Βούλγαρης Κωνσταντίνος it9876

HW5

1a.Multinomial Naive Bayes

Υπολογίζω τις πιθανότητες των κλάσεων TV|yes και TV|no

P(TV|yes) = (3/5) = 0,6

P(TV|no) = (2/5) = 0,4

Υπολογίζω τις πιθανότητες των λέξεων στα έγγραφα στις κλάσεις TV|yes και TV|no

P(πρόγραμμα|yes) = (3+1)/(14+8) = 0,182

P(πρόγραμμα|no) = (0+1)/(6+8) = 0,071

P(επεισόδιο|yes) = (5+1)/(14+8) = 0,273

P(επεισόδιο|no) = (1+1)/(6+8) = 0,143

P(σειρά|yes) = (1+1)/(14+8) = 0,091

P(σειρά|no) = (0+1)/(6+8) = 0,071

P(κανάλι|yes) = (3+1)/(14+8) = 0,182

P(κανάλι|no) = (0+1)/(6+8) = 0,071

P(ταινία|yes) = (1+1)/(14+8) = 0,091

P(ταινία|no) = (0+1)/(6+8) = 0,071

P(ειδήσεις|yes) = (1+1)/(14+8) = 0,091

P(ειδήσεις|no) = (1+1)/(6+8) = 0,143

P(γήπεδο|yes) = (0+1)/(14+8) = 0,045

P(γήπεδο|no) = (2+1)/(6+8) = 0,214

P(ομάδα|yes) = (0+1)/(14+8) = 0,045

P(ομάδα|no) = (2+1)/(6+8) = 0,214

Υπολογίζω τις πιθανότητες του κατηγοριοποιητή

P(d6|Yes) = P(TV|yes) * P(επεισόδιο|yes)^2 * P(γήπεδο|yes)^3 * P(ειδήσεις|yes) = 0,6 * (0,273)^2 * (0,045)^2 * 0,091 = **0,0000003810199028**

P(d6|No) = P(TV|no) * P(επεισόδιο|no)^2 * P(γήπεδο|no)^3 * P(ειδήσεις|no) = 0,4 * (0,143)^2 * (0,214)^3 * 0,143 = **0,00001147481067**

Άρα το έγγραφο d6 θα κατηγοριοποιηθεί στην κατηγορία **P(TV|no)** με την μεγαλύτερη πιθανότητα.

1β. Bernulli Naive Bayes

Υπολογίζω τις πιθανότητες των κλάσεων TV|yes και TV|no

P(TV|yes) = (3/5) = 0,6

P(TV|no) = (2/5) = 0,4

Υπολογίζω τις πιθανότητες της μοναδικής παρουσίας των λέξεων στα έγγραφα στις κλάσεις TV|yes και TV|no

P(πρόγραμμα|yes) = (2+1)/(3+2) = 0,6

P(πρόγραμμα|no) = (0+1)/(2+2) = 0,25

P(επεισόδιο|yes) = (3+1)/(3+2) = 0,8

P(επεισόδιο|no) = (1+1)/(2+2) = 0,5

P(σειρά|yes) = (1+1)/(3+2) = 0,4

P(σειρά|no) = (0+1)/(2+2) = 0,25

P(κανάλι|yes) = (2+1)/(3+2) = 0,6

P(κανάλι|no) = (0+1)/(2+2) = 0,25

P(ταινία|yes) = (1+1)/(3+2) = 0,4

P(ταινία|no) = (0+1)/(2+2) = 0,25

P(ειδήσεις|yes) = (1+1)/(3+2) = 0,4

P(ειδήσεις|no) = (1+1)/(2+2) = 0,5

P(γήπεδο|yes) = (0+1)/(3+2) = 0,2

P(γήπεδο|no) = (2+1)/(2+2) = 0,75

P(ομάδα|yes) = (0+1)/(3+2) = 0,2

P(ομάδα|no) = (2+1)/(2+2) = 0,75

Υπολογίζω τις πιθανότητες του κατηγοριοποιητή

P(d6|Yes) = P(TV|yes) * P(επεισόδιο|yes) * P(ειδήσεις|yes) * P(γήπεδο|yes) * (1- P(πρόγραμμα|yes)) * (1- P(σειρά|yes)) * (1- P(κανάλι|yes)) * (1- P(ταινία|yes)) * (1- P(ομάδα|yes)) =

0,6*0,8*0,4*0,2*(1-0,6)*(1-0,4)*(1-0,6)*(1-0,4)*(1-0,2) = **0,001769472**

P(d6|No) = P(TV|no) * P(επεισόδιο|no) * P(γήπεδο|no) * P(ειδήσεις|no) * (1- P(πρόγραμμα|no)) * (1- P(σειρά|no)) * (1- P(κανάλι|no)) * (1- P(ταινία|no)) * (1- P(ομάδα|no)) =

0,4*0,5*0,75*0,5*(1-0,25)*(1-0,25)*(1-0,25)*(1-0,25)*(1-0,75) = **0,005932617188**

Άρα το έγγραφο d6 θα κατηγοριοποιηθεί στην κατηγορία **P(TV|no)** με την μεγαλύτερη πιθανότητα.

2. Κατηγοριοποίηση κειμένου με Rapid Miner

2.1 Εισαγωγή

Έχουμε ένα σύνολο δεδομένων από τη συλλογή με Newsgroups άρθρα.

Έχουμε 9 κατηγορίες:

Pc Hardware

Mac Hardware

Sport Baseball

Sport Hockey

Science Space

Misc ForSale

Politics

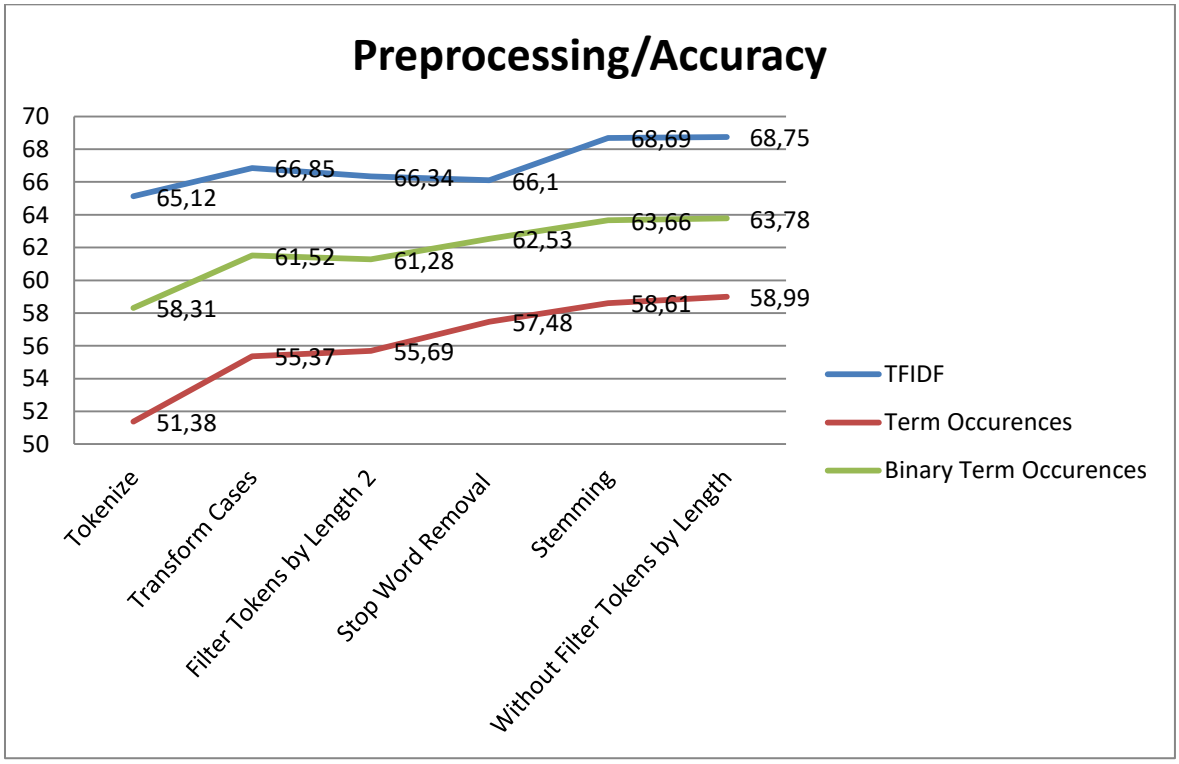
Religion

Atheism

2.2 Preprocessing

Σχεδίασα την διαδικασία της προεπεξεργασίας ως εξής:

Δοκίμασα τις τεχνικές αυξητικά με κριτήριο το accuracy και με τα 3 διανύσματα, TFIDF, Term Occurrences και Binary Term Occurrences. Επίσης το pruning το έθεσα στο 3%/97% για να έχω γρήγορους χρόνους επεξεργασίας (~10sec). Αποτέλεσμα το παρακάτω γράφημα:



Γράφημα 1. Preprocessing

Μετά από το βασικό Tokenize, εφάρμοσα το Transform Cases, το οποίο βελτίωσε το accuracy και στα 3 διανύσματα, στην συνέχεια φιλτράρισα τις λέξεις με 1 στοιχείο αλλά εδώ δεν είχα βελτίωση, στο TFIDF και στο Binary Term Occurrences υπήρξε ελαφριά μείωση του accuracy. Στη συνέχεια εφάρμοσα τα Stop Word Removal και Stemming(Porter), τα οποία βελτίωσαν την ακρίβεια αρκετά.

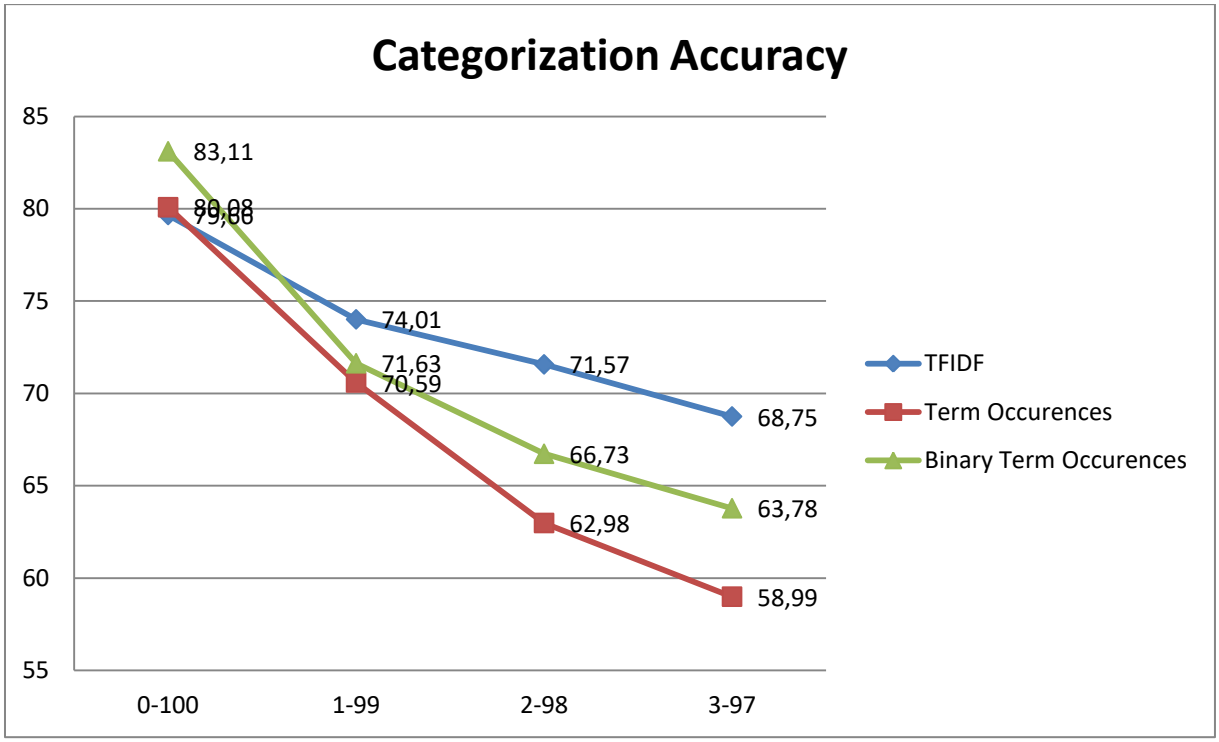
Τέλος δοκίμασα να αφαιρέσω το φίλτρο Filter Tokens by Length και είδα βελτίωση στο accuracy οπότε αποφάσισα να μην το χρησιμοποιήσω για την επεξεργασία.

Τελικό setup για την περαιτέρω επεξεργασία:

Tokenize, Transform Cases, Stop Word Removal, Stemming

2.2 Κατηγοριοποίηση

Με δεδομένο το Preprocessing Pipeline και την εφαρμογή του Naive Bayes classifier πήρα τα παρακάτω αποτελέσματα για διαφορετικές τιμές Prunning και τα 3 διαφορετικά διανύσματα.

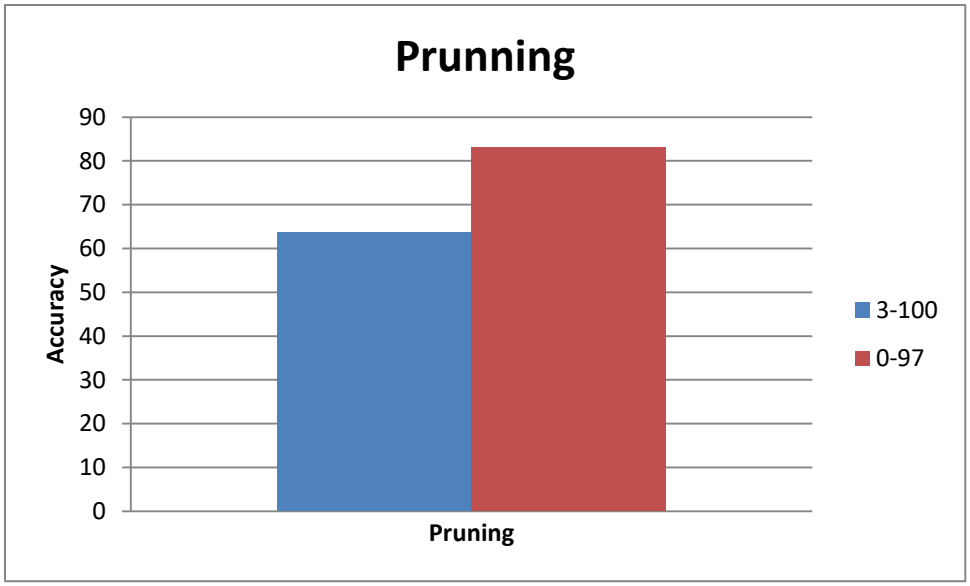


Γράφημα 2. Prunning levels

Όπως φαίνεται κι απ το γράφημα, το Prunning έχει αρνητικά αποτελέσματα στην ακρίβεια του μοντέλου. Το Binary Term Occurences διάνυσμα πετυχαίνει τη μεγαλύτερη ακρίβεια με απουσία Prunning.

Θέλω να δω όμως και σε ποια πλευρά του κλαδέματος οφείλεται η απώλεια κουρέματος, στις λέξεις που εμφανίζονται σπάνια ή στις λέξεις που εμφανίζονται συχνά;

Οπότε με δοκιμή στο Binary Term Occurences με την υψηλότερη ακρίβεια και κλάδεμα διαδοχικά 3% και 97% έχουμε:



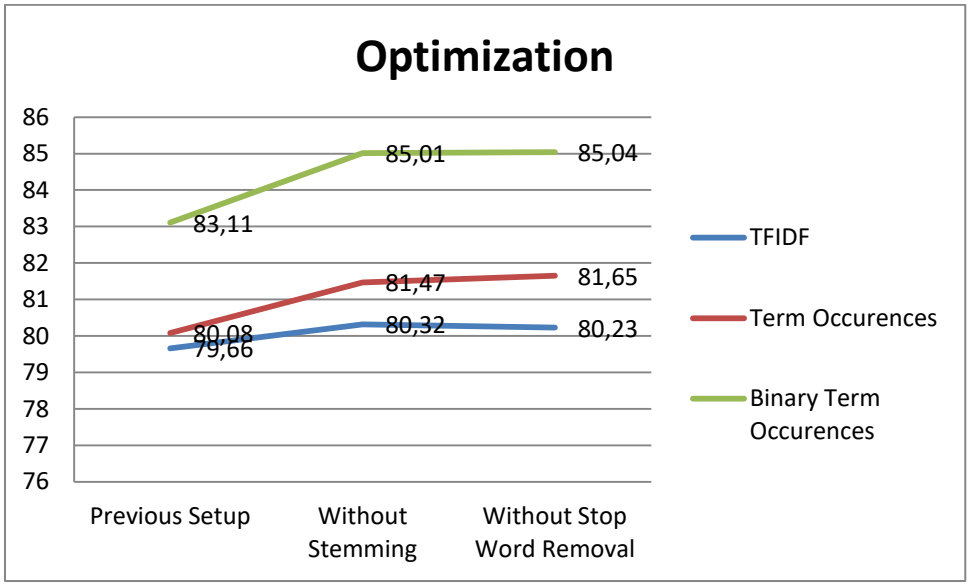
Γράφημα 3. Pruning Sides

Βλέπουμε λοιπόν ότι με κλάδεμα στις σπάνιες λέξεις στο 3% του κειμένου, η ακρίβεια πέφτει στο 63% περίπου ενώ με κλάδεμα στο 97% των συχνών λέξεων η ακρίβεια μένει σχεδόν ανεπηρέαστη.

Αυτό οφείλεται στο γεγονός κάποιες από τις κατηγορίες που έχουμε είναι κοντινές μεταξύ τους, π.χ. οι pc hardware/mac hardware, atheism/religion, baseball/hockey συνεπώς χρειαζόμαστε σπάνιες λέξεις στα έγγραφα μας για να πετύχουμε καλύτερη κατηγοριοποίηση.

2.3 Βελτιστοποίηση Κατηγοριοποίησης

Θέλω να δώ αν είναι δυνατό ένα καλύτερο ποσοστό ακρίβειας οπότε με δεδομένο το κλάδεμα στο 0/100, θα γυρίσω στις παραμέτρους της προεπεξεργασίας και θα αφαιρέσω το Stemming και το Stop Word Removal διαδοχικά.



Γράφημα 4. Βελτιστοποίηση

Η αφαίρεση του Stemming έχει θετική επιρροή στην ακρίβεια όλων των διανυσμάτων αλλά το η αφαίρεση του Stop Word Removal δεν έφερε μεγάλη βελτίωση στην ακρίβεια.

Συνεπώς το τελικό Pipeline και μοντέλο που πετυχαίνει την μεγαλύτερη ακρίβεια είναι 85,01%:

Preprocessing: Tokenize, Transform Cases, Filter Stopwords

Categorization: Naive Bayes, No Pruning, Binary Term Occurences

2.4 Recall Precision

accuracy 85,01%	true hardwarePC	true sportBaseball	true hardwareMac	true sportHockey	true sciSpace	true miscForsale	true politics	true religion	true atheism	class precision
pred. hardwarePC	303	0	45	1	4	38	0	0	4	76.71%
pred. sportBaseball	1	349	1	6	1	6	0	0	0	95.88%
pred. hardwareMac	58	3	301	2	1	40	1	0	0	74.14%
pred. sportHockey	1	22	3	379	3	8	3	0	0	90.45%
pred. sciSpace	11	10	23	2	366	21	13	0	5	81.15%
pred. miscForsale	15	4	10	3	3	267	0	0	0	88.41%
pred. politics	1	7	2	5	13	6	248	0	10	84.93%
pred. religion	2	2	0	0	1	2	31	375	29	84.84%
pred. atheism	0	0	0	1	2	2	14	2	271	92.81%
class recall	77.30%	87.91%	78.18%	94.99%	92.89%	68.46%	80.00%	99.47%	84.95%	

Πίνακας 1. Recall/Precision

Ο πίνακας 1 μας δείχνει τις τιμές precision-recall για κάθε κατηγορία του μοντέλου με την μεγαλύτερη ακρίβεια(85,01%).

Το μοντέλο μας έχει πολύ καλές επιδόσεις στο precision σε κάποιες κατηγορίες, baseball, hockey, misc ForSale, atheism, politics και religion ενώ στα pc hardware και mac hardware δεν τα πήγε και τόσο καλά. Εντυπωσιακό είναι το ποσοστό του recall για την κατηγορία religion με 99,47%.

Στο recall πάλι έχουμε καλές επιδόσεις στις κατηγορίες baseball, hockey, science space, religion και atheism ενώ στις κατηγορίες hardware pc, hardware mac, misc ForSale και politics είχε μέτρια αποτελέσματα.