**Τεχνητη Νοημοσυνη Εργασια 2**

**“Λογιστικη Παλινδρομηση”**

Το προγραμμα αυτο δεχεται ως input αρχεια τυπου txt, απο τα οποια το καθενα αντιστοιχει σε ενα διαφορετικο email. Επειτα, χωριζει τα emails σε δεδομενα εκπαιδευσης και δεδομενα αξιολογησης σε αναλογια 7 προς 1. Αγνοωντας το κομματι “Subject :” καθε γραπτου μηνυματος, καθως ολα περιερχουν την παραπανω ακολουθια χαρακτηρων στο αρχειο τους, ελεγχοντας ολα τα αρχεια των δεδομενων ελεγχου ενα προς ενα δημιουργει ενα Hash-Map  
(“**Dictionary**/**DictionaryShorted**)(ταξινομημενο με βαση την συχνοτητα εμφανισης), στο οποιο περιεχονται ολες οι λεξεις που εμφανιστηκαν στον ελεγχο αυτο. Απο το παραπανω κραταει τις “NWords” πιο συχνες λεξεις στην μεταβλητη **words,** οπου το NWords καθοριζεται απο τον χρηστη ως μεταβλητη.  
Στην συνεχεια δημιουργει μια λιστα για καθε τυπο δεδομενων (training/testing).

Για καθε ενα email δημιουργειται ενα “instance” με τα χαρακτηρηστικα:

* Name (ονομα αρχειου που περιεχει το email)
* Label (0 για “ham” / 1 για “spam”)
* X (Vector που περιεχει τα χαρακτηρηστικα του καθε instance)

Τα χαρακτηρηστικα του καθε email ειναι ενας πινακας που αντιστοιχει τις λεξεις του πινακα **words** με τις λεξεις που περιεχει το παραπανω.  
Για παραδειγμα, αν στην θεση 1 του πινακα words υπαρχει η λεξη “hello” και στο νεο μηνυμα που εξεταζουμε εμφανιζεται η ιδια λεξη 3 φορες, τοτε x[1]=3

Επειτα το προγραμμα εκπαιδευει τα δεδομενα που προοριζονται για εκπαιδευση με χρηση της λογιστικης παλινδρομησης.

Αυτο λειτουργει ως εξης:

H λογιστικη παλινδρομηση χρησιμοποιει την “σιγμοιδη συναρτηση” για να υπολογισει τις πιθανοτητες, ο μαθηματικος τυπος της οποιας ειναι:   
S(x)=1/1+e^(-x), οπου x= weights[1]\*x[1]+weights[2]\*x[2]+….....+weights[n]\*x[n], το οποιο ειναι η συναρτηση της λογιστικης παλινδρομησης.

Δημιουργειται ενας πινακας weights μεγεθους οσο και οι “NWords” συχνοτερες λεξεις που επιλεξαμε. Καθε στοιχειο του πινακα weights αντιστοιχει σε μια λεξη του πινακα words. Κατα την εκπαιδευση δεδομενων γινονται ITERATIONS επαναληψεις, οι οποιες προκαθοριζονται απο τον χρηστη (σε εμας 200).

Σε καθε μια επαναληψη, για καθε email προβλεπει το “label” του εν λογω email, και ενημερωνει τον πινακα των weights της καθε λεξης αναλογα με το αν η προβλεψη ηταν σωστη η οχι. Ο αλγοριθμος δινει εμφαση σε email για τα οποια η προβλεψη ηταν λαθος, και αυξανει τα βαρη των λεξεων τους.

Παραλληλα, υπολογιζει το Maximum Likelihood. Oσο πιο μεγαλο ειναι το Maximum Likelihood, τοσο πιο πιθανο ειναι ο αλγοριθμος να προβλεψει τα σωστα αποτελεσματα. Ιδανικα, το Maximum Likelihood αυξανεται μετα απο καθε ITERATION.

Στο τελος, εχοντας ενημερωσει τον πινακα των βαρων πληρως, τον χρησιμοποιει για να κανει προβλεψεις και να αξιολογησει τα δεδομενα προς testing, χρησιμοποιωντας την σιγμοιδη συναρτηση. Πρωτα, Εφαρμοζει τον τυπο της συναρτησης, και επειτα αναλογα με το αν η πιθανοτητα P του να ειναι “spam” το email, δηλαδη η τιμη της σιγμοιδης συναρτησης, ειναι P>0.5 η P<0.5 το κατατασσει στα spam η ham αντιστοιχα.

Ο αλγοριθμος εμφανιζει τα αποτελεσμετα τοσο για τα δεδομενα εκπαιδευσης οσο και για τα δεδομενα αξιολογησης, υπολογιζοντας το ποσοστο σφαλματος, αποτελεσματα ακριβειας, ανακλησης και F1.

Ενα παραδειγμα χρησης του αλγοριθμου, με συνολο δεδομενων τον φακελο Ling-Spam, και διαχωρισμος των δεδομενων εκπαιδευσης απο τα δεδομενα αξιολογησης με αναλογια 7 προς 1 ειναι το εξης.

**Train dataset Results**

Accuracy: 0.9966730517189233.

**Positive class:**

Presicion: 0.9818181818181818

Recall: 0.9981515711645101

F1 Score: 0.9899175068744271

**Negative class:**

Presicion: 0.9996371991776515

Recall: 0.9963837994214079

F1 Score: 0.9980078478720192

**Test dataset Results**

Accuracy: 0.9951632406287787.

**Positive class:**

Presicion: 0.9744525547445255

Recall: 0.996268656716418

F1 Score: 0.985239852398524

**Negative class:**

Presicion: 0.9992753623188406

Recall: 0.9949494949494949

F1 Score: 0.9971077368040492