```
---
title: "Qualifying_6"
author: "Chathrua Gunasekara"
date: "December 22, 2014"
output: pdf_document
---

1). Data exploration using visualizations
```{r,echo=FALSE}

rm(list=ls())
data = read.table("http://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data",
        sep=",",header=F,col.names=c("age", "type_employer", "fnlwgt", "education",
                "education_num","marital", "occupation", "relationship", "race","sex",
                "capital_gain", "capital_loss", "hr_per_week","country", "income"),
        fill=FALSE,strip.white=T)


testdata = read.table("http://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data",
        sep=",",header=F,col.names=c("age", "type_employer", "fnlwgt", "education",
                "education_num","marital", "occupation", "relationship", "race","sex",
                "capital_gain", "capital_loss", "hr_per_week","country", "income"),
        fill=FALSE,strip.white=T)


data[["education_num"]]=NULL
data[["fnlwgt"]]=NULL

data$type_employer = as.character(data$type_employer)
data$occupation = as.character(data$occupation)
data$country = as.character(data$country)
data$education = as.character(data$education)
data$race = as.character(data$race)
data$marital = as.character(data$marital)

data$type_employer = gsub("^Federal-gov","Federal-Govt",data$type_employer)
data$type_employer = gsub("^Local-gov","Other-Govt",data$type_employer)
data$type_employer = gsub("^State-gov","Other-Govt",data$type_employer)
data$type_employer = gsub("^Private","Private",data$type_employer)
data$type_employer = gsub("^Self-emp-inc","Self-Employed",data$type_employer)
```
```

```
data$type_employer = gsub("^Self-emp-not-inc","Self-
Employed",data$type_employer)
data$type_employer = gsub("^Without-pay","Not-
Working",data$type_employer)
data$type_employer = gsub("^Never-worked","Not-
Working",data$type_employer)

data$occupation = gsub("^Adm-clerical","Admin",data$occupation)
data$occupation = gsub("^Armed-Forces","Military",data$occupation)
data$occupation = gsub("^Craft-repair","Blue-Collar",data$occupation)
data$occupation = gsub("^Exec-managerial","White-
Collar",data$occupation)
data$occupation = gsub("^Farming-fishing","Blue-Collar",data$occupation)
data$occupation = gsub("^Handlers-cleaners","Blue-
Collar",data$occupation)
data$occupation = gsub("^Machine-op-inspct","Blue-
Collar",data$occupation)
data$occupation = gsub("^Other-service","Service",data$occupation)
data$occupation = gsub("^Priv-house-serv","Service",data$occupation)
data$occupation = gsub("^Prof-specialty","Professional",data$occupation)
data$occupation = gsub("^Protective-serv","Other-
Occupations",data$occupation)
data$occupation = gsub("^Sales","Sales",data$occupation)
data$occupation = gsub("^Tech-support","Other-
Occupations",data$occupation)
data$occupation = gsub("^Transport-moving","Blue-
Collar",data$occupation)

data$country[data$country=="Cambodia"] = "SE-Asia"
data$country[data$country=="Canada"] = "British-Commonwealth"
data$country[data$country=="China"] = "China"
data$country[data$country=="Columbia"] = "South-America"
data$country[data$country=="Cuba"] = "Other"
data$country[data$country=="Dominican-Republic"] = "Latin-America"
data$country[data$country=="Ecuador"] = "South-America"
data$country[data$country=="El-Salvador"] = "South-America"
data$country[data$country=="England"] = "British-Commonwealth"
data$country[data$country=="France"] = "Euro_1"
data$country[data$country=="Germany"] = "Euro_1"
data$country[data$country=="Greece"] = "Euro_2"
data$country[data$country=="Guatemala"] = "Latin-America"
data$country[data$country=="Haiti"] = "Latin-America"
data$country[data$country=="Holand-Netherlands"] = "Euro_1"
data$country[data$country=="Honduras"] = "Latin-America"
data$country[data$country=="Hong"] = "China"
data$country[data$country=="Hungary"] = "Euro_2"
data$country[data$country=="India"] = "British-Commonwealth"
data$country[data$country=="Iran"] = "Other"
data$country[data$country=="Ireland"] = "British-Commonwealth"
data$country[data$country=="Italy"] = "Euro_1"
```

```
data$country[data$country=="Jamaica"] = "Latin-America"
data$country[data$country=="Japan"] = "Other"
data$country[data$country=="Laos"] = "SE-Asia"
data$country[data$country=="Mexico"] = "Latin-America"
data$country[data$country=="Nicaragua"] = "Latin-America"
data$country[data$country=="Outlying-US(Guam-USVI-etc)"] = "Latin-
America"
data$country[data$country=="Peru"] = "South-America"
data$country[data$country=="Philippines"] = "SE-Asia"
data$country[data$country=="Poland"] = "Euro_2"
data$country[data$country=="Portugal"] = "Euro_2"
data$country[data$country=="Puerto-Rico"] = "Latin-America"
data$country[data$country=="Scotland"] = "British-Commonwealth"
data$country[data$country=="South"] = "Euro_2"
data$country[data$country=="Taiwan"] = "China"
data$country[data$country=="Thailand"] = "SE-Asia"
data$country[data$country=="Trinadad&Tobago"] = "Latin-America"
data$country[data$country=="United-States"] = "United-States"
data$country[data$country=="Vietnam"] = "SE-Asia"
data$country[data$country=="Yugoslavia"] = "Euro_2"

data$education = gsub("^10th","Dropout",data$education)
data$education = gsub("^11th","Dropout",data$education)
data$education = gsub("^12th","Dropout",data$education)
data$education = gsub("^1st-4th","Dropout",data$education)
data$education = gsub("^5th-6th","Dropout",data$education)
data$education = gsub("^7th-8th","Dropout",data$education)
data$education = gsub("^9th","Dropout",data$education)
data$education = gsub("^Assoc-acdm","Associates",data$education)
data$education = gsub("^Assoc-voc","Associates",data$education)
data$education = gsub("^Bachelors","Bachelors",data$education)
data$education = gsub("^Doctorate","Doctorate",data$education)
data$education = gsub("^HS-Grad","HS-Graduate",data$education)
data$education = gsub("^Masters","Masters",data$education)
data$education = gsub("^Preschool","Dropout",data$education)
data$education = gsub("^Prof-school","Prof-School",data$education)
data$education = gsub("^Some-college","HS-Graduate",data$education)
data$marital[data$marital=="Never-married"] = "Never-Married"
data$marital[data$marital=="Married-AF-spouse"] = "Married"
data$marital[data$marital=="Married-civ-spouse"] = "Married"
data$marital[data$marital=="Married-spouse-absent"] = "Not-Married"
data$marital[data$marital=="Separated"] = "Not-Married"
data$marital[data$marital=="Divorced"] = "Not-Married"
data$marital[data$marital=="Widowed"] = "Widowed"

data$race[data$race=="White"] = "White"
data$race[data$race=="Black"] = "Black"
data$race[data$race=="Amer-Indian-Eskimo"] = "Amer-Indian"
data$race[data$race=="Asian-Pac-Islander"] = "Asian"
data$race[data$race=="Other"] = "Other"
```

````
data[["capital_gain"]] <- ordered(cut(data$capital_gain,c(-Inf, 0,
            median(data[["capital_gain"]][data[["capital_gain"]] >0]),
            Inf)),labels = c("None", "Low", "High"))
data[["capital_loss"]] <- ordered(cut(data$capital_loss,c(-Inf, 0,
            median(data[["capital_loss"]][data[["capital_loss"]] >0]),
            Inf)), labels = c("None", "Low", "High"))

is.na(data) = data=='?'
is.na(data) = data==' ?'
data = na.omit(data)

data$marital = factor(data$marital)
data$education = factor(data$education)
data$country = factor(data$country)
data$type_employer = factor(data$type_employer)
data$occupation = factor(data$occupation)
data$race = factor(data$race)
data$sex = factor(data$sex)
data$relationship = factor(data$relationship)
data$income = as.factor(ifelse(data$income==data$income[1],0,1))


data$age = scale(data$age)
data$hr_per_week = scale(data$hr_per_week)

dummy <-dummyVars(income~.-age -hr_per_week,data=data)
newdata<-as.data.frame(predict(dummy,data))
newdata<-cbind(newdata,data$age,data$hr_per_week)

#-------------------------------------------
```
1)
```{r,echo=FALSE}
plot(as.factor(testdata$education),testdata$education_num,main="Education
 vs Education Num",xlab="Education level",ylab="Numecial value for
education")
newdata <-newdata[,-nearZeroVar(newdata)]
corrrelation <- cor(newdata)
newdata<-newdata[,-findCorrelation(corrrelation,cutoff=0.75)]
library(corrplot)
corrplot(corrrelation, order = "hclust")

```
2)Outliars
```{r,echo=FALSE}
boxplot(data$age,main="Box plot for age")
boxplot(data$hr_per_week,main="Box plot for hours per week")

```
````
````

````
3)skewness
```{r,echo=FALSE}
library(e1071)
#Capital _gain
boxplot(testdata$capital_gain,main="Box plot for capital gain")
skewness(testdata$capital_gain)
hist(testdata$capital_gain,main="Histogram for Capital gain")
barplot(table(data$capital_gain),main="Capital Gain")
skewness(testdata$capital_loss)
hist(testdata$capital_loss,main="Histogram for Capital Loss")
barplot(table(data$capital_loss),main="Capital Loss")
#Here I block capital gains and losses, rather than do a transformation.
Both variables are heavily skewed to the point that I think a numerical
transformation would not have been appropriate. So I choose to block
them into “None”, “Low”, and “High”. For both variables, none means they
don’t play the market. Low means they have some investments. High means
they have significant investments.
```
4)
```{r,echo=FALSE}
par(mfrow=c(2,2))
barplot(table(data$type_employer),main="type_employer")
barplot(table(data$education),main="education")
barplot(table(data$marital),main="martial")
barplot(table(data$occupation),main="occupaion")
barplot(table(data$relationship),main="relationship")
dev.off()
par(mfrow=c(2,2))
barplot(table(data$race),main="race")
barplot(table(data$sex),main="sex")
barplot(table(data$capital_gain),main="capital gain")
barplot(table(data$capital_loss),main="capital loss")
dev.off()
par(mfrow=c(1,2))
barplot(table(data$country),main="country")
barplot(table(data$income),main="income")

data <-data[,-nearZeroVar(data)]



```
5)
```{r,echo=FALSE}
dev.off()
testdata[ testdata == "?" ] = NA

image(is.na(testdata), main = "Missing Values", xlab = "Observation",
ylab = "Variable",xaxt = "n", yaxt = "n", bty = "n")
axis(1, seq(0, 1, length.out = nrow(testdata)), 1:nrow(testdata), col =
````
````

````
"white")
testdata[,15]

table(testdata$type_employer)
table(testdata$education)
table(testdata$marital)
table(testdata$occupation)
table(testdata$relationship)
table(testdata$race)
table(testdata$sex)
table(testdata$country)

```
6)
```{r,echo=FALSE}
#removed the missing data, still there are over 30000
```

7)
```{r,echo=FALSE}

# AUC of ROC will be used as a classification statistic.

```
8)
```{r,echo=FALSE}
barplot(table(data$income),main="Predictor frequncy distribution")
#unbalnced



```
9)
```{r,echo=FALSE}
#describe the preprocessing
#13 predictors are left
```
10)
```{r,echo=FALSE}
dummy <-dummyVars(income~.-age -hr_per_week -capital_gain -
capital_loss,data=data)
dummy_data<-predict(dummy,data)
dummy_data<-as.data.frame(dummy_data)

trainIndex <- createDataPartition(data$income, p = .75,list =
FALSE,times = 1)
y<-data$income
x<-data[,-10]
trainX<-x[trainIndex,]
trainY<-factor(y[trainIndex])
````

````
testX<- x[-trainIndex,]
testY<-factor(y[-trainIndex])

dummy <-dummyVars(income~.-age -hr_per_week -capital_gain -
capital_loss,data=x)
dummy <-dummyVars(income~.,data=data)
dummy_data<-predict(dummy,data)
dummy_data<-as.data.frame(dummy_data)

xD<-dummy_data[,-40]
xD<-xD[,-nearZeroVar(xD)]
trainXD<-xD[trainIndex,]
testXD<- xD[-trainIndex,]

```
Logistic Regression
```{r,echo=FALSE}
ctrl <- trainControl(method = "LGOCV",
                     summaryFunction = twoClassSummary,
                     classProbs = TRUE,
                     savePredictions = TRUE)
set.seed(476)
lrModel<- train(trainX,y = trainY,method = "glm",metric =
"ROC",trControl = ctrl)
lrModel
#confusionMatrix(predict(lrModel,testX),testY,positive="1")
```
LDA
```{r,echo=FALSE}

ldaModel<- train(x = trainXD,y = as.factor(trainY),method = "lda",metric
= "ROC",trControl = ctrl)
#confusionMatrix(predict(ldaModel,testX),testY,positive ="1")
ldaModel
```
PLSDA
```{r,echo=FALSE}
library(pls)


plsFit <- train(x = trainXD,y = trainY,method = "pls",tuneGrid =
expand.grid(.ncomp = 1:10),metric = "ROC",trControl = ctrl)
plsFit
plsImpGrant <- varImp(plsFit, scale = FALSE)
plsImpGrant
```
Penalized Models - GLMnet
```{r,echo=FALSE}
glmnGrid <- expand.grid(.alpha = c(0,.2, .6, .8),.lambda=c(.1, .2,
length = 20))
````

~~~~
glmnTuned <- train(trainXD,y = trainY,method = "glmnet",tuneGrid =
glmnGrid,metric = "ROC",trControl = ctrl)
glmnTuned
```

Nearest Shrunken Centrods
```{r,echo=FALSE}
nscGrid <- data.frame(.threshold = 0:25)
set.seed(476)
nscTuned <- train(x = trainXD,y = trainY,method = "pam",preProc =
c("center", "scale"),tuneGrid = nscGrid,metric = "ROC",trControl = ctrl)
nscTuned
varImp(nscTuned, scale = FALSE)
```
11) Non Linear models

Non linear discriminant analysis
```{r,echo=FALSE}


set.seed(476)

mdaFit <- train(trainX, trainY,method = "mda",metric = "ROC",tuneGrid =
expand.grid(.subclasses = 1:8),trControl = ctrl)
mdaFit
```

NeuralNet
```{r,echo=FALSE}

nnetGrid <- expand.grid(.size = 1:5,.decay = c(0,0.001,0.1,1))
maxSize <- max(nnetGrid$.size)
numWts <- 1*(maxSize * (9 + 1) + maxSize + 1)
set.seed(476)
nnetFit <- train(x = trainX,y = trainY,
                 method = "nnet",
                  metric = "ROC",
                 tuneGrid = nnetGrid,
                 trace = FALSE,
                 maxit = 10000,
                 MaxNWts = numWts,
                 ## ctrl was defined in the previous chapter
                  trControl = ctrl)

nnetFit



```
FDA
~~~~

````
```{r,echo=FALSE}
fdaFit <- train(trainX, trainY,method = "fda",metric = "ROC",trControl =
ctrl)
fdaFit
plsImp <- varImp(fdaFit,scale=FALSE)
```
SVM
```{r,echo=FALSE}
library(e1071)
library(kernlab)
library(klaR)
set.seed(202)

sigmaRange <- sigest(as.matrix(trainXD),na.action = na.omit)
svmRGrid <- expand.grid(.sigma = sigmaRange[1],.C = 2^(seq(-2, 2)))
set.seed(476)
svmRModel <- train(trainXD, trainY,
                    method = "svmRadial",
                    metric = "ROC",
                    tuneGrid = svmRGrid,
                    fit = FALSE,
                    trControl = ctrl)

svmRModel
varImp(svmRModel, scale = FALSE)
```
KNN
```{r,echo=FALSE}
set.seed(476)
knnFit <- train(trainXD, trainY,
                   method = "knn",
                   metric = "ROC",
                   tuneGrid = data.frame(.k = c(3,5,9)),
                   trControl = ctrl)
knnFit
```

NaiveBayes
```{r,echo=FALSE}
library(klaR)
ctrl1 <- trainControl(
                     summaryFunction = twoClassSummary,
                     classProbs = TRUE,
                     savePredictions = TRUE)
nb<-naiveBayes(trainX, trainY)
NBFit <- train(trainX, trainY,method = "nb",metric = "ROC",trControl =
ctrl1)

confusionMatrix(predict(nb, testX),testY)
```
````

13)

````
```{r,echo=FALSE}

confusionMatrix(predict(lrModel, testX),testY)

confusionMatrix(predict(ldaModel, testXD),testY)

confusionMatrix(predict(plsFit, testXD),testY)

confusionMatrix(predict(nnetFit, testX),testY)

confusionMatrix(predict(knnFit, testX),testY)

confusionMatrix(predict(svmRModel, testX),testY)

confusionMatrix(predict(lrModel, testX),testY)
```
````