```
library(dplyr)
library(irr)
library(caret)
library(rpart)
library(rpart.plot)
library(ROCR)
library(randomForest)


#data prep
>creditcard<-read.csv("D:/creditcard.csv")
>summary(creditcard)

#There are no missing values and all variables are numeric
> table(creditcard$Class)

     0      1
284315    492

#This is highly imbalanced dataset, only 492 out of 284807 observations are fraud

#data partitioning
>set.seed(100)
>index <- sample(1:nrow(creditcard), nrow(creditcard)*0.7)
>training<- creditcard[index,]
>validation<- creditcard[-index,]
> table(training$Class)

     0      1
198998    366


# Decision Tree
# Apply decision tree model on training dataset
mod<-rpart(Class~.,data=training, method="class")

#Evaluation of model on validation dataset using area under ROC curve
predicted <- predict(mod ,validation, type="prob")
area_under_curve<-auc(validation$Class, predicted[,2])
area_under_curve
Area under the curve: 0.89


#Random Forest
#Implementing Random forest on training dataset
>n <- names(training)
>rf.form <- as.formula(paste("Class ~", paste(n[!n %in% "Class"],           collapse
= " + ")))
> trainset.rf <- randomForest(rf.form, training ,ntree=100,importance=T)

#Evaluation of model on validation dataset using area under ROC curve

> predicted0<- predict(trainset.rf ,validation, type="prob")
> area_under_curve0<-auc(validation$Class, predicted0[,2])
> area_under_curve0
Area under the curve: 0.9181


# This is imbalanced classification problem, so use function ROSE from    library ROSE
for synthetically generating data from training dataset

> library(ROSE)
> training$Class<-as.factor(training$Class)
> data.rose <- ROSE(Class ~ ., data = training, seed = 100)$data
> table(data.rose$Class)
    0     1
99849 99515
```

```
# Applying Decision Tree model on This synthetically generated data

> mod1<-rpart(Class~.,data=data.rose, method="class")

#Accuracy on validation dataset
> predicted1 <- predict(mod1 ,validation, type="prob")
> area_under_curve1<-auc(validation$Class, predicted1[,2])
> area_under_curve1
Area under the curve: 0.9011
# AUC value for decision tree is increased from 0.89 to 0.9011


# Applying Random Forest model on This synthetically generated data

> n <- names(data.rose)
> rf.form1 <- as.formula(paste("Class ~", paste(n[!n %in% "Class"],         collapse =
" + ")))
> trainset.rf1 <- randomForest(rf.form1,data.rose,ntree=100,importance=T)

#Accuracy on validation dataset
>predicted2<- predict(trainset.rf1 ,validation, type="prob")              > area_under
_curve1<-auc(validation$Class, predicted2[,2])
> area_under_curve1
Area under the curve: 0.9224
# AUC value for Random forest is increased from 0.9181 to 0.9224




#sampling
#We use function ovun.sample from library ROSE on imbalanced training dataset and take
sample of 10000 observations.

> library(ROSE)
> training$Class<-as.factor(training$Class)
> data_balanced_under <- ovun.sample(Class ~ ., data = training, method = "both" ,N=10
000 , seed = 300)$data
> table(data_balanced_under$Class)
   0    1
4894 5106



# Applying Decision Tree model on sampled data

> mod2<-rpart(Class~.,data=data_balanced_under, method="class")

#Accuracy on validation dataset
> predicted3 <- predict(mod2 ,validation, type="prob")
> area_under_curve3<-auc(validation$Class, predicted3[,2])
> area_under_curve3
Area under the curve: 0.9336
#AUC value on original dataset was 0.89 and is increased to 0.9336


# Applying Random Forest on This sampled data

> n <- names(data_balanced_under)
> rf.form2<- as.formula(paste("Class ~", paste(n[!n %in% "Class"], collapse = " + ")))
> trainset.rf2 <- randomForest(rf.form2, data_balanced_under,ntree=500,importance=T)

#Accuracy on validation dataset
> predicted4 <- predict(trainset.rf2 ,validation, type="prob")
> area_under_curve4<-auc(validation$Class, predicted4[,2])
> area_under_curve4
Area under the curve: 0.9647
#AUC value on original dataset was 0.9181 and is increased to 0.9647
```

#XGBoost

#Loading required libraries

```
>library(xgboost)
>library(magrittr)
>library(Matrix)
```

#we need to make data such that it is used in xgboost model
#training and test dataset should be in xgb.DMatrix (xgboost's own datatype)

```
>train_label<-training[,"Class"]
>data.train<- xgb.DMatrix(as.matrix(training[, colnames(training) != "Class"]), label
= train_label)
>test_label<-validation[,"Class"]
>data.test<- xgb.DMatrix(as.matrix(validation[, colnames(validation) != "Class"]), lab
el = test_label)
```

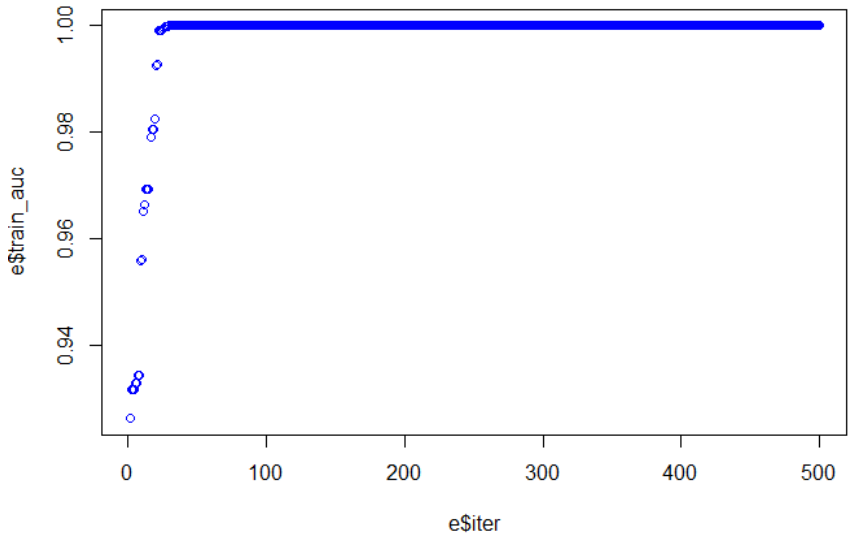
#parameters
#Here problem is classification type so objective function will be "binary:logistic",
#we set evaluation metric as auc. Then train the model using xgb.train function

```
>n1<-length(unique(train_label))
>parameters<-list("objective"="binary:logistic","eval_metric"="auc","numclass"=n1)
>watchlist<-list(train= data.train, test= data.test)


>XGB_model<-xgb.train(params=parameters,   data=data.train,   nrounds=500,
+                       watchlist=watchlist)
```

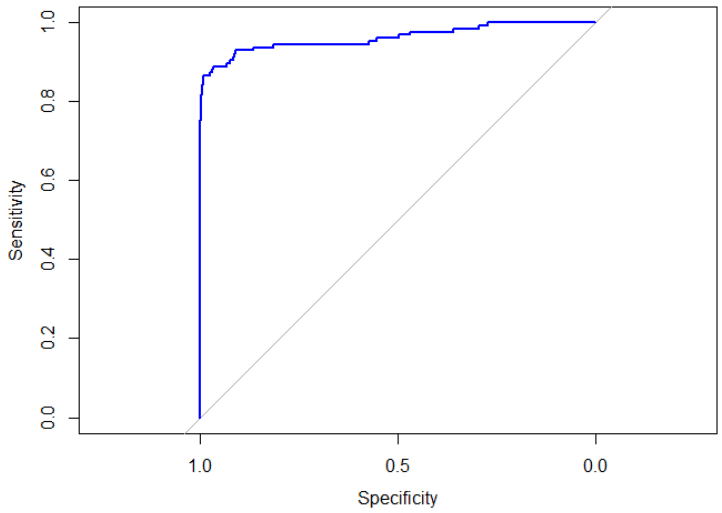
##training and test error plot

```
> error<-data.frame(XGB_model$evaluation_log)
> plot(error$iter, error$train_auc ,col='blue')
```



##validation of model

```
>predicted5 = predict(bst_model, newdata = as.matrix(validation[, colnames(validation)
!= "Class"], ntreelimit = bst_model$bestInd)
>library(pROC)
>area_under_curve5 = roc(validation$Class, predicted5, plot = TRUE, col = "blue")
```




```
>print(area_under_curve5)

Call:
roc.default(response = validation$Class, predictor = predicted5,     plot = TRUE, col
= "blue")

Data: predicted5 in 85317 controls (validation$Class 0) < 126 cases (validation$Class
1).

Area under the curve: 0.9617
```