Uttam Parts

creditcard\_data$Amount=scale(creditcard\_data$Amount)

NewData=creditcard\_data[,-c(1)]

head(NewData)

///

data\_sample = sample.split(NewData$Class,SplitRatio=0.80)

train\_data = subset(NewData,data\_sample==TRUE)

test\_data = subset(NewData,data\_sample==FALSE)

dim(train\_data)

dim(test\_data)

//

library(rpart.plot)

decisionTree\_model <- rpart(Class ~ . , creditcard\_data, method = 'class')

predicted\_val <- predict(decisionTree\_model, creditcard\_data, type = 'class')

probability <- predict(decisionTree\_model, creditcard\_data, type = 'prob')

rpart.plot(decisionTree\_model)

Sohaib Parts

library(caret)

library(ggplot2)

library(smotefamily)

library(pROC)

library(rpart)

library(rpart.plot)

set.seed(0)

# load dataset and scale 'Amount' and 'time' columns then add values as new col

creditcard <- read.table("creditcard.csv",sep=',',header=T)

creditcard['Amount\_scaled']<-scale(creditcard$Amount)

creditcard['time\_scaled']<-scale(creditcard$Time)

# remove unscaled amoyunt and time columns

creditcard<-creditcard[ -c(1,30) ]

# check split between positive and negative class

table(creditcard$Class)

# create 80/20 split

trainIndex <- createDataPartition(creditcard$Class, p = .8, list = FALSE, times = 1)

train <- creditcard[trainIndex,]

test <- creditcard[-trainIndex,]

# check probability of classes in train and test split

prop.table(table(train$Class))

prop.table(table(test$Class))

# use SMOTE to create increase minority class in training dataset

# create an approximate 65/35 split

train <- SMOTE(train[,-29],train$Class,dup\_size = 310)

train<-train$data

prop.table(table(train$class))

#view distribution of response variable in train set

table(train$class)

hist(as.numeric(train$class))

#convert train and test class to factor

train$class<-as.factor(train$class)

test$Class<-as.factor(test$Class)

# build logistic model

logfit<-glm(class~., data = train,family = 'binomial')

summary(logfit)

# plot residuals

plot(predict(logfit),residuals(logfit))

abline(h=0,lty=2,col="grey")

lines(lowess(predict(logfit),residuals(logfit)),col="black",lwd=2)

# plot qqplot

qqnorm(residuals(logfit))

qqline(residuals(logfit))

#predict

logfit\_pred<- predict(logfit, test[,-29],type ='response' )

logfit\_pred<- ifelse(logfit\_pred>0.5,1,0)

# output confusion matrix

confusionMatrix(as.factor(logfit\_pred),as.factor(test$Class),positive = '1')

# Create ROC curve

roc\_score=roc(as.numeric(test$Class), as.numeric(logfit\_pred)) #AUC score

plot(roc\_score ,main ="ROC curve -- Logistic Regression ")

# decision tree with SMOTE

decisionTree\_model <- rpart(class ~ . , data= train, method = 'class')

decisionTree\_model

predicted\_val <- predict(decisionTree\_model, test, type = 'class')

# table of test predictions from decision tree

table(predicted\_val, test$Class)

# print decision tree

rpart.plot(decisionTree\_model)