US Adult Census Income

Dawit Tsadik Berhe

1/8/2021

Contents

1. Overview

The purpose of this project is to build a model or an algorithm that would predict whether a individual's income would exceed 50K/year, based on the census data for the individual. It is based on the 1994 Census bureau database (downloaded from https://www.kaggle.com/uciml/adult-census-income). The data-set consists 32,561 observations with 15 variables. The data-set was split into two parts – the first part (90% of the data-set) will be used for training and cross-validating the model, while the second part (10%) will be used to test the model developed. I will develop four models – Logistic regression (glm), k-nearest neighbors (knn), Classification Trees (rpart) and Ensembles, and compare the result of the four models.

The report has four sections. In section 2, the data-set will be analyzed to get a better understanding and insight of the various variables and their relationships and effects on the predictions. In section 3, different models will be tested, and their results will be compared. In section 4, overall observations and lesson learned will be discussed.

2. Analysis and Model development approach

In this section I will try to explore, clean-up and analysis the data-set to get a better insights and understanding of the data, and examine the effect of the different variables on the predictions.

2.1. General data analysis and data validation.

```
#load the required libraries.
library(tidyverse)
library(caret)
library(dplyr)

#Read the data-set from my repository in github
url <- "https://raw.githubusercontent.com/dtsadik/EDX_CYO_Project/d886694086edb3b6110a971f28709f01004c29
adult<-read_csv(url)


#Explore adult data set
adult %>% as_tibble()
```

```
## # A tibble: 32,561 x 15
##      age workclass fnlwgt education education.num marital.status occupation
##    <dbl> <chr>      <dbl> <chr>              <dbl> <chr>          <chr>
##  1    90 ?          77053 HS-grad                9 Widowed        ?
##  2    82 Private   132870 HS-grad                9 Widowed        Exec-mana~
##  3    66 ?         186061 Some-col~              10 Widowed        ?
##  4    54 Private   140359 7th-8th                 4 Divorced       Machine-o~
##  5    41 Private   264663 Some-col~              10 Separated      Prof-spec~
##  6    34 Private   216864 HS-grad                9 Divorced       Other-ser~
##  7    38 Private   150601 10th                   6 Separated      Adm-cleri~
##  8    74 State-gov  88638 Doctorate             16 Never-married  Prof-spec~
##  9    68 Federal-~ 422013 HS-grad                9 Divorced       Prof-spec~
## 10    41 Private    70037 Some-col~              10 Never-married  Craft-rep~
## # ... with 32,551 more rows, and 8 more variables: relationship <chr>,
## #   race <chr>, sex <chr>, capital.gain <dbl>, capital.loss <dbl>,
## #   hours.per.week <dbl>, native.country <chr>, income <chr>
```

```
colSums(is.na(adult))
```

```
##            age      workclass         fnlwgt      education  education.num
##              0              0              0              0              0
## marital.status     occupation   relationship           race            sex
##              0              0              0              0              0
##   capital.gain   capital.loss hours.per.week native.country         income
##              0              0              0              0              0
```

```
colSums(adult == "?")
```

```
##            age      workclass         fnlwgt      education  education.num
##              0           1836              0              0              0
## marital.status     occupation   relationship           race            sex
##              0           1843              0              0              0
##   capital.gain   capital.loss hours.per.week native.country         income
##              0              0              0            583              0
```

As it can be seen from script's output above, the adult data-set has 32,561 observations and 15 variables. Each row contains individual's census data and an indication whether that individual's income is more than

the 50K/year or not. There are no n/a in any of the columns, but we have few records with "?". I will clean-up the data after I decide which columns I will keep. First I will remove the variables with no or almost no variability, since they will not add value to the prediction. I use nearZero function to identify these columns.

```
adult<-adult[-nearZeroVar(adult)]
```

"education" and "education.num" variables provide the same or similar information, so it doesn't add value to keep both variables, so I will keep only one variable (education.num). Similarity "marital.status" and "relationship" variables provide the same or similar information, so I will keep "marital.status" and remove "relationship" from the data-set.

I will keep sex, race, education.num, workclass, occupation, age, marital.status, hours.per.week, income variables in the data-set, and do data clean-up on the new data-set.

```
#keep the following variables.
adult<-adult%>% select(sex, race, education.num, workclass, occupation, age, marital.status, hours.per.

#check if we still have "?" in the remaining variables.
colSums(adult == "?")
```

```
##            sex           race  education.num      workclass     occupation
##              0              0              0           1836           1843
##            age marital.status hours.per.week         income
##              0              0              0              0
```

```
#remove records with "?"
adult<-adult[rowSums(adult == "?")==0,]

adult %>% as_tibble()
```

```
## # A tibble: 30,718 x 9
##    sex   race  education.num workclass occupation   age marital.status
##    <chr> <chr>         <dbl> <chr>     <chr>      <dbl> <chr>
##  1 Fema~ White             9 Private   Exec-mana~    82 Widowed
##  2 Fema~ White             4 Private   Machine-o~    54 Divorced
##  3 Fema~ White            10 Private   Prof-spec~    41 Separated
##  4 Fema~ White             9 Private   Other-ser~    34 Divorced
##  5 Male  White             6 Private   Adm-cleri~    38 Separated
##  6 Fema~ White            16 State-gov Prof-spec~    74 Never-married
##  7 Fema~ White             9 Federal-~ Prof-spec~    68 Divorced
##  8 Male  White            10 Private   Craft-rep~    41 Never-married
##  9 Fema~ Black            16 Private   Prof-spec~    45 Divorced
## 10 Male  White            15 Self-emp~ Prof-spec~    38 Never-married
## # ... with 30,708 more rows, and 2 more variables: hours.per.week <dbl>,
## #   income <chr>
```

```
all<-length(adult$income)  #total number of the observations
gt50<- adult %>% filter(income==">50K") %>%  count()  #total number of 50k plus incomes
gt50Perc<-mean(adult$income==">50K") #percentage of 50k plus incomes in the data-set
```

The cleaned and modified data-set has now 30,718 observations and 9 variables.

2.2. Analysis of the variables/predictors.

Here I will analyze the different variables/predictors in the data-set. The general approach I am going to follow is to compare the proportion of a variables in the whole data-set to the proportion of the same variable in the ">50K" data-set, and see if they are over or under-represented in the ">50K" income group when compared to their representation in the whole data-set.

sex

sex_group	percentage_in_data_set	percentage_in_gt50K	gt50K_over_data_set
Male	0.6767368	0.8526797	1.2599873
Female	0.3232632	0.1473203	0.4557285

As it can be seen from the above, men's proportion in the whole data-set is 0.68, while their proportion in the >50K income group is 0.85. So men are over-represented in the ">50K" income group (i.e. more than their proportion in the data-set). The right hand side column provides the proportion to how much they are over-represented in the >50K income group (i.e. proportion in the >50k divided by the proportion in the whole data-set).

race

race_group	percentage_in_data_set	percentage_in_gt50K	gt50K_over_data_set
Asian-Pac-Islander	0.0317078	0.035424837	1.1172281
White	0.8562081	0.907712418	1.0601540
Black	0.0947002	0.049411765	0.5217706
Amer-Indian-Eskimo	0.0093105	0.004444444	0.4773582
Other	0.0080734	0.003006536	0.3723983

Asian-Pac-Islander and White people are over-represented in the ">50K" income group (when compared to their proportion in the whole data-set).

education.num

education.num_group	percentage_in_data_set	percentage_in_gt50K	gt50K_over_data_set
15	0.0181652	0.0542483660	2.9863823
16	0.0129566	0.0385620915	2.9762571
14	0.0545283	0.1230065359	2.2558297
13	0.1686959	0.2844444444	1.6861375
11	0.0430041	0.0454901961	1.0578106
12	0.0332053	0.0338562092	1.0196030
10	0.2205547	0.1767320261	0.8013069
9	0.3245003	0.2129411765	0.6562126
8	0.0127938	0.0040522876	0.3167383
6	0.0270525	0.0078431373	0.2899224
4	0.0186536	0.0049673203	0.2662934
7	0.0343772	0.0078431373	0.2281491
5	0.0150726	0.0033986928	0.2254882
3	0.0098639	0.0018300654	0.1855312
2	0.0050785	0.0007843137	0.1544394

People with education.num_group of more 10 years are over-represented in the ">50K" income group when compared to their representation in the total data-set.

workclass

workclass_group	percentage_in_data_set	percentage_in_gt50K	gt50K_over_data_set
Self-emp-inc	0.0363305	0.08130719	2.2379877
Federal-gov	0.0312520	0.04849673	1.5517944
Local-gov	0.0681359	0.08065359	1.1837158
Self-emp-not-inc	0.0827202	0.09464052	1.1441037
State-gov	0.0422554	0.04614379	1.0920223
Private	0.7388502	0.64875817	0.8780646

People in Private workclass are under-represented in the ">50K" income group when compared to their representation in the total data-set.

occupation

occupation_group	percentage_in_data_set	percentage_in_gt50K	gt50K_over_data_set
Exec-managerial	0.1323654	0.257254902	1.94352092
Prof-specialty	0.1347744	0.243006536	1.80306154
Protective-serv	0.0211277	0.027581699	1.30547710
Tech-support	0.0302103	0.036993464	1.22453150
Sales	0.1188228	0.128496732	1.08141441
Craft-repair	0.1334397	0.121437908	0.91005847
Transport-moving	0.0519891	0.041830065	0.80459358
Adm-clerical	0.1227293	0.066274510	0.54000541
Machine-op-inspct	0.0651735	0.032679739	0.50142668
Farming-fishing	0.0323589	0.015032680	0.46456122
Armed-Forces	0.0002930	0.000130719	0.44615832
Handlers-cleaners	0.0445993	0.011241830	0.25206316
Other-service	0.1072661	0.017908497	0.16695393
Priv-house-serv	0.0048506	0.000130719	0.02694916

Exec-managerial,Prof-specialty,Protective-serv,Tech-support, Sales are over-represented in the ">50K" income group.

age

age_group	percentage_in_data_set	percentage_in_gt50K	gt50K_over_data_set
50	0.1623478	0.2650980392	1.63290186
40	0.2834820	0.3886274510	1.37090699
60	0.0989973	0.1294117647	1.30722479
70	0.0210300	0.0205228758	0.97588498
30	0.2405430	0.1769934641	0.73580799
90	0.0013022	0.0009150327	0.70269935
80	0.0054040	0.0037908497	0.70148988
20	0.1868937	0.0146405229	0.07833611

People in age group 50,40 & 60 (i.e. ages 35-64) are over-represented in the ">50K" income group.

marital.status

marital.status_group	percentage_in_data_set	percentage_in_gt50K	gt50K_over_data_set
Married-AF-spouse	0.0006836	0.001307190	1.9121071
Married-civ-spouse	0.4667947	0.852418301	1.8261096
Divorced	0.1386158	0.059738562	0.4309650
Widowed	0.0273455	0.010588235	0.3872017
Married-spouse-absent	0.0126636	0.004313725	0.3406402
Separated	0.0312195	0.008627451	0.2763483
Never-married	0.3226773	0.063006536	0.1952618

People in the Married-AF-spouse & Married-civ-spouse group are over-represented in the ">50K" income group.

hours.per.week

hours.per.week_group	percentage_in_data_set	percentage_in_gt50K	gt50K_over_data_set
60	0.0830458	0.147843137	1.7802609
50	0.1183345	0.199215686	1.6834959
80	0.0085292	0.012810458	1.5019528
70	0.0126310	0.018431373	1.4592137
100	0.0032554	0.004052288	1.2447817
90	0.0013022	0.001437908	1.1042418
40	0.6178137	0.573333333	0.9280037
0	0.0037112	0.001568627	0.4226763
30	0.0474315	0.015163399	0.3196907
10	0.0197278	0.005228758	0.2650445
20	0.0842177	0.020915033	0.2483448

People in hours.per.week_group of 40 hrs or less (i.e less than 44 hrs) are under-represented in the ">50K" income group.

2.3. Model developing approach

As it can be seen from the analysis in the previous section, all the 8 variables have some effect on the prediction, with some variable's effect being higher than the others. I will include all the 9 variables in the model. I will split the data set into train (90% of the data-set) and test (10% of the data-set). The train data-set will be used for both training and cross-validating the models, while the test will be used to test the final model only. Since the data-set is not very large, I have decided to assign 90% of the data set to the train data-set in order to have enough data for training and cross-validations.

3. Results

In this section we will test and compare the different models.

```
# Represent the ">50K" income group by "1" and the "<=50" income group by "0". Also change the variable
adult$income = as.factor(ifelse(adult$income=='>50K',1,0))

#split 90/10 the data-set into adult_train & adult_test data sets.
set.seed(1, sample.kind="Rounding")
adult_test_index <- as.vector(createDataPartition(adult$income, times = 1,
                                      p = 0.1, list = FALSE))
adult_train <- adult[-adult_test_index,]
adult_test <- adult[adult_test_index,]
```

\newline

glm

```
#glm
fit_glm <- train(income ~ ., method="glm", data=adult_train)
pred_glm <- predict(fit_glm, adult_test, "raw")
cm_glm<-confusionMatrix(pred_glm, factor(adult_test$income))
cm_glm$overall[["Accuracy"]]
```

```
## [1] 0.8268229
```

As it can be seen from the above, glm provides accuracy of around 0.83.

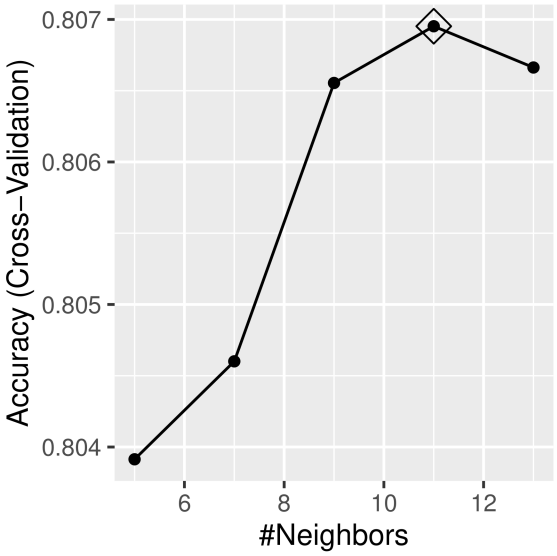
knn

```
#see what parameter can be tuned in knn.
modelLookup("knn")
```

```
##   model parameter      label forReg forClass probModel
## 1   knn         k #Neighbors   TRUE     TRUE      TRUE
```

```
# use 5-fold cross-validation to tune k.
control <- trainControl(method = "cv", number = 5, p = .9)
train_knn <- train(income ~ ., method = "knn", data = adult_train,
                   tuneGrid = data.frame(k = c(5,7,9,11,13)),
                   trControl = control)

ggplot(train_knn, highlight = TRUE)
```




```
train_knn$bestTune
```

```
##    k
## 4 11
```

```
pred_knn <- predict(train_knn, adult_test, type="raw")
cm_knn <- confusionMatrix(pred_knn, factor(adult_test$income))
cm_knn$overall[["Accuracy"]]
```

```
## [1] 0.8154297
```

As it can be seen from the above, knn provided accuracy of around 0.82, when k=11.

Classification Trees

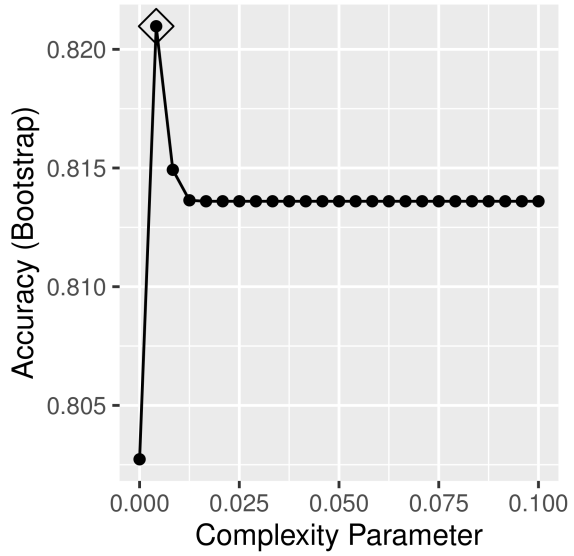
```
##Classification Trees

# see what parameters can be tuned in rpart.
modelLookup("rpart")
```

```
##   model parameter                label forReg forClass probModel
## 1 rpart        cp Complexity Parameter   TRUE     TRUE      TRUE
```

```
#cross-validate/tune cp
train_rpart <-  train(income ~ ., data = adult_train,
                      method = "rpart",
                      tuneGrid = data.frame(cp = seq(0.0, 0.1, len = 25)))

#get the best cp
ggplot(train_rpart, highlight = TRUE)
```




```
train_rpart$bestTune
```

```
##           cp
## 2 0.004166667
```

```
pred_rpart <- predict(train_rpart, adult_test, type="raw")
cm_rpart <- confusionMatrix(pred_rpart, factor(adult_test$income))
cm_rpart$overall[["Accuracy"]]
```

```
## [1] 0.8173828
```

As it can be seen from the above, rpart provided accuracy of around 0.82, when cp=0.004166667.

Ensembles Use the predictions from the previous three models and predict like what the majority predicted (i.e. like 2 out 3 model predicted).

```
## Ensembles
# predict like the majority

# change the prediction outputs to numeric as we will need to do some calculation
# on them.
pred_glm_num<-as.numeric(as.character(pred_glm))
pred_knn_num<-as.numeric(as.character(pred_knn))
pred_rpart_num<-as.numeric(as.character(pred_rpart))

# if at least two models predicts 1 (i.e. sum of three prediction is more than 1), then predict 1.
pred_ensembles<-
ifelse((pred_glm_num + pred_knn_num + pred_rpart_num) > 1, 1, 0) %>% factor(levels = levels(adult_test$
cm_ensembles<-confusionMatrix(pred_ensembles, factor(adult_test$income))
cm_ensembles$overall[["Accuracy"]]
```

```
## [1] 0.8261719
```

Ensembles provided an accuracy of around 0.83 - a little better than the other models.

The result from the four models is summarized in the following table.

```
# Print out the result from the four models
CM_accuracy %>% knitr::kable()
```

method	accuracy
glm	0.8268229
knn	0.8154297
rpart	0.8173828
ensembles	0.8261719

4. Conclusion

I was able to build four different models and compare their results. In this particular case, the glm model seem to perform better. If needed, the prediction could have be further improved, by having a larger dataset, by further tuning the model's parameters, and also by implementing more models and build ensembles model on all model's result.