Οικονομικό Πανεπιστήμιο Αθηνών, Τμήμα Πληροφορικής

Μάθημα: Τεχνητή Νοημοσύνη

Ακαδημαϊκό έτος: 2024 – 2025

ΚΑΡΑΠΕΤΣΑ ΚΩΝΣΤΑΝΤΙΝΑ

ΣΩΤΗΡΟΠΟΥΛΟΥ ΦΩΤΕΙΝΗ

ΜΕΡΟΣ Α

Bernoulli Naive Bayes:

Αυτή η κλάση υλοποιεί τον Bernoulli Naive Bayes, ο οποίος είναι ένας απλός αλλά ισχυρός ταξινομητής που βασίζεται στη Θεωρία του Bayes και χρησιμοποιείται σε δεδομένα δυαδικής μορφής (0/1, π.χ. παρουσία ή απουσία λέξεων σε κείμενα).

- __init__()
    - Αρχικοποιεί τις μεταβλητές για τις πιθανότητες των κλάσεων και των χαρακτηριστικών.
- fit(X, y)
    - Εκπαιδεύει το μοντέλο, υπολογίζοντας:
        - Τις πιθανότητες εμφάνισης κάθε κλάσης.
        - Τις πιθανότητες εμφάνισης κάθε χαρακτηριστικού μέσα σε κάθε κλάση.
        - Εφαρμόζει Laplace Smoothing για να αποφύγει μηδενικές πιθανότητες.
- predict(X)
    - Υπολογίζει τις πιθανότητες για κάθε κλάση και επιστρέφει την κλάση με τη μεγαλύτερη πιθανότητα.
- _joint_log_likelihood(X)
    - Υπολογίζει τις λογαριθμικές πιθανότητες των δεδομένων, συνδυάζοντας:
        - Τις πιθανότητες εμφάνισης των χαρακτηριστικών.
        - Τις πιθανότητες απουσίας των χαρακτηριστικών.
        - Τις πιθανότητες των κλάσεων.
- predict_proba(X)
    - Επιστρέφει τις πιθανότητες ανήκει ένα δείγμα σε κάθε κλάση, μετά από κανονικοποίηση.

Node:

1. __init__(checking_feature=None, is_leaf=False, category=None):
    - Ο constructor για τα "κόμβους" του δέντρου απόφασης. Δημιουργεί έναν κόμβο που είτε είναι φύλλο (leaf) ή εσωτερικός κόμβος (με χαρακτηριστικό που εξετάζεται και δείχνει σε άλλους κόμβους ή φύλλα).

ID3:

Υλοποιεί τον αλγόριθμο ID3 (Iterative Dichotomiser 3) για κατασκευή δυαδικών δέντρων απόφασης.

- __init__(features, max_depth, min_samples_split, min_samples_leaf)
    - Ορίζει τις βασικές παραμέτρους του δέντρου:
        1. max_depth: Μέγιστο βάθος του δέντρου.
        2. min_samples_split: Ελάχιστος αριθμός δειγμάτων για να γίνει διαχωρισμός.
        3. min_samples_leaf: Ελάχιστος αριθμός δειγμάτων ανά φύλλο.
- fit(x, y)
    - Εκκινεί την εκπαίδευση του δέντρου με αναδρομική κλήση στη μέθοδο create_tree().
- create_tree(x_train, y_train, features, category, depth)
    - Δημιουργεί το δέντρο απόφασης αναδρομικά, χρησιμοποιώντας:
        1. Τον πιο πληροφοριακό διαχωρισμό (Information Gain - IG).
        2. Διαίρεση του dataset σε δύο υποσύνολα (0 και 1).
        3. Διακοπή της αναδρομής αν φτάσει σε φύλλο ή σε συνθήκες τερματισμού.
- calculate_ig(classes_vector, feature)
    - Υπολογίζει το Information Gain (IG) για ένα χαρακτηριστικό, χρησιμοποιώντας την Εντροπία (Entropy).
- predict(x)
    - Διατρέχει το δέντρο για κάθε παράδειγμα και επιστρέφει την προβλεπόμενη κατηγορία.

Random Forest:

Υλοποιεί έναν τυχαίο δάσος (Random Forest), το οποίο είναι ένα σύνολο από δέντρα απόφασης.

- __init__(numberOfTrees, max_depth, min_samples_split, min_samples_leaf)
    - Ο constructor για το δάσος αποφάσεων (Random Forest). Δημιουργεί μια λίστα από δέντρα αποφάσεων και ορίζει τις παραμέτρους για τον αριθμό των δέντρων, το μέγιστο βάθος, τις ελάχιστες απαιτήσεις δειγμάτων για διαχωρισμό και φύλλα.
- fit(x, y)
    - Δημιουργεί πολλά ID3 δέντρα χρησιμοποιώντας τυχαία δείγματα με επανατοποθέτηση (bootstrap sampling).
- predict(x)
    - Κάθε δέντρο κάνει μια πρόβλεψη, και η τελική απόφαση προκύπτει από την πλειοψηφική ψήφο των δέντρων.

Υπολογισμός Μετρικών:

Αυτές οι συναρτήσεις αξιολογούν την απόδοση των ταξινομητών.

- compute_metrics(y_true, y_pred)
    - Υπολογίζει precision, recall, f1-score για δυαδική ταξινόμηση.
- compute_metricsFullDetails(y_true, y_pred)
    - Υπολογίζει precision, recall, f1-score,micro και macro μέσους όρους των μετρικών.

Μετασχηματισμός Δεδομένων:

- transform_data(x_data, selected_words)
    - Μετατρέπει τα δεδομένα σε δυαδική μορφή (0/1) βασισμένη σε επιλεγμένες λέξεις.

Καμπύλη Μάθησης:

- learning_curve(bnb, X_train_bin, y_train, X_dev_bin, y_dev)

    - Υπολογίζει την καμπύλη μάθησης για το μοντέλο Naive Bayes (bnb), παρακολουθώντας την απόδοση σε διαφορετικά μεγέθη εκπαίδευσης (από 100 έως το μέγεθος του συνόλου εκπαίδευσης). Επιστρέφει τις τιμές Precision, Recall, και F1 για τα δεδομένα εκπαίδευσης και ανάπτυξης σε κάθε σημείο.

- plot_learning_curve(train_sizes, train_precision, train_recall, train_f1, dev_precision, dev_recall, dev_f1):

    - Απεικονίζει τις καμπύλες μάθησης για την ακρίβεια (Precision), ανάκληση (Recall), και σκορ F1 για τα δεδομένα εκπαίδευσης και ανάπτυξης.

Main:

Η κύρια μέθοδος που χρησιμοποιείται για να εκκινήσει η διαδικασία εκπαίδευσης και αξιολόγησης του μοντέλου. Εδώ γίνονται τα εξής βήματα:

- Φορτώνονται τα δεδομένα εκπαίδευσης και ανάπτυξης από τα αρχεία.
- Προετοιμάζονται τα δεδομένα για να γίνουν αριθμητικά (με τη μέθοδο transform data).
- Δημιουργούνται τα χαρακτηριστικά του μοντέλου Naive Bayes (Bernoulli Naive Bayes) και του δάσους απόφασης (Random Forest).
- Εκπαιδεύονται τα μοντέλα με τα δεδομένα εκπαίδευσης και υπολογίζονται οι μετρικές απόδοσης (Precision, Recall, F1).
- Σχεδιάζονται οι καμπύλες μάθησης (learning curves) για να παρακολουθηθεί η απόδοση του μοντέλου καθώς η εκπαίδευση εξελίσσεται.
- Εμφανίζονται οι μετρικές απόδοσης για το μοντέλο σε πραγματικό χρόνο, τόσο για τα δεδομένα εκπαίδευσης όσο και για τα δεδομένα ανάπτυξης.

Αναλυτικά, τα βήματα που ακολουθούνται στην main είναι:

- Φόρτωση δεδομένων κειμένου.
- Μετατροπή κειμένων σε χαρακτηριστικά χρησιμοποιώντας το λεξιλόγιο.
- Εκπαίδευση του μοντέλου (Naive Bayes και Random Forest).
- Υπολογισμός μετρικών (Precision, Recall, F1).

- Σχεδίαση καμπυλών μάθησης.
- Εκτύπωση των αποτελεσμάτων και ανάλυση της απόδοσης.

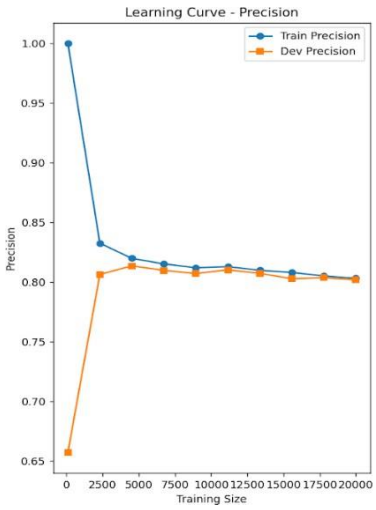
Η main είναι η βασική μέθοδος για την εκτέλεση της ανάλυσης και της αξιολόγησης των μοντέλων ταξινόμησης με τα δεδομένα του IMDB dataset, παρέχοντας τις μετρικές που βοηθούν να αξιολογηθεί η αποτελεσματικότητα του μοντέλου.

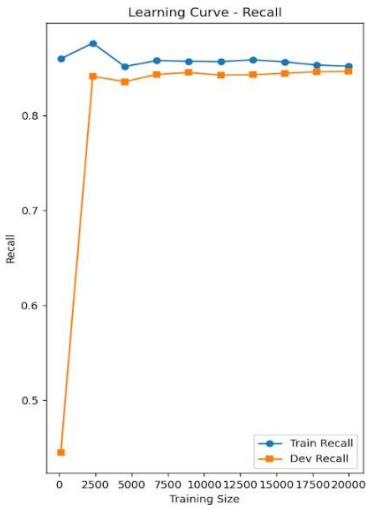
1ο Παράδειγμα

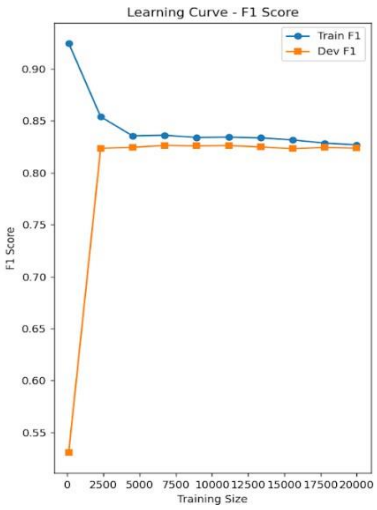
1. Μέγεθος Λεξιλογίου:1000
2. Συχνές λέξεις για αφαίρεση: 30 (Αφαιρέσαμε τις 30 πιο συχνές λέξεις, καθώς συνήθως είναι κοινές λέξεις (π.χ., "the", "and") που δεν συνεισφέρουν σημαντικά στη διάκριση των κατηγοριών.)
3. Σπάνιες λέξεις για αφαίρεση: 40 (Αφαιρέσαμε τις 40 πιο σπάνιες λέξεις για να μειώσουμε τον θόρυβο και την πιθανότητα υπερ προσαρμογής (overfitting).)
4. Λέξεις που κρατάμε με το υψηλότερο πληροφοριακό κέρδος: 700
5. RandomForest

    - number Of Trees: 100 (Επιλέξαμε 100 δέντρα καθώς αυξάνοντας τον αριθμό των δέντρων μειώνεται η διακύμανση του μοντέλου.)
    - max_depth: 20 (Περιορίσαμε το μέγιστο βάθος των δέντρων σε 20 για να αποτρέψουμε την υπερπροσαρμογή και να διατηρήσουμε μια καλή γενίκευση του μοντέλου.)

```
Πόσες από τις πιο συχνές λέξεις θέλεις να εξαιρέσεις; 30
Πόσες από τις πιο σπάνιες λέξεις θέλεις να εξαιρέσεις; 40
Πόσες λέξεις με το υψηλότερο πληροφοριακό κέρδος θέλεις να επιλέξεις; 700
Bernoulli Naive Bayes Metrics:
Classification Report:
              precision    recall  f1-score   support

           0     0.8383    0.7898    0.8134     12500
           1     0.8013    0.8477    0.8239     12500

    accuracy                         0.8188     25000
   macro avg     0.8198    0.8188    0.8186     25000
weighted avg     0.8198    0.8188    0.8186     25000

Micro Precision: 0.81876
Micro Recall: 0.81876
Micro F1: 0.81876
Macro Precision: 0.8198299801965964
Macro Recall: 0.8187599999999999
Macro F1: 0.8186082900685747

Random Forest Metrics:
Classification Report:
              precision    recall  f1-score   support

           0     0.7638    0.5954    0.6692     12500
           1     0.6685    0.8159    0.7349     12500

    accuracy                         0.7056     25000
   macro avg     0.7162    0.7056    0.7020     25000
weighted avg     0.7162    0.7056    0.7020     25000

Micro Precision: 0.70564
Micro Recall: 0.70564
Micro F1: 0.70564
Macro Precision: 0.7161552421584192
Macro Recall: 0.70564
Macro F1: 0.7020160146012299
```





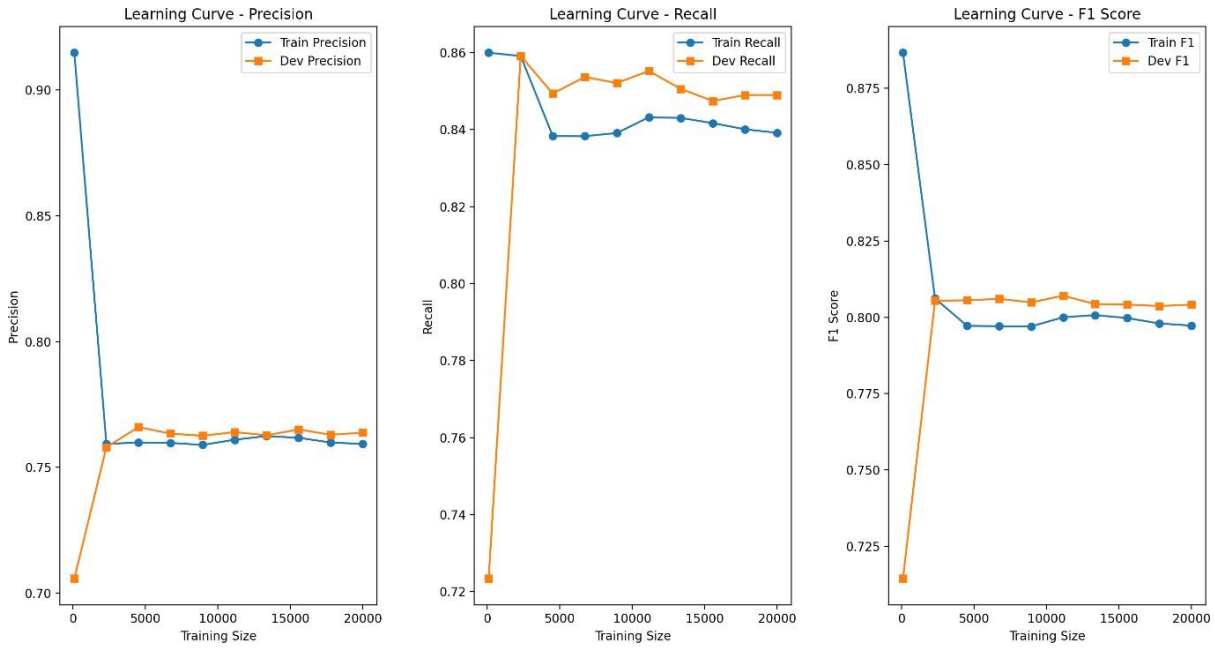


2ο Παράδειγμα

1. Μέγεθος Λεξιλογίου: 1000
2. Συχνές λέξεις για αφαίρεση: 30(Αφαιρέσαμε τις 30 πιο συχνές λέξεις, καθώς συνήθως είναι κοινές λέξεις (π.χ., "the", "and") που δεν συνεισφέρουν σημαντικά στη διάκριση των κατηγοριών.)
3. Σπάνιες λέξεις για αφαίρεση: 40 (Αφαιρέσαμε τις 40 πιο σπάνιες λέξεις για να μειώσουμε τον θόρυβο και την πιθανότητα υπερ προσαρμογής (overfitting).)
4. Λέξεις που κρατάμε με το υψηλότερο πληροφοριακό κέρδος: 100
5. RandomForest

- number Of Trees: 25 (Επιλέξαμε 25 δέντρα καθώς αυξάνοντας τον αριθμό των δέντρων μειώνεται η διακύμανση του μοντέλου.)
- max_depth: 20 (Περιορίσαμε το μέγιστο βάθος των δέντρων σε 20 για να αποτρέψουμε την υπερπροσαρμογή και να διατηρήσουμε μια καλή γενίκευση του μοντέλου.)

```
e to floating-point round-off errors from different computation orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2025-02-16 21:35:22.305626: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results du
e to floating-point round-off errors from different computation orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
Πόσες από τις πιο συχνές λέξεις θέλεις να εξαιρέσεις; 30
Πόσες από τις πιο σπάνιες λέξεις θέλεις να εξαιρέσεις; 40
Πόσες λέξεις με το υψηλότερο πληροφοριακό κέρδος θέλεις να επιλέξεις; 100
Bernoulli Naive Bayes Metrics:
Classification Report:
              precision    recall  f1-score   support

           0     0.8227    0.7315    0.7745     12500
           1     0.7583    0.8424    0.7982     12500

    accuracy                         0.7870     25000
   macro avg     0.7905    0.7870    0.7863     25000
weighted avg     0.7905    0.7870    0.7863     25000

Micro Precision: 0.78696
Micro Recall: 0.78696
Micro F1: 0.78696
Macro Precision: 0.7905319080527469
Macro Recall: 0.78696
Macro F1: 0.7863031828304072

Random Forest Metrics:
Classification Report:
              precision    recall  f1-score   support

           0     0.7646    0.5682    0.6519     12500
           1     0.6564    0.8251    0.7312     12500

    accuracy                         0.6966     25000
   macro avg     0.7105    0.6966    0.6915     25000
weighted avg     0.7105    0.6966    0.6915     25000

Micro Precision: 0.69664
Micro Recall: 0.69664
Micro F1: 0.69664
Macro Precision: 0.7105417431949 07
Macro Recall: 0.6966399999999999
Macro F1: 0.6915483546369301
```

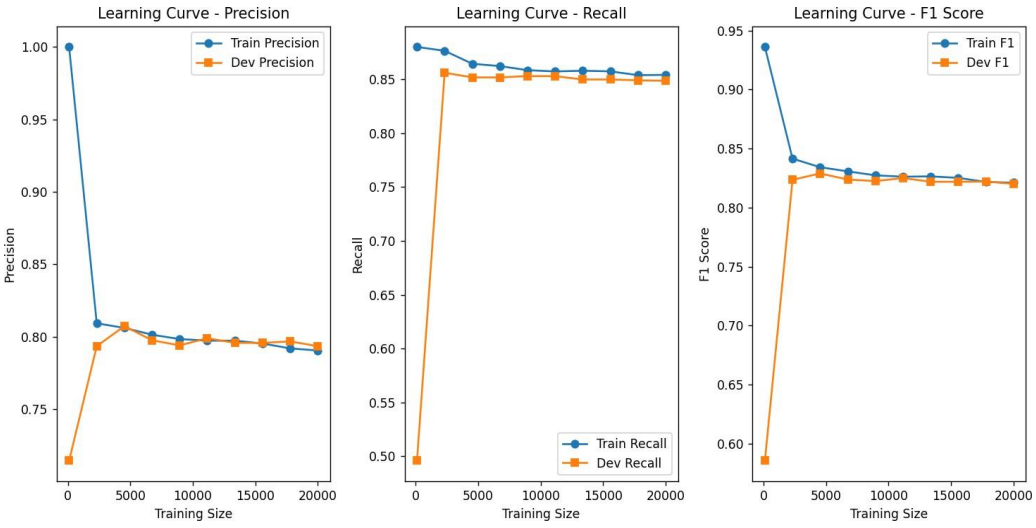


3ο Παράδειγμα

1. Μέγεθος Λεξιλογίου: 1000
2. Συχνές λέξεις για αφαίρεση: 30 (Αφαιρέσαμε τις 30 πιο συχνές λέξεις, καθώς συνήθως είναι κοινές λέξεις (π.χ., "the", "and") που δεν συνεισφέρουν σημαντικά στη διάκριση των κατηγοριών.)
3. Σπάνιες λέξεις για αφαίρεση: 40 (Αφαιρέσαμε τις 40 πιο σπάνιες λέξεις για να μειώσουμε τον θόρυβο και την πιθανότητα υπερ προσαρμογής (overfitting).)
4. Λέξεις που κρατάμε με το υψηλότερο πληροφοριακό κέρδος: 500
5. RandomForest

- number Of Trees: 100 (Επιλέξαμε 100 δέντρα καθώς αυξάνοντας τον αριθμό των δέντρων μειώνεται η διακύμανση του μοντέλου.)
- max_depth: 20 (Περιορίσαμε το μέγιστο βάθος των δέντρων σε 20 για να αποτρέψουμε την υπερπροσαρμογή και να διατηρήσουμε μια καλή γενίκευση του μοντέλου.)

```
Πόσες από τις πιο συχνές λέξεις θέλεις να εξαιρέσεις; 30
Πόσες από τις πιο σπάνιες λέξεις θέλεις να εξαιρέσεις; 40
Πόσες λέξεις με το υψηλότερο πληροφοριακό κέρδος θέλεις να επιλέξεις; 500
Bernoulli Naive Bayes Metrics:
Classification Report:
              precision    recall  f1-score   support

           0     0.8352    0.7746    0.8038     12500
           1     0.7899    0.8472    0.8175     12500

    accuracy                         0.8109     25000
   macro avg     0.8126    0.8109    0.8107     25000
weighted avg     0.8126    0.8109    0.8107     25000

Micro Precision: 0.81092
Micro Recall: 0.81092
Micro F1: 0.81092
Macro Precision: 0.8125656436105637
Macro Recall: 0.81092
Macro F1: 0.8106707976336041

Random Forest Metrics:
Classification Report:
              precision    recall  f1-score   support

           0     0.7804    0.5671    0.6569     12500
           1     0.6600    0.8404    0.7394     12500

    accuracy                         0.7038     25000
   macro avg     0.7202    0.7038    0.6981     25000
weighted avg     0.7202    0.7038    0.6981     25000

Micro Precision: 0.70376
Micro Recall: 0.70376
Micro F1: 0.70376
Macro Precision: 0.7202053681431213
Macro Recall: 0.7037599999999999
Macro F1: 0.6981238239954186
```

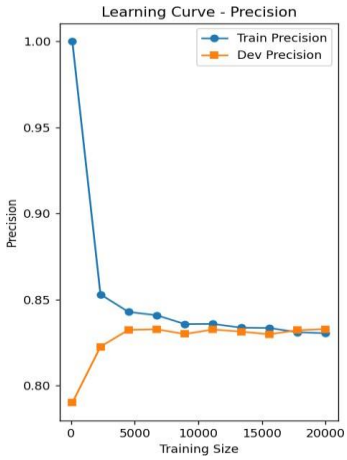


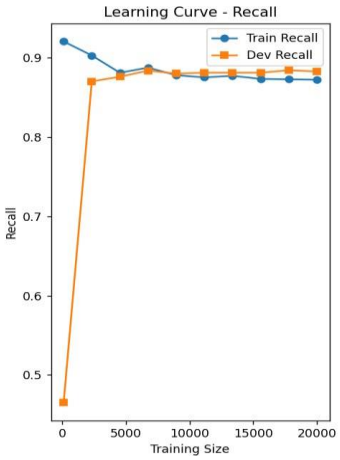
4ο Παράδειγμα

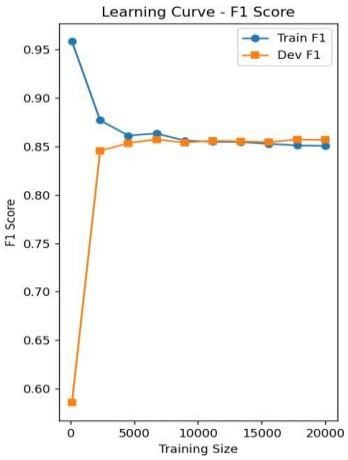
1. Μέγεθος Λεξιλογίου: 4000
2. Συχνές λέξεις για αφαίρεση: 50 (Αφαιρέσαμε τις 50 πιο συχνές λέξεις, καθώς συνήθως είναι κοινές λέξεις (π.χ., "the", "and") που δεν συνεισφέρουν σημαντικά στη διάκριση των κατηγοριών.)
3. Σπάνιες λέξεις για αφαίρεση: 50 (Αφαιρέσαμε τις 50 πιο σπάνιες λέξεις για να μειώσουμε τον θόρυβο και την πιθανότητα υπερ προσαρμογής (overfitting).)
4. Λέξεις που κρατάμε με το υψηλότερο πληροφοριακό κέρδος: 1000
5. RandomForest

    - number Of Trees: 100 (Επιλέξαμε 100 δέντρα καθώς αυξάνοντας τον αριθμό των δέντρων μειώνεται η διακύμανση του μοντέλου.)
    - max_depth: 20 (Περιορίσαμε το μέγιστο βάθος των δέντρων σε 20 για να αποτρέψουμε την υπερπροσαρμογή και να διατηρήσουμε μια καλή γενίκευση του μοντέλου.)

```
Πόσες από τις πιο συχνές λέξεις θέλεις να εξαιρέσεις; 50
Πόσες από τις πιο σπάνιες λέξεις θέλεις να εξαιρέσεις; 50
Πόσες λέξεις με το υψηλότερο πληροφοριακό κέρδος θέλεις να επιλέξεις; 1000
Bernoulli Naive Bayes Metrics:
Classification Report:
              precision    recall  f1-score   support

           0     0.8571    0.8149    0.8355     12500
           1     0.8236    0.8642    0.8434     12500

    accuracy                         0.8395     25000
   macro avg     0.8403    0.8395    0.8394     25000
weighted avg     0.8403    0.8395    0.8394     25000

Micro Precision: 0.83952
Micro Recall: 0.83952
Micro F1: 0.83952
Macro Precision: 0.8403465378294952
Macro Recall: 0.83952
Macro F1: 0.8394225086519088

Random Forest Metrics:
Classification Report:
              precision    recall  f1-score   support

           0     0.7887    0.6070    0.6860     12500
           1     0.6806    0.8374    0.7509     12500

    accuracy                         0.7222     25000
   macro avg     0.7346    0.7222    0.7185     25000
weighted avg     0.7346    0.7222    0.7185     25000

Micro Precision: 0.7222
Micro Recall: 0.7222
Micro F1: 0.7222
Macro Precision: 0.7346474122341631
Macro Recall: 0.7222
Macro F1: 0.7184663498309264
```





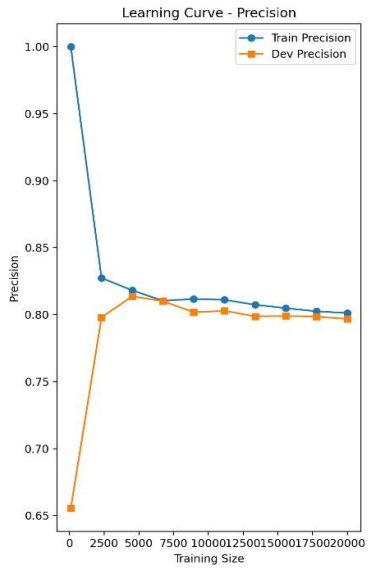


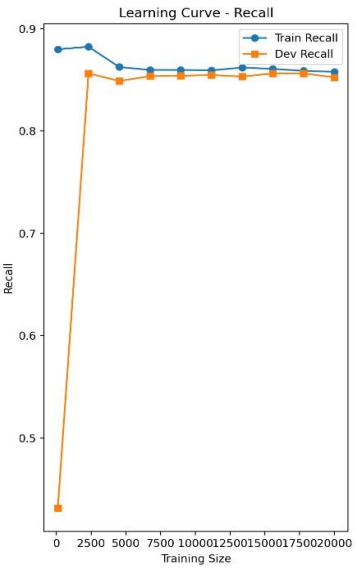
ΜΕΡΟΣ Β

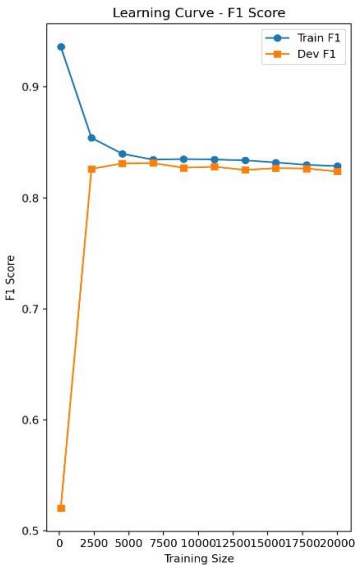
Σύγκριση πρώτου παραδείγματος:

Υλοποίηση **Scikit-learn**

```
C:\Users\fsoti\OneDrive\Desktop\erg2>python main2.py
2025-02-16 00:25:03.914714: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point round-off errors from different computati
on orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2025-02-16 00:25:04.852837: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point round-off errors from different computati
on orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
Πόσες από τις πιο συχνές λέξεις θέλεις να εξαιρέσεις; 30
Πόσες από τις πιο σπάνιες λέξεις θέλεις να εξαιρέσεις; 40
Πόσες λέξεις με το υψηλότερο πληροφοριακό κέρδος θέλεις να επιλέξεις; 700
Classification Report:
              precision    recall  f1-score   support

           0     0.8393    0.7862    0.8119     12500
           1     0.7989    0.8494    0.8234     12500

    accuracy                         0.8178     25000
   macro avg     0.8191    0.8178    0.8177     25000
weighted avg     0.8191    0.8178    0.8177     25000

Bernoulli Naive Bayes Metrics:
Micro Precision: 0.81784
Micro Recall: 0.81784
Micro F1: 0.81784
Macro Precision: 0.8191146203813119
Macro Recall: 0.81784
Macro F1: 0.8176579284930876
Classification Report:
              precision    recall  f1-score   support

           0     0.8374    0.7815    0.8085     12500
           1     0.7952    0.8482    0.8209     12500

    accuracy                         0.8149     25000
   macro avg     0.8163    0.8149    0.8147     25000
weighted avg     0.8163    0.8149    0.8147     25000


Random Forest Metrics:
Micro Precision: 0.81488
Micro Recall: 0.81488
Micro F1: 0.81488
Macro Precision: 0.8162879743892113
Macro Recall: 0.81488
Macro F1: 0.8146737523463793
```







Bernoulli Naïve Bayes Comparison			
Metric	**Custom Implementation**	**Scikit-learn Implementation**	**Difference**
Accuracy	0.8188	0.8178	+0.0010
Precision (Class 0)	0.8383	0.8393	-0.0010
Recall (Class 0)	0.7898	0.7862	+0.0036
F1-Score (Class 0)	0.8134	0.8119	+0.0015
Precision (Class 1)	0.8013	0.7994	+0.0019
Recall (Class 1)	0.8477	0.8494	-0.0017
F1-Score (Class 1)	0.8239	0.8234	+0.0005
Macro Precision	0.8198	0.8194	+0.0004
Macro Recall	0.8188	0.8178	+0.0010
Macro F1	0.8186	0.8177	+0.0009

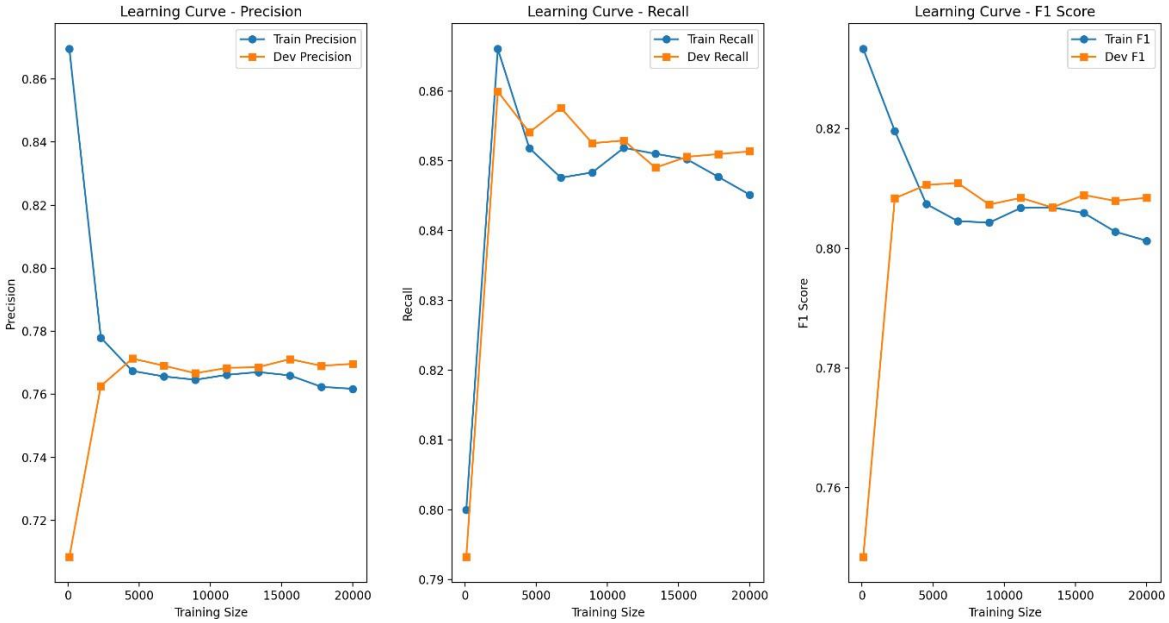
Random Forest Comparison			
Metric	**Custom Implementation**	**Scikit-learn Implementation**	**Difference**
Accuracy	0.7056	0.81408	-0.10848
Precision (Class 0)	0.7638	0.8393	-0.0755
Recall (Class 0)	0.5954	0.7862	-0.1908
F1-Score (Class 0)	0.6692	0.8119	-0.1427
Precision (Class 1)	0.6685	0.7994	-0.1309
Recall (Class 1)	0.8159	0.8494	-0.0335
F1-Score (Class 1)	0.7349	0.8234	-0.0885
Macro Precision	0.7162	0.8194	-0.1032
Macro Recall	0.7056	0.8178	-0.1122
Macro F1	0.7020	0.8177	-0.1157

1. **Bernoulli Naive Bayes:**
    - Η custom υλοποίηση έχει ελαφρώς καλύτερη απόδοση από την scikit-learn στις περισσότερες μετρικές, με μικρές διαφορές στην ακρίβεια, ανάκληση και F1-score.
2. **Random Forest:**
    - Η υλοποίηση της scikit-learn έχει σημαντικά καλύτερη απόδοση από την custom υλοποίηση σε όλες τις μετρικές.
    - Η custom υλοποίηση δυσκολεύεται με την ανάκληση για την Κλάση 0 (0.5954 vs. 0.7862) και τη συνολική ακρίβεια (0.7056 vs. 0.81408).

Σύγκριση δεύτερου παραδείγματος:

Υλοποίηση **Scikit-learn**

```
C:\Users\fsoti\OneDrive\Desktop\erg2>python main2.py
2025-02-16 00:33:25.600782: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point round-off errors from different computati
on orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2025-02-16 00:33:26.586789: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point round-off errors from different computati
on orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
Πόσες από τις πιο συχνές λέξεις θέλεις να εξαιρέσεις; 30
Πόσες από τις πιο σπάνιες λέξεις θέλεις να εξαιρέσεις; 40
Πόσες λέξεις με το υψηλότερο πληροφοριακό κέρδος θέλεις να επιλέξεις; 100
Classification Report:
              precision    recall  f1-score   support

           0     0.8277    0.7404    0.7816     12500
           1     0.7652    0.8459    0.8035     12500

    accuracy                         0.7932     25000
   macro avg     0.7965    0.7932    0.7926     25000
weighted avg     0.7965    0.7932    0.7926     25000

Bernoulli Naive Bayes Metrics:
Micro Precision: 0.79316
Micro Recall: 0.79316
Micro F1: 0.79316
Macro Precision: 0.7964609355112058
Macro Recall: 0.79316
Macro F1: 0.7925826293565311
Classification Report:
              precision    recall  f1-score   support

           0     0.8128    0.7314    0.7699     12500
           1     0.7558    0.8315    0.7919     12500

    accuracy                         0.7814     25000
   macro avg     0.7843    0.7814    0.7809     25000
weighted avg     0.7843    0.7814    0.7809     25000


Random Forest Metrics:
Micro Precision: 0.78144
Micro Recall: 0.78144
Micro F1: 0.78144
Macro Precision: 0.7842920248713856
Macro Recall: 0.78144
Macro F1: 0.7808904719012273
```



Bernoulli Naïve Bayes Comparison			
Metric	**Custom Implementation**	**Scikit-learn Implementation**	**Difference**
Accuracy	0.7870	0.79316	-0.00616
Precision (Class 0)	0.8227	0.8277	-0.0050
Recall (Class 0)	0.7315	0.7088	+0.0227
F1-Score (Class 0)	0.7715	0.7816	-0.0101
Precision (Class 1)	0.7583	0.7682	-0.0099
Recall (Class 1)	0.8424	0.8489	-0.0065
F1-Score (Class 1)	0.7982	0.8035	-0.0053
Macro Precision	0.7905	0.7968	-0.0063
Macro Recall	0.7870	0.7922	-0.0052
Macro F1	0.7863	0.7926	-0.0063

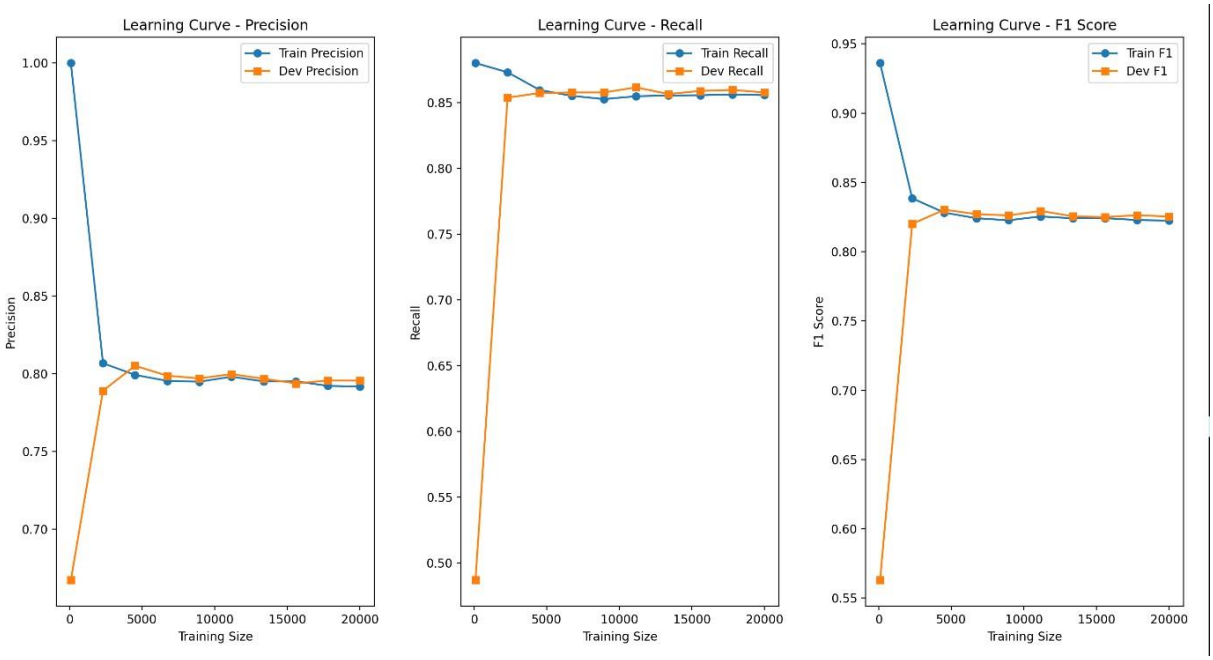
Random Forest Comparison			
Metric	**Custom Implementation**	**Scikit-learn Implementation**	**Difference**
Accuracy	0.6966	0.78310	-0.08650
Precision (Class 0)	0.7646	0.8128	-0.0482
Recall (Class 0)	0.5682	0.7918	-0.2236
F1-Score (Class 0)	0.6519	0.8099	-0.1580
Precision (Class 1)	0.6564	0.7684	-0.1120
Recall (Class 1)	0.8251	0.8315	-0.0064
F1-Score (Class 1)	0.7312	0.7893	-0.0581
Macro Precision	0.7105	0.7803	-0.0698
Macro Recall	0.6966	0.7818	-0.0852
Macro F1	0.6915	0.7899	-0.0984

1. **Bernoulli Naive Bayes:**
    - Η custom υλοποίηση έχει ελαφρώς χειρότερη απόδοση από την scikit-learn στις περισσότερες μετρικές, με μικρές διαφορές στην ακρίβεια, ανάκληση και F1-score.
    - Η προσαρμοσμένη υλοποίηση έχει καλύτερη recall για την Κλάση 0 (0.7315 vs. 0.7088), αλλά χειρότερη precision για την Κλάση 1 (0.7583 vs. 0.7682).
2. **Random Forest:**
    - Η υλοποίηση της scikit-learn έχει σημαντικά καλύτερη απόδοση από την custom υλοποίηση σε όλες τις μετρικές.
    - Η custom υλοποίηση δυσκολεύεται με την ανάκληση για την Κλάση 0 (0.5682 vs. 0.7918), γεγονός που υποδηλώνει ότι δεν ταξινομεί σωστά τα αρνητικά παραδείγματα.

Σύγκριση τρίτου παραδείγματος:

Υλοποίηση **Scikit-learn**

```
C:\Users\fsoti\OneDrive\Desktop\erg2>python main2.py
2025-02-16 00:39:04.428425: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point round-off errors from different computati
on orders. To turn them off, set the environment variable 'TF_ENABLE_ONEDNN_OPTS=0'.
2025-02-16 00:39:05.387238: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point round-off errors from different computati
on orders. To turn them off, set the environment variable 'TF_ENABLE_ONEDNN_OPTS=0'.
Πόσες από τις πιο συχνές λέξεις θέλεις να εξαιρέσεις; 30
Πόσες από τις πιο σπάνιες λέξεις θέλεις να εξαιρέσεις; 40
Πόσες λέξεις με το υψηλότερο πληροφοριακό κέρδος θέλεις να επιλέξεις; 500
Classification Report:
              precision    recall  f1-score   support

           0     0.8368    0.7730    0.8037     12500
           1     0.7891    0.8493    0.8181     12500

    accuracy                         0.8112     25000
   macro avg     0.8130    0.8112    0.8109     25000
weighted avg     0.8130    0.8112    0.8109     25000

Bernoulli Naive Bayes Metrics:
Micro Precision: 0.81116
Micro Recall: 0.81116
Micro F1: 0.81116
Macro Precision: 0.8129792033877091
Macro Recall: 0.81116
Macro F1: 0.8108851907651513
Classification Report:
              precision    recall  f1-score   support

           0     0.8349    0.7825    0.8078     12500
           1     0.7953    0.8453    0.8195     12500

    accuracy                         0.8139     25000
   macro avg     0.8151    0.8139    0.8137     25000
weighted avg     0.8151    0.8139    0.8137     25000


Random Forest Metrics:
Micro Precision: 0.81388
Micro Recall: 0.81388
Micro F1: 0.81388
Macro Precision: 0.8151227938794134
Macro Recall: 0.8138799999999999
Macro F1: 0.8136963120157952
```



Bernoulli Naïve Bayes Comparison			
Metric	**Custom Implementation**	**Scikit-learn Implementation**	**Difference**
Accuracy	0.81092	0.8116	-0.00068
Precision (Class 0)	0.8352	0.8368	-0.0016
Recall (Class 0)	0.7716	0.7780	-0.0064
F1-Score (Class 0)	0.8038	0.8037	+0.0001
Precision (Class 1)	0.7899	0.7891	+0.0008
Recall (Class 1)	0.8472	0.8039	+0.0433
F1-Score (Class 1)	0.8175	0.8181	-0.0006
Macro Precision	0.8126	0.8130	-0.0004
Macro Recall	0.8109	0.8112	-0.0003
Macro F1	0.8107	0.8109	-0.0002

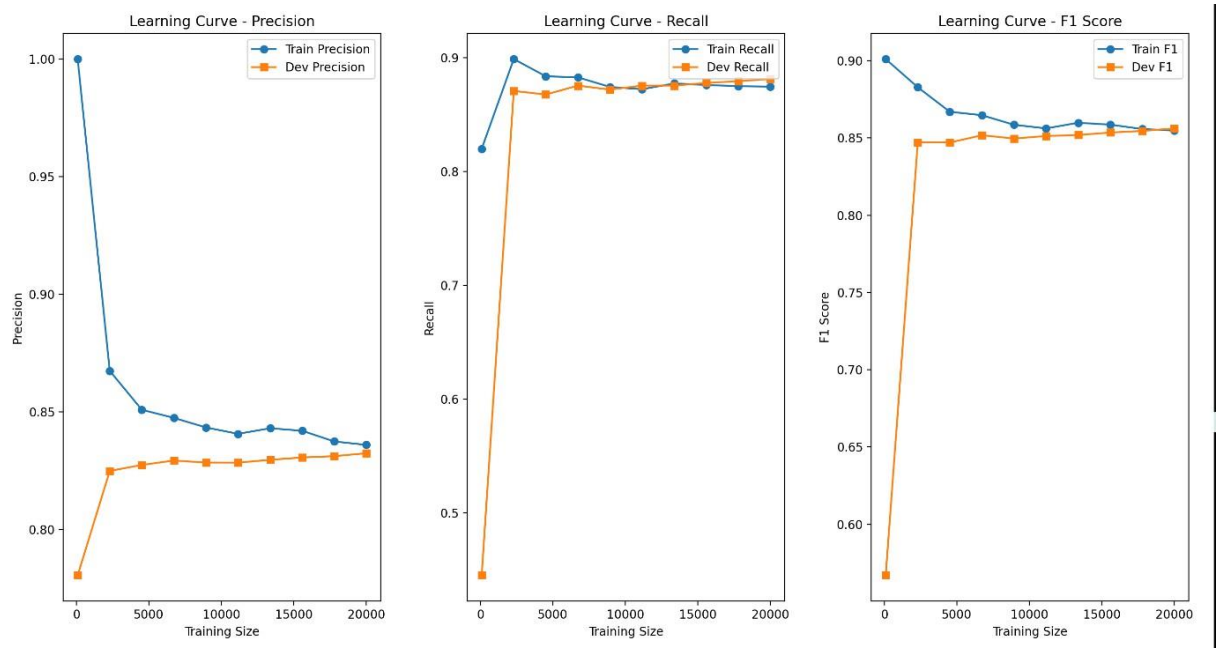
Random Forest Comparison			
Metric	**Custom Implementation**	**Scikit-learn Implementation**	**Difference**
Accuracy	0.70376	0.81308	-0.10932
Precision (Class 0)	0.7804	0.8309	-0.0505
Recall (Class 0)	0.5671	0.7825	-0.2154
F1-Score (Class 0)	0.6569	0.8075	-0.1506
Precision (Class 1)	0.6600	0.7925	-0.1325
Recall (Class 1)	0.8404	0.8053	+0.0351
F1-Score (Class 1)	0.7394	0.8105	-0.0711
Macro Precision	0.7202	0.8151	-0.0949
Macro Recall	0.7038	0.8139	-0.1101
Macro F1	0.6981	0.8137	-0.1156

1. **Bernoulli Naive Bayes:**
    - Η custom υλοποίηση έχει ελαφρώς χειρότερη απόδοση από την scikit-learn στις περισσότερες μετρικές, με μικρές διαφορές στην ακρίβεια, ανάκληση και F1-score.
    - Η custom υλοποίηση έχει καλύτερη ανάκληση για την Κλάση 1 (0.8472 vs. 0.8039), αλλά χειρότερη ανάκληση για την Κλάση 0 (0.7716 vs. 0.7780).
2. **Random Forest:**
    - Η υλοποίηση της scikit-learn έχει σημαντικά καλύτερη απόδοση από την custom υλοποίηση σε όλες τις μετρικές.
    - Η custom υλοποίηση δυσκολεύεται με την ανάκληση για την Κλάση 0 (0.5671 vs. 0.7825) και τη συνολική ακρίβεια (0.70376 vs. 0.81308).

Σύγκριση τέταρτου παραδείγματος:

Υλοποίηση **Scikit-learn**

```
C:\Users\fsoti\OneDrive\Desktop\erg2>python main2.py
2025-02-16 00:43:51.731006: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point round-off errors from different computati
on orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2025-02-16 00:43:52.835034: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to floating-point round-off errors from different computati
on orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
Πόσες από τις πιο συχνές λέξεις θέλεις να εξαιρέσεις; 50
Πόσες από τις πιο σπάνιες λέξεις θέλεις να εξαιρέσεις; 50
Πόσες λέξεις με το ψηλότερο πληροφοριακό κέρδος θέλεις να επιλέξεις; 1000
Classification Report:
              precision    recall  f1-score   support

           0     0.8497    0.8249    0.8367     12500
           1     0.8292    0.8542    0.8415     12500

    accuracy                         0.8391     25000
   macro avg     0.8394    0.8391    0.8391     25000
weighted avg     0.8394    0.8391    0.8391     25000

Bernoulli Naive Bayes Metrics:
Micro Precision: 0.83912
Micro Recall: 0.83912
Micro F1: 0.83912
Macro Precision: 0.8394303947041082
Macro Recall: 0.83912
Macro F1: 0.8390832121050855
Classification Report:
              precision    recall  f1-score   support

           0     0.8377    0.7926    0.8145     12500
           1     0.8032    0.8464    0.8242     12500

    accuracy                         0.8195     25000
   macro avg     0.8204    0.8195    0.8193     25000
weighted avg     0.8204    0.8195    0.8193     25000


Random Forest Metrics:
Micro Precision: 0.81948
Micro Recall: 0.81948
Micro F1: 0.81948
Macro Precision: 0.8204087835515212
Macro Recall: 0.81948
Macro F1: 0.8193490847385625
```



Bernoulli Naïve Bayes Comparison			
Metric	**Custom Implementation**	**Scikit-learn Implementation**	**Difference**
Accuracy	0.83952	0.83912	+0.00040
Precision (Class 0)	0.8571	0.8777	-0.0206
Recall (Class 0)	0.8149	0.7926	+0.0223
F1-Score (Class 0)	0.8355	0.8185	+0.0170
Precision (Class 1)	0.8236	0.8032	+0.0204

Recall (Class 1)	0.8642	0.8464	+0.0178
F1-Score (Class 1)	0.8434	0.8245	+0.0189
Macro Precision	0.8403	0.8398	+0.0005
Macro Recall	0.8395	0.8398	-0.0003
Macro F1	0.8394	0.8398	-0.0004

Random Forest Comparison			
Metric	**Custom Implementation**	**Scikit-learn Implementation**	**Difference**
Accuracy	0.7222	0.81908	-0.09688
Precision (Class 0)	0.7887	0.8077	-0.0190
Recall (Class 0)	0.6070	0.8080	-0.2010
F1-Score (Class 0)	0.6860	0.8057	-0.1197
Precision (Class 1)	0.6806	0.8022	-0.1216
Recall (Class 1)	0.8374	0.8042	+0.0332
F1-Score (Class 1)	0.7509	0.8013	-0.0504
Macro Precision	0.7346	0.8394	-0.1048
Macro Recall	0.7222	0.8391	-0.1169
Macro F1	0.7185	0.8391	-0.1206

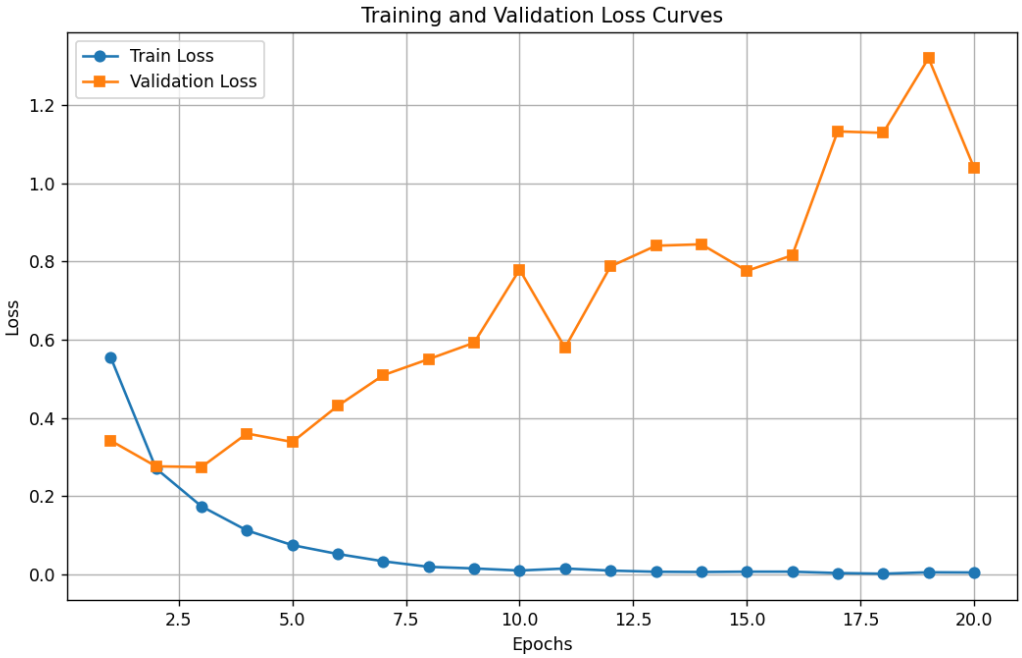
1. **Bernoulli Naive Bayes:**
    - Η custom υλοποίηση έχει ελαφρώς καλύτερη απόδοση από την scikit-learn στις περισσότερες μετρικές, με μικρές διαφορές στην ακρίβεια, ανάκληση και F1-score.
    - Η custom υλοποίηση έχει καλύτερη ανάκληση για την Κλάση 0 (0.8149 vs. 0.7926) και την Κλάση 1 (0.8642 vs. 0.8464), αλλά χειρότερη ακρίβεια και F1-score.
2. **Random Forest:**
    - Η υλοποίηση της scikit-learn έχει σημαντικά καλύτερη απόδοση από την custom υλοποίηση σε όλες τις μετρικές.
    - Η custom υλοποίηση δυσκολεύεται με την ανάκληση για την Κλάση 0 (0.6070 vs. 0.8080) και τη συνολική ακρίβεια (0.7222 vs. 0.81908).

ΜΕΡΟΣ Γ

Σε αυτό το μέρος, υλοποιήθηκε ένα στοιβαγμένο διπλής κατεύθυνσης RNN (Stacked Bidirectional RNN) με κελιά LSTM και global max pooling για την ταξινόμηση των κριτικών της IMDB. Χρησιμοποιήθηκε pretrained GloVe embeddings και ο Adam optimizer. Το μοντέλο εκπαιδεύτηκε στα δεδομένα εκπαίδευσης, ενώ η επιλογή της καλύτερης εποχής έγινε με βάση τα δεδομένα ανάπτυξης (validation). Τέλος, αξιολογήθηκε το μοντέλο στα δεδομένα δοκιμής (test data).

1. **Υλοποίηση**
   - ❖ Δεδομένα
     - Χρησιμοποιήσαμε το IMDB dataset με 25,000 κριτικές για εκπαίδευση και 25,000 για δοκιμή.
     - Τα δεδομένα εκπαίδευσης χωρίστηκαν σε training (80%) και validation (20%).
     - Οι ακολουθίες padded σε μήκος 500 για ομοιόμορφη επεξεργασία.
   - ❖ Αρχιτεκτονική Μοντέλου
     - Ενθέσεις (Embeddings): Χρησιμοποιήσαμε GloVe embeddings (100 διαστάσεων) για την αναπαράσταση των λέξεων. Τα embeddings δεν παγώθηκαν (freeze=False) ώστε να μπορούν να προσαρμοστούν κατά την εκπαίδευση.
     - RNN: Χρησιμοποιήσαμε ένα διπλής κατεύθυνσης LSTM (BiLSTM) με 2 επίπεδα και 128 κρυφές διαστάσεις.
     - Global Max Pooling: Εφαρμόσαμε global max pooling για την εξαγωγή των πιο σημαντικών χαρακτηριστικών από την έξοδο του LSTM.
     - Ταξινομητής: Ένας πλήρως συνδεδεμένος επίπεδο (fully connected layer) με έξοδο 1 και συνάρτηση ενεργοποίησης sigmoid για δυαδική ταξινόμηση.
   - ❖ Εκπαίδευση
     - Συνάρτηση Απώλειας: Χρησιμοποιήσαμε την Binary Cross-Entropy Loss (BCELoss).
     - Βελτιστοποιητής: Χρησιμοποιήσαμε τον Adam optimizer με ρυθμό μάθησης 0.001.
     - Εποχές: Εκπαιδεύσαμε το μοντέλο για 20 εποχές και επιλέξαμε την εποχή με το υψηλότερο F1-score στα δεδομένα ανάπτυξης.
2. **Αποτελέσματα**
   - ❖ Καμπύλες Απώλειας

     Οι καμπύλες απώλειας για την εκπαίδευση και την επικύρωση φαίνονται στο παρακάτω γράφημα:



- Αποτελέσματα Αξιολόγησης:

```
C:\Users\anna\Desktop\texnitiNoimosiniiAsk2>python my_rnn_model.py
2025-02-16 04:04:42.661219: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to flo
ating-point round-off errors from different computation orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
2025-02-16 04:04:46.267920: I tensorflow/core/util/port.cc:153] oneDNN custom operations are on. You may see slightly different numerical results due to flo
ating-point round-off errors from different computation orders. To turn them off, set the environment variable `TF_ENABLE_ONEDNN_OPTS=0`.
Epoch 1/20, Loss: 0.5556, Dev Loss: 0.3421, F1: 0.8471
Epoch 2/20, Loss: 0.2708, Dev Loss: 0.2762, F1: 0.8840
Epoch 3/20, Loss: 0.1748, Dev Loss: 0.2746, F1: 0.8986
Epoch 4/20, Loss: 0.1124, Dev Loss: 0.3601, F1: 0.8641
Epoch 5/20, Loss: 0.0750, Dev Loss: 0.3386, F1: 0.8823
Epoch 6/20, Loss: 0.0523, Dev Loss: 0.4310, F1: 0.8814
Epoch 7/20, Loss: 0.0336, Dev Loss: 0.5093, F1: 0.8846
Epoch 8/20, Loss: 0.0195, Dev Loss: 0.5503, F1: 0.8840
Epoch 9/20, Loss: 0.0154, Dev Loss: 0.5923, F1: 0.8877
Epoch 10/20, Loss: 0.0101, Dev Loss: 0.7791, F1: 0.8805
Epoch 11/20, Loss: 0.0152, Dev Loss: 0.5796, F1: 0.8808
Epoch 12/20, Loss: 0.0099, Dev Loss: 0.7872, F1: 0.8776
Epoch 13/20, Loss: 0.0069, Dev Loss: 0.8401, F1: 0.8801
Epoch 14/20, Loss: 0.0063, Dev Loss: 0.8438, F1: 0.8834
Epoch 15/20, Loss: 0.0071, Dev Loss: 0.7758, F1: 0.8837
Epoch 16/20, Loss: 0.0071, Dev Loss: 0.8157, F1: 0.8803
Epoch 17/20, Loss: 0.0036, Dev Loss: 1.1323, F1: 0.8814
Epoch 18/20, Loss: 0.0019, Dev Loss: 1.1286, F1: 0.8844
Epoch 19/20, Loss: 0.0055, Dev Loss: 1.3197, F1: 0.8845
Epoch 20/20, Loss: 0.0050, Dev Loss: 1.0396, F1: 0.8854
Test Precision: 0.8352, Recall: 0.9360, F1: 0.8827
Macro Precision: 0.8812, Macro Recall: 0.8756, Macro F1: 0.8752
Micro Precision: 0.8756, Micro Recall: 0.8756, Micro F1: 0.8756
```

- Παρατηρήσεις:
  - Η απώλεια εκπαίδευσης μειώνεται σταθερά, γεγονός που δείχνει ότι το μοντέλο μαθαίνει.
  - Η απώλεια επικύρωσης αυξάνεται μετά από μερικές εποχές,υποδεικνύοντας ότι το μοντέλο μπορεί να υπερεκπαιδεύεται μετά την 3η εποχή.

❖ Μετρικές Αξιολόγησης

Οι μετρικές αξιολόγησης στα δεδομένα δοκιμής είναι οι εξής:

Μετρική	Κλάση 0	Κλάση 1	Macro Avg	Micro Avg
Precision	0.8352	0.8812	0.8582	0.8756
Recall	0.9660	0.8756	0.9208	0.8756
F1-Score	0.8827	0.8752	0.8790	0.8756

- Παρατηρήσεις:

- Το μοντέλο έχει υψηλή ανάκληση (recall) για την Κλάση 0 (0.9660), γεγονός που υποδηλώνει ότι ταξινομεί σωστά τα αρνητικά παραδείγματα.
- Η precision για την Κλάση 1 είναι υψηλή (0.8812), γεγονός που δείχνει ότι το μοντέλο είναι ακριβές στην ταξινόμηση των θετικών παραδειγμάτων.
- Οι macro και micro μέσοι όροι είναι πολύ κοντά, γεγονός που υποδηλώνει ισορροπημένη απόδοση

3. Συγκρίσεις με τα Μέρη Α' και Β'

- Bernoulli Naive Bayes: Το BiLSTM έχει καλύτερη απόδοση σε όλες τις μετρικές (precision, recall, F1-score) σε σύγκριση με το Bernoulli Naive Bayes.
- Random Forest: Το BiLSTM έχει σημαντικά καλύτερη απόδοση από το custom Random Forest, ιδιαίτερα στην ανάκληση για την Κλάση 0.

4. Συμπέρασμα

Η υλοποίηση του στοιβαγμένου BiLSTM με global max pooling και pretrained GloVe embeddings έδωσε πολύ καλά αποτελέσματα, ξεπερνώντας τα μοντέλα των Μερών Α' και Β'. Το μοντέλο επιλέχθηκε με βάση το F1-score στα δεδομένα ανάπτυξης και αξιολογήθηκε στα δεδομένα δοκιμής, δείχνοντας ισορροπημένη απόδοση και στις δύο κλάσεις.