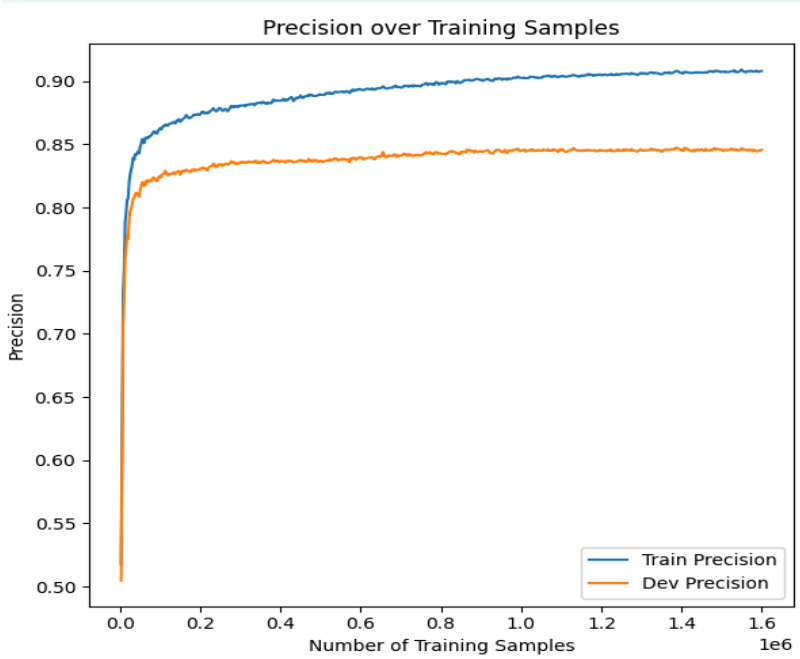
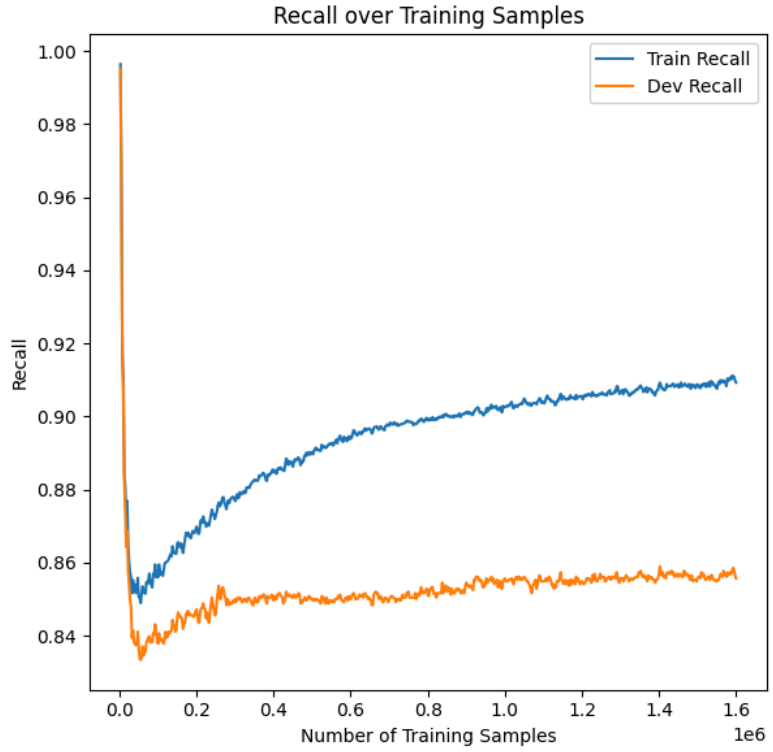
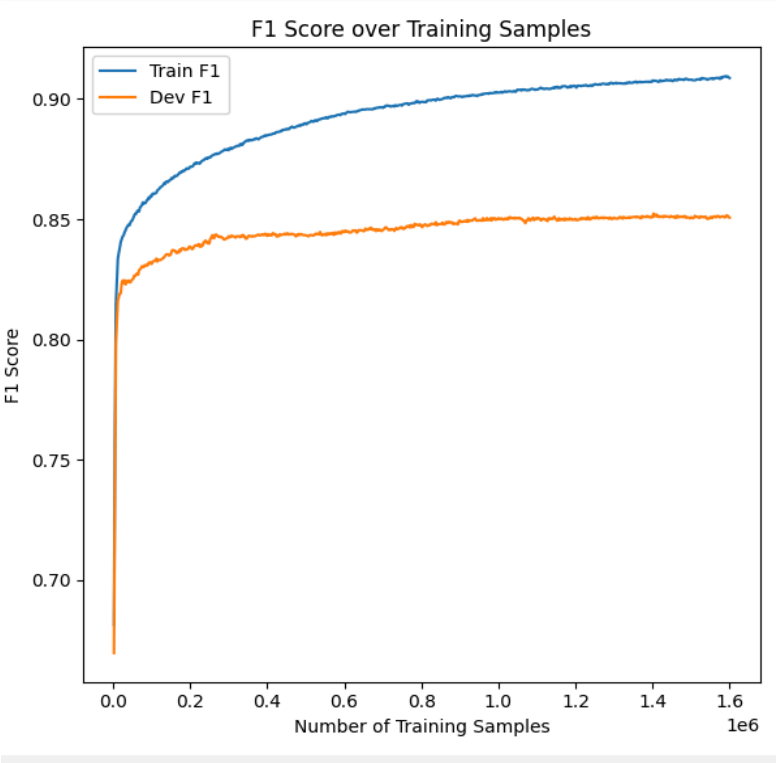
Αποτελέσματα Logistical Regression:

Λογιστική παλινδρόμηση: Αφού δημιουργηθεί το vocabulary με βάση την αντίστοιχη παράγραφο που εξηγεί την διαδικασία χρησιμοποιεί την μέθοδο convert_to_vector ώστε να μπορεί να αναπαραστήσει τα δεδομένα για την λειτουργία του. Το vector για κάθε λέξη του vocabulary έχει αντιστοιχιστεί από ένα index. Στην αρχή όλα τα index είναι 0. Για κάθε review, κάθε λέξη του που εμπεριέχεται στο vocabulary κάμει την θέση της στο αντίστοιχο vector του review 1. Για κάθε review, δημιουργείται παράλληλα και ένα label το οποίο δείχνει την κατηγορία που ανήκει (0 αρνητικό , 1 θετικό) και θα χρησιμοποιηθεί για τους υπολογισμούς. Τα δεδομένα στο τελικό στάδιο προετοιμασίας γίνονται shuffle για να υπάρχει μια ομοιόμορφη κατανομή αφού με τον αρχικό τρόπο έχουμε όλα τα positive ακολουθούμενα από όλα τα negative. Φροντίζουμε παράλληλα να μην χάνεται η αντιστοιχία με τα labels καθώς άμα δεν υπάρχει η αντιστοιχία πραγματικότητας με πρόβλεψης, δεν θα είναι ακριβής κανένας υπολογισμός. Το 80% το κρατάμε για τα δεδομένα εκπαίδευσης(training_data) και το υπόλοιπο 20% για τα δεδομένα ανάπτυξης (dev data). Στο σημείο αυτό καλείται η μέθοδος train για να εκπαιδεύσει τα δεδομένα και να αρχίσει να βελτιώνει τις προβλέψεις του. Στην αρχή της μεθόδου του train δημιουργεί ένα array ίσο με το μέγεθος του vocabulary και αρχικοποιεί τα βάρη κάθε λέξης την πρώτη φορά με τυχαίο τρόπο, ο οποίος με τις επαναλήψεις θα πλησιάσει την πραγματικότητα. Κάθε index στο array weights αντιστοιχεί στο index του array vectors. Έπειτα για κάθε εποχή υπολογίζει το αποτέλεσμα της σιγμοειδής συνάρτησης το οποίο χρησιμοποιείται για την πρόβλεψη του αποτελέσματος. Με βάση την τιμή της πρόβλεψης υπολογίζουμε την απώλεια και ενημερώνουμε τα βάρη ανάλογα. Στο τέλος προσθέτουμε ομαλοποίηση και ενημερώνουμε τα δεδομένα που είναι απαραίτητα για το διάγραμμα. Στην τελική φάση πραγματοποιείται αξιολόγηση του αλγορίθμου για να δούμε το πόσο καλά τα καταφέρνει με βάση την εκπαίδευση που του κάναμε. Κάνουμε προβλέψεις για τα test data αυτή τη φορά και υπολογίζουμε τα αντίστοιχα metrics που χρειάζονται για την αξιολόγηση της απόδοσης. Στην υλοποίηση αυτή ως παράμετροι χρησιμοποιήθηκαν k = 750, n =1000 και m = 30000 και λ = 0.001. Οι τιμές αυτές επιλέχθηκαν μετά από μια σειρά δοκιμών ώστε να επιτευχθεί όσο το δυνατό μεγαλύτερη ακρίβεια γίνεται. Το λ είναι αρκετά μικρό ώστε με την πάροδο των epochs να κάνει σταθερά πρόοδο χωρίς να παρεκκλίνει πολύ. Η ακρίβεια μετά από 100 epochs ανέρχεται περίπου στο 83%.

Αυτά είναι τα διάγραμμα στα δεδομένα μας και συγκεκριμένα στην κατηγορία των positive reviews. Η εκτέλεση διήρκησε συνολικά 80 εποχές με τελικό Avg Loss: 0.3044134530308932 και “σπάσιμο” στο διάγραμμα ανά 25000 δείγματα.


Precision over Training Samples
Precision
Number of Training Samples
1e6
Train Precision
Dev Precision
0.90
0.85
0.80
0.75
0.70
0.65
0.60
0.55
0.50
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6


Recall over Training Samples
Recall
Number of Training Samples
1e6
Train Recall
Dev Recall
1.00
0.98
0.96
0.94
0.92
0.90
0.88
0.86
0.84
0.0
0.2
0.4
0.6
0.8
1.0
1.2
1.4
1.6



Logistic Regression Classification Report:

	precision	**recall**	**f1-score**	**support**
Class 0	0.82	0.84	0.83	12500
Class 1	0.83	0.82	0.83	12500
accuracy			0.83	25000
macro-avg	0.83	0.83	0.83	25000
weighted-avg	0.83	0.83	0.83	25000

Και εδώ είναι τα δεδομένα όσον αφορά τα υπόλοιπα metrics.

Micro-Averaged Precision: 0.82708

Micro-Averaged Recall: 0.82708

Micro-Averaged F1: 0.82708

Macro-Averaged Precision: 0.8272406966490444

Macro-Averaged Recall: 0.82708

Macro-Averaged F1: 0.8270587686276129