# 1. Import dataset to R using stringsAsFactors = FALSE.

# Run the following transformations:

dt1 <- read.csv(file.choose(), stringsAsFactors = F)

dt1[c(2,3,6,7,9,13,14)]<-data.frame(lapply(dt1[c(2,3,6,7,9,13,14)],factor))

dt1 <- dt1[c(2,3,9,10,11,12,14)]

# 2. Pick 70% records randomly as training dataset.

nrows <- nrow(dt1)

nrows

train.size<- floor(0.7 \* nrows)

train.size

set.seed(1234)

train.index <- sample(1:nrows,train.size,replace = F)

dt1.train<- dt1[train.index,] # 189 training observations

dt1.test<- dt1[-train.index,] # 81 testing observations

str(dt1.train)

str(dt1.test)

# 3. Build a logit regression model where statlogheart is the target variable.

model <- glm(statlogheart ~ . , data = dt1.train, family='binomial')

summary(model)

# 4. Evaluate the performance of training dataset by using cross table.

#What percentage of people who do not have heart disease are wrongly predicted?

predicted\_train<-predict(model, dt1.train, type= "response")

predicted\_train<- ifelse(predicted\_train>=0.5,2,1)

summary(predicted\_train)

install.packages('gmodels')

library(gmodels)

CrossTable(dt1.train$statlogheart, predicted\_train)

#25

# 5. Evaluate the performance of testing dataset by using cross table. What percentage of people who have heart disease are wrongly predicted?

# Hint for predict: predicted<-predict(model, dt.test, type= "response")

predicted\_test <-predict(model, dt1.test, type= "response")

summary(predicted\_test)

install.packages('gmodels')

library(gmodels)

predicted\_test<- ifelse(predicted\_test>=0.5,2,1)

CrossTable(dt1.test$statlogheart, predicted\_test)

# 16

# 6. Predict using naiveBayes

# a. Predict classification for the training dataset and create cross table.

# Note for predict: parameter type=response is needed only for logit.

# Predict classification for the testing dataset and create cross table.

install.packages('e1071')

library(e1071)

model <- naiveBayes(x=dt1.train[-7], y=dt1.train$statlogheart, laplace = 1)

predicted.train.naive <- predict(model, dt1.train[-7])

CrossTable(dt1.train[[7]], predicted.train.naive)

predicted.test.naive <- predict(model, dt1.test[-7])

CrossTable(dt1.test[[7]], predicted.test.naive)

# 7. Predict using decision tree

# a. Predict classification for the training dataset and create cross table.

# Predict classification for the testing dataset and create cross table.

install.packages("C50")

library(C50)

model1 <- C5.0(dt1.train[-7],dt1.train$statlogheart)

model1

statlog\_train\_pred <- predict(model1,dt1.train)

install.packages('gmodels')

library(gmodels)

CrossTable(dt1.train$statlogheart, statlog\_train\_pred, dnn=c("Actual", "Predicted"))

statlog\_test\_pred <- predict(model1,dt1.test)

install.packages('gmodels')

library(gmodels)

CrossTable(dt1.test$statlogheart, statlog\_test\_pred, dnn=c("Actual", "Predicted"))

# 8. Predict using knn at k=3

# a. Predict classification for the training dataset and create cross table.

# Hint: train=dt.train, test=dt.train

library(class)

library(gmodels)

predicted.train <- knn(train=dt1.train[-7], test=dt1.train[-7],

cl=dt1.train$statlogheart, k=3, prob= T)

CrossTable(dt1.train$statlogheart, predicted.train, dnn=c("Actual", "Predicted"))

# b. Predict classification for the testing dataset and create cross table.

# Hint: train=dt.train, test=dt.test

predicted.test <- knn(train=dt1.train[-7], test=dt1.test[-7],

cl=dt1.train$statlogheart, k=3)

CrossTable(dt1.test$statlogheart, predicted.test, dnn=c("Actual", "Predicted"))

# 9. Compare performance of models using training dataset

#naiveBayes <- FN = 9 FP = 17 , total = 26

#decision tree <- FN = 13 FP = 9 , total =22

#knn <- FN = 11 FP = 10 , total = 21

#logit <- FN = 13 FP = 12 , total = 25

# 10. Compare performance of models using testing dataset

#naiveBayes <- FN = 1 FP = 11 , total = 12

#decision tree <- FN = 1 FP = 11 , total = 12

#knn <- FN = 2 FP = 15 , total = 17

#logit <- FN = 1 FP = 15 , total = 16

# 11. Which model would you recommend overall?

# Hint: Typically you want to give preference to a model with good test dataset performance. If 2 models perform equally well over test dataset then look for model that performs well for training dataset.

# If you are looking for attribution and not prediction then you try to get the performance of regression model as close to the best performing other model, and then use regression model.

##Decision tree performes better

# 12. Plot ROC curves for all the above 3 models (naïve Bayes, decision tree, knn).

#Also add a reference line to indicate the performance of a classifier with no predictive value.

#

predicted.test.prob1<-predict(model.naive,dt1.test,type="raw")

pred\_naive<-prediction(predicted.test.prob[,2],dt1.test$statlogheart)

pref\_naive<- performance(pred\_naive,"tpr","fpr")

plot(pref\_naive,main="ROC Curve for Naive Model",col="blue")

abline(a=0,b=1)

predicted.test.tree.prob<-predict(model1,dt.test,type="prob")

pred\_tree<-prediction(predicted.test.tree.prob1[,2],dt1.test$statlogheart)

pref\_tree<- performance(pred\_tree,"tpr","fpr")

plot(pref\_tree,main="ROC Curve for Decision Tree",col="blue")

abline(a=0,b=1)

predicted.test.knn.prob2<- attributes(predicted.test)$prob

head(predicted.test.knn.prob)

pred\_knn<-prediction(predicted.test.knn.prob2,dt1.test$statlogheart)

pref\_knn<- performance(pred\_knn,"tpr","fpr")

plot(pref\_knn,main="ROC Curve for KNN Model",col="blue")

abline(a=0,b=1)