Γλυκός Αθανάσιος Παναγιώτης p3190046

Γλυκού Χριστίνα Μαρία p3190047

Τεχνητή Νοημοσύνη: Εργασία 2

- αφελής ταξινομητής Bayes με πολυμεταβλητή μορφή Bernoulli

Το αρχείο μας ονομάζεται imdb

Αρχικά η πρώτος αλγόριθμος μάθησης που επιλέξαμε να υλοποιήσουμε είναι ο αφελής ταξινομητής Bayes με πολυμεταβλητή μορφή Bernoulli. Εισάγουμε στο πρόγραμμα μας τα δεδομένα από το imdb που αφορούν κάποιες κριτικές ταινιών και δημιουργούμε το λεξιλόγιο ανάλογα με τις παραμετρους μας .

Ύστερα δημιουργούμε τον πίνακα **x** ο οποίος περιέχει για κάθε review του x_train και για κάθε λέξη του λεξιλογίου αν περιέχεται στο κάθε review. Στην συνέχεια φτιάχνουμε έναν αντίστοιχο πίνακα x2 αλλά για τα **x_test**.

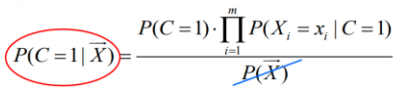
Ύστερα με τη χρήση της μεθόδου **find_possibility** δημιουργούμε έναν πίνακα με πιθανότητες : **possibility** οποίος θα περιέχει 4 πιθανότητες, την πιθανότητα η λέξη να υπάρχει μέσα στο review και να κατατάσσεται στα θετικά reviews , η δεύτερη είναι να μην υπάρχει η λέξη και να κατατάσσεται στα θετικά reviews, η τρίτη να υπάρχει η λέξη και να κατατάσσεται στα αρνητικά reviews και τέλος να μην υπάρχει η λέξη και να κατατάσσεται στα αρνητικά reviews. Για να υπολογίσουμε τις πιθανότητες αυτές χρησιμοποιούμε τις μεταβλητές **yes_exists** , **yes_den_exists** , **no_exists** , **no_den_exists** και **sumyes** (πλήθος θετικών reviews) , **sumno** (πλήθος αρνητικών reviews). Χρησιμοποιούμε επίσης την εκτιμήτρια Laplace για να μην μηδενίζονται οι πιθανότητες.

Στη συνέχεια υπολογίζουμε τα ποσοστά των θετικών και των αρνητικών review που θα μας χρειαστούν αργότερα με τη μέθοδο **possibility_of_pos_neg**.

Έπειτα καλούμε τη μέθοδο **results** όπου υπολογίζουμε τις πιθανότητες **p_yes_y** και **p_no_y** που αντιστοιχούν στις πιθανότητες ενώ γνωρίζουμε

το review y, το review να είναι θετικό και αντίστοιχα ενώ γνωρίζουμε το review y,το review να είναι αρνητικό. Αυτό επιτυγχάνεται ως εξής : πχ για την πιθανότητα p_yes_y

Δημιουργούμε το γινόμενο



(ο παρονομαστής δεν μας απασχολεί)

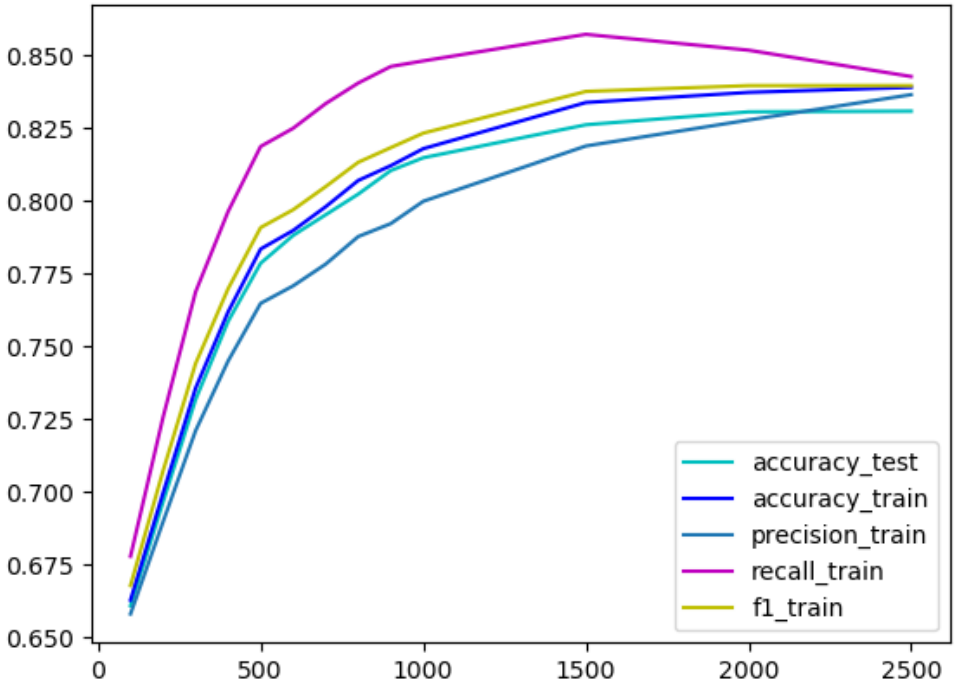
Όπου έχουμε ήδη υπολογίζει την πιθανότητα των θετικών review ,οπότε υπολογίζουμε το γινόμενο των ΠΡ(X=xi|C=1) το οποίο το βρίσκουμε από τον πίνακα possibility που δημιουργήσαμε πριν. Ομοίως και για το p_no_y.

Αν το p_yes_y είναι μεγαλύτερο από το p_no_y τότε κατηγοριοποιούμε το review y ως θετικό αλλιώς ως αρνητικό, και αποθηκεύουμε το αποτέλεσμα σε έναν πίνακα.

Στη συνέχεια συγκρίνουμε τον πίνακα των αποτελεσμάτων που υπολογίσαμε με τον πίνακα των κανονικών αποτελεσμάτων και υπολογίζουμε τα ποσοστά

Αυτή την διαδικασία την εκτελούμε για τα x_train και y_train και στο τέλος εμφανίζουμε τα accuracy κτλ.

λέξεις	100	200	300	400	500	600	700	800	900	1000	1500	2000	2500
Accuracy test	0.66104	0.6966	0.73176	0.7586	0.77856	0.78812	0.7952	0.80228	0.81036	0.81476	0.82608	0.83052	0.83076
Accuracy train	0.66296	0.69968	0.73568	0.76188	0.78344	0.7898	0.79796	0.80696	0.81204	0.81788	0.83372	0.8372	0.83892
precision	0.6582013047530288	0.6898098859315589	0.7210715893741558	0.7450041164583349	0.7647982062780269	0.7708006279434851	0.7782592454239821	0.7877174565086983	0.7921504007190473	0.7998189089262808	0.8188001528467711	0.8277095319544595	0.8364170570952116
recall	0.678	0.72568	0.76872	0.79632	0.81864	0.82488	0.83336	0.8404	0.84608	0.848	0.85712	0.85168	0.84264
F1	0.6679539722572508	0.7072904483430799	0.7441338186323386	0.7698078187231738	0.7908037094281299	0.7969239092630521	0.8048676839868648	0.8132063786963926	0.8182275347181928	0.8232050634877489	0.8375219855384014	0.8395236968693332	0.8395169967720082



- ID3, με πριόνισμα

Το αρχείο μας ονομάζεται imdb2

Αρχικά, διαβάζουμε τα δεδομένα μας με τον ίδιο τρόπο που τα διαβάσαμε και στον bayes και δημιουργούμε ξανά δυο πίνακες, τον χ που αφορά τα train και τον x2 που αφορά τα test, όπου για κάθε feature του λεξιλογίου έχουν τιμή 1 αν το περιέχουν και τιμή 0 αν όχι.

Στη συνέχεια δημιουργούμε τις εξής μεθόδους:

- twoCEntropy: η μέθοδος αυτή επιστρέφει την εντροπία 2 μεταβλητών σύμφωνα με γνωστό μαθηματικό τύπο.
- calculateIG: η συνάρτηση αυτή υπολογίζει έναν πίνακα με τα information gain των ιδιοτήτων μας χρησιμοποιώντας την εντροπία (twoCEntropy) και τις πιθανότητες που υπολογίζουμε.
- most_informative_feature: αρχικά υπολογίζουμε τα IG με τη χρήση της συνάρτησης calculateIG, βρίσκουμε το max μεταξύ αυτών και παράλληλα ελέγχουμε αν αυτό το feature έχει ξαναχρησιμοποιηθεί(χρησιμοποιώντας τον πίνακα xrisimopoihmenes_theseis)
- make_sub_tree: Αρχικά υπολογίζουμε το πληθος των reviews που περιέχουν το συγκεκριμένο feature (count_1) και το πλήθος αυτών που δεν το έχουν (count_0) . Επίσης ξεχωρίζουμε σε 2 πίνακες τα reviews που έχουν το feature και τα reviews που δεν το έχουν . Για κάθε περίπτωση υπολογίζουμε πόσα reviews είναι θετικά και πόσα είναι αρνητικά . Αναλυτικότερα, έχουμε δηλαδή 4 περιπτώσεις οι οποίες είναι :
  class0_count1 : όσα στην στήλη feature =1 το y_train=1
  class0_count0 : όσα στην στήλη feature =1 το y_train=0
  class1_count1 : όσα στην στήλη feature =0 το y_train=1
  class0_count1 : όσα στην στήλη feature =0 το y_train=0

  Ύστερα σε έναν πίνακα most_common κρατάμε για το συγκεκριμένο feature αν τα περισσότερα reviews είναι θετικά ή αρνητικά το οποίο το χρειαζόμαστε για την συνέχεια. Δηλαδή

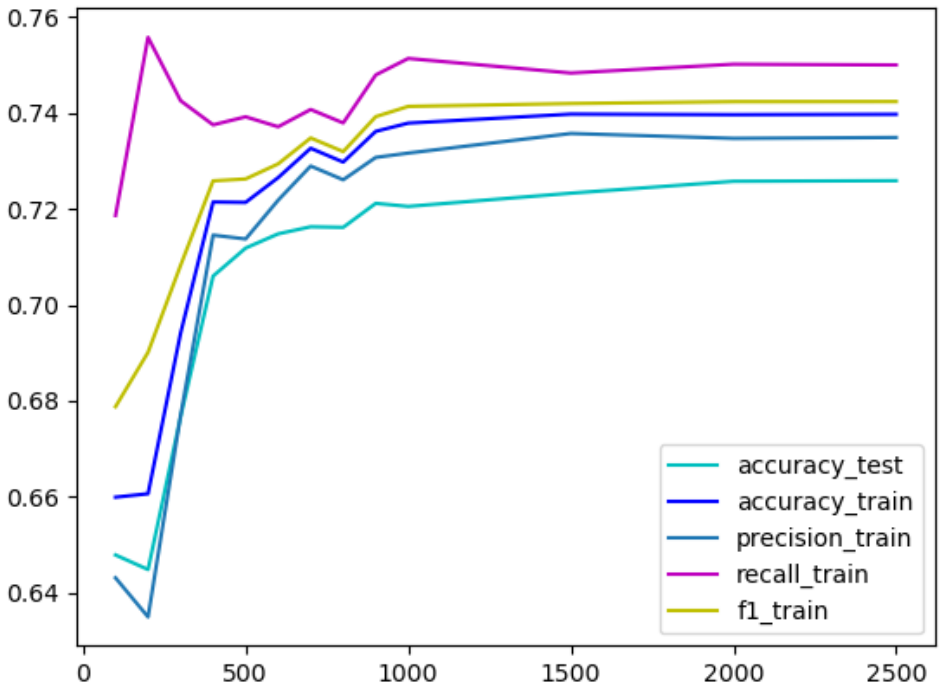
στην περίπτωση που μας τελειώσουν είτε οι ιδιότητες είτε τα παραδείγματα και ενώ ο κόμβος έπρεπε να επεκταθεί δεν μπορούμε να τον επεκτείνουμε περεταίρω.

Στην συνέχεια αποφασίζουμε αν ο κόμβος στον οποίο βρισκόμαστε θα επεκταθεί ή όχι χρησιμοποιώντας την πιθανότητα για τις περιπτώσεις το feature να είναι 1 και αν είναι 0 .Αν το ποσοστό των θετικών ή των αρνητικών reviews είναι ίσο ή ξεπερνα το 70% τότε δεν τον επεκτείνουμε άλλο και του κατατάσσουμε το review ως θετικό ή αρνητικό αντίστοιχα. Αλλιώς το μαρκάρουμε με « ? » που σημαίνει ότι μπορεί να επεκταθεί παραπάνω.

- generate_tree: μέσω αναδρομικών κλήσεων όσο δεν μας έχουν τελειώσει τα reviews και οι ιδιότητες που μπορούμε να χρησιμοποιήσουμε, βρίσκουμε το most informative feature και υπολογίζουμε το δέντρο και το νέο πίνακα με reviews και αποτελέσματα, βρίσκουμε το νέο κόμβο που αντιστοιχεί στο δέντρο μας και αν είναι δυνατή η επέκτασή του συνεχίζουμε αναδρομικά της κλήσεις μας.
- Id3: σε αυτή τη συνάρτηση απλά καλούμε την generate_tree και επιστρέφουμε το δέντρο μας.
- decision: μας δίνεται ως είσοδος το δέντρο που δημιουργήσαμε πριν και ένα review. Προσπελαύνουμε τους κόμβους του δέντρου ελέγχοντας την αντιστοιχία με το review ώστε να καταλήξουμε σε φύλλο του δέντρου που θα είναι το αποτέλεσμα του υπολογισμού κατηγοριοποίησης του review.
- Evaluate: Τέλος έχουμε την μέθοδο evaluate όπου για κάθε review ενός set υπολογίζουμε τα ποσοστά επιτυχίας κτλ.(Αν ένα φύλλο είχε την τιμή ? χρησιμοποιούμε τον πίνακα most common και αποφασίζουμε αν το review θα κατηγοριοποιηθεί ως θετικό ή αρνητικό.

Οι διαδικασία εκτελείται για τα test και τα train και εμφανίζουμε τις μετρήσεις που μας ζητούνται.

λέξεις	100	200	300	400	500	600	700	800	900	1000	1500	2000	2500
Accuracy test	0.64788	0.64484	0.67664	0.706	0.71184	0.7148	0.71628	0.71616	0.72116	0.72052	0.72328	0.72576	0.72588
Accuracy train	0.65992	0.66064	0.69408	0.72144	0.72136	0.72652	0.73264	0.72976	0.73616	0.73788	0.73976	0.73964	0.73972
precision	0.6431128293241696	0.6349643769323834	0.6769253208868145	0.7145403813361227	0.7137339718832072	0.7218174696435566	0.7289403243583688	0.7260705289672544	0.7307331561669532	0.7316351172392304	0.7357243983010854	0.7347018726004858	0.7348906482715372
recall	0.71864	0.75576	0.74256	0.73752	0.7392	0.73712	0.74072	0.73792	0.74792	0.75136	0.74832	0.75016	0.75
F1	0.6787819253438114	0.6901161516546132	0.7082252403479322	0.7258483583969767	0.7262438104220702	0.7293884820898476	0.7347829537338306	0.7319473099508016	0.7392266940776469	0.7413663811881671	0.7419687475212183	0.7423504730237899	0.7423684523102507



Τα προγραμματα τα τρέχουμε **δίνοντας 100,…,2500 ως number of words και 0 ως skip top words**. Δηλαδη τα διαγραμματα αφορούν αυτές τις περιπτώσεις.