HarvardX:PH125.9x Data Science: Capstone part2 - Choose your own

Ioannis Dimitriou

February 16, 2020

Introduction

In this part of the HarvardX Data Science Capstone there is a much bigger challenge than the first one, as we have to choose our dataset from the web and generally act more independently on the data exploration. On this purpose I chose the Adult Census Income database from: "https://www.kaggle.com" .

Dataset

The Adult Census Income dataset was extracted from the website mentioned in the introduction. The first extraction of the data was made by Ronny Kohavi and Barry Becker, on the 1994 Census bureau database. In this dataset each row represents a person and there are several variables as columns. The aim of the dataset is to combine the variables in a machine learning algorithm and predict whether a person's income is greater than $50k or not.

Methods and Analysis

Downloading the Dataset

My fisrt step was to download the dataset from: "https://www.kaggle.com/uciml/adult-census-income" to my system. Then, I uploaded it to my personal github account in order to import it to my code.The URL of the data file on my github account is: "https://github.com/IoannisDim/HarvardX-Data-Science-Capstone-part2/blob/master/adult.csv" .

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

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

```
## -- Attaching packages ------------------------------------------- tidyverse 1.3.0 --
```

```
## <U+2713> ggplot2 3.2.1     <U+2713> purrr   0.3.3
## <U+2713> tibble  2.1.3     <U+2713> dplyr   0.8.3
## <U+2713> tidyr   1.0.0     <U+2713> stringr 1.4.0
## <U+2713> readr   1.3.1     <U+2713> forcats 0.4.0
```

```
## -- 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
```

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

```
## Loading required package: rpart
```

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

```
## Loading required package: randomForest
```

```
## randomForest 4.6-14

## Type rfNews() to see new features/changes/bug fixes.

##
## Attaching package: 'randomForest'

## The following object is masked from 'package:dplyr':
##
##     combine

## The following object is masked from 'package:ggplot2':
##
##     margin
```

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

```
## Loading required package: matrixStats

##
## Attaching package: 'matrixStats'

## The following object is masked from 'package:dplyr':
##
##     count
```

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

```
## Loading required package: gbm

## Loaded gbm 2.1.5
```

```r
#Download the dataset
data<- read.csv("https://raw.githubusercontent.com/IoannisDim/HarvardX-Data-Science-Capstone-part2/maste
```

Data Exploration

Now we can have a first touch with our data by seeing the dimensions of the dataset, the structure and the first 6 observations of it. We can see that there are 32561 observations as rows and 15 variables as columns. We can also observe the category of each variable and the first 6 observations.

```r
#Dimensions
dim(data)
```

```
## [1] 32561    15
```

```r
#Structure
str(data)
```

```
## 'data.frame':    32561 obs. of  15 variables:
##  $ age           : int  90 82 66 54 41 34 38 74 68 41 ...
##  $ workclass     : Factor w/ 9 levels "?","Federal-gov",..: 1 5 1 5 5 5 5 8 2 5 ...
##  $ fnlwgt        : int  77053 132870 186061 140359 264663 216864 150601 88638 422013 70037 ...
##  $ education     : Factor w/ 16 levels "10th","11th",..: 12 12 16 6 16 12 1 11 12 16 ...
##  $ education.num : int  9 9 10 4 10 9 6 16 9 10 ...
##  $ marital.status: Factor w/ 7 levels "Divorced","Married-AF-spouse",..: 7 7 7 1 6 1 6 5 1 5 ...
##  $ occupation    : Factor w/ 15 levels "?","Adm-clerical",..: 1 5 1 8 11 9 2 11 11 4 ...
##  $ relationship  : Factor w/ 6 levels "Husband","Not-in-family",..: 2 2 5 5 4 5 5 3 2 5 ...
##  $ race          : Factor w/ 5 levels "Amer-Indian-Eskimo",..: 5 5 3 5 5 5 5 5 5 5 ...
##  $ sex           : Factor w/ 2 levels "Female","Male": 1 1 1 1 1 1 2 1 1 2 ...
##  $ capital.gain  : int  0 0 0 0 0 0 0 0 0 0 ...
```

```
##  $ capital.loss  : int  4356 4356 4356 3900 3900 3770 3770 3683 3683 3004 ...
##  $ hours.per.week: int  40 18 40 40 40 45 40 20 40 60 ...
##  $ native.country: Factor w/ 42 levels "?","Cambodia",..: 40 40 40 40 40 40 40 40 40 1 ...
##  $ income        : Factor w/ 2 levels "<=50K",">50K": 1 1 1 1 1 1 1 2 1 2 ...
```

```r
#First 6 Observations
head(data)
```

```
##   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
##          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
##   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
```

Data cleaning

The next step is to "clean" our data in order not to have any NAs or missing values. We are going to remove all the observations that have missing values shown as "?". Observing the structure we can easily see that this happens in 3 variables: workclass, occupation, and native.country. After cleaning the dataset we can see that there are 30162 observations left.

```r
data<- data%>% filter(!workclass=="?", !occupation=="?", !native.country=="?")
dim(data)
```

```
## [1] 30162    15
```

Summary of the data

The summary of the data shows that the vast majority of the observations have an income less than or equal to 50k dollars. Specifically 22654 persons have an income <=50k dollars, while the rest 7508 earn more than 50k. The proportion of the majority is 75.01%.

```r
summary(data)
```

```
##       age                    workclass         fnlwgt               education   
##  Min.   :17.00   Private         :22286   Min.   :  13769   HS-grad     :9840  
##  1st Qu.:28.00   Self-emp-not-inc: 2499   1st Qu.: 117627   Some-college:6678  
##  Median :37.00   Local-gov       : 2067   Median : 178425   Bachelors   :5044  
##  Mean   :38.44   State-gov       : 1279   Mean   : 189794   Masters     :1627  
##  3rd Qu.:47.00   Self-emp-inc    : 1074   3rd Qu.: 237629   Assoc-voc   :1307  
##  Max.   :90.00   Federal-gov     :  943   Max.   :1484705   11th        :1048  
```

```
##                   (Other)         :   14                     (Other)     :4618
##  education.num                marital.status           occupation
##  Min.   : 1.00   Divorced             : 4214   Prof-specialty :4038
##  1st Qu.: 9.00   Married-AF-spouse    :   21   Craft-repair   :4030
##  Median :10.00   Married-civ-spouse   :14065   Exec-managerial:3992
##  Mean   :10.12   Married-spouse-absent:  370   Adm-clerical   :3721
##  3rd Qu.:13.00   Never-married        : 9726   Sales          :3584
##  Max.   :16.00   Separated            :  939   Other-service  :3212
##                  Widowed              :  827   (Other)        :7585
##          relationship                   race            sex
##  Husband       :12463   Amer-Indian-Eskimo:  286   Female: 9782
##  Not-in-family : 7726   Asian-Pac-Islander:  895   Male  :20380
##  Other-relative:  889   Black             : 2817
##  Own-child     : 4466   Other             :  231
##  Unmarried     : 3212   White             :25933
##  Wife          : 1406
##
##   capital.gain    capital.loss     hours.per.week        native.country
##  Min.   :    0   Min.   :   0.00   Min.   : 1.00   United-States:27504
##  1st Qu.:    0   1st Qu.:   0.00   1st Qu.:40.00   Mexico       :  610
##  Median :    0   Median :   0.00   Median :40.00   Philippines  :  188
##  Mean   : 1092   Mean   :  88.37   Mean   :40.93   Germany      :  128
##  3rd Qu.:    0   3rd Qu.:   0.00   3rd Qu.:45.00   Puerto-Rico  :  109
##  Max.   :99999   Max.   :4356.00   Max.   :99.00   Canada       :  107
##                                                    (Other)      : 1516
##    income
##  <=50K:22654
##  >50K : 7508
##
##
##
##
##
```

Before we go further to our analysis we should remove some variables that are unnecessary to it. These are "fnlwgt" variable which is an estimation measure of the units of population that are representative of the observation, and the "education" variable as we have also the "educatio.num"

Remove unnecessary variables

```
data<- data%>% select(-c(education, fnlwgt))
```

Create Train and Validation sets

The next step is to create the train and validation sets. Validation set will proportionally the 25% of the data and the rest 75% will get into the train set.

```
set.seed(1,sample.kind = "Rounding")  #if using R3.5 or earlier set.seed(1)
test_index <- createDataPartition(data$income, times = 1, p = 0.25, list = FALSE)
validation<- data[test_index, ]
train_set<- data[-test_index, ]
```

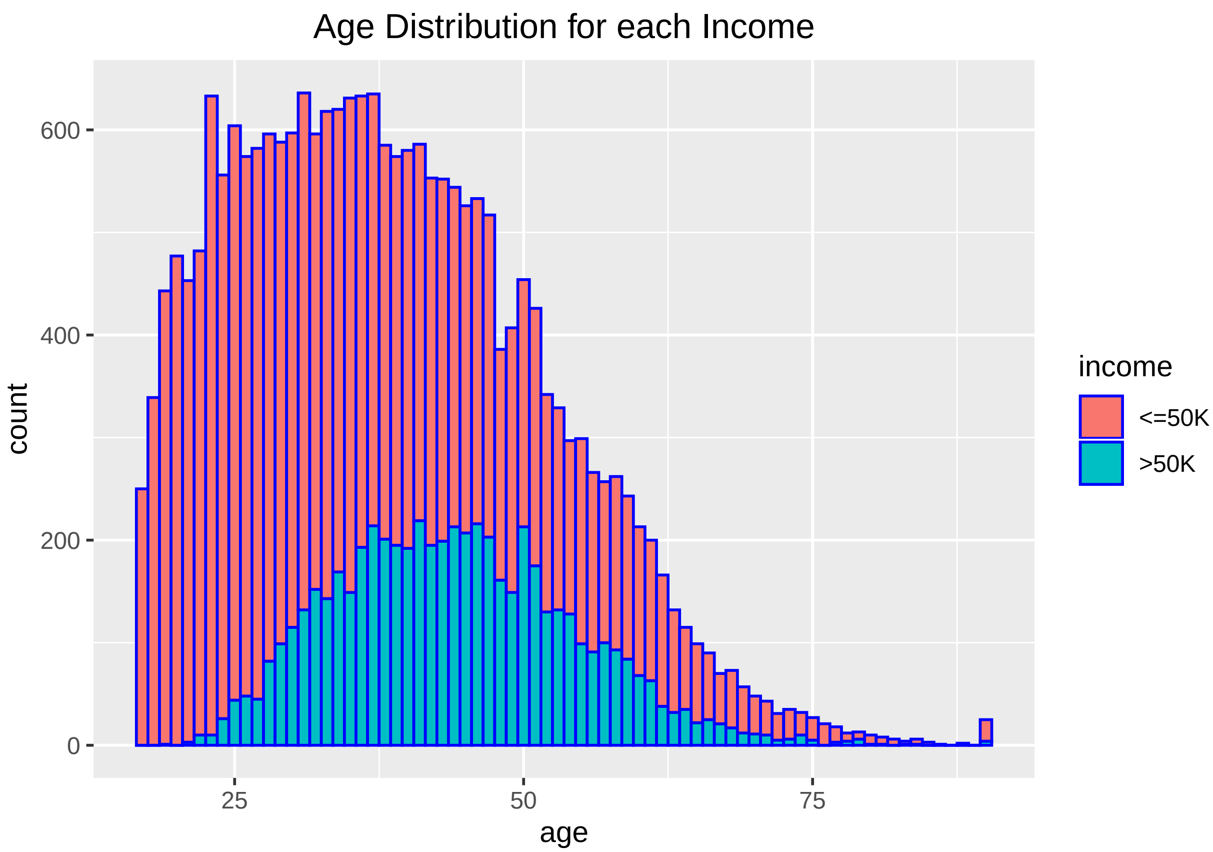
Data Visualization

Through the data visualization we can inspect several variables in order to get good predictors.

Age

The age variable can be a good predictor as it has a large variavility. We can see that on the following histogram.

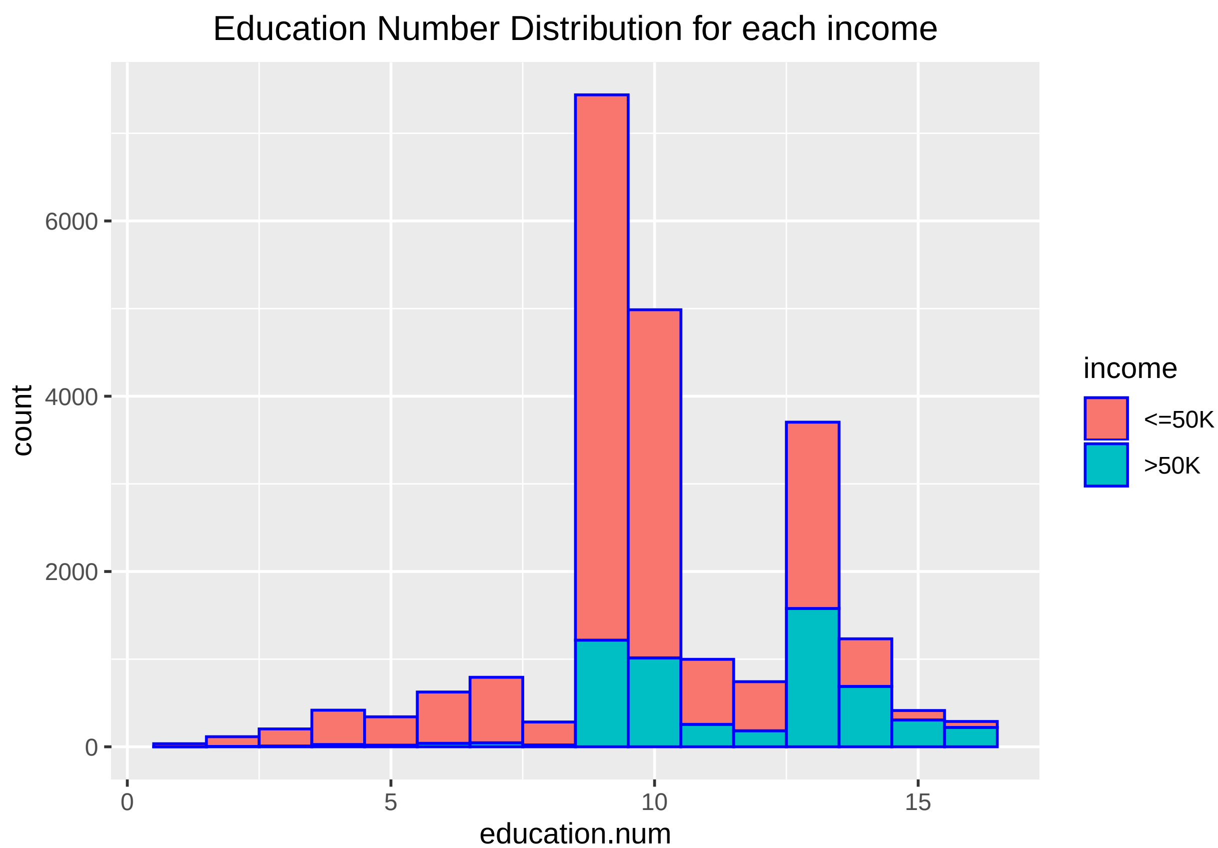
```
train_set%>% ggplot(aes(age)) +
  geom_histogram(aes(fill=income),color='blue',binwidth=1) +
  labs(title= "Age Distribution for each Income")+
  theme(plot.title = element_text(hjust = 0.5))
```



Education.num

Education Number is a variable showing the education level from 1 (Preschool) to 16 (Doctorate). It can be inferred by the following histogram that the higher the education level is, the higher the proportion of people having an income more than 50k gets.

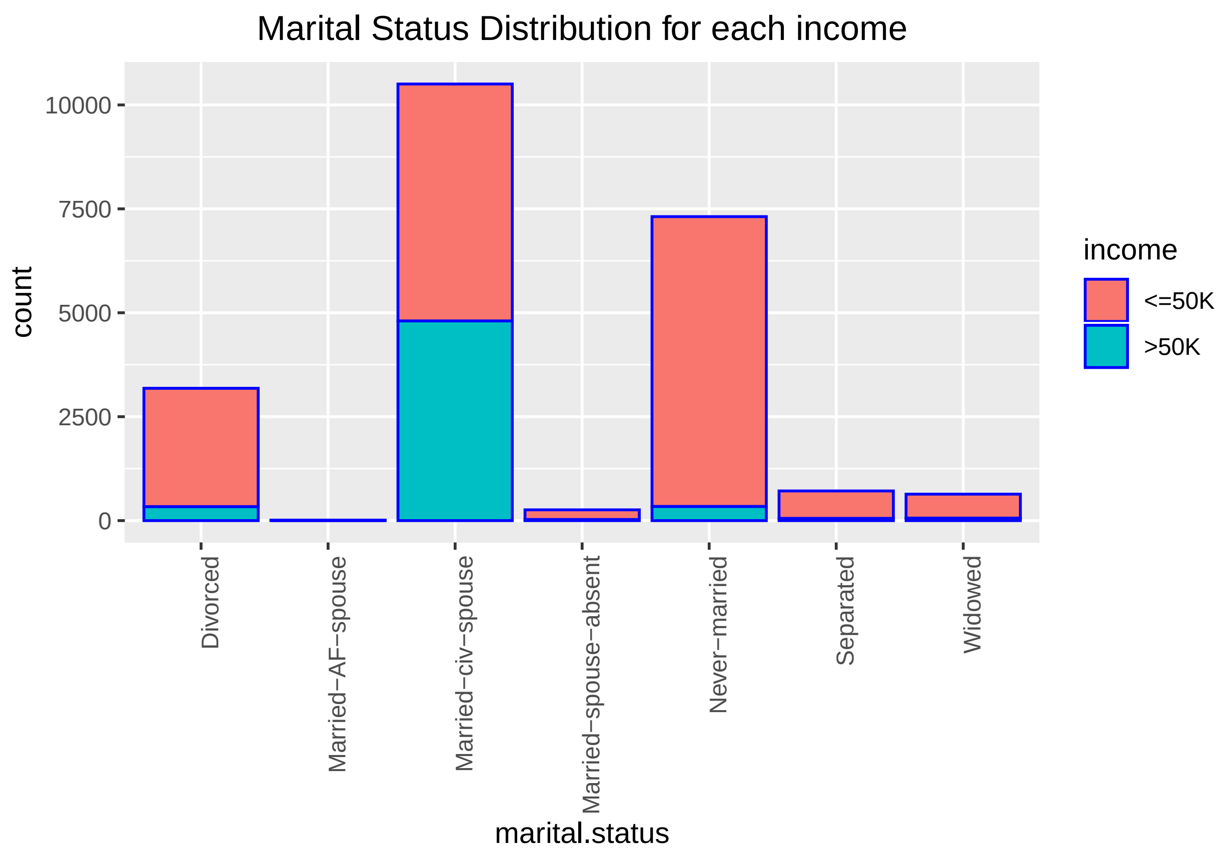
```
train_set%>% ggplot(aes(education.num))+
  geom_histogram(aes(fill=income), color='blue', binwidth = 1)+
  labs(title = "Education Number Distribution for each income")+
  theme(plot.title = element_text(hjust = 0.5))
```



Marital.status

We can see that the proportion of people with more than 50k as income are well distributed according to their marital status. An exemption is people with marital status "Married-civ-spouse". In this category belong the most people of those having >50k income(at about 5000 out of 7508).

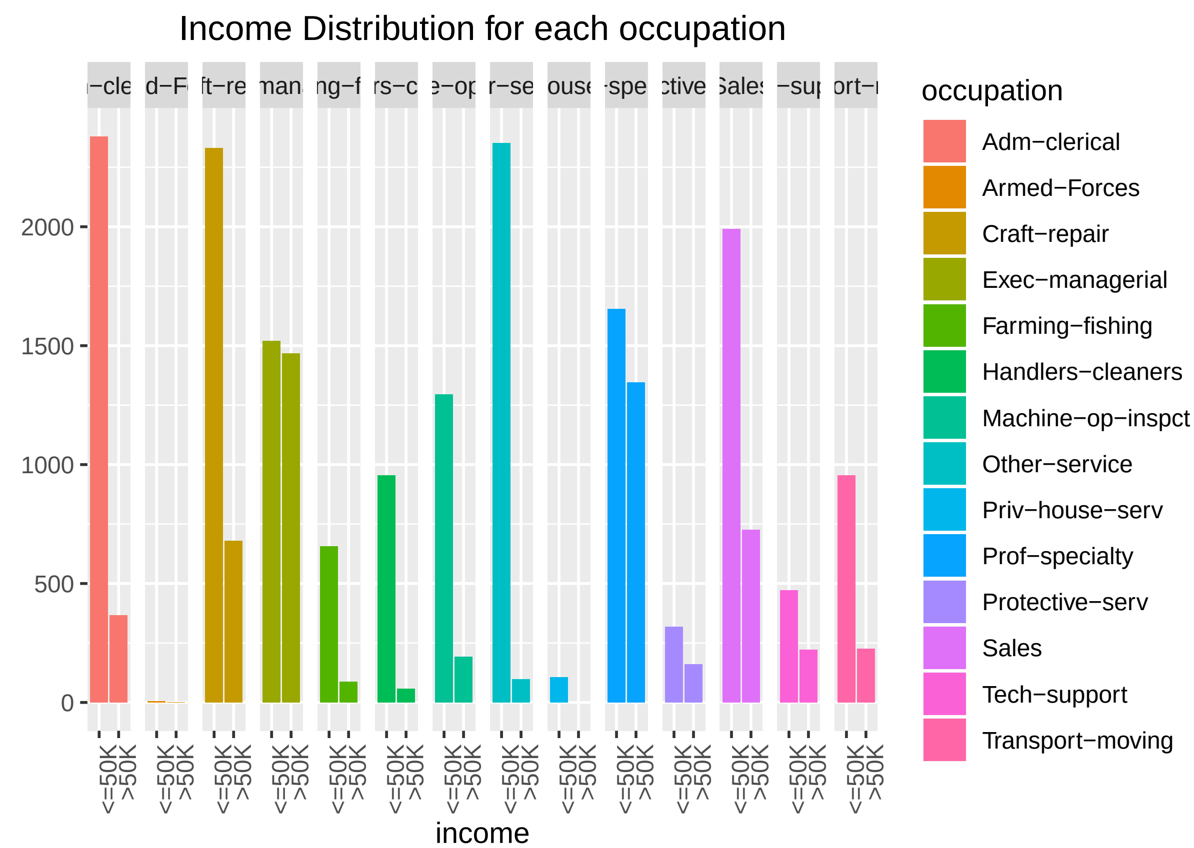
```
train_set%>% ggplot(aes(marital.status))+
  geom_histogram(aes(fill=income),stat = "count", color='blue')+
  labs(title = "Marital Status Distribution for each income")+
  theme(plot.title = element_text(hjust = 0.5))+
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



Occupation

It can be inferred that certain occupations have a bigger proportion of people >50k.

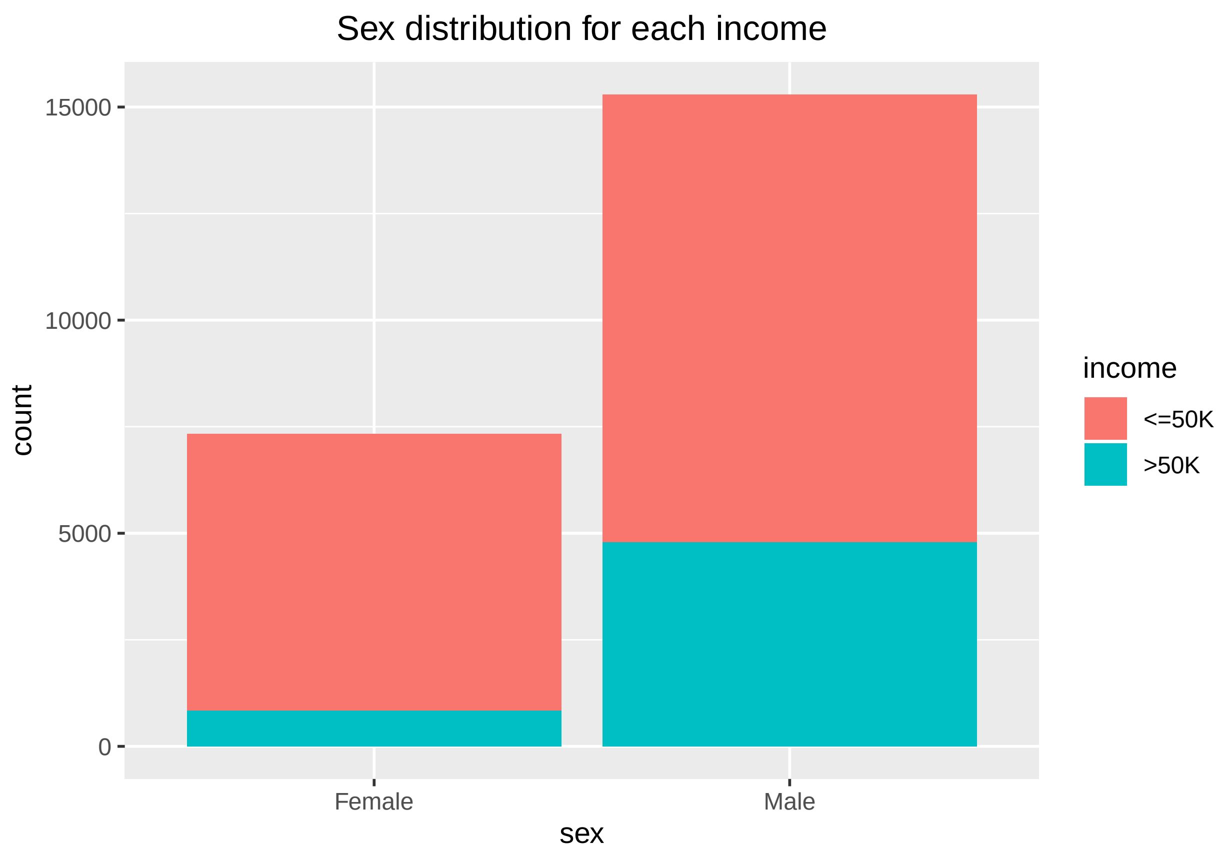
```
qplot(income,data = train_set, fill=occupation)+ facet_grid(.~occupation)+
  labs(title = "Income Distribution for each occupation")+
  theme(plot.title = element_text(hjust = 0.5))+
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



Sex

Here we can see that the vast majority of people having an income greater than 50000 dollars are males.

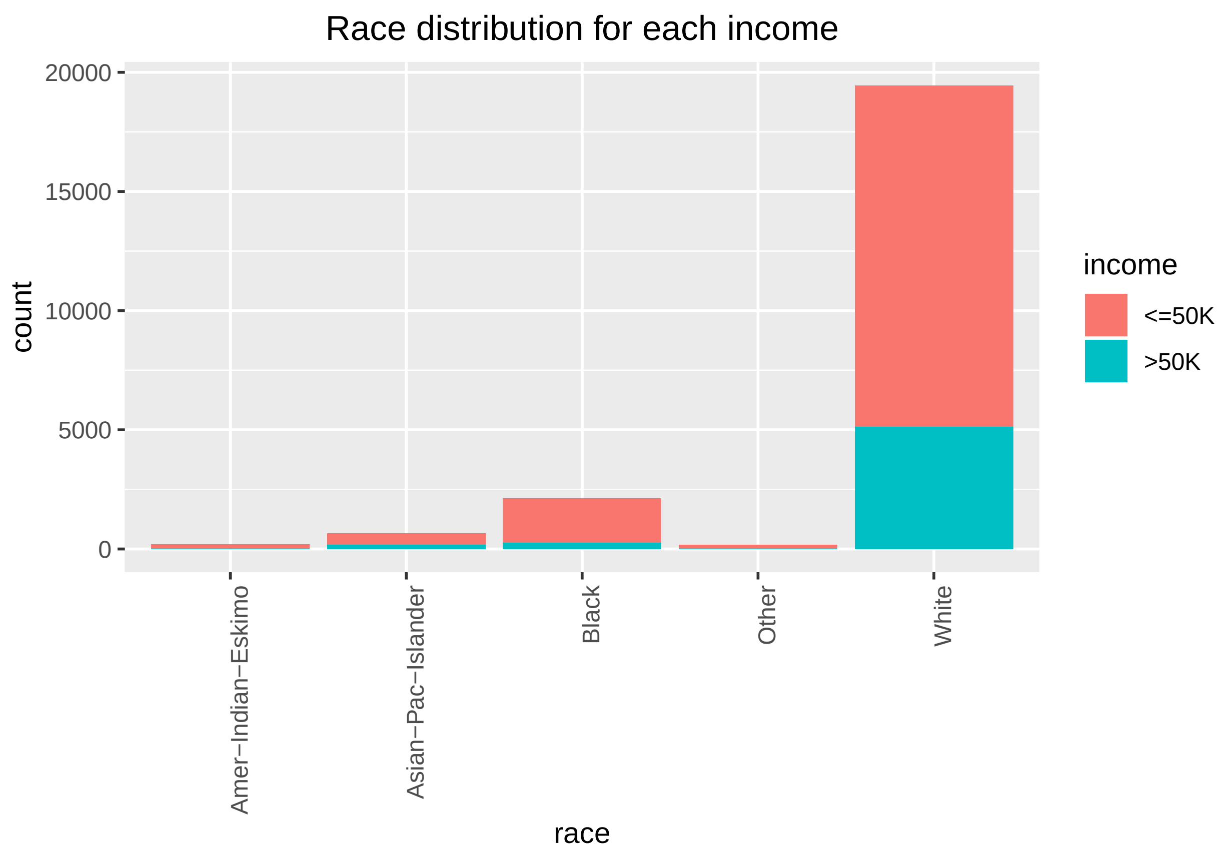
```
train_set%>% ggplot(aes(sex))+
  geom_bar(aes(fill=income), stat = "count")+
  labs(title = "Sex distribution for each income")+
  theme(plot.title = element_text(hjust = 0.5))
```



Race

We can see that almost all people having greater income than 50k are white.

```
train_set %>% ggplot(aes(race))+
  geom_histogram(aes(fill=income),stat="count")+
  labs(title = "Race distribution for each income")+
  theme(plot.title = element_text(hjust = 0.5))+
  theme(axis.text.x = element_text(angle = 90, hjust = 1))
```



Machine Learning Models

After inspecting the dataset and several variables of it , it is time to proceed to our Machine Learning Models in order to predict whether a person has an income lower than or equal to 50k dollars, or greater than this. We are going to inspect the Accuracy of each model so as to find the best predictive model with the highest accuracy.

Split the Train set to run models more efficiently

Before proceeding with the predicting models we are going to split the train set to training and testing set, so as to make our system perform more efficiently.

```
set.seed(10,sample.kind = "Rounding")  #if using R3.5 or earlier set.seed(10)
test_split_index <- createDataPartition(train_set$income, times = 1, p = 0.2, list = FALSE)
testing <- train_set[test_split_index, ]
training <- train_set[-test_split_index, ]
```

Knn (K nearest neighbors) Model

We are going to use a 10-fold cross-validation, have 10 samples and use 10% of the observations in each set.

```
#Using a 10 fold cross-validation
 set.seed(9,sample.kind = "Rounding")
    control <- trainControl(method = "cv", number = 10, p = .9)
    train_knn <- train(income ~ .,
                   method = "knn",
                   data = training,
```

```r
                   tuneGrid = data.frame(k = seq(5,33,2)),
                   trControl = control)
    #See the best k value
    train_knn$bestTune
```

```
##    k
## 6 15
```

```r
    #Compute the accuracy of the knn model on the validation dataset
    knn_accuracy <- confusionMatrix(predict(train_knn, testing, type = "raw"),
                                    testing$income)$overall["Accuracy"]
    #Create a table to save our results for each model
    accuracy_results <- tibble(method = "knn", Accuracy = knn_accuracy)
    #View the knn accuracy results in our table
    accuracy_results %>% knitr::kable()
```

method	Accuracy
knn	0.8457459

Classification Tree Model

The second model that we are going to inspect is The Classification Tree Model. Cross-validation will be used to choose the best cp(complexity parameter).

```r
#Train a Classification Tree model
set.seed(300,sample.kind = "Rounding")  #if using R3.5 or earlier set.seed(300)
 train_rpart <- train(income ~ .,
                      method = "rpart",
                      tuneGrid = data.frame(cp = seq(0, 0.01, len=100)),
                      data = training)
 #See the best cp value
 train_rpart$bestTune
```

```
##             cp
## 12 0.001111111
```

```r
 #Compute the accuracy of the Classification Tree model on the testing dataset
 rpart_accuracy <- confusionMatrix(predict(train_rpart, testing),
                                   testing$income)$overall["Accuracy"]

 #Save the Classification Tree model accuracy results to our table
 accuracy_results <- bind_rows(accuracy_results,
                               tibble(method="rpart", Accuracy = rpart_accuracy))
 #View the rpart accuracy results in our table
 accuracy_results %>% knitr::kable()
```

method	Accuracy
knn	0.8457459
rpart	0.8552486

Random Forest Model

Last but not least, we will inspect the Random Forest Model.

```r
set.seed(3,sample.kind = "Rounding")  #if using R3.5 or earlier set.seed(3)
    train_rf <- randomForest(income ~ ., data = training)
    #Compute the accuracy of the random forest model on the testing dataset
    rf_accuracy <- confusionMatrix(predict(train_rf, testing),
                                  testing$income)$overall["Accuracy"]
    #Save the random forest accuracy results to our table
    accuracy_results <- bind_rows(accuracy_results,
                                 tibble(method="random forest", Accuracy = rf_accuracy))
    #View the random forest accuracy results in our table
    accuracy_results %>% knitr::kable()
```

method	Accuracy
knn	0.8457459
rpart	0.8552486
random forest	0.8585635

Testing the most accurate model with the validation set

From the results table we can see that the model having the highest accuracy is the Random Forest model. Our final step is to test that model using the validation set so as to see the final overall accuracy.

```r
  set.seed(3, sample.kind = "Rounding") #if using R3.5 or earlier set.seed(3)
  final_train_rf <- randomForest(income ~ ., data =training)

 #Compute the accuracy of our final random forest model on the validation set
  final_accuracy <- confusionMatrix(predict(final_train_rf,
                              validation),
                        validation$income)$overall["Accuracy"]

 ##Save the random forest accuracy results to our table.
  accuracy_results <- bind_rows(accuracy_results,
                              tibble(method="Final Random Forest Model",
                                     Accuracy = final_accuracy))
 #View the final random forest model accuracy results in our table
  accuracy_results %>% knitr::kable()
```

method	Accuracy
knn	0.8457459
rpart	0.8552486
random forest	0.8585635
Final Random Forest Model	0.8575786

Results

As we can see from our results table we set up 3 models to predict whether a person has an income greater than 50k dollars or not. The model with the highest accuracy is the Random Forest model having an accuracy of 0.859, after being tested with the split testing set. After that, the model mentioned above was tested with the validation set and we found the final overall accuracy.

method	Accuracy
knn	0.8457459
rpart	0.8552486
random forest	0.8585635
Final Random Forest Model	0.8575786

Conclusion

Summarizing, we inspected the Adult Census Income dataset and our goal was to make a machine learning algorithm, predicting whether a person's income is greater than 50k dollars or not. We achieved that after forming three models and choosing the model with the best accuracy. That was the Random Forest model achieving a 0.858 final overall accuracy after being tested with the validation set. This accuracy is satisfying and adequate for a predictive model.