Τεχνητή Νοημοσύνη

2η Εργασία

Ομάδα:
Κουτσομπίνας Γιώργος 3150251
Βελαώρας Απόστολος 3180249
Καλδής Αργύριος 3160045

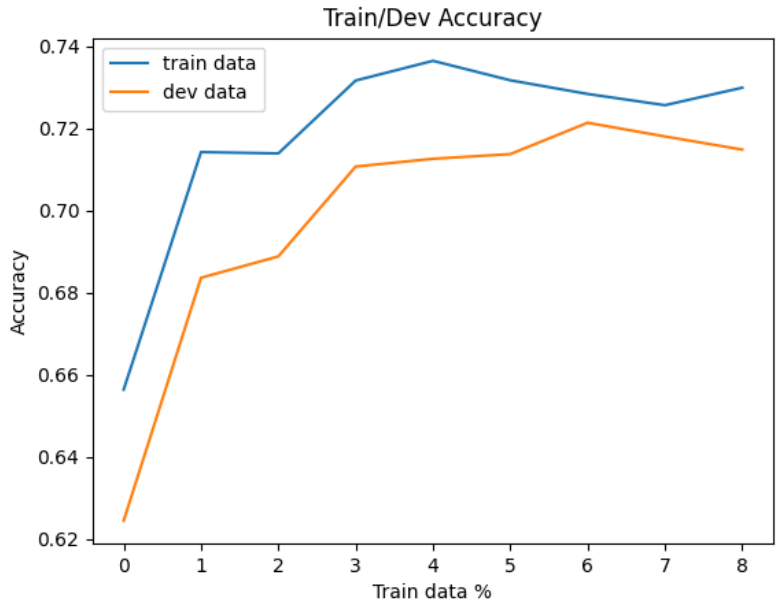
Λογιστική παλινδρόμηση

- data.py: χρησιμοποιείται για την δημιουργία λεξικού και ανάγνωση των .txt αρχείων. Για κάθε λέξη που διαβάζεται, επιστρέφει ένα διάνυσμα με τιμές 0 ή 1 ανάλογα με το αν υπάρχει η λέξη στο λεξικό. Στο τέλος κάθε διανύσματος μπαίνει η τιμή 0 αν το σχόλιο ανήκει στην αρνητική κατηγορία, ή η τιμή 1 αν το σχόλιο ανήκει στην θετική κατηγορία
- logistic_regression.py: περιλαμβάνει τις μεθόδους για την υλοποίηση της λογιστικής παλινδρόμησης:
    - logistic_function(x, w): σιγμοειδής συνάρτηση, επιστρέφει έναν αριθμό από 0 έως 1.
    - train(train_total): Διαβάζει τα σχόλια από τον φάκελο train, τα χωρίζει σε train και dev και εκτελεί την στοχαστική ανάβαση κλίσης.
    - test(): Διαβάζει τα σχόλια από τον φάκελο test και δοκιμάζει τα βάρη σε αυτά μέσω της σιγμοειδούς συνάρτησης για να τα αξιολογήσει.
    - test_external(path): Αξιολογεί ένα μεμονωμένο σχόλιο.
    - cost(w, l_val): Επιστρέφει το κανονικοποιημένο συνολικό κόστος.
    - single_cost(x, l_val): Επιστρέφει το κανονικοποιημένο κόστος ενός μεμονομένου σχολίου.

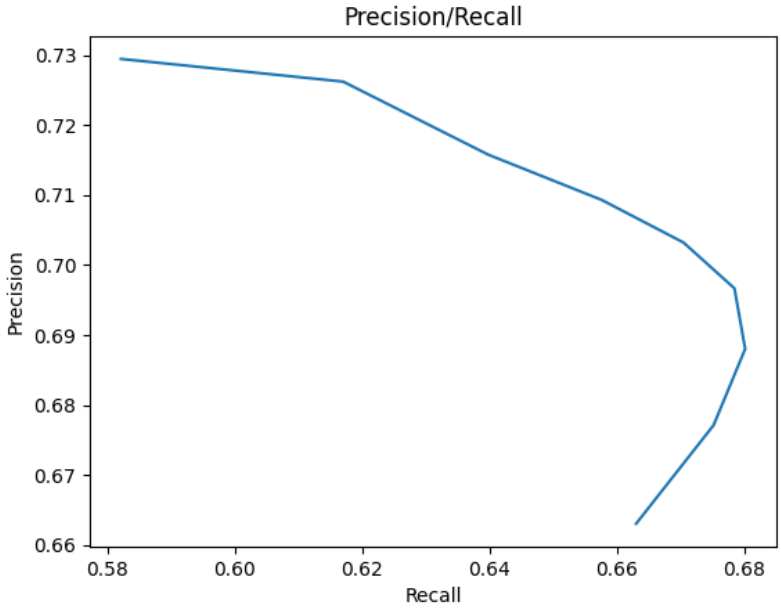
- evaluate(pos): Επιστρέφει την κατηγορία στην οποία ανήκει ένα σχόλιο train/test.
- evaluate_dev(pos): Επιστρέφει την κατηγορία στην οποία ανήκει ένα σχόλιο dev.
- select_model(): Διαλέγει τα βάρη με το μικρότερο error rate στα dev δεδομένα, σε περίπτωση που δοθούν πάνω από μια λ τιμές.
- converges(old, new): Ελέγχει αν συγκλίνει το κόστος.
- get_stats(): Επιστρέφει διαγράμματα με καμπύλες μάθησης.

- Η επιλογή του όρου κανονικοποίησης λ = 4 έγινε μετά από παρακολούθηση των καμπυλών μάθησης και από πολλαπλές δοκιμές σε ποσοστά των δεδομένων εκπαίδευσης και
- Η επιλογή του learning rate a = 0.01 στην στοχαστική κατάβαση κλίσης έγινε μετά από παρακολούθηση των τιμών του συνολικού κόστους σε κάθε ενημέρωση βαρών.

Καμπύλη μάθησης ορθότητας:



Καμπύλη μάθησης ακρίβειας/ανάκλησης:



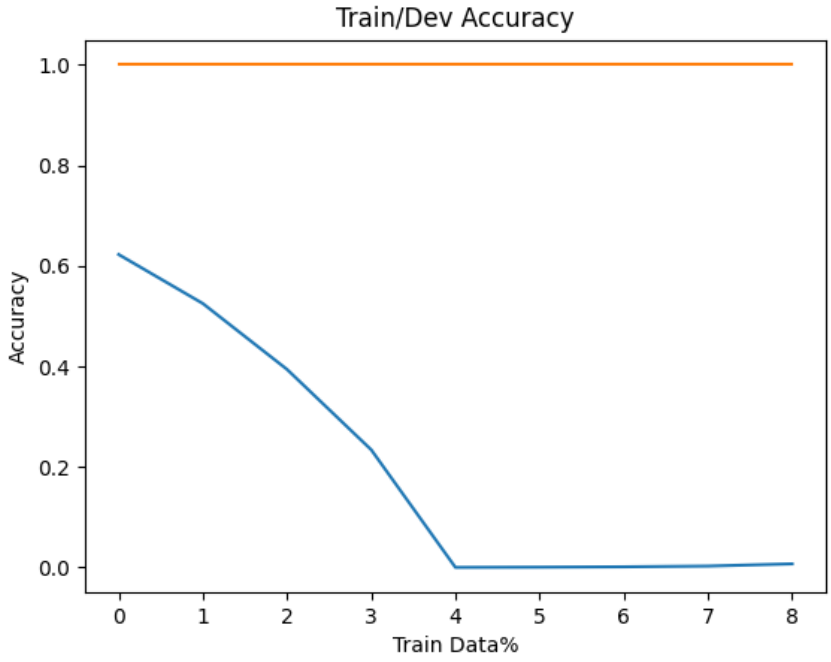
Αφελής ταξινομητής Bayes

- **NayveBayes.py:** Περιέχη την υλοποίηση του αλγορίθμου Αφελης ταξινομητής Bayes.
    - **#TRAIN**
    - **countWords(**) : Μετράει τις λέξεις που περιέχονται στο dictionary.txt
    - **countPosNeg()**: Επιστρέφει τον αριθμό το θετικών και αρνητικών reviews στη βάση (train)
    - **calcPossibility():** υπολογίζει την πιθανότητα P(C=1) k P(C=0) (Οπού 1 θετικό , 0 αρνητικό review)
    - **countWordsPerVector():** Επιστρέφει το άθροισμα τον λέξεων του dictionary για κάθε μια από τις δυο κατηγορίες (Θετικές αρνητικές κριτικές)
    - **calcPossibilityOfWords():** Επιστρέφει τις πιθανότητες κάθε λέξεις για κάθε μια απ τις κατηγορίες (P(X|C=1) , P(X|C=0) οπού X η κάθε λέξη)
    - **summer():** Επιστρέφει το άθροισμα όλων τον λέξεων σε κάθε κατηγορία.

TEST

- **getFilesVector():** Επιστρέφει το vector για ένα δοσμένο review αλλά σε διάνυσμα που μετράει το πλήθος τον λέξεων (όχι μόνο τιμές 0 και 1)
- **getFilesVector2():** Επιστρέφει το vector για ένα δοσμένο review σε διάνυσμα 0 κ 1 (δεν χρησιμοποιείτε , ελάχιστα καλυτέρα αποτελέσματα η προηγουμένη)
- **finalCalc():** Υπολογίζει την τελική πιθανότητα (P(X1|C)*P(X2|C)…..P(Xi|C)*P(C) για I = πλήθος λέξεων και C= 0 ή 1) Με βάση το vector του review υπολογίζει τη πιθανότητα και ανάλογα πόσες φορές έχει εμφανιστεί ή δεδομένη λέξη υπολογίζει την δύναμη της με την Power() και επιστρεφει 0 αν το review είναι αρνητικό ή 1 αν είναι θετικο.
- **Power(Element,N):** Επιστρέφει το Element στη δύναμη N
- **TestResult():** Επιστρέφει ένα άθροισμα με βάση ποσά από τα review ήταν accurate.
- **Accuracy():** Επιστρέφει το ποσοστό ακριβείας των τεστ.

Καμπύλη μάθησης ορθότητας:


Train/Dev Accuracy
1.0
0.8
0.6
0.4
0.2
0.0
Accuracy
0
1
2
3
4
5
6
7
8
Train Data%