CME 4403 Hw3 - Due Date: November 4th 2019, 23:59 AKİF CAKAR

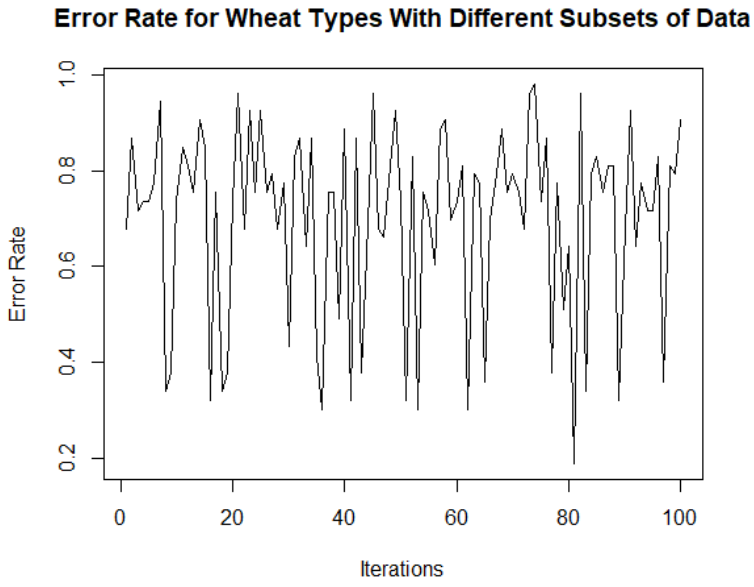
Please send your source codes and results (figures, replies etc.) in one text file (e.g., YourName.docx / YourName.pdf), then submit it to the Classroom.

Group homework is not allowed!

1. You will code a **decision tree** (DT) that will be similar with the one given in the lab examples. The input data is provided in the text file "wheat_types.txt". The target feature is the "type" column. You will train and test the DT by using all features except "type". Apply the following steps and reply the questions in your report.
    - You can use a different decision tree library apart from "tree".
    - Use 75% of samples for training and 25% of them for test purposes.
    - Perform this sample division (cross-validation) process for "100" iterations.
    - Draw a plot to show how accuracy changes over 100 iterations.
    - Report average accuracy of DT after 100 iterations.
    - Show the DT (with parent and internal node's decision criteria) which has the highest accuracy over 100 iterations.

ANSWERS FOR 1st –(The R Code is the bottom of the page)

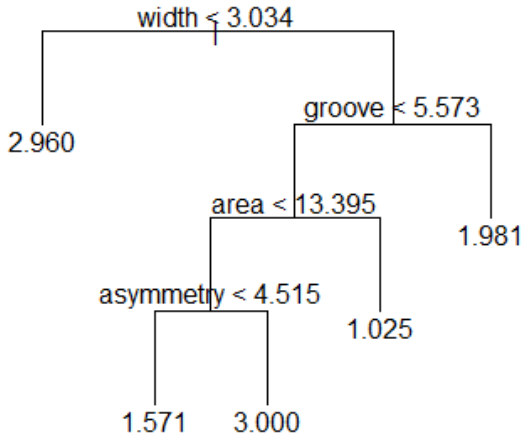
(Draw a plot to show how accuracy changes over 100 iterations.)



(Report average accuracy of DT after 100 iterations.)

```
> #Report average accuracy of DT after 100 iterations.
> mean(dt_acc)
[1] 0.2998113
```

(Show the DT which has the highest accuracy over 100 iterations.)



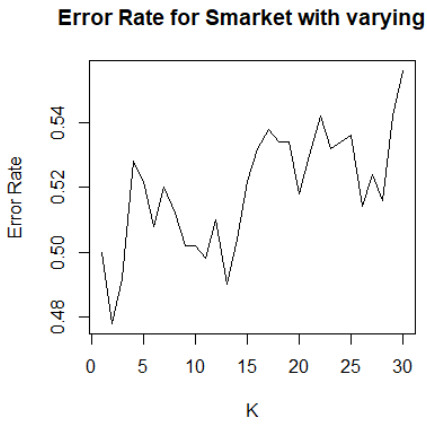
2. You will code a **k-nearest neighbor** (kNN) algorithm, which will be similar with the one given in the lab examples. You will use built-in data set "Smarket" that is provided in the "ISLR" library. The target feature is the "Direction" column. You will use various combinations of descriptive features and different ***k*** values. Apply the following steps and reply the questions in your report.
    - You need to test several combinations of descriptive features to improve the prediction accuracy of the target feature. The combinations:
        - Lag1, Lag2, Lag3
        - Lag3, Lag4, Volume
        - Lag2, Lag5, Volume
    - Use 60% of samples for training and 40% of them for test purposes (for each combination).
    - Test different ***k*** values (1 to 30) for each combination.
    - Make a plot to show how accuracy changes for different ***k*** values (for each combination).

- Which ***k*** value did provide the highest accuracy for each combination? Report both ***k*** values and the highest accuracy value.
- What is the highest accuracy to predict the "Direction" column when you set a specific ***k*** value and the descriptive feature combination?

ANSWERS FOR 2nd –(The R Code is the bottom of the page)

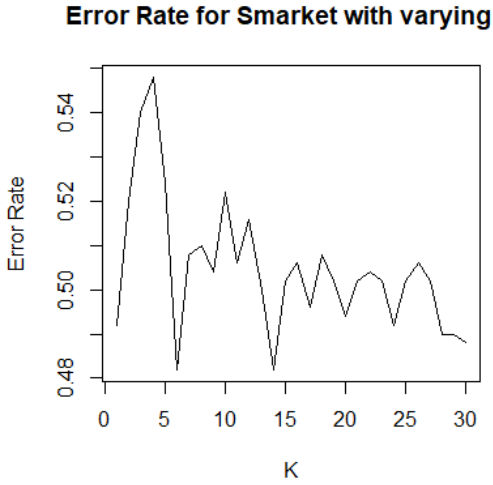
(plot to show how accuracy changes for different ***k*** values) (for each combination)

(Lag1, Lag2, Lag3)



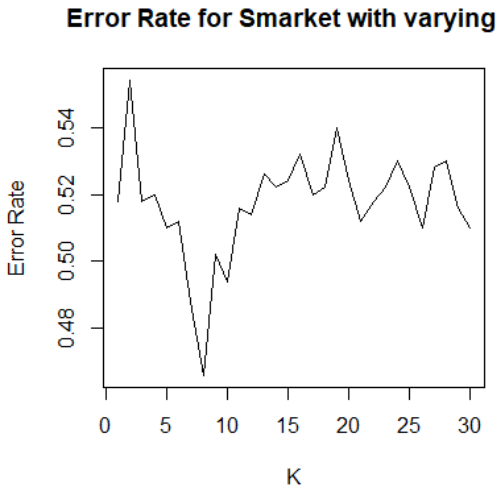

```
> #highest accuracy
> highest_accuracy1
[1] 0.522
>
> #k value has highest accuracy
> highest_k_accuracy1
[1] 2
```

(Lag3, Lag4, Volume)




```
> #highest accuracy
> highest_accuracy2
[1] 0.518
>
> #k value has highest accuracy
> highest_k_accuracy2
[1] 6
```

(Lag2, Lag5, Volume)




```
> #highest accuracy
> highest_accuracy3
[1] 0.534
>
> #k value has highest accuracy
> highest_k_accuracy3
[1] 8
```

(The highest accuracy to predict the "Direction" column when set a specific ***k*** value and the descriptive feature combination)

```
+ }
The highest accuracy(0.534000) provided by the combinations that are Lag2, Lag5 and Volume
>
```

R Codes For 1st Question

```r
setwd('C:\\Users\\akifc\\Desktop')
options(max.print=999999)

library(tree) # Contains the "tree" function

wheat_types <- read.table("wheat_types.txt", header = TRUE, sep = ";", dec = ".")


#Set the seed for reproducibility
# Use 75% of samples for traning, the rest for testing
# the indices (row ids) are saved in the "sub" vector
set.seed(579642)
sub <- sample(1:nrow(wheat_types), size=nrow(wheat_types)*0.75)

wt.tr <- tree(type ~ ., data = wheat_types, subset = sub)
```

```r
summary(wt.tr)

#################################
#Cross-validation version - Construct a new DT
#for different partitions of the samples - 100 times

dt_acc <- numeric()
set.seed(1815850)

highest_accuracy <- -99

for(i in 1:100){
  sub <-  sample(1:nrow(wheat_types), size=nrow(wheat_types)*0.75)
  fit2 <- tree(type ~ ., data = wheat_types, subset = sub)
  test_predict <- table(predict(fit2, wheat_types[-sub, ]), wheat_types[-sub, "type"])
  dt_acc <- c(dt_acc, sum(diag(test_predict)) / sum(test_predict))

  if(highest_accuracy < dt_acc[i]){ #DT which has the highest accuracy over 100 iterations.
    highest_accuracy <- dt_acc[i]
    highest_acc_tr <- fit2
  }
}

#Report average accuracy of DT after 100 iterations.
mean(dt_acc)


#Draw a plot to show how accuracy changes over 100 iterations.
plot(1-dt_acc, type="l", ylab="Error Rate", xlab="Iterations", main="Error Rate for Wheat Types With Different Subsets of Data")

#################################

#The value 'highest_acc_tr' filled above.
# Show the DT (with parent and internal node's decision criteria) which has the highest accuracy over 100 iterations.
plot(highest_acc_tr,  type = "uniform")
text(highest_acc_tr)
```

R Codes For 2nd Question

```r
install.packages("ISLR")
library(ISLR)

install.packages('class')
library(class) # Contains the "knn" function

###############################################
###############################################

#Create partitions in the Smarket data set (60% for training, 40% for testing/evaluation)
sm_sample <- sample(1:nrow(Smarket), size=nrow(Smarket)*0.6)
sm_train <- Smarket[sm_sample, ] #Select the 60% of rows
sm_test <- Smarket[-sm_sample, ] #Select the 40% of rows

```

```
##############################################
##############################################

#Here 2,3,4 in the vector represent Lag1, Lag2 and Lag3 features
train.X <- sm_train[,c(2,3,4)]
test.X <- sm_test[,c(2,3,4)]

#Seed must set in order to get reproducible result
set.seed(4985912356) #Set the seed for reproducibility

#First try to determine the right K-value
smarket_acc <- numeric() #holding variable


highest_accuracy1 <- -99
highest_k_accuracy1 <- -1


for(i in 1:30){
  predict <- knn(train=train.X, test=test.X, cl=sm_train$Direction, k=i)
  smarket_acc <- c(smarket_acc, mean(predict==sm_test$Direction))
  if(highest_accuracy1 < smarket_acc[i]){
#To find the value which has the highest accuracy.
    highest_accuracy1 <- smarket_acc[i]
    highest_k_accuracy1 <- i
  }
}

#Plot error rates for k=1 to 30
plot(1-
smarket_acc, type="l", ylab="Error Rate",  xlab="K", main="Error Rate for Smarket wi
th varying K")

#highest accuracy
highest_accuracy1

#k value has highest accuracy
highest_k_accuracy1

##############################################
##############################################

#Here 4,5,7 in the vector represent Lag3, Lag4 and Volume features
train.X <- sm_train[,c(4,5,7)]
test.X <- sm_test[,c(4,5,7)]

#Seed must set in order to get reproducible result

#First try to determine the right K-value
smarket_acc <- numeric() #holding variable


highest_accuracy2 <- -99
highest_k_accuracy2 <- -1


for(i in 1:30){
  predict <- knn(train=train.X, test=test.X, cl=sm_train$Direction, k=i)
  smarket_acc <- c(smarket_acc, mean(predict==sm_test$Direction))
  if(highest_accuracy2 < smarket_acc[i]){
#To find the value which has the highest accuracy.
    highest_accuracy2 <- smarket_acc[i]
    highest_k_accuracy2 <- i
  }
}

```

```r
#Plot error rates for k=1 to 30
plot(1-
smarket_acc, type="l", ylab="Error Rate",  xlab="K", main="Error Rate for Smarket wi
th varying K")

#highest accuracy
highest_accuracy2

#k value has highest accuracy
highest_k_accuracy2

##############################################
##############################################

#Here 3,6,7 in the vector represent Lag2, Lag5 and Volume features
train.X <- sm_train[,c(3,6,7)]
test.X <- sm_test[,c(3,6,7)]

#Seed must set in order to get reproducible result

#First try to determine the right K-value
smarket_acc <- numeric() #holding variable


      highest_accuracy3 <- -99
      highest_k_accuracy3 <- -1


      for(i in 1:30){
        predict <- knn(train=train.X, test=test.X, cl=sm_train$Direction, k=i)
        smarket_acc <- c(smarket_acc, mean(predict==sm_test$Direction))
        if(highest_accuracy3 < smarket_acc[i]){
      #To find the value which has the highest accuracy.
          highest_accuracy3 <- smarket_acc[i]
          highest_k_accuracy3 <- i
        }
      }

      #Plot error rates for k=1 to 30
      plot(1-
smarket_acc, type="l", ylab="Error Rate",  xlab="K", main="Error Rate for Smarket wi
th varying K")

      #highest accuracy
      highest_accuracy3

      #k value has highest accuracy
      highest_k_accuracy3

      ##############################################
      max_value <- max(highest_k_accuracy1,highest_accuracy2,highest_k_accuracy3)

      if(highest_k_accuracy1==max_value){
        cat(sprintf("The highest accuracy(%f) provided by the combinations that are
 Lag1, Lag2 and Lag3",highest_accuracy1))
      } else if(highest_k_accuracy2==max_value){
        cat(sprintf("The highest accuracy(%f) provided by the combinations that are
 Lag3, Lag4 and Volume",highest_accuracy2))
      } else {
        cat(sprintf("The highest accuracy(%f) provided by the combinations that are
 Lag2, Lag5 and Volume",highest_accuracy3))
      }
```
