Analysis on Adult Income Dataset using Logistic Regression and BLB

Yifei Chen, Shozen Dan, Weixiang Wang, Yuyan Wu

3/11/2020

1. Introduction

The purpose of this project is to analyze the data to find the predictors that are significant and important in classifying whether a person earns more or less than 50K a year. We will also estimate the confidence intervals of the coefficients by utilizing the bag of little bootstrap method.

Our response variable, which is an indicator of whether the person earns over 50K per year, is categorical. Thus, we will use a logistic regression model. We first use forward subset selection with AIC as our criterion to filter out unnecessary predictors. Through this process, we identified a model 5 predictors that had the smallest AIC value among other subsets of predictors.

We use the Bag of Little Bootstrap method instead of normal Bootstrap is because BLB is computationally less expensive. Although bootstrap is more accurate, it is not practical to perform bootstrap for massive datasets that are too large to store in memory. Therefore, despite the fact that bootstrap is possible for our dataset, we will use the BLB approach for academic purposes.

2. Description of the Dataset

We obtained the data from the UCI Machine Learning Repository[1]. The data extraction was conducted by Barry Becker, who worked at Silicon Graphics at the time, from the 1994 Census database. The data contains approximately 32,000 observations with over 15 variables and can be downloaded from http://archive.ics.uci.edu/ml/datasets/Adult. A brief description of each of the variables and their data type is as follows:

- Age: Age of the individual (Numerical)
- Workclass: Class of work (Categorical)
- fnlwgt: Final Weight Determined by Census Org (Numerical)
- Education: Education of the individual (Categorical)
- Education-num: Number of years of education (Numerical)
- Marital-status: Marital status of the individual (Categorical)
- Occupation: Occupation of the individual (Categorical)
- Relationship: Present relationship (Categorical)
- Race: Race of the individual (Categorical)
- Sex: Sex of the individual (Categorical)
- Capital-gain: Capital gain made by the individual (Numerical)
- Capital-loss: Capital loss made by the individual (Numerical)
- Hours-per-week: Average number of hours spent by the individual on work (Numerical)
- Native-country: Native country of origin (Categorical)

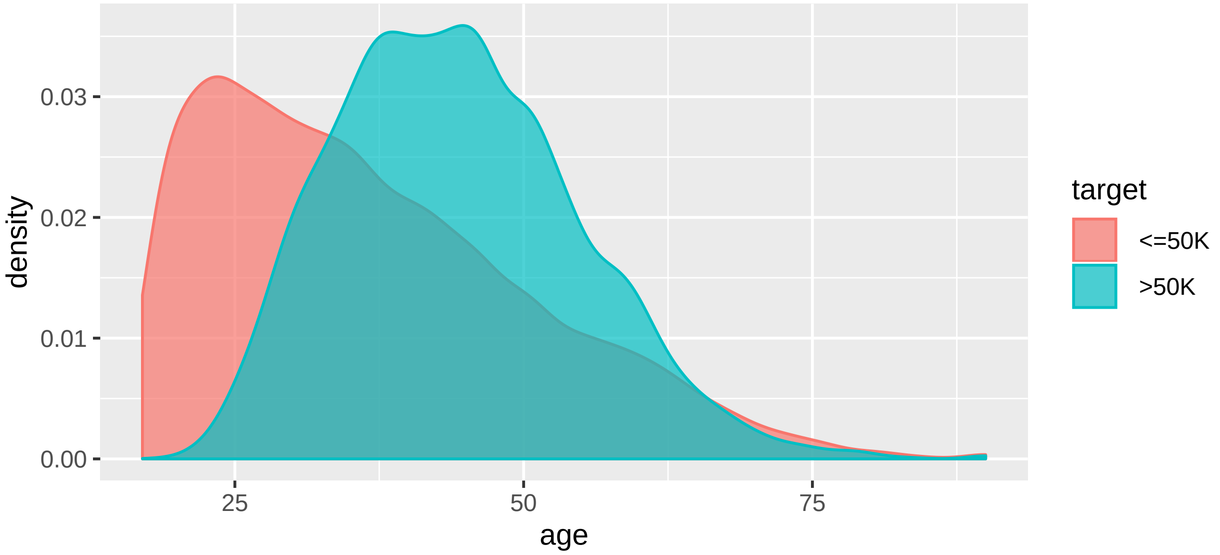
- Income Level: Whether income is over 50K or not (Categorical)

```
##       age         workclass             fnlwgt         education
##  Min.   :17.00   Length:32561       Min.   :  12285   Length:32561
##  1st Qu.:28.00   Class :character   1st Qu.: 117827   Class :character
##  Median :37.00   Mode  :character   Median : 178356   Mode  :character
##  Mean   :38.58                      Mean   : 189778
##  3rd Qu.:48.00                      3rd Qu.: 237051
##  Max.   :90.00                      Max.   :1484705
##  education.num   marital.status      occupation         relationship
##  Min.   : 1.00   Length:32561       Length:32561       Length:32561
##  1st Qu.: 9.00   Class :character   Class :character   Class :character
##  Median :10.00   Mode  :character   Mode  :character   Mode  :character
##  Mean   :10.08
##  3rd Qu.:12.00
##  Max.   :16.00
##      race               sex             capital.gain    capital.loss
##  Length:32561       Length:32561       Min.   :    0   Min.   :   0.0
##  Class :character   Class :character   1st Qu.:    0   1st Qu.:   0.0
##  Mode  :character   Mode  :character   Median :    0   Median :   0.0
##                                        Mean   : 1078   Mean   :  87.3
##                                        3rd Qu.:    0   3rd Qu.:   0.0
##                                        Max.   :99999   Max.   :4356.0
##  hours.per.week  native.country        target
##  Min.   : 1.00   Length:32561       Length:32561
##  1st Qu.:40.00   Class :character   Class :character
##  Median :40.00   Mode  :character   Mode  :character
##  Mean   :40.44
##  3rd Qu.:45.00
##  Max.   :99.00
```

The table above displays the summary statistics for the training dataset. We can see from it that there are 9 categorical variables: workclass, education, native.country, target, race, marital.status, occupation, relationship, and sex. The data also contains 6 numerical variables: age, fnlwgt, education.num, capital.gain, capital.loss, and hours.per.week.

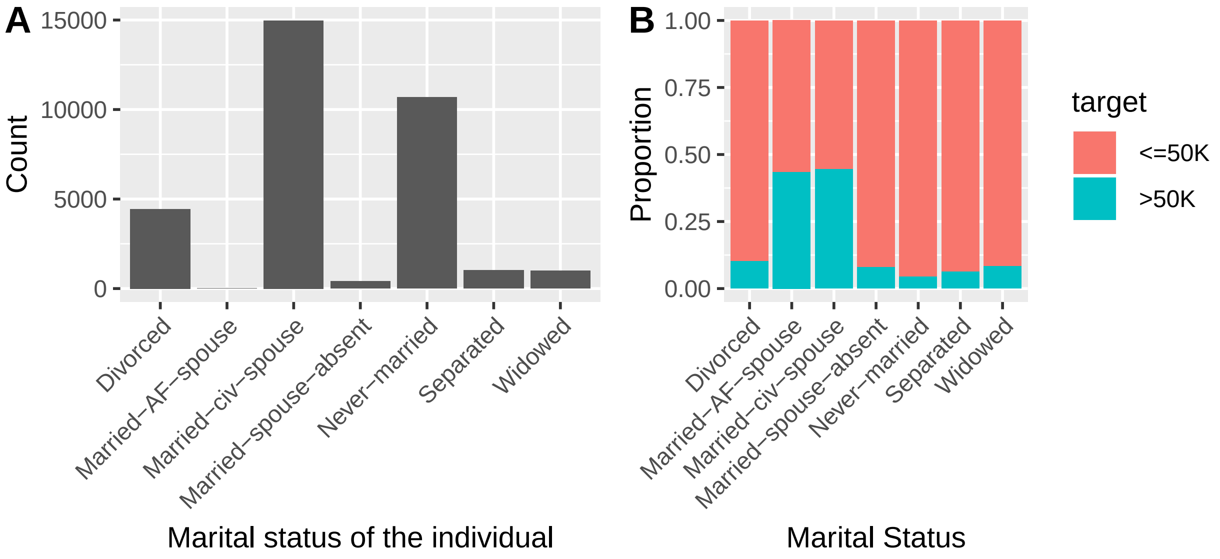
3. Exploratory Data Analysis

3.1 Age and Income



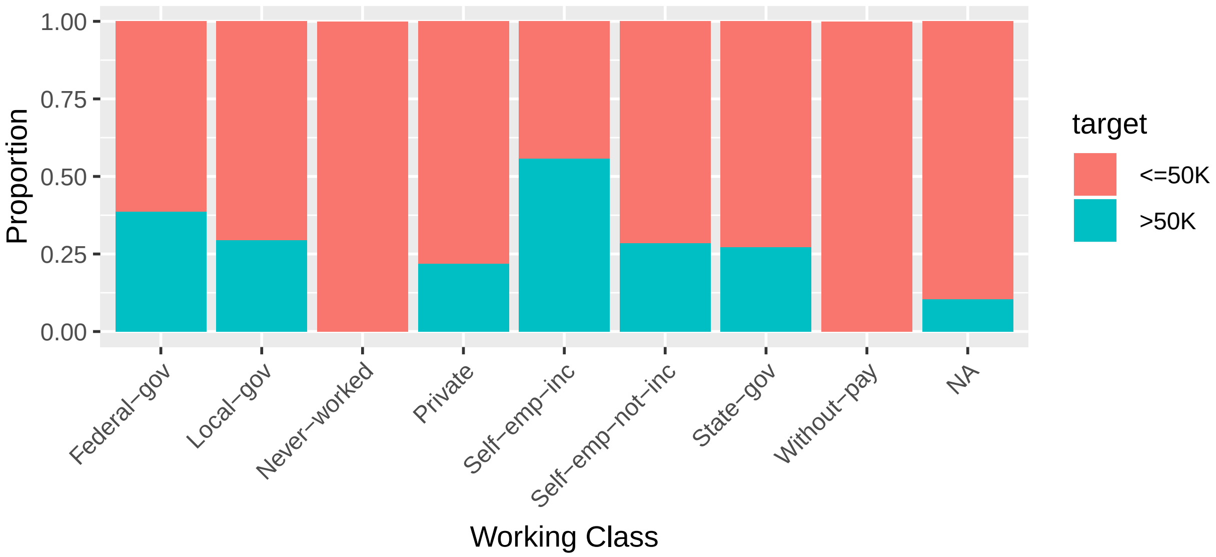
The histogram above displays the density distribution of age for the two income groups. We can see that the average age of people who earn more than 50K is higher than the average age of those who earn below 50K. This is predictable, as job salaries increase as people age due to promotions and career changes. Although the probability distribution of age does seem to differ between the two groups, there is still a large overlap between them. Therefore we need other variables to further differentiate the two groups.

3.2 Marital Status and Income



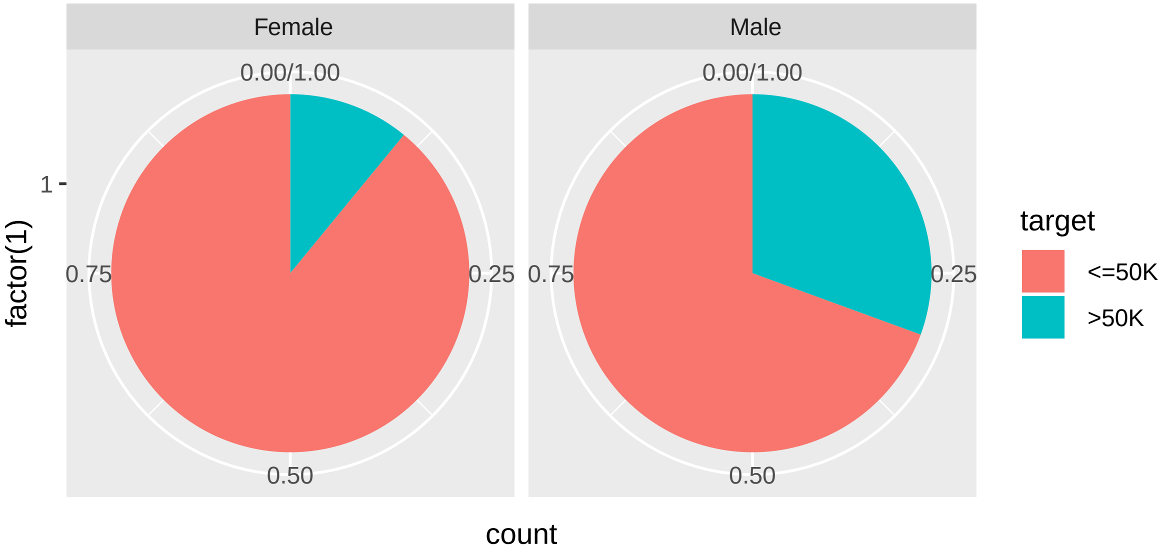
From the figure above, we can see that the proportion of people with annual incomes above 50K is especially high among the married. Both groups have around 40% over income 50K. For simplicity and interpretability, we will combine Married-AF-spouse and Married-civ-spouse into one group called “married” and the other categories into another called “separated”.

3.3 Working Class and Income



From the figure above we can see that the proportion of people with incomes over 50K is highest among the self-employed. People who work in public sectors such as federal, local, and state governments also receive high pay, with more than 25% of people with income over 50K.

3.3 Gender and Income

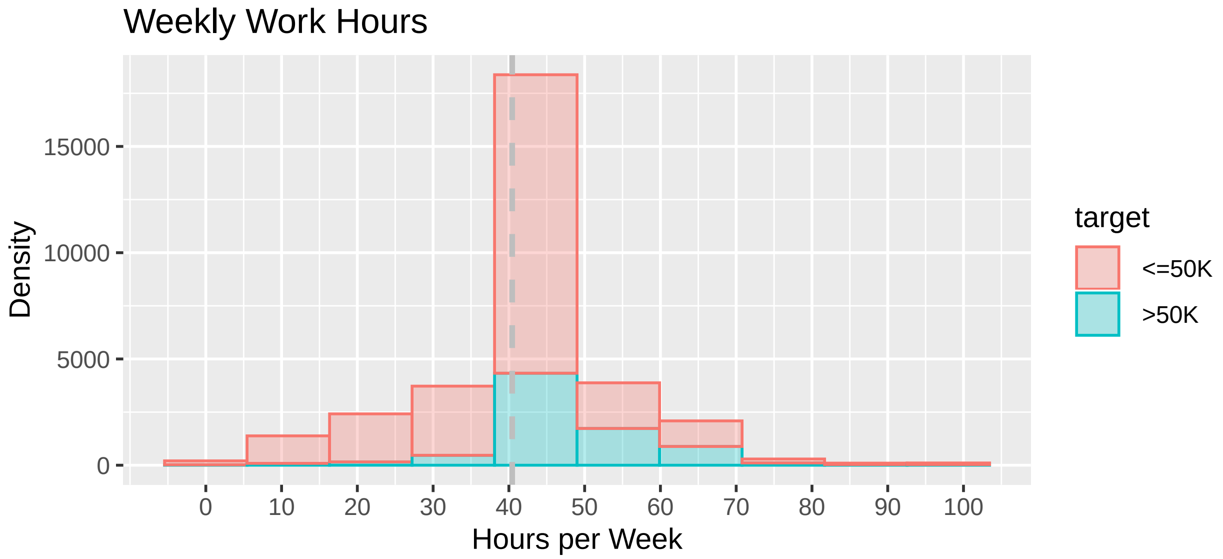


```
##
##  Pearson's Chi-squared test with Yates' continuity correction
```

```
##
## data:  train$target and train$sex
## X-squared = 1517.8, df = 1, p-value < 2.2e-16
```

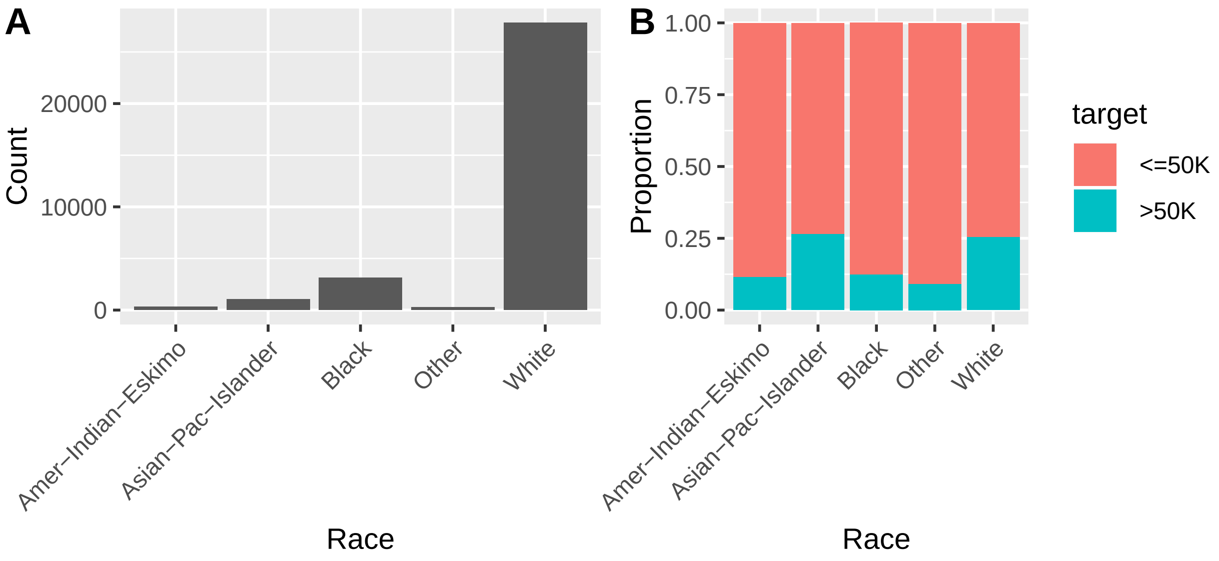
From the pie chart, we can see that the proportion of people earning over 50K is higher among men is higher than in females. Given that this data was collected in 1994 when gender equality was not as good as it is today, this was expected. The Chi-squared test for independence indicates this as well with a p-value less than 2.2^{-16}.

3.4 Work Hours and Income



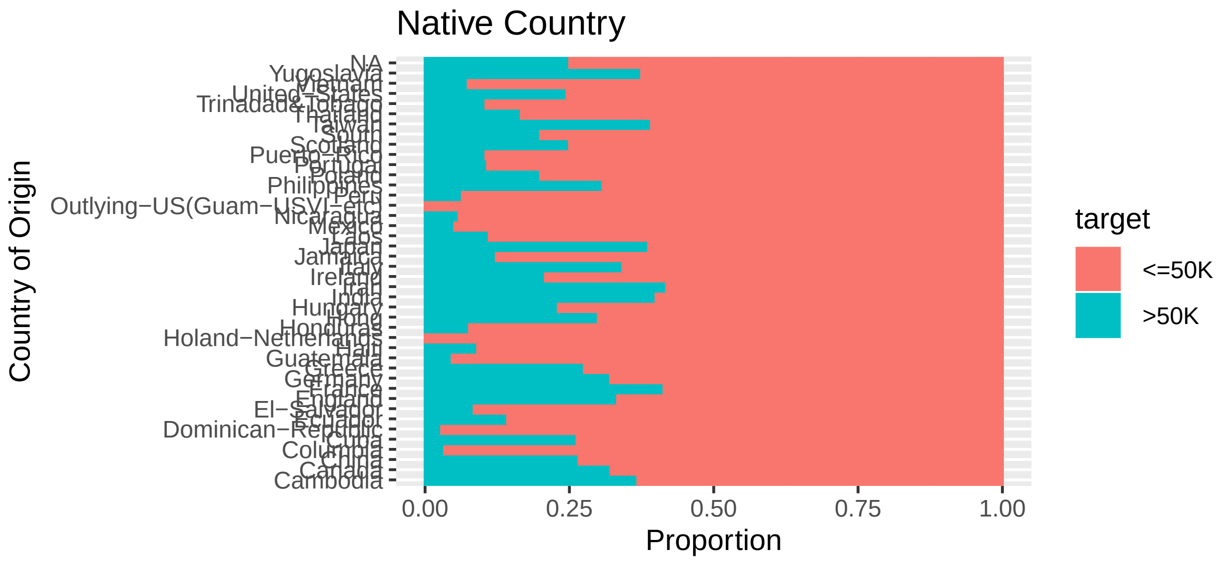
The figure above is a density histogram of the work hours per week. We can see that most people work for 40 to 50 hours per week which is 8 hours per day, assuming 5 weekdays. It seems that most of the people from both income groups work for 40 to 70 hours a week.

3.5 Race and Income



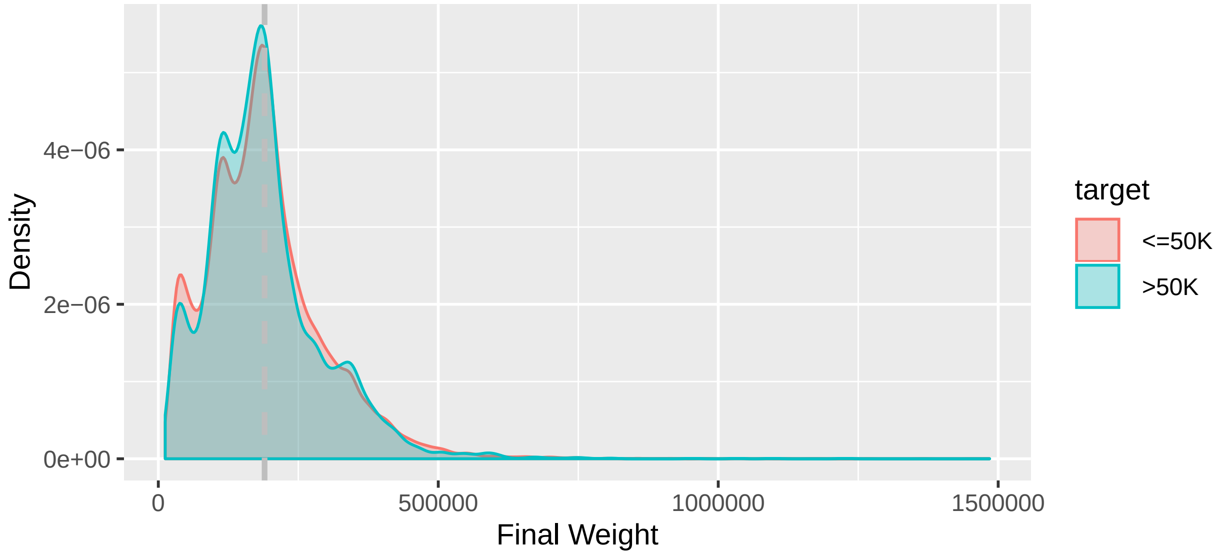
The figure above displays the count for each race and the income proportion of each race. We can see that the dataset is comprised overwhelmingly of white subjects. When we look at the income proportion within each group, we see that Asian Pacific Islanders and White have a relatively high probability of earning over 50K compared to American Indian Eskimo, Black, and Other ethnicities. For simplicity, we will combine Asian-Pac-Islander into one group named “asian-white”, and the others into a group called “others”.

3.6 Country and Income



The figure above displays the proportion of people who earn over 50K according to the country. Although it seems like people from European countries have a higher probability of earning more than 50K, it is difficult to say if there is a distinct pattern in the region. The native country is very similar to race (ethnicity), and because the number of different categories in the native country will make the model difficult to interpret we will exclude it from our model.

3.7 Final Weight and Income

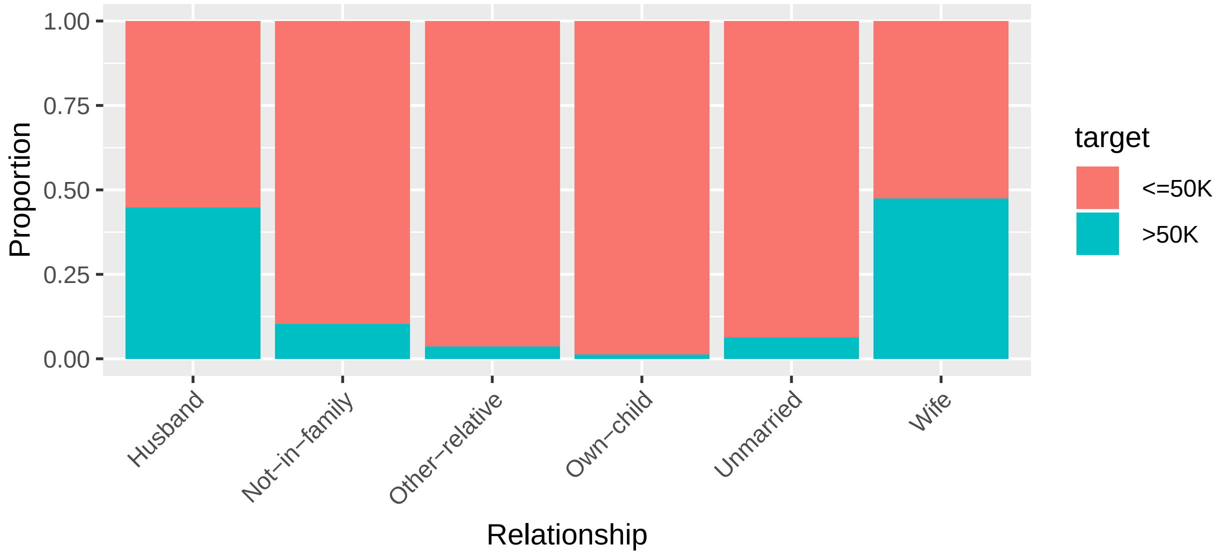


Seems like fnlwgt only has a small effect on the outcome. Both curves have positive skewness and high kurtosis. The data mostly concentrates around 190,000, which is very close to the mean value.

What to do with the Weights(fnlwgt)?

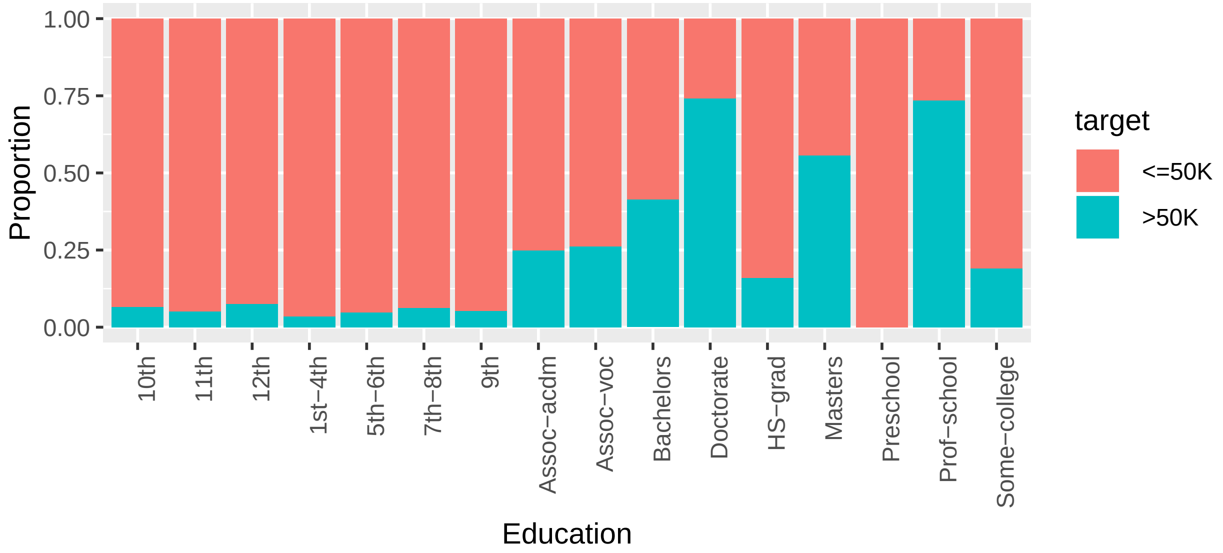
The weight in surveys estimates the number of units in the target population that the observation represents. The weights vary due to the differential sampling rates that vary depending on the subpopulation and by using them we can account for these different sampling rates giving us estimates that better reflect the target population(US Census Bureau). The UCI adult income dataset contains a variety of different subpopulations based on education, marital status, workclass, etc. and the sampling rates for each of these populations are different. This is defined as a sample selection bias in Zadrozney(2004)[3]. While we can potentially utilize the weights to account for sample selection bias, there are statistical learning models that are not affected by sample selection bias such as logistic regression(Zadrozney, 2004). Therefore we can simply ignore the weights.

3.8 Relationship and Income

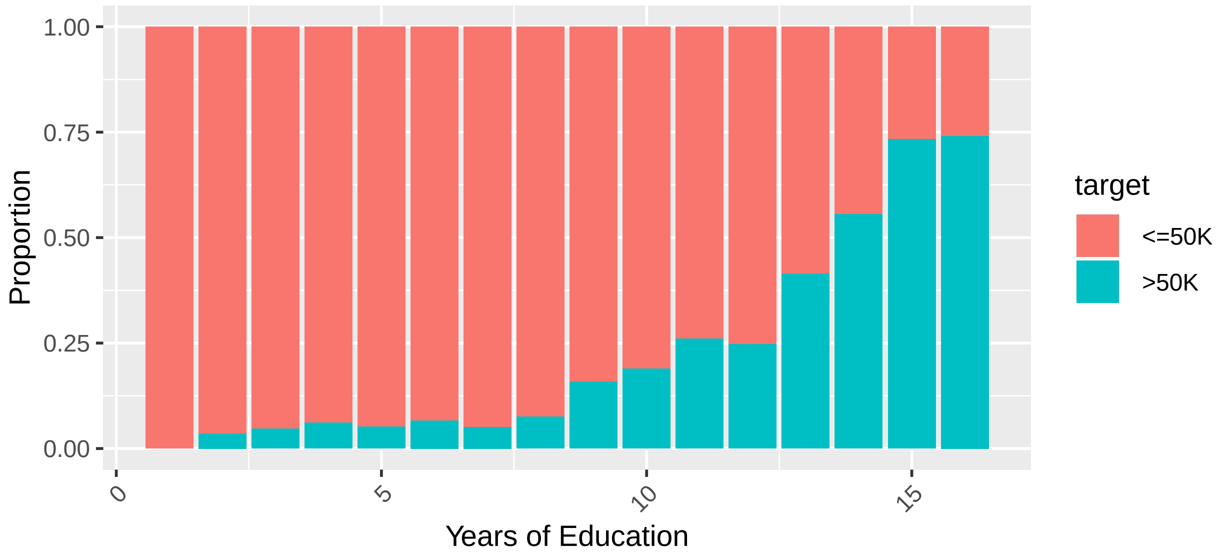


From the plot above we can see that people with a husband or wife i.e. married is likely to earn more than those who are unmarried. We can also see this in the analysis of marital status and income performed in section 1.1. Since the relationship and marital status are similar, we will only include marital status within our model.

3.9 Education and Income

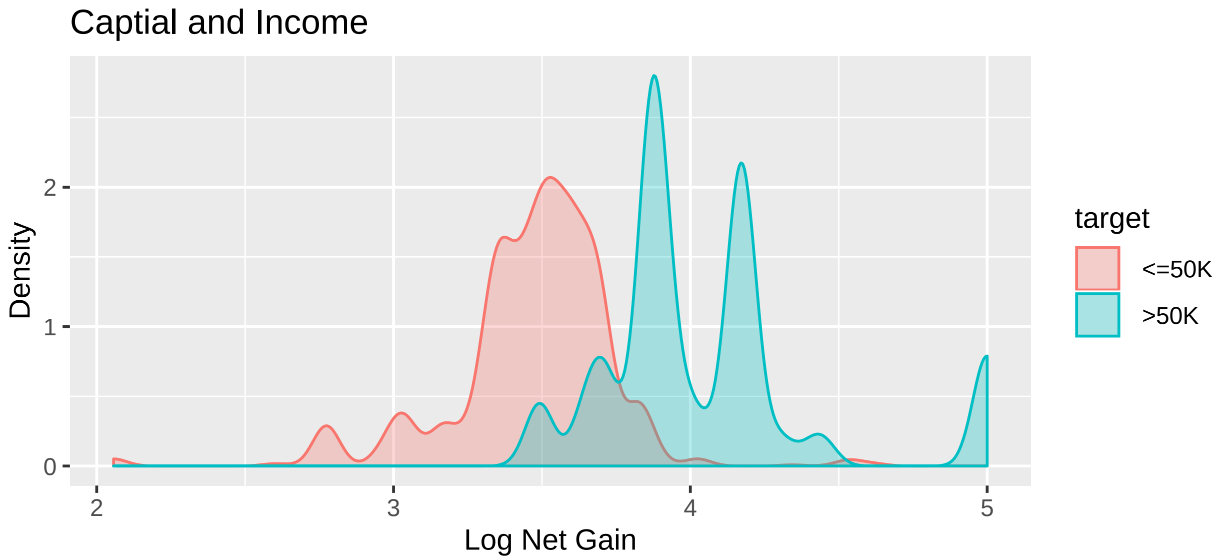


The figure above shows the income proportions for each education group. As we can expect, people with more education (bachelors, masters, doctorate, professional school) tend to earn more than people with only a high school diploma or less.



We can see that the education number and education class has a lot of similarities. People whose class number greater than 13 has a higher chance of gaining higher income. Because of the strong relationship between education and education number, we decide to only keep the education numbers in our model.

3.10 Capital and Income



The graph above displays the log-transformed(base = 10) densities of net capital gain, with NA and nonfinite values excluded. We can see a clear split in the two distributions, indicating that those who earn over 50K per year have more net gain from the capital. Because the untransformed net gain is an extremely skewed distribution, we will conduct a log10 transformation so that it is easier for the model to process.

4. Data Cleaning and Engineering

- delete the dots of labels

- combine train and test set
- delete the blank space of labels
- set the factors
- replace missing values with medians.

5. Model Selection

First, we perform forward selection to pick up a logistic regression model with 10 predictors.

We find that the the model with relationship, education, capital gain, occupation, hours per week, capital loss, age, sex, marital status and work class has the smallest AIC 2092.7 among all the models with 10 variables.

Then we consider the results in exploratory data analysis and decide to fit a logistic regression model with age, marital.status, sex, hours.per.week, race, education.num, and net.capital.gain.

6. Using Bag of Little Bootstrap to Find the Confidence Intervals of the Coefficents

6.1 Why BLB?

Although bootstrap provides a simple way of estimating the confidence intervals of estimators, the computational cost increase notable when data size increases. Bag of Little Bootstraps (BLB) on the other hand is a new procedure that incorporates features of both the bootstrap and subsampling to yield a robust and computationally efficient way of assessing the quality of the estimators. BLB is suitable for parallel and distributed computing while keeping the applicability and statistical efficiency of the bootstrap(Kleiner et al., 2012).

The general algorithm of Bag of Little Bootstrap is described below:

(a) Sample without replacement the sample s times into sizes of b
(b) For each subsample 1.resample each until the sample size is n, r times 2.compute the bootstrap logistic coefficients for each bootstrap sample 3.compute the confidence interval from the bootstrap statistics
(c) take the average of the confidence intervals from all subsamples

In practice, to be more efficient, we generate weights following a multinomial distribution to replace resampling. We also use parallel computing to speed up the computation. We use parLapplyLb function from the parallel package. It applies load balancing which means once a worker finishes its work, it can help other workers with their work. This will help increase the computational efficiency.

6.2 Compute and Display the Confidence Interval

```
## [1] "(Intercept)"
##       2.5%      97.5%
## -0.7697262 -0.6312046
## [1] "age"
##      2.5%     97.5%
## 0.3617901 0.4130615
## [1] "marital.statusseparated"
##      2.5%     97.5%
## -2.384980 -2.268645
## [1] "sexMale"
##       2.5%      97.5%
## 0.03484525 0.16383630
## [1] "hours.per.week"
##      2.5%     97.5%
## 0.3827483 0.4288319
## [1] "raceothers"
##       2.5%      97.5%
## -0.3446108 -0.2037257
## [1] "net.capital.gain"
##      2.5%     97.5%
## 0.9396326 0.9900741
```

The confidence intervals for each of the predictors are shown above. They give us insight into the uncertainty of the model, as well as which predictors are important in classifying income level. We can see from the confidence intervals that all of our predictor coefficients are significant at $\alpha = 0.05$, indicating that our model selection process was correct.

7 Predicting with BLB

7.1 The Prediction Algorithm

The general of predicting with BLB is as follows:

(a) Sample the data without replacement S times into samples of size B
(b) For each sample S

    1. Resample with replacement until each sample is size N. Repeat this procedure to R times to obtain R subsamples from each sample:
    2. Fit a regression model on each of the subsamples and use it to obtain a prediction value
    3. Reduce the prediction value of the R subsample into one by taking the average.

(c) Reduce the prediction computed from all the samples into one by taking the average.

7.2 Accuracy of the Model

```
## [1] "Confusion Matrix"
```

```
##
##         FALSE  TRUE
##   <=50K 11902   533
##   >50K   2280  1566
```

The table above is the confusion matrix of our model.

```
## [1] "Accuracy"
```

```
## [1] 0.8272219
```

Our model has an accuracy of approximately 83% on the test dataset.

7.3 Prediction Interval

In this section, we will construct a 95% prediction interval for a new data point using BLB

```
##             age workclass     fnlwgt education education.num marital.status
## 32661 -1.355131   Private -0.1866972   HS-grad    -0.4178727      separated
##         occupation  relationship        race    sex capital.gain capital.loss
## 32661 Adm-clerical Not-in-family asian-white Female   -0.1426575   -0.2180554
##       hours.per.week native.country target net.capital.gain
## 32661    -0.03143088  United-States  <=50K       -0.2951884
```

Here we arbitrarily chose an observation from the test data set and displayed it in the table above.

```
##       2.5%      97.5%
## 0.01512678 0.01723918
```

As we can see, the prediction interval is (0.015, 0.017) which means that we are approximately 95% confident that the probability of this person having an income over 50K is 1.5% to 1.7%. Since the income of this person is indeed less than 50K, we can see that our model is reasonably accurate in this case.

Reference

1. http://archive.ics.uci.edu/ml/datasets/Adult
2. US Census Bureau. “Weighting.” The United States Census Bureau, 7 May 2018, www.census.gov/programs-surveys/sipp/methodology/weighting.html#par_textimage_4.
3. Zadrozny, Bianca. (2004). Learning and Evaluating Classifiers under Sample Selection Bias. Proceedings, Twenty-First International Conference on Machine Learning, ICML 2004. 2004. 10.1145/1015330.1015425.
4. Kleiner, A., Talwalkar, A., Sarkar, P., & I., M. (2012, June 28). A Scalable Bootstrap for Massive Data. Retrieved from https://arxiv.org/abs/1112.5016

Code Appendix

```r
library(tidyverse)
library(data.table)
library(mlr)
library(parallel)
library(stringr)
library(psych)
library(cowplot)
# Import data
setcol <- c("age",
            "workclass",
            "fnlwgt",
            "education",
            "education-num",
            "marital-status",
            "occupation",
            "relationship",
            "race","sex",
            "capital-gain",
            "capital-loss",
            "hours-per-week",
            "native-country",
            "target")

train <- read.table("adult.data",
                    header = F,
                    sep = ",",
                    col.names = setcol,
                    na.strings = c(" ?"),
                    stringsAsFactors = F)

test <- read.table("adult.test",
                   header = F,
                   sep = ",",
                   col.names = setcol,
                   skip = 1,
                   na.strings = c(" ?"),
                   stringsAsFactors = F)

# summary statistics
summary(train)

# Age and Income
ggplot(train, aes(x=age, color=target, fill=target)) +
  geom_density(alpha=0.7)
# Marital Status and Income
his1 <- ggplot(data=train, aes(x=marital.status))+
  geom_bar() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(x = "Marital status of the individual", y = "Count")

his2 <- ggplot(data=train, aes(x = marital.status, fill = target)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  geom_bar(position="fill") +
  labs(x = "Marital Status", y = "Proportion")

plot_grid(his1, his2, labels = "AUTO")

# Working Class and Income
ggplot(data=train, aes(x = workclass, fill = target)) +
  geom_bar(position="fill") +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(x = "Working Class", y = "Proportion")
# Gender and Income
gender.t <- select(train, target, sex)
ggplot(data = gender.t, aes(x=factor(1), stat="sex", fill = target)) +
  geom_bar(position = "fill") +
  coord_polar(theta="y") +
  facet_grid(. ~ sex)
```

```r
chisq.test(train$target, train$sex)

# Work Hours and Income
ggplot(train) +
  geom_histogram(aes(x = hours.per.week, color = target, fill = target), alpha=.3, bins=10) +
  geom_vline(aes(xintercept = mean(hours.per.week), colour=target), linetype="dashed", color="grey", si
  scale_x_continuous("Hours per Week", seq(0, 100, 10)) +
  labs(title = "Weekly Work Hours", y = "Density")

# Race and Income
his1 <- ggplot(data=train, aes(x=race))+
  geom_bar() +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  labs(x = "Race", y = "Count")

his2 <- ggplot(data=train, aes(x = race, fill = target)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  geom_bar(position="fill") +
  labs(x = "Race", y = "Proportion")

plot_grid(his1, his2, labels = "AUTO")

# Country and Income
ggplot(data = train, aes( x = native.country, color = target, fill = target)) +
  geom_bar(position = "fill") +
  coord_flip() +
  labs(title = "Native Country", x = "Country of Origin", y = "Proportion")

# Final Weight and Income
ggplot(train, aes(x=fnlwgt, colour=target, fill=target)) +
  geom_density(alpha=.3) +
  geom_vline(aes(xintercept=mean(fnlwgt),  colour=target), linetype="dashed",color="grey", size=1) +
  labs(x = "Final Weight", y = "Density")

ggplot(data=train, aes(x=relationship, fill = target)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  geom_bar(position="fill") +
  labs(x = "Relationship", y = "Proportion")

# Education and Income 1
ggplot(data=train, aes(x = education, fill = target)) +
  theme(axis.text.x = element_text(angle = 90, hjust = 1)) +
  geom_bar(position="fill") +
  labs(x = "Education", y = "Proportion")

# Education and Income 2
ggplot(data=train, aes(x=education.num, fill = target)) +
  theme(axis.text.x = element_text(angle = 45, hjust = 1)) +
  geom_bar(position="fill") +
  labs(x = "Years of Education", y = "Proportion")

# Capital and Income
capital <- select(train, capital.gain, capital.loss, target)
```

```r
capital$net.gain <- capital$capital.gain - capital$capital.loss
capital$log.net.gain <- log10(capital$net.gain)
ggplot(capital, aes(x = log.net.gain, colour = target, fill = target)) +
  geom_density(alpha=.3) +
  labs(title = "Captial and Income", x = "Log Net Gain", y = "Density")

# Delete dots within the test labels
test$target <- as.character(test$target)
test$target <- substr(test$target, start = 2, stop = nchar(test$target)-1)

# Combine train and test set
total <- rbind(train, test)

# Delete the blank space within the train labels
char_col <- colnames(total)[sapply(total, is.character)]
for (i in char_col) {
  set(total, j = i,value = str_trim(total[[i]], side = "left"))
}

# Data engineering (combining multiple categories)
total$marital.status <- total$marital.status %>%
  map_chr(~ {
    if(. == "Married-AF-spouse") { "married" }
    else if (. == "Married-civ-spouse") { "married" }
    else { "separated" }
  })

total$race <- total$race%>%
  map_chr(~ {
    if(. == "White") { "asian-white" }
    else if (. == "Asian-Pac-Islander") { "asian-white" }
    else { "others" }
  })

total$net.capital.gain <- total$capital.gain - total$capital.loss
total$net.capital.gain <- total$net.capital.gain %>% map_dbl(~{
  if (. <= 0) {0}
  else {log10(.)}
})

# Set categorical variables as factors
for(i in char_col) {
  set(total, j=i, value = factor(total[[i]]))
}

# Splitting training and testing data
train <- total[1:32561,]
test <- total[32562:48842,]

# Standardizing numerical variables
num_col <- colnames(total)[sapply(total, is.numeric)]
for (col in num_col){
  train[col] <- scale(train[col])
  test[col] <- scale(test[col])
}

# Replace missing values with median
imp1 <- impute(obj = train, target = "target",classes = list(numeric = imputeMedian(),
                                                            factor = imputeMode()))
imp2 <- impute(obj = test, target = "target",classes = list(numeric = imputeMedian(),
                                                           factor = imputeMode()))
train <- imp1$data
test <- imp2$data
# split the data set into S samples
set.seed(141)
s <- 10
groups <- sample(seq_len(s), nrow(train), replace = TRUE)
dir.create("train/", showWarnings = FALSE)
for (i in seq_len(s)) {
  write_csv(filter(train, groups == i), str_c("train/", i, ".csv"))
}

file_names <- file.path("train", list.files("train")) # list of file names under "train" dir
# define each_boot
each_boot <- function(i, data, j){
  freqs <- rmultinom(1, nrow(train), rep(1, nrow(data)))
  fit <- glm(target ~
               age +
               marital.status +
               sex +
               hours.per.week +
               race +
               education.num +
               net.capital.gain,
           data = data,
           weights = freqs,
           family = "binomial")
  (fit$coefficients)[j]
}

r <- 8 # Sample r times for each subsample
each_subsample <- function(file_name, j){
  subsample <- read.csv(file_name)
  head(subsample)
  seq_len(r) %>%
    map_dbl(each_boot, data = subsample, j = j) %>%
    quantile(c(0.025, 0.975))
}

# define jth_coef: non-parallel
jth_coef <- function(j){
  file_names %>%
    map(each_subsample, j = j) %>%
    reduce(`+`) / s
}
# define jth_coef: parallel
```

```r
jth_coef <- function(j) {
  cl <- makeCluster(8)
  clusterExport(cl, c("file_names",
                      "%>%",
                      "r",
                      "s",
                      "map",
                      "reduce",
                      "jth_coef",
                      "map_dbl",
                      "each_boot",
                      "train"))
  coef_ci <- (parLapplyLB(cl, file_names, each_subsample, j=j) %>%
              reduce(`+`)/s)
  stopCluster(cl)
  coef_ci
}

# 95% Confidence interval for the j-th coefficient
coef_names <- c("(Intercept)",
                "age",
                "marital.statusseparated",
                "sexMale",
                "hours.per.week",
                "raceothers",
                "net.capital.gain")
for(i in 1:length(coef_names)) {
  print(coef_names[i])
  print(jth_coef(i))
}

# define each_boot3
each_boot3 <- function(i, data){
  freqs <- rmultinom(1, nrow(train), rep(1, nrow(data)))
  fit <- glm(target ~
               age +
               marital.status +
               sex +
               hours.per.week +
               race +
               education.num +
               net.capital.gain,
             data = data,
             weights = freqs,
             family = "binomial",
             na.action = na.omit)
  predict(fit, test, type = 'response')
}

# define each_subsample3
r <- 8
each_subsample3 <- function(file_name){
  subsample <- read.csv(file_name)
  seq_len(r) %>%
    map(each_boot3, data = subsample) %>%
    reduce(`+`) / s
}

# define calc_pred
calc_pred <- function(){
  cl <- makeCluster(8)
  clusterExport(cl, c("file_names",
                      "%>%",
                      "r",
                      "s",
                      "map",
                      "reduce",
                      "map_dbl",
                      "each_boot3",
                      "train",
                      "test"))
  prediction <- (parLapplyLB(cl, file_names, each_subsample3) %>%
                   reduce(`+`) / s)
  stopCluster(cl)
  prediction
}

predict <- calc_pred()
# Confusion Matrix
print("Confusion Matrix")
table(test$target, predict > 0.5)

# Accuracy
print("Accuracy")
mean((test$target == ">50K") == (predict > 0.5))

# New Observation
newdata.point <- test[100,]
newdata.point

# define each_boot4
each_boot4 <- function(i,data){
  freqs <- rmultinom(1, nrow(train), rep(1,nrow(data)))
  fit<-glm(target ~
             age +
              marital.status +
              sex +
              hours.per.week +
              race +
              education.num +
              net.capital.gain,
           data = data,
           weights = freqs,
           family = "binomial")
  predict(fit, newdata.point, type = 'response') # predicts using the new data point
}
```

```r
# define each_subsample4
r <- 8
each_subsample4 <- function(file_name){
  subsample <- read.csv(file_name)
  seq_len(r) %>%
    map_dbl(each_boot4, data = subsample) %>%
    quantile(c(0.025,0.975))
}

# define pred_ci
pred_ci <- function(){
  cl <- makeCluster(8)
  clusterExport(cl, c("file_names",
                      "%>%",
                      "r",
                      "s",
                      "map",
                      "reduce",
                      "map_dbl",
                      "each_boot4",
                      "train",
                      "newdata.point"))
  prediction <- (parLapplyLB(cl, file_names, each_subsample4) %>%
                   reduce(`+`)/s)
  stopCluster(cl)
  prediction
}

# Calculate prediction inteval
pred_ci()
```