Assignment 10.1

1. Use the below given data set

DataSet

2. Perform the below given activities:

a. Create classification model using different classifiers

b. Verify model goodness of fit

c. Apply all the model validation techniques.

**ANS**

library(C50)

data(churn)

head(churnTrain)

head(churnTest)

# load libraries

library(caret)

library(rpart)

# define training control

train\_control<- trainControl(method="cv", number=10)

# train the model

model<- train(churn~., data=churnTrain, trControl=train\_control, method="rf")

model

# make predictions

predictions<- predict(model,churnTest)

# append predictions

pred<- cbind(churnTest,predictions)

# summarize results

confusionMatrix<- confusionMatrix(pred$predictions,pred$churn)

confusionMatrix

#how do we create a cross validation scheme

control <- trainControl(method = 'repeatedcv',

number = 10,

repeats = 3)

seed <-7

metric <- 'Accuracy'

set.seed(seed)

mtry <- sqrt(ncol(churnTrain))

tunegrid <- expand.grid(.mtry=mtry)

rf\_default <- train(churn~.,

data = churnTrain,

method = 'rf',

metric = metric,

tuneGrid = tunegrid,

trControl = control)

print(rf\_default)

#-------------------------------

# make predictions

predictions<- predict(rf\_default,churnTest)

# append predictions

pred<- cbind(churnTest,predictions)

# summarize results

confusionMatrix<- confusionMatrix(pred$predictions,pred$churn)

confusionMatrix

varImp(rf\_default)

#----------------

# random search for parameters

control <- trainControl(method = 'repeatedcv',

number = 10,

repeats = 3,

search = 'random')

set.seed(seed)

mtry <- sqrt(ncol(churnTrain))

rf\_random <- train(churn~.,

data = churnTrain,

method = 'rf',

metric = metric,

tuneLength = 15,

trControl = control)

print(rf\_random)

plot(rf\_random)

#--------------------

# make predictions

predictions<- predict(rf\_random,churnTest)

# append predictions

pred<- cbind(churnTest,predictions)

# summarize results

confusionMatrix<- confusionMatrix(pred$predictions,pred$churn)

confusionMatrix

varImp(rf\_random)

#--------------------

#--------------------------------------------------------------

# Grid search

control <- trainControl(method = 'repeatedcv',

number = 10,

repeats = 3,

search = 'grid')

set.seed(seed)

tunegrid <- expand.grid(.mtry=c(1:15))

#mtry <- sqrt(ncol(x))

rf\_gridsearch <- train(churn~.,

data = churnTrain[1:200,],

method = 'rf',

metric = metric,

tuneGrid = tunegrid,

trControl = control)

print(rf\_gridsearch)

plot(rf\_gridsearch)

#-------------------------

# make predictions

predictions<- predict(rf\_gridsearch,churnTest)

# append predictions

pred<- cbind(churnTest,predictions)

# summarize results

confusionMatrix<- confusionMatrix(pred$predictions,pred$churn)

confusionMatrix

varImp(rf\_gridsearch)

#---------------------------

# Boosting

# ---------------------------------------

# Boosting model requires three things

#1- a loss function to be optimized

#2- a weak learner to make predictions

#3- an additive model to add the weak learners to minimize the loss function

# gradient boosting

control <- trainControl(method = 'repeatedcv',

number = 5,

repeats = 3,

search = 'grid')

seed <- 7

library(C50)

set.seed(seed)

metric <- 'Accuracy'

gbm\_mod <- train(churn~.,

data = churnTrain,

method = 'gbm',

metric = metric,

trControl = control)

print(gbm\_mod)

plot(gbm\_mod)

summary(gbm\_mod)

# make predictions

predictions<- predict(gbm\_mod,churnTest)

# append predictions

pred<- cbind(churnTest,predictions)

# summarize results

confusionMatrix<- confusionMatrix(pred$predictions,pred$churn)

confusionMatrix