# Analysis of Adult Income Dataset

Benjamin Khoo

2024-11-06

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

```
## Loading required package: knitr
```

```r
library(knitr)
# Attempt to keep code tidy
opts_chunk$set(tidy.opts = list(width.cutoff=60), tidy=TRUE)
knitr::opts_chunk$set(echo = TRUE)
```

## Introduction

The aim of this project is to design a machine learning algorithm to predict whether an individual earns more or less than $50k/year using the adult income dataset. This is a dataset containing 32561 observations of 15 variables. The first few lines of code have been provided to download the code from a GitHub repository. The original dataset may be found on Kaggle at the following website:

https://www.kaggle.com/datasets/wenruliu/adult-income-dataset

## Methods/Analysis

### Load libraries using the if!require function, to download and install required packages only if required.

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

```r
if (!require(caret)) install.packages("caret", repos = "http://cran.us.r-project.org")
library(caret)
if (!require(RCurl)) install.packages("RCurl", repos = "http://cran.us.r-project.org")
library(RCurl)
if (!require(ggplot2)) install.packages("ggplot2", repos = "http://cran.us.r-project.org")
library(ggplot2)
if (!require(dplyr)) install.packages("dplyr", repos = "http://cran.us.r-project.org")
library(dplyr)
```

```r
if (!require(randomForest)) install.packages("randomForest",
    repos = "http://cran.us.r-project.org")
library(randomForest)
if (!require(e1071)) install.packages("e1071", repos = "http://cran.us.r-project.org")
library(e1071)
if (!require(rpart)) install.packages("rpart", repos = "http://cran.us.r-project.org")
library(rpart)
if (!require(rpart.plot)) install.packages("rpart", repos = "http://cran.us.r-project.org")
library(rpart.plot)
if (!require(ROSE)) install.packages("ROSE", repos = "http://cran.us.r-project.org")
library(ROSE)
```

### Load dataset from Github and gain an overview of the dataset

```r
options(timeout = 120)
x <- getURL("https://raw.githubusercontent.com/bkhooze/CYO/refs/heads/main/adult.csv")
salary <- read.csv(text = x)
head(salary)
```

```
##   age workclass fnlwgt    education education.num marital.status
## 1  90         ?  77053      HS-grad             9        Widowed
## 2  82   Private 132870      HS-grad             9        Widowed
## 3  66         ? 186061 Some-college            10        Widowed
## 4  54   Private 140359      7th-8th             4       Divorced
## 5  41   Private 264663 Some-college            10      Separated
## 6  34   Private 216864      HS-grad             9       Divorced
##          occupation  relationship  race    sex capital.gain capital.loss
## 1                 ? Not-in-family White Female            0         4356
## 2   Exec-managerial Not-in-family White Female            0         4356
## 3                 ?     Unmarried Black Female            0         4356
## 4 Machine-op-inspct     Unmarried White Female            0         3900
## 5    Prof-specialty     Own-child White Female            0         3900
## 6     Other-service     Unmarried White Female            0         3770
##   hours.per.week native.country income
## 1             40  United-States  <=50K
## 2             18  United-States  <=50K
## 3             40  United-States  <=50K
## 4             40  United-States  <=50K
## 5             40  United-States  <=50K
## 6             45  United-States  <=50K
```

```r
glimpse(salary)
```

```
## Rows: 32,561
## Columns: 15
## $ age            <int> 90, 82, 66, 54, 41, 34, 38, 74, 68, 41, 45, 38, 52, 32,~
## $ workclass      <chr> "?", "Private", "?", "Private", "Private", "Private", "~
## $ fnlwgt         <int> 77053, 132870, 186061, 140359, 264663, 216864, 150601, ~
## $ education      <chr> "HS-grad", "HS-grad", "Some-college", "7th-8th", "Some-~
## $ education.num  <int> 9, 9, 10, 4, 10, 9, 6, 16, 9, 10, 16, 15, 13, 14, 16, 1~
```

```
## $ marital.status <chr> "Widowed", "Widowed", "Widowed", "Divorced", "Separated~
## $ occupation     <chr> "?", "Exec-managerial", "?", "Machine-op-inspct", "Prof~
## $ relationship   <chr> "Not-in-family", "Not-in-family", "Unmarried", "Unmarri~
## $ race           <chr> "White", "White", "Black", "White", "White", "White", "~
## $ sex            <chr> "Female", "Female", "Female", "Female", "Female", "Fema~
## $ capital.gain   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ capital.loss   <int> 4356, 4356, 4356, 3900, 3900, 3770, 3770, 3683, 3683, 3~
## $ hours.per.week <int> 40, 18, 40, 40, 40, 45, 40, 20, 40, 60, 35, 45, 20, 55,~
## $ native.country <chr> "United-States", "United-States", "United-States", "Uni~
## $ income         <chr> "<=50K", "<=50K", "<=50K", "<=50K", "<=50K", "<=50K", "~
```

The 15 variables in the dataset are: 1. Age 2. Workclass 3. Fnlwgt 4. Education 5. Education numerical 6. Marital status 7. Occupation 8. Relationship 9. Race 10. Sex 11. Capital gain 12. Capital loss 13. Hours per week 14. Native country 15. Income

### Recode values and drop missing values.

From the preliminary exploration, noted values coded as ?. This is recoded to NA.

```r
# Noted in this dataset missing values coded as ?, recode
# this to NA
salary[salary == "?"] <- NA
salary %>%
    summarise_all(~sum(is.na(.)))
```

```
##   age workclass fnlwgt education education.num marital.status occupation
## 1   0      1836      0         0             0              0       1843
##   relationship race sex capital.gain capital.loss hours.per.week native.country
## 1            0    0   0            0            0              0            583
##   income
## 1      0
```

```r
# Most columns have no missing values except workclass,
# occupation and native.country
sum(is.na(salary$occupation))/length(salary$occupation) * 100
```

```
## [1] 5.660146
```

```r
table(salary$workclass)
```

```
##
##      Federal-gov        Local-gov     Never-worked          Private
##              960             2093                7            22696
##     Self-emp-inc Self-emp-not-inc        State-gov      Without-pay
##             1116             2541             1298               14
```

```r
table(salary$occupation)
```

```
##
##      Adm-clerical      Armed-Forces      Craft-repair   Exec-managerial
```

```
##              3770                 9              4099              4066
##   Farming-fishing Handlers-cleaners Machine-op-inspct     Other-service
##               994              1370              2002              3295
##   Priv-house-serv    Prof-specialty   Protective-serv             Sales
##               149              4140               649              3650
##      Tech-support  Transport-moving
##               928              1597
```

```r
# Following code changes the NA values in the column
# workclass to 'Private', which is the most common
# observation.
salary <- salary %>%
    mutate(workclass = ifelse(is.na(workclass), "Private", workclass))
salary <- na.omit(salary)
salary %>%
    summarise_all(~sum(is.na(.)))
```

```
##   age workclass fnlwgt education education.num marital.status occupation
## 1   0         0      0         0             0              0          0
##   relationship race sex capital.gain capital.loss hours.per.week native.country
## 1            0    0   0            0            0              0              0
##   income
## 1      0
```

```r
glimpse(salary)
```

```
## Rows: 30,162
## Columns: 15
## $ age            <int> 82, 54, 41, 34, 38, 74, 68, 45, 38, 52, 32, 46, 45, 57,~
## $ workclass      <chr> "Private", "Private", "Private", "Private", "Private", ~
## $ fnlwgt         <int> 132870, 140359, 264663, 216864, 150601, 88638, 422013, ~
## $ education      <chr> "HS-grad", "7th-8th", "Some-college", "HS-grad", "10th"~
## $ education.num  <int> 9, 4, 10, 9, 6, 16, 9, 16, 15, 13, 14, 15, 7, 14, 13, 1~
## $ marital.status <chr> "Widowed", "Divorced", "Separated", "Divorced", "Separa~
## $ occupation     <chr> "Exec-managerial", "Machine-op-inspct", "Prof-specialty~
## $ relationship   <chr> "Not-in-family", "Unmarried", "Own-child", "Unmarried",~
## $ race           <chr> "White", "White", "White", "White", "White", "White", "~
## $ sex            <chr> "Female", "Female", "Female", "Female", "Male", "Female~
## $ capital.gain   <int> 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0~
## $ capital.loss   <int> 4356, 3900, 3900, 3770, 3770, 3683, 3683, 3004, 2824, 2~
## $ hours.per.week <int> 18, 40, 40, 45, 40, 20, 40, 35, 45, 20, 55, 40, 76, 50,~
## $ native.country <chr> "United-States", "United-States", "United-States", "Uni~
## $ income         <chr> "<=50K", "<=50K", "<=50K", "<=50K", "<=50K", ">50K", "<~
```

Most columns have no missing values except workclass, occupation and native country, of which workclass and occupation have the most missing values. As the category with most observations for workclass is "Private", missing values for workclass were recoded to "Private". For occupation, as missing data was 5.7%, decision to proceed with complete case analysis for this project. The rows with NA values for occupation and native country were dropped. After processing, there are 30162 rows remaining in the dataset (original 32561).

### Changing the columns in the dataset to appropriate variable type - numeric and factor respectively.

```
summary(salary)
```

```
##       age         workclass             fnlwgt          education        
##  Min.   :17.00   Length:30162       Min.   :  13769   Length:30162      
##  1st Qu.:28.00   Class :character   1st Qu.: 117627   Class :character  
##  Median :37.00   Mode  :character   Median : 178425   Mode  :character  
##  Mean   :38.44                      Mean   : 189794                     
##  3rd Qu.:47.00                      3rd Qu.: 237629                     
##  Max.   :90.00                      Max.   :1484705                     
##  education.num   marital.status      occupation        relationship      
##  Min.   : 1.00   Length:30162       Length:30162       Length:30162      
##  1st Qu.: 9.00   Class :character   Class :character   Class :character  
##  Median :10.00   Mode  :character   Mode  :character   Mode  :character  
##  Mean   :10.12                                                           
##  3rd Qu.:13.00                                                           
##  Max.   :16.00                                                           
##      race               sex             capital.gain    capital.loss    
##  Length:30162       Length:30162       Min.   :    0   Min.   :   0.00  
##  Class :character   Class :character   1st Qu.:    0   1st Qu.:   0.00  
##  Mode  :character   Mode  :character   Median :    0   Median :   0.00  
##                                        Mean   : 1092   Mean   :  88.37  
##                                        3rd Qu.:    0   3rd Qu.:   0.00  
##                                        Max.   :99999   Max.   :4356.00  
##  hours.per.week  native.country        income         
##  Min.   : 1.00   Length:30162       Length:30162      
##  1st Qu.:40.00   Class :character   Class :character  
##  Median :40.00   Mode  :character   Mode  :character  
##  Mean   :40.93                                        
##  3rd Qu.:45.00                                        
##  Max.   :99.00                                        
```

```
salary[] <- lapply(salary, trimws)
num <- c(1, 3, 5, 11, 12, 13)
salary[num] <- sapply(salary[num], as.numeric)
# Transform appropriate columns to numeric type
cat <- c(2, 4, 6, 7, 8, 9, 10, 14)
salary[, cat] <- lapply(salary[, cat], factor)
# Transform appropriate columns to factor type
str(salary)
```

```
## 'data.frame':    30162 obs. of  15 variables:
##  $ age           : num  82 54 41 34 38 74 68 45 38 52 ...
##  $ workclass     : Factor w/ 7 levels "Federal-gov",..: 3 3 3 3 3 6 1 3 5 3 ...
##  $ fnlwgt        : num  132870 140359 264663 216864 150601 ...
##  $ education     : Factor w/ 16 levels "10th","11th",..: 12 6 16 12 1 11 12 11 15 10 ...
##  $ education.num : num  9 4 10 9 6 16 9 16 15 13 ...
##  $ marital.status: Factor w/ 7 levels "Divorced","Married-AF-spouse",..: 7 1 6 1 6 5 1 1 5 7 ...
##  $ occupation    : Factor w/ 14 levels "Adm-clerical",..: 4 7 10 8 1 10 10 10 10 8 ...
##  $ relationship  : Factor w/ 6 levels "Husband","Not-in-family",..: 2 5 4 5 5 3 2 5 2 2 ...
```

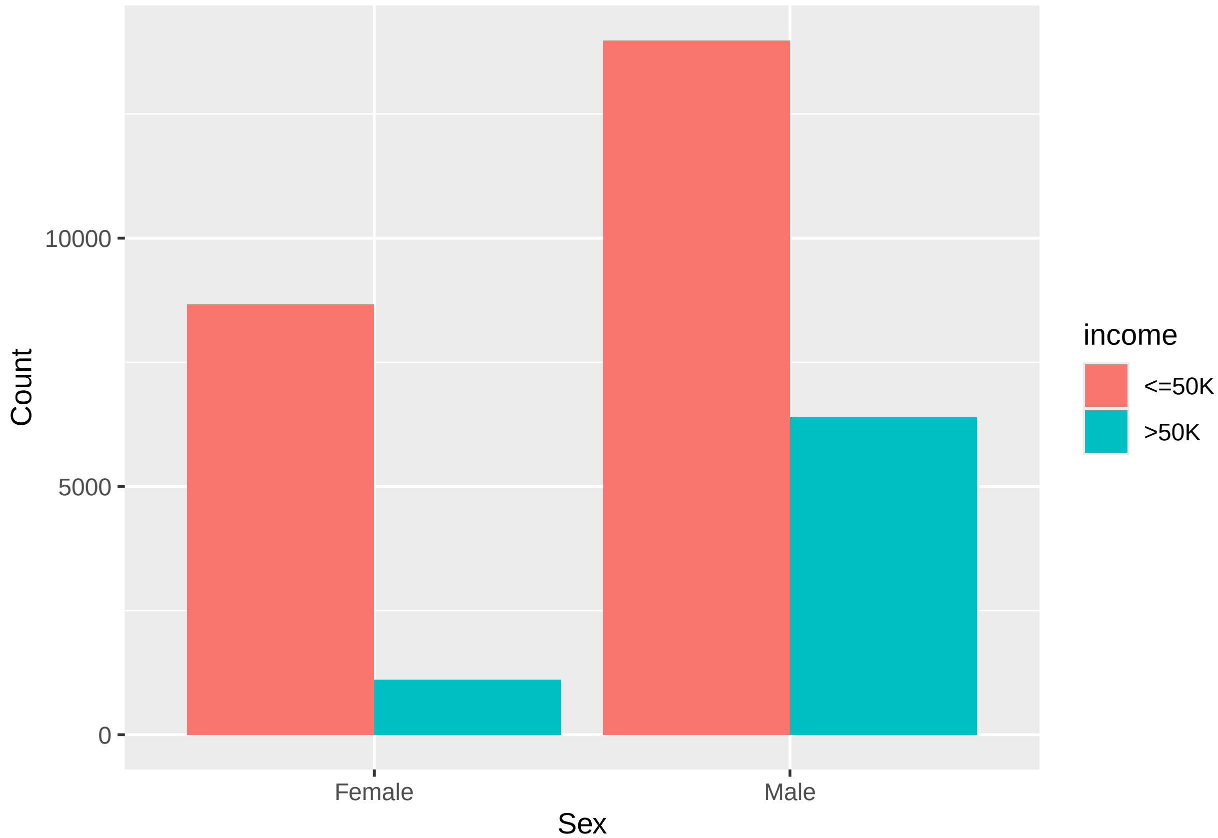
```
##  $ race          : Factor w/ 5 levels "Amer-Indian-Eskimo",..: 5 5 5 5 5 5 5 3 5 5 ...
##  $ sex           : Factor w/ 2 levels "Female","Male": 1 1 1 1 2 1 1 1 2 1 ...
##  $ capital.gain  : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ capital.loss  : num  4356 3900 3900 3770 3770 ...
##  $ hours.per.week: num  18 40 40 45 40 20 40 35 45 20 ...
##  $ native.country: Factor w/ 41 levels "Cambodia","Canada",..: 39 39 39 39 39 39 39 39 39 39 ...
##  $ income        : chr  "<=50K" "<=50K" "<=50K" "<=50K" ...
##  - attr(*, "na.action")= 'omit' Named int [1:2399] 1 3 10 15 19 25 45 49 50 66 ...
##   ..- attr(*, "names")= chr [1:2399] "1" "3" "10" "15" ...
```

### Pre-processing data - removing columns with multiple repeated values.

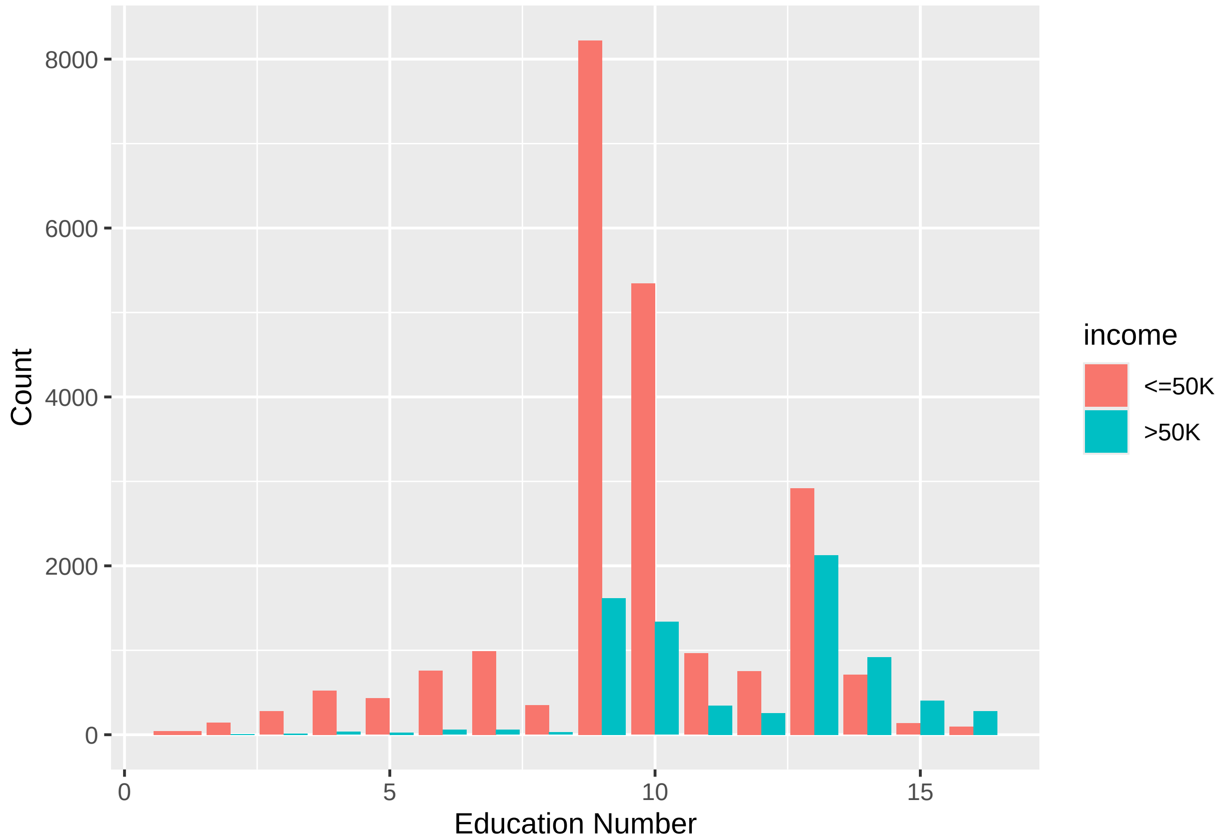
```r
table(salary$capital.gain)
table(salary$capital.loss)
table(salary$fnlwgt)
# Removed columns fnlwgt, capital.gain and capital.loss in
# view of multiple repeated values, with more than 20000
# values are '0'. Also uncertain of how these affects the
# outcome variable, income.
salary <- salary[-c(3, 11, 12)]
str(salary)
```

### Data visualisation

Various features in the dataset which may affect the outcome are presented here graphically.



10000
5000
0
Count
Female
Male
Sex
income
<=50K
>50K



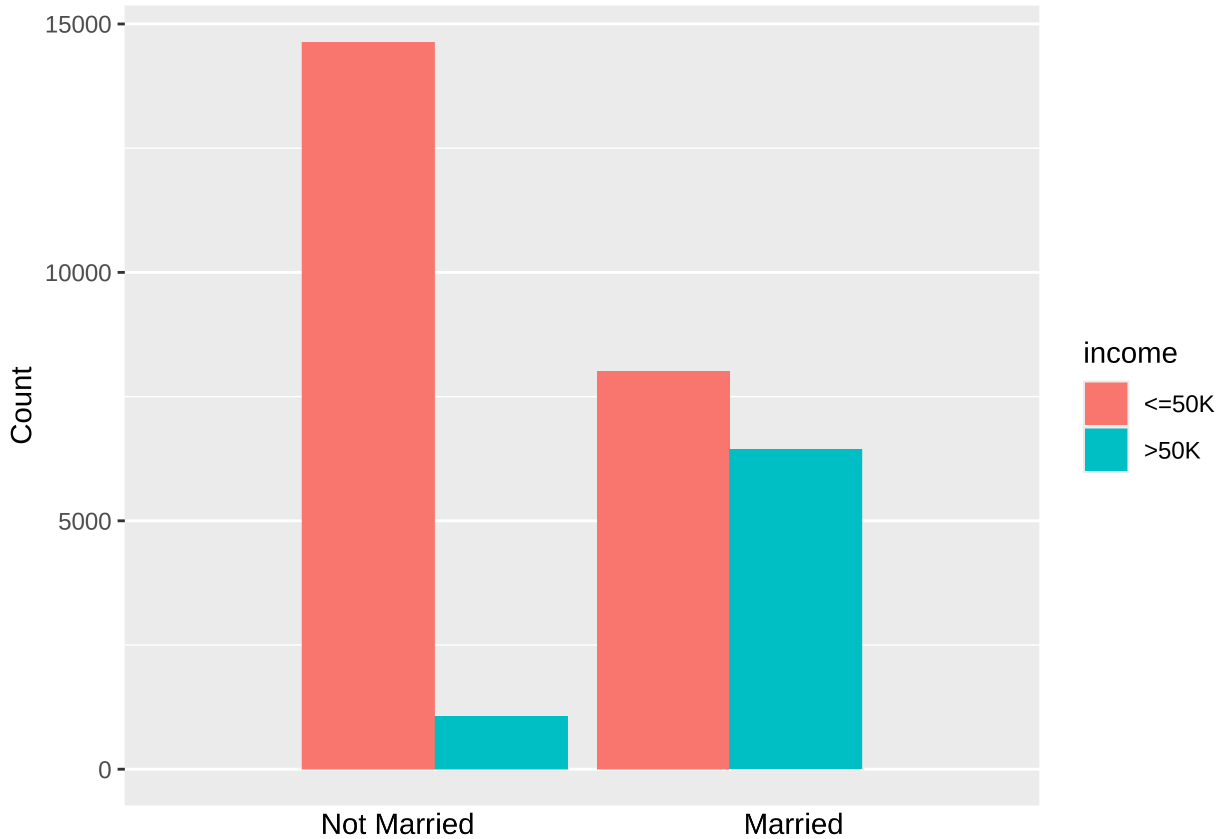

A greater proportion of males earned >$50k, compared to females. With increasing level of education, the proportion of people who have income > $50k increases. These feature may be used in subsequent model development.

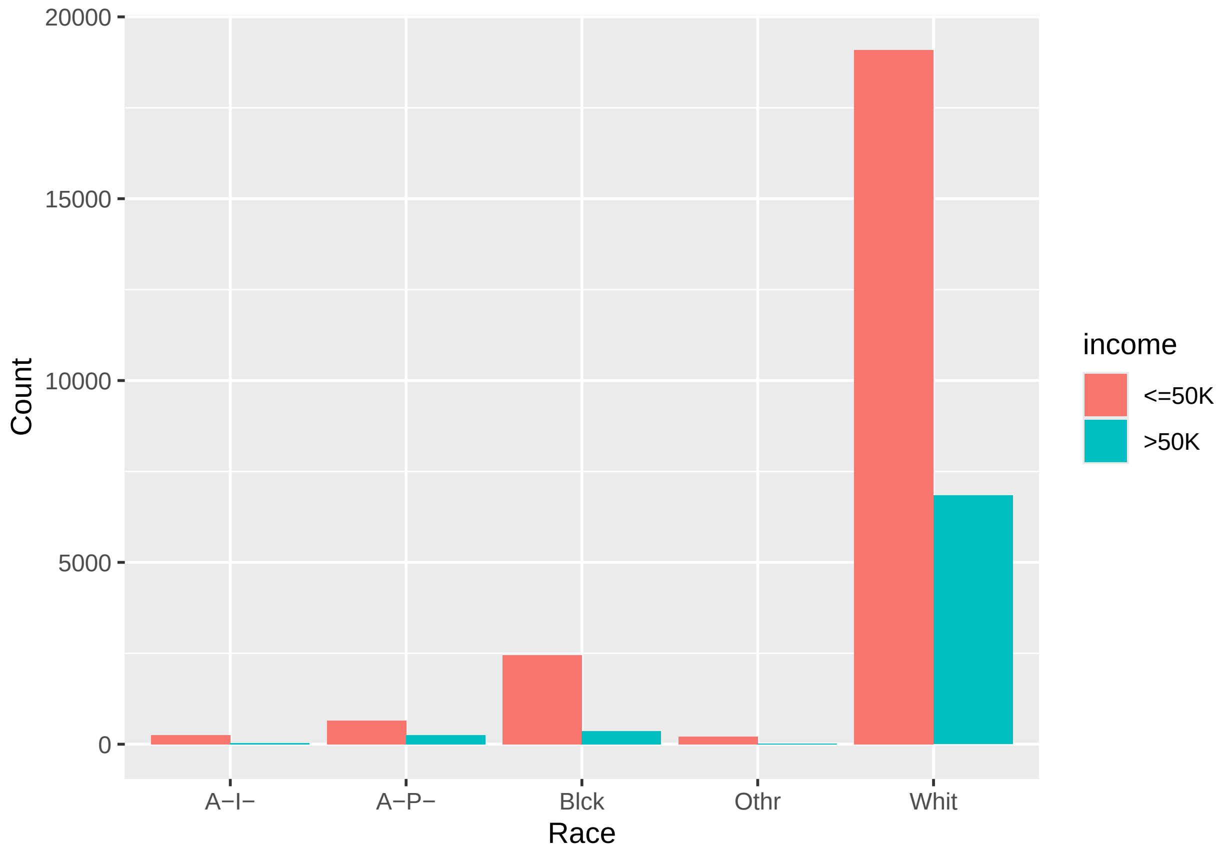
### Changing marital status to binary outcomes - 1 for married and 0 for not married

```
# Marital status has multiple categories - aim to recode as
# binary
table(salary$marital.status)
```

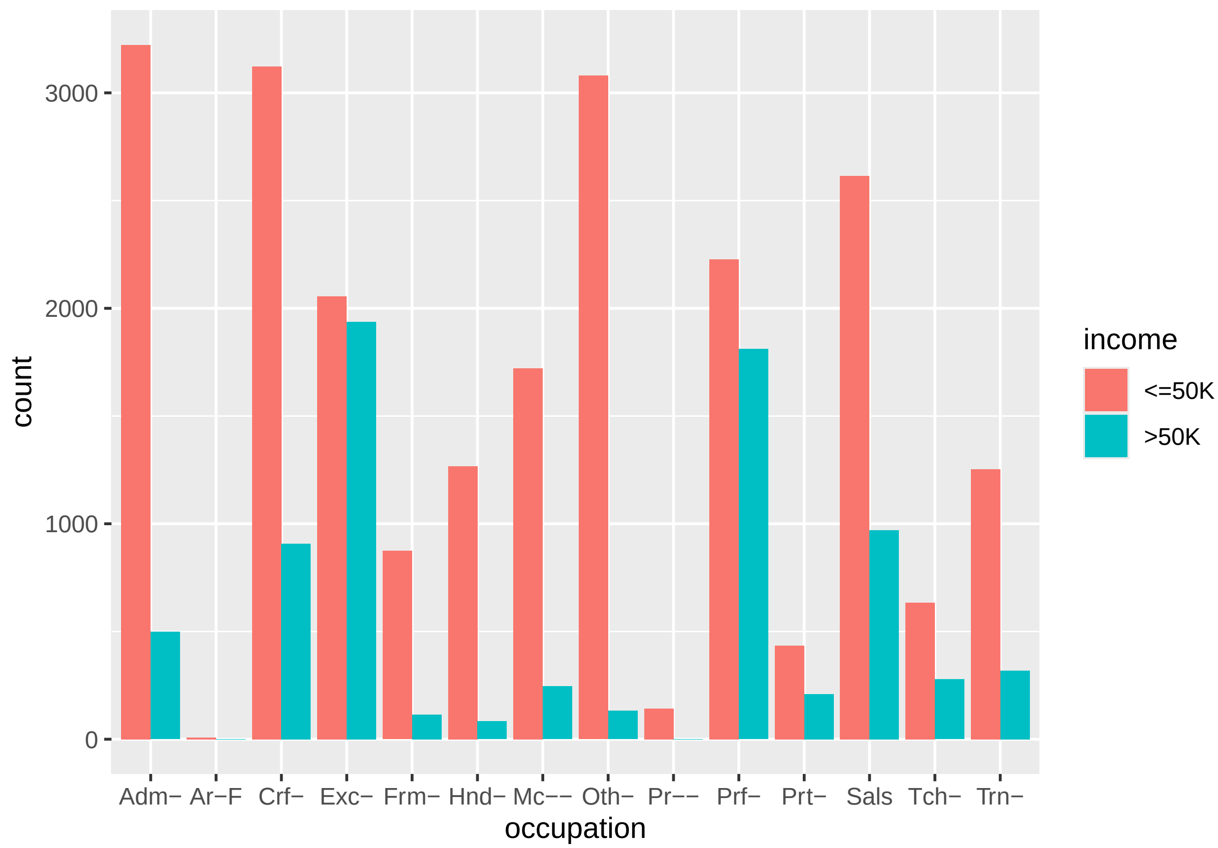
```
##
##              Divorced     Married-AF-spouse    Married-civ-spouse
##                  4214                    21                 14065
## Married-spouse-absent         Never-married             Separated
##                   370                  9726                   939
##               Widowed
##                   827
```

```
# Noted marital status consists of multiple values, to
# convert this to people who are married vs not
salary <- salary %>%
    mutate(marriage_binary = ifelse(marital.status %in% c("Married-civ-spouse",
        "Married-AF-spouse", "Married-spouse-absent"), 1, 0))
```


15000
10000
5000
0
Count
Not Married
Married
income
<=50K
>50K




```
##
##      Adm-clerical      Armed-Forces      Craft-repair   Exec-managerial
##              3721                 9              4030              3992
##   Farming-fishing Handlers-cleaners Machine-op-inspct     Other-service
##               989              1350              1966              3212
##   Priv-house-serv    Prof-specialty   Protective-serv             Sales
##               143              4038               644              3584
##      Tech-support  Transport-moving
##               912              1572
```




Refer to the table of occupations for the legend. Executive, professional and sales jobs seem to have the highest proportion of income earners > $50k.


Individuals in the 30 to 50 age range have the highest proportion of people earning >$50k a year.

```
# Recode income into binary outcomes, 0 if income <$50K and
# 1 if income >$50K. Drop marital status column as this has
# been coded into binary. Drop education column as this is
# coded in education.num.
salary <- salary %>%
    mutate(income = ifelse(income == c("<=50K"), 0, 1))
salary <- salary[-c(3, 5)]
str(salary)
```

```
## 'data.frame':    30162 obs. of  11 variables:
##  $ age             : num  82 54 41 34 38 74 68 45 38 52 ...
##  $ workclass       : Factor w/ 7 levels "Federal-gov",..: 3 3 3 3 3 6 1 3 5 3 ...
##  $ education.num   : num  9 4 10 9 6 16 9 16 15 13 ...
##  $ occupation      : Factor w/ 14 levels "Adm-clerical",..: 4 7 10 8 1 10 10 10 10 8 ...
##  $ relationship    : Factor w/ 6 levels "Husband","Not-in-family",..: 2 5 4 5 5 3 2 5 2 2 ...
##  $ race            : Factor w/ 5 levels "Amer-Indian-Eskimo",..: 5 5 5 5 5 5 5 3 5 5 ...
##  $ sex             : Factor w/ 2 levels "Female","Male": 1 1 1 1 2 1 1 1 2 1 ...
##  $ hours.per.week : num  18 40 40 45 40 20 40 35 45 20 ...
##  $ native.country : Factor w/ 41 levels "Cambodia","Canada",..: 39 39 39 39 39 39 39 39 39 39 ...
##  $ income          : num  0 0 0 0 0 1 0 1 1 1 ...
##  $ marriage_binary: num  0 0 0 0 0 0 0 0 0 0 ...
```

```
# Check if there is correlation between columns which are
# numeric correlate with each other
```

```
correlation_var <- c("age", "education.num", "hours.per.week",
    "marriage_binary", "income")
correlation <- round(cor(salary[correlation_var]), 2)
correlation
```

```
##                  age education.num hours.per.week marriage_binary income
## age             1.00          0.04           0.10            0.31   0.24
## education.num   0.04          1.00           0.15            0.07   0.34
## hours.per.week  0.10          0.15           1.00            0.22   0.23
## marriage_binary 0.31          0.07           0.22            1.00   0.44
## income          0.24          0.34           0.23            0.44   1.00
```

In the numeric variables in the dataset for analysis, there are no variables that are highly correlated with each other and thus all variables were included for the analysis.

```
# Partition into test and train set. Decision to use 0.3
# for test set as outcome is somewhat unbalanced.
set.seed(1, sample.kind = "Rounding")
test_index <- createDataPartition(salary$income, times = 1, p = 0.3,
    list = FALSE)
salary_train <- salary[-test_index, ]
salary_test <- salary[test_index, ]
```

### Dealing with unbalanced dataset

From the earlier exploration, it is noted that most individuals in the dataset earn <$50k, and therefore the dataset is imbalanced. An oversampling strategy was selected to deal with this issue, using the ROSE package.

```
oversampled_data <- ovun.sample(income ~ ., data = salary_train,
    method = "over", N = 31656)$data
table(oversampled_data$income)
```

```
##
##     0     1
## 15828 15828
```

### Results

For this project, the base model using logistic regression was compared with the following machine learning methods: Random Forest, Support Vector Machines and Decision Tree. The outcome of interest was income as a binary variable i.e. more or less than $50k. These analyses were run using the balanced dataset created above.

```
# Model 1: Logistic regression using oversampling to
# correct for imbalanced dataset
fit_glm <- glm(income ~ ., data = oversampled_data, family = binomial("logit"))
p_glm <- predict(fit_glm, salary_test, type = "response")
p_glm <- as.factor(ifelse(p_glm > 0.5, "1", "0"))
salary_test$income <- as.factor(salary_test$income)
m1 <- confusionMatrix(p_glm, salary_test$income)
m1
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 5215  369
##          1 1611 1854
##
##                Accuracy : 0.7812
##                  95% CI : (0.7725, 0.7897)
##     No Information Rate : 0.7543
##     P-Value [Acc > NIR] : 1.007e-09
##
##                   Kappa : 0.5032
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.7640
##             Specificity : 0.8340
##          Pos Pred Value : 0.9339
##          Neg Pred Value : 0.5351
##              Prevalence : 0.7543
##          Detection Rate : 0.5763
##    Detection Prevalence : 0.6171
##       Balanced Accuracy : 0.7990
##
##        'Positive' Class : 0
##
```

The accuracy of logistic regression to predict the outcome using balanced data was 0.781.

```r
# Model 2: Random Forest
fit_rf <- randomForest(income ~ ., data = oversampled_data, ntree = 500)
pred_rf <- predict(fit_rf, salary_test, type = "response")
pred_rf <- as.factor(ifelse(pred_rf > 0.5, "1", "0"))
m2 <- confusionMatrix(pred_rf, salary_test$income)
m2
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 5650  544
##          1 1176 1679
##
##                Accuracy : 0.8099
##                  95% CI : (0.8017, 0.818)
##     No Information Rate : 0.7543
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.532
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.8277
```

```
##          Specificity : 0.7553
##       Pos Pred Value : 0.9122
##       Neg Pred Value : 0.5881
##           Prevalence : 0.7543
##       Detection Rate : 0.6244
## Detection Prevalence : 0.6845
##    Balanced Accuracy : 0.7915
##
##     'Positive' Class : 0
##
```

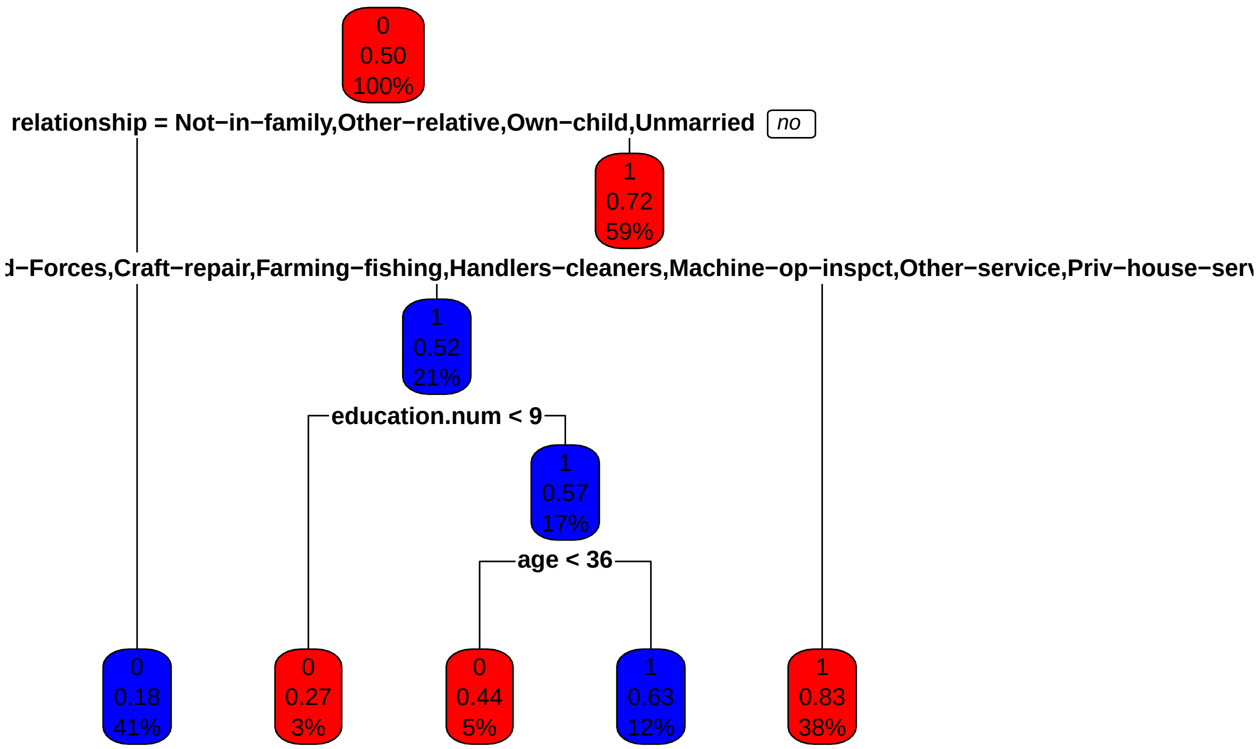
The accuracy of random forest to predict the outcome using balanced data was 0.810.

```
# Model 3: Support Vector Machines
fit_svm <- svm(income ~ ., data = oversampled_data)
pred_svm <- predict(fit_svm, newdata = salary_test, type = "response")
pred_svm <- as.factor(ifelse(pred_svm > 0.5, "1", "0"))
m3 <- confusionMatrix(pred_svm, salary_test$income)
m3
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 4815  370
##          1 2011 1853
##
##                Accuracy : 0.7369
##                  95% CI : (0.7277, 0.7459)
##     No Information Rate : 0.7543
##     P-Value [Acc > NIR] : 0.9999
##
##                   Kappa : 0.4315
##
##  Mcnemar's Test P-Value : <2e-16
##
##             Sensitivity : 0.7054
##             Specificity : 0.8336
##          Pos Pred Value : 0.9286
##          Neg Pred Value : 0.4796
##              Prevalence : 0.7543
##          Detection Rate : 0.5321
##    Detection Prevalence : 0.5730
##       Balanced Accuracy : 0.7695
##
##        'Positive' Class : 0
##
```

The accuracy of support vector machines to predict the outcome using balanced data was 0.737.

```
# Model 4: Decision tree
fit_dectree <- rpart(income ~ ., data = oversampled_data, method = "class")
rpart.plot(fit_dectree, box.col = c("red", "blue"))
```




```r
pred_dectree <- predict(fit_dectree, newdata = salary_test, type = "class")
m4 <- confusionMatrix(pred_dectree, salary_test$income, positive = "1")
m4
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 5262  478
##          1 1564 1745
##
##                Accuracy : 0.7743
##                  95% CI : (0.7656, 0.7829)
##     No Information Rate : 0.7543
##     P-Value [Acc > NIR] : 4.34e-06
##
##                   Kappa : 0.4772
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.7850
##             Specificity : 0.7709
##          Pos Pred Value : 0.5273
##          Neg Pred Value : 0.9167
##              Prevalence : 0.2457
##          Detection Rate : 0.1928
```

```
##    Detection Prevalence : 0.3657
##       Balanced Accuracy : 0.7779
##
##        'Positive' Class : 1
##
```

The accuracy of decision tree to predict the outcome using balanced data was 0.774.

# Calculating the F1 score

F1 score was also calculated as this balances precision and recall, and provides a useful metric to assess a dataset where there is some imbalance and also provides a more stable model performance.

```
f1_score <- function(model) {
    precision <- model$byClass["Pos Pred Value"]
    recall <- model$byClass["Sensitivity"]
    f1 <- 2 * (precision * recall)/(precision + recall)
}
f1m1 <- f1_score(m1)
f1m2 <- f1_score(m2)
f1m3 <- f1_score(m3)
f1m4 <- f1_score(m4)
```

```
options(digits = 3)
results <- tibble(Model = c("Logistic Regression Balanced", "Random Forest",
    "Support Vector Machines", "Decision Tree"), Accuracy = c(m1$overall["Accuracy"],
    m2$overall["Accuracy"], m3$overall["Accuracy"], m4$overall["Accuracy"]),
    Sensitvity = c(m1$byClass["Sensitivity"], m2$byClass["Sensitivity"],
        m3$byClass["Sensitivity"], m4$byClass["Sensitivity"]),
    Specificity = c(m1$byClass["Specificity"], m2$byClass["Specificity"],
        m3$byClass["Specificity"], m4$byClass["Specificity"]),
    F1score = c(f1m1, f1m2, f1m3, f1m4))
results
```

```
## # A tibble: 4 x 5
##   Model                        Accuracy Sensitvity Specificity F1score
##   <chr>                           <dbl>      <dbl>       <dbl>   <dbl>
## 1 Logistic Regression Balanced    0.781      0.764       0.834   0.840
## 2 Random Forest                   0.810      0.828       0.755   0.868
## 3 Support Vector Machines         0.737      0.705       0.834   0.802
## 4 Decision Tree                   0.774      0.785       0.771   0.631
```

The accuracy, sensitivity and specificity and F1 score of the various models to predict whether an individual earns $50k or more is displayed in the table above. These results were derived using the oversampling method to deal with an unbalanced dataset. Overall, the random forest model had the best accuracy of 0.810 and F1 score of 0.868 to predict the outcome.

## Conclusion

Overall, machine learning using random forest model has modest improvement over logistic regression to predict whether the income of a person would be more or less than $50k. The advantage of accurate income classification would allow stakeholders to accurately predict income. This has multiple use cases - for finance institutions to cater for high or low income earners, for governments to plan for appropriate services for individuals earning <$50k a year. However, a binary classification of income is likely too broad, and having more income bands may be helpful. Another approach would be to treat income as a continuous variable, and machine learning approaches used to predict income.

Other potential options to deal with an unbalanced dataset include the use of SMOTE (Synthetic Minority Oversampling Technique). Undersampling using the ROSE package is also an option, however, runs the risk of loss of statistical power. Other machine learning approaches include using K-nearest neighbour and XGboost. Combining models in the form of ensembles may also help to improve performance.

## Executive summary

This project using an adult income dataset was a classification project to predict a binary outcome of whether an individual's income was more or less than \$50k/year. The original dataset contained 32561 observations of 15 variables. With data cleaning, missing data was identified and imputed or removed. 3 columns fnlwgt, capital gain and capital loss with multiple repeated values and unclear relation to the outcome were removed.

Following this, data visualisation was performed to analyse the relationship of features with the outcome. There were no highly correlated numeric values within the dataset. The dataset was then split into train and test sets, with oversampling method used to augment the train set given the unbalanced dataset.

The various machine learning models that were performed included logistic regression, random forest, support vector machines and decision tree, with the findings as follows. The random forest model was had the best prediction accuracy of 0.810 and F1 score of 0.868. Improving the prediction of income may have benefits for banking and government sectors among others. Further options for future projects include using income as a continuous variable, as well as other machine learning approaches such as k-nearest neighbour, XGboost and the use of ensembles.

Results of Machine Learning Approaches

```
## # A tibble: 4 x 5
##   Model                         Accuracy Sensitvity Specificity F1score
##   <chr>                            <dbl>      <dbl>       <dbl>   <dbl>
## 1 Logistic Regression Balanced     0.781      0.764       0.834   0.840
## 2 Random Forest                    0.810      0.828       0.755   0.868
## 3 Support Vector Machines          0.737      0.705       0.834   0.802
## 4 Decision Tree                    0.774      0.785       0.771   0.631
```

## References


1. Introduction to Data Science. Rafael A Irizarry. 2019. https://rafalab.dfci.harvard.edu/dsbook/
2. OpenAI. (2024). ChatGPT 3.5.
3. https://www.kaggle.com/datasets/wenruliu/adult-income-dataset
4. https://developers.google.com/machine-learning/crash-course/classification/accuracy-precision-recall