**Εργασία 2 Τεχνητής Νοημοσύνης**

Κωνσταντίνος Ιωάννης Κορναράκης 3170074

Γεώργιος Σπινάρης 3170151

Ηλίας Μπεζαντάκος 3160241

**Bernoulli Naive Bayes**

Δέχεται 2 Arraylists από Hashsets που περιέχουν τις λέξεις κάθε μηνύματος. Το list0 περιέχει ham μηνύματα και το list1

spam. Έπειτα, στο στάδιο της εκπαίδευσης χρησιμοποιείται ένα Arraylist από Hashmaps που αρχικά περιέχει τη συχνότητα

εμφάνισης κάθε λέξης στα μηνύματα της κάθε κατηγορίας, όπου η πρώτη θέση του Arraylist αντιστοιχεί το Hashmap της

κατηγορίας ham και η δεύτερη στο Hashmap της κατηγορίας spam. Επιλέγονται οι 200 λέξεις με την μεγαλύτερη συχνότητα

από κάθε κατηγορία υπολογίζονται οι πιθανότητες. Κάτα την κατάταξη, δίνεται ένα Hashset που περιέχει τις λέξεις του

μηνύμα προς κατάταξη και πολλαπλασιάζει τις πιθανότητες ανάλογα με την υπάρξη ή όχι της κάθε λέξης των Hashmap

στο μήνυμα και επιστρέφει 0 αν το μήνυμα πιθανότατα είναι ham και 1 αν θεωρηθεί spam.

**ID3**

Δέχεται 2 Arraylists από Hashsets που περιέχουν τις λέξεις κάθε μηνύματος. Το list0 περιέχει ham μηνύματα και το list1

spam. Έπειτα, στο στάδιο της εκπαίδευσης δημιουργείται ένα δέντρο, όπου σε κάθε κόμβο επιλέγεται η λέξη που

διαχωρίζει καλύτερα τα παραδείγματα στις 2 κατηγορίες χρησιμοποιώντας το information gain. Κάτα την κατάταξη, δίνεται

ένα Hashset που περιέχει τις λέξεις του μηνύματος προς κατάταξη και το δέντρο το κατατάσσει ελέγχοντας την ύπαρξη ή

όχι της λέξης κάθε κόμβου.

**Adaboost με δέντρα βάθους 1 (stumps)**

Δέχεται 2 Arraylists από Hashsets που περιέχουν τις λέξεις κάθε μηνύματος. Το list0 περιέχει ham μηνύματα και το list1

spam. Έπειτα, στο στάδιο της εκπαίδευσης χρησιμοποιούνται Μ stumps για την κατάταξη των παραδειγμάτων και μεταβάλλονται

τα βάρη του κάθε παραδείγματος ανάλογα με το αν κατατάχθηκε σωστά το συγκεκριμένο παράδειγμα.

**Multinomial Naïve Bayes**

Αρχικά, υπάρχει ένα hash map και 2 array list.

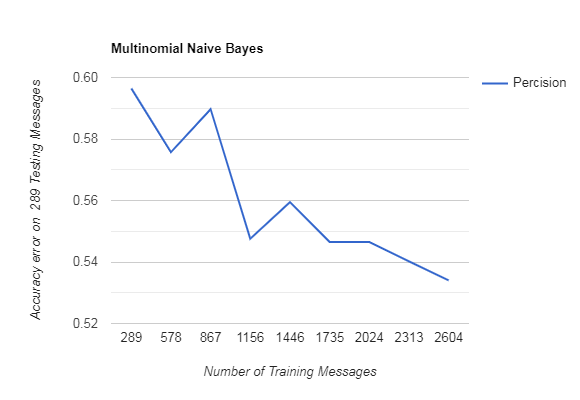
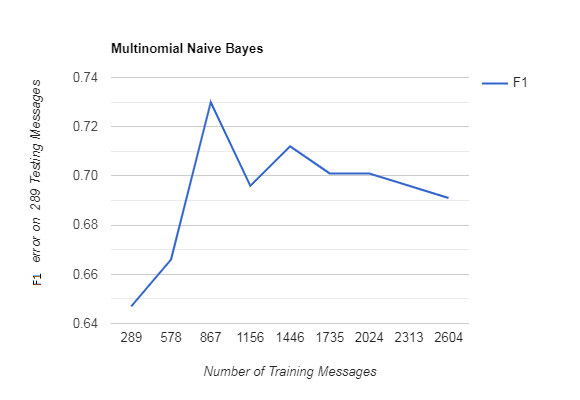
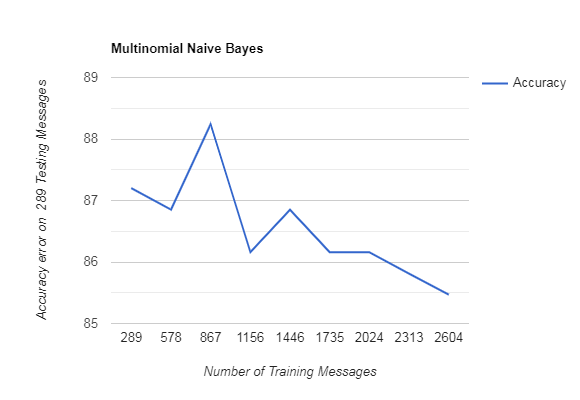
Στο hash map vocabulary αποθηκεύονται σαν κλειδί όλες οι μοναδικές λέξεις που συναντώνται μέσα από το σύνολο των μηνυμάτων εκπαίδευσης. Και αντίστοιχα σαν τιμή του κάθε κλειδιού/λέξης ένας αριθμός ο οποίος αντιπροσωπεύει την θέση της αντίστοιχης λέξης στα arraylist (η οποία θέση είναι όμοια για όλα τα arraylist).

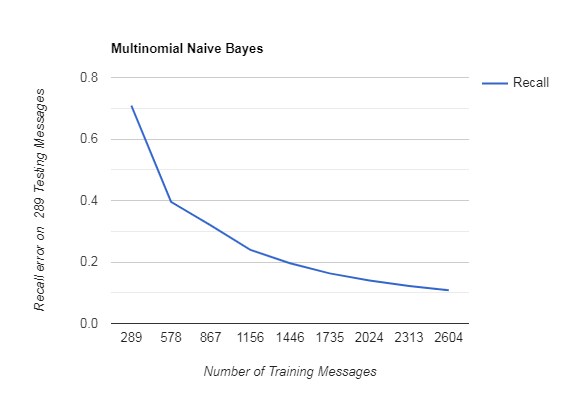
Στο arraylist spamWordCounter, αποθηκεύεται το πόσες φορές η λέξη εμφανίστηκε σε μήνυμα σπαμ. Στο arraylist hamWordCounter, αποθηκεύεται αντιστοίχως το πόσες φορές η λέξη εμφανίστηκε σε μήνυμα χαμ.

Η εκτέλεση ξεκινά ως εξής:  
Διαβάζονται τα μηνύματα εκπαίδευσης. Κάθε νέα λέξη προστίθεται στο vocabulary και τα 2 arraylist στην θέση που αντιστοιχεί η λέξη παίρνουν τις κατάλληλες τιμές. Όταν μία υπάρχουσα λέξη στο vocabulary εντοπίζεται, τότε αυξάνεται ο κατάλληλος μετρητής στο κατάλληλο arraylist στην θέση που αντιστοιχεί στην λέξη.

Για το τεστ, χρησιμοποιείται ένα arraylist postStatsTemp (μετέπειτα postStats) με μήκος όσα και τα μηνύματα τεστ και πλάτος 2 όπου η πρώτη θέση δείχνει αν το μήνυμα έχει εκτιμηθεί σαν σπαμ(true αν εκτιμήθηκε σαν σπαμ, false αν όχι) και στη δεύτερη θέση δείχνει αν το μήνυμα ήταν όντως σπαμ(true αν ήταν όντως σπαμ, false αν όχι). Στο τέλος του διαβάσματος ενός μηνύματος, όλες οι λέξεις του μηνύματος προστίθενται σε ένα arraylist sentence (κάθε λέξη έχει μία θέση). Αυτό το arraylist περνάει σαν όρισμα στην calculate μέθοδο η οποία υπολογίζει τις πιθανότητες να είναι το μήνυμα χαμ και σπαμ με τους γνωστούς τύπους και τις επιστρέφει. Αποφασίζεται τότε βάσει των 2 αυτών πιθανοτήτων αν το μήνυμα είναι σπαμ ή χαμ και σε ενημερώνεται το arraylist postStatsTemp. Αφού διαβαστούν όλα τα μηνύματα τεστ, το postStatsTemp αντιγράφεται στον πίνακα postStats.

Τέλος, υπολογίζονται τα στατιστικά και εμφανίζονται. (Στον υπολογισμό των πιθανοτήτων χρησιμοποιήθηκε εξομάλυνση Laplace με +1 στον αριθμητή και +(πλήθος λέξεων στο vocabulary) στον παραονομαστή). Επίσης, έγινε χρήση λογαρίθμου στις πιθανότητες ώστε να αποφευχθεί το underflow.





Training data: lingspam – bare – part1

Testing data: lingspam – bare – part2

+-------------------------+----------+-----------+--------+--------+

| | Accuracy | Precision | Recall | F1 |

+-------------------------+----------+-----------+--------+--------+

| Bernoulli Naive Bayes | 81.31% | 46.87 | 93.75% | 0.62 |

+-------------------------+----------+-----------+--------+--------+

| Multinomial Naive Bayes | 87.2% | 64.1% | 52.08% | 57.47% |

+-------------------------+----------+-----------+--------+--------+

| ID3 | 83.39% | 50% | 100% | 0.66% |

+-------------------------+----------+-----------+--------+--------+

| Adaboost | 83.04% | 46.15% | 12.5% | 0.19% |

+-------------------------+----------+-----------+--------+--------+

Training data & testing data: lingspam – bare – part1

+-------------------------+----------+-----------+--------+--------+

| | Accuracy | Precision | Recall | F1 |

+-------------------------+----------+-----------+--------+--------+

| Bernoulli Naive Bayes | 96.89% | 95.34% | 85.41% | 0.9% |

+-------------------------+----------+-----------+--------+--------+

| Multinomial Naive Bayes | 92.04% | 100% | 52.08% | 68.49% |

+-------------------------+----------+-----------+--------+--------+

| ID3 | 100% | 100% | 100% | 1% |

+-------------------------+----------+-----------+--------+--------+

| Adaboost | 90.66% | 100% | 43.75% | 0.6% |

+-------------------------+----------+-----------+--------+--------+