PROJECT CENSUS DATA

## Data Import ##

CensusData <- read.csv("C:/Users/user/Downloads/CensusData.csv")

View(CensusData)

CensusData[CensusData == ' ?'] = NA

CensusData = na.omit(CensusData)

summary(CensusData)

## Data Exploration ##

bp1 = boxplot(CensusData$age ~ CensusData$Income)

bp4 = boxplot(CensusData$fnlwgt ~ CensusData$Income)

bp2 = boxplot(CensusData$education.num ~ CensusData$Income)

bp3 = boxplot(CensusData$hours.per.week ~ CensusData$Income)

## Replace Outliers ##

for(i in 1:nrow(CensusData)){

if(CensusData[i,"age"]>73)

CensusData[i,"age"]=73

}

Mean<-mean(CensusData$fnlwgt)

SD<-sd(CensusData$fnlwgt)

MinLimit<-Mean-2\*SD

MaxLimit<-Mean+2\*SD

for(i in 1:nrow(CensusData)){

if(CensusData[i,"fnlwgt"]>=MaxLimit){

CensusData[i,"fnlwgt"]<-MaxLimit

}

if(CensusData[i,"fnlwgt"]<=MinLimit){

CensusData[i,"fnlwgt"]<-MinLimit

}

}

Mean<-mean(CensusData$education.num)

SD<-sd(CensusData$education.num)

MinLimit<-Mean-2\*SD

MaxLimit<-Mean+2\*SD

for(i in 1:nrow(CensusData)){

if(CensusData[i,"education.num"]>=MaxLimit){

CensusData[i,"education.num"]<-MaxLimit

}

if(CensusData[i,"education.num"]<=MinLimit){

CensusData[i,"education.num"]<-MinLimit

}

}

Mean<-mean(CensusData$hours.per.week)

SD<-sd(CensusData$hours.per.week)

MinLimit<-Mean-2\*SD

MaxLimit<-Mean+2\*SD

for(i in 1:nrow(CensusData)){

if(CensusData[i,"hours.per.week"]>=MaxLimit){

CensusData[i,"hours.per.week"]<-MaxLimit

}

if(CensusData[i,"hours.per.week"]<=MinLimit){

CensusData[i,"hours.per.week"]<-MinLimit

}

}

## Stratified Sampling ##

mydata\_0 = CensusData[(CensusData$Income == ' <=50K'), ]

mydata\_1 = CensusData[(CensusData$Income == ' >50K'), ]

train\_row\_0 = round(nrow(mydata\_0)\*0.8,0)

train\_row\_1 = round(nrow(mydata\_1)\*0.8,0)

set.seed(2)

index\_train0 = sample(1:nrow(mydata\_0), train\_row\_0, replace=FALSE)

index\_train1 = sample(1:nrow(mydata\_1), train\_row\_1, replace=FALSE)

train\_mydata\_0 = mydata\_0[index\_train0,]

train\_mydata\_1 = mydata\_1[index\_train1,]

train\_mydata = rbind(train\_mydata\_0, train\_mydata\_1)

test\_mydata\_0 = mydata\_0[-index\_train0,]

test\_mydata\_1 = mydata\_1[-index\_train1,]

test\_mydata = rbind(test\_mydata\_0, test\_mydata\_1)

table(CensusData$Income)

table(train\_mydata$Income)

table(test\_mydata$Income)

## Decision Tree Model ##

library(ROCR)

library("rpart")

mytree = rpart(Income~ age+workclass+fnlwgt+education+education.num+marital.status+occupation+relationship+race+sex+capital.gain+capital.loss+hours.per.week+native.country, data=train\_mydata)

pred\_tree = predict(mytree, test\_mydata, type="class")

combined\_tree = cbind(test\_mydata$Income, pred\_tree)

colnames(combined\_tree)= c("Actual", "Predicted")

df\_combined\_tree = data.frame(combined\_tree)

table(df\_combined\_tree$Actual, df\_combined$Predicted)

preds\_tree = prediction(as.numeric(pred\_tree), as.numeric(test\_mydata$Income))

perf\_tree = performance(preds\_tree, "tpr", "fpr")

plot(perf\_tree)

## Logistic Regression ##

mylogit = glm(Income~ age+workclass+fnlwgt+education+education.num+marital.status+occupation+relationship+race+sex+capital.gain+capital.loss+hours.per.week+native.country, data=train\_mydata, family="binomial")

summary(mylogit)

pred = predict(mylogit, test\_mydata, type="response")

pred1 = round(pred, 0)

combined = cbind(test\_mydata$Income, pred1)

colnames(combined)= c("Actual", "Predicted")

df\_combined = data.frame(combined)

table(df\_combined$Actual, df\_combined$Predicted)

preds = prediction(as.numeric(pred), as.numeric(test\_mydata$Income))

perf = performance(preds, "tpr", "fpr")

plot(perf)