­

Ch4. Statistical Learning

-Lab & Exercises #4,10,11

­­

**학번 : 182STG12**

**이름 : 오혜윤**

c

Problem Description

In Chapter 4 lab, we learn about Logistic Regression, LDA, QDA, and KNN. We fit various models in order to predict Direction using Smarket data. The first model is logistic regression and we use the function called glm(). In Smarket data, there are many variables such as Lag1, Lag2 and so on. We fit two glm models using different variables, Lag1 to Lag5 and Lag1 to Lag2 separately. As a result, the model with less variables has a higher accuracy rate. It is surprising because most people think that the model with many variables will be more correct. Also, we use LDA and QDA models to predict Direction. Interestingly, the QDA predictions are accurate almost 60% of the time, even though the 2005 data was not used to fit the model. Compared to the result of LDA (56%), the QDA model has the highest accuracy, 60%. This suggests that the quadratic form assumed by QDA may capture the true relationship more accurately than the linear forms assumed by LDA and logistic regression. After that, we use KNN models. At first, we use K=1 but the results were not good. Using K=3, the accuracy rate has improved slightly.

Finally, we can conclude that the best model is QDA because the accuracy rate is 60% (the highest rate).

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | glm1 | glm2 | LDA | QDA | KNN (K=1) | KNN (K=3) |
| accuracy | 52% | 58% | 56% | 60% | 50% | 54% |

Similarly, we apply the KNN approach to Caravan data. Any variables that are on a large scale can be a problem when using KNN so we should standardize the data. By standardizing the variables, all variables can be on a comparable scale. Anyway, in this dataset, as we increase K, we can get higher accuracy rate.

In Chapter 4 exercises, we compare the results of many models such as glm, LDA and QDA. In the first question, we calculate the fraction of available observations we will use to predict. In the second question, using Weekly dataset, we want to find the best model. By computing the model accuracy, we can find the highest accuracy by trying various models. In the third question, using Auto dataset, we can compare the test errors by trying four models, LDA, QDA, logistic regression and KNN. We split the data into a training dataset and a test dataset. After that, by fitting the model into a training dataset and compare the result of the model prediction and a test dataset, we can compute the test error. We can conclude that the best model is the model which has the lowest test error.

Results

**CH4. Lab Review**

In the lab, it was mostly about fitting the model to the given dataset. I already knew such models, so it was not that surprising. However, the results were quite surprising because the QDA model was the best model when we used the Smarket datset. I thought that fitting the simple model to dataset was better, but the results were not. Anyway, most of models regarding classification work better than the logistic regression models.

We can apply many approaches to the dataset in this lab. Comparing each model, we can choose the best model which fits well to the dataset. We can learn a lot about modeling in this section, so it was useful.

**CH4 Exercises**

**4. When the number of features p is large, there tends to be a deterioration in the performance of KNN and other local approaches that perform prediction using only observations that are near the test observation for which a prediction must be made. This phenomenon is known as the curse of dimensionality, and it ties into the fact that curse of dinon-parametric approaches often perform poorly when p is large. We will now investigate this curse.**

**(a) Suppose that we have a set of observations, each with measurements on p = 1 feature, X. We assume that X is uniformly (evenly) distributed on [0, 1]. Associated with each observation is a response value. Suppose that we wish to predict a test observation’s response using only observations that are within 10% of the range of X closest to that test observation. For instance, to predict the response for a test observation with X = 0.6, we will use observations in the range [0.55, 0.65]. On average, what fraction of the available observations will we use to make the prediction?**

It is clear that if x∈[0.05,0.95] then the observations are in the interval [x−0.05,x+0.05] and represents a length of 0.1 which is a fraction of 10%. If x<0.05, then we will use observations in the interval [0,x+0.05] which is a fraction of 100\*(x+0.05)%. Similarly, if x>0.95, then we will use observations in the interval [x-0.05,1] which is a fraction of 100\*(1.05-x)%. To compute the average fraction we will use to make the prediction we have to evaluate the following expression

So, we can conclude that, on average, the fraction of available observations we will use to make the prediction is about 10%.

**(b) Now suppose that we have a set of observations, each with measurements on p = 2 features, X1 and X2. We assume that (X1,X2) are uniformly distributed on [0, 1] × [0, 1]. We wish to predict a test observation’s response using only observations that are within 10% of the range of X1 and within 10% of the range of X2 closest to that test observation. For instance, in order to predict the response for a test observation with X1 = 0.6 and X2 = 0.35, we will use observations in the range [0.55, 0.65] for X1 and in the range [0.3, 0.4] for X2. On average, what fraction of the available observations will we use to make the prediction?**

The fraction of the available observations we will use to make the prediction is 0.1×0.1=0.01=1% because X1 and X2 are independent.

**(c) Now suppose that we have a set of observations on p = 100 features. Again the observations are uniformly distributed on each feature, and again each feature ranges in value from 0 to 1. We wish to predict a test observation’s response using observations within the 10% of each feature’s range that is closest to that test observation. What fraction of the available observations will we use to make the prediction?**

Like (b), if p is 100, then all Xs are independent. Therefore, The fraction of the available observations we will use to make the prediction is %

**(d) Using your answers to parts (a)–(c), argue that a drawback of KNN when p is large is that there are very few training observations “near” any given test observation.**

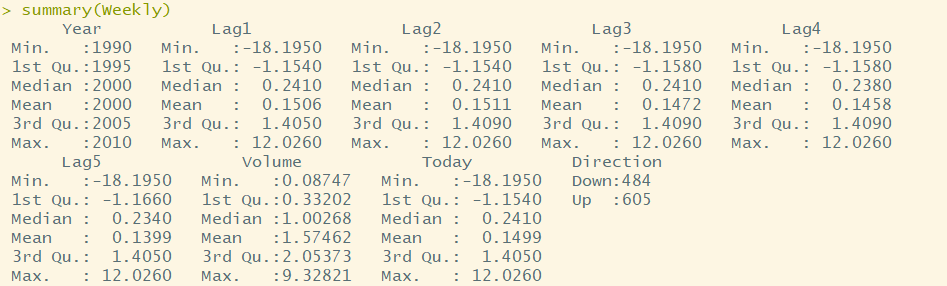
As seen in the previous answers, the fraction of available observations we will use to make the prediction decreases exponentially as p increases. Therefore, when p is large, there are very few training observations near any given test observation, so we cannot use KNN when p is large.

**(e) Now suppose that we wish to make a prediction for a test observation by creating a p-dimensional hypercube centered around the test observation that contains, on average, 10% of the training observations. For p = 1, 2, and 100, what is the length of each side of the hypercube? Comment on your answer.**

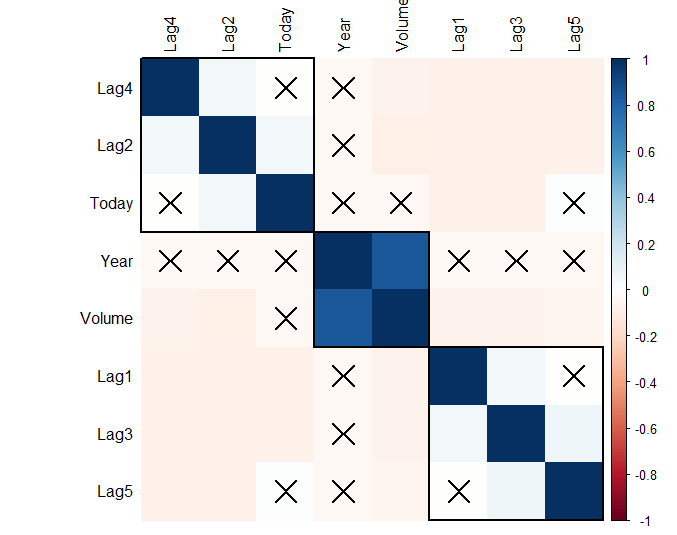
Considering that all predictors are distributed uniformly into the space, the hypercube’s volume will be 10% of the total volume. The length of each side of the hypercube will be the value of hypercube volume divided by . So, the length of each side of the hypercube is . Therefore, when p=1, length = 1. Also, for p=2, length=2.5 and for p=100, length =.

**10. This question should be answered using the Weekly data set, which is part of the ISLR package. This data is similar in nature to the Smarket data from this chapter’s lab, except that it contains 1, 089 weekly returns for 21 years, from the beginning of 1990 to the end of 2010.**

1. **Produce some numerical and graphical summaries of the Weekly data. Do there appear to be any patterns?**

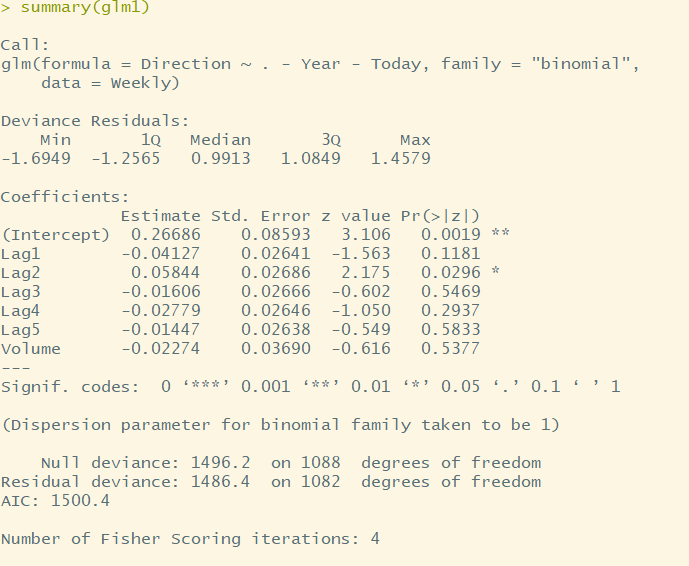


The lags show no clear pattern. There is little skew to be seen from the difference between median and mean.



The clustered correlation plot seems to insinuate that the stocks go up and down in two day pattern since the even days, today (lag0), lag2, lag4, have higher than 0 correlation and the same for the odd. The significance levels of the correlations and their absolute values are low, though.

1. **Use the full data set to perform a logistic regression with Direction as the response and the five lag variables plus Volume as predictors. Use the summary function to print the results. Do any of the predictors appear to be statistically significant? If so, which ones?**



The only statistically significant variable is Lag2 because its pvalue is 0.0296 which is less than 0.05. Lag1 is not very far from being significant as its pvalue is 0.1181 which is slightly larger than 0.05.

1. **Compute the confusion matrix and overall fraction of correct predictions. Explain what the confusion matrix is telling you about the types of mistakes made by logistic regression.**

|  |  |  |
| --- | --- | --- |
|  | Down | Up |
| Down | 54 | 48 |
| Up | 430 | 557 |

**Accuracy : 0.5611**

The chart above is the confusion matrix. The diagonal is where we get the prediction right but off the diagonal are the mistakes we’ve made. Therefore, we can conclude that the accuracy of prediction is (54+557)/1089=0.5611 (56.11%). Also, the lower left (predicted Up really Down) is what is called false positive and the upper right (predicted Down really Up) is false negative.

1. **Now fit the logistic regression model using a training data period from 1990 to 2008, with Lag2 as the only predictor. Compute the confusion matrix and the overall fraction of correct predictions for the held out data (that is, the data from 2009 and 2010).**

|  |  |  |
| --- | --- | --- |
|  | Down | Up |
| Down | 9 | 5 |
| Up | 34 | 56 |

**Accuracy : 0.625**

It seems the accuracy of the model was increased by leaving out unnecessary predictors. The prediction accuracy is 0.625 (62.5%) and it is quite a reasonable result.

|  |  |  |
| --- | --- | --- |
|  | Down | Up |
| Down | 9 | 5 |
| Up | 34 | 56 |

1. **Repeat (d) using LDA.**

**Accuracy : 0.625**

The prediction accuracy is 0.625 (62.5%) and it is the same as the result of (d).

|  |  |  |
| --- | --- | --- |
|  | Down | Up |
| Down | 0 | 0 |
| Up | 43 | 61 |

1. **Repeat (d) using QDA.**

**Accuracy : 0.5865**

The QDA model is not better than the LDA model. Its accuracy is only 58% which is less than 62.

|  |  |  |
| --- | --- | --- |
|  | Down | Up |
| Down | 21 | 30 |
| Up | 22 | 31 |

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

**Accuracy : 0.5**

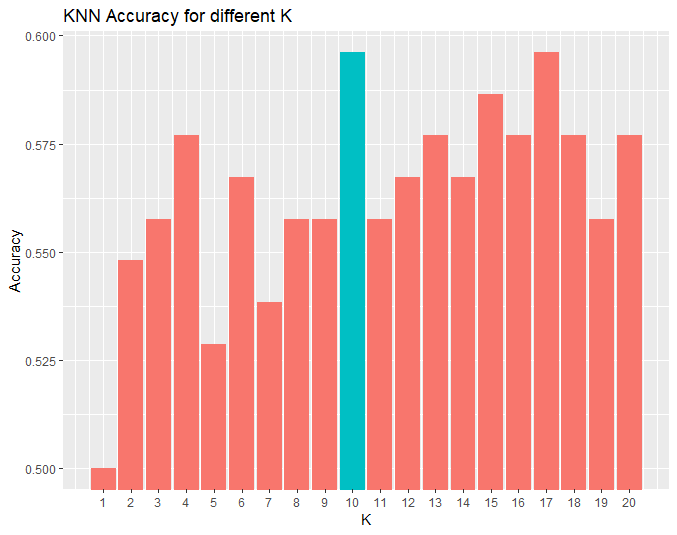
The prediction accuracy is 0.5 (50%) and it is the worst result among all models.

1. **Which of these methods appears to provide the best results on this data?**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| *method* | GLM1 | GLM2 | LDA | QDA | KNN(K=1) |
| Accuracy | 0.5611 | 0.625 | 0.625 | 0.5865 | 0.5 |

The logistic regression model and LDA model with Lag2 as its only predictor did the best. Their accuracies are 62.5%.

1. **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 classifier.**



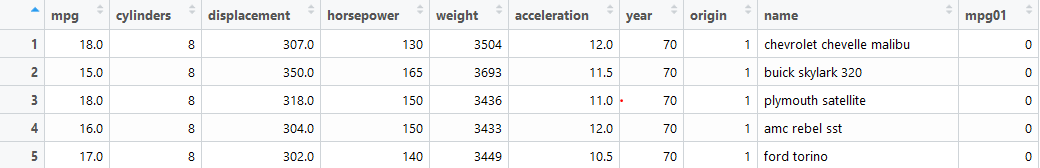
The KNN model when K is 10 has the highest accuracy (0.5962). Compared to the result when K is 1, its accuracy is much better. However, it is less than 0.625 which is the highest accuracy among all models.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| *method* | GLM1 | LDA | QDA | KNN(K=10) |
| Accuracy | 0.5769 | 0.625 | 0.5577 | 0.5962 |

The results above represent that the test accuracy is not good when the models have many variables. Only LDA model is the same as the result of the LDA model with Lag2 as only predictor. The test accuracy is better when the models have only Lag2 variable.

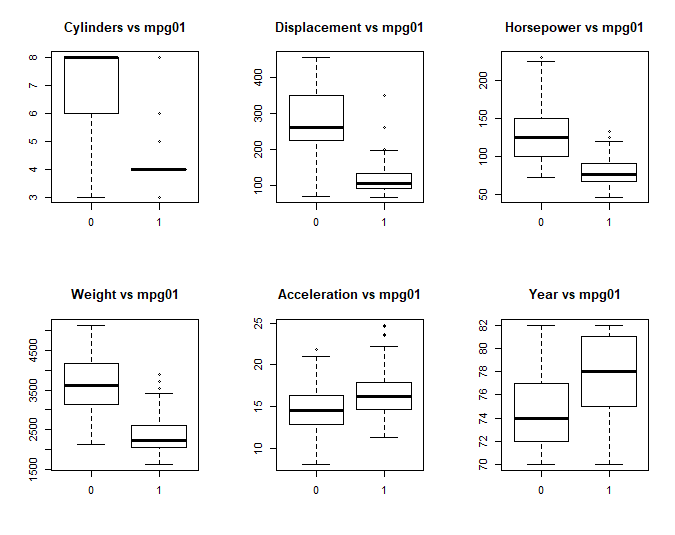
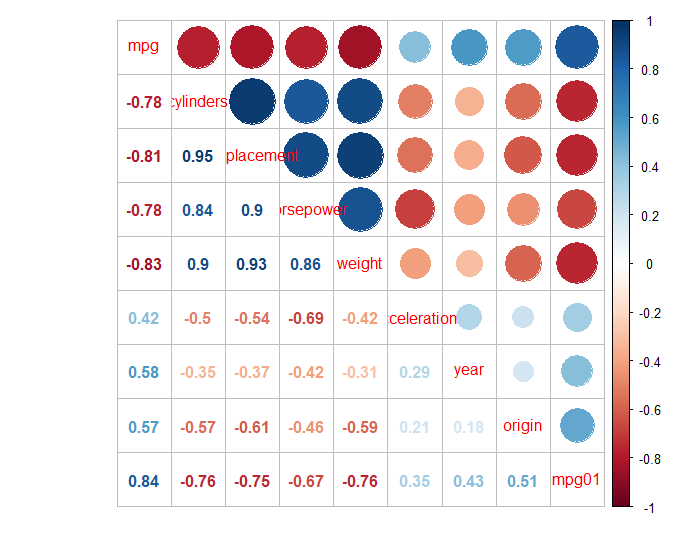
**11. In this problem, you will develop a model to predict whether a given car gets high or low gas mileage based on the Auto data set.**

1. **Create a binary variable, mpg01, that contains a 1 if mpg contains a value above its median, and a 0 if mpg contains a value below its median.**

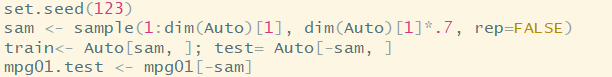


1. **Explore the data graphically in order to investigate the association between mpg01 and the other features. Which of the other features seem most likely to be useful in predicting mpg01? Scatterplots and boxplots may be useful tools to answer this question. Describe your findings.**

Correlation and scatter plot matrix below represent there exists some association between “mpg01”and “cylinders”, “weight”, “displacement” and “horsepower”.



1. **Split the data into a training set and a test set.**



A training set (70%) and a test set (30%).

|  |  |  |
| --- | --- | --- |
|  | 0 | 1 |
| 0 | 53 | 3 |
| 1 | 11 | 51 |

1. **Perform LDA on the training data in order to predict mpg01 using the variables that seemed most associated with mpg01 in (b). What is the test error of the model obtained?**

**Test error : 0.1186**

The test error of the LDA model is 0.1186 (11%).

|  |  |  |
| --- | --- | --- |
|  | 0 | 1 |
| 0 | 56 | 3 |
| 1 | 8 | 51 |

1. **Perform QDA on the training data in order to predict mpg01 using the variables that seemed most associated with mpg01 in (b). What is the test error of the model obtained?**

**Test error : 0.0932**

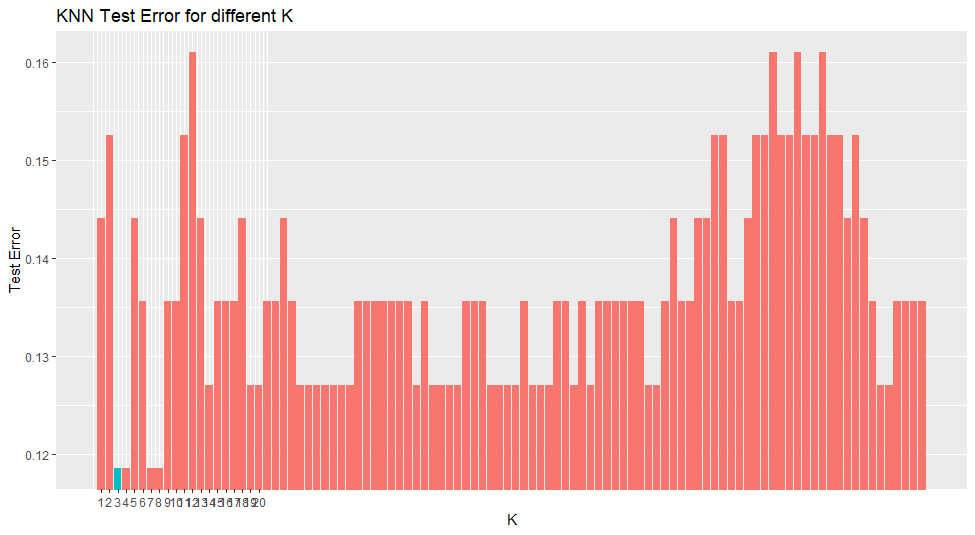
The test error of the QDA model is 0.0932 (9%).

|  |  |  |
| --- | --- | --- |
|  | 0 | 1 |
| 0 | 54 | 3 |
| 1 | 10 | 51 |

1. **Perform logistic regression on the training data in order to predict mpg01 using the variables that seemed most associated with mpg01 in (b). What is the test error of the model obtained?**

**Test error : 0.1102**

The test error of the logistic regression model is 0.1102 (11%).

1. **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 mpg01 in (b). What test errors do you obtain? Which value of K seems to perform the best on this data set?**

The test error of the KNN model when K =3 has the lowest test error. The histogram above represents that the test error is low when K =3, 4, 7, 8. When K=3, 4, 7, 8, the test error is 0.1186 (11%). Therefore, the best model is the KNN model with K=3.

river?

Discussion

After reviewing the datasets and using graphical analysis, we found many relationships between the variables and model accuracies. In the Weekly dataset, the LDA and the logistic regression model with only one variable have the highest accuracies. I think this is because there is only one variable “Lag2” and the simplest model is the best. In the Auto dataset, by using four model options, we can find that the QDA model is the best model for predicting mpg01. Also, by comparing the KNN models, we can conclude that model with few K is better. When K is fewer than 4, the model is better.

By investigating the datasets, we can find many interesting things. Especially, by creating correlation matrix and graphs, we can understand datasets better. Also, by splitting training set and test dataset, we can predict the variable with many variables and compare the models.

Appendix (R

**R codes**

**#ch4------------------------------------------------------------------------**

**##lab----------------------------------------------------------------------**

**# The Stock Market Data**

**library(ISLR);names(Smarket);dim(Smarket)**

**summary(Smarket);pairs(Smarket);cor(Smarket);cor(Smarket[,-9]);**

**attach(Smarket);plot(Volume)**

**# Logistic Regression**

**glm.fits=glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume,data=Smarket,family=binomial)**

**summary(glm.fits);coef(glm.fits);summary(glm.fits)$coef;summary(glm.fits)$coef[,4]**

**glm.probs=predict(glm.fits,type="response");glm.probs[1:10]**

**contrasts(Direction)**

**glm.pred=rep("Down",1250);glm.pred[glm.probs>.5]="Up"**

**table(glm.pred,Direction);(507+145)/1250;mean(glm.pred==Direction);train=(Year<2005)**

**Smarket.2005=Smarket[!train,];dim(Smarket.2005);Direction.2005=Direction[!train]**

**glm.fits=glm(Direction~Lag1+Lag2+Lag3+Lag4+Lag5+Volume,data=Smarket,family=binomial,subset=train)**

**glm.probs=predict(glm.fits,Smarket.2005,type="response");glm.pred=rep("Down",252);glm.pred[glm.probs>.5]="Up"**

**table(glm.pred,Direction.2005);mean(glm.pred==Direction.2005);mean(glm.pred!=Direction.2005) #52% correct rate**

**glm.fits=glm(Direction~Lag1+Lag2,data=Smarket,family=binomial,subset=train)**

**glm.probs=predict(glm.fits,Smarket.2005,type="response");glm.pred=rep("Down",252)**

**glm.pred[glm.probs>.5]="Up";table(glm.pred,Direction.2005);mean(glm.pred==Direction.2005)**

**106/(106+76) #58% correct rate**

**predict(glm.fits,newdata=data.frame(Lag1=c(1.2,1.5),Lag2=c(1.1,-0.8)),type="response")**

**# Linear Discriminant Analysis**

**library(MASS)**

**lda.fit=lda(Direction~Lag1+Lag2,data=Smarket,subset=train);lda.fit;plot(lda.fit)**

**lda.pred=predict(lda.fit, Smarket.2005);names(lda.pred);lda.class=lda.pred$class**

**table(lda.class,Direction.2005);mean(lda.class==Direction.2005) #56% correct rate**

**sum(lda.pred$posterior[,1]>=.5);sum(lda.pred$posterior[,1]<.5)**

**lda.pred$posterior[1:20,1];lda.class[1:20];sum(lda.pred$posterior[,1]>.9)**

**# Quadratic Discriminant Analysis**

**qda.fit=qda(Direction~Lag1+Lag2,data=Smarket,subset=train);qda.fit**

**qda.class=predict(qda.fit,Smarket.2005)$class;table(qda.class,Direction.2005)**

**mean(qda.class==Direction.2005)#60% correct rate**

**# K-Nearest Neighbors**

**library(class)**

**train.X=cbind(Lag1,Lag2)[train,];test.X=cbind(Lag1,Lag2)[!train,];train.Direction=Direction[train]**

**set.seed(1)**

**knn.pred=knn(train.X,test.X,train.Direction,k=1)#K=1**

**table(knn.pred,Direction.2005)**

**(83+43)/252 #50% correct rate**

**knn.pred=knn(train.X,test.X,train.Direction,k=3) #K=3**

**table(knn.pred,Direction.2005);mean(knn.pred==Direction.2005) #54% correct rate**

**# An Application to Caravan Insurance Data**

**dim(Caravan);attach(Caravan);summary(Purchase);348/5822**

**standardized.X=scale(Caravan[,-86])**

**var(Caravan[,1]);var(Caravan[,2]);var(standardized.X[,1]);var(standardized.X[,2])**

**test=1:1000;train.X=standardized.X[-test,];test.X=standardized.X[test,]**

**train.Y=Purchase[-test];test.Y=Purchase[test]**

**set.seed(1)**

**knn.pred=knn(train.X,test.X,train.Y,k=1);mean(test.Y!=knn.pred);mean(test.Y!="No")**

**table(knn.pred,test.Y);9/(68+9);knn.pred=knn(train.X,test.X,train.Y,k=3)**

**table(knn.pred,test.Y);5/26;knn.pred=knn(train.X,test.X,train.Y,k=5)**

**table(knn.pred,test.Y);4/15**

**glm.fits=glm(Purchase~.,data=Caravan,family=binomial,subset=-test)**

**glm.probs=predict(glm.fits,Caravan[test,],type="response")**

**glm.pred=rep("No",1000); glm.pred[glm.probs>.5]="Yes"**

**table(glm.pred,test.Y);glm.pred=rep("No",1000)**

**glm.pred[glm.probs>.25]="Yes";table(glm.pred,test.Y);11/(22+11)**

**##10----------------------------------------------------------------------**

**###a)------------------------------------------------------**

**require(ISLR); require(tidyverse) ;require(corrplot)**

**data('Weekly') ;glimpse(Weekly); summary(Weekly)**

**cor\_test <- cor.mtest(Weekly[,1:8], conf.level = .90)**

**corrplot(cor(Weekly[,1:8]), method = 'color',**

**order = 'hclust', addrect = 3,**

**p.mat = cor\_test$p, sig.level = 0.1, tl.col = 'black')**

**###b)------------------------------------------------------**

**glm1 <- glm(Direction ~ . - Year - Today, data = Weekly, family = 'binomial')**

**summary(glm1)**

**###c)------------------------------------------------------**

**attach(Weekly)**

**prd <- predict(glm1, type = "response");prd\_v <- ifelse(prd >= 0.5, 'Up', 'Down')**

**table(prd\_v, Direction);acc <- paste('Accuracy:', round(mean(pred.glm == Weekly$Direction),4));acc**

**###d)------------------------------------------------------**

**train <- Weekly[Weekly$Year <= 2008,]**

**test <- Weekly[Weekly$Year > 2008,]**

**glm2 <- glm(Direction ~ Lag2, data = train, family = 'binomial')**

**prd2 <- predict(glm2, newdata = test, type = 'response');prd\_v2 <- ifelse(prd2 >= 0.5, 'Up', 'Down');**

**table(prd\_v2, test$Direction);acc <- paste('Accuracy:', round(mean(prd\_v2 == test$Direction),4));acc**

**kable(table(pred\_values, test$Direction),**

**format = 'html') %>%**

**kable\_styling() %>%**

**add\_header\_above(c('Predicted' = 1, 'Observed' = 2)) %>%**

**column\_spec(1, bold = T) %>%**

**add\_footnote(label = acc)**

**###e)------------------------------------------------------**

**library(MASS)**

**mlda <- lda(Direction ~ Lag2, data = train)**

**prd3 <- predict(mlda, newdata = test);prd\_v3 <- prd3$class**

**table(prd\_v3, test$Direction);acc <- paste('Accuracy:', round(mean(prd\_v3 == test$Direction),4));acc**

**###f)------------------------------------------------------**

**mlda <- qda(Direction ~ Lag2, data = train)**

**prd3 <- predict(mlda, newdata = test);prd\_v3 <- prd3$class**

**table(prd\_v3, test$Direction);acc <- paste('Accuracy:', round(mean(prd\_v3 == test$Direction),4));acc**

**###g)------------------------------------------------------**

**library(class)**

**mknn <- knn(train = data.frame(train$Lag2), test = data.frame(test$Lag2),cl = train$Direction, k = 1)**

**table(mknn, test$Direction);acc <- paste('Accuracy:', round(mean(mknn == test$Direction),4));acc**

**###i)------------------------------------------------------**

**#knn==========================**

**acc <- list('1' = 0.5)**

**for (i in 1:20) {**

**mknn <- knn(train = data.frame(train$Lag2), test = data.frame(test$Lag2), cl = train$Direction, k = i)**

**acc[as.character(i)] = round(mean(mknn == test$Direction),4)**

**}**

**acc <- unlist(acc)**

**data\_frame(acc = acc) %>%**

**mutate(k = row\_number()) %>%**

**ggplot(aes(k, acc)) +**

**geom\_col(aes(fill = k == which.max(acc))) +**

**labs(x = 'K', y = 'Accuracy', title = 'KNN Accuracy for different K') +**

**scale\_x\_continuous(breaks = 1:20) +**

**coord\_cartesian(ylim = c(min(acc), max(acc))) +**

**guides(fill = FALSE)**

**#lda==============================**

**mlda2 <- lda(Direction ~ Lag2 + Lag4, data = train)**

**prd5 <- predict(mlda2, newdata = test); prd\_v5 <- prd5$class**

**acc <- paste('Accuracy:', round(mean(prd\_v5 == test$Direction),4));acc**

**#qda==============================**

**mqda2 <- qda(Direction ~ Lag2 + Lag1, data = train)**

**prd6 <- predict(mqda2, newdata = test);prd\_v6 <- prd6$class**

**acc <- paste('Accuracy:', round(mean(prd\_v6 == test$Direction),4));acc**

**#glm==============================**

**mglm3 <- glm(Direction ~ (. - Today - Volume)\*(. - Today - Volume), data = train, family = 'binomial')**

**step <- stepAIC(mglm3, direction = 'both', trace = 0)**

**prd7 <- predict(step, newdata = test, type = 'response');prd\_v7 <- ifelse(prd7 >= 0.5, 'Up', 'Down')**

**acc <- paste('Accuracy:', round(mean(prd\_v7 == test$Direction),4));acc**

**##11----------------------------------------------------------------------**

**###a)------------------------------------------------------**

**attach(Auto)**

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

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

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

**###b)------------------------------------------------------**

**cor(Auto[, -9])**

**library(corrplot) ; corrplot.mixed(cor(Auto[, -9]), upper="circle")**

**par(mfrow=c(2,3))**

**boxplot(cylinders ~ mpg01, data = Auto, main = "Cylinders vs mpg01")**

**boxplot(displacement ~ mpg01, data = Auto, main = "Displacement vs mpg01")**

**boxplot(horsepower ~ mpg01, data = Auto, main = "Horsepower vs mpg01")**

**boxplot(weight ~ mpg01, data = Auto, main = "Weight vs mpg01")**

**boxplot(acceleration ~ mpg01, data = Auto, main = "Acceleration vs mpg01")**

**boxplot(year ~ mpg01, data = Auto, main = "Year vs mpg01")**

**###c)------------------------------------------------------**

**set.seed(123)**

**sam <- sample(1:dim(Auto)[1], dim(Auto)[1]\*.7, rep=FALSE)**

**train<- Auto[sam, ]; test= Auto[-sam, ]**

**mpg01.test <- mpg01[-sam]**

**###d)------------------------------------------------------**

**mlda3 <- lda(mpg01 ~ cylinders + weight + displacement + horsepower, data = train)**

**prd.lda <- predict(mlda3, test)**

**table(prd.lda$class, mpg01.test);round(mean(prd.lda$class != mpg01.test),4)**

**###e)------------------------------------------------------**

**mqda3 = qda(mpg01 ~ cylinders + horsepower + weight + acceleration, data=train)**

**prd.qda <- predict(mqda3, test)**

**table(prd.qda$class, mpg01.test);round(mean(prd.qda$class != mpg01.test),4)**

**###f)------------------------------------------------------**

**mglm3 <- glm(mpg01 ~ cylinders + weight + displacement + horsepower, data = train, family = binomial)**

**summary(mglm3);prob <- predict(mglm3, test, type = "response")**

**prd.glm <- rep(0, length(prob));prd.glm[prob > 0.5] <- 1**

**table(prd.glm, mpg01.test) ; round(mean(prd.glm != mpg01.test),4)**

**###g)------------------------------------------------------**

**train.X <- cbind(cylinders, weight, displacement, horsepower)[sam, ]**

**test.X <- cbind(cylinders, weight, displacement, horsepower)[-sam, ]**

**train.mpg01 <- mpg01[sam]**

**set.seed(1)**

**err <- list('1' = 0.5)**

**for (i in 1:100) {**

**mknn <- knn(train = train.X ,test = test.X, cl = train.mpg01, k = i)**

**err[as.character(i)] = round(mean(mknn != mpg01.test),4)**

**}**

**err <- unlist(err)**

**data\_frame(err = err) %>%**

**mutate(k = row\_number()) %>%**

**ggplot(aes(k, err)) +**

**geom\_col(aes(fill = k == which.min(err))) +**

**labs(x = 'K', y = 'Test Error', title = 'KNN Test Error for different K') +**

**scale\_x\_continuous(breaks = 1:20) +**

**coord\_cartesian(ylim = c(min(err), max(err))) +**

**guides(fill = FALSE)**

**min(err)**