Machine Learning and Data Mining Assignment 2

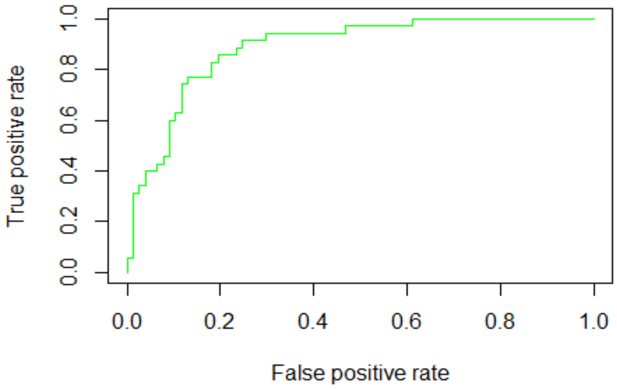
Surya Balakrishnan Ramakrishnan (18231072)

MSc Computer Science (Data Analytics)

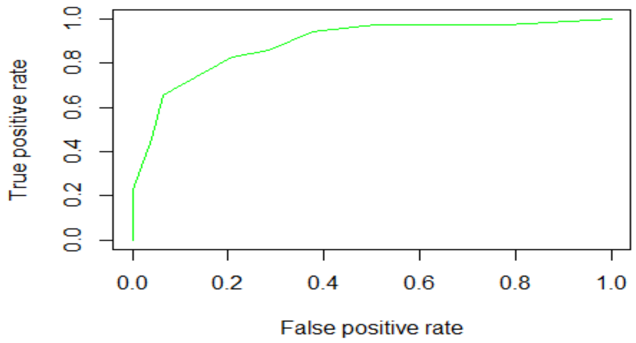
2nd November 2018

```r
# Importing the Libraries necessary for the assignment
library(lattice)
library(caret)
library(e1071)
library(ggplot2)
library(ROCR)
library(gplots)

#Reading the CSV file which was used in the last assignment Autoimmune.csv
data <- read.csv(file ="Autoimmune.csv",header=TRUE, sep =",")

# seperate the data into training and validation data (test dataset)

set.seed(101)
index <- createDataPartition(data$Autoimmune_Disease, p = 0.7, list = F )
train <- data[index,]
validation <- data[-index,]

# Setting levels for both training and validation data
levels(train$Autoimmune_Disease) <- make.names(levels(factor(train$Autoimmune_Disease)))

levels(validation$Autoimmune_Disease) <- make.names(levels(factor(validation$Autoimmune_Disease)))

# Performing the 10 fold cross validation
repeats <- 3
numbers <- 10
tunel <- 10

set.seed(1234)
x <- trainControl(method = "repeatedcv",
          number <- numbers,
          repeats <- repeats,
          classProbs = TRUE,
          summaryFunction = twoClassSummary)

# Performing the Naive Bays Classifier
Naive_Bayes <- naiveBayes(Autoimmune_Disease~., data=data)
nbprediction <- predict(Naive_Bayes, validation, type='raw')

# Printing out the confusion matrix
Naive_Bayes1 <- predict(Naive_Bayes,validation)
print(table(Naive_Bayes1,validation$Autoimmune_Disease))
```

```
##
## Naive_Bayes1 negative positive
##     negative      70       13
##     positive       7       22

# Generating the score and ROC Curve
score <- nbprediction[, "positive"]
actual.class <- validation$Autoimmune_Disease
pred <- prediction(score, actual.class)
nb.prff  <-  performance(pred, "tpr", "fpr")
plot(nb.prff, col = "green")
```



```
# Calculating the Area under the curve (AUC)
auc <- performance(pred,"auc")
print(auc@y.values[[1]])

## [1] 0.8875696

# Perfoming the KNN algorithm
knnmodel <- train(Autoimmune_Disease~. , data = train, method = "knn",
            preProcess <- c("center","scale"),
            trControl <- x,
            metric <- "ROC",
            tuneLength <- tunel)

# Validation
valid_pred <- predict(knnmodel,validation, type = "prob")

#Storing Model Performance Scores

pred_val <-prediction(valid_pred[,2],validation$Autoimmune_Disease)

# Plot the ROC curve
perf_val <- performance(pred_val, "tpr", "fpr")
plot(perf_val, col = "green", lwd = 1.5)
```



```
# Calculating Area under Curve (AUC)
perf_val <- performance(pred_val,"auc")
print(perf_val@y.values[[1]])

## [1] 0.8883117
```

Methodology used in the construction of the ROC Curve.

1) After loading the packages required and given data set into the R interface the first step involves separating the data into training and validation (test) data sets, the createDataPartition() is used to partition the data. Here validation is the test data set with 1/3rd of the overall data set.

2) The nest step involves setting the levels for both training and validation data (test dataset) and performing 10 fold cross validation of the data set.

3) In the next step we implement both the KNN and Naive Bayes Classifier and print out the confusion matrix of the classification. In case of KNN we also determine which is the most optimal value for N (Number of neighbors)

4) The next step involves scoring the algorithm by determining how many records the algorithm predicted correctly. Once the score of the algorithm is obtained we generate the ROC curve along with the AUC (area under the curve.)

Observations from the two ROC curves.

1) The ROC curve of the KNN algorithm is smoother than the Naive Bayes algorithms ROC curve. One of the implications could be that the Naive Bayes algorithm is providing discrete predictions rather than a continuous score. Another probable reason could be that the Naive Bayes classifier needs more data to efficiently classify the given data set.

2) KNN performs better in the initial stages as compared to the Naive Bayes classifier, which is indicated by the steep upward curve in the beginning. More the angle made by the curve between X and Y axis better is the chance that the algorithm predicted it correctly.

3) The area under the curve (AUC) is similar for both the algorithms which suggests that both the algorithm have a similar accuracy rates. Even though the KNN had an advantage initially but it ends up giving similar results as compared to the Naive Bayes classifier. This suggests that the KNN best works for a small dataset and as the data size increases the performance decreases. On the contrary more the data better the Naive Bayes classifier performs which is evident from the unsmooth curve obtained.

References:

1) https://www.r-bloggers.com/k-nearest-neighbor-step-by-step-tutorial/

2) https://blog.revolutionanalytics.com/2016/08/roc-curves-in-two-lines-of-code.html

3) https://stats.stackexchange.com/questions/191805/r-plotting-a-roc-curve-for-a-naive-bayes-classifier-using-rocr-not-sure-if-i

4) http://gim.unmc.edu/dxtests/roc2.html

5) https://www.quora.com/What-does-it-mean-when-an-ROC-curve-is-not-smooth

6) http://fastml.com/what-you-wanted-to-know-about-auc/