Bayes Argorithm

Υλοποίηση 1 αλγορίθμου αφελή ταξινομητή Bayes με την παραδοχή ότι οι τιμές/λέξεις είναι ανεξάρτητες μεταξύ τους.
Για να μπορέσουν να βγουν τα αποτελέσματα σωστά χρησιμοποιήθηκε και η εκτιμήτρια Laplace η οποία πρόσθεση (+) μια ψεύτο εμφάνιση σε όλες τις λέξεις.
Για την υλοποιήση του αλγορίθμου χρησιμοποιήθηκαν τα εξής:

- Ένα HashMap ονόματι MailHash.

  > Ο MailHash περιέχει για όλο το λεξιλόγιο τον αριθμό τον mails τα οποία περιέχουν την συγκεκριμένη λέξη.

- Ένα HashSet ονόματι tempSet.

  > Στο οποίο αποθηκεύονται προσωρινά όλες οι μοναδικές λέξεις για κάθε mail. Και έπειτα προσθέτονται στο MailHash(Ένα HashMap)

Διάβασμα στοιχείων

Για να επιτευχθεί αυτό χρησιμοποιούνται δύο συναρτήσεις: inputToHashMap() και UpdateMailHash()

- inputToHashMap

  > Παίρνει δύο ορίσματα το path το οποίο αναφέρεται σε ποια δεδομένα θα επεξεργαστεί ο αλγόριθμος και το posost που αναφέρεται για πόσο της % τον δεδομένων θα εκπαιδευτεί.Η inputToHashMap() για κάθε directory ανοίγει ένα mail το καταγράφει αν είναι spam η ham και το εισάγει στο τοπικό tempSet, και τέλος ενημερώνει τον λεξιλόγιο δηλαδή τον MailHash για της νέες η τις υπάρχουσες λέξεις αντίστοιχα. *UpdateMailHash Παίρνει ένα όρισμα 0 για spam και 1 για ham και ελέγχει αν η λέξη υπάρχει η όχι και αναλόγως αυξάνει τον κατάλληλο μετρητή.

Μετά το διάβασμα τον δεδομένων καλείται η συνάρτηση.

Training

- Laplace()

> η οποία προσθέτει (+) μια ψευτό εμφάνιση σε όλες τις λέξεις.

- Probability()

> η οποία αλλάζει την τιμή μέσα στον MailHash(Ένα HashMap) και το κάνει πιθανότητα εμφάνιση της λέξεις σύμφωνα με τον τύπο

$$SpamPropability_{lexi} = log(1 + lexi^{\varphi o \rho \acute{\epsilon} \varsigma \epsilon \mu \varphi \acute{\alpha}} / SpamCounter^{\sigma \upsilon \nu o \lambda \iota \kappa o \varsigma})$$

$$HamPropability_{lexi} = log(1 + lexi^{\varphi o \rho \acute{\epsilon} \varsigma \epsilon \mu \varphi \acute{\alpha}} / HamCounter^{\sigma \upsilon \nu o \lambda \iota \kappa o \varsigma})$$

Testing

Κατά την διάρκεια του testing ο αλγόριθμος διαβάζει το τελευταίο κομμάτι των δεδομένων που είναι άγνωστα μέχρι στιγμής στον αλγόριθμο και προσπαθεί με βάση τα trainig δεδομένα να κάνει μία πρόβλεψη σε ποια κατηγορία ανήκει μέσω των 2 παρακάτω τύπων ανάλογα με πιο έχει την μεγαλύτερη πιθανότητα.

- εάν η λέξη υπάρχει στο mail.

$$P(C = 1/X) = log(P(C = 1)) + log(\prod_{i=1}^{forallwords} P(X_i = x_i/C = 1))$$

$$P(C = 0/X) = log(P(C = 0)) + log(\prod_{i=1}^{forallwords} P(X_i = x_i/C = 0))$$

- εάν η λέξη δεν υπάρχει στο mail.
- $$P(C = 1/X) = log(P(C = 1)) + log(1/TotalHamCounter))$$
- $$P(C = 0/X) = log(P(C = 0)) + log(1/TotalSpamCounter))$$

Υπολογισμός (Accuracy PRecision Recall)

- Accuracy

$$\frac{100 * (TP + TN)}{TP + TN + FP + FN}$$

- HamPRecision

$$\frac{100 * TP}{TP + FP}$$

- SpamPRecision

$$\frac{100 * TN}{TN + FN}$$

- RecallHam

$$\frac{100 * TP}{TP + FN}$$

- RecallSpam

$$\frac{100 * TN}{TN + FP}$$

TP=0,TN=1,FP=2,FN=3