Project 2 Classification

Ashish Lamichhane

Prediction of Income Level based on Age, Race, Sex and Education (and other predictors)

Reading Data into R

Citation : Dua, D. and Graff, C. (2019). UCI Machine Learning Repository [http://archive.ics.uci.edu/ml]. Irvine, CA: University of California, School of Information and Computer Science. Link of the data : https://archive.ics.uci.edu/ml/datasets/Adult

I used the parameter na.strings="NA" to tell R to fill missing cells with NA. This data is a bit messy because some things that should be factors, like Income or Sex or Race, are not. If we had not used the stringsAsFactors=FALSE parameter, strings like Occupation or Native country would be encoded as factors. I have made some changes to the data set after importing. For instance, I am converting Income to factor using as.factor().

```
#reading the data into R. First row contains variable names and comma is separator.
df <- read.table("adult.csv", na.strings = "NA", stringsAsFactors = FALSE, header = TRUE, strip.white =
```

Data Exploration Functions (Before data cleaning)

```
names(df) # lists the column names.
```

```
##  [1] "Age"            "Work_Class"     "fnlwgt"         "Education"
##  [5] "Education_num"  "Marital_Status" "Occupation"     "Relationship"
##  [9] "Race"           "Sex"            "Capital_gain"   "Capital_loss"
## [13] "Hours_per_week" "Native_Country" "Income"
```

```
head(df, n = 10) #see first 10 rows.
```

```
##    Age       Work_Class fnlwgt Education Education_num
## 1   39        State-gov  77516 Bachelors            13
## 2   50 Self-emp-not-inc  83311 Bachelors            13
## 3   38          Private 215646   HS-grad             9
## 4   53          Private 234721      11th             7
## 5   28          Private 338409 Bachelors            13
## 6   37          Private 284582   Masters            14
## 7   49          Private 160187       9th             5
## 8   52 Self-emp-not-inc 209642   HS-grad             9
## 9   31          Private  45781   Masters            14
## 10  42          Private 159449 Bachelors            13
##           Marital_Status        Occupation  Relationship  Race    Sex
## 1          Never-married      Adm-clerical Not-in-family White   Male
## 2     Married-civ-spouse   Exec-managerial       Husband White   Male
```

```
## 3               Divorced Handlers-cleaners Not-in-family White   Male
## 4     Married-civ-spouse Handlers-cleaners       Husband Black   Male
## 5     Married-civ-spouse    Prof-specialty          Wife Black Female
## 6     Married-civ-spouse   Exec-managerial          Wife White Female
## 7  Married-spouse-absent     Other-service Not-in-family Black Female
## 8     Married-civ-spouse   Exec-managerial       Husband White   Male
## 9          Never-married    Prof-specialty Not-in-family White Female
## 10    Married-civ-spouse   Exec-managerial       Husband White   Male
##    Capital_gain Capital_loss Hours_per_week Native_Country Income
## 1          2174            0             40  United-States      0
## 2             0            0             13  United-States      0
## 3             0            0             40  United-States      0
## 4             0            0             40  United-States      0
## 5             0            0             40           Cuba      0
## 6             0            0             40  United-States      0
## 7             0            0             16        Jamaica      0
## 8             0            0             45  United-States      1
## 9         14084            0             50  United-States      1
## 10         5178            0             40  United-States      1
```

```r
tail(df, n = 5) # see last 5 rows.
```

```
##       Age   Work_Class fnlwgt Education Education_num     Marital_Status
## 48838  39      Private 215419 Bachelors            13           Divorced
## 48839  64         <NA> 321403   HS-grad             9            Widowed
## 48840  38      Private 374983 Bachelors            13 Married-civ-spouse
## 48841  44      Private  83891 Bachelors            13           Divorced
## 48842  35 Self-emp-inc 182148 Bachelors            13 Married-civ-spouse
##            Occupation   Relationship               Race    Sex
## 48838  Prof-specialty  Not-in-family              White Female
## 48839            <NA> Other-relative              Black   Male
## 48840  Prof-specialty        Husband              White   Male
## 48841    Adm-clerical      Own-child Asian-Pac-Islander   Male
## 48842 Exec-managerial        Husband              White   Male
##       Capital_gain Capital_loss Hours_per_week Native_Country Income
## 48838            0            0             36  United-States      0
## 48839            0            0             40  United-States      0
## 48840            0            0             50  United-States      0
## 48841         5455            0             40  United-States      0
## 48842            0            0             60  United-States      1
```

```r
str(df) #finding the structure of the data set.
```

```
## 'data.frame':    48842 obs. of  15 variables:
##  $ Age           : int  39 50 38 53 28 37 49 52 31 42 ...
##  $ Work_Class    : chr  "State-gov" "Self-emp-not-inc" "Private" "Private" ...
##  $ fnlwgt        : int  77516 83311 215646 234721 338409 284582 160187 209642 45781 159449 ...
##  $ Education     : chr  "Bachelors" "Bachelors" "HS-grad" "11th" ...
##  $ Education_num : int  13 13 9 7 13 14 5 9 14 13 ...
##  $ Marital_Status: chr  "Never-married" "Married-civ-spouse" "Divorced" "Married-civ-spouse" ...
##  $ Occupation    : chr  "Adm-clerical" "Exec-managerial" "Handlers-cleaners" "Handlers-cleaners" ...
##  $ Relationship  : chr  "Not-in-family" "Husband" "Not-in-family" "Husband" ...
##  $ Race          : chr  "White" "White" "White" "Black" ...
```

```
##  $ Sex           : chr  "Male" "Male" "Male" "Male" ...
##  $ Capital_gain  : int  2174 0 0 0 0 0 0 0 14084 5178 ...
##  $ Capital_loss  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Hours_per_week: int  40 13 40 40 40 40 16 45 50 40 ...
##  $ Native_Country: chr  "United-States" "United-States" "United-States" "United-States" ...
##  $ Income        : int  0 0 0 0 0 0 0 1 1 1 ...
```

```
summary(df) # summary() function provides a number of useful statistics including range, median, and me
```

```
##       Age         Work_Class            fnlwgt          Education        
##  Min.   :17.00   Length:48842       Min.   :  12285   Length:48842      
##  1st Qu.:28.00   Class :character   1st Qu.: 117550   Class :character  
##  Median :37.00   Mode  :character   Median : 178144   Mode  :character  
##  Mean   :38.64                      Mean   : 189664                     
##  3rd Qu.:48.00                      3rd Qu.: 237642                     
##  Max.   :90.00                      Max.   :1490400                     
##  Education_num   Marital_Status      Occupation        Relationship      
##  Min.   : 1.00   Length:48842       Length:48842       Length:48842      
##  1st Qu.: 9.00   Class :character   Class :character   Class :character  
##  Median :10.00   Mode  :character   Mode  :character   Mode  :character  
##  Mean   :10.08                                                           
##  3rd Qu.:12.00                                                           
##  Max.   :16.00                                                           
##      Race               Sex             Capital_gain    Capital_loss   
##  Length:48842       Length:48842       Min.   :    0   Min.   :   0.0  
##  Class :character   Class :character   1st Qu.:    0   1st Qu.:   0.0  
##  Mode  :character   Mode  :character   Median :    0   Median :   0.0  
##                                        Mean   : 1079   Mean   :  87.5  
##                                        3rd Qu.:    0   3rd Qu.:   0.0  
##                                        Max.   :99999   Max.   :4356.0  
##  Hours_per_week  Native_Country         Income      
##  Min.   : 1.00   Length:48842       Min.   :0.0000  
##  1st Qu.:40.00   Class :character   1st Qu.:0.0000  
##  Median :40.00   Mode  :character   Median :0.0000  
##  Mean   :40.42                      Mean   :0.2393  
##  3rd Qu.:45.00                      3rd Qu.:0.0000  
##  Max.   :99.00                      Max.   :1.0000
```

```
dim(df) #gives the row, col dimensions
```

```
## [1] 48842    15
```

```
sapply(df, function(x) sum(is.na(x))) #checking # of NAs per column
```

```
##            Age     Work_Class         fnlwgt      Education  Education_num 
##              0           2799              0              0              0 
## Marital_Status     Occupation   Relationship           Race            Sex 
##              0           2809              0              0              0 
##   Capital_gain   Capital_loss Hours_per_week Native_Country         Income 
##              0              0              0            857              0
```

Data Cleaning Process

The original data set had Income level as <=50K(0) and >50K(1). It was converted to 0 and 1 respectively because reading the input as character and converting it to a factor created 4 levels whereas, only 2 levels were needed. Amelia library was installed to check the graph of missing values vs observed values. We saw that there was only 1% missing values. Work_Class and Occupation had alot of missing values. We will discard those columns. We will also not use Education because there is another column called Education_num that specifies number to those columns. Similary, Marital_status, Relationship, Occupation is also discarded becuase those are character inputs and couldn't be used for logistic regression. By using the subset function we are selecting only releveant columns (9 columns). Sex and Race are both converted to contain numeric values rather than characters. The correlation between numeric columns is then checked and the findCorrelation() function suggested that there was no correlation among those columns.

```r
library(Amelia)
```

```
## Loading required package: Rcpp
```

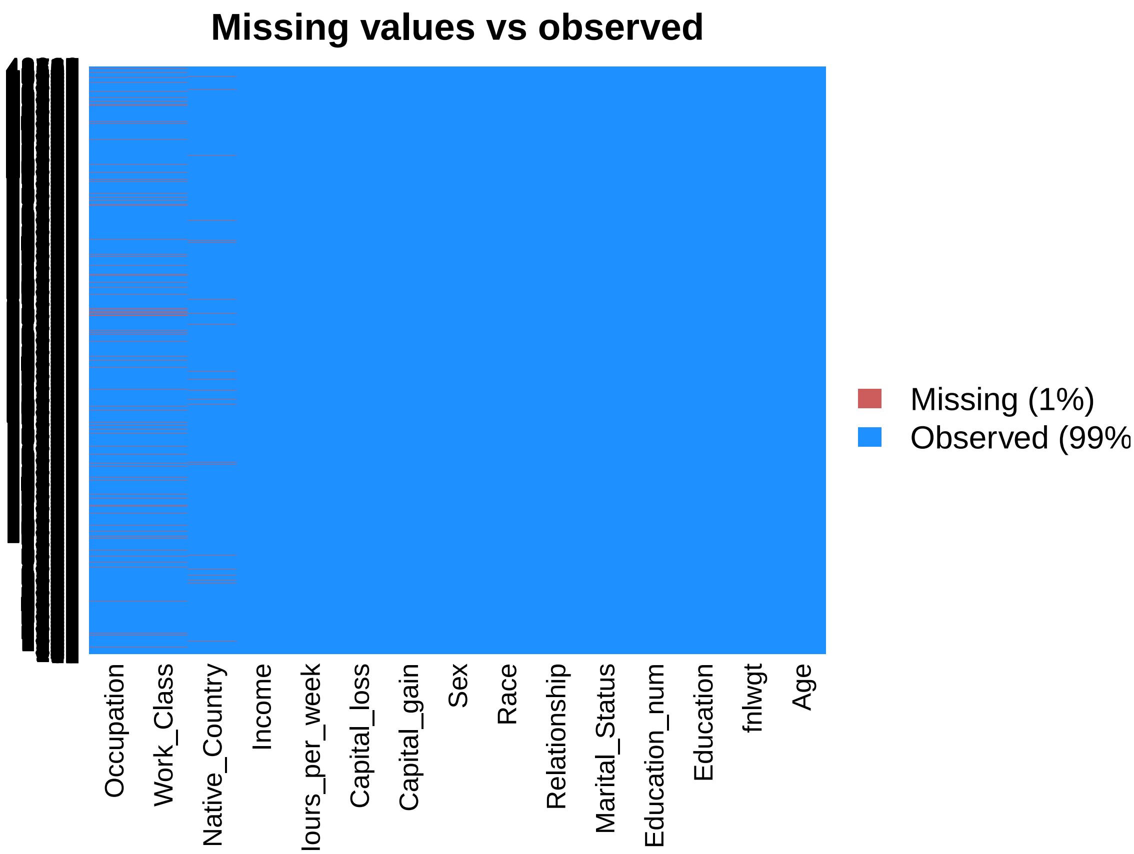
```
## ##
## ## Amelia II: Multiple Imputation
## ## (Version 1.7.5, built: 2018-05-07)
## ## Copyright (C) 2005-2019 James Honaker, Gary King and Matthew Blackwell
## ## Refer to http://gking.harvard.edu/amelia/ for more information
## ##
```

```r
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```r
missmap(df, main = "Missing values vs observed")
```




```r
data <- subset(df,select=c(1,5,9:13,15))
attach(data)

sapply(data, function(x) sum(is.na(x))) #double checking # of NAs per column
```

```
##            Age  Education_num           Race            Sex   Capital_gain
##              0              0              0              0              0
##   Capital_loss Hours_per_week         Income
##              0              0              0
```

```r
data$Income <- as.factor(data$Income) # we are predicting the income so it is converted to a factor.
levels(data$Income) #checking levels.
```

```
## [1] "0" "1"
```

```r
contrasts(data$Income) #checking encodings.
```

```
##   1
## 0 0
## 1 1
```

```r
#Converting Sex to a numeric data.
data$Sex <- ifelse(data$Sex == "Male",1,0)
## Converting Race to numeric data as well.
```

```r
raceType <- c("Amer-Indian-Eskimo" = 0, "Asian-Pac-Islander" = 1, "Black" = 2, "White" = 3, "Other" = 4)
data$Race <- as.numeric(raceType[data$Race])
# The findCorrelation() function suggests that there is no co-relation among any of the columns tested.
corMatrix <- cor(data[,c(1:7)])
findCorrelation(corMatrix, cutoff=0.5, verbose=TRUE)
```

```
## All correlations <= 0.5
```

```
## integer(0)
```

Data Exploration Functions (Applied on selected subset of the original data)

```r
names(data) # lists the column names.
```

```
## [1] "Age"            "Education_num"  "Race"           "Sex"
## [5] "Capital_gain"   "Capital_loss"   "Hours_per_week" "Income"
```

```r
head(data, n = 10) #see first 10 rows.
```

```
##    Age Education_num Race Sex Capital_gain Capital_loss Hours_per_week
## 1   39            13    3   1         2174            0             40
## 2   50            13    3   1            0            0             13
## 3   38             9    3   1            0            0             40
## 4   53             7    2   1            0            0             40
## 5   28            13    2   0            0            0             40
## 6   37            14    3   0            0            0             40
## 7   49             5    2   0            0            0             16
## 8   52             9    3   1            0            0             45
## 9   31            14    3   0        14084            0             50
## 10  42            13    3   1         5178            0             40
##    Income
## 1       0
## 2       0
## 3       0
## 4       0
## 5       0
## 6       0
## 7       0
## 8       1
## 9       1
## 10      1
```

```r
tail(data, n = 5) # see last 5 rows.
```

```
##       Age Education_num Race Sex Capital_gain Capital_loss Hours_per_week
## 48838  39            13    3   0            0            0             36
## 48839  64             9    2   1            0            0             40
```

```
## 48840  38            13    3   1            0            0              50
## 48841  44            13    1   1         5455            0              40
## 48842  35            13    3   1            0            0              60
##       Income
## 48838      0
## 48839      0
## 48840      0
## 48841      0
## 48842      1
```

```r
str(data) #finding the structure of the data set.
```

```
## 'data.frame':    48842 obs. of  8 variables:
##  $ Age           : int  39 50 38 53 28 37 49 52 31 42 ...
##  $ Education_num : int  13 13 9 7 13 14 5 9 14 13 ...
##  $ Race          : num  3 3 3 2 2 3 2 3 3 3 ...
##  $ Sex           : num  1 1 1 1 0 0 0 1 0 1 ...
##  $ Capital_gain  : int  2174 0 0 0 0 0 0 0 14084 5178 ...
##  $ Capital_loss  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Hours_per_week: int  40 13 40 40 40 40 16 45 50 40 ...
##  $ Income        : Factor w/ 2 levels "0","1": 1 1 1 1 1 1 1 2 2 2 ...
```

```r
summary(data) # summary() function provides a number of useful statistics including range, median, and 
```

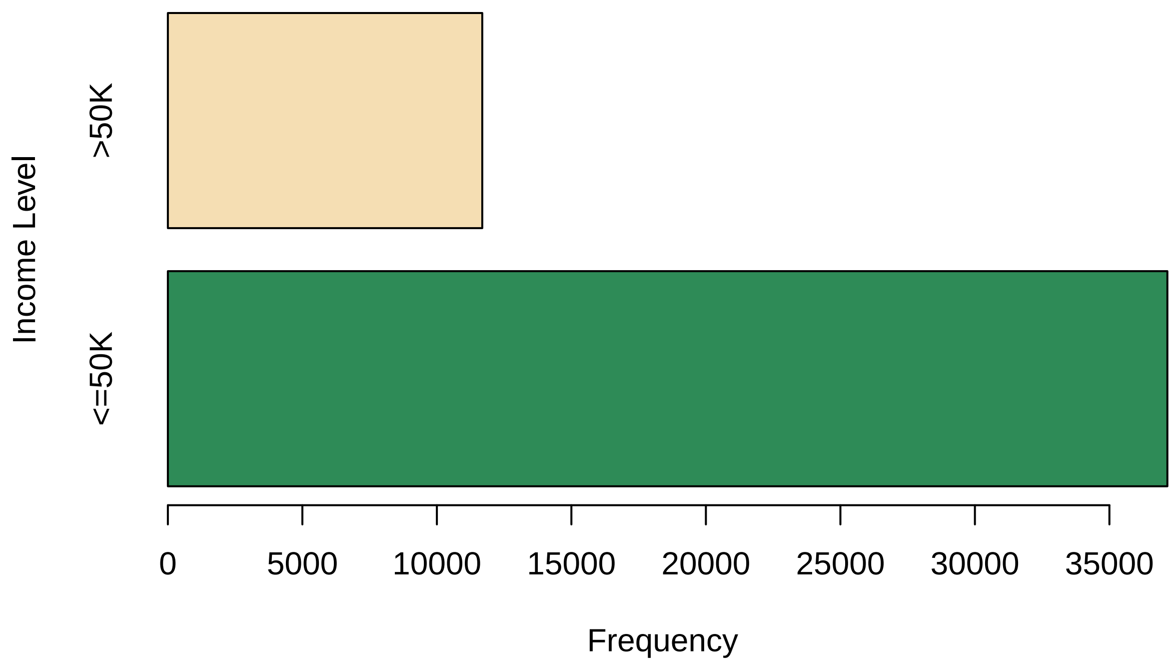
```
##       Age        Education_num        Race            Sex        
##  Min.   :17.00   Min.   : 1.00   Min.   :0.000   Min.   :0.0000  
##  1st Qu.:28.00   1st Qu.: 9.00   1st Qu.:3.000   1st Qu.:0.0000  
##  Median :37.00   Median :10.00   Median :3.000   Median :1.0000  
##  Mean   :38.64   Mean   :10.08   Mean   :2.821   Mean   :0.6685  
##  3rd Qu.:48.00   3rd Qu.:12.00   3rd Qu.:3.000   3rd Qu.:1.0000  
##  Max.   :90.00   Max.   :16.00   Max.   :4.000   Max.   :1.0000  
##   Capital_gain    Capital_loss    Hours_per_week  Income   
##  Min.   :    0   Min.   :   0.0   Min.   : 1.00   0:37155  
##  1st Qu.:    0   1st Qu.:   0.0   1st Qu.:40.00   1:11687  
##  Median :    0   Median :   0.0   Median :40.00            
##  Mean   : 1079   Mean   :  87.5   Mean   :40.42            
##  3rd Qu.:    0   3rd Qu.:   0.0   3rd Qu.:45.00            
##  Max.   :99999   Max.   :4356.0   Max.   :99.00            
```

```r
dim(data) #gives the row, col dimensions
```

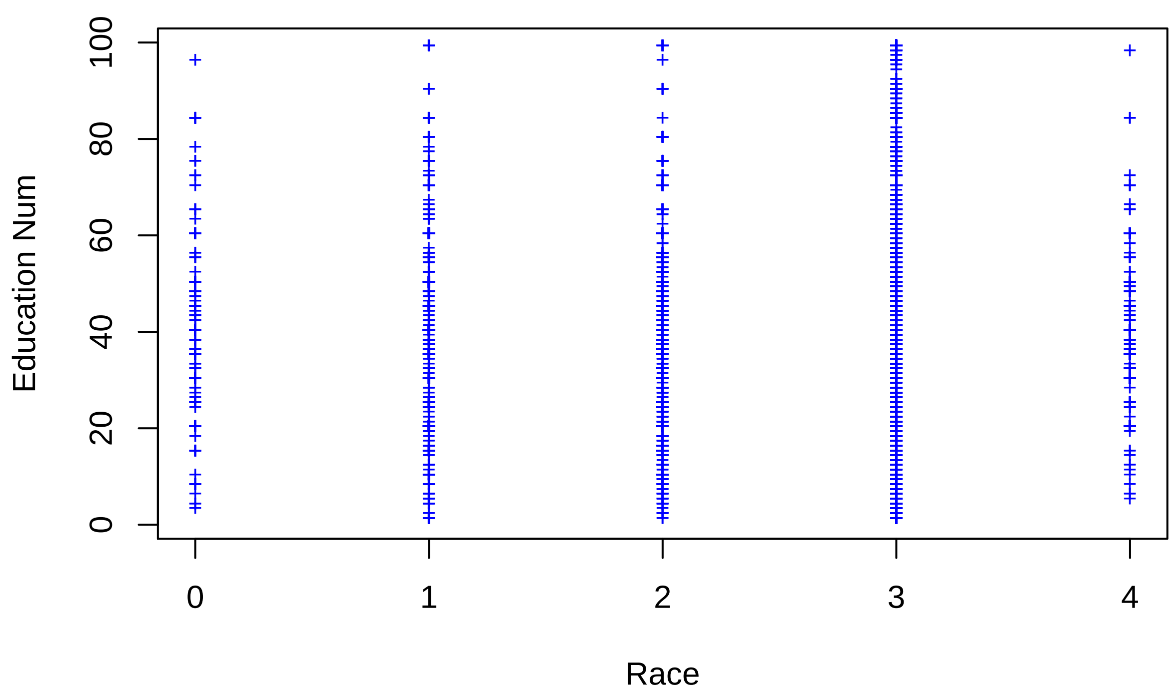
```
## [1] 48842     8
```

Visual Data Exploration

```r
#Plotting appearances ( or count ) of two Income Levels.
counts <- table(data$Income)
barplot(counts, horiz=TRUE, names=c("<=50K", ">50K"), col=c("seagreen","wheat"), ylab="Income Level", xl
```

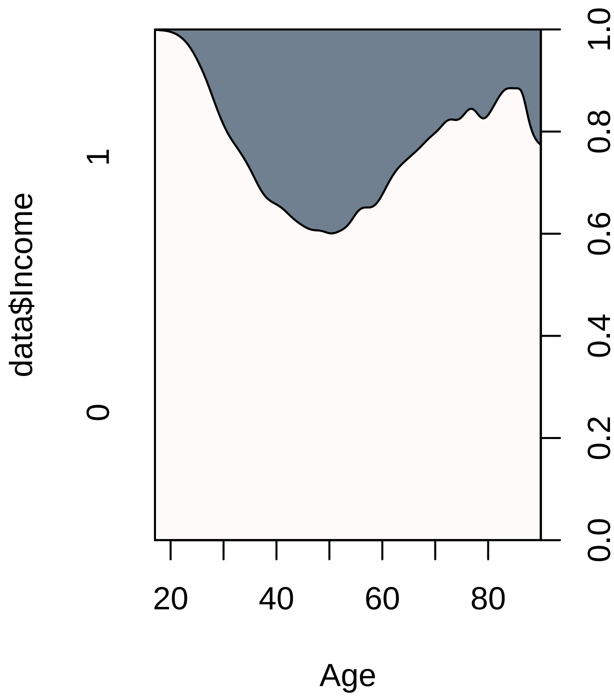


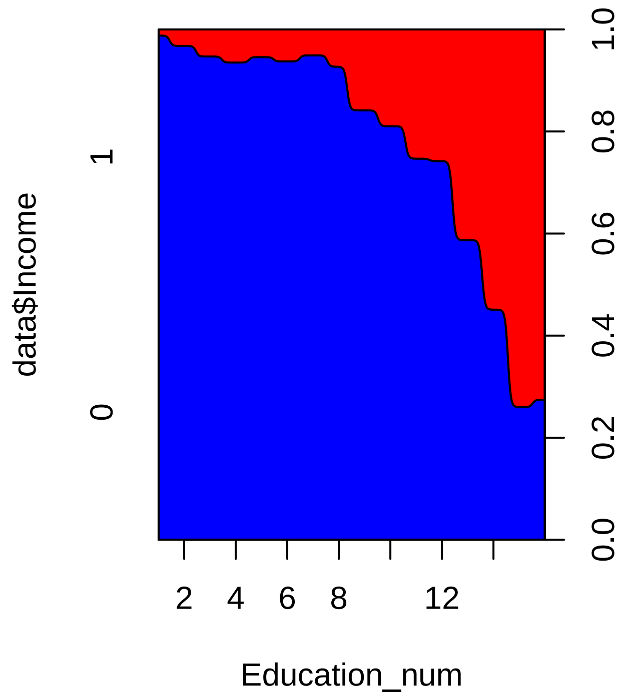

```
# Scatter plot for Race and Hours worked per week. 0 to 4 suggests different type of race. Here
#"Amer-Indian-Eskimo" = 0
#"Asian-Pac-Islander" = 1
#"Black" = 2
#"White" = 3
#"Other" = 4
plot(data$Race, df$Hours_per_week, pch='+', cex=0.75, col="blue", xlab="Race", ylab="Education Num")
```




```
#plotting Income (qualitative) against Age and Education Num (both quantitatives)
par(mfrow=c(1,2))
```

```r
cdplot(data$Income~Age, col=c("snow", "slategray"))
cdplot(data$Income~Education_num, col = c("blue","red"))
```





Logistic Regression

Divide into train and test (Using the same sample for all algorithms).

Features selected are : a. Age b. Education_num c.Race d.Sex e.Capital_gain f.Capital_loss g. Hours_per_week h. Income The reason for selecting those features is as follows: - Only selecting numeric data (or data converted to numeric after importing). - There is little to no correlation between the selected columns. - I am predicting Income Level based on Age, Sex, Race, Education and the # of hours they work per week.

```r
# Randomly sample the data set to let 2/3 be training and 1/3 test.
set.seed(1958)  # setting a seed gets the same results every time
i <- sample(1: nrow(data),0.67 * nrow(data), replace = TRUE)
```

```r
#Creating train and test for logistic regression.
logistic_train <- data[i,]
logistic_test <- data[-i,]
```

Key points: -I got the error message : Warning message: glm.fit: fitted probabilities numerically 0 or 1 occurred This means that the data is perfectly or nearly perfectly linearly separable and the error occured due to the inability to maximize the likelihood which already has separated the data perfectly. -Since, null deviance considers the intercept alone, and the residual deviance considers all predictors. The drop in the value of residual deviance indicates that our predictors are good predictors. -82% accuracy is achieved. -p-value is good for all predictors except Race.

Build the model

```
logistic_model <- glm(Income~. ,data=logistic_train, family=binomial)
```

```
## Warning: glm.fit: fitted probabilities numerically 0 or 1 occurred
```

```
summary(logistic_model)
```

```
##
## Call:
## glm(formula = Income ~ ., family = binomial, data = logistic_train)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -5.0596  -0.6199  -0.3751  -0.1040   3.3677
##
## Coefficients:
##                  Estimate Std. Error z value Pr(>|z|)
## (Intercept)    -9.503e+00  1.520e-01 -62.502  < 2e-16 ***
## Age             4.152e-02  1.241e-03  33.447  < 2e-16 ***
## Education_num   3.398e-01  6.974e-03  48.723  < 2e-16 ***
## Race            1.895e-01  3.269e-02   5.797 6.75e-09 ***
## Sex             1.199e+00  4.014e-02  29.879  < 2e-16 ***
## Capital_gain    3.253e-04  1.012e-05  32.164  < 2e-16 ***
## Capital_loss    6.312e-04  3.377e-05  18.691  < 2e-16 ***
## Hours_per_week  3.246e-02  1.321e-03  24.560  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 36141  on 32723  degrees of freedom
## Residual deviance: 25575  on 32716  degrees of freedom
## AIC: 25591
##
## Number of Fisher Scoring iterations: 7
```

```
probs <- predict(logistic_model, newdata=logistic_test, type="response")
pred <- ifelse(probs>0.5, 1, 0)
acc1 <- mean(pred==logistic_test$Income)
print(paste("Logistic model accuracy = ", acc1))
```

```
## [1] "Logistic model accuracy =  0.822224890917097"
```

```
table(pred, logistic_test$Income)
```

```
##
## pred     0     1
##    0 17923  3349
##    1  1092  2617
```

Additional Metrics : Confusion Matrix

Accuracy for logistic regression is 0.8222
Confusion Matrix :: Reference Prediction 1 0 0 17923 3349 1 1092 2617

Sensitivity calculated as 0.9426 Specificity calculated as 0.4378
Kappa calculated as 0.4381. The Kappa value sugessts that it is a "moderate agreement".

```
library(caret)
#Confusion Matrix, Sensitivity, Specificity, Kappa calculation, Accuracy and Error Rate calculation.
confusionMatrix(
  factor(pred, levels = 0:1),
  factor(logistic_test$Income, levels = 0:1)
)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 17923  3349
##          1  1092  2617
##
##                Accuracy : 0.8222
##                  95% CI : (0.8174, 0.8269)
##     No Information Rate : 0.7612
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4381
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9426
##             Specificity : 0.4387
##          Pos Pred Value : 0.8426
##          Neg Pred Value : 0.7056
##              Prevalence : 0.7612
##          Detection Rate : 0.7175
##    Detection Prevalence : 0.8515
##       Balanced Accuracy : 0.6906
##
##        'Positive' Class : 0
##
```

Additional Metrics: ROCR

ROC curve is the visualization of the True Positive/ False Positive rate. We would want to see the curve shooting up right from the origin. Auc (Area Under the Curve) is calculated as 0.8424941 1 would have been a perfect classifier but, 0.84 is a fair auc.

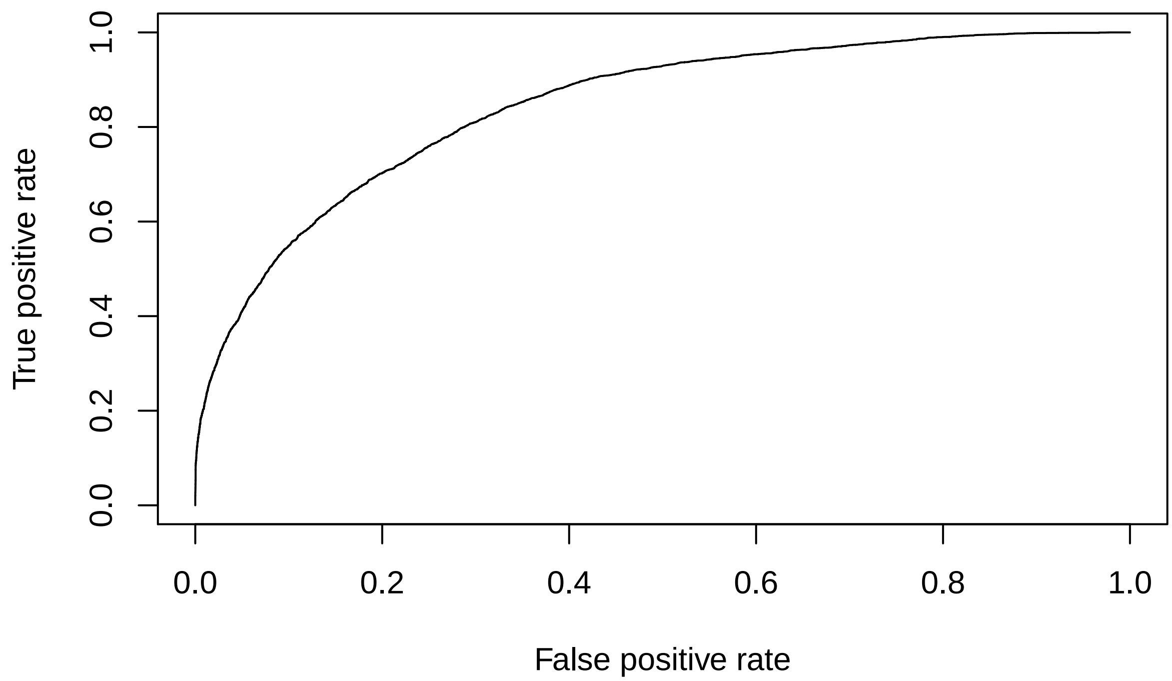
```
library(ROCR)
```

```
## Loading required package: gplots
```

```
##
## Attaching package: 'gplots'

## The following object is masked from 'package:stats':
##
##     lowess
```

```
pr <- prediction(probs, logistic_test$Income)
# TPR = sensitivity, FPR=specificity
prf <- performance(pr, measure = "tpr", x.measure = "fpr")
plot(prf)
```




```
auc <- performance(pr,measure = "auc")
auc <- auc@y.values[[1]]
auc
```

```
## [1] 0.8424941
```

The next algorithm that I am going to try is kNN.

kNN

Divide into train and test

```
knn_train <- data[i,c(1:7)] # train data
knn_test <- data[-i,c(1:7)] # test data
knn_trainlevel <-data[i,8] # train level
knn_testlevel<-data[-i,8] # test level
```

Classify

The knn() function uses Euclidean distance to find the k nearest neighbors. Classification is decided by majority vote with ties broken at random. Using an odd k can avoid some ties. I am using k = 3.

```
library(class)
knn_pred <- knn(knn_train,knn_test,cl = knn_trainlevel,k=3)
```

Compute Accuracy

```
knn_results <- knn_pred == knn_testlevel
knn_acc <- length(which(knn_results == TRUE)) / length(knn_results)
print(paste("kNN accuracy = ", knn_acc))
```

```
## [1] "kNN accuracy =  0.823746046995717"
```

There is slight increase in the accuracy but its not so significant. Logistic Regression accuray was 0.822 whereas kNN accuracy on unscaled data is 0.8237. Since, I donot see huge jump in accuracy I will try to normalize the data and run kNN on normalized data.

Trying to scale the data

Means and standard deviations of predictors are calculated and used as center and scale respectively for the train and test data.

```
#normalize data
means <- sapply(knn_train, mean)
stdvs <- sapply(knn_train, sd)
scaled_train <- scale(knn_train,center = means,scale = stdvs)
scaled_test <- scale(knn_test, center = means, scale = stdvs)
```

kNN on scaled data.

Unfortunately, scaling the data set didn't improve the accuracy rate. Rather, we have seen ~ 2% decrease in the accuracy rate. Acccuracy for scaled kNN classification is 0.80 only.

```
scaled_pred <- knn(scaled_train,scaled_test,cl = knn_trainlevel, k = 3)
scaledknn_results <- scaled_pred == knn_testlevel
scaledknn_acc <- length(which(scaledknn_results == TRUE)) / length(scaledknn_results)
print(paste("Scaled kNN accuracy = ", scaledknn_acc))
```

```
## [1] "Scaled kNN accuracy =  0.801609223009487"
```

Next, I am going to try Naive Bayes algorithm to see if it improves the accuracy.

Naive Bayes

I am using the same sample size but creating new test and train data for comparison. I am also converting Race into factor.

Divide into train and test.

```
nb_train <- data[i,]
nb_test <- data[-i,]
nb_train$Race <- as.factor(nb_train$Race) # Race is converted to factor in train data.
nb_test$Race <- as.factor(nb_test$Race) # Race is converted to factor in test data.
```

Build the naive bayes classifier

The prior for Income Level, called A-priori above, is .75 <=50K and .24 >50K. The likelihood data is shown in the output as conditional probabilities. For discrete variables like Sex and Race, there is a breakdown by income <=50K/>50K for each possible value of the attribute. For continuous data like age, education_num we are given the mean and standard deviation for the two classes.

```
library(e1071)
naive_bayes <- naiveBayes(nb_train$Income~.,data = nb_train)
naive_bayes
```

```
##
## Naive Bayes Classifier for Discrete Predictors
##
## Call:
## naiveBayes.default(x = X, y = Y, laplace = laplace)
##
## A-priori probabilities:
## Y
##         0         1
## 0.7590148 0.2409852
##
## Conditional probabilities:
##    Age
## Y       [,1]     [,2]
##   0 36.81709 14.13075
##   1 44.28684 10.32634
##
##    Education_num
## Y        [,1]     [,2]
##   0  9.575006 2.456972
##   1 11.596754 2.378131
##
##    Race
## Y             0           1           2           3           4
##   0 0.011434093 0.028907319 0.110797971 0.839198003 0.009662614
##   1 0.004818666 0.034237890 0.046284555 0.910474258 0.004184631
##
##    Sex
## Y        [,1]      [,2]
##   0 0.6082213 0.4881575
##   1 0.8479584 0.3590840
##
##    Capital_gain
```

```
## Y         [,1]      [,2]
##   0  138.4636   894.847
##   1 3939.5886 14454.949
##
##    Capital_loss
## Y         [,1]     [,2]
##   0  52.06337 305.7550
##   1 186.01725 584.0543
##
##    Hours_per_week
## Y        [,1]     [,2]
##   0 38.73412 12.47821
##   1 45.55897 11.24395
```

There is even more drop in the accuracy. Accuracy of only 0.789 is achieved.NB has higher bias but lower variance than logistic regression so it didn't do well with the data. NB also works better with smaller data set.

Evaluate on the test data.

```r
nb_pred <- predict(naive_bayes, newdata=nb_test, type="class")
table(nb_pred, nb_test$Income)
```

```
##
## nb_pred     0     1
##       0 18005  4238
##       1  1010  1728
```

```r
nb_acc <- mean(nb_pred==nb_test$Income)
print(paste("Naive Bayes accuracy = ", nb_acc))
```

```
## [1] "Naive Bayes accuracy =  0.789920339457988"
```

Additional Metric : Confusion Matrix on NB

Accuracy for naive bayes is 0.7899 Confusion Matrix :: Reference Prediction 0 1 0 18005 4238 1 1010 1728

Sensitivity calculated as 0.9469 Specificity calculated as 0.2896
Kappa calculated as 0.2904. The Kappa value sugessts that it is a "fair agreement". P-value is < 2.2e-16 which is good.

```r
confusionMatrix(nb_pred, nb_test$Income, positive="0")
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 18005  4238
##          1  1010  1728
```

```
##
##                Accuracy : 0.7899
##                  95% CI : (0.7848, 0.795)
##     No Information Rate : 0.7612
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.2904
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9469
##             Specificity : 0.2896
##          Pos Pred Value : 0.8095
##          Neg Pred Value : 0.6311
##              Prevalence : 0.7612
##          Detection Rate : 0.7207
##    Detection Prevalence : 0.8904
##       Balanced Accuracy : 0.6183
##
##        'Positive' Class : 0
##
```

Analysis of the best algorithm:

The algorithm that was able to achieve highest accuracy in this data set was logistic regression. The reason for logistic regression to outperform both kNN and Naive Bayes is that the classes were linearly separable. For NB, the accuracy is the lowest. The reason for the lowest accuracy could be NB's indpendence assumption. NB also has high bias and low variance than logistic regression. For kNN, in general, it is better to a good idea to scale the variables for better distance calculation but in my case, it performed worse than unscaled kNN classification.

What was learnt from the data

Our best model, logisitc model, suggests that all of our variables were good predictors. The income level is affected by Age, Sex, Race altogether. The model suggested that 71% people had an income level of less than 50K. The model takes all factors into consideration rather than a single variable.The response is determined by a linear combination of predictors. The linear models for classification create a linear decision boundary that is a combination of the all predictors. Based on our linear model, gender had a huge impact on the income level. It is then followed by Education_num (represents the education level and years of education as a number) which is followed by Race. Capital_gain and Capital_loss didn't have significant impact on the income level.