Adult Census Income

Abraham Verde

March/8/2022

INTRODUCTION

For this project I used a dataset that contains an Adult Census of USA in 1994. This dataset has 15 variables and a field that tell us if person's income is lower o higher than 50k usd per year.

In this project, I will perform two machine learning algorithms to predict the income in a test set. This algorithms will be GLM (generalized linear model) and Random Forest.

The data set used in this project I downloaded from kaggel website via https://www.kaggle.com/ and is also available to download from my github account https://github.com/abrahamverde/adult_census/raw/master/adult.csv.

```
if(!require(readr)) install.packages("readr", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: readr
```

```
if(!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: tidyverse
```

```
## -- Attaching packages --------------------------------------- tidyverse 1.3.1 --
```

```
## v ggplot2 3.3.5     v dplyr   1.0.6
## v tibble  3.1.1     v stringr 1.4.0
## v tidyr   1.1.3     v forcats 0.5.1
## v purrr   0.3.4
```

```
## -- Conflicts ------------------------------------------ tidyverse_conflicts() --
## x dplyr::filter() masks stats::filter()
## x dplyr::lag()    masks stats::lag()
```

```
if(!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: caret
```

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

```
##
## Attaching package: 'caret'
```

```
## The following object is masked from 'package:purrr':
##
##     lift
```

```r
if(!require(data.table)) install.packages("data.table", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: data.table
```

```
##
## Attaching package: 'data.table'
```

```
## The following objects are masked from 'package:dplyr':
##
##     between, first, last
```

```
## The following object is masked from 'package:purrr':
##
##     transpose
```

```r
if(!require(ggplot2)) install.packages("data.table", repos = "http://cran.us.r-project.org")
if(!require(dplyr)) install.packages("data.table", repos = "http://cran.us.r-project.org")
if(!require(ranger)) install.packages("ranger", repos = "http://cran.us.r-project.org")
```

```
## Loading required package: ranger
```

```r
library(readr)
library(tidyverse)
library(caret)
library(data.table)
library(ggplot2)
library(dplyr)
library(ranger)
```

ANALYSIS

The first step is getting the data from csv file.

```r
datasetURL <- "https://github.com/abrahamverde/adult_census/raw/master/adult.csv"
rawDataSet <- read.csv(datasetURL)
```

LOAD CSV DATA SET FROM MY GITHUB ACCOUNT Once I loaded the dataset, I started to do some data exploration.

```r
#GET COLUMNS NAME
names(rawDataSet)
```

```
##  [1] "age"            "workclass"      "fnlwgt"         "education"     
##  [5] "education.num"  "marital.status" "occupation"     "relationship"  
##  [9] "race"           "sex"            "capital.gain"   "capital.loss"  
## [13] "hours.per.week" "native.country" "income"
```

```r
#GET A LITTLE SAMPLE DATA
head(rawDataSet, 15)
```

```
##    age        workclass fnlwgt    education education.num marital.status
## 1   90                ?  77053      HS-grad             9        Widowed
## 2   82          Private 132870      HS-grad             9        Widowed
## 3   66                ? 186061 Some-college            10        Widowed
## 4   54          Private 140359      7th-8th             4       Divorced
## 5   41          Private 264663 Some-college            10      Separated
## 6   34          Private 216864      HS-grad             9       Divorced
## 7   38          Private 150601         10th             6      Separated
## 8   74        State-gov  88638    Doctorate            16  Never-married
## 9   68      Federal-gov 422013      HS-grad             9       Divorced
## 10  41          Private  70037 Some-college            10  Never-married
## 11  45          Private 172274    Doctorate            16       Divorced
## 12  38 Self-emp-not-inc 164526  Prof-school            15  Never-married
## 13  52          Private 129177    Bachelors            13        Widowed
## 14  32          Private 136204      Masters            14      Separated
## 15  51                ? 172175    Doctorate            16  Never-married
##           occupation   relationship  race    sex capital.gain capital.loss
## 1                  ?  Not-in-family White Female            0         4356
## 2    Exec-managerial  Not-in-family White Female            0         4356
## 3                  ?      Unmarried Black Female            0         4356
## 4  Machine-op-inspct      Unmarried White Female            0         3900
## 5     Prof-specialty      Own-child White Female            0         3900
## 6      Other-service      Unmarried White Female            0         3770
## 7       Adm-clerical      Unmarried White   Male            0         3770
## 8     Prof-specialty Other-relative White Female            0         3683
## 9     Prof-specialty  Not-in-family White Female            0         3683
## 10      Craft-repair      Unmarried White   Male            0         3004
## 11    Prof-specialty      Unmarried Black Female            0         3004
## 12    Prof-specialty  Not-in-family White   Male            0         2824
## 13     Other-service  Not-in-family White Female            0         2824
## 14   Exec-managerial  Not-in-family White   Male            0         2824
## 15                 ?  Not-in-family White   Male            0         2824
##    hours.per.week native.country income
## 1              40  United-States  <=50K
## 2              18  United-States  <=50K
## 3              40  United-States  <=50K
## 4              40  United-States  <=50K
## 5              40  United-States  <=50K
## 6              45  United-States  <=50K
## 7              40  United-States  <=50K
## 8              20  United-States   >50K
## 9              40  United-States  <=50K
## 10             60              ?   >50K
## 11             35  United-States   >50K
## 12             45  United-States   >50K
## 13             20  United-States   >50K
```

```
## 14              55  United-States   >50K
## 15              40  United-States   >50K
```

```r
#IT'S IMPORTANT TO KNOW THE LENGHT OF DATASET
nrow(rawDataSet)
```

```
## [1] 32561
```

```r
#PEOPLE WITH THEIR INCOME
peopleIncome <- table(rawDataSet$income)
peopleIncome
```

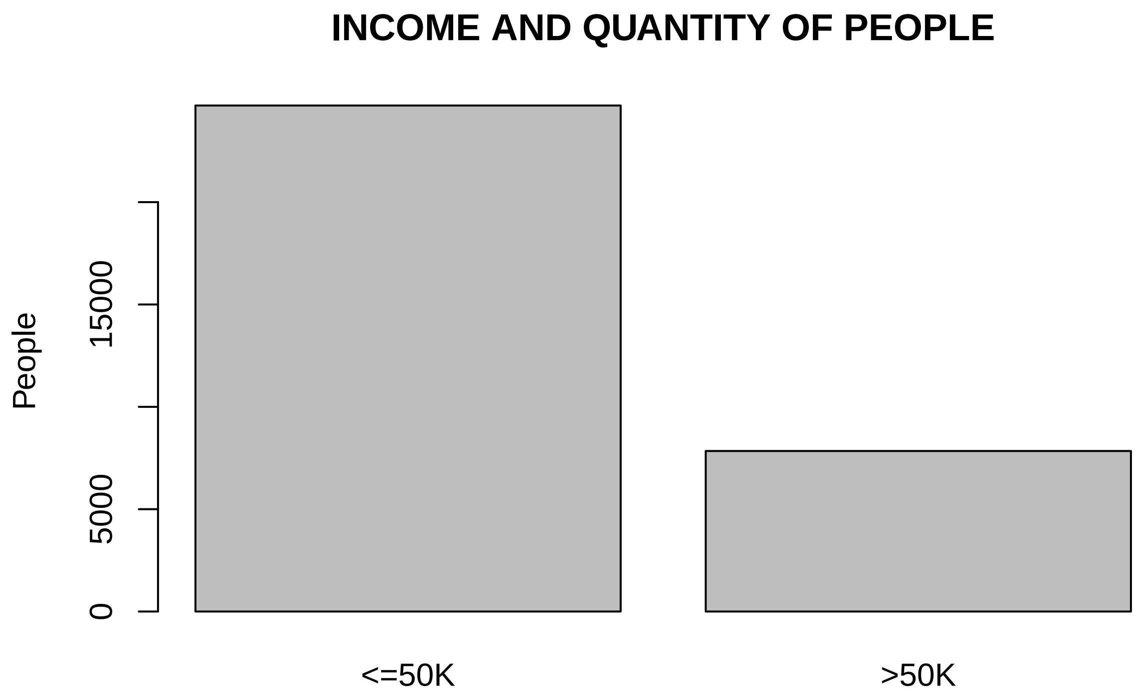
```
##
## <=50K  >50K
## 24720  7841
```

```r
#INCOME RATE
peopleIncome_rate <- prop.table(peopleIncome)
peopleIncome_rate
```

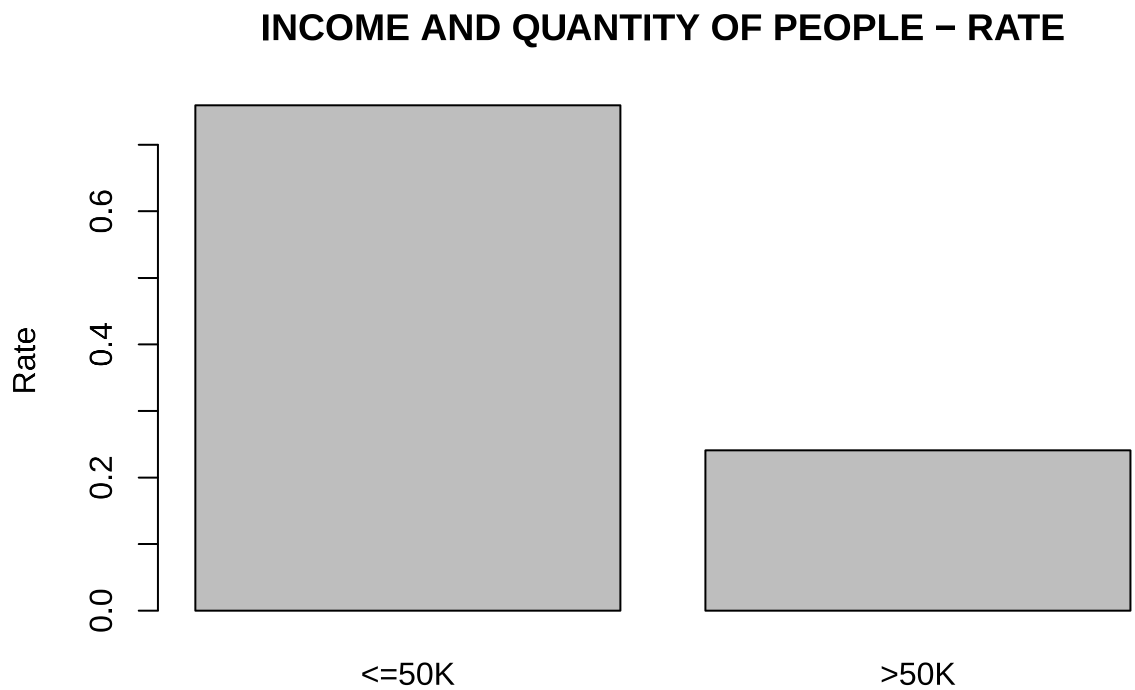
```
##
##     <=50K      >50K
## 0.7591904 0.2408096
```

In the next graphs we can easily see the diference between income values by people. Almost the 75% of the people earn an income lower than 50k per year.

```r
#GRAPH OF QTY
barplot(peopleIncome,main = 'INCOME AND QUANTITY OF PEOPLE',ylab ='People')
```




```r
#GRAPH OF RATE
barplot(peopleIncome_rate,main = 'INCOME AND QUANTITY OF PEOPLE - RATE',ylab ='Rate')
```

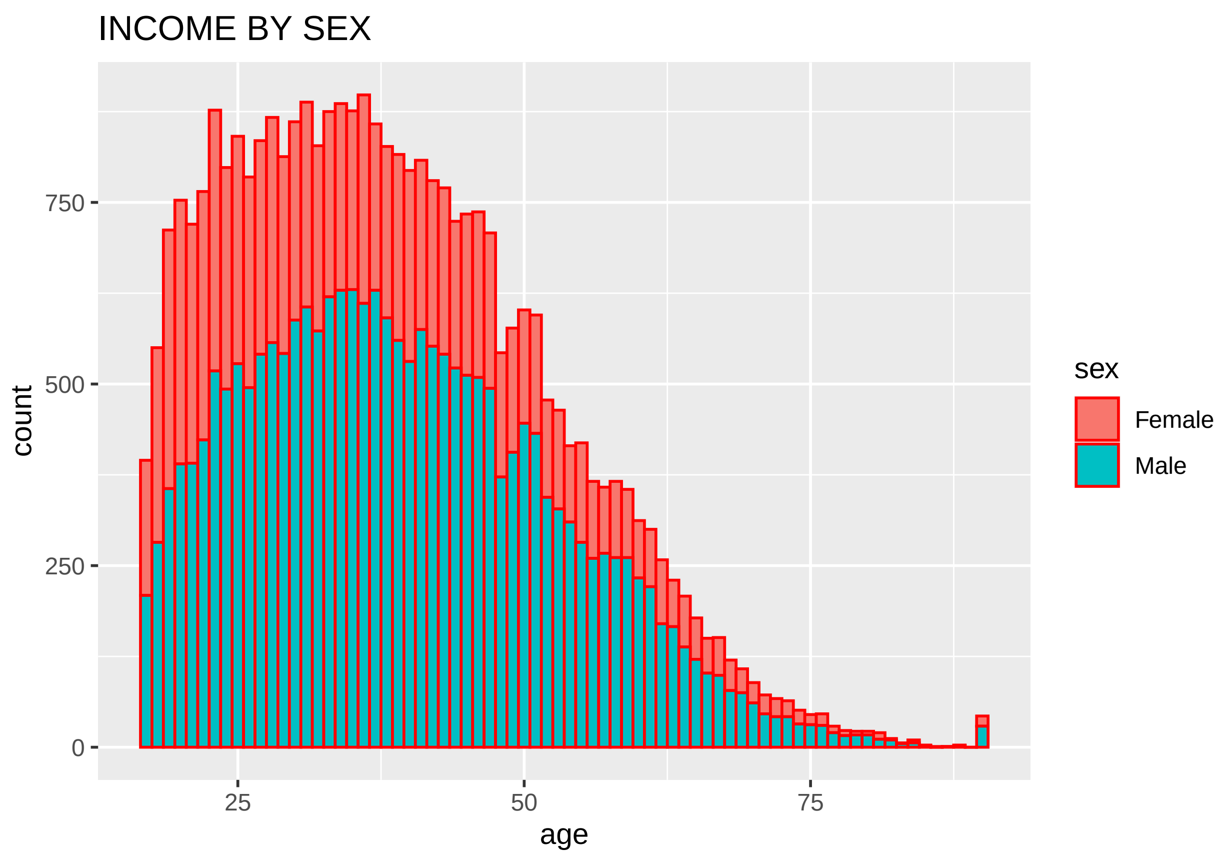


In the next graph we can see the income of the people by age. We can conclude that the higher income is between 30 to 50 years old.

```
#INCOME BY AGE
ggplot(rawDataSet) + aes(x=age, group=income, fill=income) +
  geom_histogram(binwidth=1, color='black')+
  labs(x="Age",y="Count",title = "INCOME BY AGE")
```

By the same way, we can see the higher income belong to male population.

```
ggplot(data=rawDataSet,
       aes(age,group=sex,fill=sex))+
  geom_histogram(binwidth=1, color='red')+ ggtitle('INCOME BY SEX')
```



The exploration data makes me realize there are some really relevant fields for the prediction. So, I'm going to select just the relevants ones.

```
cleanData <- rawDataSet %>% select(income, sex, age)
```

Some data exploration over the clean data dataframe.

```
#PREVIEW
head(cleanData, 50)
```

```
##    income    sex age
## 1   <=50K Female  90
## 2   <=50K Female  82
## 3   <=50K Female  66
## 4   <=50K Female  54
## 5   <=50K Female  41
## 6   <=50K Female  34
## 7   <=50K   Male  38
## 8    >50K Female  74
## 9   <=50K Female  68
## 10   >50K   Male  41
## 11   >50K Female  45
## 12   >50K   Male  38
## 13   >50K Female  52
## 14   >50K   Male  32
```

```
## 15   >50K   Male  51
## 16   >50K   Male  46
## 17   >50K   Male  45
## 18   >50K   Male  57
## 19   >50K   Male  22
## 20   >50K   Male  34
## 21   >50K   Male  37
## 22  <=50K Female  29
## 23  <=50K Female  61
## 24  <=50K   Male  51
## 25  <=50K   Male  61
## 26  <=50K   Male  21
## 27  <=50K   Male  33
## 28  <=50K   Male  49
## 29   >50K   Male  37
## 30   >50K   Male  38
## 31   >50K   Male  23
## 32   >50K Female  59
## 33   >50K   Male  52
## 34   >50K   Male  51
## 35   >50K   Male  60
## 36   >50K Female  63
## 37   >50K   Male  53
## 38   >50K Female  51
## 39   >50K Female  37
## 40   >50K Female  54
## 41   >50K   Male  44
## 42   >50K Female  43
## 43   >50K Female  51
## 44   >50K Female  43
## 45  <=50K   Male  71
## 46   >50K Female  48
## 47  <=50K   Male  71
## 48  <=50K   Male  73
## 49  <=50K Female  68
## 50  <=50K   Male  67
```

```
#CHECK FOR NA
colSums(is.na(cleanData))
```

```
## income    sex    age
##      0      0      0
```

```
#CHECK STRUCTURE
str(cleanData)
```

```
## 'data.frame':    32561 obs. of  3 variables:
##  $ income: Factor w/ 2 levels "<=50K",">50K": 1 1 1 1 1 1 1 2 1 2 ...
##  $ sex   : Factor w/ 2 levels "Female","Male": 1 1 1 1 1 1 2 1 1 2 ...
##  $ age   : int  90 82 66 54 41 34 38 74 68 41 ...
```

I realized the characters inside income field could be a problem in the next steps. I decided to change name to these factors.

```r
levels(cleanData$income)<-c("lower50", "higher50")
str(cleanData)
```

```
## 'data.frame':    32561 obs. of  3 variables:
##  $ income: Factor w/ 2 levels "lower50","higher50": 1 1 1 1 1 1 1 2 1 2 ...
##  $ sex   : Factor w/ 2 levels "Female","Male": 1 1 1 1 1 1 2 1 1 2 ...
##  $ age   : int  90 82 66 54 41 34 38 74 68 41 ...
```

CREATE DATA PARTITION For this prediction, I'm using the 70% to train_set and 30% to test_set.

```r
set.seed(1, sample.kind="Rounding") # if using R 3.5 or earlier, use `set.seed(1)`
```

```
## Warning in set.seed(1, sample.kind = "Rounding"): non-uniform 'Rounding' sampler
## used
```

```r
test_index <- createDataPartition(cleanData$income, times = 1, p = 0.3, list = FALSE)
train_set <- cleanData[-test_index,]
test_set <- cleanData[test_index,]


#EXPLORING DATA PARTITION
nrow(train_set)
```

```
## [1] 22792
```

```r
nrow(test_set)
```

```
## [1] 9769
```

```r
head(train_set, 15)
```

```
##      income    sex age
## 1   lower50 Female  90
## 2   lower50 Female  82
## 3   lower50 Female  66
## 5   lower50 Female  41
## 6   lower50 Female  34
## 7   lower50   Male  38
## 9   lower50 Female  68
## 10 higher50   Male  41
## 11 higher50 Female  45
## 12 higher50   Male  38
## 13 higher50 Female  52
## 14 higher50   Male  32
## 15 higher50   Male  51
## 18 higher50   Male  57
## 19 higher50   Male  22
```

```r
head(test_set, 15)
```

```
##      income    sex age
## 4   lower50 Female  54
## 8  higher50 Female  74
## 16 higher50   Male  46
## 17 higher50   Male  45
## 20 higher50   Male  34
## 25  lower50   Male  61
## 26  lower50   Male  21
## 27  lower50   Male  33
## 28  lower50   Male  49
## 29 higher50   Male  37
## 33 higher50   Male  52
## 37 higher50   Male  53
## 43 higher50 Female  51
## 44 higher50 Female  43
## 46 higher50 Female  48
```

```r
table(train_set$income)
```

```
##
##  lower50 higher50
##    17304     5488
```

FIT GLM MODEL Before try to fit the model, I setup the Train Control Object. This object will "control" the glm train.

```r
trainControlObject <- trainControl(method="cv", number = 10, classProbs = TRUE, summaryFunction = twoCla

#Here I try to fit the model. This process could take a while depending on your computer.
fit_glm <- train(income~., data = train_set, trControl=trainControlObject, family = binomial, method =
```

So far, I got a Logistic Regression model. The randomForest approach is a very popular approach therefore I dediced fit a model using randomForest and show both results (glm and random forest approach.).

```r
#RANDOM FOREST USING RANGER PACKAGE (THIS IS FASTER THAN OLDER PACKAGE "randomForest" )
fit_randomforest <- train(income~., data = train_set, method = "ranger", metric="ROC",num.trees=50,
                          trControl=trainControlObject)
```

```
## note: only 1 unique complexity parameters in default grid. Truncating the grid to 1 .
```

RESULTS

Finally, I have results in both approach.

```r
#GET RESULTS USING RESAMPLES FUNCTION
Results <- resamples(list(LG=fit_glm, RFOREST=fit_randomforest))

#SHOW SOME SUMMARY
summary(Results)
```

```
##
## Call:
## summary.resamples(object = Results)
##
## Models: LG, RFOREST
## Number of resamples: 10
##
## ROC
##              Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## LG      0.7079488 0.7174762 0.7212402 0.7236659 0.7329895 0.7388794    0
## RFOREST 0.7500292 0.7520444 0.7552263 0.7581409 0.7644253 0.7727453    0
##
## Sens
##              Min.   1st Qu.    Median      Mean   3rd Qu.      Max. NA's
## LG      0.9514451 0.9599711 0.9621499 0.9623783 0.9660550 0.9699596    0
## RFOREST 0.9318313 0.9364442 0.9462568 0.9447529 0.9528902 0.9560947    0
##
## Spec
##               Min.    1st Qu.     Median      Mean    3rd Qu.       Max. NA's
## LG      0.03642987 0.04373579 0.04735883 0.0497434 0.05421104 0.07468124    0
## RFOREST 0.11293260 0.14389800 0.16484517 0.1649070 0.19171220 0.20255474    0
```

CONCLUSIONS

After running both model, and taking account their accuracy. I can conclude both techniques give close results but random forest is a little bit better accuracy when is compared with GLM.