Adult Census Income Capstone Report - HarvardX Data Science

Dagvadorj Galbadrakh

1 Executive summary

In order to practice data wrangling, exploratory data analysis, and model fitting, Adult Census Income [2] data set is used. The data set includes income information as classification that consists of whether the income is more than or less than 50k per annum for people whose socio-economic and demographic information is provided. The purpose of this work is to download and prepare the data, study the variables, and try to fit models that accurately predict the income based on the socio-economic and demographic predictors.

First, the data set is downloaded from the Internet and uncertain data are filtered out. Classification data are converted into factors from characters. The data set is divided into training and test sets where a random selection of 80% of the data are stored in the training set in order to train and fit models while the rest are stored in the test set so that we can validate the accuracy of each model.

After that, we explored and visualized the data and how the predictors and the outcomes relate using the ggplot library.

Finally, we started fitting models for our data set. In doing so we fit linear models and used search algorithms to understand which variables bore the model with more quality. In order to cross validate our analysis, we also used a form of decision tree called the recursive partitioning algorithm to study the importance of the variables. Based on these analyses, we futher tried out KNN, LDA, and QDA models which are suitable for the nature of the data set which has many classification predictors.

I would like to thank Mr. Irizarry and his team as well as the peers for the great opportunity of learning and validating my understanding of machine learning.

2 Data preparation

2.1 Loading data

```r
# Include libraries

if (!require(tidyverse)) install.packages("tidyverse", repos = "http://cran.us.r-project.org")
if (!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")
if (!require(data.table)) install.packages("data.table", repos = "http://cran.us.r-project.org")
if (!require(ggridges)) install.packages("ggridges", repos = "http://cran.us.r-project.org")
if (!require(ggthemes)) install.packages("ggthemes", repos = "http://cran.us.r-project.org")
if (!require(rpart.plot)) install.packages("rpart.plot", repos = "http://cran.us.r-project.org")
if (!require(MASS)) install.packages("MASS", repos = "http://cran.us.r-project.org")
if (!require(RCurl)) install.packages("RCurl", repos = "http://cran.us.r-project.org")

library(tidyverse)
```

```r
library(caret)
library(data.table)
library(ggridges)
library(ggthemes)
library(rpart.plot)
library(MASS)
library(RCurl)

# IMPORTANT Install tinytex once if not installed before and the platform
# requires if(!require(tinytex)) install.packages('tinytex', repos =
# 'http://cran.us.r-project.org') tinytex::install_tinytex()

# Download data

incomes <- read_csv("https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data",
    col_names = c("age", "workclass", "fnlwgt", "education", "education.number",
        "marital.status", "occupation", "relationship", "race", "sex", "capital.gain",
        "capital.loss", "hours.per.week", "native.country", "income"))

str(incomes)
```

```
## tibble [32,561 x 15] (S3: spec_tbl_df/tbl_df/tbl/data.frame)
##  $ age             : num [1:32561] 39 50 38 53 28 37 49 52 31 42 ...
##  $ workclass       : chr [1:32561] "State-gov" "Self-emp-not-inc" "Private" "Private" ...
##  $ fnlwgt          : num [1:32561] 77516 83311 215646 234721 338409 ...
##  $ education       : chr [1:32561] "Bachelors" "Bachelors" "HS-grad" "11th" ...
##  $ education.number: num [1:32561] 13 13 9 7 13 14 5 9 14 13 ...
##  $ marital.status  : chr [1:32561] "Never-married" "Married-civ-spouse" "Divorced" "Married-civ-spous
##  $ occupation      : chr [1:32561] "Adm-clerical" "Exec-managerial" "Handlers-cleaners" "Handlers-cle
##  $ relationship    : chr [1:32561] "Not-in-family" "Husband" "Not-in-family" "Husband" ...
##  $ race            : chr [1:32561] "White" "White" "White" "Black" ...
##  $ sex             : chr [1:32561] "Male" "Male" "Male" "Male" ...
##  $ capital.gain    : num [1:32561] 2174 0 0 0 0 ...
##  $ capital.loss    : num [1:32561] 0 0 0 0 0 0 0 0 0 0 ...
##  $ hours.per.week  : num [1:32561] 40 13 40 40 40 40 16 45 50 40 ...
##  $ native.country  : chr [1:32561] "United-States" "United-States" "United-States" "United-States" .
##  $ income          : chr [1:32561] "<=50K" "<=50K" "<=50K" "<=50K" ...
##  - attr(*, "spec")=
##   .. cols(
##   ..   age = col_double(),
##   ..   workclass = col_character(),
##   ..   fnlwgt = col_double(),
##   ..   education = col_character(),
##   ..   education.number = col_double(),
##   ..   marital.status = col_character(),
##   ..   occupation = col_character(),
##   ..   relationship = col_character(),
##   ..   race = col_character(),
##   ..   sex = col_character(),
##   ..   capital.gain = col_double(),
##   ..   capital.loss = col_double(),
##   ..   hours.per.week = col_double(),
##   ..   native.country = col_character(),
```

```
## ..   income = col_character()
## .. )
```

```r
dim(incomes)
```

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

When we glance the data we see that some entries have values noted as "?" where the value is not available. We will get rid of these data.

```r
colSums(incomes == "?")
```

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

```r
# We see that workclass, occupation, and native.country columns have '?'
# values.

incomes <- incomes %>%
    filter(!(workclass == "?" | occupation == "?" | native.country == "?"))

dim(incomes)
```

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

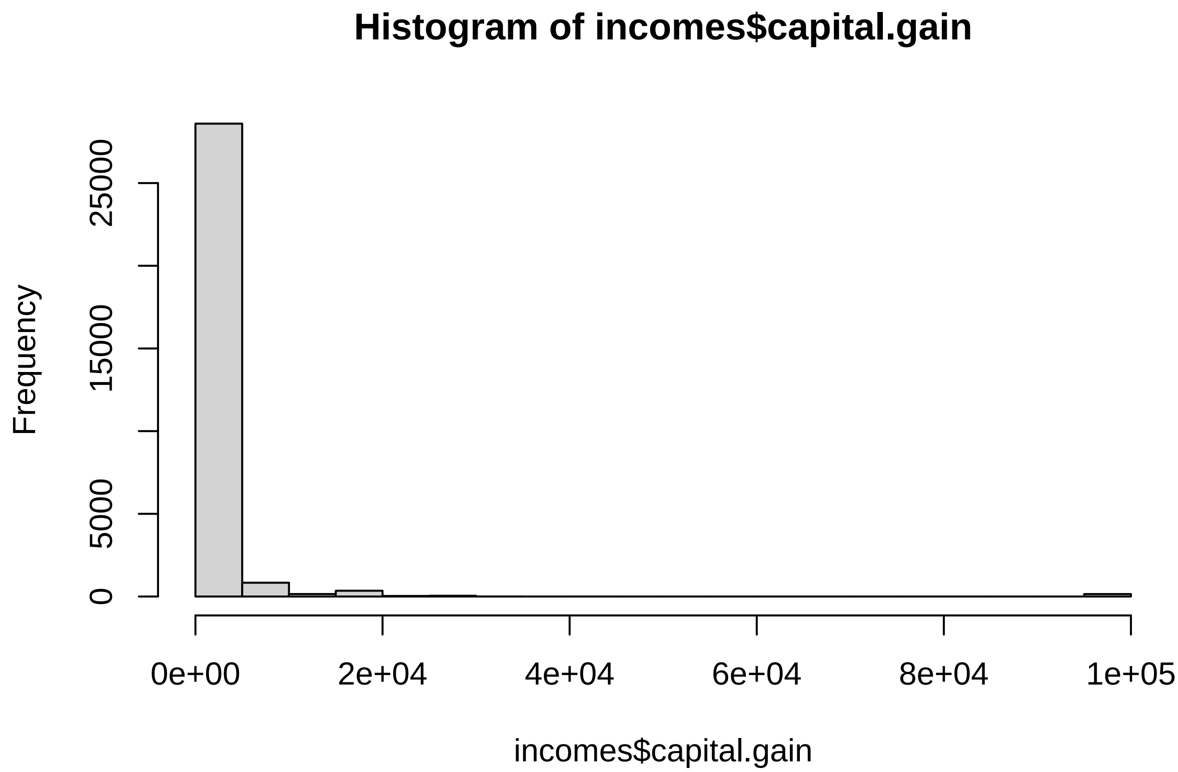
```r
colSums(incomes == "?")
```

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

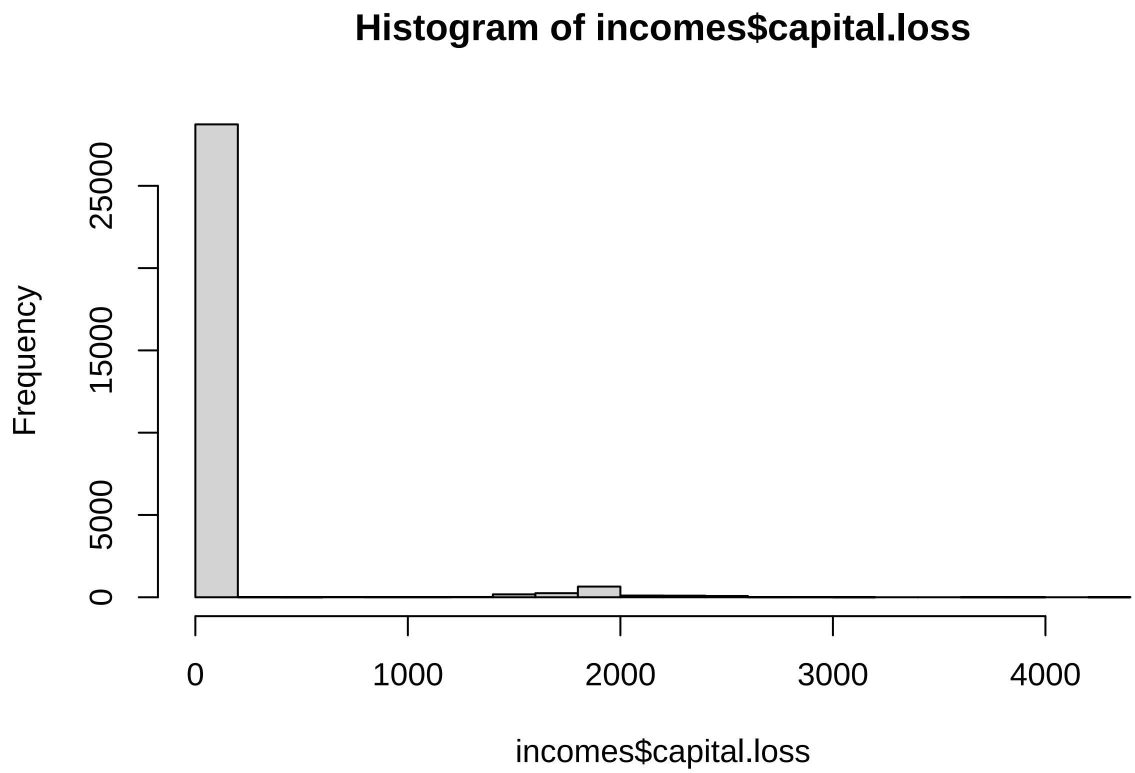
As for the predictors I decide not to use fnlwgt which is a weight that accounts for socio-economic and demographic features of individuals as calculated by CPS [2]. We are already trying to understand the effects of socio-economic and demographic features in the data set (such as age, work class, education, etc.) for the income.

Furthermore, the capital gain and capital loss predictors do not seem to characterize the data very well as there is no balance or diversification accross our data. Hence, we will not be using these predictors as well.

```r
hist(incomes$capital.gain)
```

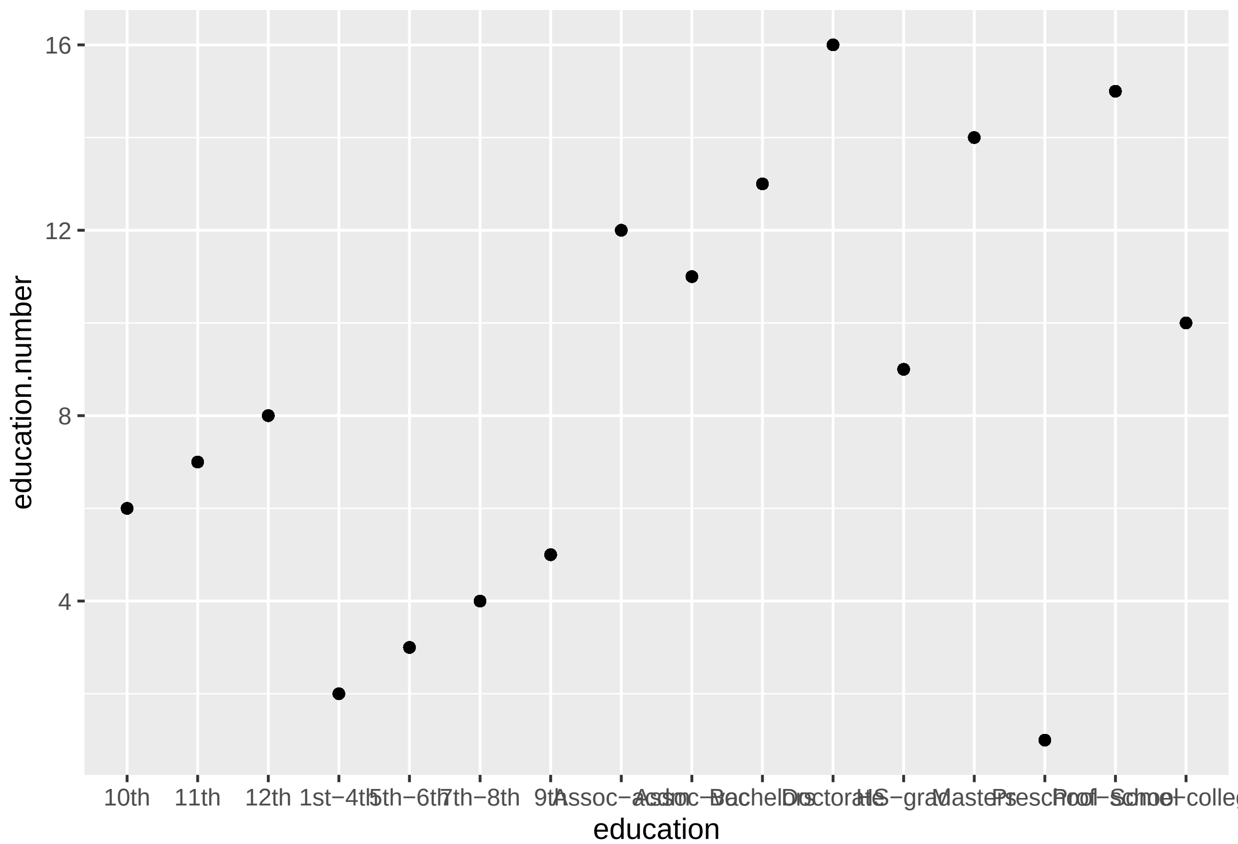



```r
hist(incomes$capital.loss)
```



Quick glance at education and education.number predictors makes it easy to see that the education.number is a one-to-one numerical representation of education. I will use education for exploratory data analysis and education.number for model fitting since education is more user-friendly because it is easilty readable and education.number is numeric and also shows the degree of education.

```
incomes %>%
    arrange(education.number) %>%
    ggplot(aes(education, education.number)) + geom_point()
```




```r
incomes <- incomes %>%
    dplyr::select(-capital.gain, -capital.loss, -fnlwgt)
```

2.2 Preparing data types

When we examine the data, we can see that some character data need to be converted to factor data type.

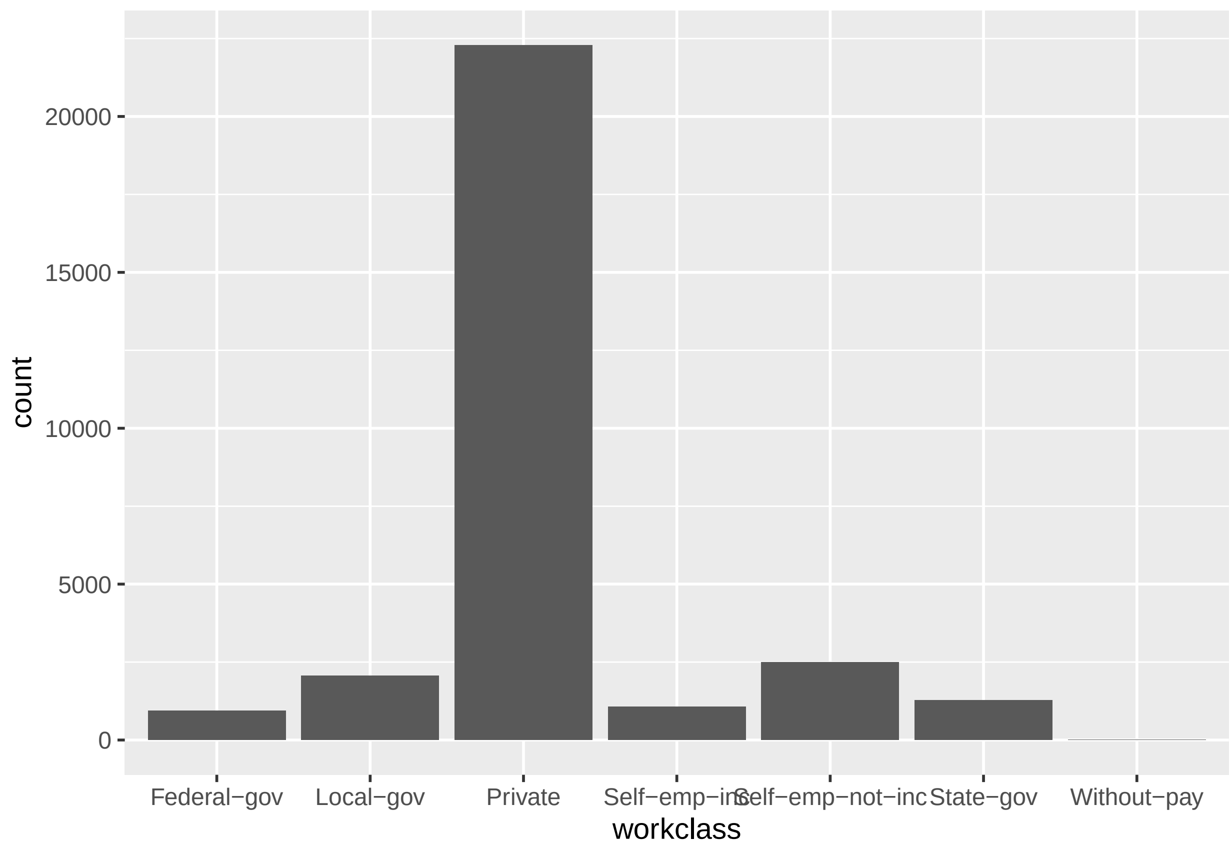
```r
summary(incomes)
```

```
##       age         workclass          education         education.number
##  Min.   :17.00   Length:30162       Length:30162       Min.   : 1.00
##  1st Qu.:28.00   Class :character   Class :character   1st Qu.: 9.00
##  Median :37.00   Mode  :character   Mode  :character   Median :10.00
##  Mean   :38.44                                         Mean   :10.12
##  3rd Qu.:47.00                                         3rd Qu.:13.00
##  Max.   :90.00                                         Max.   :16.00
##  marital.status      occupation        relationship           race
##  Length:30162       Length:30162       Length:30162       Length:30162
##  Class :character   Class :character   Class :character   Class :character
##  Mode  :character   Mode  :character   Mode  :character   Mode  :character
##
##
##
##      sex            hours.per.week  native.country        income
```

```
##  Length:30162       Min.   : 1.00   Length:30162       Length:30162      
##  Class :character   1st Qu.:40.00   Class :character   Class :character  
##  Mode  :character   Median :40.00   Mode  :character   Mode  :character  
##                     Mean   :40.93                                        
##                     3rd Qu.:45.00                                        
##                     Max.   :99.00                                        
```

For example, the workclass column consists of selection among few choices.

```r
incomes %>%
    ggplot(aes(workclass)) + geom_histogram(stat = "count")
```




```r
unique(incomes$workclass)
```

```
## [1] "State-gov"        "Self-emp-not-inc" "Private"          "Federal-gov"     
## [5] "Local-gov"        "Self-emp-inc"     "Without-pay"
```

We use the mutate function to convert the data types where necessary.

```r
incomes <- incomes %>%
    mutate(workclass = as.factor(workclass), education = as.factor(education), marital.status = as.facto
        occupation = as.factor(occupation), relationship = as.factor(relationship),
        race = as.factor(race), sex = as.factor(sex), native.country = as.factor(native.country),
        income = as.factor(income))
```

Let's examine our data set one more time.

```r
summary(incomes)
```

```
##       age                   workclass            education     education.number
##  Min.   :17.00   Federal-gov     :  943   HS-grad     :9840   Min.   : 1.00
##  1st Qu.:28.00   Local-gov       : 2067   Some-college:6678   1st Qu.: 9.00
##  Median :37.00   Private         :22286   Bachelors   :5044   Median :10.00
##  Mean   :38.44   Self-emp-inc    : 1074   Masters     :1627   Mean   :10.12
##  3rd Qu.:47.00   Self-emp-not-inc: 2499   Assoc-voc   :1307   3rd Qu.:13.00
##  Max.   :90.00   State-gov       : 1279   11th        :1048   Max.   :16.00
##                  Without-pay     :   14   (Other)     :4618
##                marital.status            occupation           relationship
##  Divorced             : 4214   Prof-specialty :4038   Husband       :12463
##  Married-AF-spouse    :   21   Craft-repair   :4030   Not-in-family : 7726
##  Married-civ-spouse   :14065   Exec-managerial:3992   Other-relative:  889
##  Married-spouse-absent:  370   Adm-clerical   :3721   Own-child     : 4466
##  Never-married        : 9726   Sales          :3584   Unmarried     : 3212
##  Separated            :  939   Other-service  :3212   Wife          : 1406
##  Widowed              :  827   (Other)        :7585
##                  race           sex        hours.per.week
##  Amer-Indian-Eskimo:  286   Female: 9782   Min.   : 1.00
##  Asian-Pac-Islander:  895   Male  :20380   1st Qu.:40.00
##  Black             : 2817                  Median :40.00
##  Other             :  231                  Mean   :40.93
##  White             :25933                  3rd Qu.:45.00
##                                            Max.   :99.00
##
##        native.country    income
##  United-States:27504   <=50K:22654
##  Mexico       :  610   >50K : 7508
##  Philippines  :  188
##  Germany      :  128
##  Puerto-Rico  :  109
##  Canada       :  107
##  (Other)      : 1516
```

We see that there is no N/A or 0 data to clean in our data set using the colSums function.

```r
colSums(is.na(incomes))
```

```
##              age        workclass        education education.number
##                0                0                0                0
##   marital.status       occupation     relationship             race
##                0                0                0                0
##              sex   hours.per.week   native.country           income
##                0                0                0                0
```

```r
colSums(incomes == 0)
```

```
##              age        workclass        education education.number
##                0                0                0                0
```

```
##   marital.status       occupation     relationship             race
##                0                0                0                0
##              sex   hours.per.week   native.country           income
##                0                0                0                0
```

2.3 Preparing the test and train sets

```
set.seed(1, sample.kind = "Rounding")
test_index <- createDataPartition(incomes$income, times = 1, p = 0.2, list = FALSE)
train_set <- incomes[-test_index, ]
test_set <- incomes[test_index, ]
```

Furthermore, let us set the fraction point to a fixed number so that the model accuracies can be easily compared.

```
options(digits = 5)
```

3 Exploratory data analysis

We can examine significant statistical figures of the column in incomes data set using the summary function.

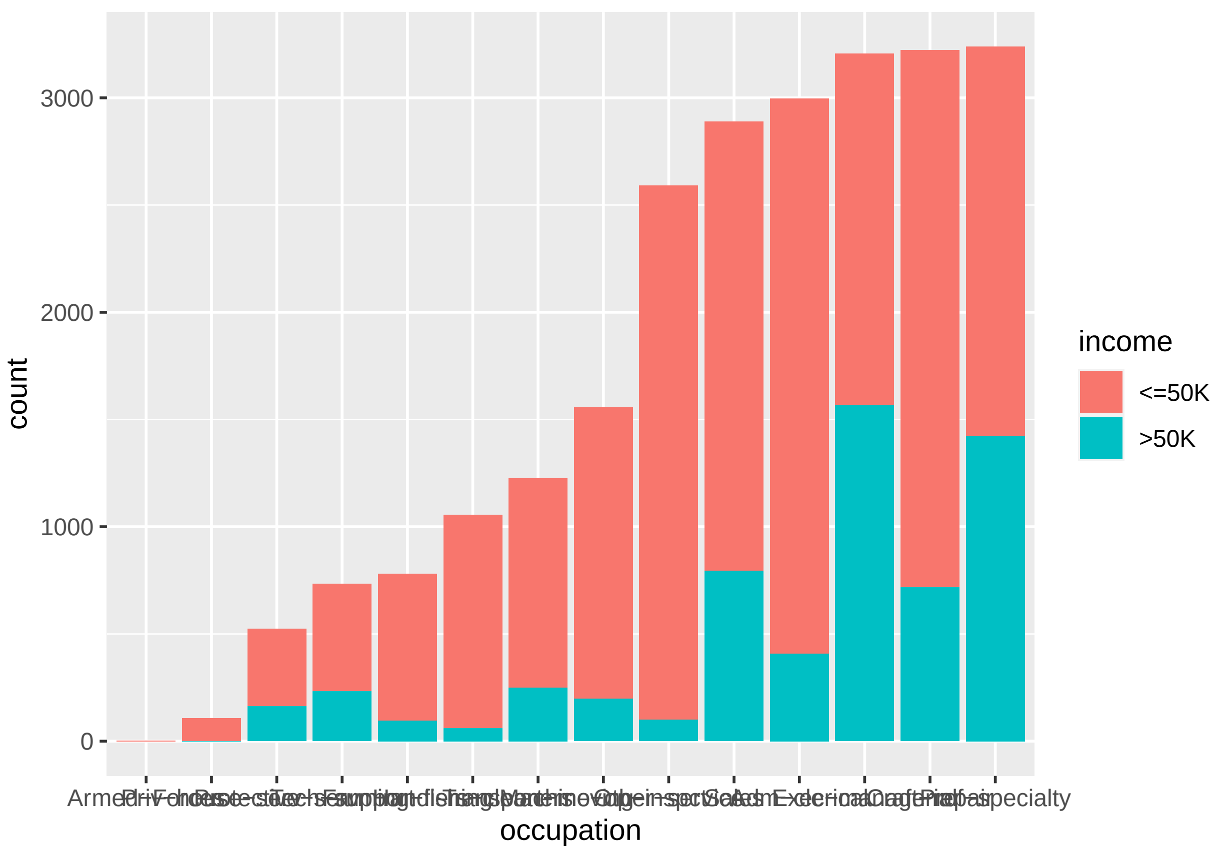
```
summary(incomes)
```

```
##       age                  workclass            education     education.number
##  Min.   :17.0   Federal-gov     :  943   HS-grad     :9840   Min.   : 1.0
##  1st Qu.:28.0   Local-gov       : 2067   Some-college:6678   1st Qu.: 9.0
##  Median :37.0   Private         :22286   Bachelors   :5044   Median :10.0
##  Mean   :38.4   Self-emp-inc    : 1074   Masters     :1627   Mean   :10.1
##  3rd Qu.:47.0   Self-emp-not-inc: 2499   Assoc-voc   :1307   3rd Qu.:13.0
##  Max.   :90.0   State-gov       : 1279   11th        :1048   Max.   :16.0
##                 Without-pay     :   14   (Other)     :4618
##                marital.status            occupation           relationship
##  Divorced             : 4214   Prof-specialty :4038   Husband       :12463
##  Married-AF-spouse    :   21   Craft-repair   :4030   Not-in-family : 7726
##  Married-civ-spouse   :14065   Exec-managerial:3992   Other-relative:  889
##  Married-spouse-absent:  370   Adm-clerical   :3721   Own-child     : 4466
##  Never-married        : 9726   Sales          :3584   Unmarried     : 3212
##  Separated            :  939   Other-service  :3212   Wife          : 1406
##  Widowed              :  827   (Other)        :7585
##                   race           sex        hours.per.week       native.country
##  Amer-Indian-Eskimo:  286   Female: 9782   Min.   : 1.0   United-States:27504
##  Asian-Pac-Islander:  895   Male  :20380   1st Qu.:40.0   Mexico       :  610
##  Black             : 2817                  Median :40.0   Philippines  :  188
##  Other             :  231                  Mean   :40.9   Germany      :  128
##  White             :25933                  3rd Qu.:45.0   Puerto-Rico  :  109
##                                            Max.   :99.0   Canada       :  107
##                                                           (Other)      : 1516
##    income
##  <=50K:22654
```

```
##  >50K : 7508
##
##
##
##
##
```

In the following diagram we see the numbers of income information for each occupation sorted by the occupation with most data to the one with least. The percentage of income factors (more than 50k and less than or equal to 50k) for occupations are different for each occupation.

```
train_set %>%
    mutate(occupation = fct_reorder(occupation, income, .fun = "length")) %>%
    ggplot(aes(occupation, fill = income)) + geom_bar()
```

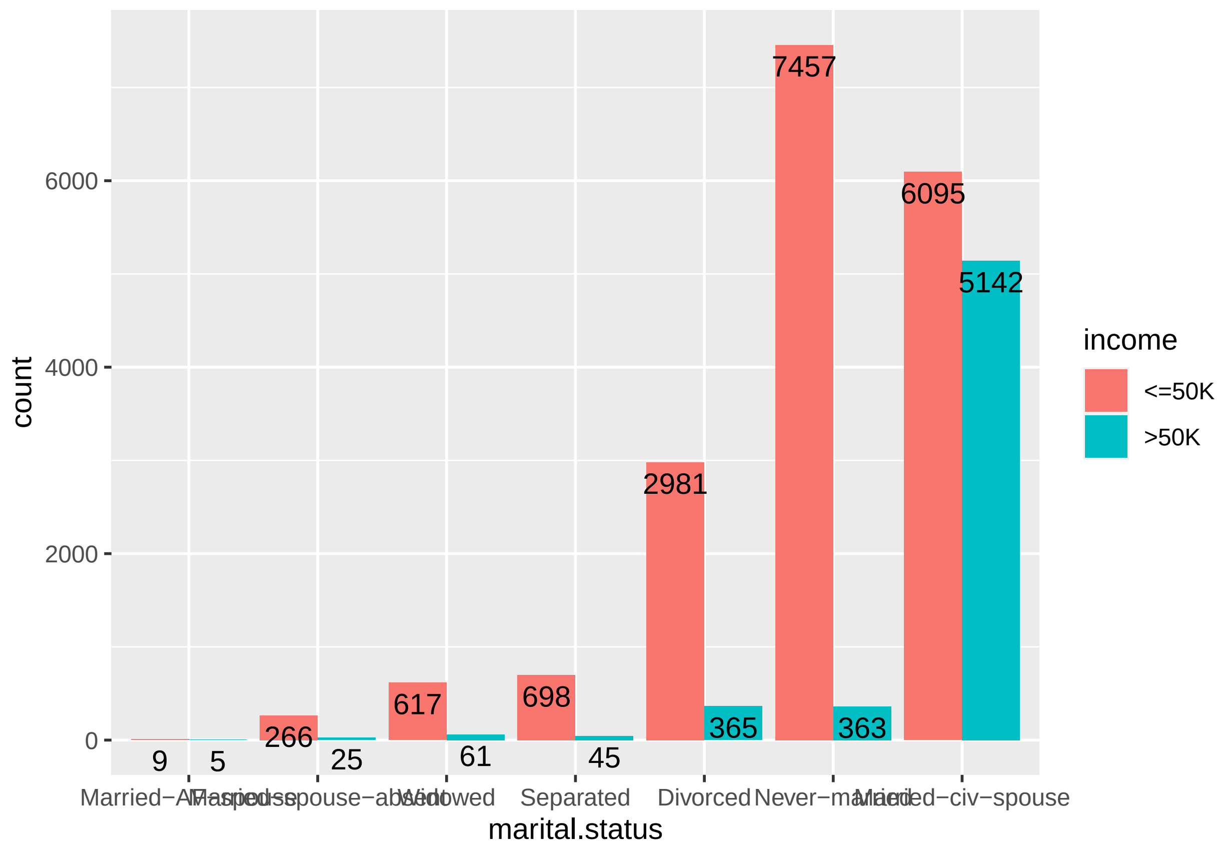


There is definitely some correlation between education and income. People with doctorate seem to have higher income regardless of the work class.

```
train_set %>%
    mutate(workclass = fct_reorder(workclass, income, .fun = "length")) %>%
    ggplot(aes(workclass, fill = income)) + geom_bar(position = "fill") + facet_wrap(~education,
    ncol = 3)
```

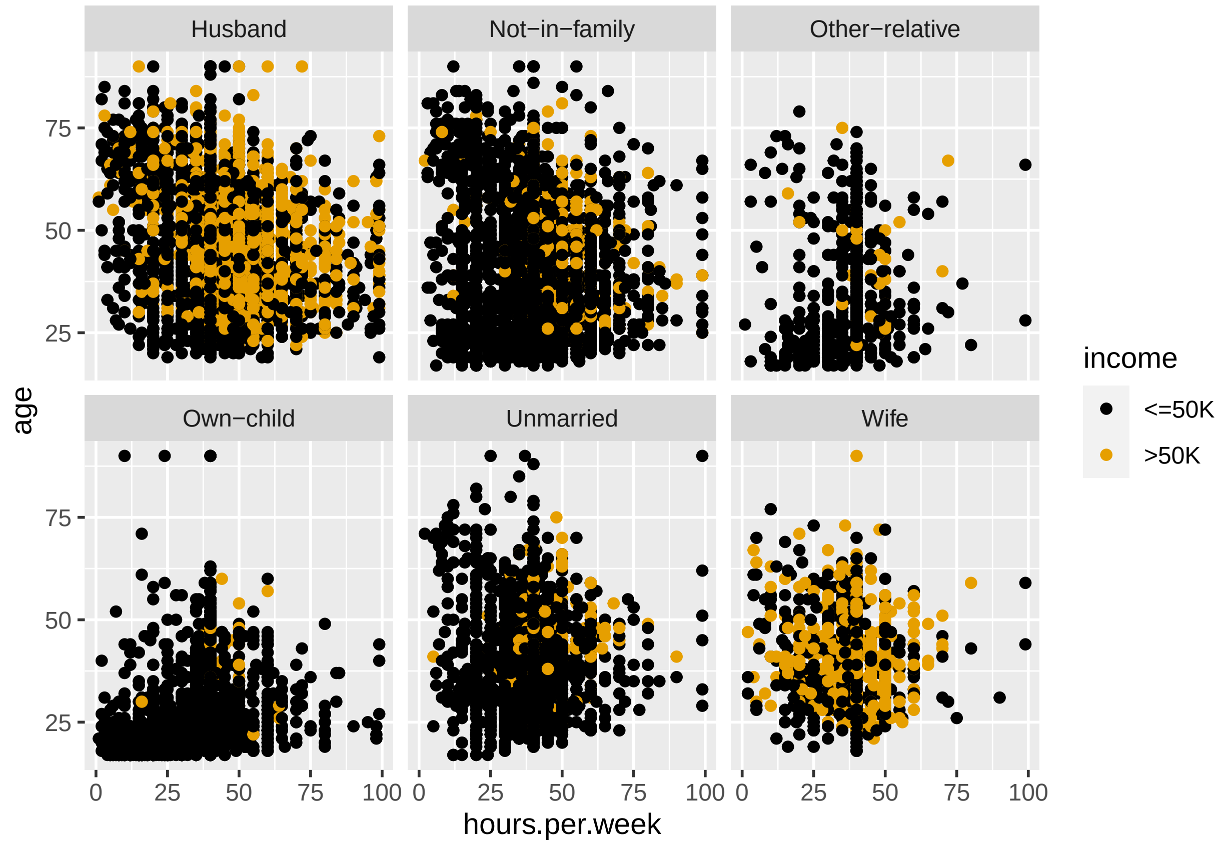
There is a higher percentage of married people who have higher income. People with some high school have less than 50k income except when they are self-employed.

```
train_set %>%
    mutate(marital.status = fct_reorder(marital.status, income, .fun = "length")) %>%
    ggplot(aes(marital.status, fill = income)) + geom_bar(stat = "count", position = "dodge") +
    geom_text(aes(label = ..count..), stat = "count", vjust = 1.5, position = position_dodge(0.9))
```



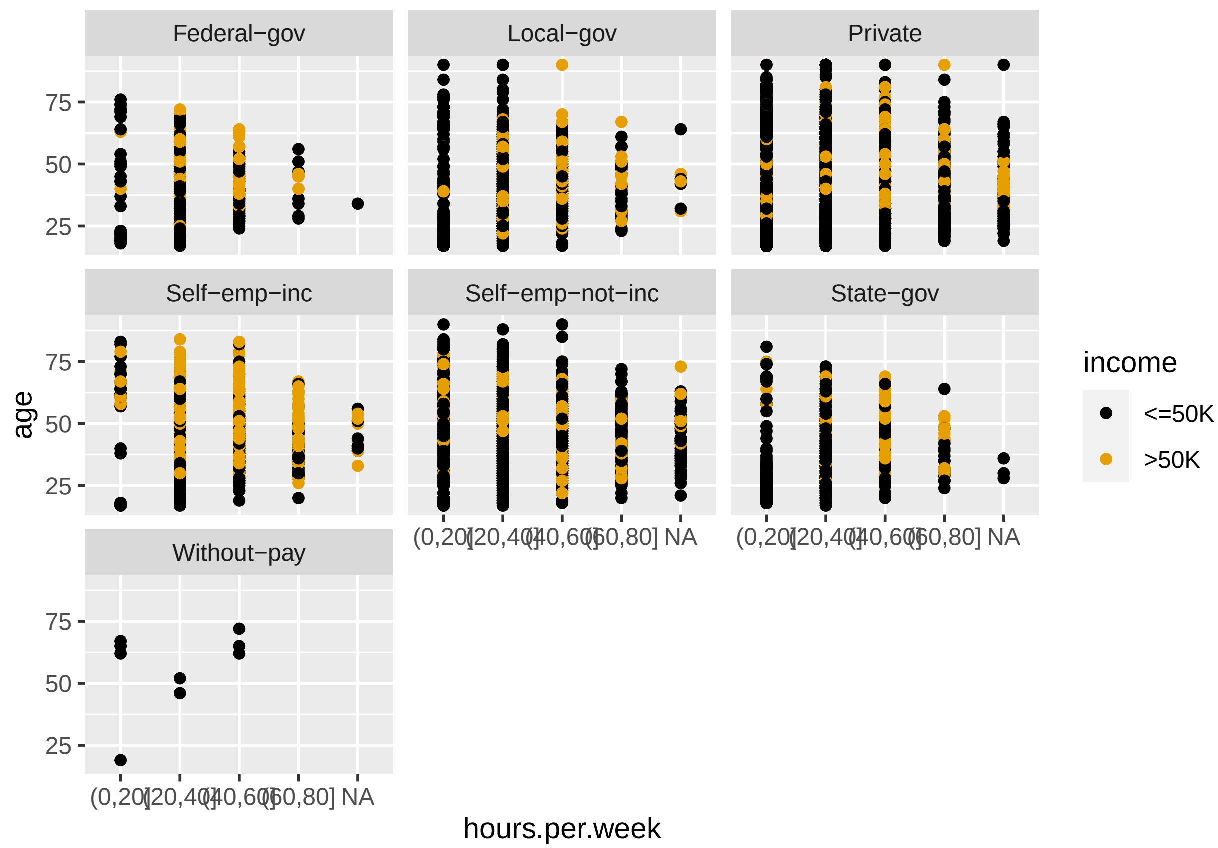
For the relationship husband the more hours worked per day the more higher income is observed. The same does not hold for the relationship wife.

```
train_set %>%
    ggplot(aes(hours.per.week, age, color = income)) + geom_point() + facet_wrap(~relationship) +
    scale_color_colorblind()
```



For government positions, there are less people with older age and the also there are less people working more hours per week. Moreover, it seems the older the age the more higher payment for the government positions are observed. There seems less correlation between age and income among self employed and private work classes.

```
train_set %>%
    mutate(hours.per.week = cut(hours.per.week, c(0, 20, 40, 60, 80))) %>%
    ggplot(aes(hours.per.week, age, color = income)) + geom_point() + facet_wrap(~workclass) +
    scale_color_colorblind()
```



4 Methods

4.1 Linear regression

Now we have 9 potential predictors. Our goal is to select the most meaningful predictors for building the best model. One way of doing this is to use step wise algorithm to test out the predictors. There are two kinds of stepwise search algorithm - backward search and forward search.

The backward search algorithm starts from a model that accounts for all predictors and tries to remove predictors one by one while not decreasing the quality of the model represented by AIC [3].

```
model.full <- glm(income ~ age + workclass + education.number + marital.status +
    occupation + relationship + race + sex + hours.per.week + native.country, data = train_set,
    family = "binomial")

model.step.backward <- stepAIC(model.full, direction = "backward")
```

```
## Start:  AIC=17276
## income ~ age + workclass + education.number + marital.status +
##     occupation + relationship + race + sex + hours.per.week +
##     native.country
##
##                      Df Deviance   AIC
## - native.country   40    17195 17273
```

```
## <none>                   17118 17276
## - race              4    17126 17276
## - workclass         6    17215 17361
## - marital.status    6    17217 17363
## - sex               1    17239 17395
## - relationship      5    17369 17517
## - age               1    17389 17545
## - hours.per.week    1    17404 17560
## - occupation       13    17691 17823
## - education.number  1    18011 18167
##
## Step:  AIC=17273
## income ~ age + workclass + education.number + marital.status +
##     occupation + relationship + race + sex + hours.per.week
##
##                    Df Deviance   AIC
## <none>                   17195 17273
## - race              4    17209 17279
## - marital.status    6    17294 17360
## - workclass         6    17295 17361
## - sex               1    17316 17392
## - relationship      5    17443 17511
## - age               1    17472 17548
## - hours.per.week    1    17483 17559
## - occupation       13    17776 17828
## - education.number  1    18123 18199
```

```
model.step.backward
```

```
##
## Call:  glm(formula = income ~ age + workclass + education.number + marital.status +
##     occupation + relationship + race + sex + hours.per.week,
##     family = "binomial", data = train_set)
##
## Coefficients:
##                         (Intercept)                                  age
##                             -9.0732                               0.0295
##                   workclassLocal-gov                     workclassPrivate
##                             -0.6339                              -0.4269
##                workclassSelf-emp-inc            workclassSelf-emp-not-inc
##                             -0.1750                              -0.8438
##                   workclassState-gov                 workclassWithout-pay
##                             -0.7962                             -12.8672
##                     education.number       marital.statusMarried-AF-spouse
##                              0.2925                               2.0003
##    marital.statusMarried-civ-spouse  marital.statusMarried-spouse-absent
##                              2.1248                              -0.1123
##         marital.statusNever-married              marital.statusSeparated
##                             -0.4966                              -0.2647
##               marital.statusWidowed                occupationArmed-Forces
##                              0.0398                             -10.6126
##              occupationCraft-repair              occupationExec-managerial
##                              0.0227                               0.8113
##           occupationFarming-fishing          occupationHandlers-cleaners
```

```
##                           -0.9371                            -0.8187
##        occupationMachine-op-inspct          occupationOther-service
##                           -0.3286                            -1.0371
##          occupationPriv-house-serv         occupationProf-specialty
##                           -2.5933                             0.5484
##          occupationProtective-serv                  occupationSales
##                            0.4303                             0.3072
##             occupationTech-support       occupationTransport-moving
##                            0.6283                            -0.1275
##          relationshipNot-in-family       relationshipOther-relative
##                            0.6074                            -0.2586
##              relationshipOwn-child            relationshipUnmarried
##                           -0.5740                             0.4276
##                   relationshipWife           raceAsian-Pac-Islander
##                            1.3595                             0.3751
##                          raceBlack                        raceOther
##                            0.4728                            -0.1925
##                          raceWhite                          sexMale
##                            0.5413                             0.8908
##                     hours.per.week
##                            0.0298
##
## Degrees of Freedom: 24128 Total (i.e. Null);  24090 Residual
## Null Deviance:       27100
## Residual Deviance: 17200     AIC: 17300
```

The backward search algorithm removes only the native.country predictor and leaves out the other nine predictors: age + workclass + education.number + marital.status + occupation + relationship + sex + hours.per.week + race

The forward search algorithm starts from a model without any predictors and tries to add predictors one by one while increasing the quality of the model represented by AIC [3].

```
model.step.forward <- stepAIC(glm(income ~ 1, data = train_set, family = "binomial"),
    direction = "forward", scope = income ~ age + workclass + education.number +
        marital.status + occupation + relationship + race + sex + hours.per.week +
        native.country)
```

```
## Start:  AIC=27081
## income ~ 1
##
##                    Df Deviance   AIC
## + relationship      5    21435 21447
## + marital.status    6    21679 21693
## + occupation       13    23975 24003
## + education.number  1    24171 24175
## + age               1    25686 25690
## + sex               1    25785 25789
## + hours.per.week    1    25789 25793
## + workclass         6    26494 26508
## + race              4    26801 26811
## + native.country   40    26767 26849
## <none>                   27079 27081
##
```

```
## Step:  AIC=21447
## income ~ relationship
##
##                    Df Deviance   AIC
## + education.number  1    18851 18865
## + occupation       13    19088 19126
## + hours.per.week    1    20953 20967
## + workclass         6    21142 21166
## + age               1    21175 21189
## + native.country   40    21157 21249
## + marital.status    6    21295 21319
## + sex               1    21341 21355
## + race              4    21347 21367
## <none>                   21435 21447
##
## Step:  AIC=18865
## income ~ relationship + education.number
##
##                  Df Deviance   AIC
## + occupation     13    18127 18167
## + age             1    18541 18557
## + hours.per.week  1    18542 18558
## + marital.status  6    18672 18698
## + workclass       6    18684 18710
## + sex             1    18731 18747
## + race            4    18815 18837
## + native.country 40    18743 18837
## <none>                 18851 18865
##
## Step:  AIC=18167
## income ~ relationship + education.number + occupation
##
##                  Df Deviance   AIC
## + age             1    17866 17908
## + hours.per.week  1    17867 17909
## + marital.status  6    17964 18016
## + sex             1    17995 18037
## + workclass       6    18012 18064
## + race            4    18105 18153
## + native.country 40    18036 18156
## <none>                 18127 18167
##
## Step:  AIC=17908
## income ~ relationship + education.number + occupation + age
##
##                  Df Deviance   AIC
## + hours.per.week  1    17528 17572
## + sex             1    17721 17765
## + workclass       6    17751 17805
## + marital.status  6    17759 17813
## + race            4    17849 17899
## + native.country 40    17781 17903
## <none>                 17866 17908
##
```

```
## Step:  AIC=17572
## income ~ relationship + education.number + occupation + age +
##     hours.per.week
##
##                  Df Deviance   AIC
## + sex             1    17406 17452
## + workclass       6    17428 17484
## + marital.status  6    17432 17488
## + race            4    17514 17566
## + native.country 40    17445 17569
## <none>                 17528 17572
##
## Step:  AIC=17452
## income ~ relationship + education.number + occupation + age +
##     hours.per.week + sex
##
##                  Df Deviance   AIC
## + workclass       6    17308 17366
## + marital.status  6    17309 17367
## + race            4    17393 17447
## + native.country 40    17322 17448
## <none>                 17406 17452
##
## Step:  AIC=17366
## income ~ relationship + education.number + occupation + age +
##     hours.per.week + sex + workclass
##
##                  Df Deviance   AIC
## + marital.status  6    17209 17279
## + race            4    17294 17360
## + native.country 40    17225 17363
## <none>                 17308 17366
##
## Step:  AIC=17279
## income ~ relationship + education.number + occupation + age +
##     hours.per.week + sex + workclass + marital.status
##
##                  Df Deviance   AIC
## + race            4    17195 17273
## + native.country 40    17126 17276
## <none>                 17209 17279
##
## Step:  AIC=17273
## income ~ relationship + education.number + occupation + age +
##     hours.per.week + sex + workclass + marital.status + race
##
##                  Df Deviance   AIC
## <none>                 17195 17273
## + native.country 40    17118 17276
```

```
model.step.forward
```

```
##
## Call:  glm(formula = income ~ relationship + education.number + occupation +
```

```
##     age + hours.per.week + sex + workclass + marital.status +
##     race, family = "binomial", data = train_set)
##
## Coefficients:
##                     (Intercept)           relationshipNot-in-family
##                         -9.0732                              0.6074
##         relationshipOther-relative              relationshipOwn-child
##                         -0.2586                             -0.5740
##            relationshipUnmarried                   relationshipWife
##                          0.4276                              1.3595
##                 education.number            occupationArmed-Forces
##                          0.2925                            -10.6126
##           occupationCraft-repair          occupationExec-managerial
##                          0.0227                              0.8113
##         occupationFarming-fishing        occupationHandlers-cleaners
##                         -0.9371                             -0.8187
##       occupationMachine-op-inspct            occupationOther-service
##                         -0.3286                             -1.0371
##         occupationPriv-house-serv          occupationProf-specialty
##                         -2.5933                              0.5484
##         occupationProtective-serv                    occupationSales
##                          0.4303                              0.3072
##           occupationTech-support         occupationTransport-moving
##                          0.6283                             -0.1275
##                              age                     hours.per.week
##                          0.0295                              0.0298
##                          sexMale                  workclassLocal-gov
##                          0.8908                             -0.6339
##                 workclassPrivate              workclassSelf-emp-inc
##                         -0.4269                             -0.1750
##         workclassSelf-emp-not-inc                 workclassState-gov
##                         -0.8438                             -0.7962
##             workclassWithout-pay      marital.statusMarried-AF-spouse
##                        -12.8672                              2.0003
##    marital.statusMarried-civ-spouse  marital.statusMarried-spouse-absent
##                          2.1248                             -0.1123
##       marital.statusNever-married           marital.statusSeparated
##                         -0.4966                             -0.2647
##            marital.statusWidowed            raceAsian-Pac-Islander
##                          0.0398                              0.3751
##                        raceBlack                          raceOther
##                          0.4728                             -0.1925
##                        raceWhite
##                          0.5413
##
## Degrees of Freedom: 24128 Total (i.e. Null);  24090 Residual
## Null Deviance:       27100
## Residual Deviance: 17200     AIC: 17300
```

The forward search algorithm omits only the native.country predictor and adds other nine predictors: age + workclass + education.number + marital.status + occupation + relationship + sex + hours.per.week + race just like the backward search algorithm.

We will check out the accuracy of linear regression model with abovementioned predictors.

```r
model.lm0 <- train_set %>%
    train(income ~ age + workclass + education.number + marital.status + occupation +
        relationship + sex + hours.per.week + race, data = ., method = "glm")
model.lm0
```

```
## Generalized Linear Model
##
## 24129 samples
##     9 predictor
##     2 classes: '<=50K', '>50K'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 24129, 24129, 24129, 24129, 24129, 24129, ...
## Resampling results:
##
##   Accuracy   Kappa
##   0.83251   0.52236
```

```r
pred.lm0 <- predict(model.lm0, test_set)
mean(pred.lm0 == test_set$income)
```

```
## [1] 0.82844
```

Unfortunately, eight predictors are too many and we will try to fit other models.

4.2 Other models

Decision trees are a good way to understand how and in what order the output is affected by the predictors. There are several ways to construct decision trees that account for different aspects of the predictors, their relations, and independent natures.

Recursive partitioning can be used to understand the importance of the predictors. The great thing about the recursive partitioning is that it recursively try out different orders of the predictors in order to come up with the best accuracy.

The caret package includes train function can is capable of training data set using different algorithms with different tuning options. Here I will use first the rpart algorithm to construct and study the predictors and try to understand which predictor(s) have more effect on the output.

```r
model.rpart <- train_set %>%
    train(income ~ age + workclass + education.number + marital.status + occupation +
        relationship + race + sex + hours.per.week + race, data = ., method = "rpart")

pred.rpart <- predict(model.rpart, test_set)
mean(pred.rpart == test_set$income)
```

```
## [1] 0.81336
```

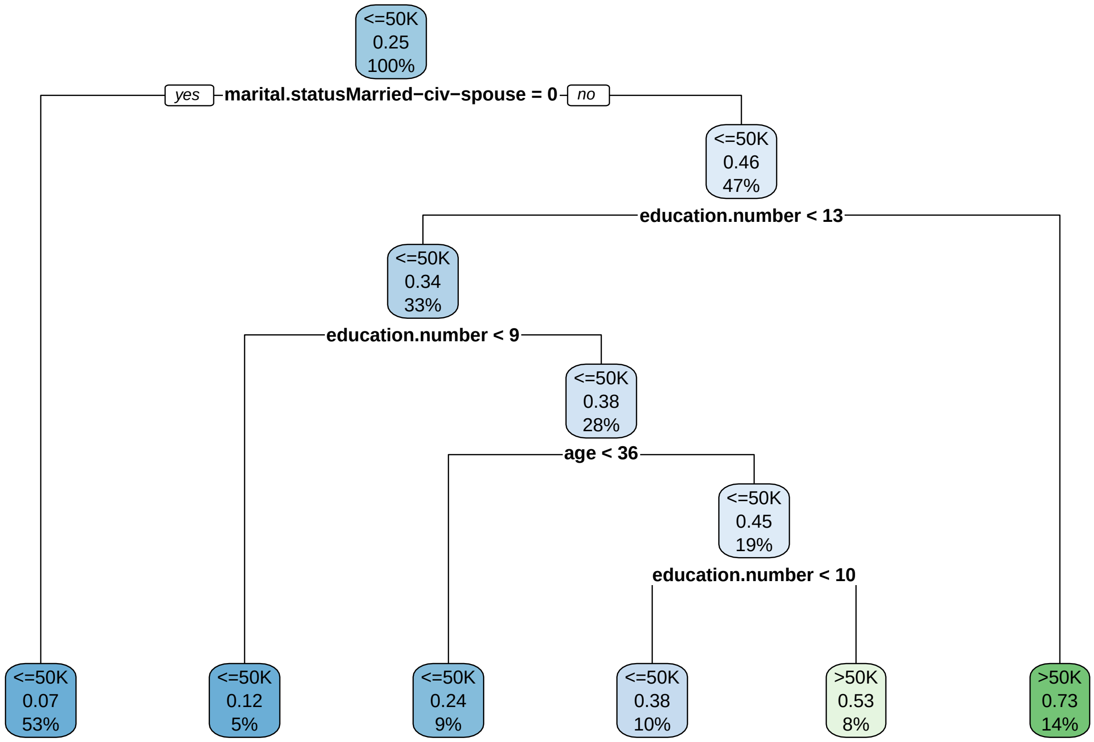
According to rpart model, the importance of the predictors for the output are:

```r
varImp(model.rpart, scale = FALSE)
```

```
## rpart variable importance
##
##   only 20 most important variables shown (out of 44)
##
##                                       Overall
## education.number                       1873.3
## marital.statusMarried-civ-spouse       1831.8
## age                                    1169.8
## marital.statusNever-married             948.8
## hours.per.week                          819.2
## occupationExec-managerial               452.7
## occupationProf-specialty                232.1
## occupationOther-service                 121.4
## workclassSelf-emp-not-inc                29.9
## `occupationFarming-fishing`               0.0
## marital.statusWidowed                     0.0
## `raceAsian-Pac-Islander`                  0.0
## `marital.statusMarried-spouse-absent`     0.0
## raceBlack                                 0.0
## `occupationProf-specialty`                0.0
## `workclassState-gov`                      0.0
## `occupationExec-managerial`               0.0
## `occupationHandlers-cleaners`             0.0
## workclassPrivate                          0.0
## `occupationProtective-serv`               0.0
```

From here we understand that the education, marital status, age, and hours per week have a higher level of importance.

```r
rpart.plot(model.rpart$finalModel)
```




```
model.rpart$results
```

```
##         cp  Accuracy   Kappa  AccuracySD  KappaSD
## 1 0.005661  0.82115 0.48813  0.0046188 0.027794
## 2 0.007770  0.81889 0.47603  0.0050778 0.035530
## 3 0.127456  0.78841 0.24799  0.0320592 0.206771
```

We note that tuning the linear regression model by modifying the predictors will not give us a better accuracy than .82778.

```
model.lm <- train_set %>%
    train(income ~ education.number + marital.status + age + hours.per.week, data = .,
        method = "glm")
model.lm
```

```
## Generalized Linear Model
##
## 24129 samples
##     4 predictor
##     2 classes: '<=50K', '>50K'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 24129, 24129, 24129, 24129, 24129, 24129, ...
## Resampling results:
```

```
##
##   Accuracy  Kappa
##   0.81827   0.47172
```

```r
pred.lm <- predict(model.lm, test_set)
mean(pred.lm == test_set$income)
```

```
## [1] 0.81121
```

```r
model.lm <- train_set %>%
    train(income ~ education.number + marital.status + age, data = ., method = "glm")
model.lm
```

```
## Generalized Linear Model
##
## 24129 samples
##     3 predictor
##     2 classes: '<=50K', '>50K'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 24129, 24129, 24129, 24129, 24129, 24129, ...
## Resampling results:
##
##   Accuracy  Kappa
##   0.81355   0.44869
```

```r
coef(model.lm)
```

```
## NULL
```

```r
pred.lm <- predict(model.lm, test_set)
mean(pred.lm == test_set$income)
```

```
## [1] 0.81319
```

```r
model.lm <- train_set %>%
    train(income ~ education.number + marital.status, data = ., method = "glm")
pred.lm <- predict(model.lm, test_set)
mean(pred.lm == test_set$income)
```

```
## [1] 0.81121
```

```r
model.lm <- train_set %>%
    train(income ~ education.number + occupation + hours.per.week * education.number,
        data = ., method = "glm")
pred.lm <- predict(model.lm, test_set)
mean(pred.lm == test_set$income)
```

```
## [1] 0.7885
```

Now let's start examining other models by using the variables of importance. K-nearest neighbors algorithm is good for examining multi-dimensional data set like ours.

```
model.knn0 <- train_set %>%
    train(income ~ education.number + marital.status + age + hours.per.week, data = .,
        method = "knn")
model.knn0
```

```
## k-Nearest Neighbors
##
## 24129 samples
##     4 predictor
##     2 classes: '<=50K', '>50K'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 24129, 24129, 24129, 24129, 24129, 24129, ...
## Resampling results across tuning parameters:
##
##   k  Accuracy  Kappa
##   5  0.79462   0.43271
##   7  0.79721   0.43735
##   9  0.79936   0.44067
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 9.
```

```
pred.knn0 <- predict(model.knn0, test_set)
mean(pred.knn0 == test_set$income)
```

```
## [1] 0.80656
```

We also note that using other sets of the important variables produce slightly better accuracy.

```
model.knn1 <- train_set %>%
    train(income ~ education.number + marital.status + age, data = ., method = "knn")
model.knn1
```

```
## k-Nearest Neighbors
##
## 24129 samples
##     3 predictor
##     2 classes: '<=50K', '>50K'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 24129, 24129, 24129, 24129, 24129, 24129, ...
## Resampling results across tuning parameters:
##
##   k  Accuracy  Kappa
##   5  0.81376   0.47764
##   7  0.81480   0.48006
```

```
##   9  0.81567   0.48271
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 9.
```

```r
pred.knn1 <- predict(model.knn1, test_set)
mean(pred.knn1 == test_set$income)
```

```
## [1] 0.81419
```

```r
model.knn2 <- train_set %>%
    train(income ~ education.number + marital.status + occupation, data = ., method = "knn")
model.knn2
```

```
## k-Nearest Neighbors
##
## 24129 samples
##     3 predictor
##     2 classes: '<=50K', '>50K'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 24129, 24129, 24129, 24129, 24129, 24129, ...
## Resampling results across tuning parameters:
##
##   k  Accuracy  Kappa
##   5  0.81888   0.48192
##   7  0.81934   0.48280
##   9  0.81944   0.48291
##
## Accuracy was used to select the optimal model using the largest value.
## The final value used for the model was k = 9.
```

```r
pred.knn2 <- predict(model.knn2, test_set)
mean(pred.knn2 == test_set$income)
```

```
## [1] 0.82082
```

We will use two more models to try to come up with a better accuracy.

```r
model.lda <- train_set %>%
    train(income ~ education.number + marital.status + age, data = ., method = "lda")
model.lda
```

```
## Linear Discriminant Analysis
##
## 24129 samples
##     3 predictor
##     2 classes: '<=50K', '>50K'
##
## No pre-processing
```

```
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 24129, 24129, 24129, 24129, 24129, 24129, ...
## Resampling results:
##
##   Accuracy  Kappa
##   0.81338   0.45963
```

```
pred.lda <- predict(model.lda, test_set)
mean(pred.lda == test_set$income)
```

```
## [1] 0.8122
```

```
model.qda <- train_set %>%
    train(income ~ education.number + marital.status + age, data = ., method = "qda")
model.qda
```

```
## Quadratic Discriminant Analysis
##
## 24129 samples
##     3 predictor
##     2 classes: '<=50K', '>50K'
##
## No pre-processing
## Resampling: Bootstrapped (25 reps)
## Summary of sample sizes: 24129, 24129, 24129, 24129, 24129, 24129, ...
## Resampling results:
##
##   Accuracy  Kappa
##   0.72188   0.41695
```

```
pred.qda <- predict(model.qda, test_set)
mean(pred.qda == test_set$income)
```

```
## [1] 0.71142
```

Results

```
table.results <- data.frame()
table.results <- rbind(table.results, data.frame(name = "Linear regression", accuracy = mean(pred.lm0 ==
    test_set$income), sensitivity = as.numeric(confusionMatrix(pred.lm0, test_set$income)$byClass["Sens
    specificity = as.numeric(confusionMatrix(pred.lm0, test_set$income)$byClass["Specificity"])))
table.results <- rbind(table.results, data.frame(name = "Recursive partitioning",
    accuracy = mean(pred.rpart == test_set$income), sensitivity = as.numeric(confusionMatrix(pred.rpart
        test_set$income)$byClass["Sensitivity"]), specificity = as.numeric(confusionMatrix(pred.rpart,
        test_set$income)$byClass["Specificity"])))
table.results <- rbind(table.results, data.frame(name = "KNN 0", accuracy = mean(pred.knn0 ==
    test_set$income), sensitivity = as.numeric(confusionMatrix(pred.knn0, test_set$income)$byClass["Sens
    specificity = as.numeric(confusionMatrix(pred.knn0, test_set$income)$byClass["Specificity"])))
table.results <- rbind(table.results, data.frame(name = "KNN 1", accuracy = mean(pred.knn1 ==
```

```
    test_set$income), sensitivity = as.numeric(confusionMatrix(pred.knn1, test_set$income)$byClass["Sens
    specificity = as.numeric(confusionMatrix(pred.knn1, test_set$income)$byClass["Specificity"])))
table.results <- rbind(table.results, data.frame(name = "KNN 2", accuracy = mean(pred.knn2 ==
    test_set$income), sensitivity = as.numeric(confusionMatrix(pred.knn2, test_set$income)$byClass["Sens
    specificity = as.numeric(confusionMatrix(pred.knn2, test_set$income)$byClass["Specificity"])))
table.results <- rbind(table.results, data.frame(name = "LDA", accuracy = mean(pred.lda ==
    test_set$income), sensitivity = as.numeric(confusionMatrix(pred.lda, test_set$income)$byClass["Sensi
    specificity = as.numeric(confusionMatrix(pred.lda, test_set$income)$byClass["Specificity"])))
table.results <- rbind(table.results, data.frame(name = "QDA", accuracy = mean(pred.qda ==
    test_set$income), sensitivity = as.numeric(confusionMatrix(pred.qda, test_set$income)$byClass["Sensi
    specificity = as.numeric(confusionMatrix(pred.qda, test_set$income)$byClass["Specificity"])))

table.results
```

```
##                     name accuracy sensitivity specificity
## 1      Linear regression  0.82844     0.92165     0.54727
## 2 Recursive partitioning  0.81336     0.89649     0.56258
## 3                  KNN 0  0.80656     0.89605     0.53662
## 4                  KNN 1  0.81419     0.90642     0.53595
## 5                  KNN 2  0.82082     0.91680     0.53129
## 6                    LDA  0.81220     0.91635     0.49800
## 7                    QDA  0.71142     0.66939     0.83822
```

Conclusion

We note that however it has has too many predictors, the linear regression model with eight predictors performed the best. We have tried to beat this model using KNN, QDA, and LDA models - models that do well with many predictors. However, in the end the linear regression model has the best accuracy. Moreover, the linear model has similar sensitivities and specificities with the other models. In other words, the model is not lagging from the other models in this area as well. If our data set included many numeric predictors in other words if the important socio-economic and demographic predictors were numeric, we would have chance to implement different analyses of clustering, matrix factorization, and component analysis as we learned in the course. I am looking forward to implement these methods and analysis for other data sets in the future. I would like to again thank Mr.Irizarry and his staff for the great opportunity.

References

[1] Irizarry. Rafael. Introduction to Data Science. 2019, found at https://leanpub.com/datasciencebook

[2] https://www.kaggle.com/uciml/adult-census-income

[3] Dalpiaz. David, Applied Statistics with R. 2021