0.83	0.84	0.83
25000	0.83	0.83	0.85	0.85	0.81	0.80	0.83	0.82

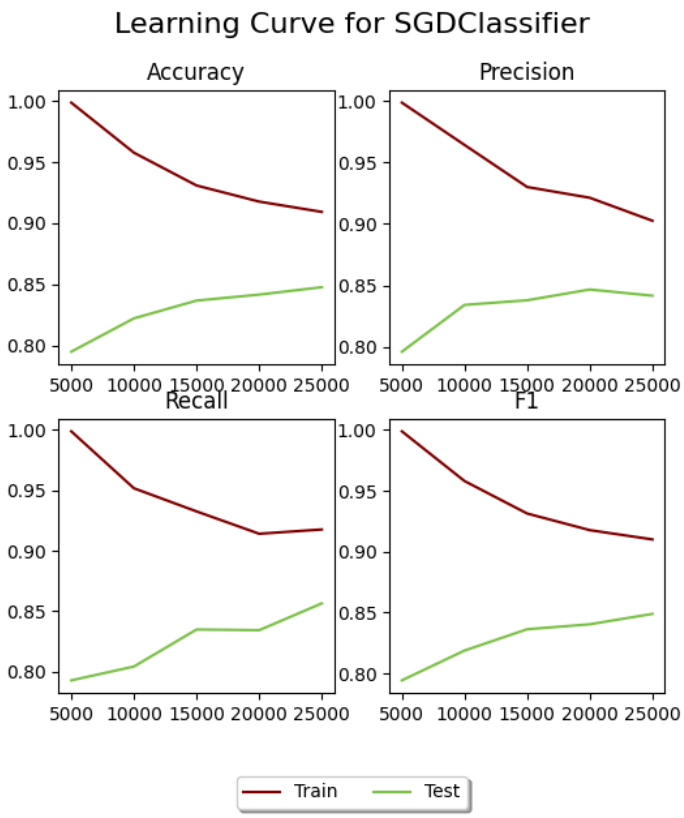
Learning, F1, Precision, Recall Curves for SkLearn's Logistic Regression

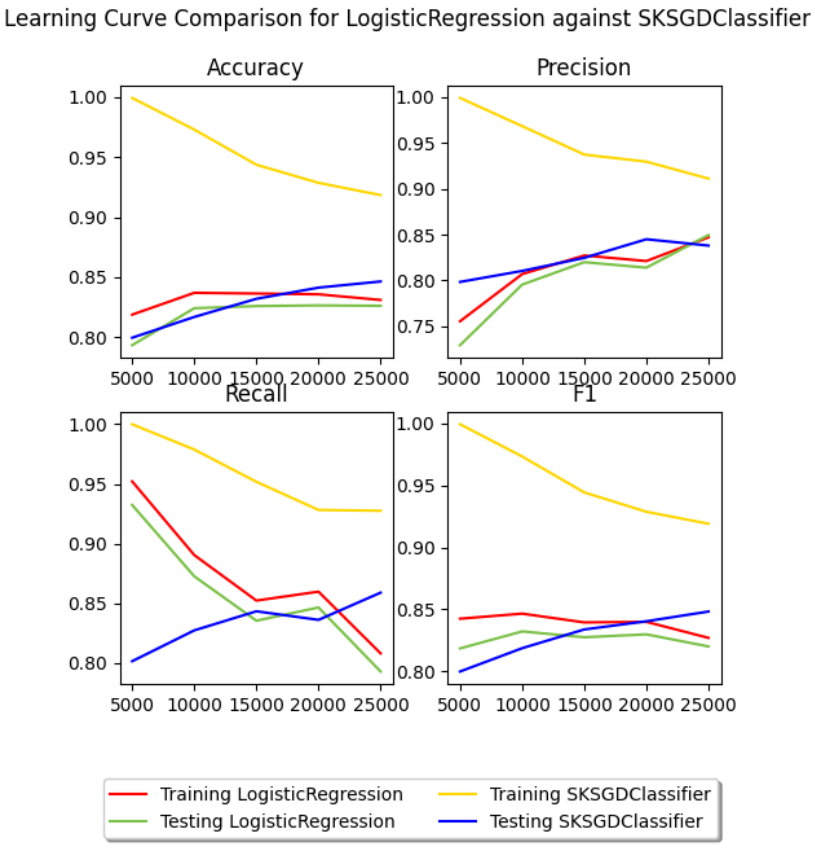






Learning, F1, Precision, Recall Curves for SkLearn's SGDClassifier





Classification Table Difference for LogisticRegression against SGDClassifier

	Train Accuracy	Test Accuracy	Precision Train	Precision Test	Recall Train	Recall Test	F1 Train	F1 Test
5000	0.180000	0.010000	0.240000	0.080000	0.050000	0.140000	0.160000	0.020000
10000	0.130000	0.000000	0.170000	0.020000	0.080000	0.060000	0.120000	0.020000
15000	0.100000	0.000000	0.110000	0.010000	0.100000	0.000000	0.100000	0.000000
20000	0.090000	0.010000	0.110000	0.030000	0.070000	0.010000	0.090000	0.010000
25000	0.090000	0.020000	0.060000	0.020000	0.120000	0.080000	0.090000	0.030000

[13]

Random Forest

Υλοποιήσαμε τον αλγόριθμο **Random Forest**, ο οποίος είναι ένας συλλογικός αλγόριθμος μάθησης που υλοποιεί την έννοια της πλειοψηφίας, μεταξύ number_of_trees δέντρα ID3.

Μέθοδοι του Random Forest:

Fit

Είναι η μέθοδος στην οποία **εκπαιδεύεται** το αλγόριθμος. Ξεκινάει, τρέχοντας μια επανάληψη(*μια για κάθε δέντρο*). Σε κάθε επανάληψη, δημιουργεί ένα καινούριο δέντρο **ID3** και το εκπαιδεύει με τυχαία παραδείγματα εκπαίδευσης με επανατοποθέτηση και τυχαίο σύνολο με ιδιότητες. Για την δημιουργία αυτών των τυχαίων συνόλων, υλοποιήθηκαν 2 νέες μέθοδοι.

Select_random_samples

Αυτή η μέθοδος, δέχεται το αρχικό dataset training δεδομένων και παράγει ένα καινούριο ίδιου μεγέθους με τυχαία παραδείγματα εκπαίδευσης, με πιθανά διπλότυπα. Τροποποιεί και τον πίνακα y για να έχει αντίστοιχα δεδομένα και αντιστοίχιση.

Select_random_features

Αυτή η μέθοδος αντίστοιχα, δέχεται αντίστοιχα το αρχικό data set training δεδομένων και επιλέγει στην τύχη m ιδιότητες(υπερπαραμέτρους) υποσύνολο του αρχικού λεξιλογίου και χωρίς επανατοποθέτηση.

Predict

Η μέθοδος αυτή είναι ο αποφασιστής του αλγορίθμου. Αποφασίζει και κατατάσσει κάθε παράδειγμα εκπαίδευσης στην κατηγορία την οποία πιστεύει ότι ανήκει κάθε κριτική. Για κάθε δέντρο αρχικά, καλεί την predict του δέντρου, δίνοντας το x test dataset. Η predict του ID3, επιστρέφει ένα σύνολο με 0, 1 που είναι οι αποκρίσεις του δέντρου σε κάθε παράδειγμα αξιολόγησης. Έπειτα η predict του random forest, τρέχει για κάθε παράδειγμα αξιολόγησης, κάθε απόκριση πρόβλεψης του id3 για το συγκεκριμένο παράδειγμα. Αν έχει τιμή 1 ή 0 αυξάνεται ο αντίστοιχος μετρητής για 0 και 1 του συγκεκριμένου παραδείγματος. Μόλις εξεταστούν όλες οι predict των id3 για κάποιο παράδειγμα, βγαίνει αν το παράδειγμα αυτό είχε περισσότερες αποκρίσεις 0 ή 1 από τα δέντρα και άρα αποθηκεύεται στον τελικό πίνακα επιστροφής του random forest, η τιμή του. Άρα κάθε παράδειγμα εκπαίδευσης έχει την πλειοψηφική απόκριση των τυχαίων δέντρων id3 που δημιούργησε ο random forest.

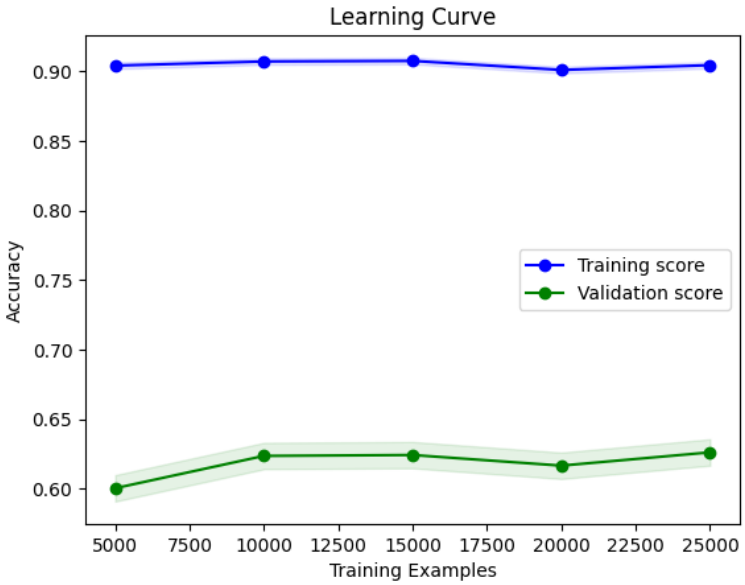
Αποτελέσματα, Συγκρίσεις και Συμπεράσματα

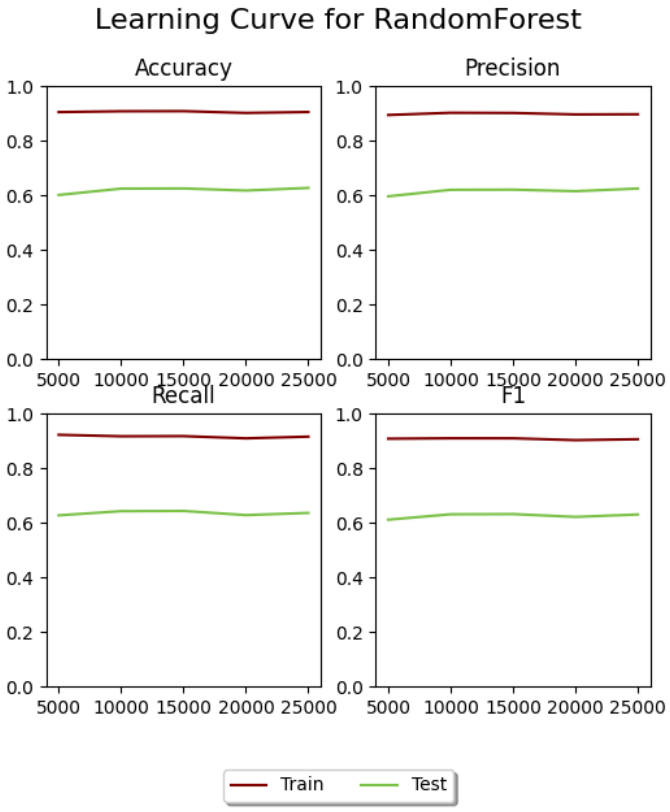
Τα αποτελέσματα που αναλύουμε έγιναν από πειράματα που είχαν τις εξής υπερπαραμέτρους **m=500, n=50, k=0**. Οι υπερπαράμετροι αυτοί βρέθηκαν, μετά απο πειραματισμό με διάφορους συνδυασμούς τιμών και επιλογή αυτού του συνδυασμού με τη μεγαλύτερη ακρίβεια. Όπως φαίνεται και στα διαγράμματα με τα αποτελέσματα του αλγορίθμου **Random Forest**, η ακρίβεια στα train δεδομένα και στα test δεδομένα είναι πολύ κοντά. Όσο αυξάνουμε το σύνολο δεδομένων εκπαίδευσης, τόσο αυξάνεται η επίδοση του αλγορίθμου στα test δεδομένα. Άρα ο αλγόριθμός μας όντως μαθαίνει και εκπαιδεύεται, εφόσον γίνεται αισθητή πλέον η έννοια της “εμπειρίας” και μάλιστα ανταποκρίνεται ιδιαίτερα καλά στα δεδομένα αξιολόγησης. Εκπαιδεύοντας και αξιολογώντας τον αλγόριθμο της βιβλιοθήκης SkLearn, και αφού τον συγκρίναμε με τον δικό μας αλγόριθμο, παρατηρήσαμε(όπως φαίνεται και παρακάτω στα διαγράμματα) ότι τα ποσοστά βρίσκονται αρκετά κοντά. Συγκεκριμένα, και από τα διαγράμματα καθώς και από τους πίνακες(ή/και καλύτερα τον πίνακα διαφοράς), φαίνεται ότι οι προσεγγίσεις είναι σχεδόν πανομοιότυπες.

Πίνακας Διαφοράς για τον RandomForest και τον SKLearn's Random Forest

	Train Accuracy	Test Accuracy	Precision Train	Precision Test	Recall Train	Recall Test	F1 Train	F1 Test
5000	0.100000	0.110000	0.130000	0.080000	0.030000	0.180000	0.090000	0.130000
10000	0.140000	0.100000	0.180000	0.060000	0.020000	0.200000	0.110000	0.120000
15000	0.140000	0.120000	0.170000	0.080000	0.060000	0.190000	0.120000	0.130000
20000	0.130000	0.120000	0.180000	0.090000	0.030000	0.210000	0.110000	0.140000
25000	0.140000	0.100000	0.180000	0.080000	0.070000	0.170000	0.130000	0.120000

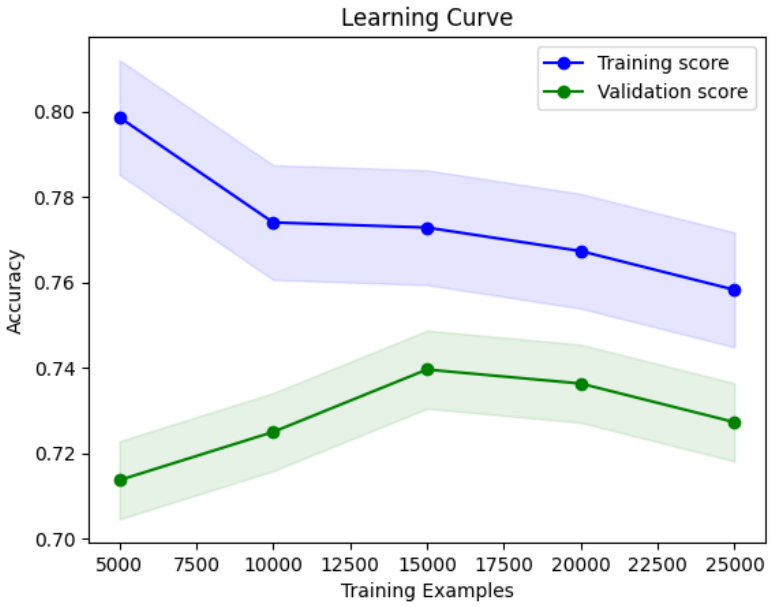
Learning, F1, Precision, Recall, F1 Curves for Random Forest

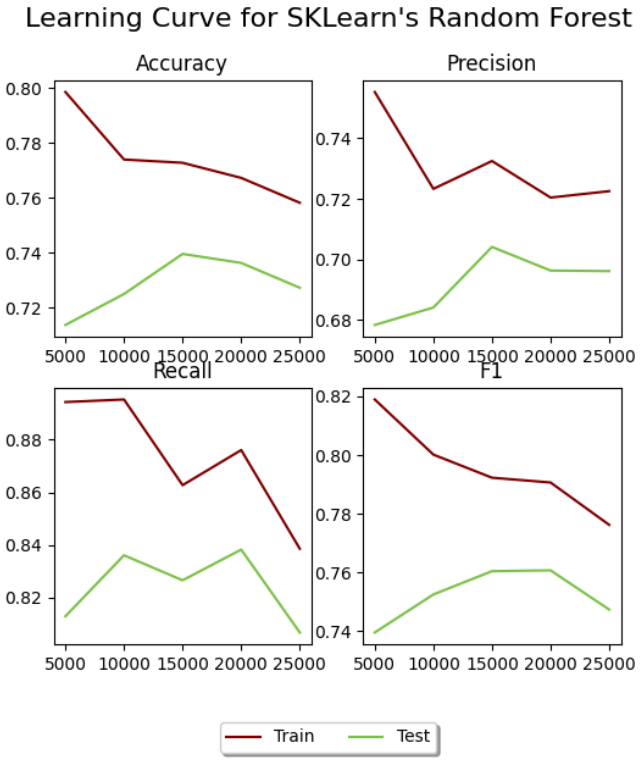


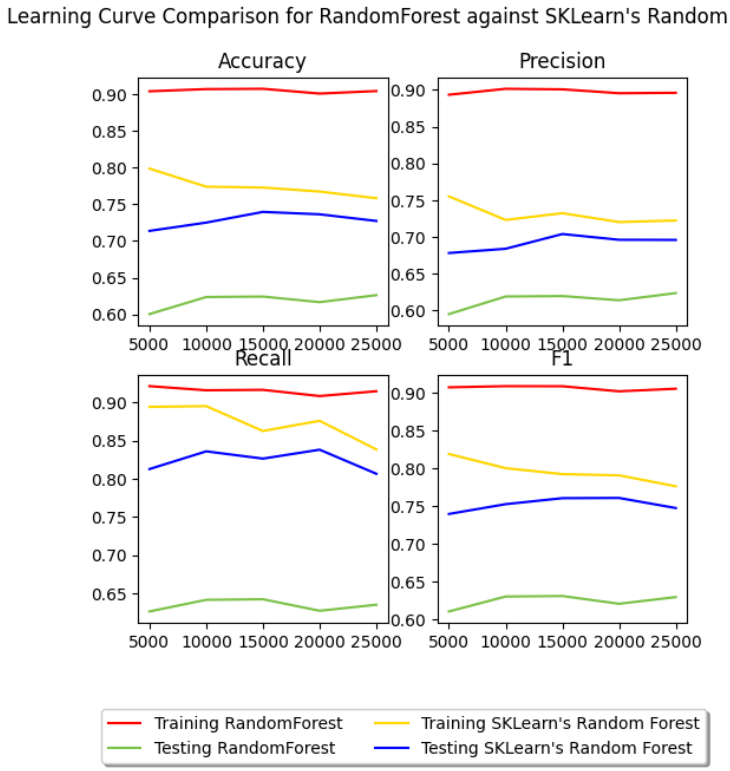


	Train Accuracy	Test Accuracy	Precision Train	Precision Test	Recall Train	Recall Test	F1 Train	F1 Test
5000	0.90	0.60	0.89	0.60	0.92	0.63	0.91	0.61
10000	0.91	0.62	0.90	0.62	0.92	0.64	0.91	0.63
15000	0.91	0.62	0.90	0.62	0.92	0.64	0.91	0.63
20000	0.90	0.62	0.90	0.61	0.91	0.63	0.90	0.62
25000	0.90	0.63	0.90	0.62	0.91	0.64	0.91	0.63

Learning, F1, Precision, Recall Curves for SkLearn's Random Forest

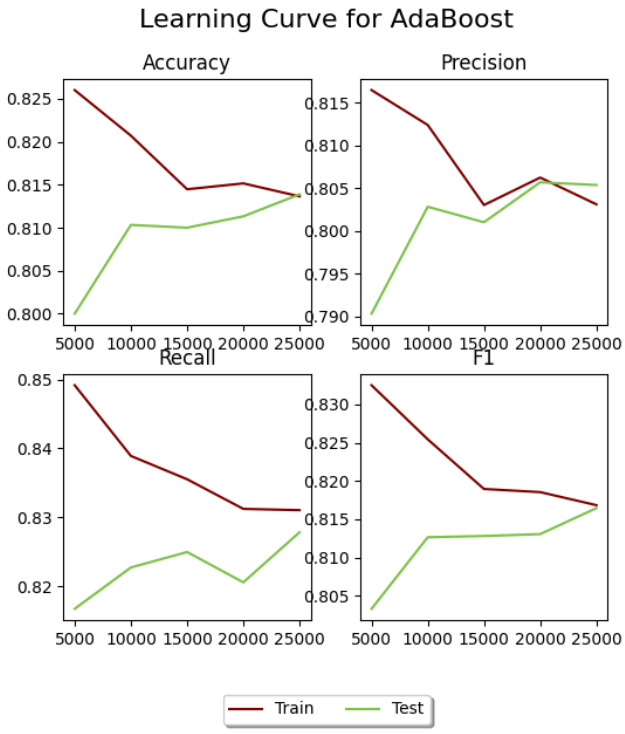


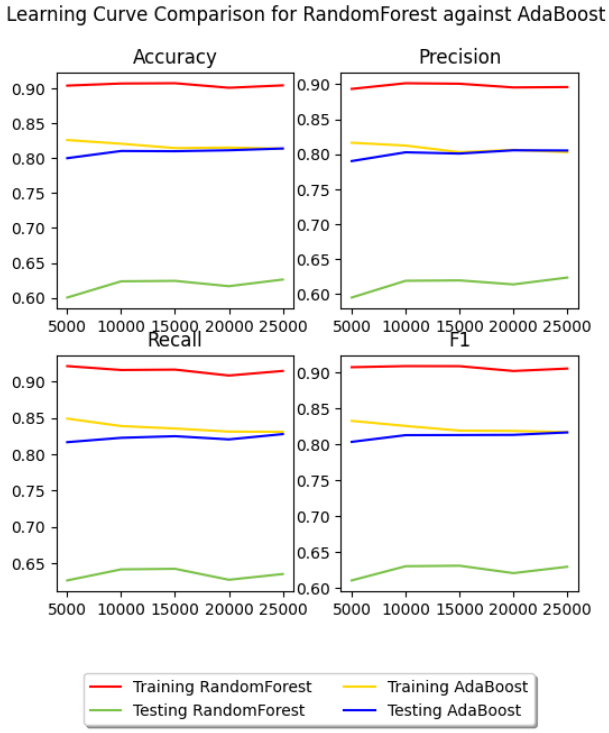




[17]

Learning, F1, Precision, Recall Curves for SkLearn's AdaBoost

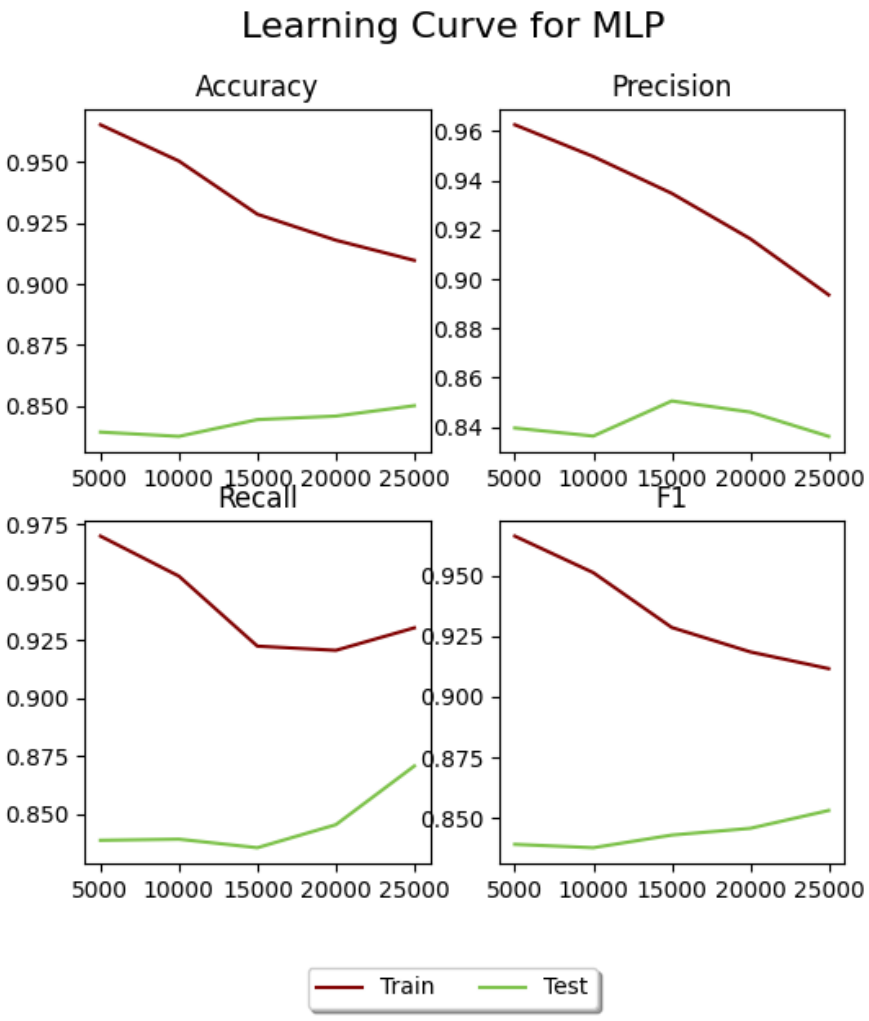




Πίνακας Διαφοράς για τον RandomForest και τον AdaBoost

	Train Accuracy	Test Accuracy	Precision Train	Precision Test	Recall Train	Recall Test	F1 Train	F1 Test
5000	0.070000	0.200000	0.070000	0.190000	0.070000	0.190000	0.080000	0.190000
10000	0.090000	0.190000	0.090000	0.180000	0.080000	0.180000	0.080000	0.180000
15000	0.100000	0.190000	0.100000	0.180000	0.080000	0.180000	0.090000	0.180000
20000	0.080000	0.190000	0.090000	0.200000	0.080000	0.190000	0.080000	0.190000
25000	0.090000	0.180000	0.100000	0.190000	0.080000	0.190000	0.090000	0.190000

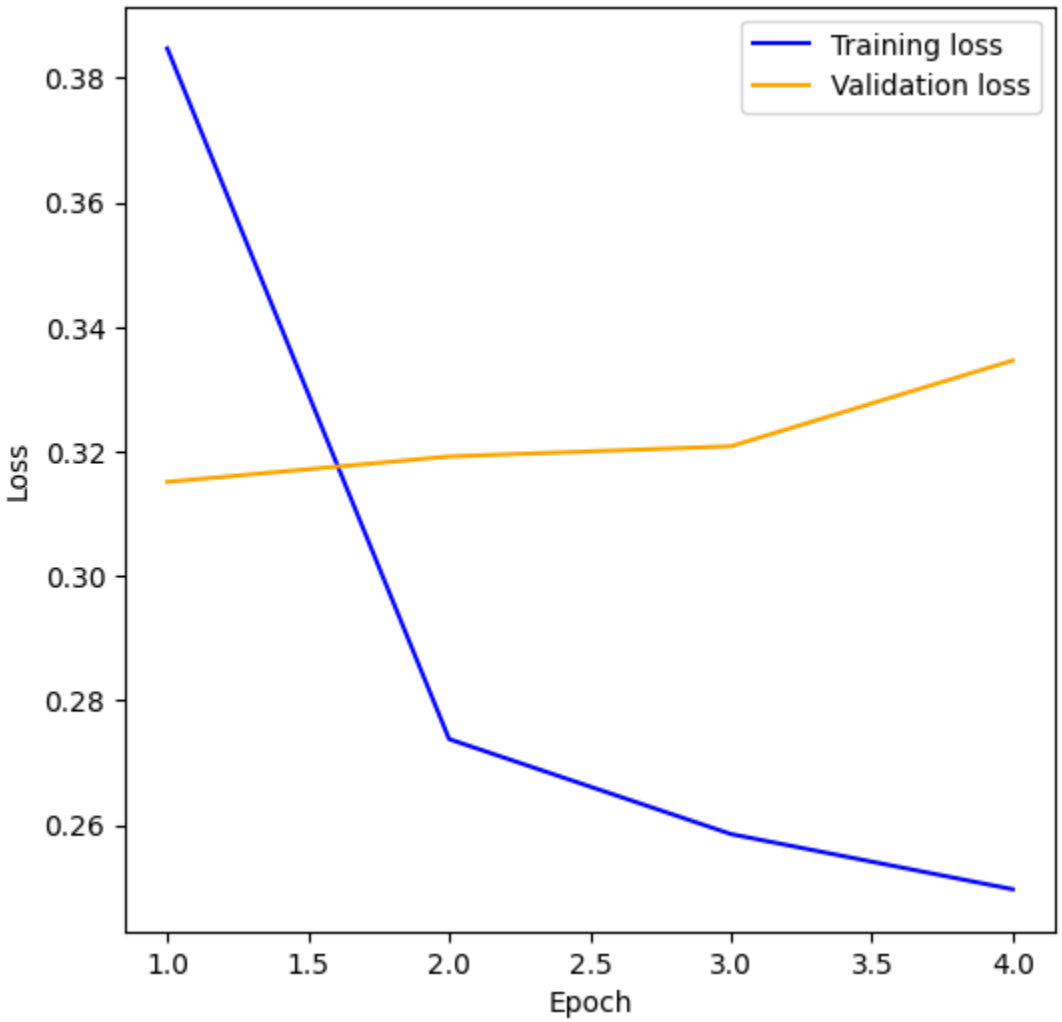
Μέρος Γ



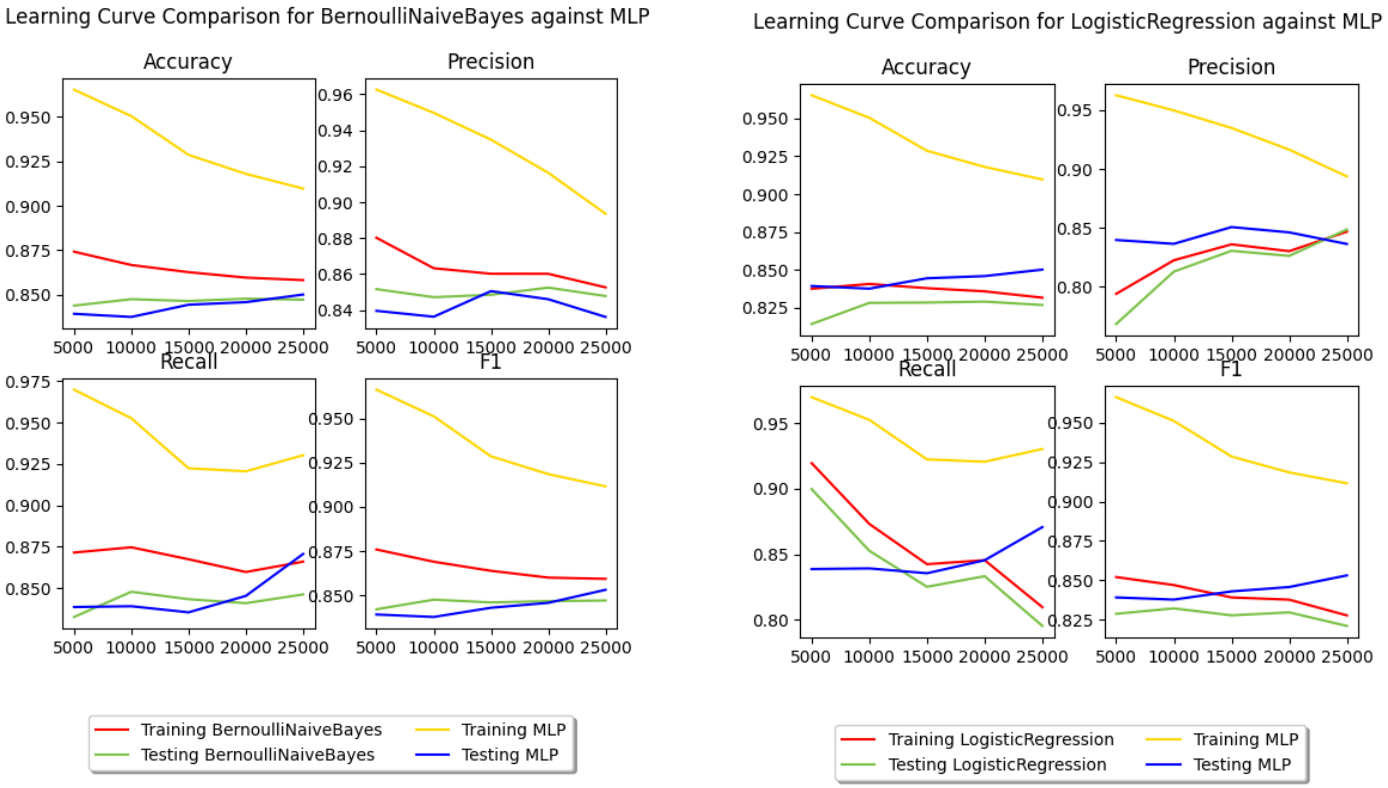
	Train Accuracy	Test Accuracy	Precision Train	Precision Test	Recall Train	Recall Test	F1 Train	F1 Test
5000	0.97	0.84	0.96	0.84	0.97	0.84	0.97	0.84
10000	0.95	0.84	0.95	0.84	0.95	0.84	0.95	0.84
15000	0.93	0.84	0.93	0.85	0.92	0.84	0.93	0.84
20000	0.92	0.85	0.92	0.85	0.92	0.85	0.92	0.85
25000	0.91	0.85	0.89	0.84	0.93	0.87	0.91	0.85

Όπως παρατηρούμε, ο αλγόριθμος του νευρωνικού δικτύου, έχει καλύτερα αποτελέσματα σε όλες τις συναρτήσεις αξιολόγησης(ακρίβεια, προσαρμοσμένη ακρίβεια, ανάκληση, F1). Αυτό ισχύει τόσο σε επίπεδο δικών μας υλοποιήσεων, καθώς και σε επίπεδο υλοποιήσεων της βιβλιοθήκης SkLearn. Ο λόγος είναι ότι οι **εποχές**, βοηθάνε το νευρωτικό στο να **μαθαίνει καλύτερα κατά την εκπαίδευση** καθώς και να **αποδίδει καλύτερα κατά την αξιολόγηση**.

Καμπύλη μεταβολής του σφάλματος



Learning, F1, Precision, Recall Curves for SkLearn's algorithms



Επίλογος

Η εργασία αυτή έγινε στα πλαίσια του μαθήματος «**Τεχνητή Νοημοσύνη**», το οποίο αποτελεί υποχρεωτικό μάθημα πυρήνα του 3ου έτους του τμήματος της Πληροφορικής. Το pdf, έχει γραφεί αποκλειστικά από τους φοιτητές της ομάδας, καθώς και όλος ο κώδικας που έχει υλοποιηθεί. Αρωγός σε αυτήν την προσπάθεια αποτέλεσαν οι σημειώσεις του μαθήματος, καθώς και των φροντιστηρίων. Χρήσιμες πληροφορίες επίσης πάρθηκαν από τα βιβλία των **S.Russell** και **P.Norvig «Τεχνητή Νοημοσύνη, μία σύγχρονη προσέγγιση»** 4η αμερικανική έκδοση και **Ι.Βλαχάβα, Π.Κεφαλά, Ν.Βασιλειάδη, Φ.Κόκκορα** και **Η.Σκελλαρίου «Τεχνητή Νοημοσύνη»** Δ έκδοση.