Big Data for Finance

Final Project

CENSUS INCOME

Income range forecast

Caio Mescouto Terra de Souza
Loan NGUYEN-Thi-Anh
Vy NGUYEN-Phuc-Bao

March 10, 2021

Abstract

The main purpose of this report is to present three prediction models - supervised learning - applied to forecast income range through USA Census data from 1994/1995. The data was splited into training (2/3) and test (1/3). It was carried out a Logistic Regression model, a Decision Tree model and a Naïve Bayes and compared accuracy and the AUC. Our conclusion is that Logistic Regression was better in predict the income range.

Keywords: Machine Learning, Supervised Learning, Logistic Regression, Decision Tree, Naive Bayes, Big Data

Contents

Chapter 1

Introduction and Motivation

Researching Census income in a mainstream topic is carried out by a lot of authorities, national institutions as well as non-government organizations due to its wide and interactive applications. Census income is the foundation information providing detailed statistics that are essential for industries and communities. Trade associations, businesses and chambers of commerce rely on this data for economic development, business decision and strategic planning. These real-life applications urge us to carry out this interesting research with Census income data-set as well as extracted from the 1994 and 1995 current population surveys conducted by the U.S. Census Bureau and implement BigData for Finance knowledge to provide, visualize and statistics information in order to solve the on-hand questions that we may have to handle in the professional career in the upcoming time.

In our work, we are making an attempt to establish a rule whereby we can classify a new individual (on the basis of observed attributes) into one of the existing classes and carry out the prediction based on observed attributes.

Specifically, we would like to have a summary of our analysis and exploration of Census income data to come up with meaningful, important and interesting attributes. After having sufficient knowledge about the attributes, we would perform a predictive task of classification to see whether the annual income of a new individual exceeds \$50K. We also would like to describe the probability of an attribute of a new investigated resident, based on prior knowledge of conditions (features) related to that attribute. (For example, The probability of a person is a Native-born in the US given that his income is lower than \$50K//year, his age is 40, he is a professor, etc...).

By practicing with this topic with such a huge data set, we hope that we can successfully apply and absorb the precious knowledge provided in the Big Data set course and truly understand how to skillfully apply such valuable knowledge to real-life topics.

Chapter 2

Data description

The data set contains weighted census data extracted from the 1994 and 1995 current population surveys conducted by the U.S. Census Bureau. It has 40 demographic and employment-related variables and one categorical variable (Income: -\$50,000; +\$50,000) in which the prediction models will work on. This set is also composed by 299,285 observations and the data was split into train/test in approximately 2/3, 1/3 proportions.

The data set were prepared for studies purposes(Machine Learning and Intelligent Systems 2021) and the tidying data process is very straightforward. (1) The data files don't have the headers, but it follows the metadata, thus it should be added before carry out the models. (2) Missing values are marked as "?", and it was modify following the built-in missing values for R (NA). (3) The variable weight (MARSUPWT) indicates the number of people in the population that each record represents due to stratified sampling and the variable year only informs if the data was collected in 1994 or 1995. These attributes should not be used in the models and were dropped out. (4) Finally, all categorical variables were identified and set as factor for further analysis.

Chapter 3

Descriptive Statistics

For statistical purpose we merge training and test data sets and started the analysis with a summary statistics. Through this process was identified four variables associated with migration status that have a huge number of missing values (147,485 each), therefore this four variables were dropped (MIGMTR1, MIGMTR3, MIGMTR4, MIGSUN). The label of Income was also changed (0 = -\$50,000; 1 = +\$50,000).

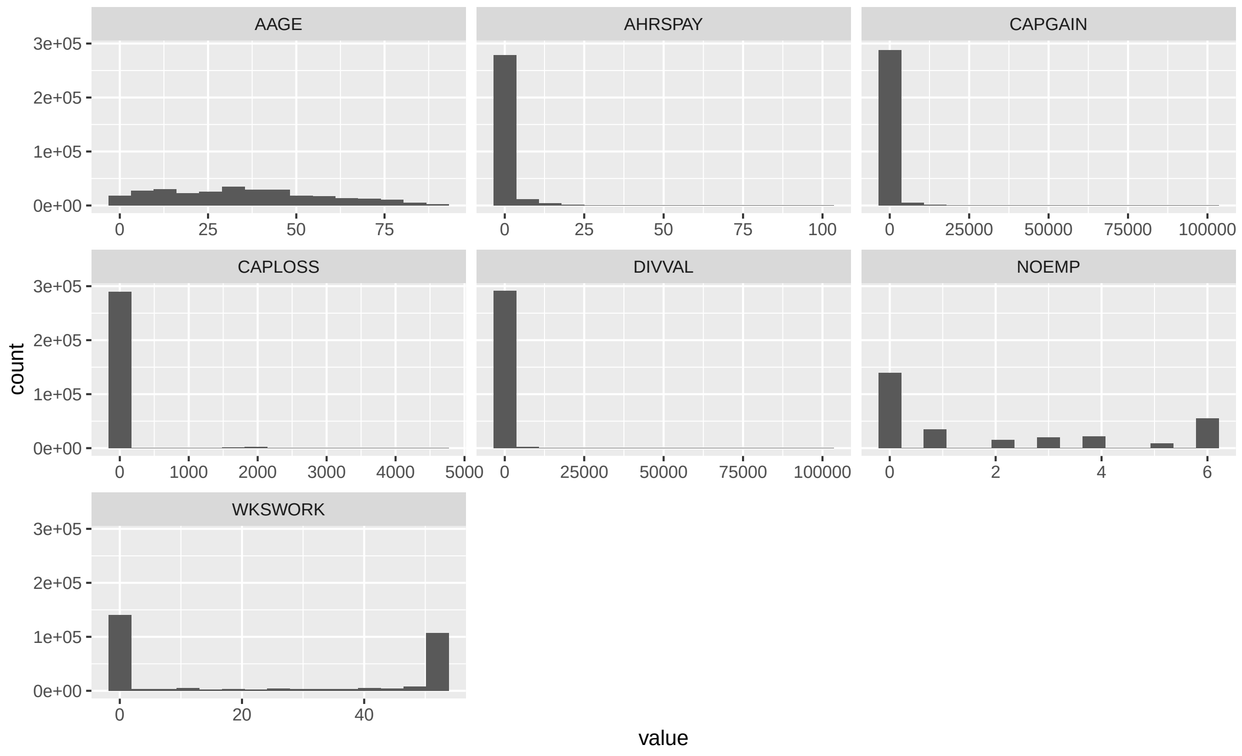


Figure 3.1: Histograms for Numerical Variables

As observed above on the histograms of numerical variables (see Figure 3.1), the data set contains three capital variables that are highly skewed, we chose aggregate these variables into one.

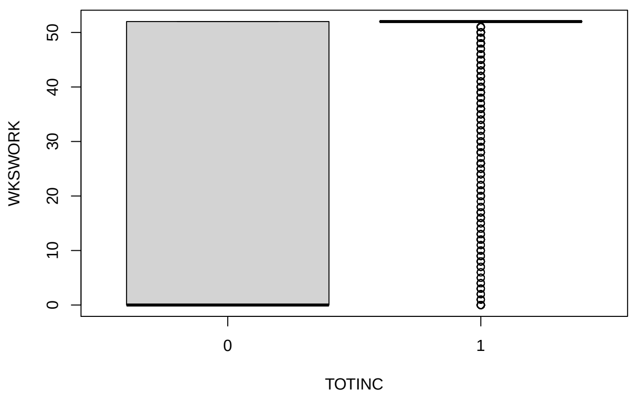
$$CAP.CHANGE = CAPGAIN - CAPLOSS + DIVVAL \tag{3.1}$$

The categorical variables were also analyzed through summary statistics and we chose drop out the following variables PEFNTVTY, PEMNTVTY, PENATVTY, GRINST, VETQVA because they have basically the same value for all observations and don't add meaning for the models.

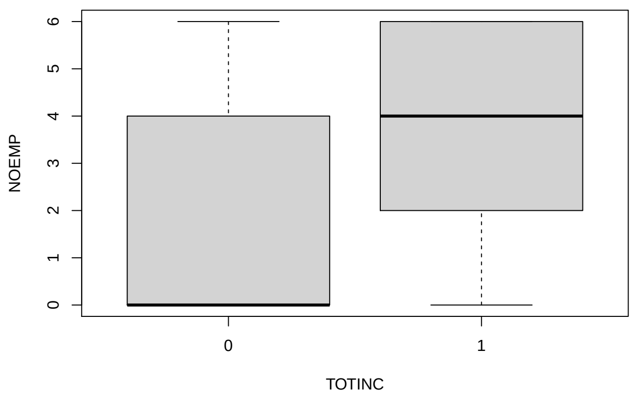
Table 3.1: Covariance Matrix

	AAGE	AHRSPAY	NOEMP	WKSWORK	CAP.CHANGE
AAGE	1.00	0.03	0.13	0.19	0.08
AHRSPAY	0.03	1.00	0.19	0.19	0.00
NOEMP	0.13	0.19	1.00	0.74	0.05
WKSWORK	0.19	0.19	0.74	1.00	0.07
CAP.CHANGE	0.08	0.00	0.05	0.07	1.00

According with the Covariance Matrix (see Table 3.1), the variable WKSWORK and NOEMP are highly correlated. But as both are also correlated with the response variable (see Figure 3.2) we decided to maintain both in the data set.



(a) weeks worked in year



(b) num persons worked for employer

Figure 3.2: Boxplots

We also analyzed more some categorical variables and the relationship between them and the response variable. Class of worker (ACLSWK), Sex (ASEX) and Race (ARECE) are presented as examples (see Figures 3.3 and 3.4).

Finally, after the descriptive analysis, we dropped out some observations with missing values and the final data set in which we developed the model has the following characteristics (see Table 3.2).

Table 3.2: Data sets

	Training	Test
Observations	196,294	98,879
Variables	36	36

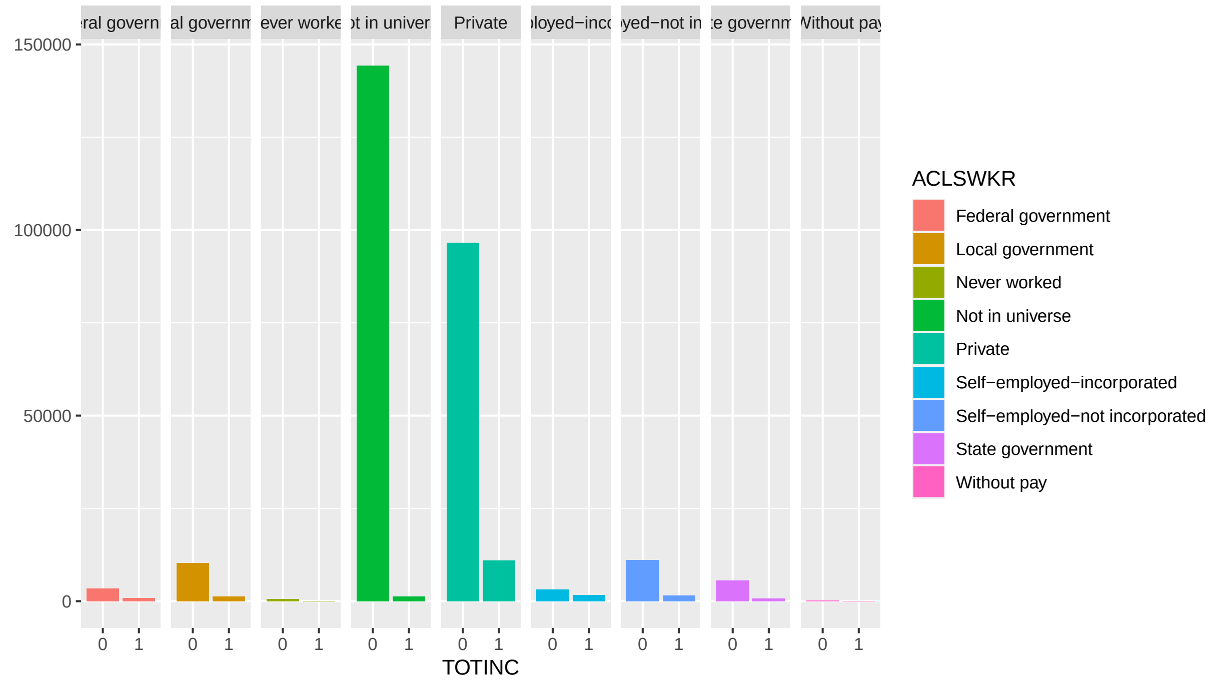
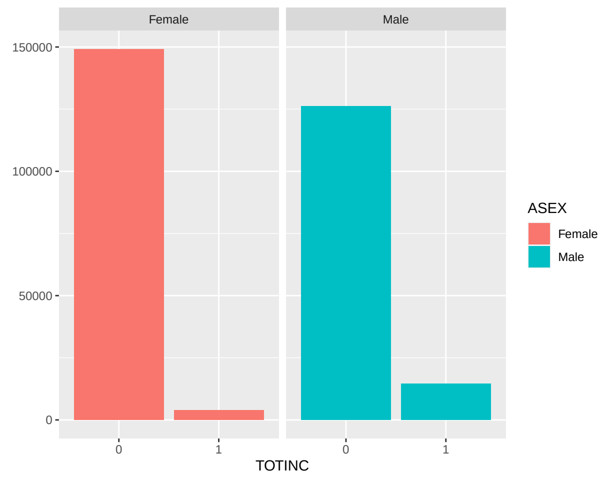
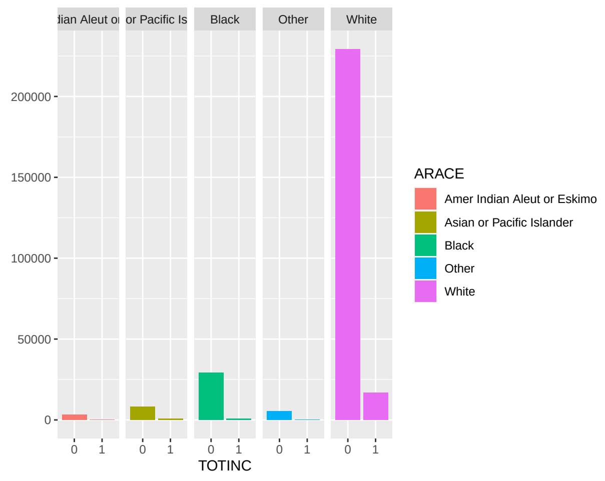


Figure 3.3: Class of worker vs. Income Range



(a) Sex



(b) Race

Figure 3.4: Variables vs. Income Range

Chapter 4

Methodology and Results

After Carried out the descriptive statistics and defined the interest variables we selected three models to predict the response variables. These model are grouped broadly as Supervised Learning Models because they should be trained first. In this kind of problem, we know the possible labels in which observations can be grouped, so we fit the model according with this labels.

4.1 Logistic Regression

The logistic regression model can be applied to predict binomial variables as our response variable (0 = -\$50,000; 1 = +\$50,000). The logistic regression is based on logistic function such as $y \to \infty, f(y) \to 1$ and $y \to -\infty, f(y) \to 0$. Therefore, the outcome is the probability of occurrence (EMC 2015). In our model if $f(y) > 0,5$ we conclude that the observation lay down into the +\$50,000 group. Below the principal indicator of our model. The Accuracy is bigger than 95% Type I Error 0,88%, Type II Error 3.9%.

Null deviance: 91977 on 195423 degrees of freedom

Residual deviance: 49625 on 195201 degrees of freedom

AIC: 50071

$$\text{Pseudo-}R^2 = 0.53953$$

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 91440  3811
##          1   869  2359
##
##                Accuracy : 0.9525
##                  95% CI : (0.9511, 0.9538)
##     No Information Rate : 0.9373
##     P-Value [Acc > NIR] : < 2.2e-16
##
```

```
##                   Kappa : 0.4796
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9906
##             Specificity : 0.3823
##          Pos Pred Value : 0.9600
##          Neg Pred Value : 0.7308
##              Prevalence : 0.9373
##          Detection Rate : 0.9285
##    Detection Prevalence : 0.9672
##       Balanced Accuracy : 0.6865
##
##        'Positive' Class : 0
##
```

Table 4.1: Confusion Matrix (%)

	FALSE	TRUE
0	0.929	0.009
1	0.039	0.024

To understand the Confusion Matrix we have on the top left the percentage of Truly < \$50,000 (98.85%) and in the bottom right the truly > \$50,000 (2.4%). Type II error is on the bottom left (3.9%). In that case, the model predicted < \$50,000 but in reality those observations are > \$50,000 and finally, type I error in the top right (0.88%), where the model predicted >\$ 50,000 but the observations are < \$50,000. All Confusion Matrix that will be presented have the same interpretation with different percentages.

4.1.1 ROC and AUC

The True Positive Rate (TPR) against the False Positive Rate (FPR) is known as the Receiver Operating Characteristic (ROC) curve. The closer the ROC curve tracks along the vertical axis and approaches the upper-left hand of the plot, the better the model performs. Therefore, a fundamental metric is to compute the area under the ROC curve (AUC) (EMC 2015, 186). The AUC is also important to compare the performance whithin models.

```
## A performance instance
##   'False positive rate' vs. 'True positive rate' (alpha: 'Cutoff')
##   with 73594 data points
```

```
## [1] 0.9437635
```

4.2 Decision Tree

The second model is the Decision Tree that use a tree structure to specify decisions in sequence. A decision tree employs a structure of test points (called nodes) and branches, which represent the decision being made. A node without further branches is called a leaf node. For our model the root

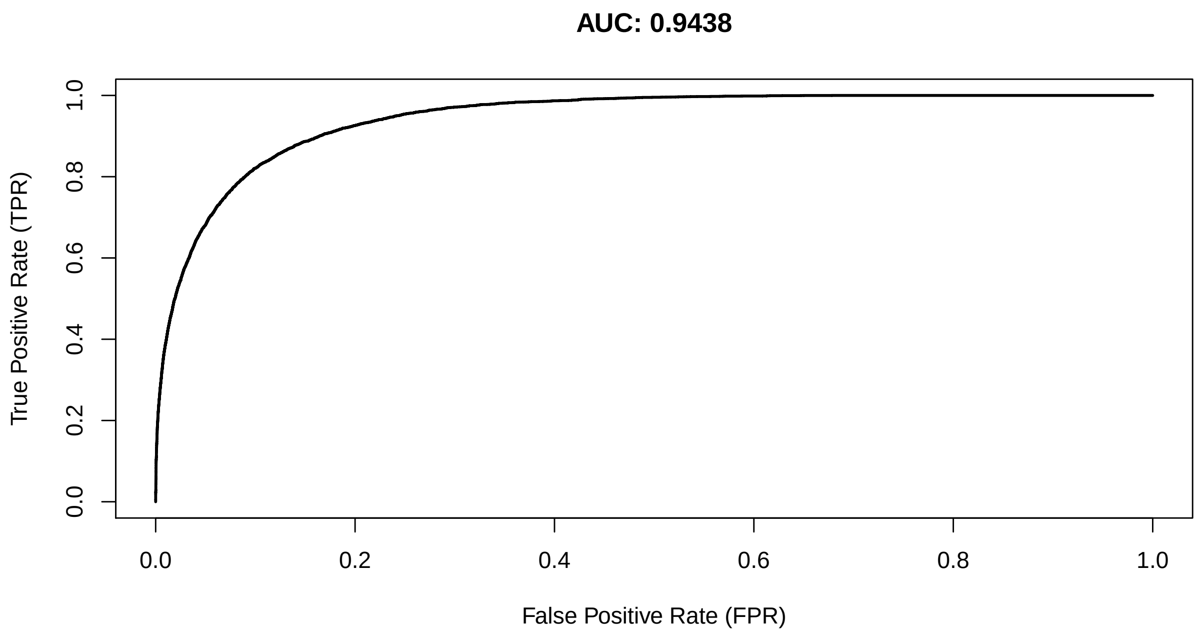
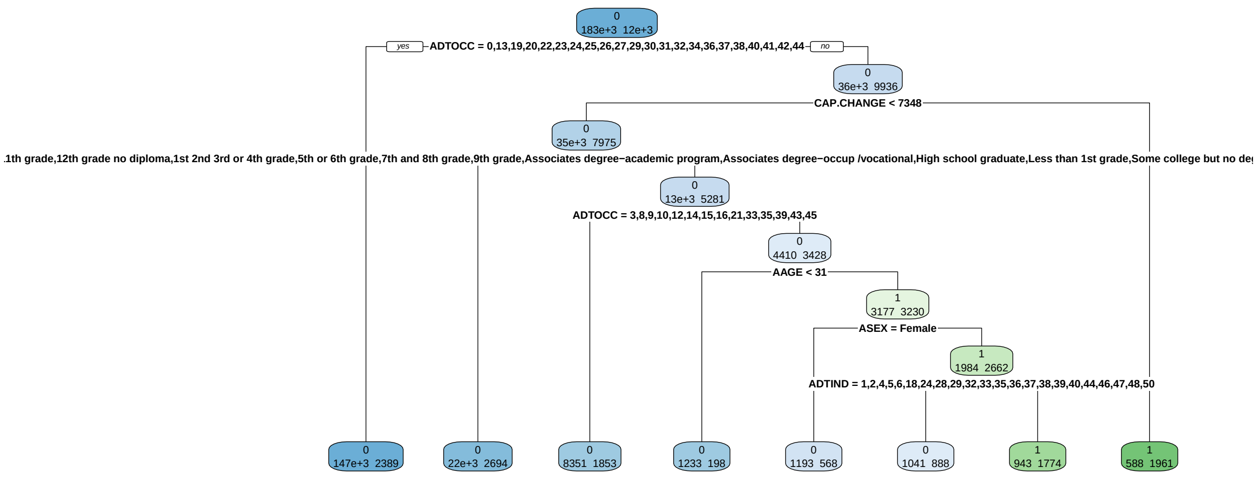


Figure 4.1: ROC and AUC - Logistic Regression

is occupation code. Yes for some of the codes goes direct to a leaf node, where the observation of income < \$50,000 is presented on the right (see Figure 4.2). In order to compare the performance of the models, we also present the Confusion Matrix for the decision tree model as follow.



0
183e+3 12e+3
yes
ADTOCC = 0,13,19,20,22,23,24,25,26,27,29,30,31,32,34,36,37,38,40,41,42,44
no
0
36e+3 9936
CAP.CHANGE < 7348
0
35e+3 7975
.1th grade,12th grade no diploma,1st 2nd 3rd or 4th grade,5th or 6th grade,7th and 8th grade,9th grade,Associates degree−academic program,Associates degree−occup /vocational,High school graduate,Less than 1st grade,Some college but no de
0
13e+3 5281
ADTOCC = 3,8,9,10,12,14,15,16,21,33,35,39,43,45
0
4410 3428
AAGE < 31
1
3177 3230
ASEX = Female
1
1984 2662
ADTIND = 1,2,4,5,6,18,24,28,29,32,33,35,36,37,38,39,40,44,46,47,48,50
0
147e+3 2389
0
22e+3 2694
0
8351 1853
0
1233 198
0
1193 568
0
1041 888
1
943 1774
1
588 1961

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction     0     1
##          0 91562  4369
##          1   747  1801
##
##                Accuracy : 0.948
##                  95% CI : (0.9466, 0.9494)
##     No Information Rate : 0.9373
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.3909
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9919
##             Specificity : 0.2919
##          Pos Pred Value : 0.9545
##          Neg Pred Value : 0.7068
##              Prevalence : 0.9373
##          Detection Rate : 0.9298
##    Detection Prevalence : 0.9741
##       Balanced Accuracy : 0.6419
##
##        'Positive' Class : 0
##
```

Table 4.2: Confusion Matrix (%)

	0	1
0	0.930	0.008
1	0.044	0.018

4.2.1 ROC and AUC

```
## A performance instance
##   'False positive rate' vs. 'True positive rate' (alpha: 'Cutoff')
##   with 3 data points
```

```
## [1] 0.6419019
```

4.3 Naïve Bayes

Naïve Bayes is a probabilistic classification method based on Bayes' theorem A naïve Bayes classifier assumes that the presence or absence of a particular feature of a class is unrelated to the presence or absence of other features. In our case, the observation that has income $>$ \$50,000 has a set of features that can be related one another, for example, education and class of worker, but this model consider all this characteristics contribute independently to the probability of the observation be or not $>$ \$50,000. This mode has a very low rate of Type II error, where the prediction set Income $<$ \$50,000 but in really is bigger. And a much higher Type I error.

```
## Confusion Matrix and Statistics
```

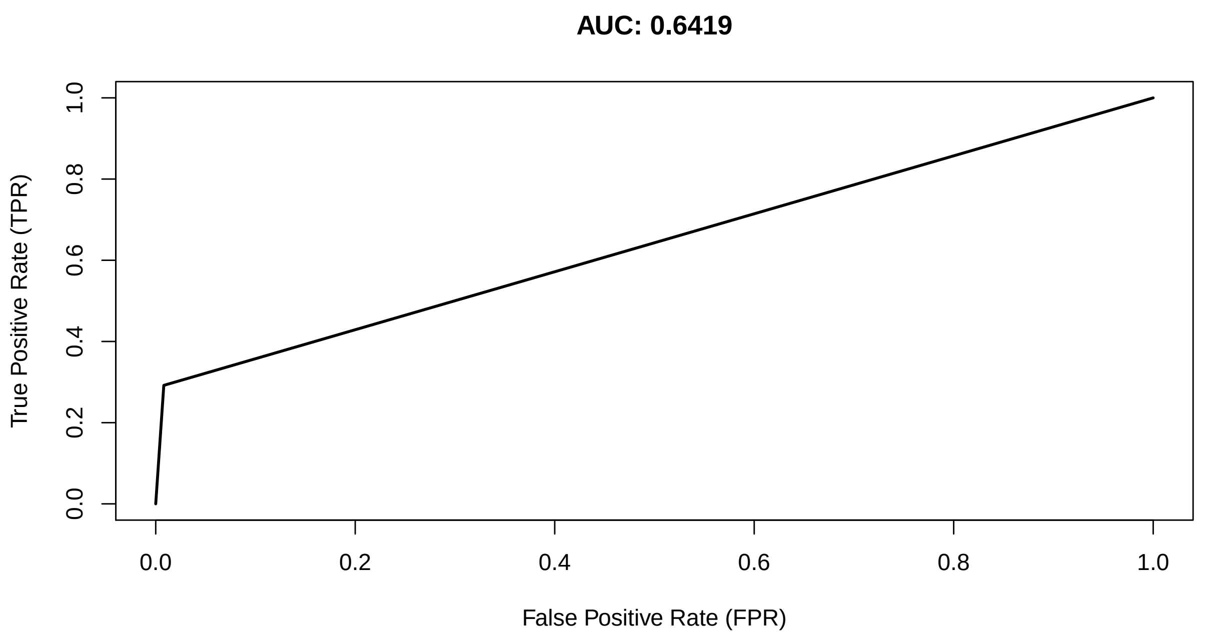


Figure 4.2: ROC and AUC - Decision Tree

```
##
##           Reference
## Prediction     0     1
##          0 73507   801
##          1 18802  5369
##
##                Accuracy : 0.8009
##                  95% CI : (0.7984, 0.8034)
##     No Information Rate : 0.9373
##     P-Value [Acc > NIR] : 1
##
##                   Kappa : 0.2823
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.7963
##             Specificity : 0.8702
##          Pos Pred Value : 0.9892
##          Neg Pred Value : 0.2221
##              Prevalence : 0.9373
##          Detection Rate : 0.7464
##    Detection Prevalence : 0.7546
##       Balanced Accuracy : 0.8332
##
##        'Positive' Class : 0
```

```
##
```

Table 4.3: Confusion Matrix (%)

	0	1
0	0.746	0.191
1	0.008	0.055

4.3.1 ROC and AUC

```
## A performance instance
##   'False positive rate' vs. 'True positive rate' (alpha: 'Cutoff')
##   with 3 data points
```

```
## [1] 0.8332464
```

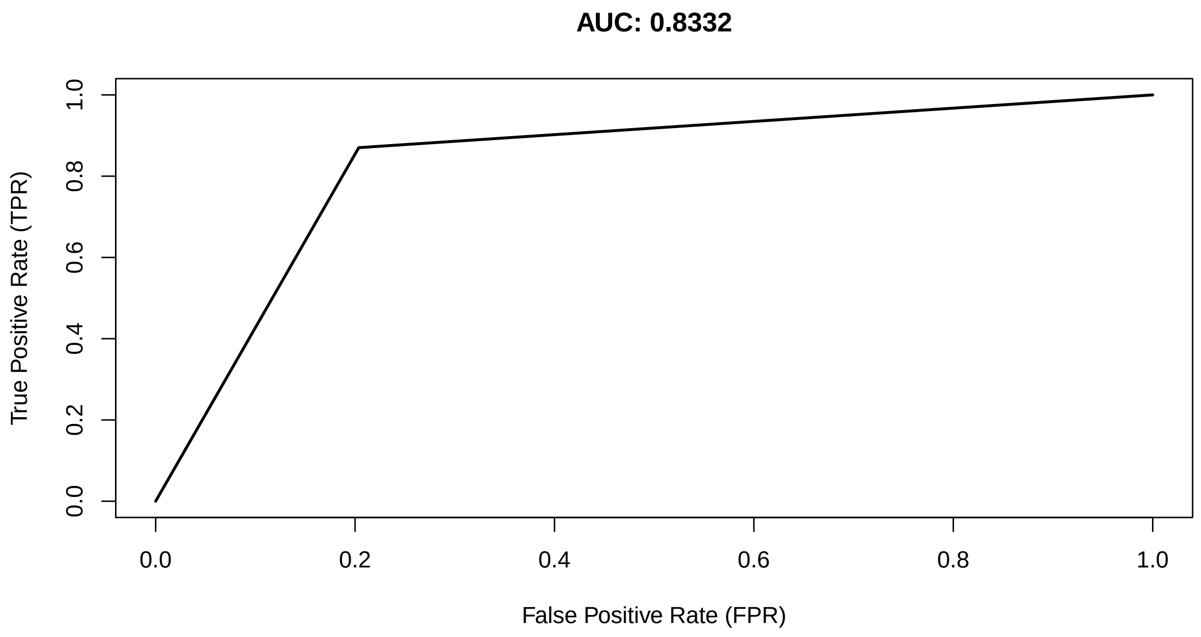


Figure 4.3: ROC and AUC - Decision Tree

Chapter 5

Conclusion

The Logistic Regression performs better than other models for our purpose. If we compare the AUC, Logistic Regression (0.9438) > Naïve Bayes (0.8332) > Decision Tree (0.6419). Comparing the Accuracy, Decision Tree (0.948) perform better than Naïve Bayes (0.8009), however the AUC of Decision tree is much lower. Logistic Regression has also a higher accuracy (0.9525). Finally it is important to highlight that the model should be applied if some specific purpose. Although the Logistic Regression had performed better than others, if the purpose would be to reduce the Type II error. As an example, one could imagine a model to charge citizens that earns more than \$50,000. In this situation, the Naïve Bayes model performs better, because has the smaller type II error and less people with more than \$50,000 is labeled wrongly and, consequently, less tax evasion. However, if we are just looking for a model that predict better without drawbacks with the type of error, our choice should be Logistic Regression.

References

EMC, Education Services. 2015. *Data Science & Big Data Analytics: Discovering, Analyzing, Visualizing and Presenting Data*. 10475 Crosspoint Boulevard Indianapolis, IN 46256: John Wiley & Sons, Inc.

Machine Learning, Center for, and UCI Machine Learning Repository Intelligent Systems. 2021. "Census-Income (Kdd) Data Set." 2021. http://archive.ics.uci.edu/ml/datasets/Census-Income+/%28KDD/%29.

Appendix

```r
knitr::opts_chunk$set(echo = TRUE)
library(data.table)
library(tidyverse)
library(R.utils)
library(rpart)
library(rpart.plot)
library(e1071)
library(lattice)
library(plyr)
library(caret)
library(ROCR)

### Read Data

column_names <- read.csv("..\\names.csv", header=FALSE) # Column names!
dt <- fread("..\\census-income.data.gz", col.names=column_names[,2],
            stringsAsFactors=TRUE)
training_df <- tibble(dt)
dt <- fread("..\\census-income.test.gz", col.names=column_names[,2],
            stringsAsFactors=TRUE)
test_df <- tibble(dt)

### Tidying data


training_df <- training_df %>%
    unique() %>% # Dropping duplicates
    mutate(AHRSPAY = AHRSPAY/100) # wage per hour in unit of dollars

training_df[training_df == '?'] = NA # Replace "?" by NA


test_df <- test_df %>%
  unique() %>% # Dropping duplicates
  mutate(AHRSPAY = AHRSPAY/100) # wage per hour in unit of dollars
```

```r
test_df[test_df == '?'] = NA # Replace "?" by NA


ag_df <- rbind(training_df, test_df) # One df for statistical description


# The instance weight (MARSUPWT) indicates the number of people in the population
# that each record represents due to stratified sampling.
# To do real analysis and derive conclusions, this field must be used.
# This attribute should *not* be used in the classifiers, so it is
# set to "ignore" in this file.


### Statistical description

summary(ag_df)

# NA's - Drop all because there is more NA's than not NA
# MIGMTR1 - 147485
# MIGMTR3 - 147485
# MIGMTR4 - 147485
# MIGSUN - 147485

ag_df <- ag_df %>%
  select(-c(MIGMTR1, MIGMTR3, MIGMTR4, MIGSUN, MARSUPWT, year))

training_df <- training_df %>%
  select(-c(MIGMTR1, MIGMTR3, MIGMTR4, MIGSUN, MARSUPWT, year))

test_df <- test_df %>%
  select(-c(MIGMTR1, MIGMTR3, MIGMTR4, MIGSUN, MARSUPWT, year))

# Num but Factor ADTIND, ADTOCC, SEOTR, VETYN + Label change 0, 1 for income

ag_df <- ag_df %>%
  mutate(ADTIND = as.factor(ADTIND)) %>%
  mutate(ADTOCC = as.factor(ADTOCC)) %>%
  mutate(SEOTR = as.factor(SEOTR)) %>%
  mutate(VETYN = as.factor(VETYN)) %>%
  mutate(TOTINC = fct_recode(TOTINC, "0" = "- 50000.", "1" = "50000+." ))

training_df <- training_df %>%
  mutate(ADTIND = as.factor(ADTIND)) %>%
  mutate(ADTOCC = as.factor(ADTOCC)) %>%
  mutate(SEOTR = as.factor(SEOTR)) %>%
  mutate(VETYN = as.factor(VETYN)) %>%
  mutate(TOTINC = fct_recode(TOTINC, "0" = "- 50000.", "1" = "50000+." ))
```

```r
test_df <- test_df %>%
  mutate(ADTIND = as.factor(ADTIND)) %>%
  mutate(ADTOCC = as.factor(ADTOCC)) %>%
  mutate(SEOTR = as.factor(SEOTR)) %>%
  mutate(VETYN = as.factor(VETYN)) %>%
  mutate(TOTINC = fct_recode(TOTINC, "0" = "- 50000.", "1" = "50000+." ))


# Histograms
ag_df <- ag_df %>%
  mutate(ADTIND = as.factor(ADTIND)) %>%
  mutate(ADTOCC = as.factor(ADTOCC)) %>%
  mutate(SEOTR = as.factor(SEOTR)) %>%
  mutate(VETYN = as.factor(VETYN)) %>%
  mutate(TOTINC = as.factor(TOTINC))

nums <- unlist(lapply(ag_df, is.numeric))

ggplot(gather(ag_df[ , nums]), aes(value)) +
  geom_histogram(bins = 15) +
  facet_wrap(~key, scales = 'free_x')
# 3 Capital Variables. They have basically the same behavior, so we group it
# in only one variable CAP.CHANGE

ggplot(gather(ag_df[ , c("CAPGAIN", "CAPLOSS", "DIVVAL")]), aes(value)) +
  geom_histogram(bins = 10) +
  facet_wrap(~key, scales = 'free_x')

ag_df <- ag_df %>%
  mutate(CAP.CHANGE = CAPGAIN - CAPLOSS + DIVVAL) %>%
  select(-c(CAPGAIN, CAPLOSS, DIVVAL))

training_df <- training_df %>%
  mutate(CAP.CHANGE = CAPGAIN - CAPLOSS + DIVVAL) %>%
  select(-c(CAPGAIN, CAPLOSS, DIVVAL))

test_df <- test_df %>%
  mutate(CAP.CHANGE = CAPGAIN - CAPLOSS + DIVVAL) %>%
  select(-c(CAPGAIN, CAPLOSS, DIVVAL))


ggplot(gather(ag_df[ , "CAP.CHANGE"]), aes(value)) +
  geom_histogram(bins = 10) +
  facet_wrap(~key, scales = 'free_x')

# distribution factors
```

```r
not_nums <- unlist(lapply(ag_df, is.factor))

ggplot(gather(ag_df[ , not_nums]), aes(value)) +
  geom_histogram(stat="count") +
  facet_wrap(~key, scales = 'free_x')

str(ag_df[ , not_nums])


# PEFNTVTY, PEMNTVTY, PENATVTY, GRINST - Lot of Factor All US much bigger
# VETQVA - Not in universe 292162!
# DROP NA

summary(ag_df[ , c("PEFNTVTY", "PEMNTVTY", "PENATVTY", "GRINST")])

ag_df <- ag_df %>%
  select(-c(PEFNTVTY, PEMNTVTY, PENATVTY, GRINST, VETQVA)) %>%
  drop_na()

training_df <- training_df %>%
  select(-c(PEFNTVTY, PEMNTVTY, PENATVTY, GRINST, VETQVA)) %>%
  drop_na()

test_df <- test_df %>%
  select(-c(PEFNTVTY, PEMNTVTY, PENATVTY, GRINST, VETQVA)) %>%
  drop_na()

# Correlation Matrix between numerical variables

nums <- unlist(lapply(ag_df, is.numeric))
res <- cor(ag_df[ , nums])

knitr::kable(res, "simple", align = "lrrrrrrr", digits = 2,
             caption = "Covariance Matrix")
par(mar = c(4, 4, .1, .1))
boxplot(WKSWORK~TOTINC, data=ag_df)
boxplot(NOEMP~TOTINC, data=ag_df)#WKSWORK is highly correlated with Income
qplot(TOTINC, data = ag_df, fill=ACLSWKR) +
  facet_grid(.~ACLSWKR)
qplot(TOTINC, data = ag_df, fill=ASEX) +
  facet_grid(.~ASEX)
qplot(TOTINC, data = ag_df, fill=ARACE) +
  facet_grid(.~ARACE)
df = data.frame(Training = c("196,294", "36"), Test=c("98,879", "36"),
                row.names = c("Observations", "Variables"))
knitr::kable(df, "simple", align = "lrr", digits = 0,
             caption = "Data sets")
```

```r
#### Logistic Regression


logit_model <- glm(TOTINC~., data=training_df, binomial(link="logit"))

sum_logit <- summary(logit_model)

pseudo_r2 = 1 - (sum_logit$deviance/sum_logit$null)

varImp(logit_model)

predict_logit_model <- predict(logit_model, newdata = test_df,
                                 type = "response")

predict_logit_model_factor <- as.factor(if_else(predict_logit_model > 0.5,
                                                 1, 0))
### Results
confusionMatrix(predict_logit_model_factor, test_df$TOTINC)


knitr::kable(prop.table(table(test_df$TOTINC, predict_logit_model > 0.5)),
             "simple", align = "lrr", digits = 3,
             caption = "Confusion Matrix (%)")
# 92.85% True < 50,000; 2.4% True > 50,000
# Type II Error 3.9% < 50,000 but in reality >50,000
# Type I Error 0,88% > 50,000 but in reality < 50,000

# ROC and AUC


pred_logit <- prediction(predict_logit_model, test_df$TOTINC)

performance(pred_logit, "tpr", "fpr")

auc_log <- performance(pred_logit, "auc")
auc_log <- unlist(slot(auc_log, "y.values"))
auc_log

perf_log <- performance(pred_logit, "tpr", "fpr")

plot(perf_log, lwd=2, xlab="False Positive Rate (FPR)",
                 ylab="True Positive Rate (TPR)",
                 main=paste("AUC:", round(auc_log, 4)))
#abline(a=0, b=1, col="gray50", lty=3)
decision_tree <- rpart(TOTINC  ~.,
                      method="class", data=training_df,
                      control=rpart.control(minsplit=1),
                      parms=list(split="information"))

predict_tree <- predict(decision_tree, newdata=test_df, type='class')

summary(decision_tree)


rpart.plot(decision_tree, type=2, extra=1)
# Results
confusionMatrix(predict_tree, test_df$TOTINC)




# 92.98% True < 50,000; 1.83% True > 50,000
# Type II Error 4.44% < 50,000 but in reality >50,000
# Type I Error 0,76% > 50,000 but in reality < 50,000

knitr::kable(prop.table(table(test_df$TOTINC, predict_tree)),
             "simple", align = "lrr", digits = 3,
             caption = "Confusion Matrix (%)")
# ROC and AUC

pred_tree <- prediction(as.numeric(predict_tree), test_df$TOTINC)

performance(pred_tree, "tpr", "fpr")

auc_tree <- performance(pred_tree, "auc")
auc_tree <- unlist(slot(auc_tree, "y.values"))
auc_tree

perf <- performance(pred_tree, "tpr", "fpr")

perf <- performance(pred_tree, "tpr", "fpr")

plot(perf, lwd=2, xlab="False Positive Rate (FPR)",
     ylab="True Positive Rate (TPR)",
                 main=paste("AUC:", round(auc_tree, 4)))
####    Naïve Bayes

model_naive <- naiveBayes(TOTINC  ~., training_df)

predict_naive <- predict(model_naive, newdata=test_df)

# Results
confusionMatrix(predict_naive, test_df$TOTINC)

knitr::kable(prop.table(table(test_df$TOTINC, predict_naive)), "simple",
             align = "lrr", digits = 3,
             caption = "Confusion Matrix (%)")

# 74.64% True < 50,000; 5.45% True > 50,000
# Type II Error 0.81% < 50,000 but in reality >50,000
# Type I Error 19,09% > 50,000 but in reality < 50,000
pred_naive <- prediction(c(predict_naive), c(test_df$TOTINC))

performance(pred_naive, "tpr", "fpr")

auc_naive <- performance(pred_naive, "auc")
auc_naive <- unlist(slot(auc_naive, "y.values"))
auc_naive
perf <- performance(pred_naive, "tpr", "fpr")

plot(perf, lwd=2, xlab="False Positive Rate (FPR)",
     ylab="True Positive Rate (TPR)", main=paste("AUC:", round(auc_naive, 4)))
```