Final Exam (Take Home)

Machine Learning in Economics

Prof. Dr. Hüseyin Taştan

Due: 12 June, 2022 (Sunday)

Contents

NAME & SURNAME:.................................. NO:............................

There are 5 questions in this exam, answer all of them. You should work on your own (not in teams). Arrange your answers in an `.Rmd` file (you can use this .Rmd file as a template) and produce a html file containing your answers (your .Rmd must knit into html without any errors). If you have any problems send me an email at huseyin.tastan@gmail.com

---

Question 1 (20 points)

(a)

Consider the following code chunk:

```r
library(tidyverse)
mpg <- mpg %>% mutate(cyl = factor(cyl))
mpg %>% group_by(cyl) %>%
  summarize(mean_cty = mean(cty),
            mean_hwy = mean(hwy)
  )
```

```
## # A tibble: 4 x 3
##   cyl   mean_cty mean_hwy
##   <fct>    <dbl>    <dbl>
## 1 4         21.0     28.8
## 2 5         20.5     28.8
## 3 6         16.2     22.8
## 4 8         12.6     17.6
```

Read the help file of the data set using `?mpg` and explain the table above.

Solution

Mpg is a dataset containing fuel economy data. cyl is the number of cylinders. The table above contains the average miles per gallon in the city in on the highway. On average, a model with 4-cylinders engine can see the highest miles per gallon(approximately hyw-29. cty-21) both on highway and in the city. The fuel consumprion increases as the number of cylinders increases, where an 8-cylinders engine has the least average miles per gallon on the highway and in the city. The less the cylinders the more fuel economy you will get.

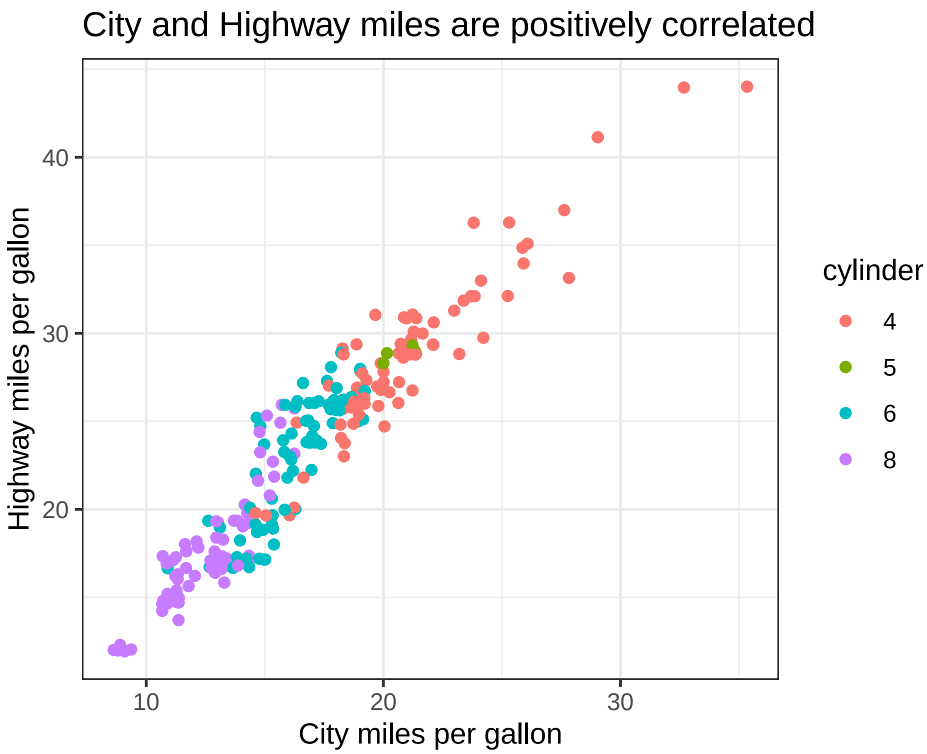
(b)

Using `ggplot2` reproduce the following graph exactly:

Then, put the following title in the plot: "City and Highway miles are positively correlated". Change x axis label to "City miles per gallon" and y axis label to "Highway miles per gallon". Also change the label title to "cylinders" instead of "cyl". (Hint: `geom_jitter()` can be useful).

Solution

```r
library(ggplot2)
ggplot(mpg, aes(x=cty, y=hwy, colour = cyl))+
  geom_jitter()+
  scale_x_continuous(breaks = seq(0,30,10))+
  scale_y_continuous(breaks = seq(0,40,10))+
  labs(title = "City and Highway miles are positively correlated",
       x = "City miles per gallon",
       y = "Highway miles per gallon",
       colour = "cylinder")+
  theme_bw()
```

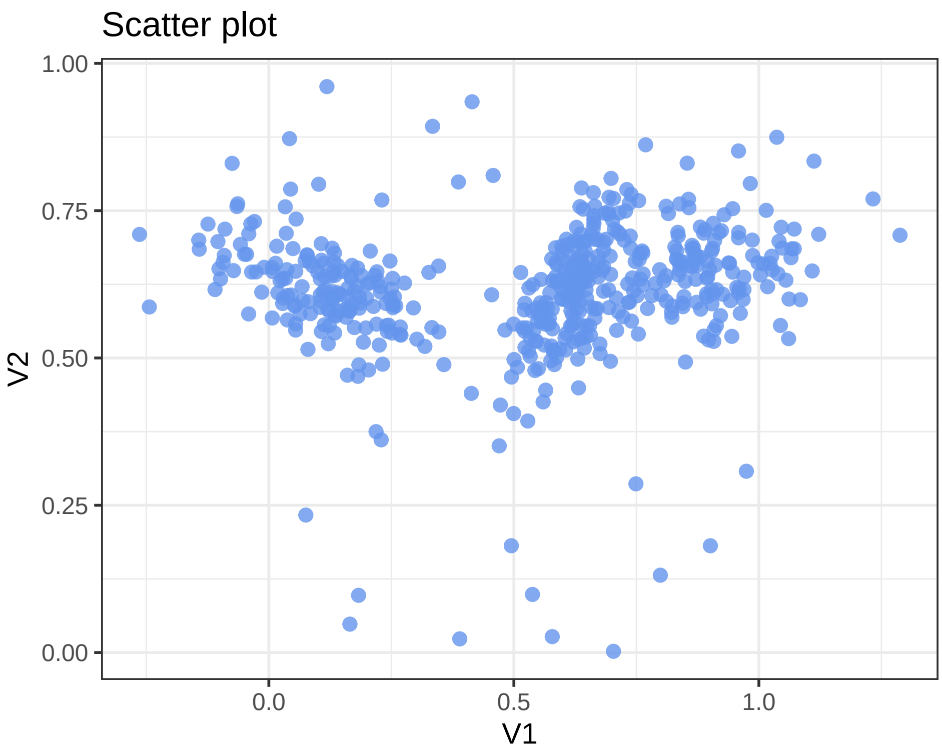


Question 2 (20 points)

The data set `fdata1.RData` contains a tibble named `fdata1` which has 510 observations on two variables `V1` and `V2`. First load the data set and then plot the scatter diagram using `ggplot2` library. We want to cluster observations into `k` groups using KNN. Consider k=2,3,4 clusters and run KNN algorithm for each k. Which one returns the lowest total within sum of squares? How many clusters are there?

Solution

```
load("fdata1.RData")
ggplot(fdata1, aes(V1, V2))+
  geom_point(color="cornflowerblue",
             size = 2,
             alpha=.8)+
  labs(title = "Scatter plot")+
  theme_bw()
```




```r
# Determine the # of clusters
wss <- (nrow(fdata1)-1)*sum(apply(fdata1,2,var))
for (i in 2:4) wss[i] <- sum(kmeans(fdata1,
   centers=i)$withinss)
plot(1:4, wss, type="b", xlab="Number of Clusters",
  ylab="Within groups sum of squares")
```


```r
# Total within sum of squares for k=2,3,4
set.seed(123)
library(cluster)
library(gridExtra)
```

```
##
## Attaching package: 'gridExtra'
## The following object is masked from 'package:dplyr':
##
##     combine
```

```
library(factoextra)
```

```
## Welcome! Want to learn more? See two factoextra-related books at https://goo.gl/ve3WBa
```

```
k2=kmeans(fdata1,2)
k3=kmeans(fdata1,3)
k4=kmeans(fdata1,4)

print(paste0("k2 total within sum of squares:", round(k2$tot.withinss,2)))
```

```
## [1] "k2 total within sum of squares:18.26"
```

```
print(paste0("k3 total within sum of squares:", round(k3$tot.withinss,2)))
```

```
## [1] "k3 total within sum of squares:11.06"
```

```
print(paste0("k4 total within sum of squares:", round(k4$tot.withinss,2)))
```

```
## [1] "k4 total within sum of squares:8.87"
```

```
print("k4 clusters size:")
```

```
## [1] "k4 clusters size:"
```

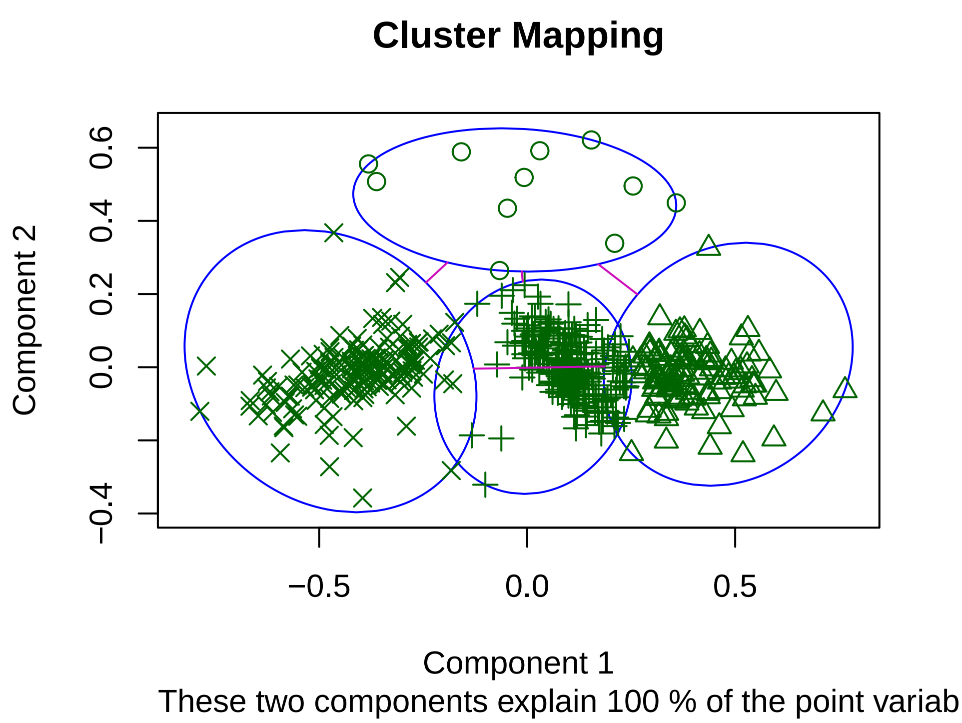
```
k4$size
```

```
## [1]  11 121 218 170
```

k = 4 returns the least total within sum of squares.

There are 4 clusters. The biggest cluster contains 218 observations.

```
par(mfrow = c(1,1))
clusplot(fdata1,k4$cluster, col.clus="blue", main="Cluster Mapping",cex=1.2)
```



Question 3 (20 points)

Use the data set `fdata1` from the previous part. This time apply hierarchical clustering method to assign observations into clusters. Use the following linkage functions: `complete`, `average`, `ward.D2`. Draw the dendrogram and interpret. How many clusters are there?

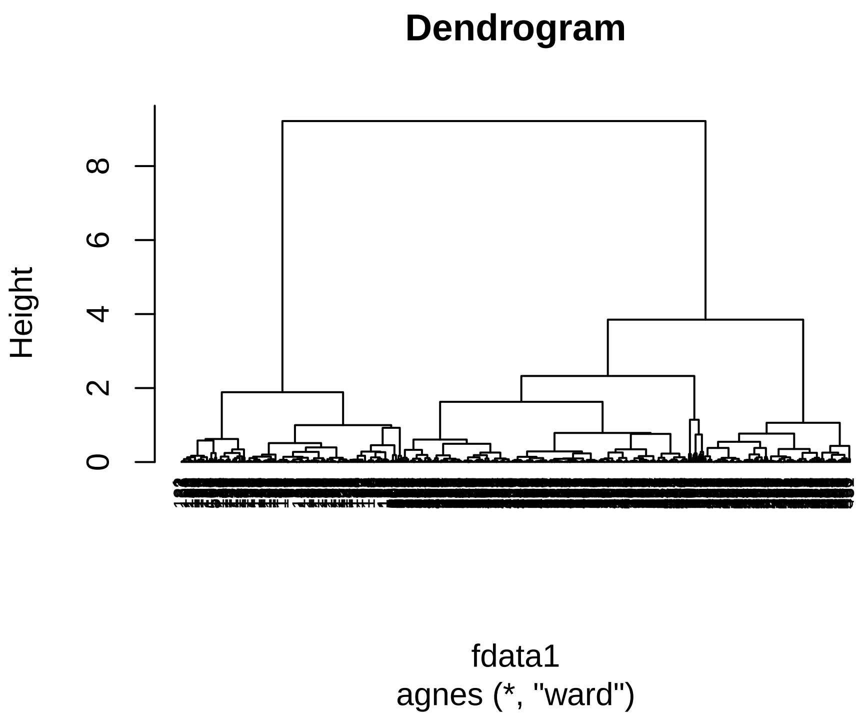
Solution

```
# linkage methods
lm <- c( "average", "complete", "ward")
names(lm) <- c( "average", "complete", "ward.D2")

# agglomerative coefficient function
ac <- function(x) {
  agnes(fdata1, method = x)$ac
}

# Find the coefficient for each linkage
sapply(lm, ac)
```

```
##   average  complete   ward.D2
## 0.9668377 0.9863157 0.9977167
```

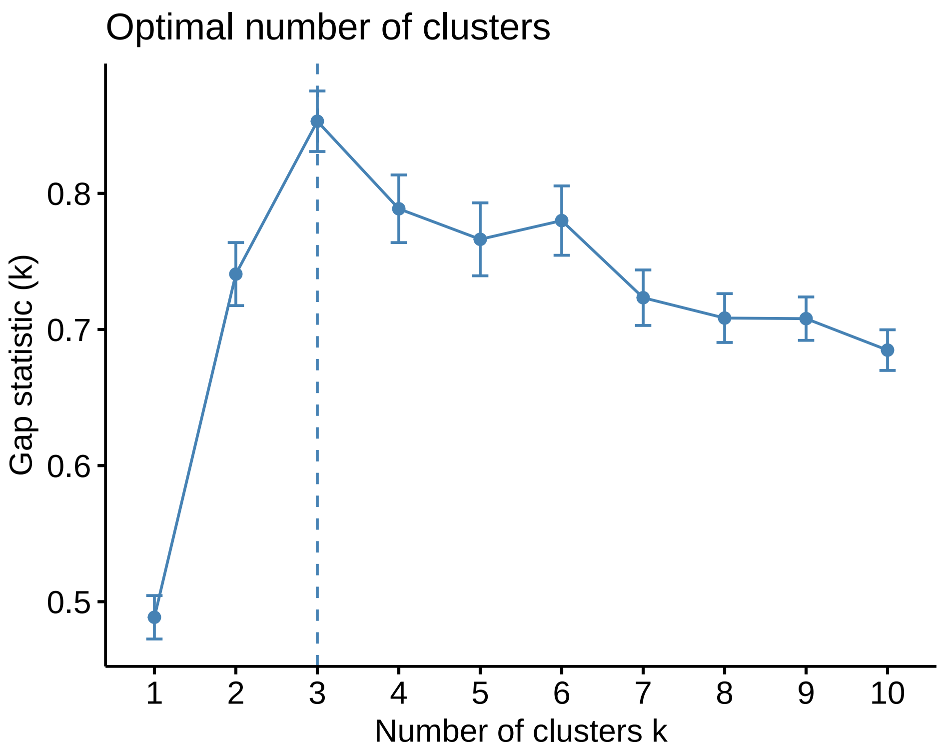
Ward.D2 linkage generates the biggest coefficient, which we'll apply in the hierarchical clustering.

```
# using Ward's minimum variance to perform hierarchical clustering
clust <- agnes(fdata1, method = "ward")

# producing a dendrogram
pltree(clust, hang = -1, cex = 0.6, main = "Dendrogram")
```



From the dendogram, the values at the bottom represents eaach observation in the data. As we climb up the tree, those observations with similarities are merged into one cluster/branch. The y-axis contains the height of dendrogram where we get the number of clusters. Fron the chart, we can see the clusters lies between 1 and 4, where we have 3 clusters.

```
#calculating gap statistics for clusters up to 10
gap_statistics <- clusGap(fdata1, FUN = hcut, K.max = 10, nstart = 25, B = 50)

# plot these clusters agaist their gap statistics
fviz_gap_stat(gap_statistics)
```



From the plot, k = 3 clusters produces the largest gap statistic hence our data is grouped into 3 clusters.

```
# Dissimilarity matrix
dis <- dist(fdata1, method = "euclidean")
# Ward.D2 method
hcl <- hclust(dis, method = "ward.D2" )

# Cut tree into 3 groups
grps <- cutree(hcl, k = 3)

# Number of members in each cluster
table(grps)
```

```
## grps
##   1   2   3
## 172 234 114
```

The first cluster contains 172 observations, the second clusters 234 while in the third cluster we have 114 observations.

Question 4 (20 points)

In this question, we are interested in predicting the direction in the foreign exchange market. To this end, you need to train a model to classify the movements in the USD/TL exchange rate. In several respects, this is similar to the `Smarket` example we saw in chapter 4 (also the exercise 10 in ch.4 that uses `Weekly` data set)

More specifically, we wish to predict the direction (Up or Down) in the USD/TL exchange rate using the last 5 days' exchange rate returns. The data set `finmarkets` contains the following variables:

```r
library(tidyverse)
library(caret)
```

```
## Loading required package: lattice

##
## Attaching package: 'caret'

## The following object is masked from 'package:purrr':
##
##     lift
```

```r
load("finmarkets.RData")
str(finmarkets)
```

```
## 'data.frame':    5086 obs. of  20 variables:
##  $ datechar     : chr  "2000-01-13" "2000-01-14" "2000-01-17" "2000-01-18" ...
##  $ year         : chr  "2000" "2000" "2000" "2000" ...
##  $ month        : chr  "01" "01" "01" "01" ...
##  $ day          : Ord.factor w/ 7 levels "Pazar"<"Pazartesi"<..: 5 6 2 3 4 6 2 3 4 5 ...
##  $ bist100      : num  18138 19110 18458 19577 19288 ...
##  $ usd          : num  0.539 0.54 0.543 0.546 0.546 ...
##  $ bistret      : num  6.87 5.22 -3.47 5.89 -1.49 ...
##  $ bistdirection: Factor w/ 2 levels "Down","Up": 2 2 1 2 1 1 1 2 2 2 ...
##  $ bistretlag1  : num  3.53 6.87 5.22 -3.47 5.89 ...
##  $ bistretlag2  : num  3.17 3.53 6.87 5.22 -3.47 ...
##  $ bistretlag3  : num  -2.27 3.17 3.53 6.87 5.22 ...
##  $ bistretlag4  : num  -4.42 -2.27 3.17 3.53 6.87 ...
##  $ bistretlag5  : num  -3.37 -4.42 -2.27 3.17 3.53 ...
##  $ usdret       : num  -0.0619 0.196 0.4419 0.6445 0.0304 ...
##  $ usddirection : Factor w/ 2 levels "Down","Up": 1 2 2 2 2 2 2 2 2 2 ...
##  $ usdretlag1   : num  0.8285 -0.0619 0.196 0.4419 0.6445 ...
##  $ usdretlag2   : num  0 0.8285 -0.0619 0.196 0.4419 ...
##  $ usdretlag3   : num  0.00299 0 0.82846 -0.06192 0.196 ...
##  $ usdretlag4   : num  -0.12587 0.00299 0 0.82846 -0.06192 ...
##  $ usdretlag5   : num  -0.92476 -0.12587 0.00299 0 0.82846 ...
##  - attr(*, "na.action")= 'omit' Named int [1:6] 1 2 3 4 5 6
##   ..- attr(*, "names")= chr [1:6] "1" "2" "3" "4" ...
```

```r
head(finmarkets)
```

```
##      datechar year month       day  bist100      usd   bistret bistdirection
## 7  2000-01-13 2000    01  Perşembe 18138.00 0.539275  6.868575            Up
## 8  2000-01-14 2000    01      Cuma 19110.00 0.540333  5.220257            Up
## 9  2000-01-17 2000    01 Pazartesi 18458.00 0.542726 -3.471388          Down
## 10 2000-01-18 2000    01      Salı 19577.00 0.546235  5.885753            Up
## 11 2000-01-19 2000    01  Çarşamba 19288.36 0.546401 -1.485360          Down
## 12 2000-01-21 2000    01      Cuma 17593.65 0.546873 -9.196376          Down
##    bistretlag1 bistretlag2 bistretlag3 bistretlag4 bistretlag5      usdret
## 7     3.527904    3.169542   -2.266227   -4.419408   -3.368104 -0.06191583
## 8     6.868575    3.527904    3.169542   -2.266227   -4.419408  0.19599713
## 9     5.220257    6.868575    3.527904    3.169542   -2.266227  0.44189724
## 10   -3.471388    5.220257    6.868575    3.527904    3.169542  0.64446976
## 11    5.885753   -3.471388    5.220257    6.868575    3.527904  0.03038523
## 12   -1.485360    5.885753   -3.471388    5.220257    6.868575  0.08634615
##    usddirection  usdretlag1  usdretlag2   usdretlag3   usdretlag4   usdretlag5
```

```
## 7          Down  0.82846414  0.00000000  0.002989822 -0.125868867 -0.924762862
## 8            Up -0.06191583  0.82846414  0.000000000  0.002989822 -0.125868867
## 9            Up  0.19599713 -0.06191583  0.828464143  0.000000000  0.002989822
## 10           Up  0.44189724  0.19599713 -0.061915834  0.828464143  0.000000000
## 11           Up  0.64446976  0.44189724  0.195997128 -0.061915834  0.828464143
## 12           Up  0.03038523  0.64446976  0.441897236  0.195997128 -0.061915834
```

The data set covers the period 13/01/2000 - 04/05/2020 and contains 5086 daily observations. `usdret` is today's return in the USD/TL exchange rate and it is defined as the daily percentage change in the USD/TL. `usddirection` is simply the sign of `usdret` and it has two levels: "Down" if `usdret` is negative, and "Up" if `usdret` is positive. The lagged usd returns are `usdretlag1-usdretlag5`. The data set also contains daily percentage returns on the BIST100 index, `bistret` and its lags `bistretlag1-bistretlag5`.

We first need to determine the train and test sets. For the purposes of this exercise, we will set the test set as years 2017-2018-2019-2020 (partly) and the previous years will be training set.

```r
finmarkets_train <- finmarkets %>% filter(year<=2016)
finmarkets_test <- finmarkets %>% filter(year>2016)
```

Note that the purpose is to successfully classify the direction in the USD/TL market.

(a)

Start by training a logistic regression model in which `usddirection` is the response variable and its lagged returns `usdretlag1` to `usdretlag5` (5 variables) as the predictor set. Do not use any other variable as a predictor. This is your first trained model.

Evaluate the performance of your model using test data only (`finmarkets_test`) and compute the confusion matrix (`caret` package can be useful). What is the accuracy rate and error rate in the test data? Is it any better than the no information rate?

Solution

```r
# Logistic Regression
model1 <- glm(usddirection ~ usdretlag1+usdretlag2+usdretlag3+usdretlag4+usdretlag5,
                   data = finmarkets_train,
                   family = binomial)
summary(model1)
```

```
##
## Call:
## glm(formula = usddirection ~ usdretlag1 + usdretlag2 + usdretlag3 +
##     usdretlag4 + usdretlag5, family = binomial, data = finmarkets_train)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.816  -1.163  -1.035   1.188   1.740
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.033688   0.030854  -1.092  0.27489
## usdretlag1   0.100343   0.033201   3.022  0.00251 **
## usdretlag2  -0.088136   0.031548  -2.794  0.00521 **
## usdretlag3   0.009199   0.031371   0.293  0.76934
## usdretlag4   0.014897   0.029842   0.499  0.61765
## usdretlag5  -0.007843   0.028918  -0.271  0.78624
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5894.8  on 4252  degrees of freedom
## Residual deviance: 5877.1  on 4247  degrees of freedom
## AIC: 5889.1
##
## Number of Fisher Scoring iterations: 3
```

The logistic regression summary shows that only the first 2 lags are significant, rest are insiginificant at 5% significance level.

```
# Predict test data based on model
model1_probabilities <- predict(model1,
                        finmarkets_test, type = "response")
model1.predictions <- ifelse(model1_probabilities > 0.5, "Up", "Down")

Direction = finmarkets_test$usddirection
# Confusion matrix
table(model1.predictions, Direction)
```

```
##                   Direction
## model1.predictions Down  Up
##               Down  270 264
##               Up    123 176
```

```
# Model accuracy
corrects <- mean(model1.predictions == Direction)
print(paste('Logistic Regression Accuracy =', round(corrects,4)))
```

```
## [1] "Logistic Regression Accuracy = 0.5354"
```

```
print(paste('Error Rate:', 1-round(corrects,4)))
```

```
## [1] "Error Rate: 0.4646"
```

The model model has made 446 correct predictions and 387 incorrect predictions in the test set. This gives an accuracy of about 54%, meaning that only 54% values has been bredicted correctly.

(b)

Now, augment your model by adding lagged returns of BIST100, that is, `bistretlag1-bistretlag5` (additional 5 variables). This is your second model. Also evaluate this model using test data and compare it to the previous model. Would you use these models in your daily exchange rate transactions and investments? In other words, is it possible to earn money using your preferred model?

Solution

```
# Logistic Regression

# Select the required columns
cols <- c("usddirection", "bistretlag1", "bistretlag2", "bistretlag3", "bistretlag4", "bistretlag5", "u

finmarkets <- finmarkets[, cols]
finmarkets_train <- finmarkets_train[, cols]
```

```r
finmarkets_test <- finmarkets_test[, cols]

# Fit the augmented logistic model
model2 <- glm(usddirection ~.,
                   data = finmarkets_train,
                   family = binomial)
summary(model2)
```

```
##
## Call:
## glm(formula = usddirection ~ ., family = binomial, data = finmarkets_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -3.4527  -1.0831  -0.5306   1.1139   2.7140
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.014355   0.032459  -0.442   0.6583
## bistretlag1 -0.290354   0.018121 -16.023   <2e-16 ***
## bistretlag2 -0.169397   0.017352  -9.762   <2e-16 ***
## bistretlag3 -0.018233   0.017319  -1.053   0.2924
## bistretlag4  0.006933   0.016664   0.416   0.6774
## bistretlag5  0.011960   0.016705   0.716   0.4740
## usdretlag1   0.004171   0.039230   0.106   0.9153
## usdretlag2  -0.082751   0.039796  -2.079   0.0376 *
## usdretlag3   0.008788   0.034903   0.252   0.8012
## usdretlag4   0.017071   0.034239   0.499   0.6181
## usdretlag5   0.027243   0.031539   0.864   0.3877
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 5894.8  on 4252  degrees of freedom
## Residual deviance: 5470.9  on 4242  degrees of freedom
## AIC: 5492.9
##
## Number of Fisher Scoring iterations: 4
```

After augmenting, this time it is only lag 1 and 2 of BST100 and lag 2 of usdret are significant. This model is slightly better as the AIC has reduced from 5889 to 5471.

```r
# Predict test data based on model
model2_probabilities <- predict(model2,
                       finmarkets_test, type = "response")
model2.predictions <- ifelse(model2_probabilities > 0.5, "Up", "Down")

Direction = finmarkets_test$usddirection
# Confusion matrix
table(Direction, model2.predictions)
```

```
##          model2.predictions
## Direction Down  Up
##      Down  267 126
```

```
##      Up    198 242
```

```
# Accuracy
classerr <- mean(model2.predictions == Direction)
print(paste('Augmented Model Accuracy =', round(classerr,4)))
```

```
## [1] "Augmented Model Accuracy = 0.611"
```

After evaluating the model, we ca see that the accuracy rate has increased to 61%, hence model2 is better. However, this is still not convincing as the error rate is still high at 39% which is not worth the risk of an investment.

Question 5 (20 points)

Continue using the data set from the previous question. This time,

(a)

Apply the bagging approach to estimate the classification tree. Evaluate the test performance as usual. Plot the variable importance graph and interpret.

Solution

```
library(rpart)
library(ipred)

set.seed(1)

#fit the bagging model
bag <- bagging(
  usddirection ~ .,
  data = finmarkets_train,
  method = "treebag",
  trControl = trainControl(method = "cv", number = 10),
  nbagg = 100,
  control = rpart.control(minsplit = 2, cp = 0)
)

#confusion matrix for bagged trees

preds <- predict(bag, finmarkets_test, type="class")
conf.Matrix <- table(Direction, preds)
conf.Matrix
```

```
##           preds
## Direction Down  Up
##      Down  255 138
##      Up    203 237
```

```
# Test performance
missed_classerr <- mean(preds != Direction)
print(paste('Bagging Accuracy =', round(1-missed_classerr,4)))
```

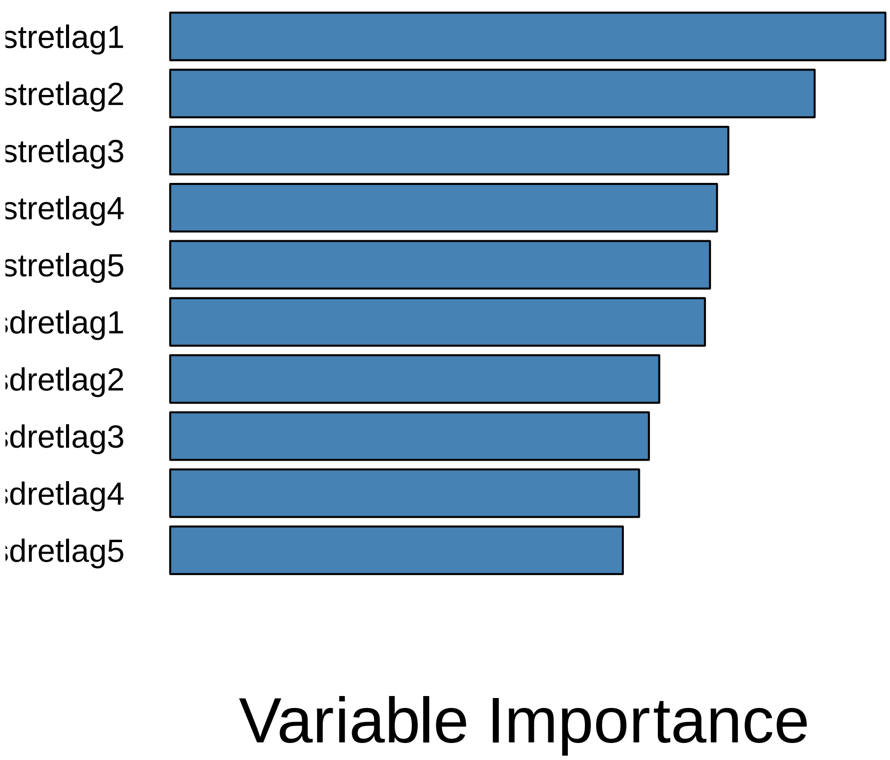
```
## [1] "Bagging Accuracy = 0.5906"
```

Bagging did not help increase accuacy. The model was only able to predict 0.59 of observations ins in the test data.

```
library(vip)
```

```
##
## Attaching package: 'vip'

## The following object is masked from 'package:utils':
##
##     vi
```

```
# variable importance
VI <- data.frame(variables = names(finmarkets[,-1]),
                 importance = varImp(bag))

# sort VI ascending
VI_plot <- VI[order(VI$Overall, decreasing=F),]

# plot the variable importance
barplot(VI_plot$Overall,
        names.arg=rownames(VI_plot),
        xlab='Variable Importance',
        horiz=TRUE,
        xaxt = "n",
        las = 2, cex.lab = 2, font.lab = 1,
        col='steelblue')
```



The figure above shows that bistrelag1 and bistretlag2 are the most important features to predict whether the USD/TL "Up" and "Down".

(b)

Apply the random forest approach to estimate the decision tree. Evaluate the test performance as usual. How did you choose the number of variables considered at each split (mtry)? Plot the variable importance graph and interpret.

Solution

```r
set.seed(123)
library(randomForest)
```

```
## randomForest 4.7-1.1

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:gridExtra':
##
##     combine

## The following object is masked from 'package:dplyr':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin
```

```r
Rf <- randomForest(
  usddirection ~ .,
  data = finmarkets_train,
  importance = TRUE,
  mtry=sqrt(ncol(finmarkets_train))-1)
importance(Rf)
```

```
##                   Down          Up MeanDecreaseAccuracy MeanDecreaseGini
## bistretlag1 40.0377925 37.10069759           53.0130770         352.0881
## bistretlag2 17.3916832 13.37551371           22.8800773         245.2242
## bistretlag3  1.2182091  0.25982190            1.1678477         191.4965
## bistretlag4  0.9226447 -0.39210843            0.4373649         191.5185
## bistretlag5 -0.3747037  3.82651897            2.5743252         193.4617
## usdretlag1  -3.0889227  5.99195223            2.0220531         195.0458
## usdretlag2   3.1954141 -3.88781583           -0.5555203         187.0555
## usdretlag3   0.5474710  4.75639537            3.9766798         193.6538
## usdretlag4  -1.5468457  2.97574514            1.1166944         188.5766
## usdretlag5   1.5768788 -0.09416964            1.0651685         187.3059
```

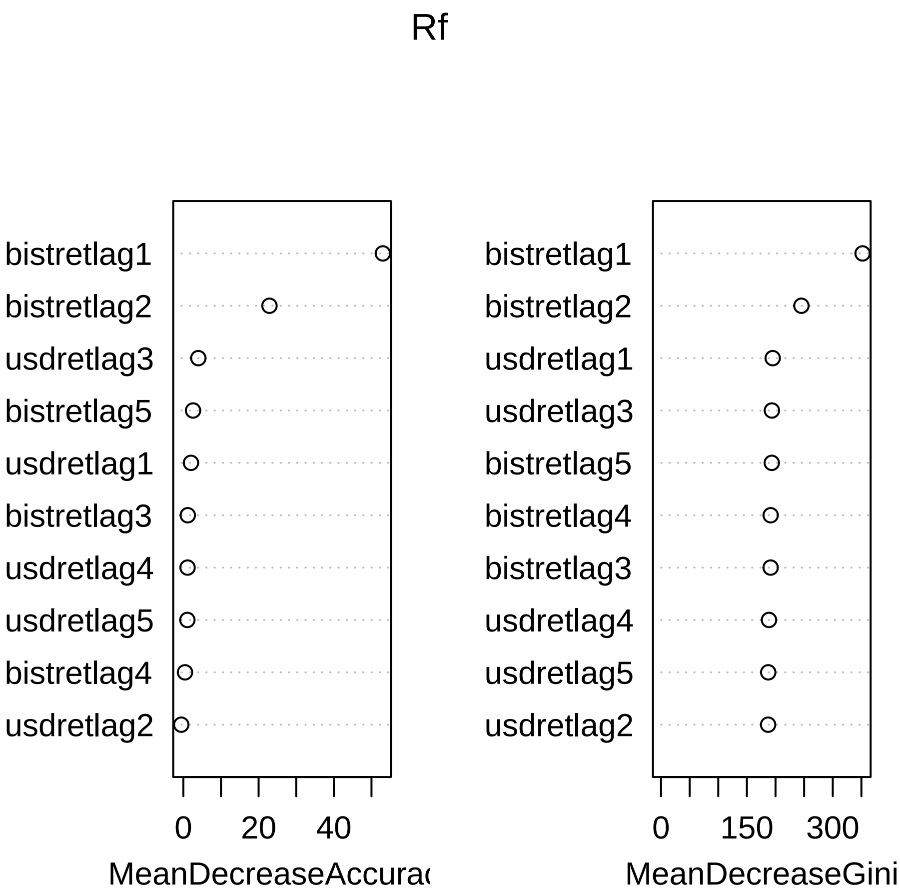
```r
#confusion matrix for RF tree
rf.preds <- predict(Rf, finmarkets_test, type="class")
table(Direction, rf.preds)
```

```
##          rf.preds
## Direction Down  Up
##      Down  257 136
##      Up    200 240
```

```r
# Test performance
missed <- mean(rf.preds != Direction)
print(paste('Accuracy =', round(1-missed,4)))
```

```
## [1] "Accuracy = 0.5966"
```

```r
# variable importance Plot
varImpPlot(Rf)
```



According to the value importance graph, bistretlag1 and bistretlag2 are the two most significant predictors having the largest mean decrease in accuracy rate as well as mean deacrese in Gini.