Δημοσθένης Γιαννακόπουλος
3170027

Αλγόριθμοι Μηχανικής Μάθησης
Σύστημα που κατατάσσει κριτικές ταινιών σε θετικές ή αρνητικές.
Χρησιμοποιήθηκε το σύνολο δεδομένων IMDB.

Υλοποιήθηκε ο Αφελής Ταξινομητής Bayes (Naive Bayes), σε πολυμεταβλητή μορφή Bernoulli. Λαμβάνεται υπόψη μόνο η παρουσία ή απουσία μιας λέξης σε μια κριτική και όχι ο αριθμός των φορών που εμφανίζεται.

Ο αλγόριθμος υλοποιήθηκε δύο φορές: μία με έτοιμες συναρτήσεις από τη βιβλιοθήκη scikit-learn και μία με δική μου υλοποίηση.
Για κάθε υλοποίηση καταγράφεται η αξιολόγηση ως εξής:

• Καμπύλες μάθησης που δείχνουν αποτελέσματα ακρίβειας (precision) και ανάκλησης (recall) για μία από τις δύο κατηγορίες, στα δεδομένα εκπαίδευσης (training data, όσα έχουν χρησιμοποιηθεί σε κάθε επανάληψη) και ανάπτυξης (development data, πάντα όλα τα δεδομένα ανάπτυξης), συναρτήσει του πλήθους των παραδειγμάτων εκπαίδευσης που χρησιμοποιούνται σε κάθε επανάληψη του πειράματος.

• Πίνακες με αποτελέσματα ακρίβειας (precision), ανάκλησης (recall) και F1 για κάθε μία από τις δύο κατηγορίες, καθώς και μέσους όρους (micro και macro-averaged), στα δεδομένα αξιολόγησης (test data), όταν χρησιμοποιούνται όλα τα δεδομένα εκπαίδευσης.

Κάθε κείμενο κριτικής παριστάνεται ως ένα διάνυσμα ιδιοτήτων με τιμές 0 ή 1, οι οποίες δείχνουν ποιες λέξεις ενός λεξιλογίου περιέχονται στο κείμενο.
Το λεξιλόγιο κατασκευάζεται παραλείποντας αρχικά τις n πιο συχνές και τις k πιο σπάνιες λέξεις των κειμένων εκπαίδευσης, θεωρώντας ότι η συχνότητα μιας λέξης ισούται με το πλήθος των κειμένων εκπαίδευσης στα οποία εμφανίζεται. Από τις λέξεις των δεδομένων εκπαίδευσης που απομένουν, επιλέγονται ως λέξεις του λεξιλογίου οι m λέξεις με το υψηλότερο πληροφοριακό κέρδος (information gain).
Το λεξιλόγιο είναι το ίδιο και για τις δύο υλοποιήσεις του αλγορίθμου μάθησης.

Σημαντικό: Για να τρέξει ο κώδικας τοπικά απαιτούνται οι σωστές εκδόσεις των πακέτων. Στο αρχείο instructions περιγράφεται τι χρειάζεται.

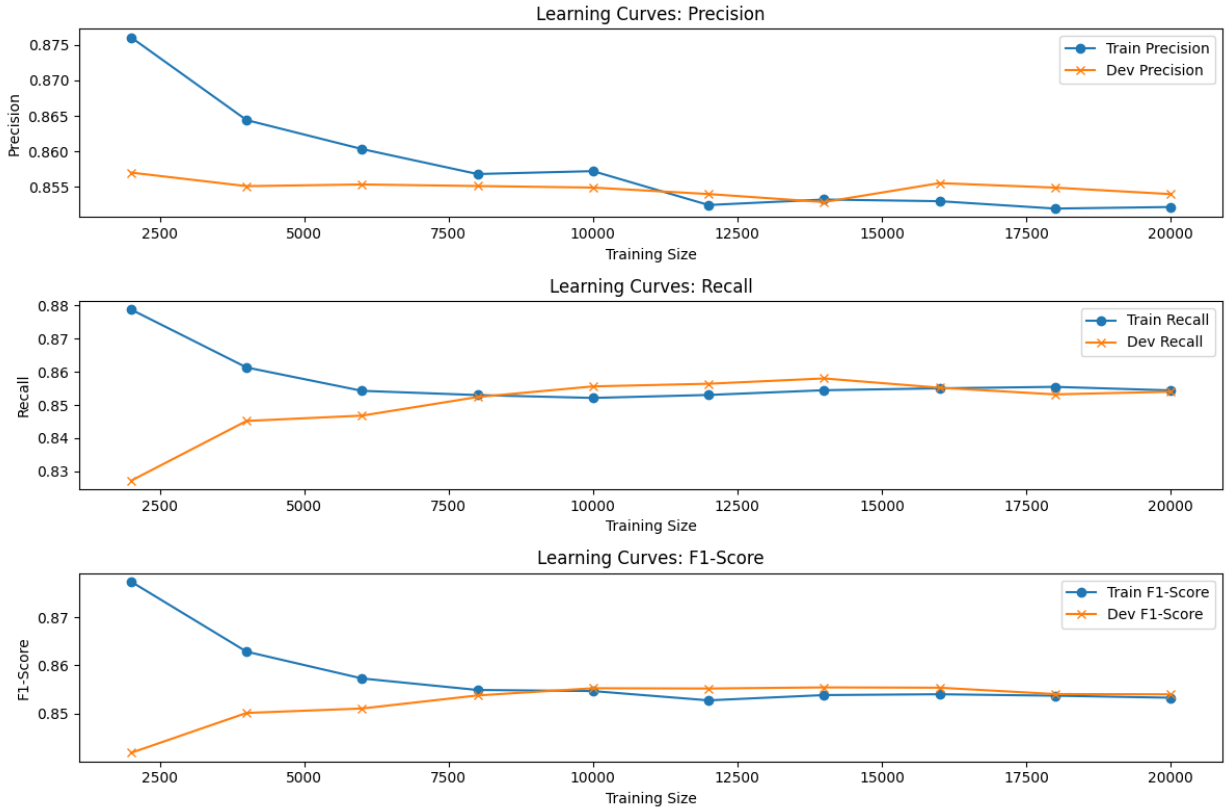
Εκτέλεση του κώδικα:
py set.py → Δημιουργία του τελικού λεξιλογίου στο αρχείο vocab.txt
py 1.py → Χρήση της δικής μου υλοποίησης
py 2.py → Χρήση της έτοιμης υλοποίησης

Τιμές υπερπαραμέτρων:
Μέγεθος λεξιλογίου: m = 456 (m_threshold = 0.0016, δηλαδή διαγράφηκαν όσες λέξεις είχαν μικρότερο information gain από 0.0016).
Κατώφλι συχνότητας εμφάνισης λέξεων: k = 34, διαγράφηκαν όσες λέξεις εμφανίζονταν λιγότερο από

34 φορές.
Κατώφλι συχνότητας εμφάνισης λέξεων: n = 100, διαγράφηκαν οι 100 συχνότερα εμφανιζόμενες λέξεις.

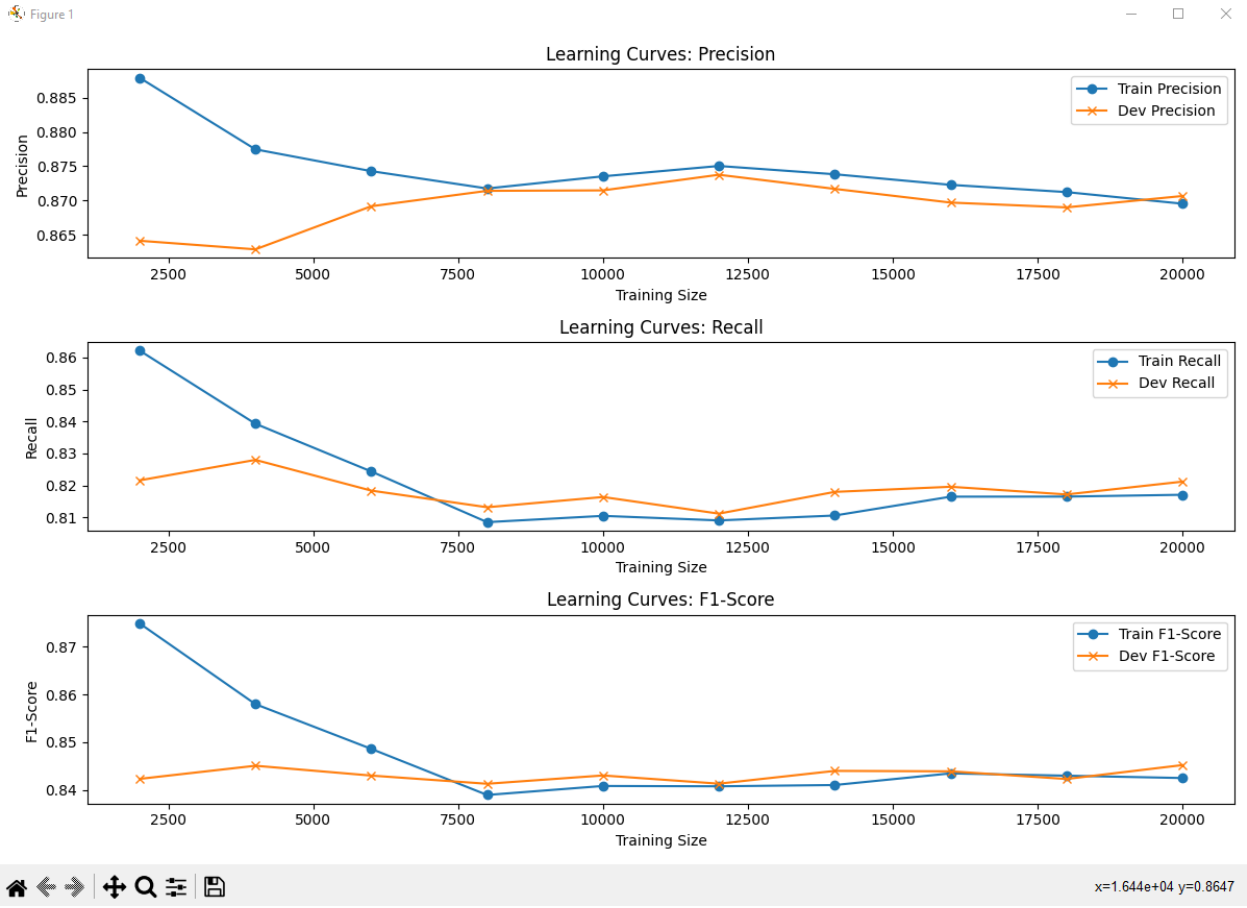
Οι υπερπαράμετροι αυτές επιλέχθηκαν από ένα σύνολο προτεινόμενων τιμών και τελικά χρησιμοποιήθηκαν εκείνες που απέδιδαν καλύτερα στην πράξη.

Β) Με έτοιμες υλοποιήσεις από τη βιβλιοθήκη Scikit-learn



Metric	Class 0	Class 1	Micro-averaged	Macro-averaged
Precision	0.8419	0.8507	0.8462	0.8463
Recall	0.8526	0.8399	0.8462	0.8462
F1-Score	0.8472	0.8453	0.8462	0.8462

A) Με δική μου υλοποίηση του Naive Bayes



Metric	Class 0	Class 1	Micro-averaged	Macro-averaged
Precision	0.8617	0.8278	0.8439	0.8448
Recall	0.8193	0.8686	0.8439	0.8439
F1-Score	0.8400	0.8477	0.8439	0.8438

Παραλείψεις: Οι υπερπαράμετροι δεν υπολογίστηκαν με τη μέθοδο των δεδομένων ανάπτυξης (development data), αλλά «με το μάτι».

Συμπέρασμα: Μια χαρά δουλεύει και η δική μου υλοποίηση