### Refer to this dataset - "germancredit with Yes and No Default values" on sub google drive link

setwd(choose.dir())

GermanCredit = read.csv("Germancredit with Yes and No Default values.csv")

View(GermanCredit)

# Exploratory data analysis

cdplot(Default~history,GermanCredit)

# Build Logistic regression model

Default\_Log = glm(Default~.,GermanCredit,family="binomial")

summary(Default\_Log)

Default\_Log = glm(Default~checkingstatus1+duration+history+purpose+amount+savings+installment+status+others+otherplans+foreign

,GermanCredit,family="binomial")

summary(Default\_Log)

# predictions

PredictedValueus = predict(Default\_Log,GermanCredit,type="response")

View(PredictedValueus)

FinalPredictedDefaultData = cbind(GermanCredit,PredictedValueus)

View(FinalPredictedDefaultData)

FinalPredictedDefaultData1 = transform(FinalPredictedDefaultData,

PredBinary = ifelse(PredictedValueus>.5,"Yes","No"))

View(FinalPredictedDefaultData1)

ncol(FinalPredictedDefaultData1)

View(FinalPredictedDefaultData1[,c(2:23,1)])

table(FinalPredictedDefaultData1$Default,FinalPredictedDefaultData1$PredBinary)

## Accuracy rate at 50% cut off is 78.4%

###### Defining cut off to 60%

FinalPredictedDefaultData1 = transform(FinalPredictedDefaultData,

PredBinary = ifelse(PredictedValueus>.6,"Yes","No"))

View(FinalPredictedDefaultData1)

ncol(FinalPredictedDefaultData1)

View(FinalPredictedDefaultData1[,c(2:23,1)])

table(FinalPredictedDefaultData1$Default,FinalPredictedDefaultData1$PredBinary)

## Accuracy rate at 60% cut off is 77.8%

###### Defining cut off to 40%

FinalPredictedDefaultData1 = transform(FinalPredictedDefaultData,

PredBinary = ifelse(PredictedValueus>.4,"Yes","No"))

View(FinalPredictedDefaultData1)

ncol(FinalPredictedDefaultData1)

View(FinalPredictedDefaultData1[,c(2:23,1)])

table(FinalPredictedDefaultData1$Default,FinalPredictedDefaultData1$PredBinary)

## Accuracy rate at 40% cut off is 76.4%

## Freeze the cut offa at 50% cut off as accuracy rate is higest is 78.4%

for(i in seq(from=.1,to=1,by=.1))

{

print(i)

}

####### Run logistic regression on data "Demo 1\_ Support Vector Machines.csv"

ChurnData = read.csv('Demo 1\_ Support Vector Machines.csv')

str(ChurnData)

ChurnData = transform(ChurnData,Churn = ifelse(Churn==1,"Yes","No"))

View(ChurnData)

str(ChurnData)

ChurnData$Churn = as.factor(ChurnData$Churn)

GlmModel = glm(Churn~DataPlan+ContractRenewal+DataUsage+CustServCalls+MonthlyCharge+RoamMins,

ChurnData,family="binomial")

summary(GlmModel)

# Support vector machine algorithm

install.packages("e1071",dependencies = T)

library(e1071)

Default\_svm = svm(Default~.,GermanCredit,kernel="linear")

PredictedSVM = predict(Default\_svm,GermanCredit,type="class")

View(PredictedSVM)

FinalSVMForecasted = cbind(GermanCredit,PredictedSVM)

View(FinalSVMForecasted)

table(FinalSVMForecasted$Default,FinalSVMForecasted$PredictedSVM)

# Accuracy rate is 78.1%

?svm

# Using Sigmoid function for SVM

Default\_svm = svm(Default~.,GermanCredit,kernel="sigmoid")

PredictedSVM = predict(Default\_svm,GermanCredit,type="class")

View(PredictedSVM)

FinalSVMForecasted = cbind(GermanCredit,PredictedSVM)

View(FinalSVMForecasted)

table(FinalSVMForecasted$Default,FinalSVMForecasted$PredictedSVM)

# Accuray rate is ~73%

### For now we will choose Logistic regression as prediction model and not SVM

## Naive Bayes theorem

library(e1071)

NaivesBayesClass = naiveBayes(Default~.,GermanCredit)

PredicetedValues = predict(NaivesBayesClass,GermanCredit)

### Decision Tree

install.packages("rpart",dependencies = T)

library(rpart)

DT = rpart(Default~.,GermanCredit)

plot(DT,margin=.1)

text(DT,cex=1)

PredicetedValues = predict(DT,GermanCredit,type="class")

View(PredicetedValues)

FinalPredicted = cbind(GermanCredit,PredicetedValues)

table(FinalPredicted$Default,FinalPredicted$PredicetedValues)

### Accuracy rate is 79.7

### DT is better than logistic and we will choose DT as final model for NOW

library(randomForest)

RF = randomForest(Default~.,GermanCredit)

PredictedVlaues = predict(RF,GermanCredit)

FinalPredicted=cbind(GermanCredit,PredictedVlaues)

table(FinalPredicted$Default,FinalPredicted$PredictedVlaues)

# 100% accuracy.

### Assignment

library(MASS)

setwd(choose.dir())

write.csv(Boston,"Boston Exported.csv")

churn\_data = read.csv("Demo 1\_ Support Vector Machines.csv")

View(churn\_data)

library(caTools)

log\_vec = sample.split(churn\_data$Churn,SplitRatio = .7)

log\_vec

ChurnDataTraining = churn\_data[log\_vec,]

ChurnDataTest = churn\_data[!log\_vec,]

#View(ChurnDataTest)

library(randomForest)

RF = randomForest(Churn~.,ChurnDataTraining)

PredictedVlaues = predict(RF,ChurnDataTest)

View(PredictedVlaues)

View(ChurnDataTest)

FinalPredicted=cbind(ChurnDataTest,PredictedVlaues,type="class")

View(FinalPredicted)

final\_table = table(FinalPredicted$Churn,FinalPredicted$PredictedVlaues)

final\_table

# accuracy rate is 92.8%