ΤΕΧΝΗΤΗ ΝΟΗΜΟΣΥΝΗ

(2η εργασία)

Ον/μο: ΓΕΩΡΓΟΠΟΥΛΟΥ ΚΩΝΣΤΑΝΤΙΝΑ
ΑΜ: 3180029

Α' ΜΕΛΟΣ:

Η main βρίσκεται στην N_B class. Αρχικά δηλώνται μεταβλητές n, m, k οι οποίες θα χρησιμοποιηθούν ως υπερπαράμετροι με n = τις πιο συχνές λέξεις, k = τις πιο σπάνιες λέξεις και m = τις συχνότερες από αυτές. Ύστερα δηλώνονται οι ArrayList A, P-R και Fl για την αποθήκευση των accurasy, precision - recall, f1 - threshold αντίστοιχα(που θα βοηθούσαν σε μια αναπαράσταση αντοίστηχων διαγραμμάτων) και ο πίνακας B για τα χαρακτηριστικά του καλύτερου test. Ξεκινώντας η main με την μέθοδο readVocabulary διαβάζει τις διαθέσιμες λέξεις που υπάρχουν στις αξιολογήσεις (μια εμφάνιση για καθε λέξη) μέσω του imdb.vocab αρχείου. Αφού φτιαχτεί το λεξιλόγιο μέσω της findHyperparametre δοκιμαστικα για κάθε δυνατο n (μεταξύ των τιμών 10, 20, 30, 40, 50, 60, 70, 80, 90 και 100), για κάθε δυνατό k (μεταξύ των τιμών 10, 20, 30, 40, 50, 60, 70, 80, 90 και 100) και για κάθε δυνατό m (μεταξύ των τιμών 2000, 2500, 3000, 3500, 4000, 4500 και 5000) διαβάζοντας όλα τα train βρίσκει σε ποιό συνδυασμό υπερπαραμέτρων το dev ειναι μεγαλύτερο, ώστε να κρατήσει αυτές τις τιμές για να χρησιμοποιηθούν στο πρόγραμμα. Μέσω της μεθόδου Dev κλάσης Dev_Test δημιουργούνται λίστες positive και negative με το 10% των train (άρα 2500 dev με 1250 θετικές και 1250 αρνητικές κριτικές). Συνεχίζοντας εκπαιδεύεται ο αλγόριθμος με Cross Validation (δηλαδή χρησιμοποιώντας διαφορετικό πλήθος train κάθε φορά). Πιο συγκεκριμένα φτιάχνονται Vocabulary λίστες pos και neg οποίες έχουν λίστες με τις λέξεις (vocab), με 1-0 για την ύπαρξη ή οχι μιας λέξης (dian), και μετρητή λέξεων (wordCount), οι οποίες ενημερώνονται και

καλούνται από αντίστοιχους setters και getters. Μέσω της readTrain διαβάζει τα trains. Πιο συγκεκριμένα, μπαίνοντας στο αρχείο labeledBow.feat, ελέγχει την κάθε σειρά ξεχωριστά γιατί αναφέρεται σε διαφορετικό train. Η διαδικασία που ακολουθείται είναι η εξής: αποθηκεύει σε πίνακα string την κάθε γραμμή και κάνοντας slit βλέπει αν πρόκειται για θετική (>=7) ή αρνητική (<=4) βαθμολογία. Για κάθε περίπτωση αγνοεί τα πρώτα 1250, καθώς καθορίζουν τα dev, για τα υπόλοιπα (που καθορίζονται από την επαναληπτική διαδικασία που αναφέρθηκε παραπάνω) ενημερώνει το pos ή neg αντίστοιχα για το εάν βρέθηκε η λέξη και τις φορές εμφάνισής της. Από αυτές τις πληροφορίες αφαιρεί τις υπερπαραμέτρους που βρέθηκαν παραπάνω μέσω της removeHyperparametre, η οποία βρίσκει τις n πιο συχνές (σε κάθε επανάληψη διαγράφεται η πιο συχνή λέξη), τις k πιο σπάνιες (σε κάθε επανάληψη διαγράφεται η πιο σπάνια λέξη) και τις m πιο συχνές λέξεις από αυτές που έμειναν (τις βάζει όλες σε μία merged λίστα, η οποία ταξινομείται και κρατιούνται οι m, και αν στην pos και neg υπάρχει λέξη πέρα από αυτές της merged την διαγράφει), μένοντας έτσι και η pos και η neg με m συχνότερες λέξεις, από τις οποίες έχουν αφαιρεθεί οι n συχνότερς και οι k σπανιότερες. Μέσω της trainVocab μετριέται το συνολικό πλήθος των θετικών και αρνητικών αξιολογήσεων και τοποθετείται στις μεταβλητές totalPos και totalNeg αντίστοιχα. Επιπλέον υπολογίζεται το πλήθος των διαφορετικών λέξεων που μετρήθηκαν και αποθηκεύεται στην total (οι τιμές αυτές επιστρέφονται από αντίστοιχους getters, δηλαδή getTotalPos, getTotalNeg και getTotal αντίστοιχα). Μετά μέσω της Train της Dev_Test κλάσης, δημιουργούνται λίστες positive και negative στις οποίες αποθηκεύεται το review με λέξεις από την readTrainReviews. Ανάλογα με ποιά λίστα από τις δύο γεμίζεται, επιλέγεται και το αντίστοιχο αρχείο από το οποίο θα παραβλέψει τα πρώτα 1250 (γιατί αναφέρονται σε dev). Από τη μέθοδο αυτή επιστρέψεται το review στο οποίο έχει

αποθηκευτεί όλη η κριτική σε string. Στην trainAcc αποθηκεύεται το accuracy που υπολογίζεται από την calculate (σύμφωνα με τον αντίστοιχο τύπο). Για να υπολογιστούν οι παράμετροι του τύπου χρησιμοποιείται η testData, η οποία για να επιστρέψει την τιμή τους υπολογίζει τις πιθανότητες (για θετική και αρνητική αξιολόγιση) του Naive Bayes με La Place (για να μη μηδενιστεί η πιθανότητα σε περιπτωση που δεν εμφανιστει μια λέξη) και με log για να αποφευχθεί underflow και τις συγκρίνει. Αντίστοιχα παίρνει τιμή και η devAcc για τα dev δεδομένα. Για κάθε μια επαναληπτική διαδικασία αποθηκεύονται τα trainAcc και devAcc στον πίνακα A. Για να ολοκληρωθεί η διαδικασία βρίσκεται το καλύτερο accuracy για τα dev και αποθηκεύεται στην bestAcc (και στην bestSize τα πόσα train χρησιμοποιήθηκαν για να μεγιστοποιηθεί το accuracy) η οποία θα χρησιμοποιηθεί για τα test. Γνωρίζοντας πλέον το bestAcc γίνονται τα test κάνοντας την ίδια διαδικασία αλλα αυτή τη φορά εκπαιδεύοντας τόσα train όσα αποθηκεύτηκαν στην bestSize. Αναλυτικότερα, φτιάχνοται λίστες για τις λέξεις μέσω της Vocabulary, διαβάζονται τα bestSize train μέσω της readTrain, όπως έγινε και παραπάνω, αφαιρούνται οι λέξεις που ορίζουν οι υπερπαράμετροι, και φτιάχνονται λίστες για τις θετικές και αρνητικές αξιολογήσεις, όπως επίσης αναφέρθηκε παραπάνω. Τέλος μέσως της calculate υπολογίζονται και αποθηκεύονται: στον B τα accuracy, precision, recall και f1-Score για τα test, για κάθε threshold τα precision και recall στον P-R και το f1-Score-threshold στον F1.

(Τα άλλα δύο μέλη της εργασίες δεν υλοποιήθηκαν)