Ömer Faruk Akgün

54076

ENGR 421- Homework 1

First two lines of the code reads the data sets into memory. Then I combine these two into one dataset such that the labels are in the 4097th row and split this data into two separate datasets as named “testset” and “trainset”. Line # 14 & 15 creates two subsets, separates the ones labeled as female and female. 18 & 19 calculates the means for each column of these two subsets and keeps them in a list. 20 combines the two lists.

I created a function named “sdpop” to calculate the population standard deviation since the sd function in R uses the denominator n-1. Following 4 lines calculates variances and creates the covariance matrices. Covariance matrices are diagonal since we assumed that the features are independent. This is the core assumption of Naive Bayes’, and this makes it a special case of multivariate parametric classification problem that we discussed in the lecture 5. Then I calculated priors using the nrow function.

The score function is:

$g(x) = X^T . W_d . X + w_d . X + w_{d0}$

Coefficients here are the difference of the ones that has been calculated for the class 1 & class 2. Instead of selecting the one with the maximum we look at the difference and choose accordingly. Wd, wd and wd0 were calculated using almost the same formulas that we derived in the lecture. There was one small adjustment I had to make due to numerical reasons. While calculating the log determinant of covariance matrix using the det() function and then getting the log does not work since the determinant is very small, but we know the determinant of a diagonal matrix is the product of all diagonal entries so I flipped the formula around and summed log-entries.

Then I evaluated the function for each column (except labels) of trainset and testset then built the confusion matrix by using the table() function.

Note that we could have used the same score function for a general multivariate parametric classification solution instead of Naïve Bayes. However, in that case our covariance matrix would not be invertible since the data set is “short and fat” (i.e N<D 400 vs 4096 in our case)