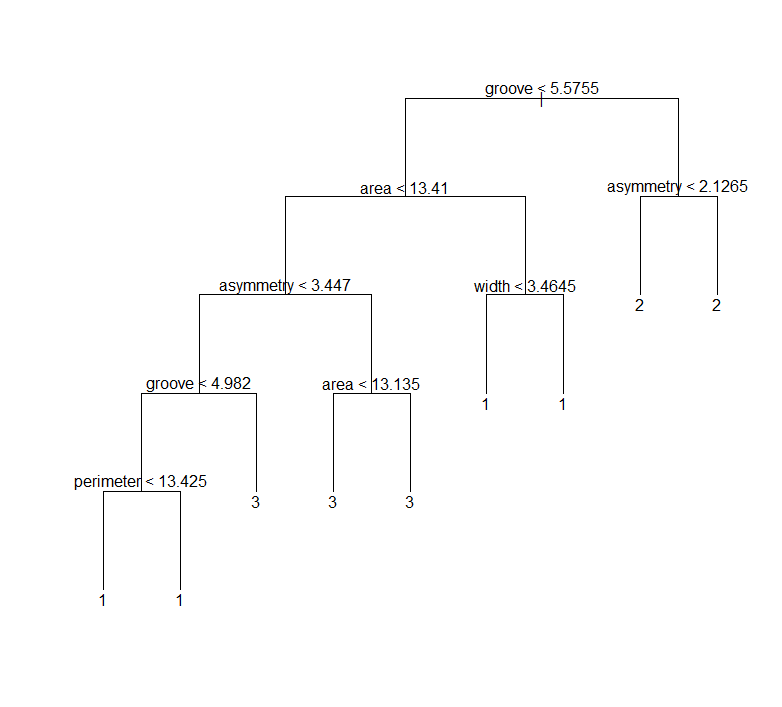
**CME4403 Introduction to Machine Learning – Homework 3 Report**

**First Part**

In the first part, the purpose is to build a Decision Tree that finds the type of the given wheat samples. “tree” library is used in this part to complete this part of the assignment. We have a wheat data set that consist of 7 descriptive and 1 target feature. Target feature is “type” and descriptive features are “area, perimeter, compactness, length, width, asymmetry, groove”. There are 3 types of wheat types. Also, this data set has 210 rows.

We split our dataset into train and test samples. %80 of samples is used for training purpose and the remaining of the samples is used for test purpose (168 samples for training phase and 42 samples for test phase). Firstly, our model needs to be trained and then, the test phase needs to be done.

When the training part finished with randomly selected samples, the Decision Tree constructed with 9 leaf nodes and 7 interior nodes. Best attribute to be chosen as top root in the decision tree has been selected as “groove” with threshold “groove < 5.5755”. To construct the tree, just "groove”, "area", "asymmetry", "perimeter" and "width" descriptive features have been used.



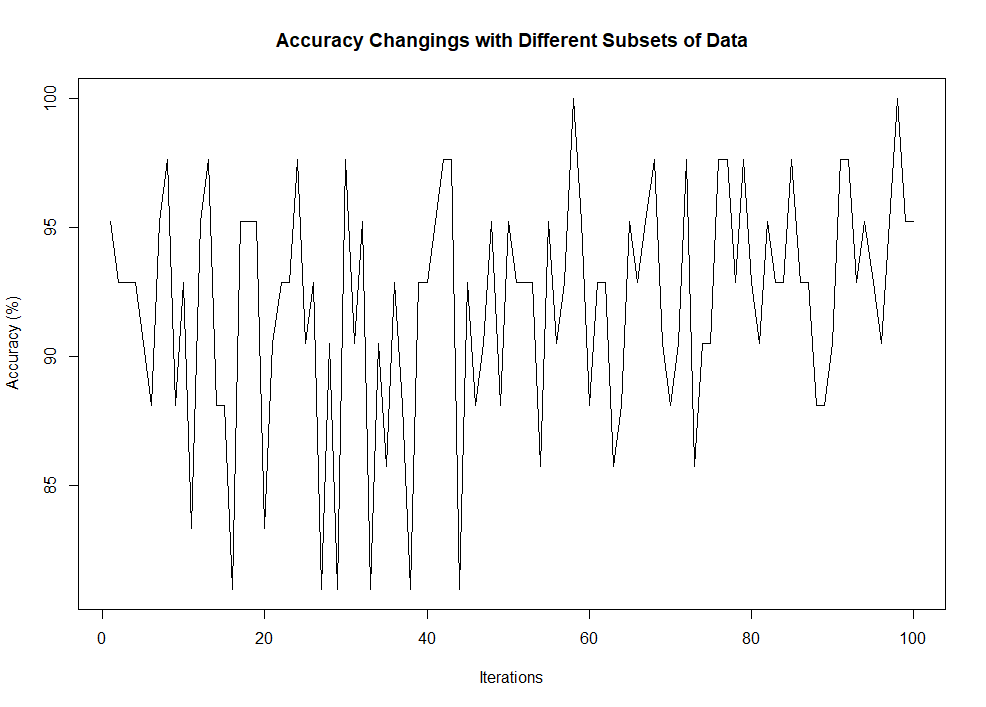
For training samples, the misclassification error rate: 0.04167 = 7 / 168. This means that types of 7 wheat samples could not be classified correctly in the decision tree. So, this is our train performance. We can observe how well our decision tree learned. The confusion matrix for training shown below.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Predicted** | | | |
| **Actual** | Type | 1 | 2 | 3 |
| 1 | **55** | **2** | **2** |
| 2 | **1** | **53** | **0** |
| 3 | **2** | **0** | **53** |

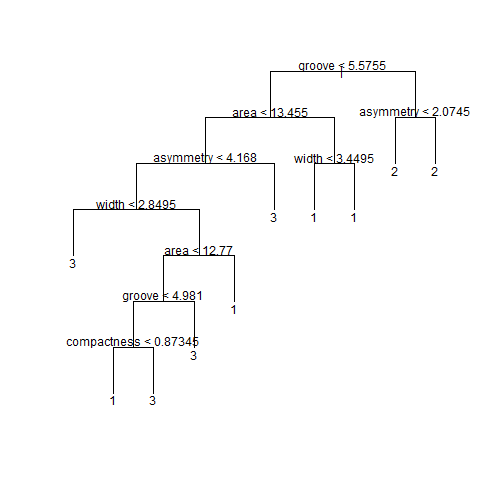
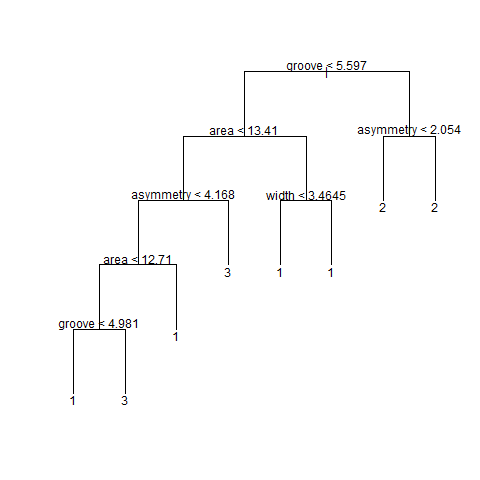
When the test phase is finished, the confusion matrix obtained shown below. We can observe our test performance from that. In the test phase, our model predicts the test sample which encounter first time to classify which type is that wheat sample. So, our misclassification error rate is 0.0476 = 2 / 42. Types of 2 wheat samples of the test samples could not be classified correctly using our decision tree.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **Predicted** | | | |
| **Actual** | Type | 1 | 2 | 3 |
| 1 | **10** | **0** | **0** |
| 2 | **0** | **15** | **0** |
| 3 | **2** | **0** | **15** |

After train and test phase is finished, cross validation needs to be performed with randomly selected different samples. Then, we can see accuracy changes and the average accuracy. The cross validation takes 100 iterations and the accuracy changing for each iteration is shown below. So, average accuracy from the cross validation was calculated 0.92 (%92)



Highest accuracy is 1 (%100) and iteration number 58 and 98 have it. Accuracy over 100 iterations changes between min accuracy 0.8095238 and highest accuracy 1. You can see below the decision trees which have the highest accuracy over 100 iterations.

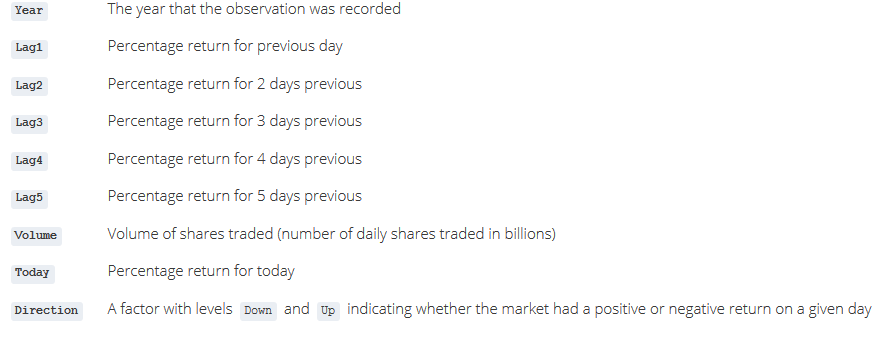


58. iteration 98. Iteration

These two trees are little bit different from each other, but we can observe that their root nodes are same (however, the threshold is little bit different). Also, most of the other trees which has different accuracy over 100 iterations has the same descriptive feature for the root node. So, we can definitely say that the best descriptive feature which information gain is the highest is “groove”. Besides, first 4 internal nodes use the same descriptive feature in two of those.

**Second Part**

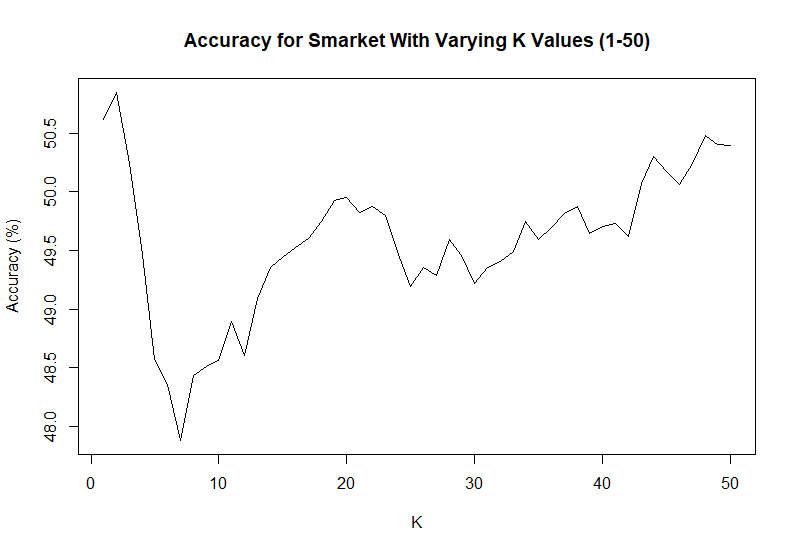
In the second part, Smarket dataset has been used. In this dataset, we have 8 descriptive features and 1 target feature. 1250 observations are included in this dataset.



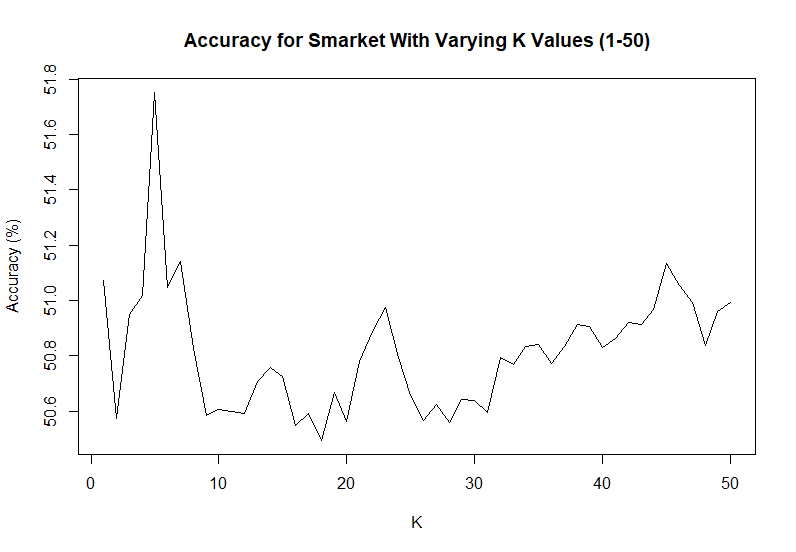
However, we have been used just 3 combination of these descriptive features; (“Lag1, Lag2, Lag3”, “Lag3, Lag4, Lag5” and “Lag1, Lag2, Lag3, Lag4, Lag5”). The target feature is “Direction”. So, the purpose is to find best descriptive feature combination to predict direction of the stock market using KNN algorithm and the best K value for each combination.

%75 of the dataset has been used for training and the remaining part has been used for test phase. Also, cross validation took 100 iterations for K values between 1 – 50.

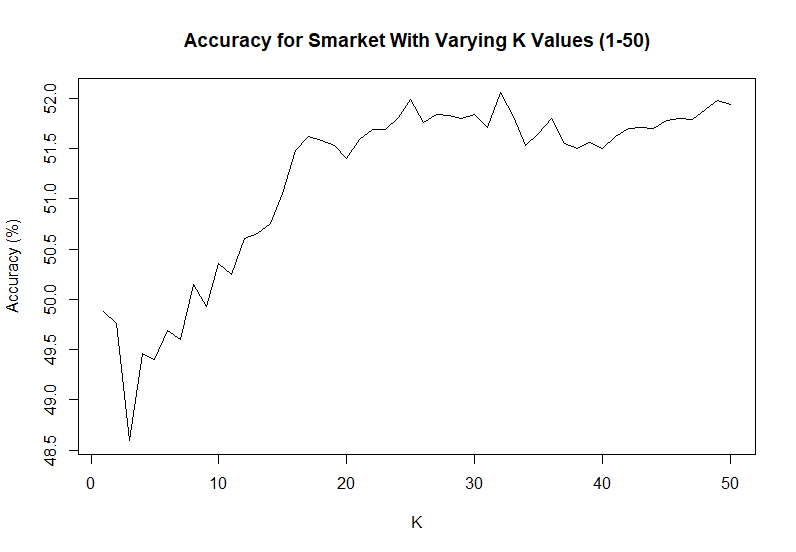
When first feature combination “Lag1, Lag2, Lag3” used, the plot below obtained. The K value that has the highest accuracy is 2 and its accuracy is 0.5084665. After that K value the accuracy fall very fast until K is 7 (lowest accuracy). Then, until K is 50, the accuracy slowly decreasing. However, the accuracy that we obtained from K=2 was never passed.



For second feature combination “Lag3, Lag4, Lag5”, the K value that has the highest accuracy got 5. As you can see below, accuracy is suddenly increasing when the K value is 5. Besides, the highest accuracy obtained 0.517508. Also, worst K value is 18 and its accuracy is 0.5049521.



When we come our last feature combination “Lag1, Lag2, Lag3, Lag4, Lag5”, the highest mean accuracy obtained among the combinations and highest accuracy also obtained. Accuracy has the peak value for K = 32 and the accuracy is 0.5205431.



Consequently, K value that has the best highest accuracy to predict the direction the stock market is 32 with “Lag1, Lag2, Lag3, Lag4, Lag5” feature combination. There is not so much difference for accuracy values among the feature combinations for different K values between 1-50, but the “Lag1, Lag2, Lag3, Lag4, Lag5” combination is the best to predict “Direction” column with K=32 value.

**R Codes**

# =========================FIRST PART==================================

#install.packages("png") # this library needs to be installed to read png files

library("png")

library(tree)

# read txt file as dataframe

wheat\_dataset <- read.table(file = "C:/Users/ABRA/Documents/RWorkingDirectory/Homeworks/hw3/wheat\_types.txt", header = TRUE, sep = ";", dec = ".")

wheat\_dataset$type <- as.factor(wheat\_dataset$type)

# construct a DT with tree function using wheat\_types dataset

original\_DT <- tree(formula = type ~ ., data=wheat\_dataset, split = "deviance")

original\_DT

# display the results

summary(original\_DT)

misclass.tree(original\_DT)

# visualize DT

plot(original\_DT, type = "uniform")

text(original\_DT)

# for reproducibility

set.seed(1233)

subset <- sample(1:nrow(wheat\_dataset), size=nrow(wheat\_dataset)\*0.8)

wh.tr <- tree(type ~ ., data = wheat\_dataset, subset = subset)

wh.tr

summary(wh.tr)

misclass.tree(wh.tr)

# plot final DT

plot(wh.tr, type = "uniform")

text(wh.tr)

# Compute training performance of the DT by using only training samples (their indices were saved in the "subset" vector)

train\_predict <- table(predict(wh.tr, wheat\_dataset[subset, ], type = "class"), wheat\_dataset[subset, "type"])

rownames(train\_predict) <- paste("Actual", rownames(train\_predict), sep = ":")

colnames(train\_predict) <- paste("Predicted", colnames(train\_predict), sep = ":")

print(train\_predict)

# Compute test performance of the DT by using only test samples

test\_predict <- table(predict(wh.tr, wheat\_dataset[-subset, ], type = "class"), wheat\_dataset[-subset, "type"])

rownames(test\_predict) <- paste("Actual", rownames(test\_predict), sep = ":")

colnames(test\_predict) <- paste("Predicted", colnames(test\_predict), sep = ":")

print(test\_predict)

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

dt\_accuracy <- numeric()

set.seed(12345)

for(i in 1:100){

subset <- sample(1:nrow(wheat\_dataset), size=nrow(wheat\_dataset)\*0.8)

temp.tr <- tree(type ~ ., data = wheat\_dataset, subset = subset)

test\_predict <- table(predict(temp.tr, wheat\_dataset[-subset, ], type = "class"), wheat\_dataset[-subset, "type"])

dt\_accuracy <- c(dt\_accuracy, sum(diag(test\_predict)) / sum(test\_predict))

dt\_name <- sprintf('dt\_%d.png', i) #iteration number to filename

plot(temp.tr, type = "uniform")

text(temp.tr)

dev.copy(png, dt\_name) # save the DT with its iteration number

dev.off()

}

cat("Mean accuracy is", mean(dt\_accuracy), "\n")

cat("Min accuracy is", min(dt\_accuracy), "\n")

cat("Max accuracy is", max(dt\_accuracy), "\n")

plot(dt\_accuracy\*100, type="l", ylab="Accuracy (%)", xlab="Iterations", main="Accuracy Changings with Different Subsets of Data")

for(i in 1:100){

if(dt\_accuracy[i] == max(dt\_accuracy)){

dt\_name <- sprintf('dt\_%d.png', i) #iteration number

cat(i, ". iteration has the highest accuracy.\n")

img <- readPNG(dt\_name) # read the DT that has the highest accuracy

plot.new()

rasterImage(img,0,0,1,1) # display DT that got highest accuracy from cross validation

}

}

# =========================SECOND PART==================================

library(ISLR)

library(class)

# Upload the Smarket data

data(Smarket)

# created a funtion for cross validation with KNN algorithm to use for each feature combination.

knnOperations <- function(descriptive\_features){

set.seed(19961)

trial\_sum <- numeric(50)

trial\_n <- numeric(50)

for(i in 1:100){

smar\_sample <- sample(1:nrow(Smarket), size=nrow(Smarket)\*0.75)

smar\_train <- Smarket[smar\_sample, descriptive\_features]

smar\_test <- Smarket[-smar\_sample, descriptive\_features]

test\_size <- nrow(smar\_test)

for(j in 1:50){

predict <- knn(smar\_train[,-length(descriptive\_features)], smar\_test[,-length(descriptive\_features)], smar\_train$Direction, k=j)

trial\_sum[j] <- trial\_sum[j] + sum(predict==smar\_test$Direction)

trial\_n[j] <- trial\_n[j] + test\_size

}

}

cv\_acc <- trial\_sum / trial\_n

cat("Max accuracy value is", max(cv\_acc), "\n")

cat("Mean accuracy value is", mean(cv\_acc), "\n")

cat("Min accuracy value is", min(cv\_acc), "\n")

k\_vals <- c(1:50)

cat("K value that got highest accuracy is ", k\_vals[cv\_acc == max(cv\_acc)], "\n")

cat("K value that got lowest accuracy is ",k\_vals[cv\_acc == min(cv\_acc)], "\n")

plot(trial\_sum / trial\_n \* 100, type="l", ylab="Accuracy (%)",xlab="K",main="Accuracy for Smarket With Varying K Values (1-50)")

}

des\_features1 <- c("Lag1", "Lag2", "Lag3", "Direction")

des\_features2 <- c("Lag3", "Lag4", "Lag5", "Direction")

des\_features3 <- c("Lag1", "Lag2", "Lag3", "Lag4", "Lag5", "Direction")

knnOperations(des\_features1)

knnOperations(des\_features2)

knnOperations(des\_features3)