

Μάθημα: Τεχνητή Νοημοσύνη

Ακαδημαϊκό έτος: 2023–24

Διδάσκων: Ι. Ανδρουτσόπουλος

Μέλη Ομάδας: Μπερντούφι Ντανιέλα (ΑΜ: 3210132)

Θανάση Ανέστης (ΑΜ: 3210273)

2^η Εργασία

Ο φάκελος μας έχουμε συμπεριλάβει πέντε αρχεία py τα οποία είναι τα ακόλουθα:

- MyBayes.ipynb με κώδικα που έχουμε υλοποιήσει εμείς τον αλγόριθμο Naïve Bayes με πολυμεταβλητή μορφή Bernoulli και αντίστοιχο κώδικα για καμπύλες και πίνακες.
- Bayes.ipynb με κώδικα που χρησιμοποιούμε το αντίστοιχο αλγόριθμο της βιβλιοθήκης και αντίστοιχο κώδικα για καμπύλες και πίνακες.
- mainMyRandomForest.ipynb με κώδικα που έχουμε υλοποιήσει εμείς τον αλγόριθμο RandomForest και αντίστοιχο κώδικα για καμπύλες και πίνακες.
- mainRandomForest.ipynb με κώδικα που χρησιμοποιούμε το αντίστοιχο αλγόριθμο της βιβλιοθήκης και αντίστοιχο κώδικα για καμπύλες και πίνακες.
- mainMLP.ipynb με κώδικα που υλοποιούμε εμείς το mlr με κυλιόμενο παράθυρο και αντίστοιχο κώδικα για καμπύλες και πίνακες.

Μέρος Α:

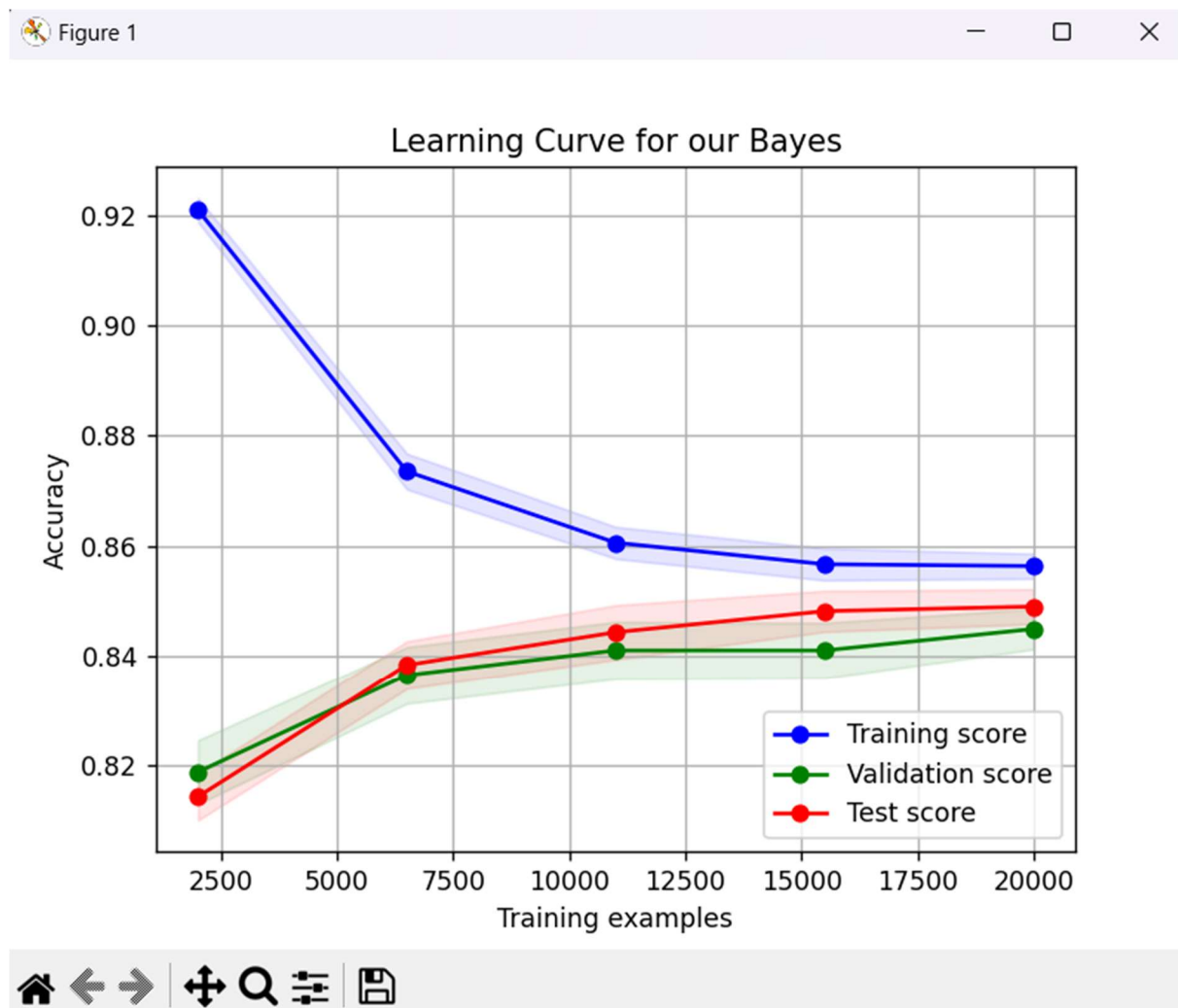
Naïve Bayes

Παράμετροι :

Βγάλαμε τις λέξεις που εμφανίζονταν πάνω από 2000 φορές (διότι για παράδειγμα λέξεις όπως το “a”, “the” που είναι συχνές δεν βοηθούν στην κατάταξη του κειμένου) η αντίστοιχη παράμετρος μας ονομάζεται max_df στον κώδικα μας, βγάλαμε τις λέξεις που εμφανίζονται λιγότερο από 100 φορές η αντίστοιχη παράμετρος μας ονομάζεται min_df στον κώδικα μας. Τέλος κρατάμε από αυτές που μένουν τις 8500 η αντίστοιχη παράμετρος μας ονομάζεται max_features στον κώδικα μας και είναι όλα όρισμα στο CountVectorizer. Ενώ για να εκπαιδεύσουμε τον αλγόριθμο μας χρησιμοποιούμε και τα 25000 παραδείγματα εκπαίδευσης

Καμπύλες μάθησης για Bayes:

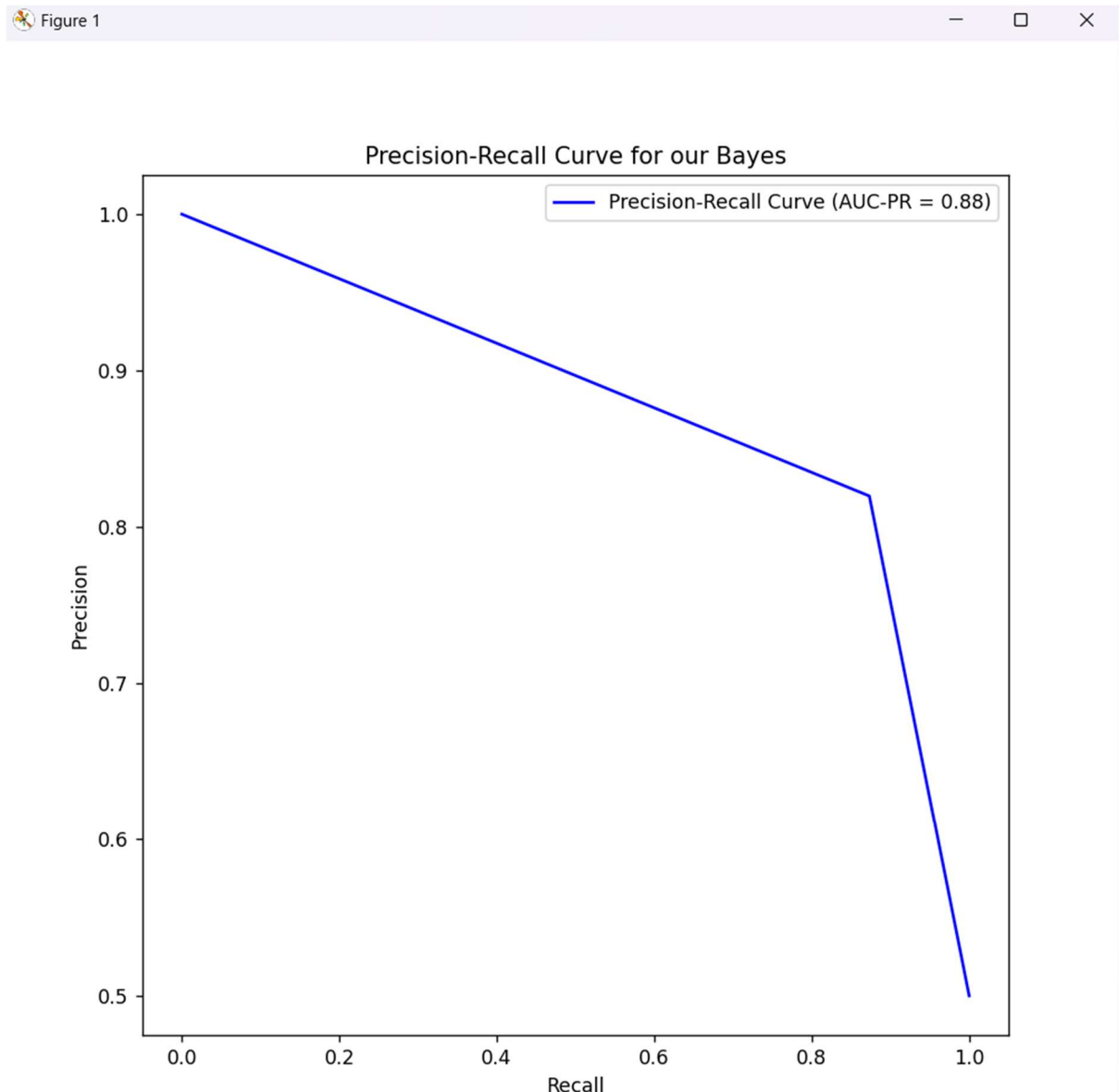
Η παρακάτω εικόνα δείχνει τις καμπύλες μάθησης για τον Naïve Bayes που έχουμε υλοποιήσει παρατηρούμε ότι όσο αυξάνονται τα παραδείγματα εκπαίδευσης τόσο η καμπύλη εκπαίδευσης μειώνεται ενώ οι καμπύλες επικύρωσης και ελέγχου αυξάνονται. Η διαφορά μεταξύ καμπύλης εκπαίδευσης και καμπύλης ελέγχου είναι μικρή προς το τέλος που χρησιμοποιούνται όλα τα παραδείγματα εκπαίδευσης το οποίο είναι καλό σημαίνει ότι το μοντέλο μας έχει εκπαιδευτεί καλά.



Καμπύλη ανάκλησης και ακρίβειας για Bayes.

Η επόμενη εικόνα δείχνει την καμπύλη ακρίβειας ανάκλησης για τον αλγόριθμο μας. Παρατηρούμε ότι καθώς αυξάνονται οι θετικές περιπτώσεις σε σύγκριση με το πλήθος τους η ακρίβεια μειώνεται με σχετικά σταθερό ρυθμό και έπειτα από ένα σημείο μειώνεται απότομα. Επιπλέον παρατηρούμε ότι κάτω από την καμπύλη ακρίβειας ανάκλησης υπάρχει πολύς χώρος το συμβολίζουμε με AUC-PR (area under the precision recall curve) το οποίο

δείχνει την συνολική απόδοση του αλγορίθμου σε σχέση με την ακρίβεια και ανάκληση και είναι 0,88 το οποίο είναι πολύ καλό σημαίνει ότι όταν κατατάσσει ένα κείμενο στην θετική κατηγορία τότε αυτό με πιθανότητα 0,88 πραγματικά ανήκει στην θετική κατηγορία.



Πίνακας με αποτελέσματα ακρίβειας, ανάκλησης και F1 (1^{ος} πίνακας) καθώς και πίνακας που να δείχνουν το ποσοστό ορθότητας (accuracy) στα δεδομένα εκπαίδευσης (training data, όσα έχουν χρησιμοποιηθεί κάθε φορά) και ελέγχου (test data) συναρτήσει του πλήθους των παραδειγμάτων εκπαίδευσης που χρησιμοποιούνται σε κάθε επανάληψη του πειράματος (2^{ος} πίνακας) για Bayes.

```

C:\Users\Daniela\Downloads\aclImdb.v1\aclImdb>python MyBayes.py
2024-01-15 14:15:15.382084: I tensorflow/core/util/port.cc:113] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point round-off errors from different computation orders. To turn them off, set the environment variable 'TF_ENABLE_ONEDNN_OPTS=0'.
WARNING:tensorflow:From C:\Users\Daniela\AppData\Local\Programs\Python\Python310\lib\site-packages\keras\src\losses.py:2976: The name tf.losses.sparse_softmax_cross_entropy is deprecated. Please use tf.compat.v1.losses.sparse_softmax_cross_entropy instead.

Vocabulary size: 3567
      precision    recall  f1-score   support

     0       0.89      0.81      0.85     12500
     1       0.83      0.90      0.86     12500

 accuracy          0.86      0.85      0.85     25000
 macro avg          0.86      0.85      0.85     25000
 weighted avg          0.86      0.85      0.85     25000

      precision    recall  f1-score   support

     0       0.86      0.81      0.84     12500
     1       0.82      0.87      0.85     12500

 accuracy          0.84      0.84      0.84     25000
 macro avg          0.84      0.84      0.84     25000
 weighted avg          0.84      0.84      0.84     25000

 Training Size  Accuracy (Training)  Accuracy (Test)
0      2500      0.909200      0.81808
1      5000      0.880000      0.82984
2      7500      0.871467      0.83388
3     10000      0.866400      0.83968
4     12500      0.860960      0.83996
5     15000      0.857800      0.83864
6     17500      0.854971      0.83912
7     20000      0.852650      0.83736
8     22500      0.854311      0.83948
9     25000      0.854520      0.84060

Naive Bayes Accuracy: 0.8406

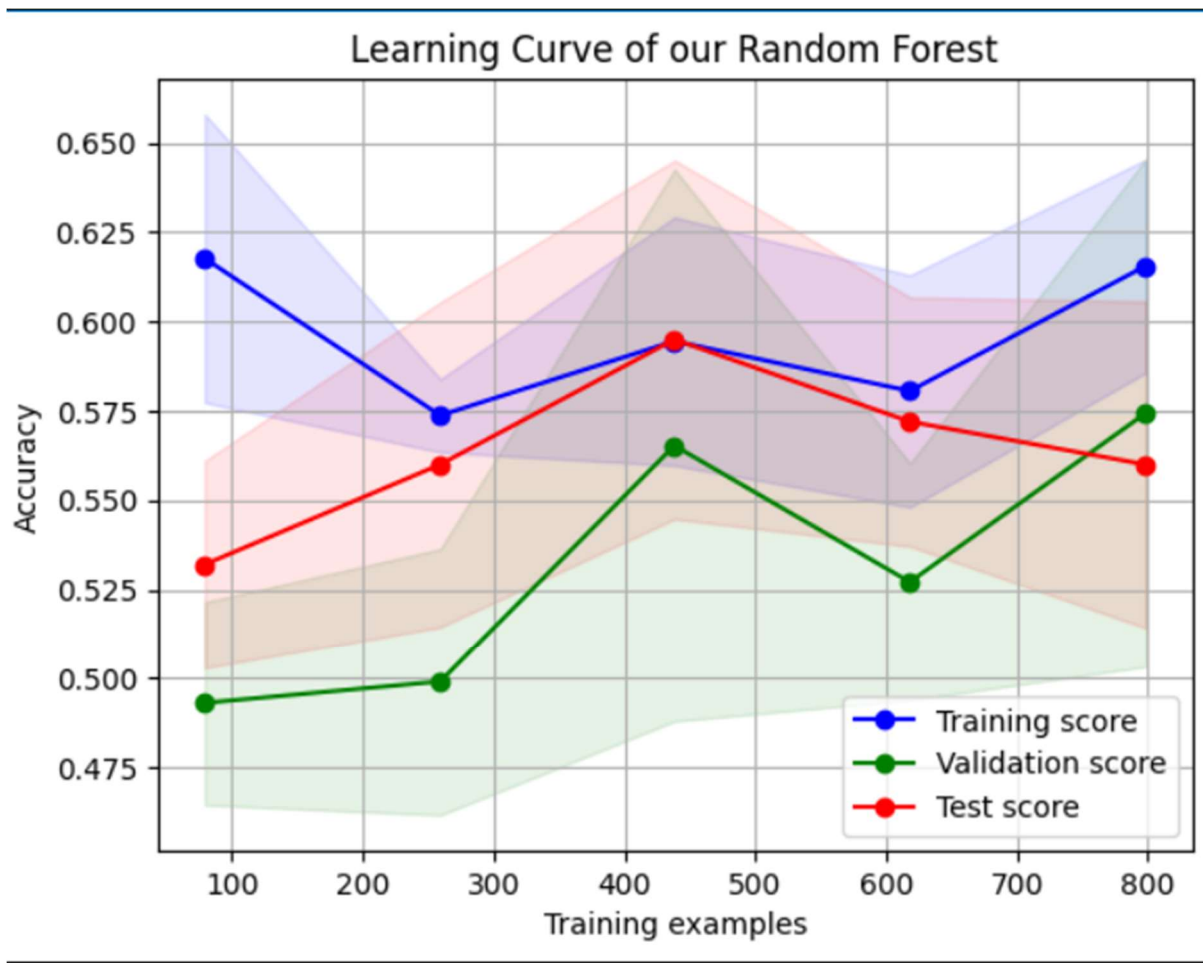
```

Random Forest

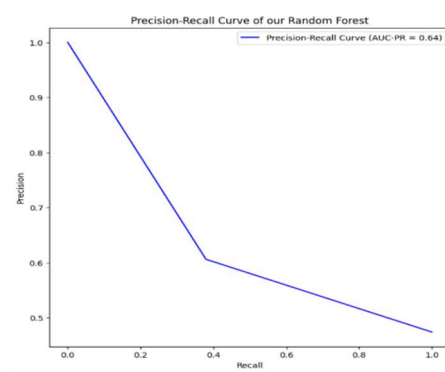
Οι παράμετροι που χρησιμοποιήσαμε για τον Random Forest είναι οι ακόλουθοι:

χρησιμοποιήσαμε 20 δέντρα για 500 παραδείγματα εκπαίδευσης έχουμε βάλει μέγιστο βάθος που μπορεί να πάει το κάθε δέντρο 4. Τέλος για το λεξιλόγιο έχουμε βγάλει τις λέξεις με συχνότητα πάνω από 2000 και όσες έχουν συχνότητα κάτω από 100 και από αυτές έχουμε κρατήσει τις 100 πιο συχνές. Έχουμε υπόψη ότι θα ήταν καλύτερο να χρησιμοποιήσουμε περισσότερα παραδείγματα εκπαίδευσης αλλά ο αλγόριθμος αργούσε πολύ να βγάλει διαγράμματα.

Η παρακάτω εικόνα δείχνει τις καμπύλες μάθησης για τον Random Forest παρατηρούμε ότι υπάρχει μια διαφορά ανάμεσα στην καμπύλη εκπαίδευσης και ελέγχου αυτό μπορεί να οφείλεται στο ότι έχουμε υπερκπαίδευση κάτι που είναι λογικό εφόσον έχουν χρησιμοποιηθεί μόνο 500 παραδείγματα εκπαίδευσης.

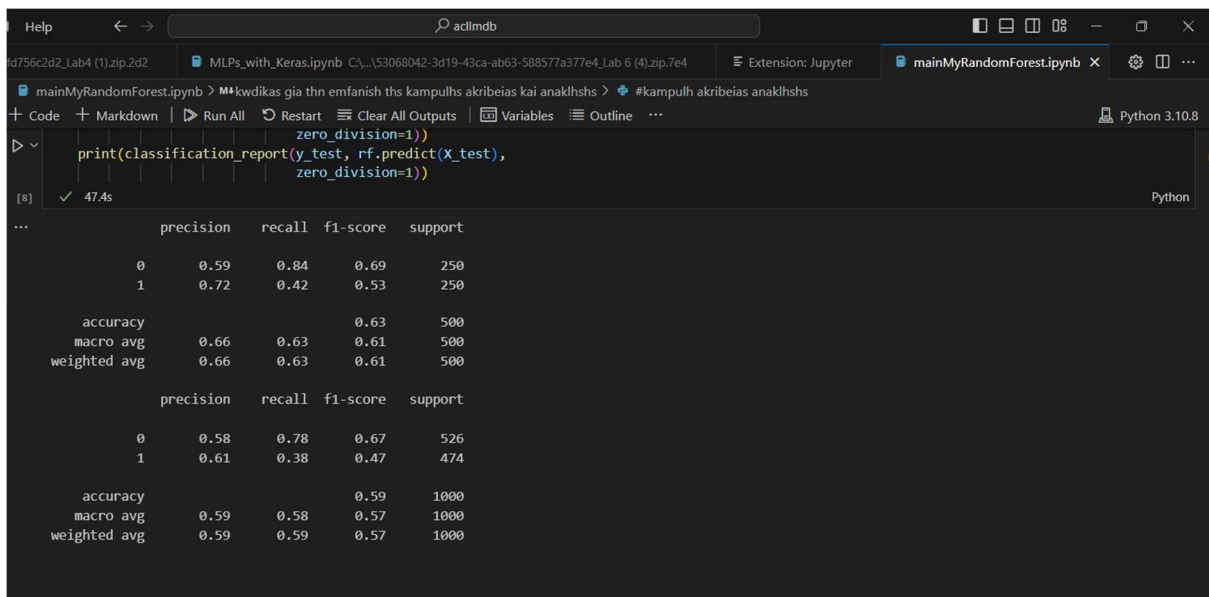


Η παρακάτω εικόνα δείχνει την καμπύλη ανάκλησης και ακρίβειας δείχνει ότι όσο αυξάνεται το ποσοστό της ανάκλησης δηλαδή των παραδειγμάτων που είναι πράγματι είναι θετικά και ο αλγόριθμος τα βρίσκει θετικά τόσο το ποσοστό ακρίβειας μειώνεται.



Η παρακάτω εικόνα δείχνει τους πίνακες της ακρίβειας, ανάκλησης και F1 καθώς και τον πίνακα με τα ποσοστά ακρίβειας ανάλογα με το πόσα παραδείγματα εκπαίδευσης έχουν χρησιμοποιηθεί.

Για τον πίνακα ακρίβειας, ανάκλησης και F1 μπορούμε να εστιάσουμε στο F1 που είναι μέτρο το οποίο δείχνει την ακρίβεια και την ανάκληση. Παρατηρούμε ότι το ποσοστό κυμαίνεται στο 50-60

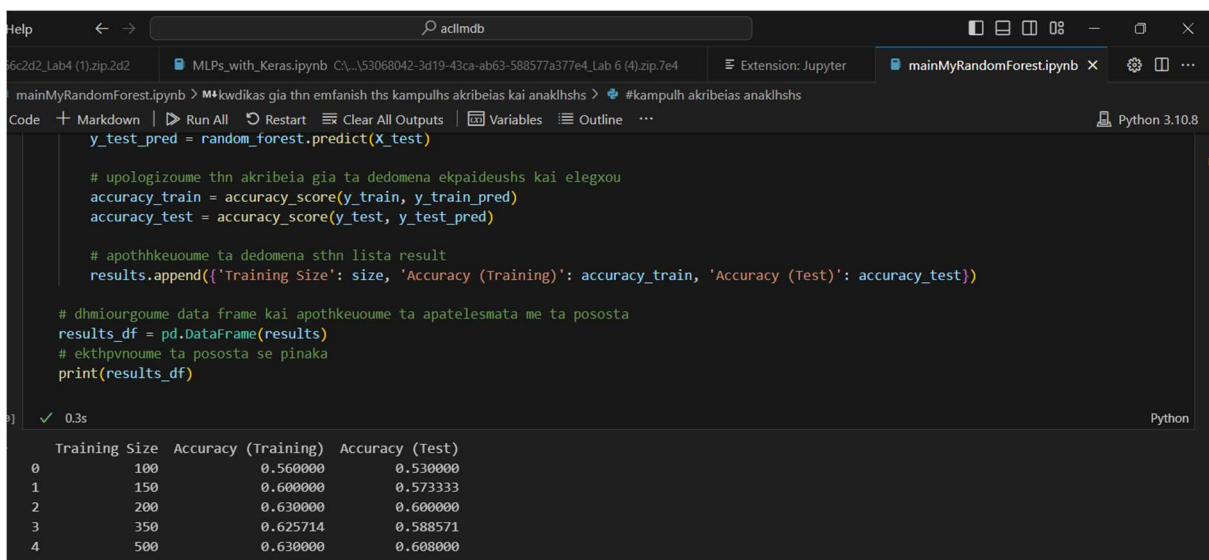


```
print(classification_report(y_test, rf.predict(X_test),
                           zero_division=1))
```

	precision	recall	f1-score	support
0	0.59	0.84	0.69	250
1	0.72	0.42	0.53	250
accuracy			0.63	500
macro avg	0.66	0.63	0.61	500
weighted avg	0.66	0.63	0.61	500

	precision	recall	f1-score	support
0	0.58	0.78	0.67	526
1	0.61	0.38	0.47	474
accuracy			0.59	1000
macro avg	0.59	0.58	0.57	1000
weighted avg	0.59	0.59	0.57	1000

Για τον πίνακα με τα ποσοστά ακρίβειας παρατηρούμε ότι καθώς αυξάνονται τα παραδείγματα εκπαίδευσης αυξάνεται και το ποσοστό και φτάνει μέχρι τα 60% περίπου για training και τεστ παραδείγματα.



```
y_test_pred = random_forest.predict(X_test)

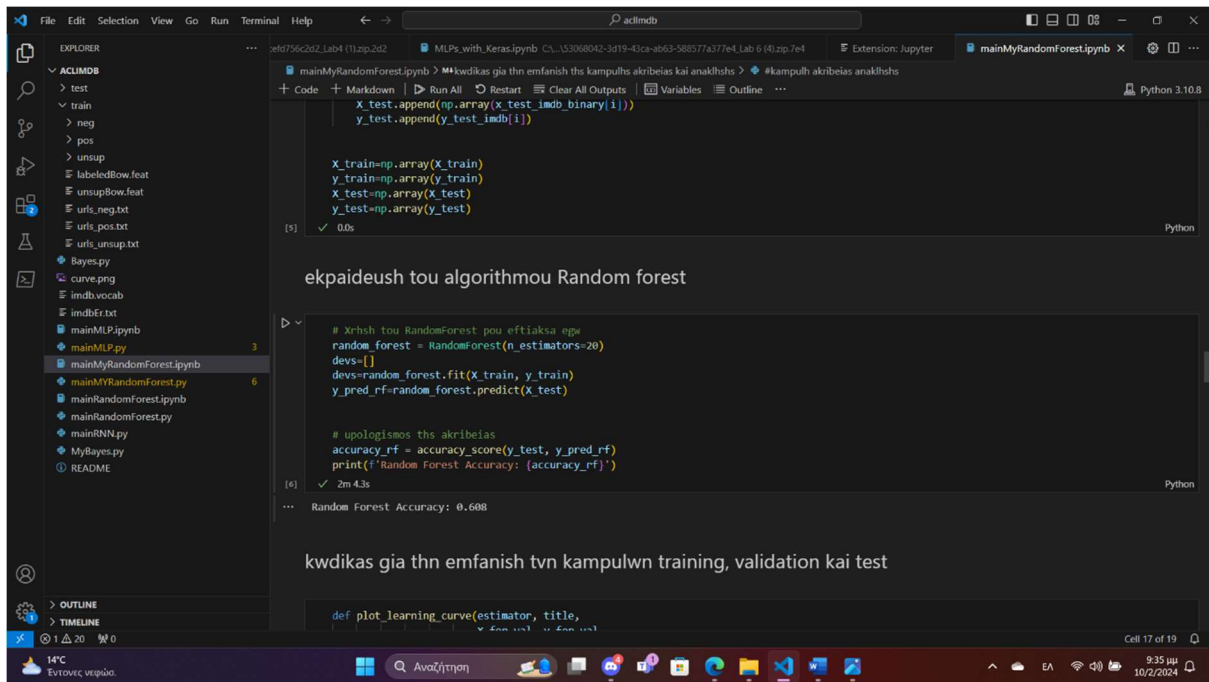
# upologizoume thn akribeia gia ta dedomena ekpaideushs kai elegxou
accuracy_train = accuracy_score(y_train, y_train_pred)
accuracy_test = accuracy_score(y_test, y_test_pred)

# apothhkeuoume ta dedomena sthn lista result
results.append({'Training Size': size, 'Accuracy (Training)': accuracy_train, 'Accuracy (Test)': accuracy_test})

# dhmiourgoume data frame kai apothhkeuoume ta apotelesmata me ta pososta
results_df = pd.DataFrame(results)
# ekthpynoume ta pososta se pinaka
print(results_df)
```

	Training Size	Accuracy (Training)	Accuracy (Test)
0	100	0.560000	0.530000
1	150	0.600000	0.573333
2	200	0.630000	0.600000
3	350	0.625714	0.588571
4	500	0.630000	0.608000

Γενικά ο αλγόριθμος μας έχει πετύχει ποσοστό ακρίβειας 60% φαίνεται στην παρακάτω εικόνα.



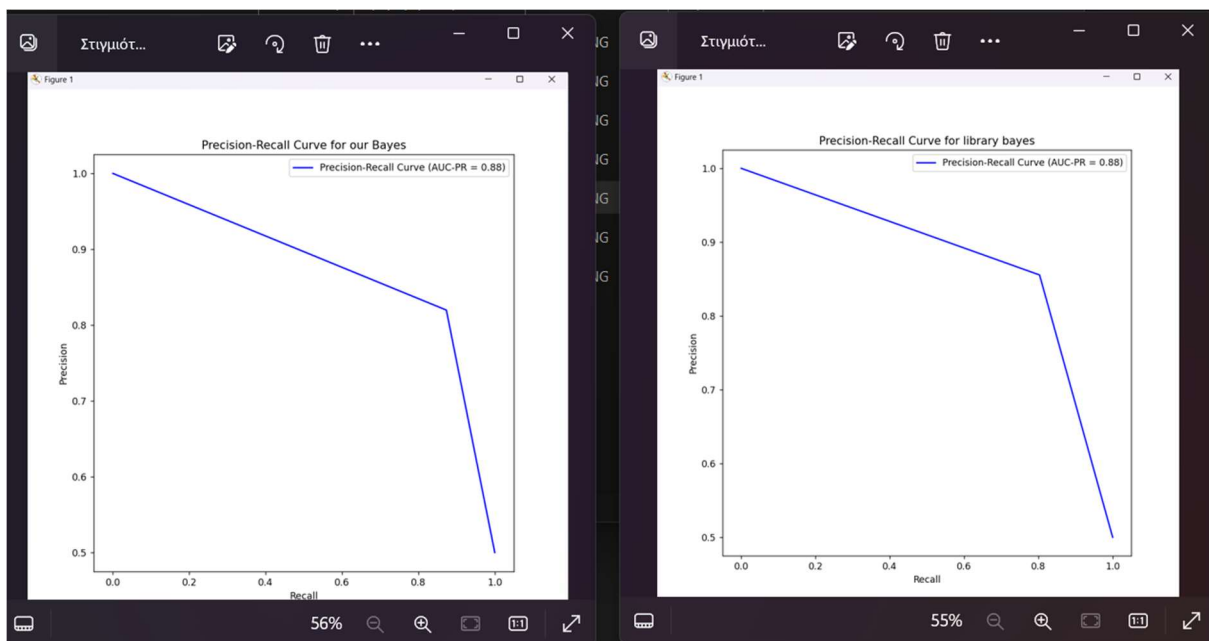
Μέρος B:

Naïve Bayes

Παρακάτω έχουμε μια εικόνα που δείχνει τις καμπύλες μάθησης που δημιουργούνται από τον αλγόριθμο Naïve Bayes που υλοποιήσαμε εμείς (είναι η εικόνα στα αριστερά και έχει τίτλο Learning curve for our Bayes) και τις καμπύλες μάθησης που δημιουργούνται από τον Naïve Bayes της βιβλιοθήκης (είναι η εικόνα δεξιά με τίτλο Learning curve for library Bayes). Παρατηρούμε ότι οι εικόνες είναι αρκετά παρόμοιες και κυμαίνονται στα ίδια περίπου ποσοστά. Συγκεκριμένα παρατηρούμε ότι το training score (ακρίβεια του training) μειώνεται καθώς αυξάνονται τα παραδείγματα εκπαίδευσης αυτό σημαίνει ότι το συνολικό σφάλμα εκπαίδευσης αυξάνεται και αυτό γίνεται επειδή μαθαίνει πολύ καλά τα συγκεκριμένα παραδείγματα εκπαίδευσης. Παρατηρούμε επίσης ότι το validation, test score (ακρίβεια του validation, test) αυξάνεται καθώς αυξάνονται τα παραδείγματα εκπαίδευσης που είναι καλό διότι σημαίνει ότι το σφάλμα τους μειώνεται. Γενικά είναι καλές εικόνες φαίνεται να έχει ο αλγόριθμός μας καλή επίδοση καθώς δεν παρατηρείται υπερφαρμογή (μεγάλη διαφορά μεταξύ καμπύλης testing και training) ούτε περιορισμένος χώρος αναζήτησης (πολύ μικρή διαφορά μεταξύ καμπυλών testing και training).



Η επόμενη εικόνα δείχνει την καμπύλη ακρίβειας ανάκλησης για τον δικό μας αλγόριθμο (αριστερά) και τον αλγόριθμο της βιβλιοθήκης (δεξιά). Παρατηρούμε ότι καθώς αυξάνονται οι θετικές περιπτώσεις σε σύγκριση με το πλήθος τους η ακρίβεια μειώνεται με σχετικά σταθερό ρυθμό και έπειτα από ένα σημείο μειώνεται απότομα αυτό συμβαίνει και στους δύο αλγορίθμους. Τέλος παρατηρούμε ότι κάτω από την καμπύλη ακρίβειας ανάκλησης υπάρχει πολύς χώρος το συμβολίζουμε με AUC-PR (area under the precision recall curve) το οποίο δείχνει την συνολική απόδοση του αλγορίθμου σε σχέση με την ακρίβεια και ανάκληση και είναι και στις δυο περιπτώσεις 0,88. Γενικά οι καμπύλες δείχνουν ότι μειώνεται η ακρίβεια καθώς αυξάνονται οι περιπτώσεις που ο αλγόριθμος τις κατατάσσει ως θετικές και είναι πράγματι θετικές.



Η παρακάτω εικόνα δείχνει τον πίνακα ακρίβειας, ανάκλησης και F1 καθώς και τον πίνακα με τα ποσοστά ακρίβειας για τα παραδείγματα εκπαίδευσης και ελέγχου με βάση τα παραδείγματα εκπαίδευσης που χρησιμοποιήθηκαν. Η αριστερή εικόνα είναι με την δική μας υλοποίηση του αλγορίθμου το καταλαβαίνουμε διότι πάνω, πάνω φαίνεται ότι τρέξαμε το MyBayes.py και το δεξιά εικόνα είναι του αλγορίθμου της βιβλιοθήκης το καταλαβαίνουμε με τον ίδιο τρόπο από το Bayes.py.

Όσον αφορά τον πίνακα ακρίβειας, ανάκλησης και F1 παρατηρούμε ότι και οι δύο αλγόριθμοι έχουν πετύχει υψηλά ποσοστά και κοντινά μεταξύ τους. Η υψηλή ανάκληση δείχνει ότι ο αλγόριθμος έχει την δυνατότητα να αναγνωρίζει μεγάλο ποσοστό των πραγματικά θετικών περιπτώσεων. Η υψηλή ακρίβεια δείχνει ότι προβλέπει θετικές περιπτώσεις με καλή ακρίβεια. Το F1 είναι ένας αριθμός που δείχνει ακρίβεια και ανάκληση μαζί και είναι αντίστοιχα και αυτό υψηλό (80%-85%).

Όσον αφορά τον πίνακα με τα ποσοστά ακρίβειας (2^{ος} πίνακας) παρατηρούμε ότι και οι δύο αλγόριθμοι έχουν βγάλει υψηλά και σχετικά κοντινά ποσοστά λίγο καλύτερα τα έχει ο δικός μας αλγόριθμος.

	precision	recall	f1-score	support
0	0.85	0.85	0.85	12500
1	0.85	0.85	0.85	12500
accuracy	0.85	0.85	0.85	25000
macro avg	0.85	0.85	0.85	25000
weighted avg	0.85	0.85	0.85	25000

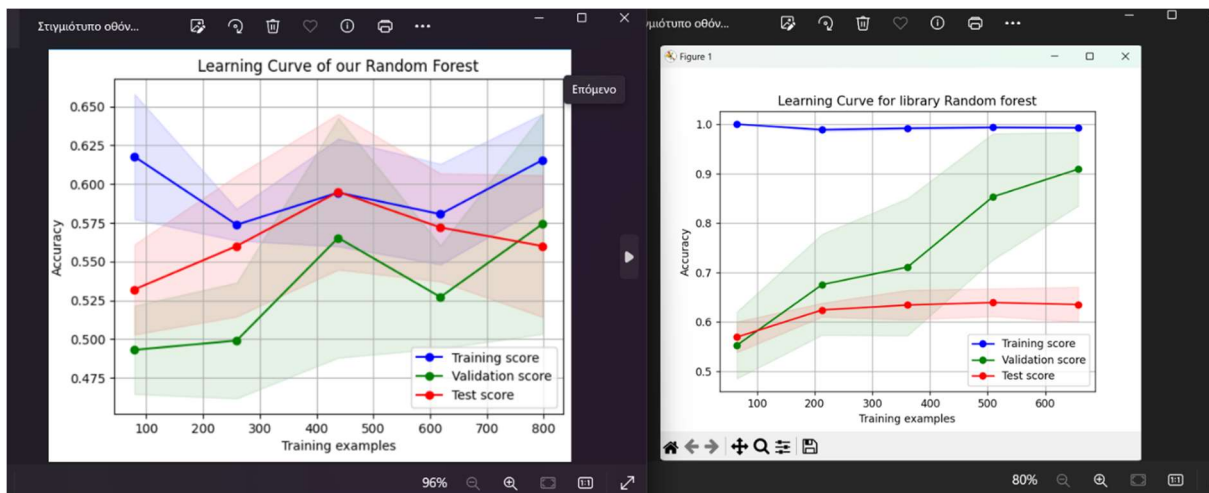
	precision	recall	f1-score	support
0	0.85	0.85	0.85	12500
1	0.85	0.85	0.85	12500
accuracy	0.85	0.85	0.85	25000
macro avg	0.85	0.85	0.85	25000
weighted avg	0.85	0.85	0.85	25000

Training Size	Accuracy (Training)	Accuracy (Test)
0	0.850000	0.850000
1	0.850000	0.850000
2	0.850000	0.850000
3	0.850000	0.850000
4	0.850000	0.850000
5	0.850000	0.850000
6	0.850000	0.850000
7	0.850000	0.850000
8	0.850000	0.850000
9	0.850000	0.850000
10	0.850000	0.850000
11	0.850000	0.850000
12	0.850000	0.850000
13	0.850000	0.850000
14	0.850000	0.850000
15	0.850000	0.850000
16	0.850000	0.850000
17	0.850000	0.850000
18	0.850000	0.850000
19	0.850000	0.850000
20	0.850000	0.850000
21	0.850000	0.850000
22	0.850000	0.850000
23	0.850000	0.850000
24	0.850000	0.850000
25	0.850000	0.850000
26	0.850000	0.850000
27	0.850000	0.850000
28	0.850000	0.850000
29	0.850000	0.850000
30	0.850000	0.850000
31	0.850000	0.850000
32	0.850000	0.850000
33	0.850000	0.850000
34	0.850000	0.850000
35	0.850000	0.850000
36	0.850000	0.850000
37	0.850000	0.850000
38	0.850000	0.850000
39	0.850000	0.850000
40	0.850000	0.850000
41	0.850000	0.850000
42	0.850000	0.850000
43	0.850000	0.850000
44	0.850000	0.850000
45	0.850000	0.850000
46	0.850000	0.850000
47	0.850000	0.850000
48	0.850000	0.850000
49	0.850000	0.850000
50	0.850000	0.850000
51	0.850000	0.850000
52	0.850000	0.850000
53	0.850000	0.850000
54	0.850000	0.850000
55	0.850000	0.850000
56	0.850000	0.850000
57	0.850000	0.850000
58	0.850000	0.850000
59	0.850000	0.850000
60	0.850000	0.850000
61	0.850000	0.850000
62	0.850000	0.850000
63	0.850000	0.850000
64	0.850000	0.850000
65	0.850000	0.850000
66	0.850000	0.850000
67	0.850000	0.850000
68	0.850000	0.850000
69	0.850000	0.850000
70	0.850000	0.850000
71	0.850000	0.850000
72	0.850000	0.850000
73	0.850000	0.850000
74	0.850000	0.850000
75	0.850000	0.850000
76	0.850000	0.850000
77	0.850000	0.850000
78	0.850000	0.850000
79	0.850000	0.850000
80	0.850000	0.850000
81	0.850000	0.850000
82	0.850000	0.850000
83	0.850000	0.850000
84	0.850000	0.850000
85	0.850000	0.850000
86	0.850000	0.850000
87	0.850000	0.850000
88	0.850000	0.850000
89	0.850000	0.850000
90	0.850000	0.850000
91	0.850000	0.850000
92	0.850000	0.850000
93	0.850000	0.850000
94	0.850000	0.850000
95	0.850000	0.850000
96	0.850000	0.850000
97	0.850000	0.850000
98	0.850000	0.850000
99	0.850000	0.850000
100	0.850000	0.850000

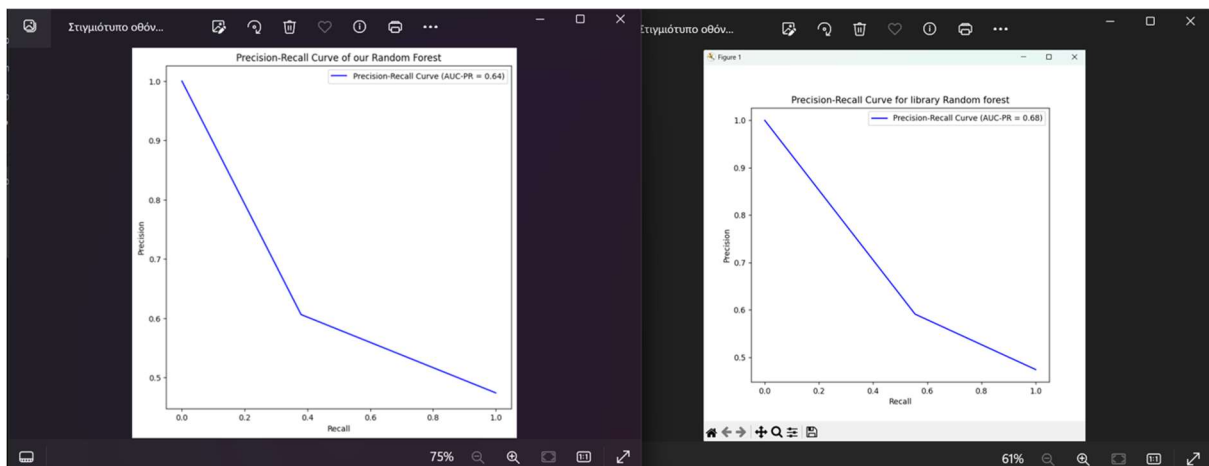
Training Size	Accuracy (Training)	Accuracy (Test)
0	0.850000	0.850000
1	0.850000	0.850000
2	0.850000	0.850000
3	0.850000	0.850000
4	0.850000	0.850000
5	0.850000	0.850000
6	0.850000	0.850000
7	0.850000	0.850000
8	0.850000	0.850000
9	0.850000	0.850000
10	0.850000	0.850000
11	0.850000	0.850000
12	0.850000	0.850000
13	0.850000	0.850000
14	0.850000	0.850000
15	0.850000	0.850000
16	0.850000	0.850000
17	0.850000	0.850000
18	0.850000	0.850000
19	0.850000	0.850000
20	0.850000	0.850000
21	0.850000	0.850000
22	0.850000	0.850000
23	0.850000	0.850000
24	0.850000	0.850000
25	0.850000	0.850000
26	0.850000	0.850000
27	0.850000	0.850000
28	0.850000	0.850000
29	0.850000	0.850000
30	0.850000	0.850000
31	0.850000	0.850000
32	0.850000	0.850000
33	0.850000	0.850000
34	0.850000	0.850000
35	0.850000	0.850000
36	0.850000	0.850000
37	0.850000	0.850000
38	0.850000	0.850000
39	0.850000	0.850000
40	0.850000	0.850000
41	0.850000	0.850000
42	0.850000	0.850000
43	0.850000	0.850000
44	0.850000	0.850000
45	0.850000	0.850000
46	0.850000	0.850000
47	0.850000	0.850000
48	0.850000	0.850000
49	0.850000	0.850000
50	0.850000	0.850000
51	0.850000	0.850000
52	0.850000	0.850000
53	0.850000	0.850000
54	0.850000	0.850000
55	0.850000	0.850000
56	0.850000	0.850000
57	0.850000	0.850000
58	0.850000	0.850000
59	0.850000	0.850000
60	0.850000	0.850000
61	0.850000	0.850000
62	0.850000	0.850000
63	0.850000	0.850000
64	0.850000	0.850000
65	0.850000	0.850000
66	0.850000	0.850000
67	0.850000	0.850000
68	0.850000	0.850000
69	0.850000	0.850000
70	0.850000	0.850000
71	0.850000	0.850000
72	0.850000	0.850000
73	0.850000	0.850000
74	0.850000	0.850000
75	0.850000	0.850000
76	0.850000	0.850000
77	0.850000	0.850000
78	0.850000	0.850000
79	0.850000	0.850000
80	0.850000	0.850000
81	0.850000	0.850000
82	0.850000	0.850000
83	0.850000	0.850000
84	0.850000	0.850000
85	0.850000	0.850000
86	0.850000	0.850000
87	0.850000	0.850000
88	0.850000	0.850000
89	0.850000	0.850000
90	0.850000	0.850000
91	0.850000	0.850000
92	0.850000	0.850000
93	0.850000	0.850000
94	0.850000	0.850000
95	0.850000	0.850000
96	0.850000	0.850000
97	0.850000	0.850000
98	0.850000	0.850000
99	0.850000	0.850000
100	0.850000	0.850000

Random Forest

Η παρακάτω εικόνα δείχνει τις καμπύλες μάθησης του αλγορίθμου που υλοποιήσαμε εμείς (αριστερή εικόνα) και του αλγορίθμου της βιβλιοθήκης. Παρατηρούμε ότι και στον δικό μας αλγόριθμο αλλά και στον αλγόριθμο της βιβλιοθήκης υπάρχει υπερεφαρμογή αυτό το καταλαβαίνουμε λόγω της μεγάλης διαφοράς μεταξύ καμπύλη εκπαίδευσης και ελέγχου αυτό οφείλεται στα λίγα παραδείγματα εκπαίδευσης που έχουμε χρησιμοποιήσει.



Η παρακάτω εικόνα δείχνει την καμπύλη ακρίβειας, ανάκλησης για τον δικό μας αλγόριθμο (αριστερή εικόνα) και για τον αντίστοιχο αλγόριθμο της βιβλιοθήκης (δεξιά εικόνα). Παρατηρούμε ότι η καμπύλη του αλγορίθμου μας είναι πιο ομαλή και άμα συγκρίνουμε το AUC-PR (area under the precision recall curve) το οποίο δείχνει την συνολική απόδοση του αλγορίθμου σε σχέση με την ακρίβεια και ανάκληση είναι λίγο μικρότερο (0,64) από ότι του αλγορίθμου της βιβλιοθήκης (0,68).



Η παρακάτω εικόνα δείχνει τους πίνακες ακρίβειας, ανάκλησης και F1 καθώς και τον πίνακα με τα ποσοστά ακρίβειας ανάλογα τα παραδείγματα εκπαίδευσης που έχουν χρησιμοποιηθεί για τον δικό μας αλγόριθμο (δεξιά εικόνα) και τον αλγόριθμο της βιβλιοθήκης (αριστερή εικόνα).

Για τον πίνακα ακρίβειας, ανάκλησης και F1 παρατηρούμε ότι καλύτερα ποσοστά έχει πετύχει ο αλγόριθμος της βιβλιοθήκης μεγάλη διαφορά υπάρχει στα δεδομένα εκπαίδευσης ο αλγόριθμος της βιβλιοθήκης έχει πετύχει πολύ καλύτερο ποσοστό.

The screenshot shows a Jupyter Notebook with two cells. The first cell contains the output of a classification report and confusion matrix for a Random Forest model. The second cell contains a print statement for the classification report.

```
print(classification_report(y_test, rf_predict(X_test),
                           zero_division=1))
```

	precision	recall	f1-score	support
0	0.59	0.84	0.69	250
1	0.72	0.42	0.53	250
accuracy	0.66	0.63	0.63	500
macro avg	0.66	0.63	0.61	500
weighted avg	0.66	0.63	0.61	500

	precision	recall	f1-score	support
0	0.58	0.78	0.67	526
1	0.61	0.38	0.47	474
accuracy	0.59	0.58	0.59	1000
macro avg	0.59	0.58	0.57	1000
weighted avg	0.59	0.59	0.57	1000

Training Size	Accuracy (Training)	Accuracy (Test)
0	1.000000	0.500000
1	0.992000	0.652000
2	0.993333	0.646667
3	0.994286	0.648571
4	0.992000	0.610000

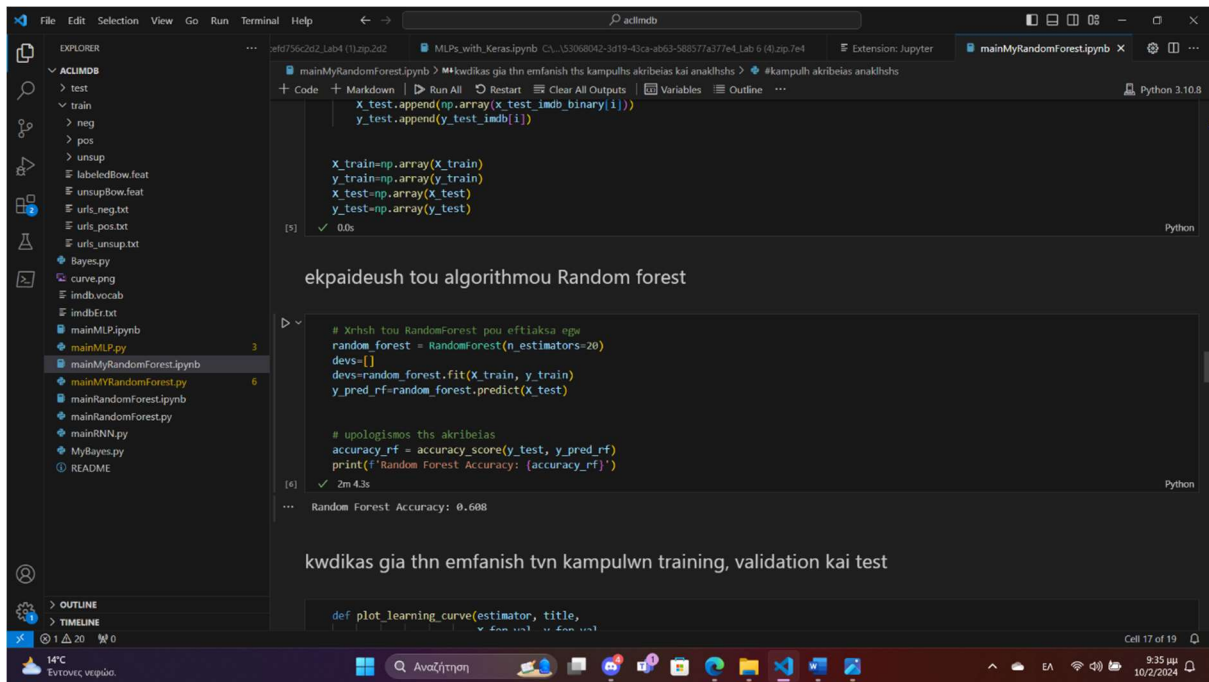
Για τον πίνακα με τα ποσοστά ακρίβειας παρατηρούμε ότι και πάλι ο αλγόριθμος της βιβλιοθήκης έχει πετύχει καλύτερα ποσοστά ειδικά στα δεδομένα εκπαίδευσης το ποσοστό είναι πολύ υψηλό. Στα δεδομένα ελέγχου τα ποσοστά του αλγορίθμου μας σε σχέση με αυτόν της βιβλιοθήκης είναι πιο κοντά.

The screenshot shows a Jupyter Notebook with two cells. The first cell contains the output of a classification report and confusion matrix for a Random Forest model. The second cell contains a print statement for the training and testing accuracy.

```
print(results_df)
```

Training Size	Accuracy (Training)	Accuracy (Test)
0	1.000000	0.500000
1	0.992000	0.652000
2	0.993333	0.646667
3	0.994286	0.648571
4	0.992000	0.610000

Γενικά άμα δούμε το ποσοστό ακρίβειας (κάτω από vocabulary size) ο αλγόριθμος μας έχει πετύχει ποσοστό ακρίβειας 0,608 (κάτω εικόνα) ενώ της βιβλιοθήκης 0,607 (πάνω εικόνα αριστερά 2^η γραμμή) .



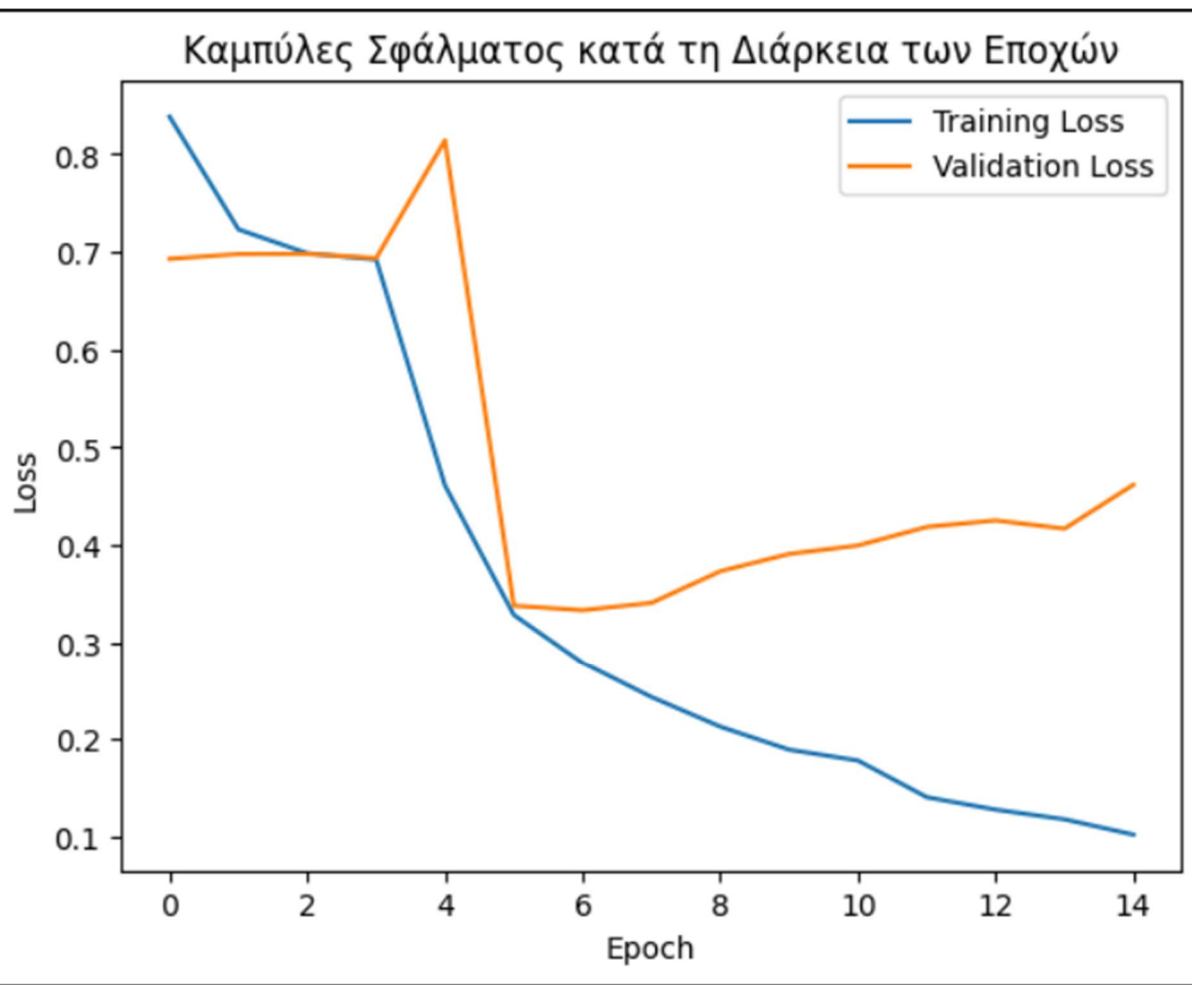
Μέρος Γ :

Για το μέρος Γ έχουμε φτιάξει MLP παριστάνοντας τις λέξεις με ενθέσεις λέξεων.

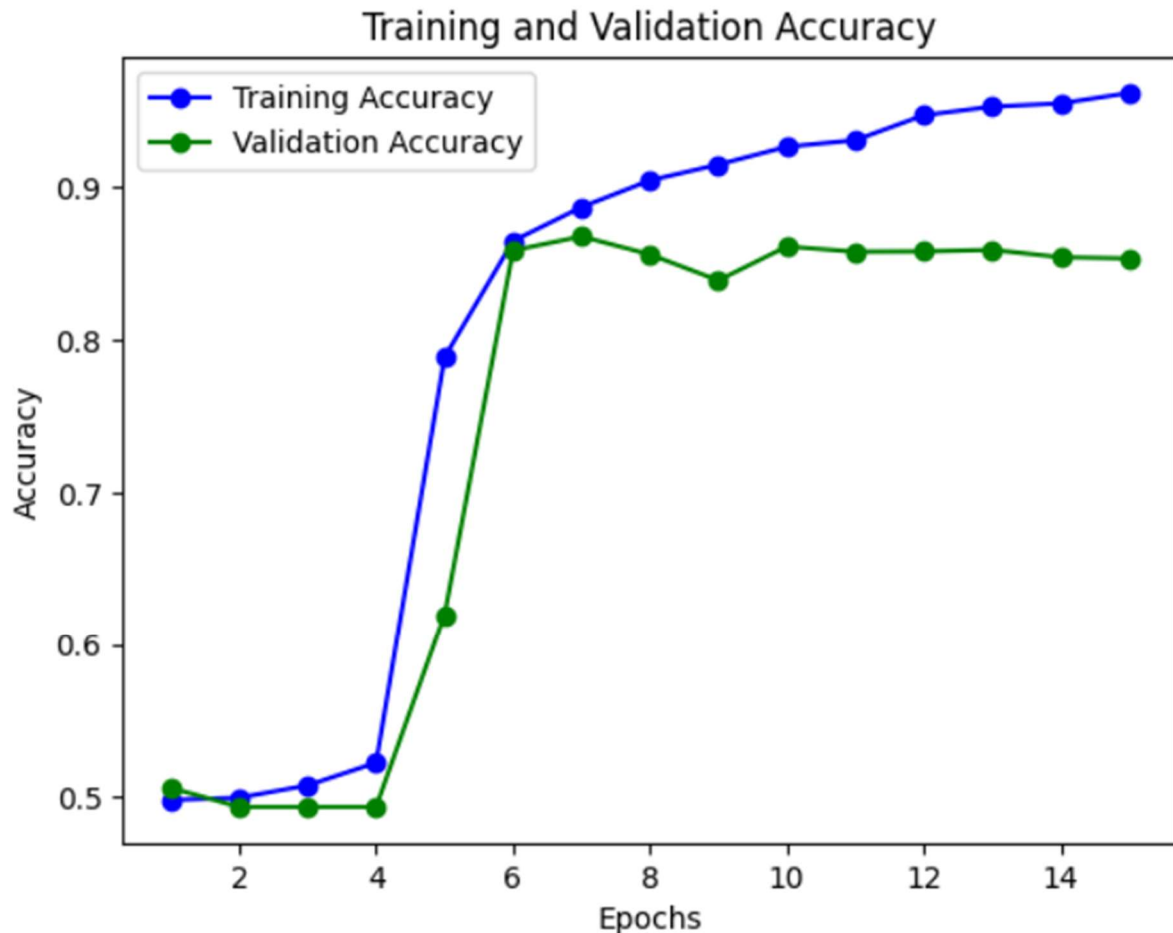
Όσον αφορά τις παραμέτρους έχουμε βάλει οι λέξεις με συχνότητα μεγαλύτερη από 2000 να βγαίνουν και λέξεις με συχνότητα μικρότερη από 100 να βγαίνουν. Από αυτές που μένουν κρατάμε τις 3567 πιο συχνές. Οι εποχές είναι 15 και τα παραδείγματα εκπαίδευσης που χρησιμοποιούμε είναι όλα δηλαδή 25000. Επιπλέον για να πετύχουμε ακόμα καλύτερα αποτελέσματα βάλαμε drop out και batch normalization μετά το αρχικό και το κρυφό επίπεδο.

Παρακάτω φαίνεται η εικόνα που δείχνει το training και validation loss του MLP.

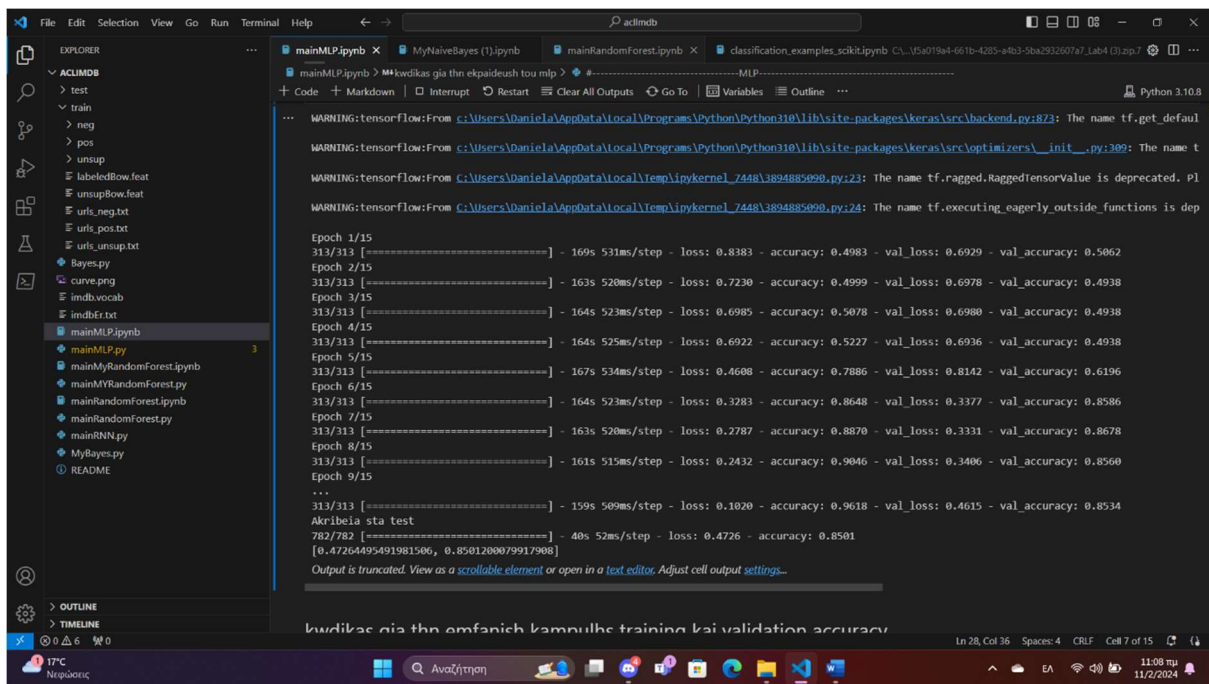
Παρατηρούμε ότι το training loss είναι μεγάλο στην αρχή και καθώς περνάνε οι εποχές σταδιακά μικραίνει και άλλο. Ενώ το validation loss μειώνεται στην αρχή απότομα και έπειτα από την εποχή 5 και μετά αρχίζει να αυξάνεται διότι αρχίζει να έχει υπερφάρμογή για αυτό σταματήσαμε στις 15 εποχές επειδή παρατηρήσαμε ότι η διαφορά training και validation loss αυξάνεται και σημαίνει ότι το μοντέλο αρχίζει να μαθαίνει καλά τα δεδομένα εκπαίδευσης. Αν παρατηρούσαμε ότι το validation loss συνεχίζει να μειώνεται τότε θα έπρεπε να βάλουμε περισσότερες εποχές για να εκπαιδεύσουμε και άλλο τον αλγόριθμο για να πετύχει καλύτερη απόδοση.



Η παρακάτω εικόνα δείχνει την ακρίβεια της καμπύλη εκπαίδευσης και επικύρωσης. Παρατηρούμε ότι η ακρίβεια στις δύο καμπύλες αυξάνεται με σχετικά παρόμοιο τρόπο και ότι από την 6^η εποχή και μετά φαίνεται να αρχίζει να υπάρχει υπερεφαρμογή αυτό σημαίνει ότι το μοντέλο αρχίζει να εξειδικεύεται στα παραδείγματα εκπαίδευσης.



Η παρακάτω εικόνα δείχνει τα ποσοστά ακρίβειας και σφάλματος για τα training και validation παραδείγματα. Παρατηρούμε ότι όσο αυξάνονται οι εποχές το μοντέλο πετυχαίνει καλύτερα ποσοστά ακρίβειας που φτάνουν μέχρι το 85%. Επιπλέον η εικόνα στο τέλος δείχνει την ακρίβεια στα παραδείγματα ελέγχου που πέτυχε το μοντέλο, το σφάλμα που έκανε είναι 0,47 και η ακρίβεια 0,85.



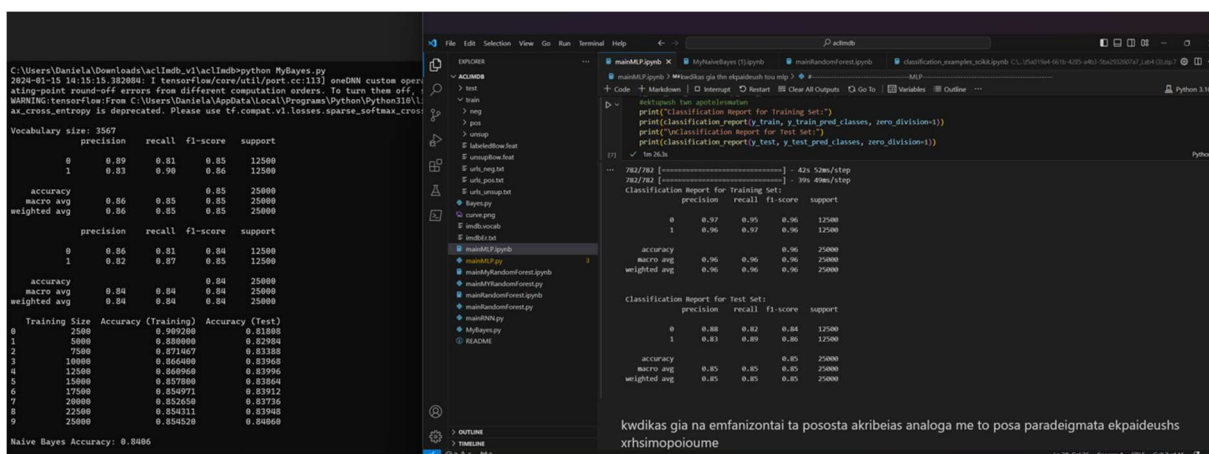
```
WARNING:tensorflow:From c:\Users\Daniela\AppData\Local\Programs\Python\Python310\lib\site-packages\keras\src\backend.py:873: The name tf.get_default_session is deprecated. Please use tf.compat.v1.get_default_session instead.
WARNING:tensorflow:From c:\Users\Daniela\AppData\Local\Programs\Python\Python310\lib\site-packages\keras\src\optimizers\tf_optimizers.py:309: The name tf.train.Optimizer is deprecated. Please use tf.compat.v1.train.Optimizer instead.
WARNING:tensorflow:From c:\Users\Daniela\AppData\Local\Temp\ipykernel_7448\3894885090.py:23: The name tf.nn.conv2d is deprecated. Please use tf.nn.conv2d_v2 instead.
WARNING:tensorflow:From c:\Users\Daniela\AppData\Local\Temp\ipykernel_7448\3894885090.py:24: The name tf.nn.conv2d is deprecated. Please use tf.nn.conv2d_v2 instead.

Epoch 1/15
313/313 [=====] - 169s 531ms/step - loss: 0.8383 - accuracy: 0.4983 - val_loss: 0.6929 - val_accuracy: 0.5062
Epoch 2/15
313/313 [=====] - 163s 520ms/step - loss: 0.7230 - accuracy: 0.4999 - val_loss: 0.6978 - val_accuracy: 0.4938
Epoch 3/15
313/313 [=====] - 164s 523ms/step - loss: 0.6985 - accuracy: 0.5078 - val_loss: 0.6980 - val_accuracy: 0.4938
Epoch 4/15
313/313 [=====] - 164s 525ms/step - loss: 0.6922 - accuracy: 0.5227 - val_loss: 0.6936 - val_accuracy: 0.4938
Epoch 5/15
313/313 [=====] - 167s 534ms/step - loss: 0.4608 - accuracy: 0.7886 - val_loss: 0.8142 - val_accuracy: 0.6196
Epoch 6/15
313/313 [=====] - 164s 523ms/step - loss: 0.3283 - accuracy: 0.8648 - val_loss: 0.3377 - val_accuracy: 0.8586
Epoch 7/15
313/313 [=====] - 163s 520ms/step - loss: 0.2787 - accuracy: 0.8870 - val_loss: 0.3331 - val_accuracy: 0.8678
Epoch 8/15
313/313 [=====] - 161s 515ms/step - loss: 0.2432 - accuracy: 0.9046 - val_loss: 0.3406 - val_accuracy: 0.8560
Epoch 9/15
...
313/313 [=====] - 159s 509ms/step - loss: 0.1020 - accuracy: 0.9618 - val_loss: 0.4615 - val_accuracy: 0.8534
Akriveia sta test
782/782 [=====] - 40s 52ms/step - loss: 0.4726 - accuracy: 0.8501
[0.4726495491981506, 0.8501200079917908]
Output is truncated. View as a scrollable element or open in a text editor. Adjust cell output settings...
```

Σύγκριση Naïve Bayes και MLP.

Η παρακάτω εικόνα δείχνει τον πίνακα ακρίβειας, ανάκλησης και F1 καθώς και τον πίνακα με τα ποσοστά ακρίβειας για δεδομένα εκπαίδευσης και ελέγχου με βάση τα παραδείγματα εκπαίδευσης που χρησιμοποιήθηκαν κάθε φορά. Η αριστερή εικόνα είναι που δείχνει τους πίνακες για Naïve Bayes και η δεξιά για MLP.

Για τον πίνακα ακρίβειας, ανάκλησης και F1 μπορούμε να κοιτάξουμε το F1 που είναι ένα μέτρο που εκφράζει ακρίβεια και ανάκληση. Παρατηρούμε ότι το F1 του mlr είναι μεγαλύτερο για τα παραδείγματα εκπαίδευσης (1^{ος} πίνακας) αλλά και για τα δεδομένα ελέγχου (2^{ος} πίνακας) σε σύγκριση με του bayes. Άρα το mlr έχει καλύτερη απόδοση στην ακρίβεια και ανάκληση απο ότι ο MLP.



	precision	recall	f1-score	support
0	0.89	0.81	0.85	12500
1	0.83	0.99	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

	precision	recall	f1-score	support
0	0.86	0.81	0.84	12500
1	0.82	0.87	0.84	12500
accuracy			0.84	25000
macro avg	0.84	0.84	0.84	25000
weighted avg	0.84	0.84	0.84	25000

Training Size	Accuracy (Training)	Accuracy (Test)
0	0.992089	0.81888
1	0.880800	0.82984
2	0.871467	0.83388
3	0.866400	0.83968
4	0.868960	0.83996
5	0.857600	0.83664
6	0.854971	0.83912
7	0.852630	0.83736
8	0.854111	0.83948
9	0.854520	0.84060

Naive Bayes Accuracy: 0.8086

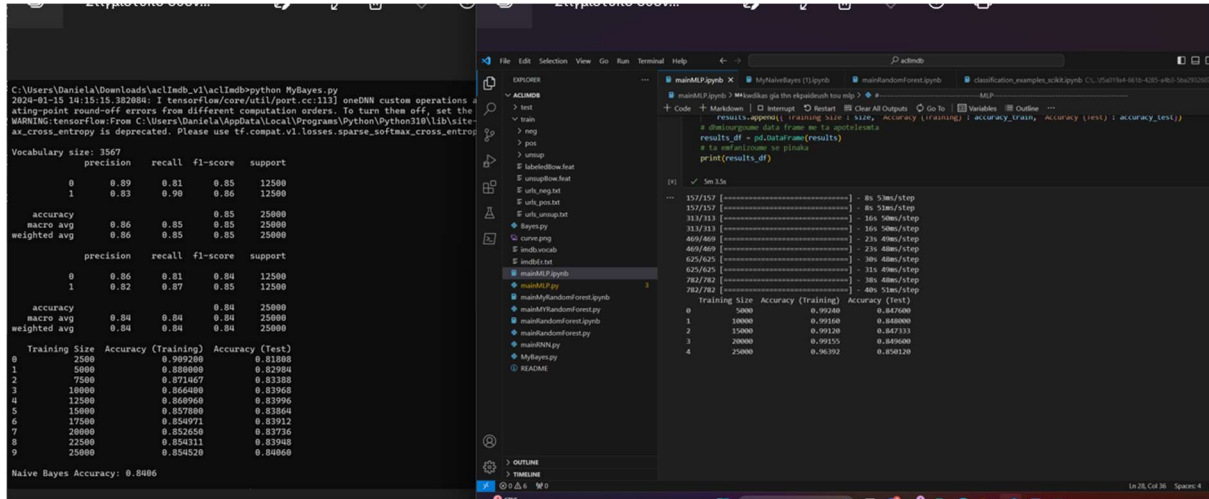
	precision	recall	f1-score	support
0	0.97	0.95	0.96	12500
1	0.96	0.97	0.96	12500
accuracy			0.96	25000
macro avg	0.96	0.96	0.96	25000
weighted avg	0.96	0.96	0.96	25000

	precision	recall	f1-score	support
0	0.88	0.82	0.84	12500
1	0.83	0.89	0.85	12500
accuracy			0.85	25000
macro avg	0.85	0.85	0.85	25000
weighted avg	0.85	0.85	0.85	25000

Classification Report for Test set:

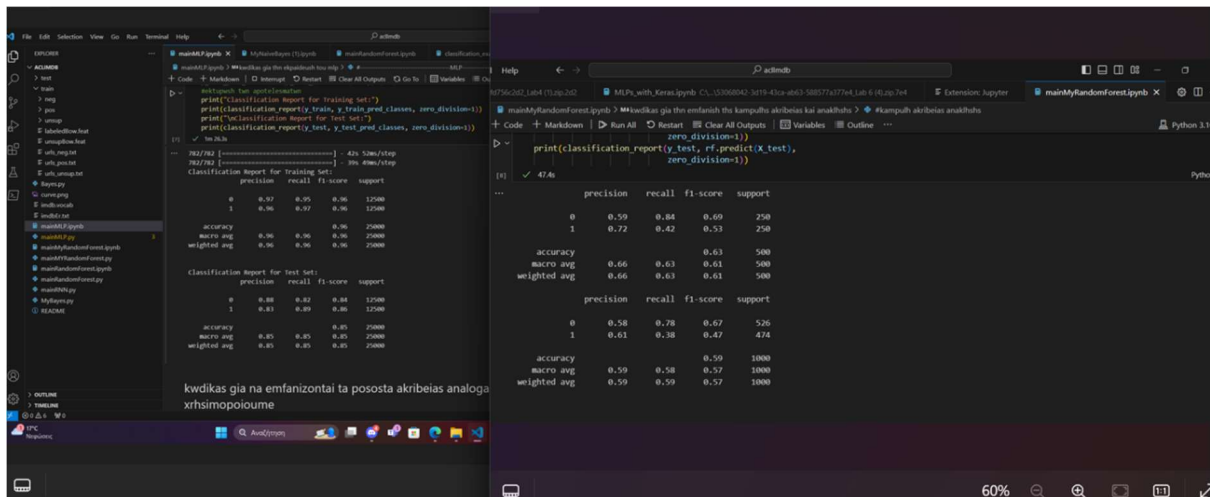
kwidias gia na emfanizontai ta posota akribeias analoga me to posa paradeigmata ekpaideusis xrhsimopoioume

Για τον πίνακα με τα ποσοστά ακρίβειας στα δεδομένα εκπαίδευσης και ελέγχου παρατηρούμε ότι άμα συγκρίνουμε τα παραδείγματα που έχουν χρησιμοποιηθεί και στους δύο αλγορίθμους διότι στον Bayes έχουμε χρησιμοποιήσει λίγα παραπάνω παραδείγματα βλέπουμε ότι έχουμε το ποσοστό που βγαίνει και για 2500 παραδείγματα ενώ στο MLP όχι αλλά θα δείτε ότι για 5000,10000,15000,20000,25000 έχουν ποσοστό και οι δύο αλγόριθμοι. Παρατηρούμε ότι και πάλι το ποσοστό ακρίβειας είναι μεγαλύτερο και άρα καλύτερο από του bayes.



Σύγκριση Random Forest με MLP

Η παρακάτω εικόνα δείχνει τους πίνακες ακρίβειας, ανάκλησης και F1 καθώς του αλγορίθμου random forest που έχουμε υλοποιήσει εμείς και του MLP παρατηρούμε και εδώ ότι το MLP έχει πετύχει πολύ καλύτερα ποσοστά από τον αλγόριθμό random forest αυτό οφείλεται στην διαφορά των παραδειγμάτων εκπαίδευσης που έχουν χρησιμοποιηθεί για τον κάθε αλγόριθμο (500 για random forest και 25000 για mlp) και πάλι τονίζουμε ότι χρησιμοποιήσαμε λίγα παραδείγματα στον random forest επειδή αργούσε πάρα πολύ να βγάλει αποτελέσματα.



Η παρακάτω εικόνα δείχνει τα ποσοστά ακρίβειας ανάλογα τον αριθμό παραδειγμάτων εκπαίδευσης που έχουμε χρησιμοποιήσει για το random forest που έχουμε υλοποιήσει εμείς (αριστερή εικόνα) και για το mlp (δεξιά εικόνα). Παρατηρούμε ότι το mlp έχει πετύχει μεγαλύτερη ακρίβεια στα training και testing παραδείγματα

