MTH6101 Introduction to Machine Learning

Laboratory week nine

The aim of this practice is to build different classifiers on the same data set and to compare them. The file `glass.data` has measurements of 10 variables in seven different types of glass. This file has 11 columns, and the last column is the type of glass. We are interested in classifying headlamps (type of glass equals seven) against all the other types of glass.

When you **start** your session, open `RStudio` and install/load the following libraries: `cvTools`, `class`, `tree`, `MASS`, `pROC`.

1. Reading the data and adapting the file for analysis.

   Read the data using the command `read.csv(file = "glass.data",header = FALSE)`, store it in a variable termed `X`. Save the last column of `X` in a variable called `Y` and remove the first column. Then center and scale the matrix `X` using the command `scale`, saving it in `X`.

   Now we will prepare the output for analysis. The variable `Y` will be used twice, first as 0/1 variable then as "Yes"/"No". Replace `Y` by `(Y==7)*1` and substitute the last column of `X` for this value of `Y`.

   At this point, the first 9 columns of `X` have the centered values and the last one has 0/1 values. Give names to the columns of `X` according to
   `colnames(X)<-c("RI","Na","Mg","Al","Si","K","Ca","Ba","Fe","Type")`
   Finally, convert the variable `Y` to "Yes"/"No" by the commands `Y[Y==1]<-"Yes"; Y[Y==0]<-"No"`. Merge `X` and `Y` in a data frame called `DAT` by `DAT<-data.frame(X,Y)`.

2. Now we create the partition of data into training and testing datasets. For this, we create a partition 66:33. Set seed equal to zero and using `cvFolds` from `cvTools`, create variables `Train` and `Test` for the training and testing partition respectively. See the notes where this has been done in virtually every example.

3. Using the training data, fit the **logistic classifier** for the response variable `Type` using all but the last column of the data (i.e. `DAT[Train,-11]`). Save the fitted model in variable `M1`. Then predict the response using the fitted model `M1` and the test partition. Save this in a variable called `P1`.

4. Examine the fitted classifier and identify variables that are not important for the response. Using only those variables that were found significant in the

first logistic model, fit a second **logistic classifier** and a second prediction set. These are to be called `M11` and `P11`.

5. Fit a K **nearest neighbors classifier**. Here use three nearest neighbors in the function `knn` from the library `class` and recall that only columns `1:9` of the data frame contain variables. Call this model `M2` and recall that `M2` already contains the predicted classes.

6. Fit a **tree classifier** to the data using the function `tree` from library `tree`. The response is `Y` and here the column ten from the data frame is not to be used (i.e. `DAT[Train,-10]`). Save the fitted model in variable called `M3`. Predict output with option `type="class"` and save it in a variable called `P3`.

7. Fit a **linear discriminant classifier** using the function `lda` from library `MASS`. Use the training data except the last column (i.e. `DAT[Train,-11]`) to fit variable `Type` and save the output in a variable termed `M4`. Using the test data, predict output using the fitted model and save results in variable termed `P4`.

8. Now we prepare to compare all classifiers. To this end, recall that we can do ROC curve for **logistic** and **linear discriminant** classifiers. For KNN and tree we will compute only points in ROC graph. Using the function `roc` from the library `pROC`, compute and ROC for models `M1`, `M11` and `M4` and save the results in variables `R1`, `R11` and `R4`.

9. For each of **KNN** and **tree** classifiers, compute the confusion matrix using the command `table`. In each case, compute the figures TPR and FPR. Save results in variables `TPR2,FPR2` and `TPR3,FPR3`.

10. In a single ROC graph, plot ROC curves for **logistic** and **linear discriminant** classifiers; add points for the **KNN** and **tree** classifiers. Compute AUC for the classifiers `M1`, `M11` and `M4` and summarize your results.

11. (**Extra**) Plot the tree for classifier `M3` and interpret it.