Lab719FinalCode

Cal Wardell

6/14/2022

Load packages

```
#Install needed libraries
library(ggplot2)
library(randomForest)
library(caret)
library(xgboost)
library(data.table)
library(mlr)
```

Goal

Determine which factors correlate with individual income to be greater or less than $50,000 per year.

About the data

The data for the analysis is extracted from the 1994 census, retrieved from the UCI Machine Learning Repository website (Dua, 2019). It provides over 32,000 rows containing information of the individual's age, working class, education, marital status, occupation, relationship to head of household on the census, race, sex, hours per week worked, native country, and whether the person made greater or less than $50,000 annually. Additionally, the data contained the following three columns that the website did not clearly explain: fnlwgt, capital-gain, and capital-loss.

This information can be used by local and national governments, non-profit groups, and other organizations that aim to raise individual incomes to decrease the rate of poverty.

Step 1: Clean and prep the data

My group and I examined the data for null values and transformed the needed columns to factors to be used in the machine learning model.

```
#Step one: Get and prepare data
adult.data <- read.csv(url("https://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data"))

#Rename columns.
```

```
col_names <- cbind("Age", "WorkClass", "Fnlwgt", "Education", "EducationNum",
"MaritalStatus", "Occupation",
                   "Relationship", "Race", "Sex", "CapitalGain",
"CapitalLoss", "HoursPerWeek", "NativeCountry",
                   "Value")

colnames(adult.data) <- col_names
str(adult.data)
```

```
## 'data.frame':    32560 obs. of  15 variables:
##  $ Age          : int  50 38 53 28 37 49 52 31 42 37 ...
##  $ WorkClass    : chr  " Self-emp-not-inc" " Private" " Private" "
Private" ...
##  $ Fnlwgt       : int  83311 215646 234721 338409 284582 160187 209642
45781 159449 280464 ...
##  $ Education    : chr  " Bachelors" " HS-grad" " 11th" " Bachelors" ...
##  $ EducationNum : int  13 9 7 13 14 5 9 14 13 10 ...
##  $ MaritalStatus: chr  " Married-civ-spouse" " Divorced" " Married-civ-
spouse" " Married-civ-spouse" ...
##  $ Occupation   : chr  " Exec-managerial" " Handlers-cleaners" " Handlers-
cleaners" " Prof-specialty" ...
##  $ Relationship : chr  " Husband" " Not-in-family" " Husband" " Wife" ...
##  $ Race         : chr  " White" " White" " Black" " Black" ...
##  $ Sex          : chr  " Male" " Male" " Male" " Female" ...
##  $ CapitalGain  : int  0 0 0 0 0 0 0 14084 5178 0 ...
##  $ CapitalLoss  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ HoursPerWeek : int  13 40 40 40 40 16 45 50 40 80 ...
##  $ NativeCountry: chr  " United-States" " United-States" " United-States"
" Cuba" ...
##  $ Value        : chr  " <=50K" " <=50K" " <=50K" " <=50K" ...
```

```
#Look for null values
sum(is.na(adult.data)) #0 so we can move forward
```

```
## [1] 0
```

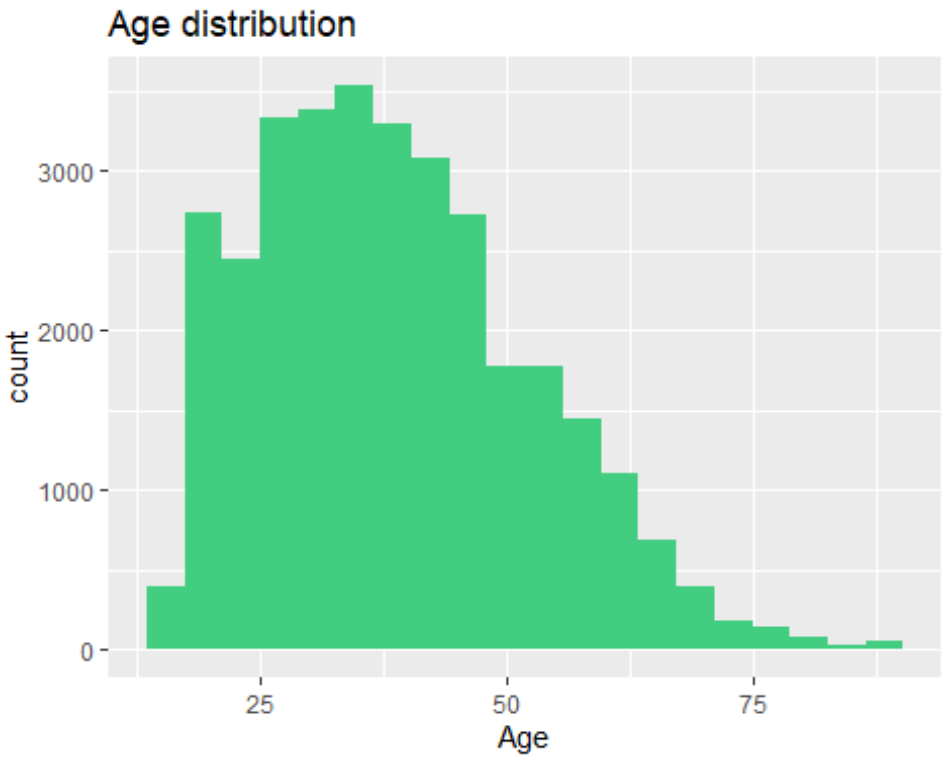
```
#Turn all the columns that are string into factors to be used for machine
learning
adult.data$WorkClass <- factor(adult.data$WorkClass)
adult.data$Education <- factor(adult.data$Education)
adult.data$MaritalStatus <- factor(adult.data$MaritalStatus)
adult.data$Occupation <- factor(adult.data$Occupation)
adult.data$Relationship <- factor(adult.data$Relationship)
adult.data$Race <- factor(adult.data$Race)
adult.data$Sex <- factor(adult.data$Sex)
adult.data$NativeCountry <- factor(adult.data$NativeCountry)
adult.data$Value <- factor(adult.data$Value)
str(adult.data)
```

```
## 'data.frame':    32560 obs. of  15 variables:
##  $ Age          : int  50 38 53 28 37 49 52 31 42 37 ...
```

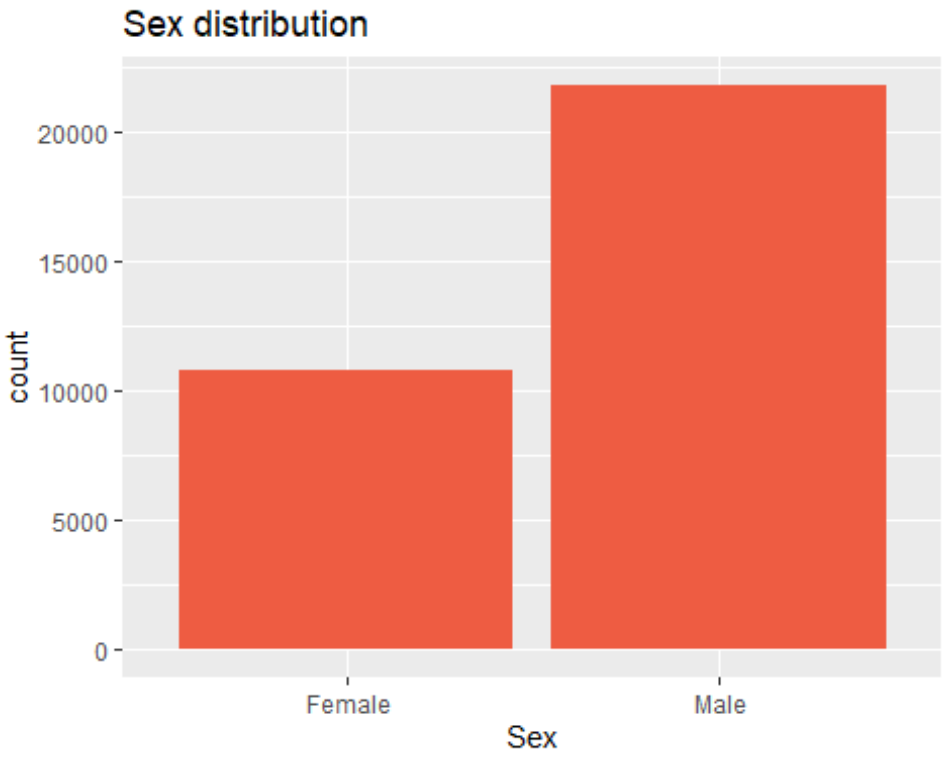
```
##  $ WorkClass    : Factor w/ 9 levels " ?"," Federal-gov",..: 7 5 5 5 5 5 7
5 5 5 ...
##  $ Fnlwgt       : int  83311 215646 234721 338409 284582 160187 209642
45781 159449 280464 ...
##  $ Education    : Factor w/ 16 levels " 10th"," 11th",..: 10 12 2 10 13 7
12 13 10 16 ...
##  $ EducationNum : int  13 9 7 13 14 5 9 14 13 10 ...
##  $ MaritalStatus: Factor w/ 7 levels " Divorced"," Married-AF-spouse",..:
3 1 3 3 3 4 3 5 3 3 ...
##  $ Occupation   : Factor w/ 15 levels " ?"," Adm-clerical",..: 5 7 7 11 5
9 5 11 5 5 ...
##  $ Relationship : Factor w/ 6 levels " Husband"," Not-in-family",..: 1 2 1
6 6 2 1 2 1 1 ...
##  $ Race         : Factor w/ 5 levels " Amer-Indian-Eskimo",..: 5 5 3 3 5 3
5 5 5 3 ...
##  $ Sex          : Factor w/ 2 levels " Female"," Male": 2 2 2 1 1 1 2 1 2
2 ...
##  $ CapitalGain  : int  0 0 0 0 0 0 0 14084 5178 0 ...
##  $ CapitalLoss  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ HoursPerWeek : int  13 40 40 40 40 16 45 50 40 80 ...
##  $ NativeCountry: Factor w/ 42 levels " ?"," Cambodia",..: 40 40 40 6 40
24 40 40 40 40 ...
##  $ Value        : Factor w/ 2 levels " <=50K"," >50K": 1 1 1 1 1 1 2 2 2 2
...
```

Step 2: Visualize the data

```
#Visualize the data
#Look at age
age <- ggplot(adult.data, aes(x = Age, fill = Age)) +
  geom_histogram(fill="seagreen3", bins = 20) +
  ggtitle('Age distribution')
age
```

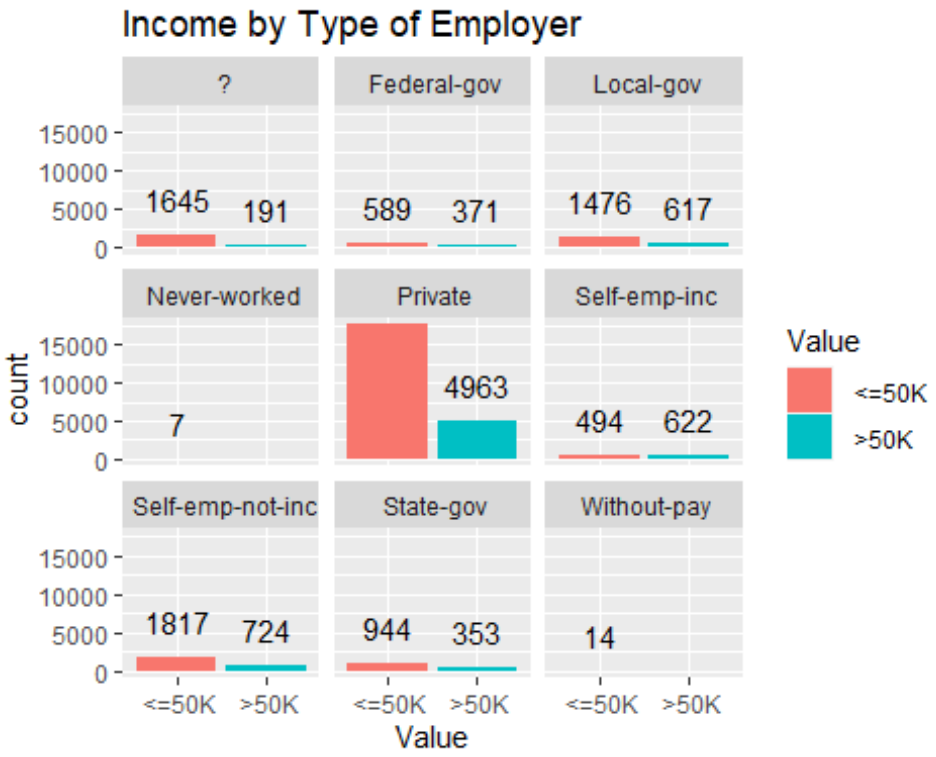



```r
#Look at sex
sex <- ggplot(adult.data, aes(x = Sex, fill = Sex)) +
  geom_bar(position = 'dodge', stat = 'count',fill="tomato2") +
  ggtitle('Sex distribution')
sex
```



We visualized the number of individuals who made more or less than $50,000 annually by type of employment.

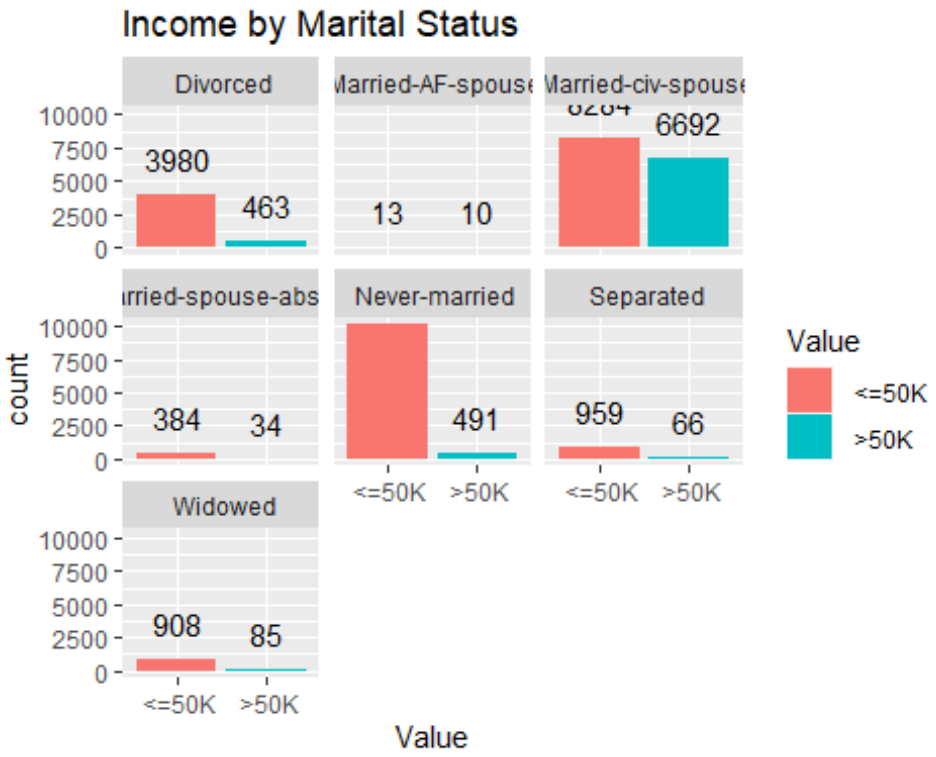
```
#Look at the income by type of employer
gWC <- ggplot(adult.data, aes(x = Value, fill = Value)) +
  geom_bar(position = 'dodge', stat = 'count') +
  stat_count(geom = 'text', color = 'black',
             aes(label = ..count.., vjust = -1)) +
  facet_wrap(~WorkClass) + ggtitle('Income by Type of Employer')

gWC
```



The data suggests that individuals who are employed by the federal government or who are self-employed with an incorporated business are more likely to be making over $50,000. However, the overall percentage of individuals in these situations is much less than the other employment situations.

Next, we visualized the number of those making more or less than $50,000 by marital status.

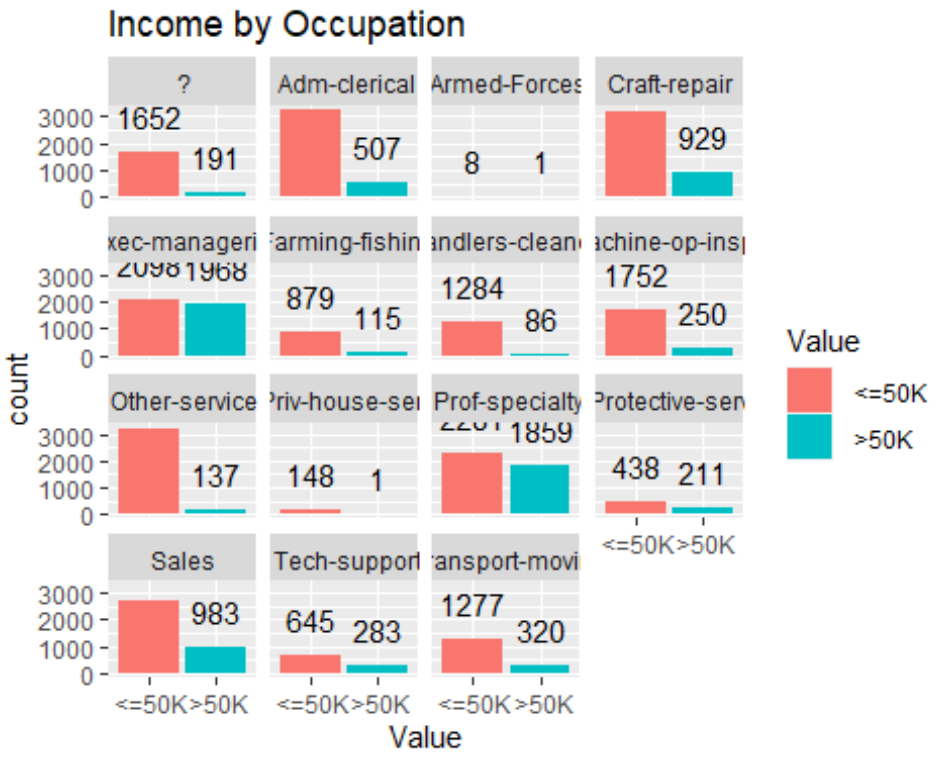
```
#By marital status
gMS <- ggplot(adult.data, aes(x = Value, fill = Value)) +
  geom_bar(position = 'dodge', stat = 'count') +
  stat_count(geom = 'text', color = 'black',
             aes(label = ..count.., vjust = -1)) +
  facet_wrap(~MaritalStatus) +
  ggtitle('Income by Marital Status')

gMS
```



The data implies that a married individual is more likely to earn over $50,000 than any other marital status.

We also Visualized the data by occupation

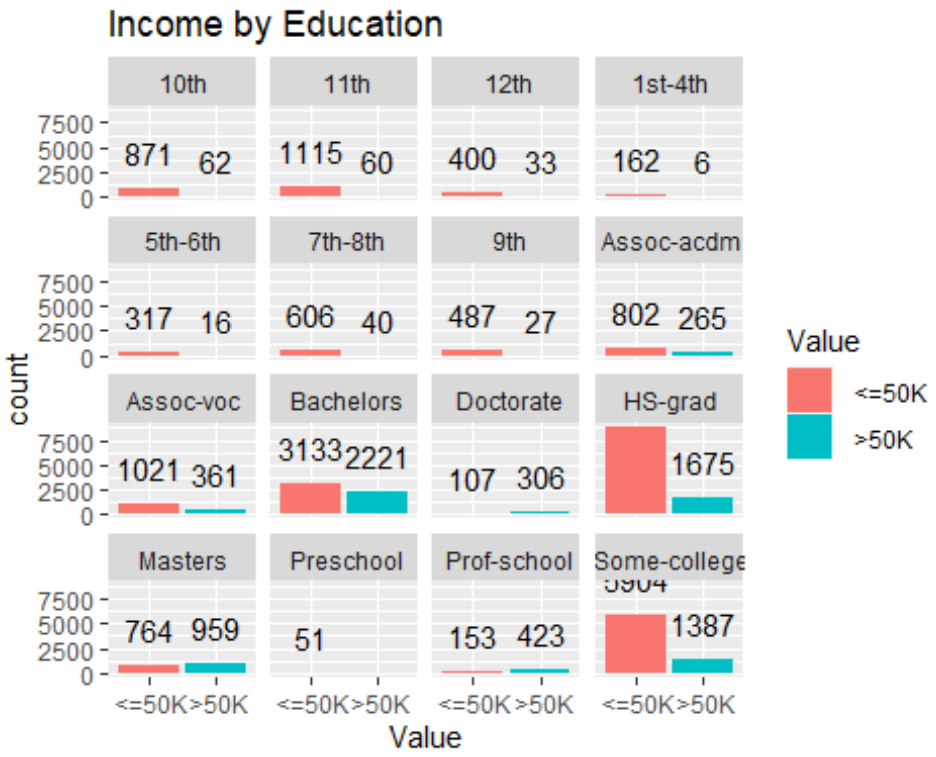
```
#By occupation
g0 <- ggplot(adult.data, aes(x = Value, fill = Value)) +
  geom_bar(position = 'dodge', stat = 'count') +
  stat_count(geom = 'text', color = 'black',
             aes(label = ..count.., vjust = -1)) +
  facet_wrap(~Occupation) +
  ggtitle('Income by Occupation')

g0
```



The occupation that has the best chance of earning over $50,000 are Managerial Executives and those who work in a speaclized profession.

Finally we visualized the data by education

```
#By Education
gEd <- ggplot(adult.data, aes(x = Value, fill = Value)) +
  geom_bar(position = 'dodge', stat = 'count') +
  stat_count(geom = 'text', color = 'black',
             aes(label = ..count.., vjust = -1)) +
  facet_wrap(~Education) +
  ggtitle('Income by Education')

gEd
```



We see that in general, the more education you received the higher chance you have at making more than $50,000.

Step 3: Create a test and training data set

Because there were 42 distinct countries in the data set, we chose to look primarily at whether the individual's native country was the United States.

```
adult.data$USAorNot <- as.integer(as.numeric(adult.data$NativeCountry))
adult.data$USAorNot <- ifelse(adult.data$USAorNot == 40, 1,0)
adult.data$USAorNot <- as.factor(adult.data$USAorNot)
head(adult.data)
```

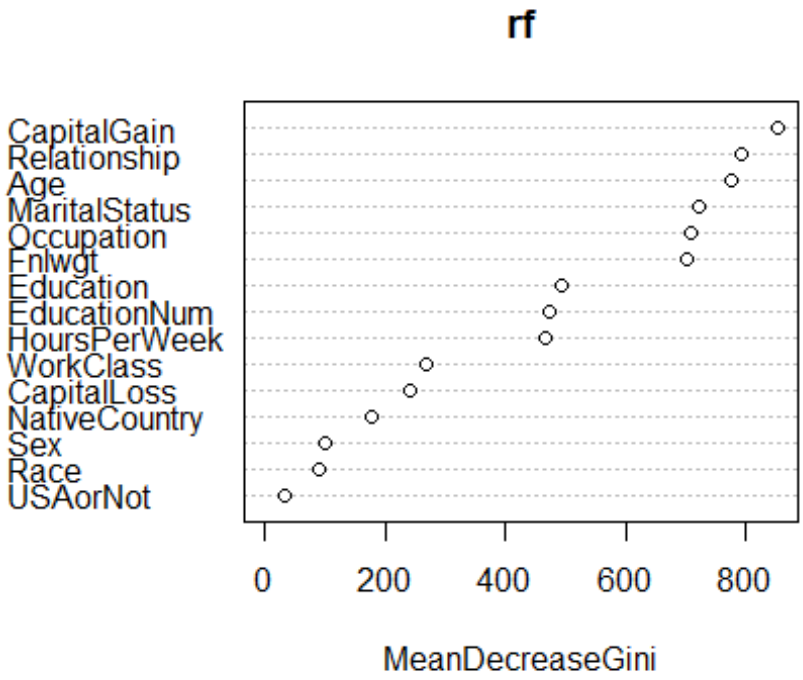
```
##   Age         WorkClass Fnlwgt  Education EducationNum
MaritalStatus
## 1  50  Self-emp-not-inc  83311  Bachelors           13     Married-civ-
spouse
## 2  38           Private 215646    HS-grad            9
Divorced
## 3  53           Private 234721       11th            7     Married-civ-
spouse
## 4  28           Private 338409  Bachelors           13     Married-civ-
spouse
## 5  37           Private 284582    Masters           14     Married-civ-
spouse
## 6  49           Private 160187        9th            5  Married-spouse-
```

```
absent
##           Occupation   Relationship   Race     Sex CapitalGain CapitalLoss
## 1    Exec-managerial        Husband  White    Male           0           0
## 2  Handlers-cleaners  Not-in-family  White    Male           0           0
## 3  Handlers-cleaners        Husband  Black    Male           0           0
## 4     Prof-specialty           Wife  Black  Female           0           0
## 5    Exec-managerial           Wife  White  Female           0           0
## 6      Other-service  Not-in-family  Black  Female           0           0
##   HoursPerWeek  NativeCountry  Value USAorNot
## 1           13  United-States  <=50K        1
## 2           40  United-States  <=50K        1
## 3           40  United-States  <=50K        1
## 4           40           Cuba  <=50K        0
## 5           40  United-States  <=50K        1
## 6           16        Jamaica  <=50K        0

set.seed(1234)
indices <- sample(nrow(adult.data), 0.70 * nrow(adult.data))
train <- adult.data[indices, ]
test <- adult.data[-indices, ]

str(train)

## 'data.frame':    22792 obs. of  16 variables:
##  $ Age          : int  53 20 42 31 43 46 29 40 23 37 ...
##  $ WorkClass    : Factor w/ 9 levels " ?"," Federal-gov",..: 5 5 2 5 5 5 5
5 5 5 ...
##  $ Fnlwgt       : int  238481 154779 284403 43716 142682 153501 34383
220563 141264 186934 ...
##  $ Education    : Factor w/ 16 levels " 10th"," 11th",..: 9 16 12 9 12 12
9 3 16 2 ...
##  $ EducationNum : int  11 10 9 11 9 9 11 8 10 7 ...
##  $ MaritalStatus: Factor w/ 7 levels " Divorced"," Married-AF-spouse",..:
3 5 1 1 3 5 3 1 5 3 ...
##  $ Occupation   : Factor w/ 15 levels " ?"," Adm-clerical",..: 5 13 2 8 15
15 15 4 5 8 ...
##  $ Relationship : Factor w/ 6 levels " Husband"," Not-in-family",..: 1 3 2
5 1 2 1 2 3 1 ...
##  $ Race         : Factor w/ 5 levels " Amer-Indian-Eskimo",..: 5 4 3 5 5 1
5 5 3 5 ...
##  $ Sex          : Factor w/ 2 levels " Female"," Male": 2 1 2 2 2 2 2 2 1
2 ...
##  $ CapitalGain  : int  0 0 0 0 0 0 0 0 0 3103 ...
##  $ CapitalLoss  : int  1485 0 0 0 0 0 0 0 0 0 ...
##  $ HoursPerWeek : int  40 40 40 43 30 40 55 40 40 44 ...
##  $ NativeCountry: Factor w/ 42 levels " ?"," Cambodia",..: 40 40 40 40 7
40 40 40 40 40 ...
##  $ Value        : Factor w/ 2 levels " <=50K"," >50K": 1 1 1 1 1 1 1 1 1 2
...
##  $ USAorNot     : Factor w/ 2 levels "0","1": 2 2 2 2 1 2 2 2 2 2 ...
```

```
str(test)

## 'data.frame':    9768 obs. of  16 variables:
##  $ Age          : int  38 28 37 31 42 30 23 32 32 43 ...
##  $ WorkClass    : Factor w/ 9 levels " ?"," Federal-gov",..: 5 5 5 5 5 8 5
5 5 7 ...
##  $ Fnlwgt       : int  215646 338409 284582 45781 159449 141297 122272
205019 186824 292175 ...
##  $ Education    : Factor w/ 16 levels " 10th"," 11th",..: 12 10 13 13 10
10 10 8 12 13 ...
##  $ EducationNum : int  9 13 14 14 13 13 13 12 9 14 ...
##  $ MaritalStatus: Factor w/ 7 levels " Divorced"," Married-AF-spouse",..:
1 3 3 5 3 3 5 5 5 1 ...
##  $ Occupation   : Factor w/ 15 levels " ?"," Adm-clerical",..: 7 11 5 11 5
11 2 13 8 5 ...
##  $ Relationship : Factor w/ 6 levels " Husband"," Not-in-family",..: 2 6 6
2 1 1 4 2 5 5 ...
##  $ Race         : Factor w/ 5 levels " Amer-Indian-Eskimo",..: 5 3 5 5 5 2
5 3 5 5 ...
##  $ Sex          : Factor w/ 2 levels " Female"," Male": 2 1 1 1 2 2 1 2 2
1 ...
##  $ CapitalGain  : int  0 0 0 14084 5178 0 0 0 0 0 ...
##  $ CapitalLoss  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ HoursPerWeek : int  40 40 40 50 40 40 30 50 40 45 ...
##  $ NativeCountry: Factor w/ 42 levels " ?"," Cambodia",..: 40 6 40 40 40
20 40 40 40 40 ...
##  $ Value        : Factor w/ 2 levels " <=50K"," >50K": 1 1 1 2 2 2 1 1 1 2
...
##  $ USAorNot     : Factor w/ 2 levels "0","1": 2 1 2 2 2 1 2 2 2 2 ...
```

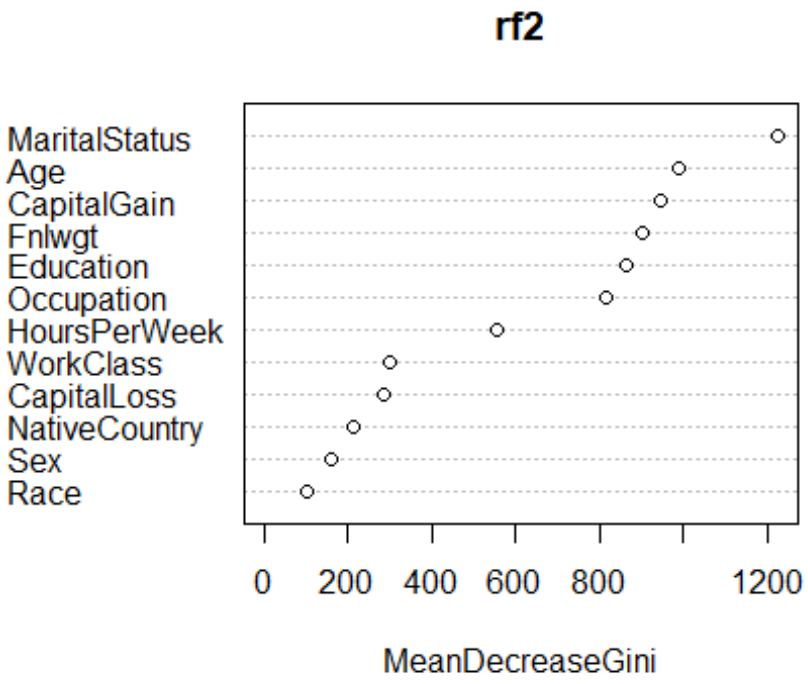
Step 4: Create and test the Random Forest model

```
#Random Forrest
rf <- randomForest(Value ~ ., data = train)
rf

##
## Call:
##  randomForest(formula = Value ~ ., data = train)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 13.76%
## Confusion matrix:
##         <=50K  >50K class.error
##  <=50K  16023  1184  0.06880921
##  >50K    1952  3633  0.34950761

varImpPlot(rf) #Shows us which variables were most important to our tree
```



First, we built a Random Forest model with all the variables. The graph above shows the significance of each variable in the model. USAorNot was insignificant, so we reran the model without it. Additionally, because Education and EducationNum both refer to the same characteristic, we did not include EducationNum in the model. Finally, we removed Relationship as we felt that Marital Status was sufficient. After running the model again, the graph below depicts the significance of the variables:

```
test2 <- subset(test, select = -c(USAorNot, EducationNum,Relationship))
train2 <- subset(train, select = -c(USAorNot, EducationNum,Relationship))

rf2 <- randomForest(Value ~ ., data = train2)
varImpPlot(rf2)  #See how meaningful the variables are
```



Here are the initial results of our model

```
rf2

##
## Call:
##  randomForest(formula = Value ~ ., data = train2)
##                Type of random forest: classification
##                      Number of trees: 500
## No. of variables tried at each split: 3
##
##         OOB estimate of  error rate: 13.91%
## Confusion matrix:
##         <=50K  >50K class.error
##  <=50K  16008  1199  0.06968094
##  >50K    1971  3614  0.35290958
```

Our model had an initial accuracy of 86.2%. It was 93.03% accurate at predicting below or equal to $50,000, but only 65.18% accurate at predicting if the individual made more than $50,000. We then tested our model; below are the results:

```
#Test our model
predrf = predict(rf2, newdata=test2[,-13])
comptablerf <- data.frame(test2[,13],predrf)
#Analyse results
confusionMatrix(comptablerf$test2...13, comptablerf$predrf)
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction  <=50K  >50K
##      <=50K   7017   495
##      >50K     802  1454
##
##                Accuracy : 0.8672
##                  95% CI : (0.8603, 0.8739)
##     No Information Rate : 0.8005
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6075
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.8974
##             Specificity : 0.7460
##          Pos Pred Value : 0.9341
##          Neg Pred Value : 0.6445
##              Prevalence : 0.8005
##          Detection Rate : 0.7184
##    Detection Prevalence : 0.7690
##       Balanced Accuracy : 0.8217
##
##        'Positive' Class :  <=50K
##
```

With a p-value well below the 0.05 mark, our model was statistically significant. It was 86.75% accurate, and more accurate at predicting those who make less than or equal to $50,000 than those who make more than $50,000.

Step 5: Create and test the XGBoost model

We then created an XGBoost model on the data. We ran it with the default parameters and here are the results when we ran it against the test:

```
#To run XG boost, we need it to be in a binary format. -1 makes it so <50k is
0 and >= 50 k is 1
#Need data to be in tables
train3 <- setDT(train2)
test3 <- setDT(test2)
y <- as.numeric(train2$Value) - 1
test_y <- as.numeric(test2$Value) - 1
prep_xg_train <- model.matrix(~.+0,data = train3[,-c("Value"),with=F]) #~.+0
leads to encoding of all categorical variables without producing an intercept
prep_xg_test <- model.matrix(~.+0,data = test3[,-c("Value"),with=F])

#Create matrix
```

```
xg_train <- xgb.DMatrix(data = prep_xg_train, label = y)
xg_test <- xgb.DMatrix(data = prep_xg_test, label = test_y)


#set parameters, use default first
params <- list(booster = "gbtree", objective = "binary:logistic", eta=0.3,
gamma=0, max_depth=6,
               min_child_weight=1, subsample=1, colsample_bytree=1)
#Calculate best number of rounds
set.seed(1234)
xgb_cv <- xgb.cv(params = params, data = xg_train, nrounds = 100, nfold = 5,
showsd = T, stratified = T,
                 print_every_n = 10, early_stopping_rounds = 20, maximize =
F)

## [19:40:05] WARNING: amalgamation/../src/learner.cc:1115: Starting in
XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
## [19:40:05] WARNING: amalgamation/../src/learner.cc:1115: Starting in
XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
## [19:40:05] WARNING: amalgamation/../src/learner.cc:1115: Starting in
XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
## [19:40:05] WARNING: amalgamation/../src/learner.cc:1115: Starting in
XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
## [19:40:05] WARNING: amalgamation/../src/learner.cc:1115: Starting in
XGBoost 1.3.0, the default evaluation metric used with the objective
'binary:logistic' was changed from 'error' to 'logloss'. Explicitly set
eval_metric if you'd like to restore the old behavior.
## [1]  train-logloss:0.546397+0.000817 test-logloss:0.547863+0.001435
## Multiple eval metrics are present. Will use test_logloss for early
stopping.
## Will train until test_logloss hasn't improved in 20 rounds.
##
## [11] train-logloss:0.298295+0.001035 test-logloss:0.312183+0.006193
## [21] train-logloss:0.268683+0.001713 test-logloss:0.294094+0.006324
## [31] train-logloss:0.255776+0.002235 test-logloss:0.289515+0.006181
## [41] train-logloss:0.246466+0.002041 test-logloss:0.287417+0.006348
## [51] train-logloss:0.238425+0.002135 test-logloss:0.286307+0.006906
## [61] train-logloss:0.231497+0.002614 test-logloss:0.286453+0.007312
## [71] train-logloss:0.225316+0.001840 test-logloss:0.286988+0.007506
## Stopping. Best iteration:
## [51] train-logloss:0.238425+0.002135 test-logloss:0.286307+0.006906
```

```r
#Best number of rounds is 51

#Run model with defaults
xg_1 <- xgb.train(params = params, data = xg_train, nrounds = 51,
                  watchlist = list(val = xg_test, train = xg_train),
print_every_n = 10,
                  maximize = T, eval_metric = "error")
```

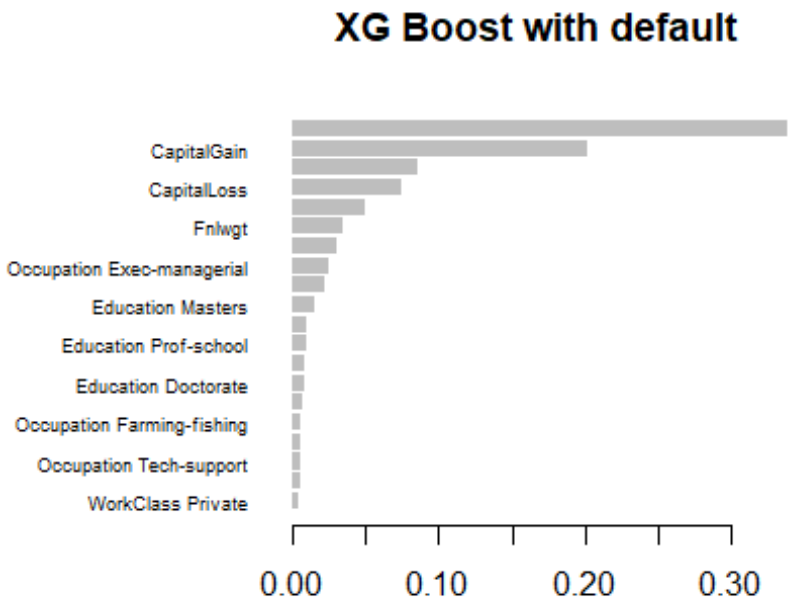
```
## [1]  val-error:0.140868  train-error:0.146938
## [11] val-error:0.131347  train-error:0.131362
## [21] val-error:0.126638  train-error:0.122499
## [31] val-error:0.124386  train-error:0.115698
## [41] val-error:0.123771  train-error:0.112408
## [51] val-error:0.124181  train-error:0.109556
```

```r
#Make the prediction
xg_1_predict <- predict(xg_1, xg_test)
xg_1_predict <- ifelse(xg_1_predict > 0.5, 1, 0)

#Confusion matrix
confusionMatrix(as.factor(xg_1_predict), as.factor(test_y))
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 7090  791
##          1  422 1465
##
##                Accuracy : 0.8758
##                  95% CI : (0.8691, 0.8823)
##     No Information Rate : 0.769
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.6292
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9438
##             Specificity : 0.6494
##          Pos Pred Value : 0.8996
##          Neg Pred Value : 0.7764
##              Prevalence : 0.7690
##          Detection Rate : 0.7258
##    Detection Prevalence : 0.8068
##       Balanced Accuracy : 0.7966
##
##        'Positive' Class : 0
##
```

This was more accurate than our Random Forest model. This model was statistically significant as it had a p-value less than 0.05. Below shows a graph depicting which variables had the greatest impact on our model. Like Random Forest, marital status had the greatest influence.

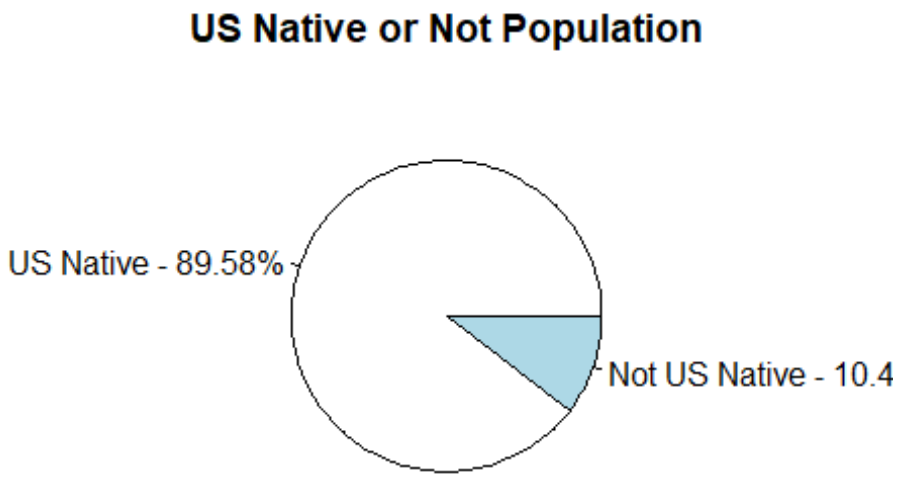
```
important <- xgb.importance(feature_names = colnames(xg_train), model = xg_1)
xgb.plot.importance(importance_matrix = important[1:20], main = 'XG Boost
with default')
```



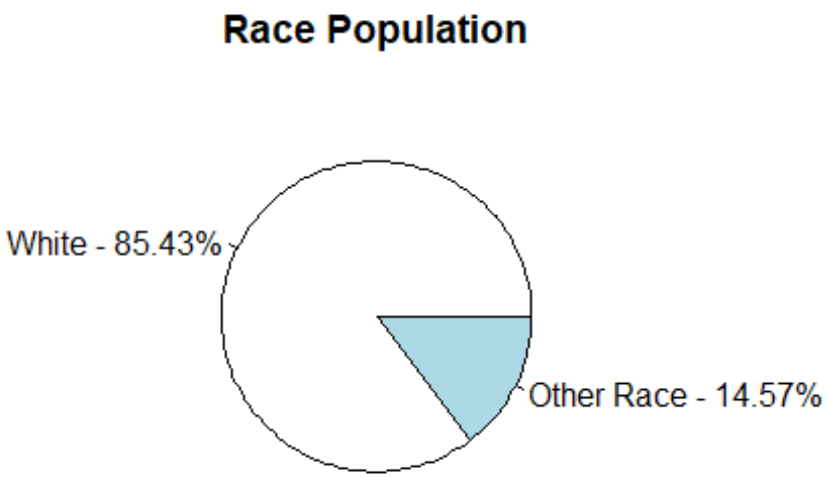
Acknowledgement of potential bias To be thorough in our analysis, we assessed the demographics to see if we had potential bias. We looked at ratio for US Native vs Not US Native, White vs Other Race, and Male vs Female, which is pictured below:

```
#Look at ratios of some of the demographics to look for potential bias
adult.data2 <- data.frame(adult.data)
adult.data2$USAorNot <- as.integer(as.numeric(adult.data2$NativeCountry))
adult.data2$USAorNot <- ifelse(adult.data2$USAorNot == 40, 1,0)
#US Native vs Non-US Native
us_ratio <- sum(adult.data2$USAorNot)/nrow(adult.data2)
us_ratio *100

## [1] 89.58538

USNative <- sum(adult.data2$USAorNot)
NotNative <- nrow(adult.data2) - USNative
native <- c(USNative, NotNative)
```

```
pie(native, labels = c("US Native - 89.58%", "Not US Native - 10.42%"), main
= "US Native or Not Population")
```




```
#Race
adult.data2$WhiteorNot <- as.integer(as.numeric(adult.data2$Race))
adult.data2$WhiteorNot <- ifelse(adult.data2$WhiteorNot == 5, 1,0)
White_ratio <- sum(adult.data2$WhiteorNot)/nrow(adult.data2)
White_ratio *100

## [1] 85.4269

wRace <- sum(adult.data2$WhiteorNot)
oRace <- nrow(adult.data2) - wRace
Race <- c(wRace, oRace)
pie(Race, labels = c("White - 85.43%", "Other Race - 14.57%"), main = "Race
Population")
```



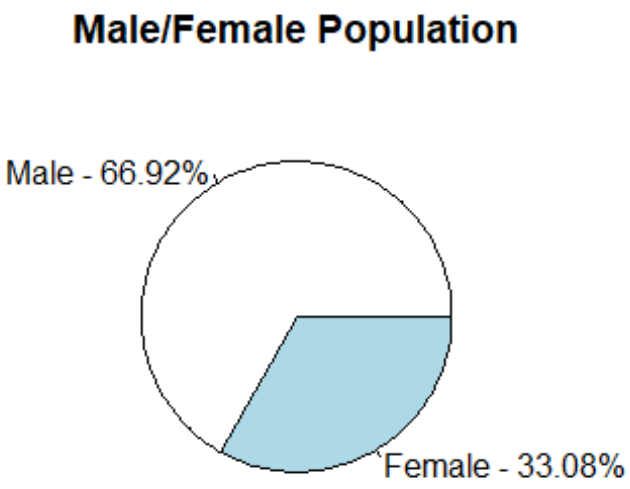

```
#Sex
adult.data2$MaleorNot <- as.integer(as.numeric(adult.data2$Sex))
adult.data2$MaleorNot <- ifelse(adult.data2$MaleorNot == 2, 1,0)
Male_ratio <- sum(adult.data2$MaleorNot)/nrow(adult.data2)
Male_ratio *100
```

```
## [1] 66.91953
```

```
male <- sum(adult.data2$MaleorNot)
female <- nrow(adult.data2) - male
malefemale <- c(male, female)
str(malefemale)
```

```
##  num [1:2] 21789 10771
```

```
pie(malefemale, labels = c("Male - 66.92%", "Female - 33.08%"), main =
"Male/Female Population")
```



Conclusion

Our XGBoost model was 87% accurate at predicting if an individual would make greater or less than $50,000 annually. The most influential variable to determine whether a person makes more or less than $50,000 was marital status. The graphs suggest that married individuals had a better chance of making greater than $50,000 than those who are not married. Our recommendation is to increase programs and incentives that encourage individuals to get married, as this will increase the likelihood of earning $50,000 or more annually.