**Assignment #3: Classification**

**Submit through link: eCampus -> Assignments->Assignment 3 Submission**

**Deadline: October 9 (Tuesday) @12:00 pm**

**The filename should have this format: LastName-FirstName-hw03.doc**

**Problem 1 (12pt)**

This question should be answered using the Weeklydata set, which is part of the ISLR package. This data is similar in nature to the Smarketdata, except that it contains 1*,*089 weekly returns for 21 years, from the beginning of 1990 to the end of 2010.

(a) Produce some numerical and graphical summaries of the Weeklydata. Do there appear to be any patterns?

**Answer: summary(Weekly)**

**pairs(Weekly)**

**The only distinctly apparent relationship is between the year and Volume of trading. We can see from the scatterplots that the volume has been increasing every year throughout this 21 years timespan**.

(b) Use the full data set to perform a logistic regression with Directionas the response and the five lag variables plus Volumeas predictors. Use the summary function to print the results. Do any of the predictors appear to be statistically significant? If so, which ones?

**Answer: log.fit<- glm(Direction~Lag1 + Lag2 + Lag3 + Lag4 + Lag5 + Volume, data=Weekly, family = binomial)**

**summary(log.fit)**

(c) Compute the confusion matrix and performance measures (accuracy, error rate, sensitivity, specificity). Explain what the confusion matrix is telling you about the types of mistakes made by logistic regression. Does the error rate represent the performance of logistic regression in prediction? (hint: is it training error rate or test error rate?)

**Answer: log.prob<-predict(log.fit,type = "response")**

**log.pred<-rep("Down",1089)**

**log.pred[log.prob>0.5]="Up"**

**table(log.pred,Direction)**

Direction

log.pred Down Up

Down 54 48

Up 430 557

**Accuracy = (54+557)/(54+48+430+557)= 56.1%**

**Error Rate= 1- Accuracy= 43.9%**

**Sensitivity= 557/(557+48)= 92.1%**

**Specificity= 54/(54+430)= 11.2%**

**The confusion matrix tells us about the True values of the class to be predicted against the predicted values by that particular model. High Accuracy/ Low error rate tells us about the overall accuracy of the model. While sensitivity and specificity tells us more about the details we are concerned with such as how much of the values predicted true by the model is actually true or how many of the false values can it identify correctly.**

**No this error rate does not represent the model performance on the prediction actually because here the training and test data are the same so we have no way to know if the model was overfitted or not without using a test data.**

(d) Now fit the logistic regression model using a training data period from 1990 to 2008, with Lag2as the only predictor. Compute the confusion matrix and performance measures (accuracy, error rate, sensitivity, specificity) for the held out data (that is, the data from 2009 and 2010).

**Answer: train<- Year< 2009**

**data.test<- Weekly[!train,]**

**log.fit<-glm(Direction~Lag2, data= Weekly, family = binomial, subset = train)**

**log.prob<-predict(log.fit, newdata = data.test,type = "response")**

**log.pred<-rep("Down", length(log.prob))**

**log.pred[log.prob>0.5]="Up"**

**table(log.pred,data.test$Direction)**

log.pred Down Up

Down 9 5

Up 34 56

**Accuracy = (9+56)/(9+56+5+34)=62.5%**

**Error Rate= 1- Accuracy= 37.5%**

**Sensitivity= 56/(56+5)= 91.8%**

**Specificity=9/(9+34)= 21%**

(e) Repeat (d) using LDA.

**Answer: library(MASS)**

**lda.fit<- lda(Direction~Lag2, data=Weekly, subset= train)**

**lda.pred<-predict(lda.fit,newdata = data.test)**

**table(lda.pred$class, data.test$Direction)**

|  |
| --- |
| Direction.test  Down Up  Down 9 5  Up 34 56 |
|  |
| |  | | --- | |  | |

**Accuracy = (9+56)/(9+56+5+34)=62.5%**

**Error Rate= 1- Accuracy= 37.5%**

**Sensitivity= 56/(56+5)= 91.8%**

**Specificity=9/(9+34)= 21%**

(f) Repeat (d) using QDA.

**Answe: qda.fit<- qda(Direction~Lag2, data=Weekly, subset= train)**

**qda.pred<-predict(qda.fit,newdata = data.test)**

**Direction.test<-data.test$Direction**

**table(qda.pred$class,Direction.test )**

Direction.test

Down Up

Down 0 0

Up 43 61

**Accuracy = (61)/(61+43)=58.6%**

**Error Rate= 1- Accuracy= 41.3%**

**Sensitivity= 61/(61+43)= 58.6%**

**Specificity=0/(0+43)= 0**

(g) Repeat (d) using KNN with *K* = 1.

**Answer:**

**library(class)**

**train.x<- as.matrix(Lag2[train])**

**test.x<-as.matrix(Lag2[!train])**

**direction.train<- as.matrix(Direction[train])**

**set.seed(1)**

**knn.pred<-knn(train = train.x,test = test.x,direction.train, k = 1)**

**table(knn.pred, Direction.test)**

Direction.test

knn.pred Down Up

Down 21 30

Up 22 31

**Accuracy = (31+21)/(30+31+21+22)=50%**

**Error Rate= 1- Accuracy= 50%**

**Sensitivity= 31/(31+30)= 50.8%**

**Specificity=21/(21+22)= 48.8%**

(h) Which of these methods appears to provide the best results on this data?

**Answer: LDA and Logistic regression seem to be the best methods from the above used methods for this particular problem.**

(i) Experiment with different combinations of predictors, including possible transformations and interactions, for each of the methods. Report the variables, method, and associated confusion matrix that appears to provide the best results on the held out data. Note that you should also experiment with values for *K* in the KNN classifiers.

**Answer: All the possible combinations tried with different predictors and different K values(k=5,10) for KNN classifiers do not reach the accuracy of the original Logistic and LDA method used in the earlier parts of the problem. So the Logistic and LDA already used in the above sections seems to offer the best results by far in terms of accuracy.**

**For logistic regression with only variable Lag 2 we have mean(log.pred == Direction) = 0.625 .**

**For LDA with only Lag 2 we have mean(lda.pred$class==Direction.test)= 0.625**

**Both equally good and best in the whole lot.**

**Problem 2 (5pt)**

Perform ROC analysis and present the results for logistic regression and LDA used for the best model chosen in Question 1(i).

**Answer**

**#######---------------------------------roc for Logistic regression-----**

**library(ROCR)**

**train<- Year< 2009**

**data.test<- Weekly[!train,]**

**log.fit<-glm(Direction~Lag2, data= Weekly, family = binomial, subset = train)**

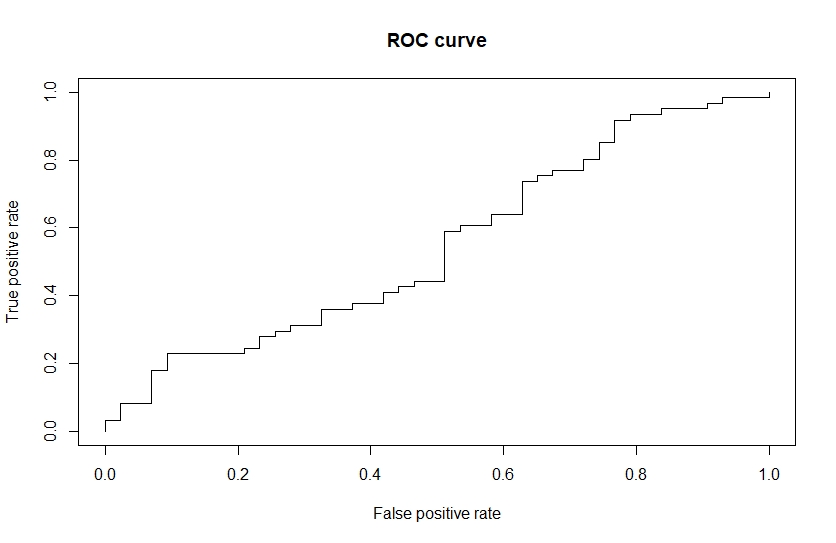
**log.prob<-predict(log.fit, newdata = data.test, type = "response")**

**log.pred= prediction(log.prob, data.test$Direction)**

**roc= performance ( log.pred, 'tpr', 'fpr')**

**plot (roc, main= 'ROC curve')**

**ablibe(a=0, b=1)**



**#######---------------------------------roc for LDA-----**

**library(MASS)**

**lda.fit<- lda(Direction~Lag2, data=Weekly, subset= train)**

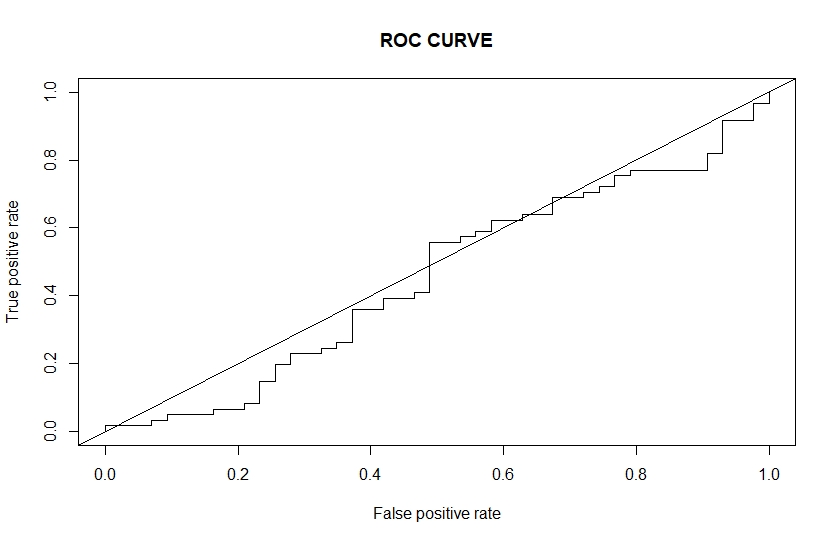
**lda.pred<-predict(lda.fit,newdata = data.test)**

**roc.pred= prediction(lda.pred$posterior[,1], data.test$Direction)**

**roc=performance(roc.pred, 'tpr', 'fpr')**

**plot(roc,main='ROC CURVE')**

**abline(a=0, b=1)**

****

**The ROC curves of both the best fitted curves are not good on ROC criteria there is not much difference in performance of the classifiers based on the value of threshold. Both the curves are not going into the top left corner of the plot and it is probably right because the data at hand is a stock market data and it is a very difficult task to find pattern in this data and predict the market direction. It has been a subject of interest for may decades but no one has come closed to actually predicting it right so the ROC curves are not that strongly evident of these models good performance.**

**Problem 3 (11pt)**

In this problem, you will develop a model to predict whether a given car gets high or low gas mileage based on the Autodata set.

(a) Create a binary variable, mpg01, that contains a 1 if mpgcontains a value above its median, and a 0 if mpgcontains a value below its median. You can compute the median using the median( )function. Note that you may find it helpful to use the data.frame( )function to create a single data set containing both mpg01and the other Autovariables.

**Answer: library(ISLR)**

**summary(Auto)**

**attach(Auto)**

**mpg01<-rep(0, length(mpg))**

**mpg01[mpg> median(mpg)]<- 1**

**Auto <-data.frame(Auto, mpg01)**

(b) Explore the data graphically in order to investigate the association between mpg01and the other features. Which of the other features seem most likely to be useful in predicting mpg01? Scatterplots and Boxplots may be useful tools to answer this question. Describe your findings.

**Answer: pairs(Auto)**

**cor(Auto)**

**by looking at the scatterplots and the correlation matrix we can say see that mpg01 is negatively correlated with cylinders, displacement, horsepower, weight.**

(c) Split the data into a training set and a test set.

**Answer: train = (year%%2 == 0)**

**data.train = Auto[train, ]**

**data.test = Auto[!train, ]**

**mpg01.test = mpg01[!train]**

(d) Perform LDA on the training data in order to predict mpg01using the variables that seemed most associated with mpg01in (b). What is the test error of the model obtained?

**Answer: lda.fit = lda(mpg01 ~ horsepower+ cylinders + displacement +weight, data = Auto,**

**subset = train)**

**lda.pred = predict(lda.fit, data.test)**

**mean(lda.pred$class = mpg01.test)**

**error rate= 1- 0.874 = 12.6%**

(e) Perform QDA on the training data in order to predict mpg01using the variables that seemed most associated with mpg01in (b). What is the test error of the model obtained?

**Answer: qda.fit = qda(mpg01 ~ horsepower+ cylinders + displacement +weight, data = Auto,**

**subset = train)**

**qda.pred = predict(qda.fit, data.test)**

**mean(qda.pred$class = mpg01.test)**

**Error rate = 1-0.869 = 13.1%**

(f) Perform logistic regression on the training data in order to predict mpg01using the variables that seemed most associated with mpg01in (b). What is the test error of the model obtained?

**Answer: glm.fit = glm(mpg01 ~ horsepower+ cylinders + displacement +weight, data = Auto,**

**family = binomial, subset = train)**

**glm.prob = predict(glm.fit, data.test, type = "response")**

**glm.pred = rep(0, length(glm.prob))**

**glm.pred[glm.prob > 0.5] = 1**

**mean(glm.pred = mpg01.test)**

**Error rate = 1- 0.879 = 12.1%**

(g) Perform KNN on the training data, with several values of *K*, in order to predict mpg01. Use only the variables that seemed most associated with mpg01in (b). What test errors do you obtain? Which value of *K* seems to perform the best on this data set?

**Answer:**

**train.X = cbind(cylinders, weight, displacement, horsepower)[train, ]**

**test.X = cbind(cylinders, weight, displacement, horsepower)[!train, ]**

**train.mpg01 = mpg01[train]**

**set.seed(1)**

**knn.pred = knn(train.X, test.X, train.mpg01, k = 1)**

**mean(knn.pred = mpg01.test)**

**Error rate= 1- 0.846 = 15.4%**

**knn.pred = knn(train.X, test.X, train.mpg01, k = 5)**

**mean(knn.pred != mpg01.test)**

**Error rate = 14.8%**

**knn.pred = knn(train.X, test.X, train.mpg01, k = 10)**

**mean(knn.pred != mpg01.test)**

**Error rate = 16.4%**

**knn.pred = knn(train.X, test.X, train.mpg01, k = 100)**

**mean(knn.pred != mpg01.test)**

**Error rate= 14.2 %**

**It has the lowest error rate for K= 100. Out of all K (1,5,10,100)**