LAB ASSESSMENT – 6

Prithviraj Guntha
20BRS1188

Decision Tree and Random Forest are both machine learning algorithms used for classification and regression tasks. The main difference between decision tree and random forest is that decision tree is built from one decision tree while random forest is built from multiple decision trees. Random Forest tries to overcome the overfitting problem of decision trees by aggregating the results of multiple trees.
The algorithms for both the models are given below:

Decision Tree:
1. Select the best attribute to split the data based on information gain or gini index.
2. Create a new decision tree node for the selected attribute and assign it as the root node.
3. Split the data into subsets based on the values of the selected attribute.
4. For each subset, repeat the process from step 1 to step 3 until all the data in the subset belongs to the same class or no more attributes are left to split.
5. Each leaf node in the decision tree represents a class label.

Random Forest:
1. Randomly select "k" data points from the training set.
2. Build a decision tree based on the selected "k" data points.
3. Repeat steps 1 and 2 "n" number of times, where "n" is the number of trees in the forest.
4. For a new data point, make a prediction by passing it through all "n" decision trees and selecting the class label that appears most frequently.

Code:

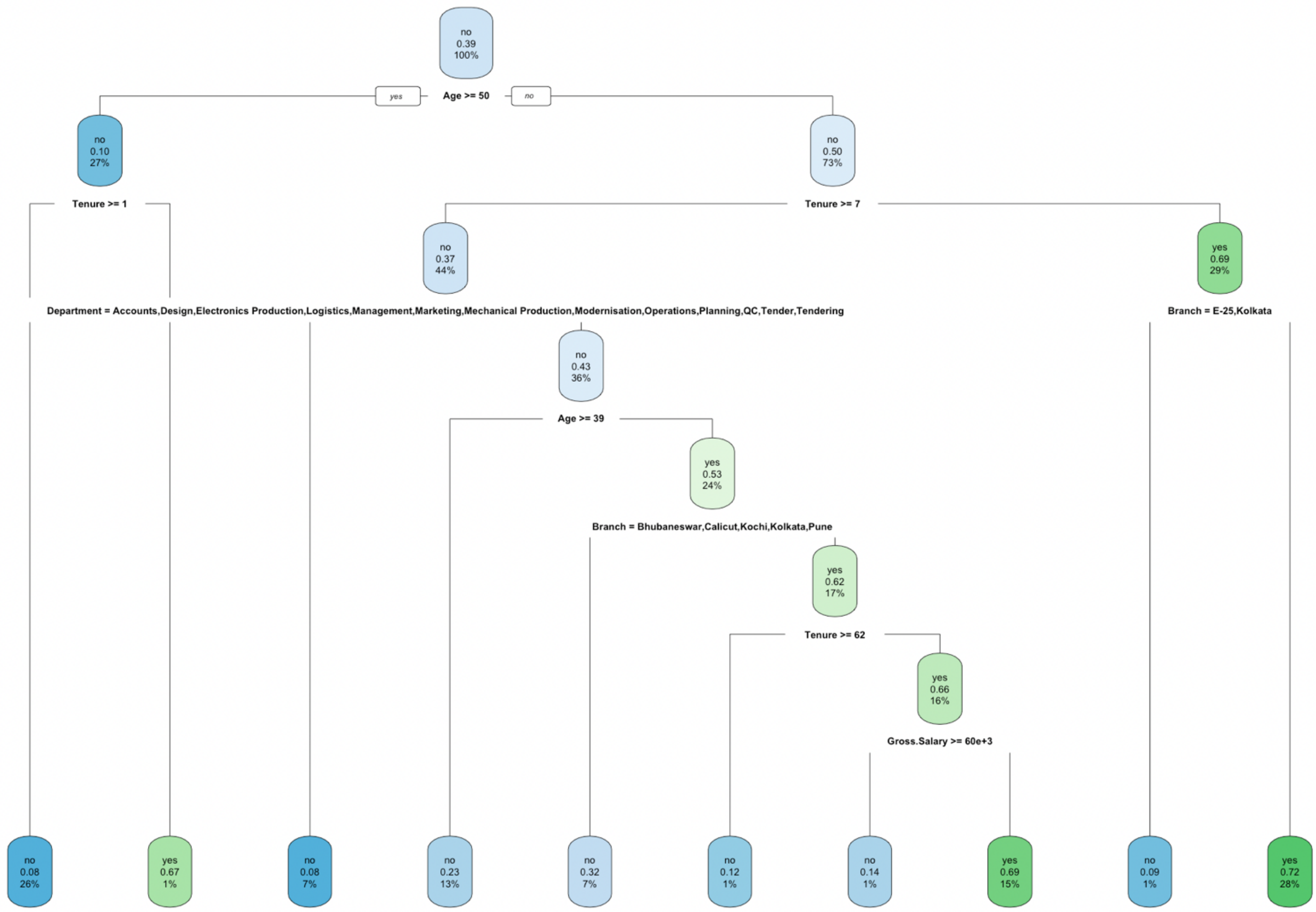
```
library(caret)
library(rpart)
library(rpart.plot)
library(randomForest)

data = read.csv("/Users/prithviraj/Desktop/sem 6/EmployeeDat.csv")
data = data[,-1]
str(data)
```

```
#since tenure is a character type, we will extract required info and normalise it
into only months.
years = c()
for( i in 1:length(data$Tenure)){
  num <- substring(data$Tenure[i],1,1)
  years[i] <- num
}
years <- as.integer(years)
yearconver = c()
for(i in 1:length(years)){
  num <- years[[i]] * 12
  yearconver <- c(yearconver , num)
}
months = c()
for( i in 1:length(data$Tenure)){
  num <- substring(data$Tenure[i],9,10)
  num <- gsub(" ","",num)
  months <- c(months,num)
}
months <- as.integer(months)
Tenure <- yearconver + months

#now lets drop the Tenure column from the dataframe and add our updated
tenure column
data <- subset(data , select = -c(Tenure))
data['Tenure'] <- Tenure

#let us convert the data into the required datatypes
data$Grade <- as.factor(data$Grade)
data$Branch <- as.factor(data$Branch)
data$Department <- as.factor(data$Department)
data$Gender <- as.factor(data$Gender)
data$Gross.Salary <- as.integer(data$Gross.Salary)
data$Age <- as.integer(data$Age)
data$resigned<- as.factor((data$resigned))
data$Tenure <- as.integer((data$Tenure))

#decision tree
index = createDataPartition(y=data$resigned, p = .80,list = FALSE )
traind = data[index,-7]
testd  = data[-index,-7]
train_labelsd <- data[index , 7]
```

```
test_labelsd <- data[-index,7]
rtree_fit <- rpart(train_labelsd~ ., traind, method='class')
rpart.plot(rtree_fit)
pred_rtree <- predict(rtree_fit, testd, type= 'class')
confusionMatrix(pred_rtree,test_labelsd)

#random forest
index = createDataPartition(y=data$resigned, p = .80,list = FALSE )
traind = data[index,-7]
testd  = data[-index,-7]
train_labelsd <- data[index , 7]
test_labelsd <- data[-index,7]
model_rf <- randomForest(train_labelsrf ~ ., data = trainrf, importance =
TRUE)
pred_rf <- predict(model_rf, testrf, type = "class")
confusionMatrix(pred_rf, test_labelsrf)
```

Output:

1. Decision tree



```
Confusion Matrix and Statistics

          Reference
Prediction no yes
       no  85  20
       yes 27  51

               Accuracy : 0.7432
                 95% CI : (0.6735, 0.8048)
    No Information Rate : 0.612
    P-Value [Acc > NIR] : 0.0001273

                  Kappa : 0.4688

 Mcnemar's Test P-Value : 0.3814706

            Sensitivity : 0.7589
            Specificity : 0.7183
         Pos Pred Value : 0.8095
         Neg Pred Value : 0.6538
             Prevalence : 0.6120
         Detection Rate : 0.4645
   Detection Prevalence : 0.5738
      Balanced Accuracy : 0.7386

       'Positive' Class : no
```

2. Random forest

```
Confusion Matrix and Statistics

          Reference
Prediction no yes
       no  93  13
       yes 19  58

               Accuracy : 0.8251
                 95% CI : (0.7622, 0.8772)
    No Information Rate : 0.612
    P-Value [Acc > NIR] : 3.485e-10

                  Kappa : 0.6374

 Mcnemar's Test P-Value : 0.3768

            Sensitivity : 0.8304
            Specificity : 0.8169
         Pos Pred Value : 0.8774
         Neg Pred Value : 0.7532
             Prevalence : 0.6120
         Detection Rate : 0.5082
   Detection Prevalence : 0.5792
      Balanced Accuracy : 0.8236

       'Positive' Class : no
```

Result:
Hence we have successfully implemented the random forest and decision tree models for predicting employee attrition.
Accuracy using decision tree = 0.74
Accuracy using random forest = 0.82

Inference:
As we can see the accuracy obtained using random forest is much better than the accuracy obtained using decision tree, hence our ideal model is the random forest model.