Τεχνητή Νοημοσύνη 2020-2021

Ομαδική Εργασία 2ο Μέρος

Ομάδα εργασίας:

Βαρέλης Ξενοφών (3170014) **Θωμάς Κωνσταντινίδης (3160074)**

Αλγόριθμος multinomial naive bayes

Ακολουθούν οι κλάσεις που δημιουργήθηκαν για την υλοποίηση του αλγορίθμου και οι μέθοδοι του καθώς και συνοπτική περιγραφή τους.

Κλάση ImportData :Κλάση η οποία χρησιμοποιείται για το διάβασμα των αρχείων .feat και είναι ίδια με αυτήν που χρησιμοποιούμε και στον ID3. Πιο συγκεκριμένα περιέχει:

- Main()
- Loadataset() η οποία με την βοήθεια του Scanner διαβάζει το dataset και αποθηκεύει τα .feat αρχεία με τα reviews σε ένα Arraylist<Hashmap<Integer,Integer>> όπου στο key κάθε φορά βάζει την λέξη και στο value πόσες φορές έχει εμφανιστεί η λέξη αυτή. Επιπλέον αποθηκεύει την βαθμολογία κάθε review σε έναν Arraylist<Integer> και χωρίζει τα train data σε 90% train και 10% development.
- Random() η οποία καλείται για να ανακατέψει τα δεδομένα του dataset .

Κλάση Bayes:Κλάση στην οποία υλοποιείται ο αλγόριθμος του Bayes. Πιο συγκεκριμένα περιέχει:

- Bayes() Κατασκευαστής ο οποίος αρχικοποιεί τα δεδομένα και τις συλλογές.
- PriorProbability() μέθοδος που επιστρέφει την ολική Πιθανότητα ένα review να είναι θετικό ή αρνητικό ανάλογα με τι του ζητάμε και σύμφωνα με τα train data.
- Train() μέθοδος που γίνονται train τα data και δημιουργεί δυο hashmap<Integer,Integer> ,ένα με λέξεις που βρίσκονται στα θετικά review και ένα με λέξεις για τα αρνητικά reviews και αποθηκεύει στο πρώτο όρισμα την λέξη και στο δεύτερο την συχνότητα της λέξης αυτής. Επιπλέον Αποθηκεύει σε δύο μεταβλητές το σύνολο των λέξεων στα θετικά reviews και αντίστοιχα για τα αρνητικά.
- getNegFrequency() μέθοδος που ψάχνει στο NegDictionary την συχνότητα μιας συγκεκριμένης λέξης.
- getPosFrequency() μέθοδος που ψάχνει στο PosDictionary την συχνότητα μιας συγκεκριμένης λέξης.
- Test() μέθοδος που παίρνει σαν είσοδο τα reviews σε ένα Arraylist<Hashmap<Integer,Integer>> και την βαθμολογία του καθενός με έναν Arraylist<Integer>, βρίσκει την δεσμευμένη πιθανότητα ένα review να έιναι για παράδειγμα θετικό δεδομένου ότι περιέχει κάποιο σύνολο λέξεων , και την αποθηκεύει σε ένα Arraylist που περιέχει τις δεσμευμένες πιθανότητες για κάθε review να είναι θετικό και αντίστοιχα για τα αρνητικά. Πιο συγκεκριμένα ο γενικός τύπος για την κάθε δεσμευμένη πιθανότητα είναι: π.χ P(positive/”review”(δηλαδή το σύνολο των λέξεων

του))=P(positive)*P(reviewWord1/positive)^frequencyOfTheWordInTheReview* P(reviewWord2/positive)^frequencyOfTheWordInTheReview)*…..Για να βρούμε την δεσμευμένη πιθανότητα του P(reviewWord1/positive) κάνουμε το εξής P(reviewWord1/positive)=(frequencyOfTheWordInAllTrainData+1)/totalPositiveWords+uniqueWords).Τέλος ανάλογα με την βαθμολογία βρέσκουμε τα truepositive,truenegative,falsepositive,falsenegative από τα οποία παραγονται τα συνολικά accuracy ,recall ,precision και f1score.

Παρακάτω παραθέτονται οι πίνακες για τα **train** και **test**:

Train data για λέξεις από 500 έως 1000 του vocab

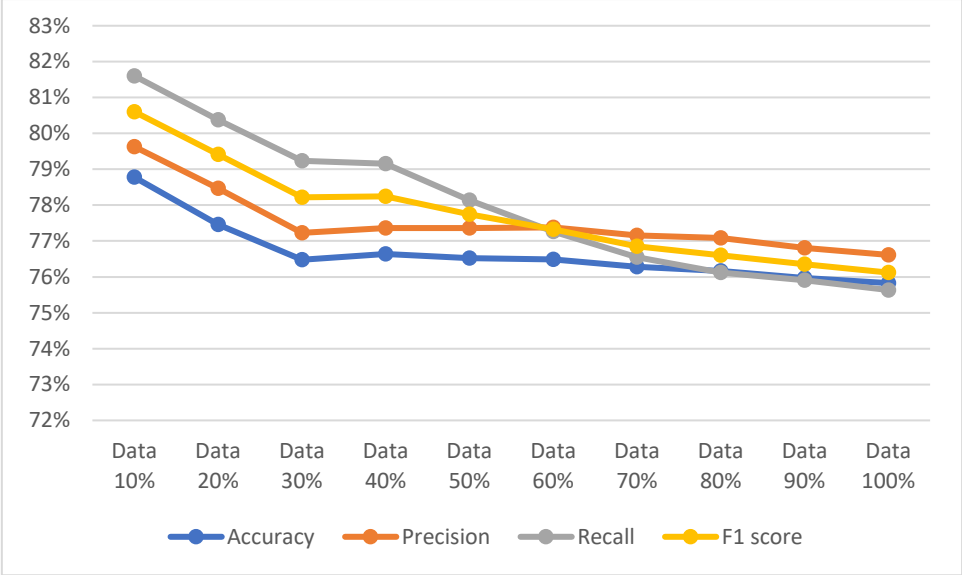
	Data 10%	Data 20%	Data 30%	Data 40%	Data 50%	Data 60%	Data 70%	Data 80%	Data 90%	Data 100%
Accuracy	0.787784 2175657 601 %	0.774632 1890325 457 %	0.764749 5913211 473 %	0.766384 3067320 553 %	0.765225 1448952 296 %	0.764898 2018130 48 %	0.762817 6549264 378 %	0.761647 3473027 195 %	0.759696 8345965 225 %	0.758270 1738742 755 %
Precision	0.796296 2962962 963 %	0.784708 2494969 819 %	0.772282 6086956 521 %	0.773650 7285040 016 %	0.773629 3306424 487 %	0.773777 0764596 185 %	0.771564 0993633 753 %	0.770865 4481391 763 %	0.768091 7827024 907 %	0.766134 7517730 497 %
Recall	0.816006 6006600 66 %	0.803792 2506183 017 %	0.792305 5478115 417 %	0.791517 9508712 996 %	0.781382 7076609 479 %	0.772668 6721100 129 %	0.765448 9164086 687 %	0.761226 4868979 015 %	0.759085 1826727 396 %	0.756345 1776649 747 %
F1 Score	0.806030 9698451 508 %	0.794135 6139279 17 %	0.782165 9556901 058 %	0.782482 3578248 236 %	0.777486 6897659 09 %	0.773222 4770642 202 %	0.768494 3429068 755 %	0.766015 6463701 515 %	0.763561 9242579 325 %	0.761208 4911477 143 %

Test data για λέξεις από 500 έως 1000 του vocab

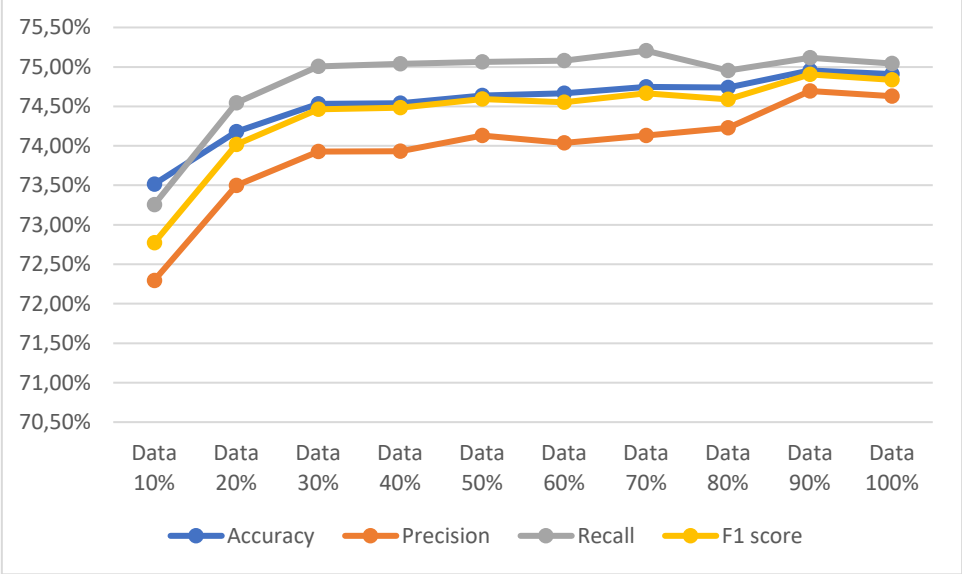
	Data 10%	Data 20%	Data 30%	Data 40%	Data 50%	Data 60%	Data 70%	Data 80%	Data 90%	Data 100%
Accuracy	0.735152 4879614 767 %	0.741773 6757624 398 %	0.745318 3520599 251 %	.7453852 3274478 33 %	0.746388 4430176 565 %	0.746655 9657570 894 %	0.747477 6427424 903 %	0.747391 6532905 297 %	0.749554 1287676 12 %	0.749077 0465489 567 %
Precision	0.722950 8196721 312 %	0.734963 9133921 412 %	0.739280 9587217 044 %	0.739321 3572854 291 %	0.741291 1473314 158 %	0.740360 2401601 067 %	0.741320 2375513 933 %	0.742291 8549763 985 %	0.746948 2313107 012 %	0.746287 8240629 264 %
Recall	0.732558 1395348 837 %	0.745424 9694997 966 %	0.750067 5493109 971 %	0.750405 1863857 374 %	0.750647 2491909 385 %	0.750777 9732106 616 %	0.752056 5403777 083 %	0.749518 3044315 993 %	0.751164 8745519 713 %	0.750443 9063761 098 %
F1 Score	0.727722 7722772 277 %	0.740157 4803149 606 %	0.744635 1931330 474 %	0.744822 0390106 576 %	0.745939 8617141 02 %	0.745532 7153029 692 %	0.746649 7958244 666 %	0.745887 5769502 472 %	0.749050 6187731 761 %	0.748360 0949736 409 %

Παρακάτω παραθέτονται οι καμπύλες για τα **train** και **test**:

Train data για λέξεις από 500 έως 1000 του vocab



Test data για λέξεις από 500 έως 1000 του vocab



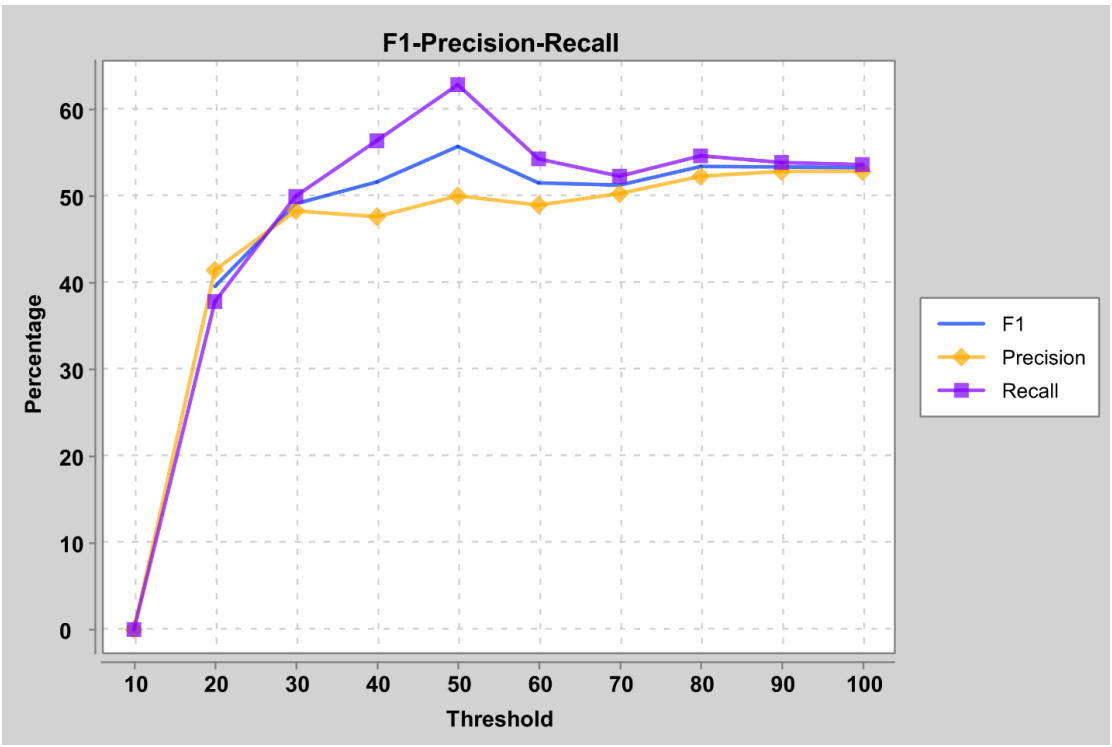
ID3

- Υπερπαράμετροι: m=13100 , n=12900
- Test accuracy

```
Test Accuracy: 52.17481066124695
```

- Καμπύλες (παρακάτω)
- Πίνακες (παρακάτω)
- Οι υπερπαράμετροι αυτοί (m=13100 , n=12900)επιλέχθηκαν ύστερα από την εξέταση πολλών παραδειγμάτων (πχ. m=10000 , n=9800). Τέλος επιλέχθηκαν 200 λέξεις, καθώς οι παραπάνω θέλουν πολλή ώρα για να τις εξετάσει ο αλγόριθμος.


Train-Dev Accuracy
Accuracy
68
66
64
62
60
58
10
20
30
40
50
60
70
80
90
100
Examples Percentage
Develop
Train


F1-Precision-Recall
Percentage
60
50
40
30
20
10
0
10
20
30
40
50
60
70
80
90
100
Threshold
F1
Precision
Recall

	Train	Develop	F1	Precision	Recall
10%	61.827%	48.917%	0	0	0
20%	63.922%	55.225%	58.278%	48.351%	73.333%
30%	66.306%	57.026%	41.484%	39.256%	43.981%
40%	67.966%	58.177%	45.974%	45.664%	46.289%
50%	68.025%	58.481%	45.781%	46.846%	44.763%
60%	67.366%	55.612%	47.274%	46.251%	48.344%
70%	67.119%	57.142%	48.095%	47.757%	48.438%
80%	67.121%	58.159%	50.288%	49.620%	50.974%
90%	67.160%	59.331%	51.187%	50.070%	52.356%
100%	66.171%	59.422%	51.080%	49.553%	52.704%