BUSINESS DATA MINING (IDS 572)

HOMEWORK 2

Ajay Pawar(676955899) & Priyanshi Patel(650927804).

Q1: Wisconsin Diagnosis Breast Cancer

Ans:

```
#Q1 (a) - Cleaning the data

#From the data, last column - 'Tissue' is the variable

#that could be used as Major Predictor of diagnosis

testX = read.csv("testX.csv",header = FALSE)

testY = read.csv("testY.csv",header = FALSE)

trainX = read.csv("trainX.csv",header = FALSE)

trainY = read.csv("trainY.csv",header = FALSE)


#Merging Test Tables

testXY = cbind(testX,testY)


#Merging Train Tables

trainXY = cbind(trainX,trainY)


#Providing Column Names

colnames(testXY) = c("R.Mean", "T.Mean","P.Mean", "A.Mean","S.Mean","CP.Mean","CC.Mean",
"NC.Mean", "SY.Mean", "F.Mean","R.SD", "T.SD","P.SD", "A.SD","S.SD","CP.SD","CC.SD", "NC.SD",
"SY.SD", "F.SD","R.LAR", "T.LAR","P.LAR", "A.LAR","S.LAR","CP.LAR","CC.LAR", "NC.LAR", "SY.LAR",
"F.LAR","Tissue")


#Making Tissues as Factor

testXY$Tissue = as.factor(testXY$Tissue)


#Providing Column names to train data
```

```
colnames(trainXY)[1:31] = c("R.Mean", "T.Mean","P.Mean", "A.Mean","S.Mean","CP.Mean","CC.Mean",
"NC.Mean", "SY.Mean", "F.Mean","R.SD", "T.SD","P.SD", "A.SD","S.SD","CP.SD","CC.SD", "NC.SD",
"SY.SD", "F.SD","R.LAR", "T.LAR","P.LAR", "A.LAR","S.LAR","CP.LAR","CC.LAR", "NC.LAR", "SY.LAR",
"F.LAR","Tissue")


#cleaning the train data

#Using Outlier

outlier = function(value)

{

  iqr = IQR(value)

  q1 = as.numeric(quantile(value,0.25))

  q3 = as.numeric(quantile(value,0.75))

  higher = q3 + 1.5 * iqr

  lower = q1 - 1.5 *iqr

  ifelse(value < higher & value >lower, value , NA)

}

trainXY_outlier = sapply(trainXY[,1:30],outlier)

View(trainXY_outlier)

train_clean = data.frame(trainXY_outlier,trainXY[31])

train_final = na.omit(train_clean)


train_final$Tissue = as.factor(train_final$Tissue)

 summary(train_final)

str(train_final)
```

Major predictor of the data is Tissue, as we found out that the last Tissue column is a variable and could be used as the major predictor for our diagnosis with the diagram.

There are infact missing values or NAs in our data and for cleaning the data we specified an outlier function which omitted the NA values and divided our data into upper and lower quartile for assigning values to it.

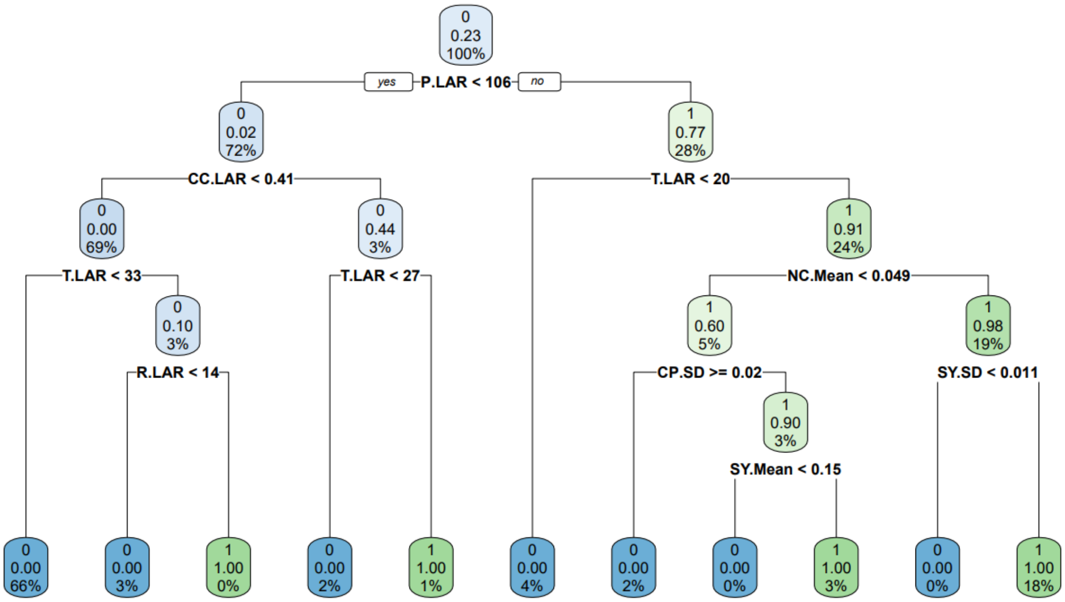
(b) Create a decision tree (using “information” for splits) to its full depth. How many leaves are in this tree?

Ans:

```
#classification and regression tree

C_Rtree=ctree(Tissue~.,data=train_final)

plot(C_Rtree)


#Create a simple decision tree using rpart using train data

#Decision Tree

C_R = rpart(Tissue~., data = train_final)

rpart.plot(C_R)


#Creating a full depth Decision tree using 'information' as split

C_R_full = rpart(Tissue~., data = train_final, parms = list(split = "information"), control =
rpart.control(minsplit = 0, minbucket = 0, cp = -1))

rpart.plot(C_R_full)

print(C_R_full)


summary(C_R_full)


#Calculating Leaf nodes

printcp(C_R_full)
```

From this we can see that there are 11 terminal nodes in our decision tree. We made a full depth decision tree using the information, where our minimum split being zero and minimum bucket being zero and cp value = -1.

(c) What are the major predictors of diagnosis suggested by your tree? Please justify your reasoning. Do these major predictors are the same as the ones you observed in part [a]?



The major predictors for the train data observed from the full depth decision tree are:

The root node: Largest Parameter. Using the summary function we know that the Largest Parameter is of greatest importance and can be considered one of the major predictors for the diagnosis along with Largest Area and Largest Radius. After evaluating the parameters we come to a conclusion that it contradicts to what we thought in PartA.

(d) Give two strong rules that describe who is likely to have cancer. Please justify your choices. (e) What is the accuracy of your decision tree model on the training data? What is the accuracy of this model on the test data?

```
rules  = arules::apriori(data = train_final, parameter = list(supp = 0.1, conf = 0.8))


arules::inspect(rules[1:10])
```

Using the data from the output of the above code, we can state that:

98.6% of people having malignant cancerous tissues have largest value for Perimeter in the range of [102,166]

97.22% of people having malignant cancerous tissues have largest value for Radius in the range of [15.5,24.6]

(e) What is the accuracy of your decision tree model on the training data? What is the accuracy of this model on the test data?

```
#prediction for train data

train_predicted = predict(C_R_full,train_final,type = "class")

print(train_predicted)


mean(train_final$Tissue == train_predicted)

#Confusion Matrix

confusionMatrix(train_predicted,train_final$Tissue)


#Prediction for test data

test_predicted_class = predict(C_R_full,testXY, type = "class")

print(test_predicted_class)


#Confusion Matrix

confusionMatrix(test_predicted_class, testXY$Tissue)
```

From the above line of code we can predict that the accuracy for the training data is more than the accuracy of test data, by 0.18.

(f) Construct the "best possible" decision tree to predict the Y labels. Explain how you construct such tree and how you evaluate its performance.

```
C_R_full_gini = rpart(Tissue~., data = train_final, parms = list(split = "gini"), control =
rpart.control(minsplit = 0, minbucket = 0, cp = -1))


summary(C_R_full_gini)


print(C_R_full_gini)


rpart.plot(C_R_full_gini)


train_predicted_gini = predict(C_R_full_gini,train_final,type = "class")


print(train_predicted_gini)


confusionMatrix(train_predicted_gini,train_final$Tissue)


model_tree_train = tree(Tissue~ P.LAR  +
              A.LAR +
              R.LAR +
              P.Mean +
              A.Mean +
              R.Mean, data = train_final)
summary(model_tree_train)
```

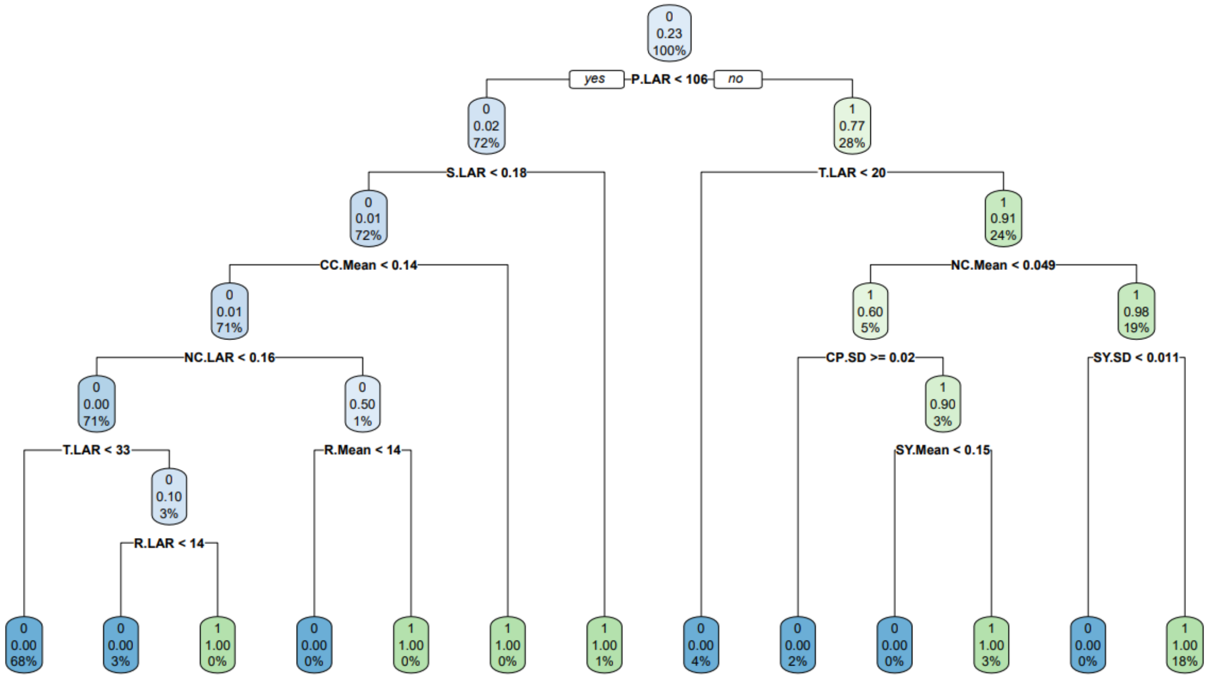
```
plot(model_tree_train, type = "uniform")


text(model_tree_train, cex = 0.8)
```

This tree can be constructed by using our major predictors of diagnosis which are Largest Perimeter, Largest Area, Largest Radius, Perimeter Mean, Area Mean, Radius Mean. We will construct a model based on these to predict our Y labels.

(g) Plot your final decision tree model and write down all decision rules that you will consider for predictions.

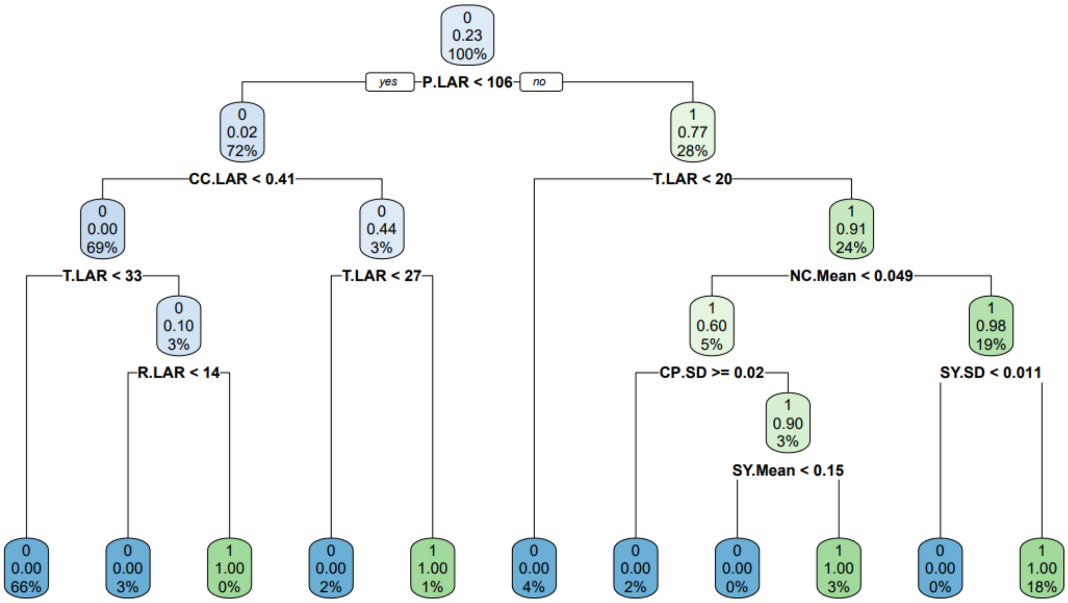
#Index Decision Tree

rpart.plot(C_R_full_gini)



#Decision tree with Information

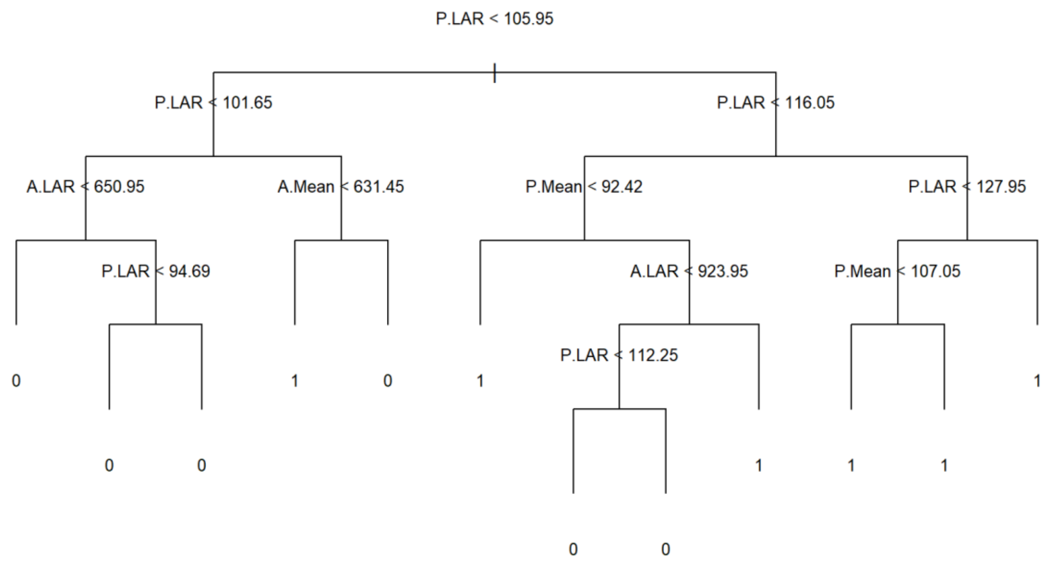
rpart.plot(C_R_full)



#Tree based on major Predictors

plot(model_tree_train, type = "uniform")

text(model_tree_train, cex = 0.8)



Q2 : Zoo Animals

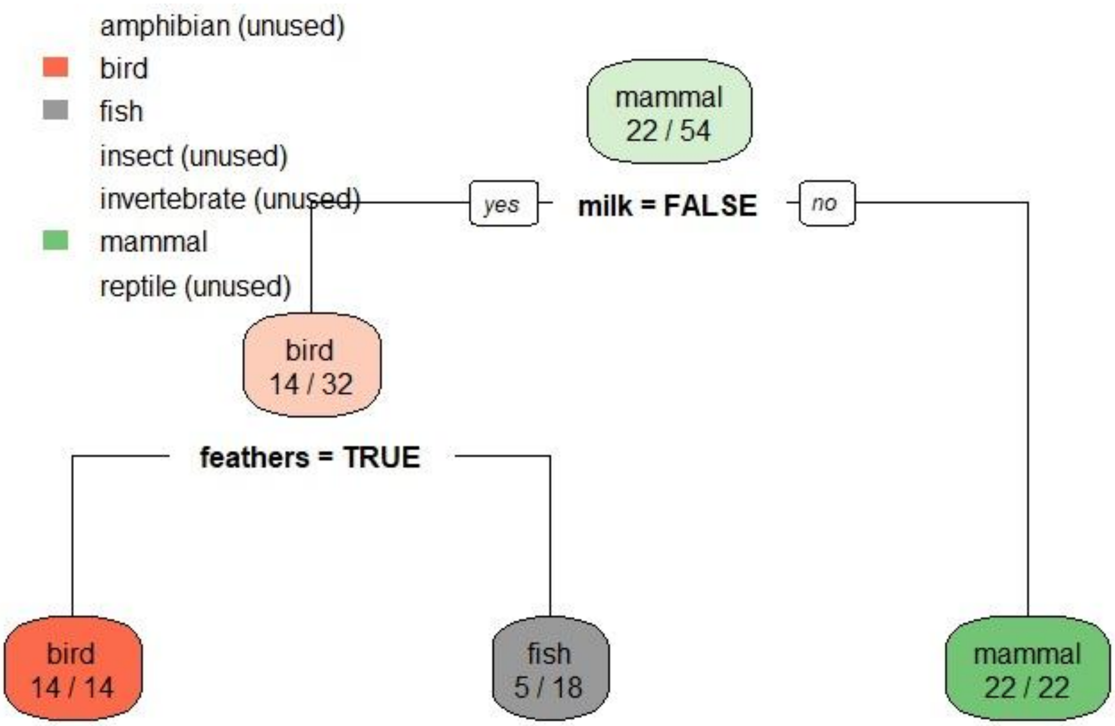
(a) Input the zoo1.csv to train the decision tree classifier and come up with a decision tree to classify a new record into one of the categories. Make sure you examine the data first and think about what variables to use for the classification scheme.

```
zoo_data = read.csv("zoo.csv")

df_zoo = data.frame(zoo_data)

df_zoo = df_zoo[-1,]

colnames(df_zoo) = c("Animals", "Hair","Feath", "Eggs","Milk","Airborn","Aqua", "Pred", "Tooth",
"Back","Breath", "Venom","Fins", "Legs","Tail","Domestic","Catsize", "Type")

test_data <- df_zoo

head(test_data)


test_data <- test_data %>%

  modify_if(is.logical, factor, levels = c(TRUE, FALSE)) %>%

  modify_if(is.character, factor)


summary(test_data)
```

```
dTree_test <- test_data %>%
  rpart(Type ~., data = .)


rpart.plot(dTree_test, extra = 2, roundint=FALSE,
           box.palette = list("Violet", "Blueviolet", "Blue", "Green", "Yellow", "Orange", "Red")) # specify 7 colors





zoo1_data = read.csv("zoo1.csv")
df_zoo1 = data.frame(zoo1_data)
df_zoo1 = df_zoo1[-1,]


head(df_zoo1)
```

We have used Type as our variable for classifying and constructed a rplot.

(b) Draw your decision tree - how many leaves does it have? What is the classification accuracy on the training and testing datasets?

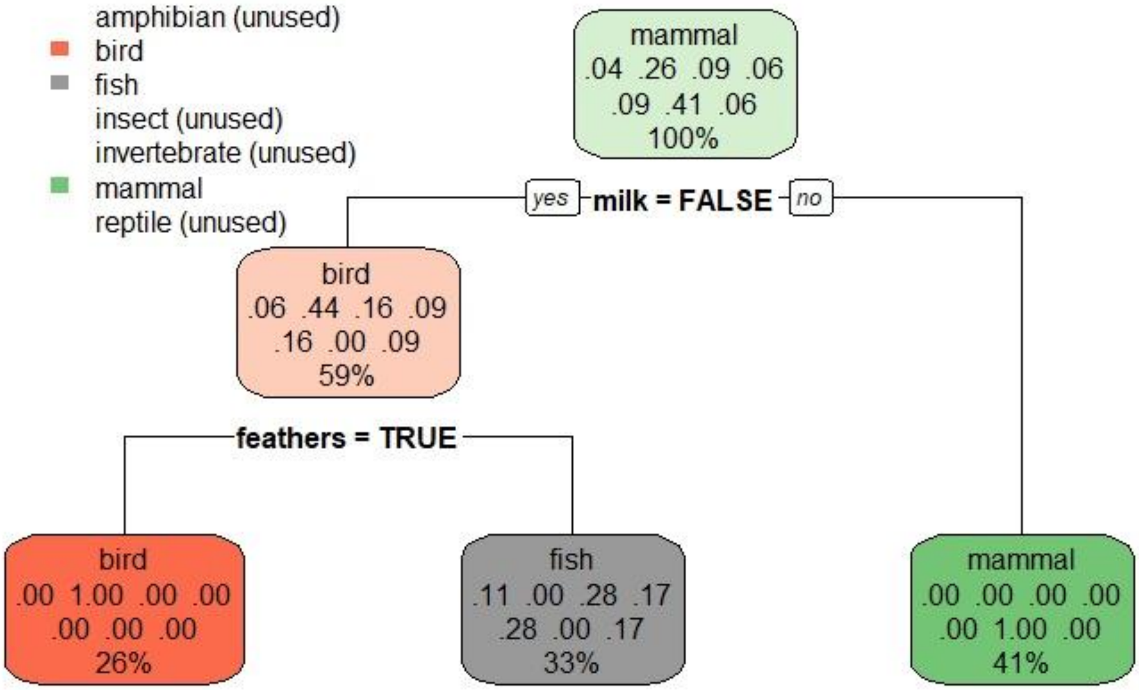


Considering different variables these were our decision tree pruning parameters

(c) Play with the decision tree pruning parameters and see how they change the performance of the tree.

```
# create tree for Response Target variable and all other variables##

mytree <- rpart(type ~ . , method='class', data= train_1[,-c(1)])

rpart.plot(mytree)

print(mytree)


rpart.plot(mytree, extra = 2)


prediction <- predict(mytree, test_1[,-c(1,18)], type="class")

test_results <- as.data.frame(prediction)

#Binding result to Column Type

test_results <- cbind(test_results, test_1$type)

colnames(test_results) <- c('predicted', 'Actual')
```

```
#confusion matrix

confusionMat <- table(test_1$type, predict)
```


amphibian (unused)
🟥 bird
■ fish
insect (unused)
invertebrate (unused)
🟩 mammal
reptile (unused)
mammal
.04 .26 .09 .06
.09 .41 .06
100%
yes
milk = FALSE
no
bird
.06 .44 .16 .09
.16 .00 .09
59%
feathers = TRUE
bird
.00 1.00 .00 .00
.00 .00 .00
26%
fish
.11 .00 .28 .17
.28 .00 .17
33%
mammal
.00 .00 .00 .00
.00 1.00 .00
41%