**Ιωάννης Χαραλάμπους**

**ΑΜ:1059685**

**Ερώτημα 1.** Προεπεξεργασία δεδομένων

Να αναφέρετε, πόσα είναι τα χαρακτηριστικά κάθε δείγματος, πόσα δείγματα εκπαίδευσης περιέχει το αρχείο και πόσα δείγματα έχει η κάθε κατηγορία.

Το εύρος τιμών των δεδομένων που σας έχουν δοθεί διαφέρει σημαντικά ανά χαρακτηριστικό. Για αυτό τον λόγο, για να μην υπερεκτιμηθεί η συνεισφορά κάποιου χαρακτηριστικού έναντι άλλων, θα πρέπει πριν την επεξεργασία των χαρακτηριστικών εισόδου να κανονικοποιηθούν στο εύρος [-1,1]. Χρησιμοποιήστε το matlab (ή όποια άλλη εφαρμογή θέλετε) τόσο για το διάβασμα του αρχείου που σας δίνεται όσο και για την κανονικοποίηση των δεδομένων εισόδου στο εύρος τιμών [-1,1].

Για να κανονικοποιήσω τον πίνακα diabetes χρησιμοποιώ την εντολή της matlab normalize(diabetes,[-1 1]; και προκύπτει ο πίνακας : 

**Ερώτημα 2.** Στο μάθημα συζητήθηκε εκτεταμένα ο ταξινομητής Bayes. Στη βιβλιογραφία, υπάρχει μια παραλλαγή του που λέγεται Αφελής Ταξινομητής Bayes (Naïve Bayes), με την υπόθεση ότι τα χαρακτηριστικά είναι στατιστικά ανεξάρτητα. Αναζητήστε τη σχετική βιβλιογραφία στο Internet, και να κάνετε μια σύντομη παρουσίαση του αλγορίθμου. Στη συνέχεια να κάνετε μια σύγκριση με τον Ταξινομητή Bayes.

Οι Naïve Bayes είναι μια οικογένεια απλών «πιθανοτικών ταξινομητών» που βασίζονται στην εφαρμογή θεωρήματος Bayes με ισχυρές (naïve) υποθέσεις ανεξαρτησίας μεταξύ των χαρακτηριστικών. Είναι από τα πιο απλά μοντέλα του Bayesian δικτύου. Οι ταξινομητές Naïve Bayes είναι εξαιρετικά κλιμακωτοί, απαιτώντας έναν αριθμό γραμμικών παραμέτρων στον αριθμό των μεταβλητών (χαρακτηριστικά / πρόβλεψη) σε ένα μαθησιακό πρόβλημα. Η εκπαίδευση μέγιστης πιθανότητας μπορεί να γίνει με την αξιολόγηση μιας έκφρασης κλειστής μορφής, η οποία παίρνει γραμμικό χρόνο, παρά με ακριβή επαναληπτική προσέγγιση, όπως χρησιμοποιείται για πολλούς άλλους τύπους ταξινομητών.

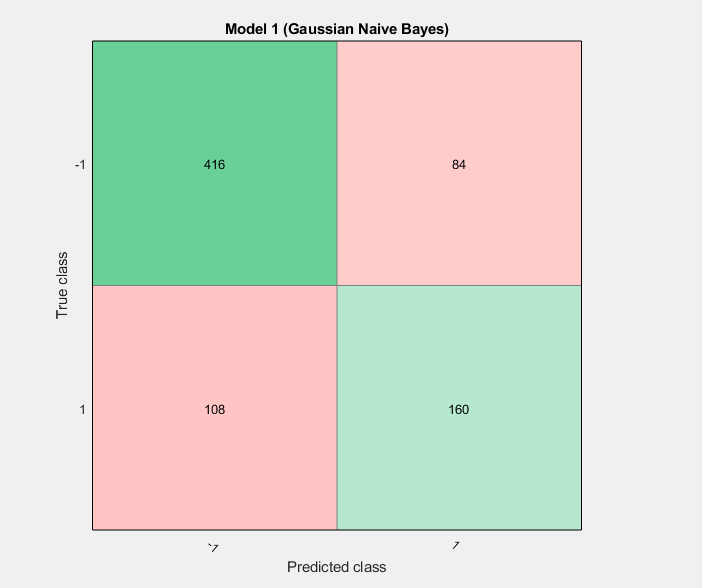
Ο Naive Bayes υποθέτει την ανεξαρτησία υπό όρους , *P*(*X*|*Y*,*Z*)=*P*(*X*|*Z*), ενώ ο γενικότερος Bayes θα επιτρέψει στον χρήστη να καθορίσει ποια χαρακτηριστικά είναι στην πραγματικότητα υπό όρους ανεξάρτητα. Ένα μεγάλο θετικό του Naive Bayes, είναι πως δεν χρειάζεται μεγάλο data set εκπαίδευσης για να εκτιμήσει τις κατάλληλες παραμέτρους για την ταξινόμηση. Είναι αρκετά χρήσιμος όταν έχουμε δυναμικά δεδομένα πολλών διαστάσεων που μπορεί να μεταβάλλονται αρκετά, καθώς προσαρμόζεται γρήγορα σε αλλαγές.

***Βιβλιογραφία ερωτήματος 2:***

1. **https://en.wikipedia.org/wiki/Naive\_Bayes\_classifier**
2. **https://stats.stackexchange.com/questions/212240/the-difference-between-the-bayes-classifier-and-the-naive-bayes-classifier**

**Ερώτημα 3.** Με χρήση της μεθόδου 5-fold cross validation, να εκπαιδεύσετε τον Naïve Bayes ταξινομητή, να παρουσιάστε και να σχολιάσετε την απόδοσή του. Μπορείτε να χρησιμοποιήσετε κατάλληλες συναρτήσεις του matlab ή οποιαδήποτε εφαρμογή επιθυμείτε (π.χ. το WEKA (http://www.cs.waikato.ac.nz/ml/weka/ ) ή να

υλοποιήσετε δικό σας κώδικα). Για την αξιολόγηση της απόδοσης του ταξινομητή να χρησιμοποιήσετε τις μετρικές του ερωτήματος 4, παρακάτω.

Χρησιμοποιώ την εφαρμογή της Matlab classification learner για να εκπαιδεύσω τον κανονοικοποιημένο πινακα diabetes .Προκύπτει ακρίβεια=75% και από το confusion matrix βρίσκω τα TP,TN,FP,FP ο οποίος ειναι : 

Για να αξιολογήσουμε την απόδοση του θα υπολογίσουμε την μετρική του γεωμετρικού μέσου της ευαισθησίας και της ειδίκευσής του. Γνωρίζουμε ότι η ευαισθησία(sensitivity) ενός ταξινομητή προκύπτει απο την σχέση TP/(TP+FN) και είναι το πόσο συχνά ο ταξινομητής αποφασίζει ναι(true), όταν και στην πραγματικότητα η τιμή είναι ναι(true).

Sensitivity= TP/(TP+FN)=416/(416+64)=0.832

Αντίστοιχα για την ειδίκευση(specificity) ή αλλιώς True Negative Rate, είναι το πόσο συχνά ο ταξινομητής αποφασίζει όχι, όταν και στην πραγματικότητα η τιμή είναι όχι και το βρίσκουμε με την σχέση TN/(TN+FP).

Specificity= TN/(TN+FP)=108/(108+160)=0.402985

Ο γεωμετρικός μέσος ειναι :

**Geometric Mean = sqrt (Sensitivity \* Specificity)=**= 0.5790367173159229‬

Ο γεωμετρικός μέσος μας και το confusion matrix μας οδηγεί στο συμπέρασμα ότι ο ταξινομητής μας δεν ειναι έμπιστος για το πρόβλημά μας.Παρατηρούμε λοιπόν ότι ο ταξινομητής έχει μεγαλύτερη επιτυχία όταν αποφασίζει true δηλαδή ότι κάποιος πάσχει από διαβήτη παρά όταν αποφασίζει false δηλαδή όταν αποφασίζει ότι κάποιος δεν πάσχει.Με αποτέλεσμα το όριο απόφασης μετατοπίζεται προς την δεύτερη κλάση, δηλαδή υπάρχουν αρκετές περιπτώσεις όπου κάποιος ταξινομείται στους μη πάσχοντες ενώ είναι.

**Ερώτημα 4.** Με χρήση της μεθόδου 5-fold cross validation και του matlab (ή οποιαδήποτε εφαρμογή επιθυμείτε π.χ. το Microsoft Azure Machine Learning Studio https://studio.azureml.net/), να εκπαιδεύσετε τους παρακάτω ταξινομητές, να παρουσιάστε και να σχολιάσετε την απόδοσή τους:

* + • Support Vector Machines (με Radial Basis Function kernel function): o Ρυθμίστε την παράμετρο C με διαδοχική αναζήτηση του βέλτιστου C στο διάστημα 1-200 με βήμα 5 και χρήση γραμμικών SVM. Στη συνέχεια, ρυθμίστε την παράμετρο γ με χρήση του βέλτιστου C που βρέθηκε από πριν, και διαδοχική αναζήτηση του βέλτιστου γ στο διάστημα 0-10 με βήμα 0.5 και χρήση RBFSVM.
  + • Ταξινομητής K-Κοντινότερου Γείτονα o Ρυθμίστε την παράμετρο Κ με διαδοχική αναζήτηση της βέλτιστης τιμής στο διάστημα 3-10.

Ποιο συγκεκριμένα ζητείται να παρουσιάσετε για κάθε ταξινομητή την μέση απόδοση του με χρήση 5 fold cross validation σε σχέση με την μετρική του γεωμετρικού μέσου (Geometric Mean) της ευαισθησίας (Sensitivity) και της ειδίκευσής (Specificity) του:

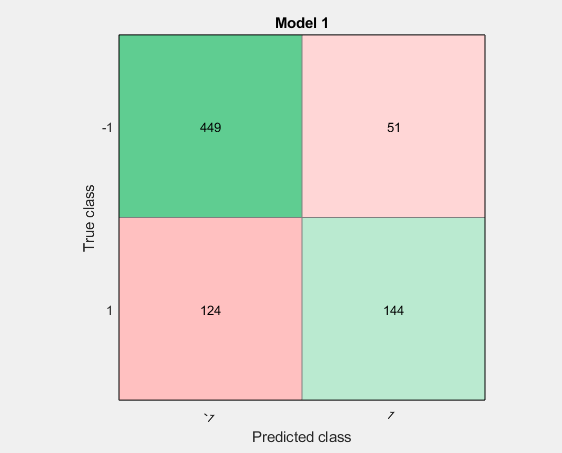
***Geometric Mean = sqrt (Sensitivity \* Specificity)***

Η μετρική αυτή χρησιμοποιείται για προβλήματα ταξινόμησης όπου παραδείγματα εκπαίδευσης της μίας κλάσης είναι περισσότερα από τα παραδείγματα εκπαίδευσης της άλλης κλάσης.

Στη συνέχεια, παρουσιάστε τα ενδιάμεσα αποτελέσματα που πήρατε από τα πειράματα για την ρύθμιση των παραμέτρων των αλγορίθμων. Γιατί η κάθε μέθοδος ταξινόμησης δίνει διαφορετικά αποτελέσματα; Με βάση τα παραπάνω αποτελέσματα ποιά από τις μεθόδους προτείνετε εσείς να χρησιμοποιηθεί για το παραπάνω πρόβλημα και γιατί;

Για να βρω τα βέλτιστα C,γ έγραψα ένα script το οποίο εκπαιδεύει 200 φορές τον ταξινομητή linear SVM και κρατάει κάθε φορά τις τιμές του σε έναν πίνακα ,οι οποίες αυξάνονται με βήμα 5 και προκύπτει C=1.Στη συνέχεια εκπαιδεύει τον RBFSVM 10 φορές με την παράμετρο C που βρήκα πριν ,με βήμα 0.5 και τα αποθηκεύει σε έναν πίνακα .Το βελτιστο γ=6.5

Προκειμένου να βρω τα Sensitivity και Specicifity χρησιμοποιώ την εφαρμογή Classification Learner της Matlab αφού ορίσω τις παραμέτρους C=1 και γ=6.5 και εκπαιδεύω τον κανονικοποιημένο πίνακα προκύπτει ότι η ταξινόμηση έχει ακρίβεια=77.2% και το ακόλουθω confusion matrix :



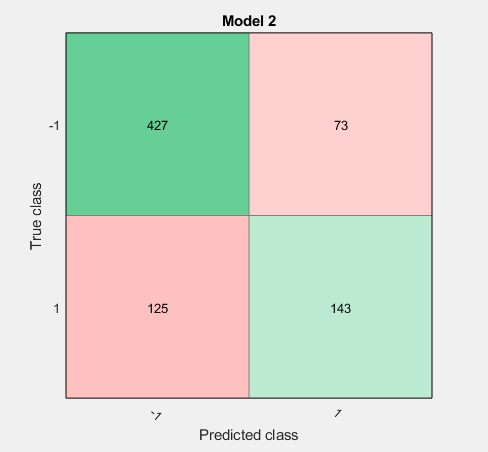
Sensitivity = 449/(449+51)=0.898

Specificity = 144/(124+144) = 0.537

*Geometric Mean=*

Για τον ταξινομητή kNN έγραψα έγραψα ενα script το οποίο εκπαιδεύει 200 φορές τον ταξινομητή για να βρούμε το βέλτιστο k με διαδοχική αναζήτηση στο διάστημα 3-10.Βρίσκω βέλτιστο k=9 .

Προκειμένου να βρω τα Sensitivity και Specicifity χρησιμοποιώ την εφαρμογή Classification Learner της Matlab αφού ορίσω τις παραμέτρους k=9και εκπαιδεύω τον κανονικοποιημένο πίνακα προκύπτει ότι η ταξινόμηση έχει ακρίβεια=74.2% και το ακόλουθω confusion matrix :



Sensitivity = 427/(427+73)=0.854

Specificity = 125/(125+143) = 0.4664

*Geometric Mean=*

Τα ενδιάμεσα αποτελέσματα των παραμέτρων κατά την εύρεσή τους βρίσκονται στους εξής πίνακες :

Συμπεραίνοντας ο RBF SVM δίνει τα καλύτερα αποτελέσματα ως προς την ακρίβεια, επομένως είναι η καλύτερη μέθοδος για αυτό το πρόβλημα. Ταξινομεί με μεγαλύτερη ακρίβεια τους ασθενείς στη σωστή κλάση.Τα διαφορετικά αποτελέσματα που παίρνουμε με κάθε μέθοδο ωφείλονται στον τρόπο με τον οποίο δουλεύει ο κάθε ταξινομητής. Ο SVM χρησιμοποιώντας βοηθητικά διανύσματα προσπαθεί να χωρίσει τα δεδομένα των 2 κλάσεων με τον πιο βέλτιστο τρόπο, ενώ ο KNN αποφασίζει για την κλάση κάθε στοιχείου σύμφωνα με την κλάση που βρίσκονται οι περισσότεροι από τους κοντινότερους γείτονες του.