Choose Your Own Project

HarvardX Data Science Capstone

Nadim Yatim

Table of Contents

Introduction

Census income is known as the income that an individual receives before completing certain payments such as personal income taxes, social security, union dues and others. In some cases, as household surveys, some individuals tend to underreport their income. Our dataset is extracted from the 1994 Census bureau database by Ronny Kohavi and Barry Becker and includes adults that have reported their census income after also getting asked to provide their information regarding characteristics such as age, work class, marital status and many others. This project requires the prediction of whether an individual makes over $50K per year or not and different machine learning models are going to be considered to achieve these predictions. The obtained predictions are going to be assessed using the obtained accuracy and the F1 score.

Methods/Analysis

Data Exploration and Visualization

Data Exploration

The dataset is made up of 32561 observations and has 15 features. Each row in this dataset is considered to have the income for each individual having a specific set of features. The features, their classes and descriptions are as follows:

Feature	Class	Description
Age	Numeric	The age of each individual
Workclass	Character	The employment status of each individual having the following possibilities: Private, State-gov, Federal-gov, Sel-emp-not-inc,Self-emp-inc,Local-gov,Without-pay,Never-worked
Fnlwgt	Numeric	The final weight referring to the population totals created by weighted tallies of any specified socio-economic characteristic of the population
Education	Character	The educational level of each individual having the following possibilities: HS-grad, Some-college, 7^{th}-8^{th} , 10^{th}-, Doctorate , Prof-school, Bachelors, Masters, 11^{th} – Assoc-acdm, Assoc-voc, 1^{st}-4^{th}, 5^{th}-6^{th}, 12^{th}, 9^{th}, Preschool
Education.num	Numeric	The educational level of each individual in numerical values ranging from 1 to 16
Marital.Status	Character	The marital status of each individual having the following possibilities: Widowed, Divorced, Seperated, Never-married, Married-civ-spouse, Married-spouse-abscent, Married-AF-spouse
Occupation	Character	The job type of each individual having he following possibilities: Exec-managerial, Machine-op-inspct, Prof-specialty, Other-service, Adm-clerical, Craft-repair,

		Transport-moving, Handlers-cleaner, Sales, Farming-fishing, Tech-support, Protective-serv, Armed-Forces, Priv-house-serv
Relationship	Character	The relationship status of each individual having the following possibilities: Not-in-family, Unmarried, Own-child, Other-relative, Husband, Wife
Race	Character	The race of each individual having the following possibilities: White, Black, Asian-Pac-Islander, Other, Amer-Indian-Eskimo
Sex	Character	The sex of each individual
Capital Gain	Numeric	The capital gain of each individual
Capital Loss	Numeric	The capital loss of each individual
Hours Per Week	Numeric	The number of hours that each individual works per week
Native Country	Character	The native country of each individual
Income	Character	The income of each individual having the following possibilities: <=50k, >50k

A sample of the data as well as a summary of each feature is as follows:

```
    age workclass fnlwgt education education.num marital.status
  <dbl> <chr>      <dbl> <chr>             <dbl> <chr>
1    90 ?          77053 HS-grad               9 Widowed
2    82 Private   132870 HS-grad               9 Widowed
3    66 ?         186061 Some-col…            10 Widowed
4    54 Private   140359 7th-8th               4 Divorced
5    41 Private   264663 Some-col…            10 Separated
6    34 Private   216864 HS-grad               9 Divorced
# … with 9 more variables: occupation <chr>, relationship <chr>,
#   race <chr>, sex <chr>, capital.gain <dbl>,
#   capital.loss <dbl>, hours.per.week <dbl>,
#   native.country <chr>, income <chr>
```

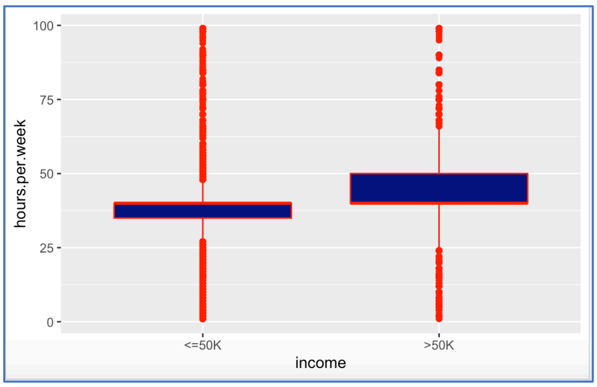
```
      age          workclass               fnlwgt
 Min.   :17.00   Length:32561       Min.   :  12285
 1st Qu.:28.00   Class :character   1st Qu.: 117827
 Median :37.00   Mode  :character   Median : 178356
 Mean   :38.58                      Mean   : 189778
 3rd Qu.:48.00                      3rd Qu.: 237051
 Max.   :90.00                      Max.   :1484705
  education          education.num   marital.status
 Length:32561       Min.   : 1.00   Length:32561
 Class :character   1st Qu.: 9.00   Class :character
 Mode  :character   Median :10.00   Mode  :character
                    Mean   :10.08
                    3rd Qu.:12.00
                    Max.   :16.00
  occupation        relationship           race
 Length:32561       Length:32561       Length:32561
 Class :character   Class :character   Class :character
 Mode  :character   Mode  :character   Mode  :character



     sex              capital.gain    capital.loss
 Length:32561       Min.   :    0   Min.   :   0.0
 Class :character   1st Qu.:    0   1st Qu.:   0.0
 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        income
 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
```

Data Visualization

Moreover, we now need to assess and visualize the effect of the features on income. The effect of numerical features are going to be visualized using boxplots while character features are visualized using bar plots.

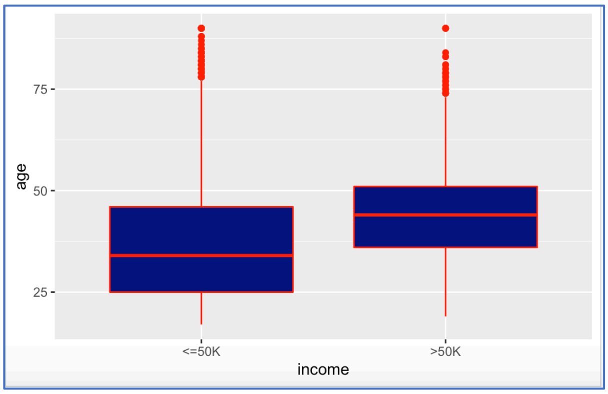
Effect of Working Hours per week

It is clear that an increased income which is more than 50k is associated with having higher number of working hours per week



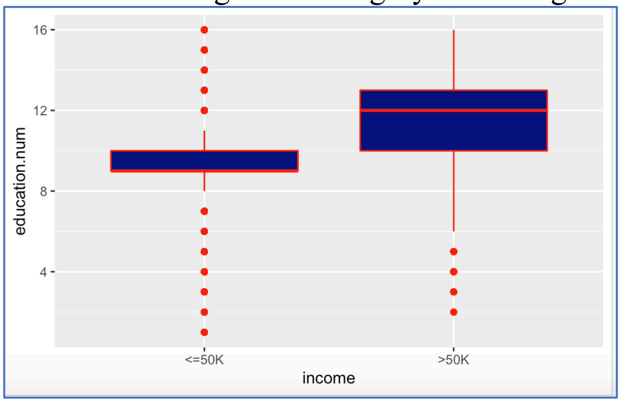
Effect of age

As for age, we can see that as individuals get older they are more likely to earn more than 50k than to earn less than this amount



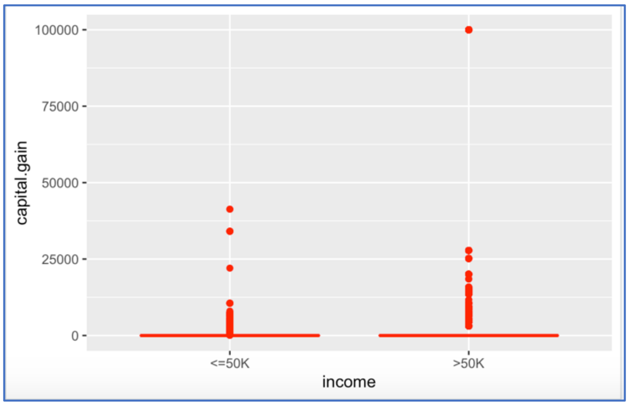
Effect of education level

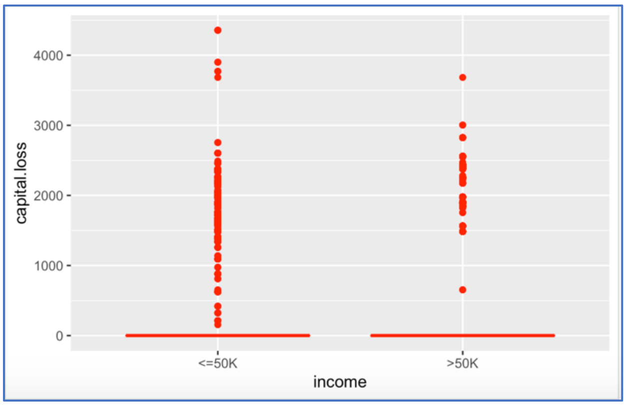
Moreover, higher levels of education are more likely to result in earning higher levels of income and being in the category of earning more than 50k



Effect of capital gain and capital loss

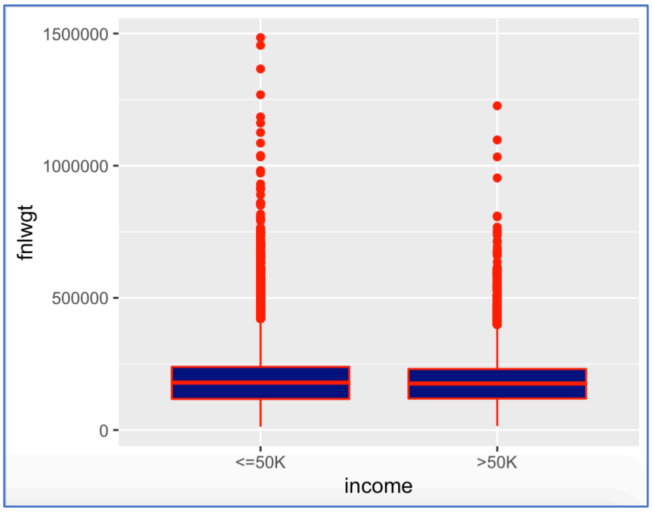
Furthermore, capital gain and loss can affect the level of income and higher capital gain and capital loss are associated with an income of more than 50k





Effect of fnlwgt

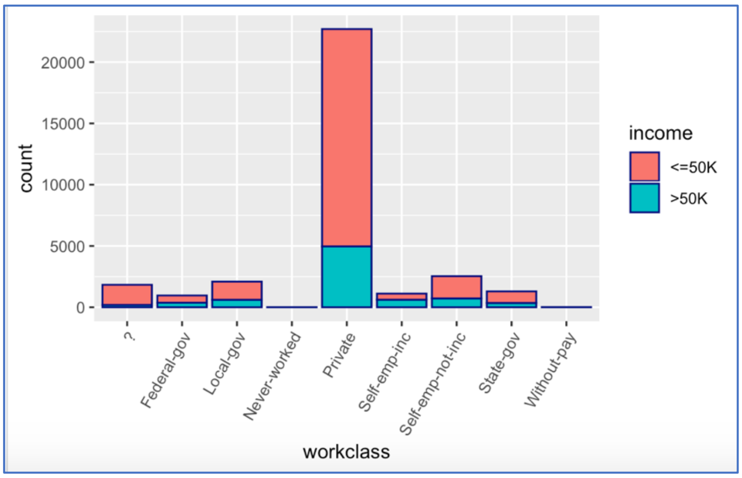
Regarding the fnlwgt feature, we can notice, as seen in the boxplot below, that individuals earning more than or less than 50k per year are of the same weights approximately.

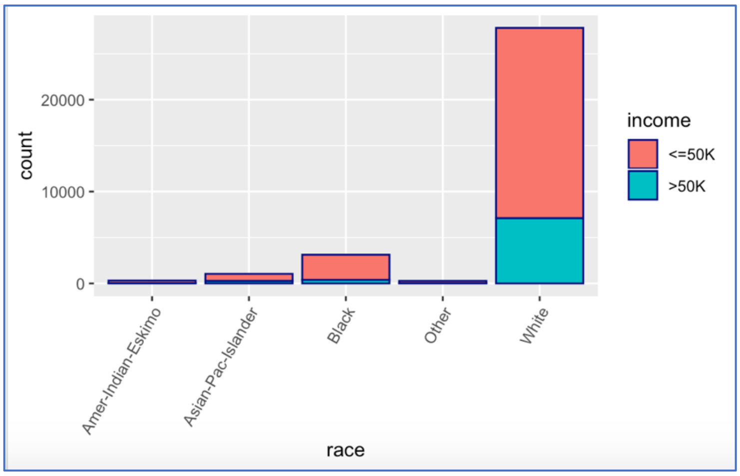


As for the features having the type character, their effect on income are shown in the following bar plots.

Effect of Workclass and Race

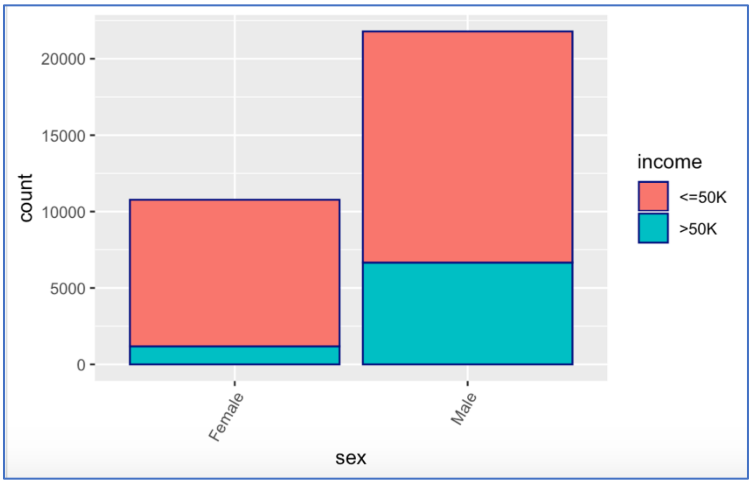
As seen below, working for the private sector increases the chances of the individual for earning more than 50k. Moreover, individual from the white race are more than others that earn more than 50k

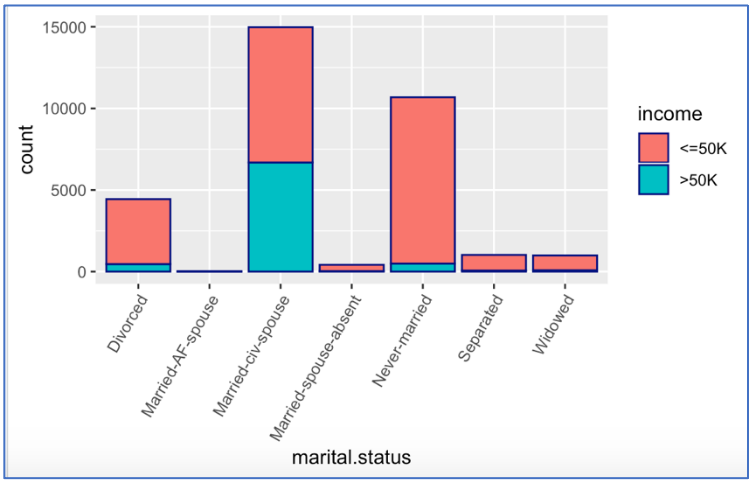




Effect of Sex and Marital Status

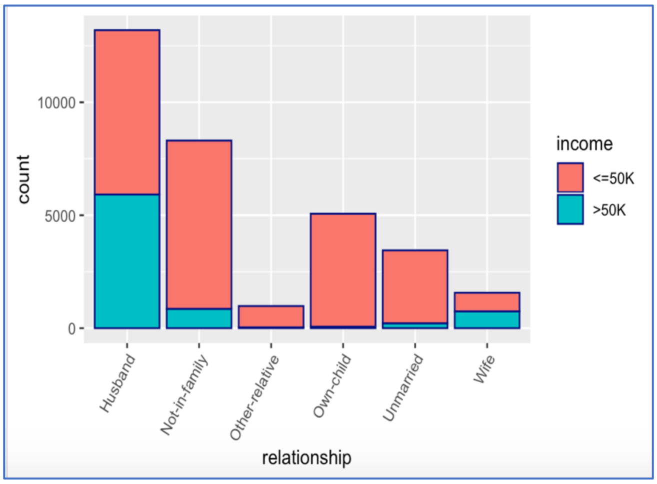
According to our dataset, 50k and more incomes are earned more by males than females and being of the marital status "Married-civ-spouse" also appear to earn more than the other categories.

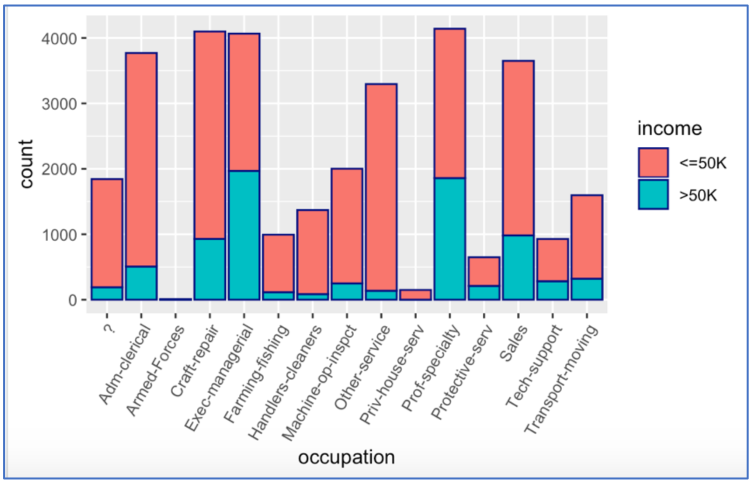




Effect of Relationship and Occupation

From our individuals in the dataset, Husbands more than any other relationship category, by a percentage of approximately 50% earn more than 50k. Whereas the number of individuals having an occupation of "Exec-managerial" and "Prof-specialty" are by far greater than those earning more than 50k in other categories.





Data Split: Training and Test Sets

In order to mimic the evaluation process of machine learning algorithms we need to split our data into two parts which are the training set(for which we pretend to know the outcome) and the test set(for which we pretend not to know the outcome) . That's why we decide on splitting the data into both sets having 90% of the data in the training set and 10% of the data in the test set. This is better than using a 50/50 split among training and test sets in our case because it will allow us to improve our predictions based on the metrics, such as accuracy and F1 score, while evaluating the machine learning algorithms.

Modeling Approach

Metrics

For the assessment of each model, we will use two metrics which are overall accuracy and the F1 score. Overall accuracy shows us how much the algorithm that is being tested is able to correctly predict a certain outcome (whether income is <=50K or >50K in our case) based on feature values that are taken as input. In addition, the F1 score is a measure that allows us to have a harmonic average of specificity and sensitivity and in our case a higher F1 score is preferred and can be an indicator about the performance of the machine learning model

Models

Logistic Regression

Being an extension of the linear regression, the logistic regression model will be able in our case to have an estimate of the conditional probability to be between 0 and 1. It also allows for the usage of the logistic transformation which converts probabilities to log odds as seen below

$$g\,(p) = log\frac{p}{1-p}$$

This transformation also allows for the probabilities to become symmetric around 0. In order to fit the logistic regression model, we have to use the maximum likelihood estimate. The model is fit as follows:

```
train_glm <- train(income ~ .,
                   method = "glm",
                   data = train_set)
```

After fitting the model and completing the predictions, the obtained confusion matrix is shown below. The accuracy of the logistic regression model on the test set is **0.8480196** and the calculated F1 score is **0.9025015**

```
Confusion Matrix and Statistics

          Reference
Prediction <=50K >50K
     <=50K  2291  314
     >50K    181  471

               Accuracy : 0.848
                 95% CI : (0.8352, 0.8602)
    No Information Rate : 0.759
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5591

 Mcnemar's Test P-Value : 2.975e-09

            Sensitivity : 0.9268
            Specificity : 0.6000
         Pos Pred Value : 0.8795
         Neg Pred Value : 0.7224
             Prevalence : 0.7590
         Detection Rate : 0.7034
   Detection Prevalence : 0.7998
      Balanced Accuracy : 0.7634

       'Positive' Class : <=50K
```

```
|Model               |  Accuracy|   F1score|
|:-------------------|---------:|---------:|
|Logistic Regression | 0.8480196| 0.9025015|
```

Linear Discriminate Analysis

The quadratic discriminant analysis model is known to be an extension to the naïve Byes which assumes that the conditional probabilities are considered to be multivariate normal. This will allow the assumption of the conditional distributions to be bivariate normal. But due to the large number of predictors the QDA model is replaced by the LDA model which assumes the same correlation structure for all classes reducing the number of parameters that need to be estimated leading to the same standard deviation and correlations.

Fitting the model is done as the code shown below:

```
train_lda <- train(income ~ .,
                   method = "lda",
                   data = train_set)
```

After fitting the model and completing the predictions, the obtained confusion matrix is shown below. Also, as expected the accuracy, having a value of **0.8369665**, is not considered to be high which is due to the lack of flexibility and the F1 score was calculated to be **0.8959028**

```
Confusion Matrix and Statistics

          Reference
Prediction <=50K >50K
     <=50K  2285  344
     >50K    187  441

               Accuracy : 0.837
                 95% CI : (0.8238, 0.8495)
    No Information Rate : 0.759
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5217

 Mcnemar's Test P-Value : 1.289e-11

            Sensitivity : 0.9244
            Specificity : 0.5618
         Pos Pred Value : 0.8692
         Neg Pred Value : 0.7022
             Prevalence : 0.7590
         Detection Rate : 0.7016
   Detection Prevalence : 0.8072
      Balanced Accuracy : 0.7431

       'Positive' Class : <=50K
```

```
|Model                        |  Accuracy|   F1score|
|:----------------------------|---------:|---------:|
|Logistic Regression          | 0.8480196| 0.9025015|
|Linear Discriminant Analysis | 0.8369665| 0.8959028|
```

Decision Tree

The outcome in our case, which we are basing our prediction on, is the income. As seen previously, this feature is considered to be categorical. Thus, using classification(decision) trees are valid in this case. At the end of each node, the prediction is based on the class that has the majority vote.

This model, which could be used for modeling decision processes, is known for the ease at which it can be visualized and the high interpretability property that specializes it.

The code that is used in order to fit the decision tree model is shown below

```
train_rpart <- train(income ~ .,
                     method = "rpart",
                     data = train_set)
```

Upon constructing the confusion matrix, and as expected upon calculation, we obtain a low value of accuracy of **0.8308259** and a value of **0.8947067** for the F1 score. The low accuracy is explained by being not very flexible and the high instability to changes that are in the training set.

```
Confusion Matrix and Statistics

          Reference
Prediction <=50K >50K
     <=50K  2341  420
     >50K    131  365

               Accuracy : 0.8308
                 95% CI : (0.8175, 0.8436)
    No Information Rate : 0.759
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.4712

 Mcnemar's Test P-Value : < 2.2e-16

            Sensitivity : 0.9470
            Specificity : 0.4650
         Pos Pred Value : 0.8479
         Neg Pred Value : 0.7359
             Prevalence : 0.7590
         Detection Rate : 0.7188
   Detection Prevalence : 0.8477
      Balanced Accuracy : 0.7060

       'Positive' Class : <=50K
```

Model	Accuracy	F1score
Logistic Regression	0.8480196	0.9025015
Linear Discriminant Analysis	0.8369665	0.8959028
Decision Tree	0.8308259	0.8947067

Random Forest

As seen in the previous model, the classification(decision) tree, there are several flaws. Random forests can be used to address those shortcomings by reducing the instability and improving the obtained prediction performance. This is accomplished by averaging several decision trees, and thus obtaining a forest which is characterized by its randomness. We make sure that the trees that are obtained are unique and different from one another by using bootstrap to include the factor of randomness.

The random forest model is fit as follows and as we can see we indicate the number of trees to be equal to 7.

```
train_rforest <- train(income ~ .,
                          method = "rf",
                          data = train_set,
                          ntree= 5,
                          importance=TRUE)
```

As expected, and after the construction of the confusion matrix, we have an improvement of the accuracy to reach a value of **0.8455634.** Also, the F1 score increases from the previous model and has a value of **0.8995808**

```
Confusion Matrix and Statistics

          Reference
Prediction <=50K >50K
     <=50K  2253  284
     >50K    219  501

               Accuracy : 0.8456
                 95% CI : (0.8327, 0.8578)
    No Information Rate : 0.759
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5656

 Mcnemar's Test P-Value : 0.004322

            Sensitivity : 0.9114
            Specificity : 0.6382
         Pos Pred Value : 0.8881
         Neg Pred Value : 0.6958
             Prevalence : 0.7590
         Detection Rate : 0.6917
   Detection Prevalence : 0.7789
      Balanced Accuracy : 0.7748

       'Positive' Class : <=50K
```

Model	Accuracy	F1score
Logistic Regression	0.8480196	0.9025015
Linear Discriminant Analysis	0.8369665	0.8959028
Decision Tree	0.8308259	0.8947067
Random Forest	0.8455634	0.8995808

Ensemble

For further enhancements and improvements to the results obtained above by the predictions made from various machine learning methods, we can combine these results obtained.

The ensemble model, its accuracy, the confusion matrix and the corresponding F1 score are obtained as follows

```r
#Caclulating the accuracy and constructing the confusion matrix
ensemble <- cbind(glm = glm_preds=="<=50K" , lda = lda_preds=="<=50K", decision=rpart_preds=="<=50K", randomforest=rforest_preds=="<=50K")

ensemble_preds <- ifelse(rowMeans(ensemble) > 0.5, "<=50K", ">50K")
ensemble_accuracy<-mean(ensemble_preds == test_set$income)
confusionMatrix(factor(ensemble_preds), reference = factor(test_set$income))
#Calculating the F1 score
ensemble_F1<- F_meas(factor(ensemble_preds), factor(test_set$income))
```

As seen below, the accuracy obtained is **0.8520111** which is an improvement among all other models and the F1 score is **0.9045922**.

```
Confusion Matrix and Statistics

          Reference
Prediction <=50K >50K
     <=50K  2285  295
     >50K    187  490

               Accuracy : 0.852
                 95% CI : (0.8393, 0.864)
    No Information Rate : 0.759
    P-Value [Acc > NIR] : < 2.2e-16

                  Kappa : 0.5756

 Mcnemar's Test P-Value : 1.095e-06

            Sensitivity : 0.9244
            Specificity : 0.6242
         Pos Pred Value : 0.8857
         Neg Pred Value : 0.7238
             Prevalence : 0.7590
         Detection Rate : 0.7016
   Detection Prevalence : 0.7921
      Balanced Accuracy : 0.7743

       'Positive' Class : <=50K
```

```
|Model                        |  Accuracy|   F1score|
|:----------------------------|---------:|---------:|
|Logistic Regression          | 0.8480196| 0.9025015|
|Linear Discriminant Analysis | 0.8369665| 0.8959028|
|Decision Tree                | 0.8308259| 0.8947067|
|Random Forest                | 0.8455634| 0.8995808|
|Ensemble                     | 0.8520111| 0.9045922|
```

Results

After trying 5 different models of machine learning, we obtained different values for both the accuracy F1 score that varied between 1 model and the other. Moreover, the highest value for accuracy and F1 score were obtained using the ensemble model having a value of 0.8520111 and 0.9045922 respectively. All the obtained results from accuracy and F1 score across the 5 models are found in the table shown below

Model	Accuracy	F1 Score
Logistic Regression	0.8480196	0.9025015
Linear Discriminant Analysis	0.8369665	0.8959028
Decision Tree	0.8308259	0.8947067
Random Forest	0.8455634	0.8995808
Ensemble	0.8520111	0.9045922

Conclusion

In order to predict whether an individual has yearly income of over $50K per year, we took into consideration several machine learning models including Logistic Regression, Linear Discriminant Analysis, Decision Tree, Random Forest and finally an Ensemble of the previous models. The performance of each model was based on 2 metrics which are accuracy and the F1 score. The performance varied among the models and the Logistic Regression was achieving the highest accuracy and F1 score of 0.8480196 and 0.9025015 respectively. These were the highest among the other models until the Ensemble model was considered which increased both accuracy and the F1 score to reach 0.8520111 and 0.9045922 respectively. Additional machine learning algorithms could have been considered and might have resulted in increases in both accuracy and F1 score but limitations such as computer power and ability were an obstacle for running such algorithms and models in addition to considering only 7 trees in as a parameter in the random forest model.