rm(list = ls())

options(digits=4)

install.packages("tree")

library(tree)

install.packages("e1071")

library(e1071)

install.packages(("ROCR"))

library(ROCR)

install.packages("randomForest")

library(randomForest)

install.packages("adabag")

library(adabag)

install.packages("rpart")

library(rpart)

GC=read.csv("german\_credit\_data.csv",header=TRUE)

attach(GC)

#converts numbers to class

GC$Class.label=as.factor(GC$Class.label)

View(GC)

GoodC=length(which(GC$Class.label==1))

BadC=length(which(GC$Class.label==2))

summary(GC)

set.seed(26883694) #random seed

train.row = sample(1:nrow(GC), 0.7\*nrow(GC))

GC.train = GC[train.row,]

GC.test= GC[-train.row,]

# Calculate a decision tree

GC.tree = tree(Class.label ~., data = GC.train)

#print(summary(GCD.tree))

plot(GC.tree)

text(GC.tree, pretty = 0)

# do predictions as classes and draw a table

GC.predtree = predict(GC.tree, GC.test, type = "class")

t1=table(Predicted\_Class = GC.predtree, Actual\_Class = GC.test$Class.label)

cat("\n#Decision Tree Confusion\n")

print(t1)

treeac=(194-33)/300

# do predictions as probabilities and draw ROC

GC.pred.tree = predict(GC.tree, GC.test, type = "vector")

# computing a simple ROC curve (x-axis: fpr, y-axis: tpr)

# labels are actual values, predictors are probability of class

GCDpred <- prediction( GC.pred.tree[,2], GC.test$Class.label)

GCDperf <- performance(GCDpred,"tpr","fpr")

plot(GCDperf)

abline(0,1)

#AUC calculations

cauc = performance(GCDpred, "auc")

print(as.numeric(cauc@y.values))

# Calculate naive bayes

GC.bayes = naiveBayes(Class.label ~. , data = GC.train)

GC.predbayes = predict(GC.bayes, GC.test)

t2=table(Predicted\_Class = GC.predbayes, Actual\_Class = GC.test$Class.label)

cat("\n#NaiveBayes Confusion\n")

print(t2)

bayesac=(206-39)/300

# outputs as confidence levels

GCpred.bayes = predict(GC.bayes, GC.test, type = 'raw')

GCBpred <- prediction( GCpred.bayes[,2], GC.test$Class.label)

GCBperf <- performance(GCBpred,"tpr","fpr")

plot(GCBperf, add=TRUE, col = "blueviolet")

#AUC calculations

cauc = performance(GCBpred, "auc")

print(as.numeric(cauc@y.values))

# Bagging

GC.bag <- bagging(Class.label ~. , data = GC.train, mfinal=5)

GCpred.bag <- predict.bagging(GC.bag, GC.test)

# JCpred.bag

GCBagpred <- prediction( GCpred.bag$prob[,2], GC.test$Class)

GCBagperf <- performance(GCBagpred,"tpr","fpr")

plot(GCBagperf, add=TRUE, col = "blue")

cat("\n#Bagging Confusion\n")

print(GCpred.bag$confusion)

RandomBag=(208-29)/300

#AUC calculations

cauc = performance(GCBagpred, "auc")

print(as.numeric(cauc@y.values))

#Boosting

GC.Boost <- boosting(Class.label ~. , data = GC.train, mfinal=10)

GCpred.boost <- predict.boosting(GC.Boost, newdata=GC.test)

# JCpred.boost

GCBoostpred <- prediction( GCpred.boost$prob[,2], GC.test$Class.label)

GCBoostperf <- performance(GCBoostpred,"tpr","fpr")

plot(GCBoostperf, add=TRUE, col = "red")

cat("\n#Boosting Confusion\n")

print(GCpred.boost$confusion)

boosac=(204-35)/300

#AUC calculations

cauc = performance(GCBoostpred, "auc")

print(as.numeric(cauc@y.values))

# Random Forest

GC.rf <- randomForest(Class.label ~. , data = GC.train, na.action = na.exclude)

GCpredrf <- predict(GC.rf, GC.test)

t3=table(Predicted\_Class = GCpredrf, Actual\_Class = GC.test$Class.label)

cat("\n#Random Forest Confusion\n")

print(t3)

Randomacu=(211-29)/300

GCpred.rf <- predict(GC.rf, GC.test, type="prob")

# GCpred.rf

GCFpred <- prediction( GCpred.rf[,2], GC.test$Class.label)

GCFperf <- performance(GCFpred,"tpr","fpr")

plot(GCFperf, add=TRUE, col = "darkgreen")

#table for comparison

GC <- as.data.frame(as.table(by(DH, DH[1], function(df) cor(df[3],

df[4], use = "GC"))))

#Attribute importance

cat("\n#Decision Tree Attribute Importance\n")

print(summary(GC.tree))

cat("\n#Baging Attribute Importance\n")

print(GC.bag$importance)

cat("\n#Boosting Attribute Importance\n")

print(GC.Boost$importance)

cat("\n#Random Forest Attribute Importance\n")

print(GC.rf$importance)

#pruning