**Assignment 6**

**Performance classifiers of KNN**

**Name: Motbhare Subodh Uddhav**

**Roll No: 92**

**Branch & Div: Computer Engineering & C**

**Gr No: 12120132**

**Source Code**

library(ISLR)

library(caTools)

library(class)

?Smarket

df<-Smarket

View(df)

df = df[,2:9]

head(df)

str(df)

summary(df)

#scaling the input variables

df\_scaled = as.data.frame(scale(df[,1:7]))

df\_scaled$Direction = df$Direction

head(df\_scaled)

head(df)

set.seed(100)

split<-sample.split(df\_scaled$Direction, SplitRatio = 0.7)

table(split)

tr\_data<-subset(df\_scaled,split == T)

ts\_data<-subset(df\_scaled,split == F)

#KNN

pred<-knn(tr\_data[,1:7],ts\_data[,1:7],tr\_data$Direction, k = 3)

pred

head(pred)

ts\_data$pred = pred

View(ts\_data)

con\_matrix\_knn<-table(Actual = ts\_data$Direction,predicted = pred)

print(con\_matrix\_knn)

#performance of knn

acc\_knn<-sum(diag(con\_matrix\_knn))/sum(con\_matrix\_knn)

print(acc\_knn)

sens\_knn<-sum(con\_matrix\_knn[1,1])/sum(con\_matrix\_knn[1,])

print(sens\_knn)

specificity\_knn<-sum(con\_matrix\_knn[1,1])/sum(con\_matrix\_knn[,1])

print(specificity\_knn)

precision<-(con\_matrix\_knn[2,2])/sum(con\_matrix\_knn[,2])

print(precision)

cat("The accuracy of knn is : ",acc\_knn)

cat("The sensitivity of knn is : ",sens\_knn)

cat("The specitivity of knn is : ",specificity\_knn)

cat("The precision of knn is : ",precision)

**Output:**

> library(ISLR)

> library(caTools)

> library(class)

>

> ?Smarket

>

> df<-Smarket

> View(df)

> df = df[,2:9]

> head(df)

Lag1 Lag2 Lag3 Lag4 Lag5 Volume Today Direction

1 0.381 -0.192 -2.624 -1.055 5.010 1.1913 0.959 Up

2 0.959 0.381 -0.192 -2.624 -1.055 1.2965 1.032 Up

3 1.032 0.959 0.381 -0.192 -2.624 1.4112 -0.623 Down

4 -0.623 1.032 0.959 0.381 -0.192 1.2760 0.614 Up

5 0.614 -0.623 1.032 0.959 0.381 1.2057 0.213 Up

6 0.213 0.614 -0.623 1.032 0.959 1.3491 1.392 Up

> str(df)

'data.frame': 1250 obs. of 8 variables:

$ Lag1 : num 0.381 0.959 1.032 -0.623 0.614 ...

$ Lag2 : num -0.192 0.381 0.959 1.032 -0.623 ...

$ Lag3 : num -2.624 -0.192 0.381 0.959 1.032 ...

$ Lag4 : num -1.055 -2.624 -0.192 0.381 0.959 ...

$ Lag5 : num 5.01 -1.055 -2.624 -0.192 0.381 ...

$ Volume : num 1.19 1.3 1.41 1.28 1.21 ...

$ Today : num 0.959 1.032 -0.623 0.614 0.213 ...

$ Direction: Factor w/ 2 levels "Down","Up": 2 2 1 2 2 2 1 2 2 2 ...

> summary(df)

Lag1 Lag2 Lag3 Lag4

Min. :-4.922000 Min. :-4.922000 Min. :-4.922000 Min. :-4.922000

1st Qu.:-0.639500 1st Qu.:-0.639500 1st Qu.:-0.640000 1st Qu.:-0.640000

Median : 0.039000 Median : 0.039000 Median : 0.038500 Median : 0.038500

Mean : 0.003834 Mean : 0.003919 Mean : 0.001716 Mean : 0.001636

3rd Qu.: 0.596750 3rd Qu.: 0.596750 3rd Qu.: 0.596750 3rd Qu.: 0.596750

Max. : 5.733000 Max. : 5.733000 Max. : 5.733000 Max. : 5.733000

Lag5 Volume Today Direction

Min. :-4.92200 Min. :0.3561 Min. :-4.922000 Down:602

1st Qu.:-0.64000 1st Qu.:1.2574 1st Qu.:-0.639500 Up :648

Median : 0.03850 Median :1.4229 Median : 0.038500

Mean : 0.00561 Mean :1.4783 Mean : 0.003138

3rd Qu.: 0.59700 3rd Qu.:1.6417 3rd Qu.: 0.596750

Max. : 5.73300 Max. :3.1525 Max. : 5.733000

>

> #scaling the input variables

> df\_scaled = as.data.frame(scale(df[,1:7]))

> df\_scaled$Direction = df$Direction

> head(df\_scaled)

Lag1 Lag2 Lag3 Lag4 Lag5 Volume Today

1 0.3319247 -0.1724215 -2.3058832 -0.9278718 4.3609337 -0.7964461 0.8411801

2 0.8405937 0.3318555 -0.1701199 -2.3056696 -0.9242381 -0.5045135 0.9054218

3 0.9048373 0.8405328 0.3330842 -0.1700390 -2.2914985 -0.1862181 -0.5510161

4 -0.5516457 0.9047775 0.8406793 0.3331338 -0.1722013 -0.5614015 0.5375722

5 0.5369763 -0.5517294 0.9047873 0.8406973 0.3271233 -0.7564857 0.1846830

6 0.1840762 0.5369105 -0.5486207 0.9048013 0.8308050 -0.3585471 1.2222300

Direction

1 Up

2 Up

3 Down

4 Up

5 Up

6 Up

> head(df)

Lag1 Lag2 Lag3 Lag4 Lag5 Volume Today Direction

1 0.381 -0.192 -2.624 -1.055 5.010 1.1913 0.959 Up

2 0.959 0.381 -0.192 -2.624 -1.055 1.2965 1.032 Up

3 1.032 0.959 0.381 -0.192 -2.624 1.4112 -0.623 Down

4 -0.623 1.032 0.959 0.381 -0.192 1.2760 0.614 Up

5 0.614 -0.623 1.032 0.959 0.381 1.2057 0.213 Up

6 0.213 0.614 -0.623 1.032 0.959 1.3491 1.392 Up

>

> set.seed(100)

> split<-sample.split(df\_scaled$Direction, SplitRatio = 0.7)

> table(split)

split

FALSE TRUE

375 875

> tr\_data<-subset(df\_scaled,split == T)

> ts\_data<-subset(df\_scaled,split == F)

>

> #KNN

> pred<-knn(tr\_data[,1:7],ts\_data[,1:7],tr\_data$Direction, k = 3)

> pred

[1] Up Down Down Up Up Down Down Down Up Up Down Up Up Up Up Down

[17] Up Up Down Down Up Down Down Up Down Up Up Up Up Down Down Up

[33] Down Down Up Down Down Down Up Down Up Down Down Up Up Down Up Down

[49] Up Down Down Up Down Up Up Down Up Up Down Up Up Down Down Up

[65] Up Up Up Up Up Up Up Down Down Down Down Up Up Down Up Down

[81] Up Up Up Up Up Up Up Down Down Down Up Down Up Down Down Down

[97] Down Up Up Down Down Up Down Up Up Up Up Down Down Down Down Up

[113] Up Up Up Down Down Down Down Up Up Up Up Down Down Down Down Down

[129] Down Up Down Up Down Up Up Down Down Down Down Down Up Down Down Up

[145] Down Down Down Up Up Down Down Down Down Up Down Up Up Down Down Down

[161] Down Down Up Up Up Down Down Down Down Up Down Down Up Up Up Down

[177] Down Down Down Up Down Up Down Down Up Down Down Down Up Down Down Down

[193] Up Down Down Up Down Down Down Down Up Up Up Down Up Up Up Down

[209] Up Up Down Down Up Up Up Down Down Up Down Up Down Up Down Up

[225] Up Down Up Down Down Down Down Up Down Up Down Down Down Down Up Up

[241] Down Down Down Up Up Up Up Down Down Up Up Up Up Up Down Up

[257] Up Down Up Down Up Up Up Down Down Up Down Up Up Down Up Up

[273] Up Down Up Up Up Up Up Down Down Down Down Down Up Up Down Up

[289] Up Up Up Up Up Down Down Down Up Up Up Up Up Up Down Down

[305] Up Down Down Up Up Up Up Down Up Up Down Down Down Down Up Up

[321] Down Up Up Up Down Up Down Up Down Up Up Up Down Up Up Up

[337] Up Up Up Up Down Down Up Down Down Up Down Up Down Down Up Down

[353] Up Up Up Up Down Up Up Up Up Down Down Up Up Up Up Up

[369] Up Up Down Up Up Up Up

Levels: Down Up

> head(pred)

[1] Up Down Down Up Up Down

Levels: Down Up

> ts\_data$pred = pred

> View(ts\_data)

>

> con\_matrix\_knn<-table(Actual = ts\_data$Direction,predicted = pred)

> print(con\_matrix\_knn)

predicted

Actual Down Up

Down 148 33

Up 28 166

>

> #performance of knn

> acc\_knn<-sum(diag(con\_matrix\_knn))/sum(con\_matrix\_knn)

> print(acc\_knn)

[1] 0.8373333

> sens\_knn<-sum(con\_matrix\_knn[1,1])/sum(con\_matrix\_knn[1,])

> print(sens\_knn)

[1] 0.8176796

>

> specificity\_knn<-sum(con\_matrix\_knn[1,1])/sum(con\_matrix\_knn[,1])

> print(specificity\_knn)

[1] 0.8409091

>

> precision<-(con\_matrix\_knn[2,2])/sum(con\_matrix\_knn[,2])

> print(precision)

[1] 0.8341709

>

> cat("The accuracy of knn is : ",acc\_knn)

The accuracy of knn is : 0.8373333> cat("The sensitivity of knn is : ",sens\_knn)

The sensitivity of knn is : 0.8176796> cat("The specitivity of knn is : ",specificity\_knn)

The specitivity of knn is : 0.8409091> cat("The precision of knn is : ",precision)