Classification of 1994 Census Income Data

Problem Statement: To build a model that will predict if the income of any individual in the US is greater than or less than USD 50,000 based on the data available about that individual.

Data Set Description: This Census Income dataset was collected by Barry Becker in 1994 and given to the site *http://archive.ics.uci.edu/ml/datasets/Census+Income*. This data set will help you understand how the income of a person varies depending on various factors such as the education background, occupation, marital status, geography, age, number of working hours/week, etc.

Here's a list of the independent or predictor variables used to predict whether an individual earns more than USD 50,000 or not:

- Age
- Work-class
- Final-weight
- Education
- Education-num (Number of years of education)
- Marital-status
- Occupation
- Relationship
- Race
- Sex
- Capital-gain
- Capital-loss
- Hours-per-week
- Native-country

The dependent variable is the "income-level" that represents the level of income. This is a categorical variable and thus it can only take two values:

1. <=50k
2. >=50k

Now that we've defined our objective and collected the data, it is time to start with the analysis.

Step 1: Import the data

#Downloading train and test data

```
>trainFile = "adult.data"; testFile = "adult.test"

>if (!file.exists (trainFile))

 +download.file (url = "<a href="http://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data">http://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.data</a>",

        destfile = trainFile)

>if (!file.exists (testFile))

 +download.file (url = "<ahref="http://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.test">http://archive.ics.uci.edu/ml/machine-learning-databases/adult/adult.test</a>",

        destfile = testFile)
```

Here our predictor variables are unlabelled. Therefore, variable names are assigned to each predictor variable so that the data more are readable, and get rid unnecessary white spaces.

```
>#Assigning column names

>colNames = c ("age", "workclass", "fnlwgt", "education",

        "educationnum", "maritalstatus", "occupation",

        "relationship", "race", "sex", "capitalgain",

        "capitalloss", "hoursperweek", "nativecountry",

        "incomelevel")


>#Reading training data

>training = read.table (trainFile, header = FALSE, sep = ",",

            strip.white = TRUE, col.names = colNames,

            na.strings = "?", stringsAsFactors = TRUE)

>#Display structure of the data

>str (training)

>#Removing NAs

>TrainSet = training [!is.na (training$workclass) & !is.na (training$occupation), ]
```

```
>TrainSet = TrainSet [!is.na (TrainSet$nativecountry), ]

>#Removing unnecessary variables


>TrainSet$fnlwgt = NULL
```

So, after importing and transforming the data into a readable format, we'll move to the next crucial step in Data Processing, which is Data Cleaning.

Step 2: Data Cleaning

The data cleaning stage is considered to be one of the most time-consuming tasks in Data Science. This stage includes removing NA values, getting rid of redundant variables and any inconsistencies in the data.

We'll begin the data cleaning by checking if our data observations have any missing values:

```
>table (complete.cases (training))

>FALSE TRUE

>2399 30162
```

The above indicates that 2399 sample cases have NA values. In order to fix this, let's look at the summary of all our variables and analyze which variables have the greatest number of null values. The reason why we must get rid of NA values is that they lead to wrongful predictions and hence decrease the accuracy of our model.

```
> summary  (training [!complete.cases(training),])

age             workclass      fnlwgt          education    educationnum

Min.   :17.00   Private       : 410   Min.   : 12285   HS-grad     :661   Min.   : 1.00

1st Qu.:22.00   Self-emp-inc   :  42   1st Qu.:121804   Some-college:613   1st Qu.: 9.00

Median :36.00   Self-emp-not-inc:  42   Median :177906   Bachelors   :311   Median :10.00

Mean   :40.39   Local-gov      :  26   Mean   :189584   11th        :127   Mean   : 9.57

3rd Qu.:58.00   State-gov      :  19   3rd Qu.:232669   10th        :113   3rd Qu.:11.00

Max.   :90.00   (Other)        :  24   Max.   :981628   Masters     : 96   Max.   :16.00

NA's           :1836                  (Other)     :478
```

```
 maritalstatus          occupation          relationship              race
 Divorced           :229   Prof-specialty : 102   Husband      :730   Amer-Indian-Eskimo:  25
 Married-AF-spouse    :  2   Other-service  :  83   Not-in-family :579   Asian-Pac-Islander: 144
 Married-civ-spouse   :911   Exec-managerial:  74   Other-relative: 92   Black           : 307
 Married-spouse-absent: 48   Craft-repair   :  69   Own-child     :602   Other           :  40
 Never-married        :957   Sales          :  66   Unmarried     :234   White           :1883
 Separated            : 86   (Other)        : 162   Wife          :162
 Widowed              :166   NA's           :1843
sex         capitalgain      capitalloss      hoursperweek       nativecountry
Female: 989   Min.   :    0.0   Min.   :   0.00   Min.   : 1.00   United-States
Median :    0.0   Median :   0.00   Median :40.00   Canada
Mean   :  897.1   Mean   :  73.87   Mean   :34.23   Philippines
3rd Qu.:    0.0   3rd Qu.:   0.00   3rd Qu.:40.00   Germany
Max.   :99999.0   Max.   :4356.00   Max.   :99.00   (Other)
NA's         : 583
```

From the above summary, it is observed that three variables have a good amount of NA values:

1. Workclass – 1836
2. Occupation – 1843
3. Nativecountry – 583

These three variables must be cleaned since they are significant variables for predicting the income level of an individual.

```
#Removing NAs
TrainSet = training [!is.na (training$workclass) & !is.na (training$occupation), ]
TrainSet = TrainSet [!is.na (TrainSet$nativecountry), ]
```

Once we've gotten rid of the NA values, our next step is to get rid of any unnecessary variable that isn't essential for predicting our outcome. It is important to get rid of such variables because they only increase the complexity of the model without improving its efficiency.

One such variable is the 'fnlwgt' variable, which denotes the population totals derived from CPS by calculating "weighted tallies" of any particular socio-economic characteristics of the population.

This variable is removed from our data set since it does not help to predict our resultant variable:

```
#Removing unnecessary variables

TrainSet$fnlwgt = NULL
```

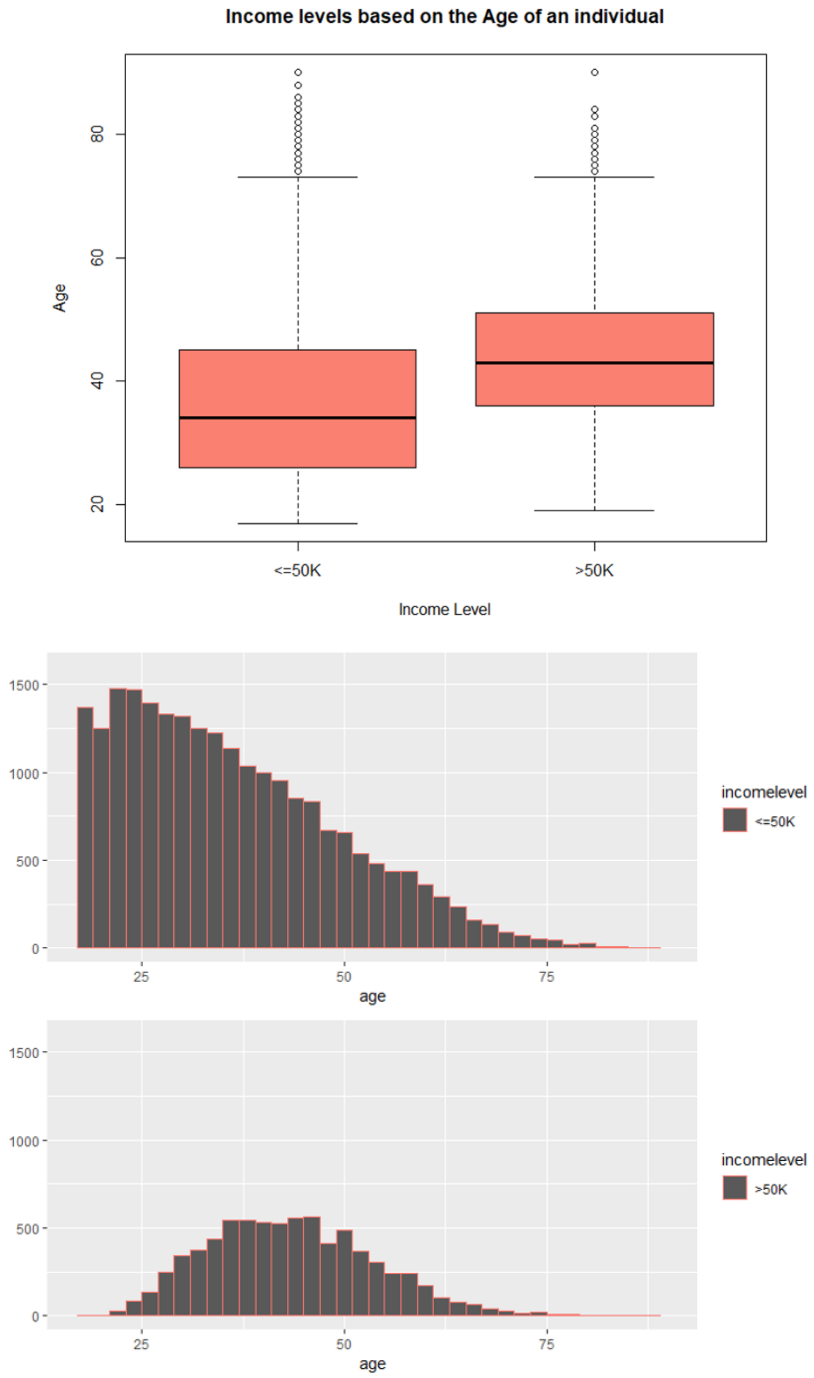
So that was all for Data Cleaning, our next step is Data Exploration.

Step 3: Data Exploration

Data Exploration involves analyzing each feature variable to check if the variables are significant for building the model.

Exploring the age variable

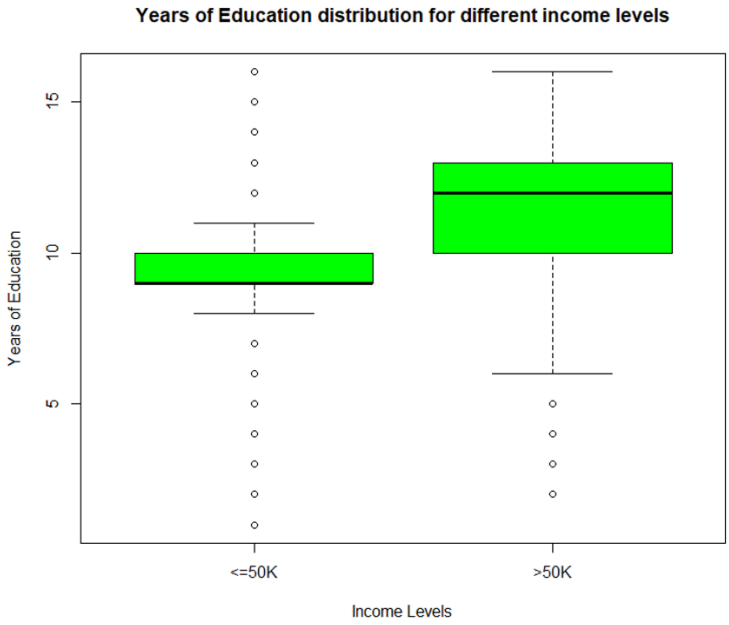
```
#Data Exploration

#Exploring the age variable

summary (TrainSet$age)

#Boxplot for age variable

boxplot (age ~ incomelevel, data = TrainSet,

      main = "Income levels based on the Age of an individual",

      xlab = "Income Level", ylab = "Age", col = "salmon")

#Histogram for age variable

incomeBelow50K = (TrainSet$incomelevel == "<=50K")

xlimit = c (min (TrainSet$age), max (TrainSet$age))

ylimit = c (0, 1600)

hist1 = qplot (age, data = TrainSet[incomeBelow50K,], margins = TRUE,

          binwidth = 2, xlim = xlimit, ylim = ylimit, colour = incomelevel)


hist2 = qplot (age, data = TrainSet[!incomeBelow50K,], margins = TRUE,

          binwidth = 2, xlim = xlimit, ylim = ylimit, colour = incomelevel)

grid.arrange (hist1, hist2, nrow = 2)
```



The above illustrations show that the age variable is varying with the level of income and hence it is a strong predictor variable.

Exploring the 'educationnum' variable

This variable denotes the number of years of education of an individual. Let's see how the 'educationnum' variable varies with respect to the income levels.



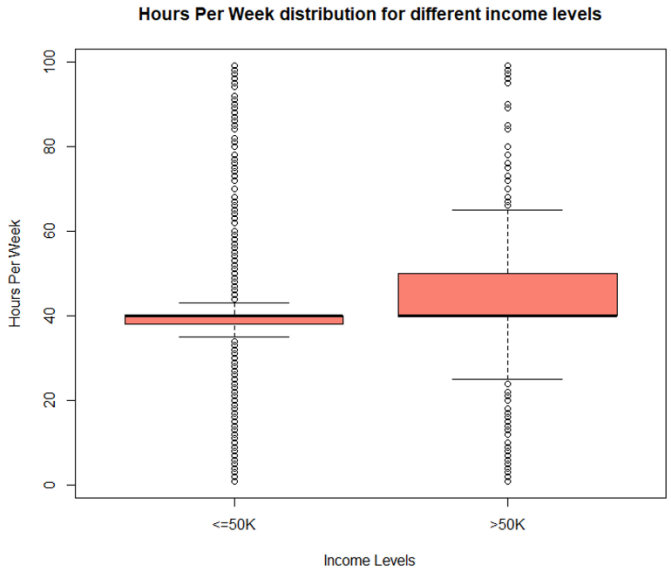
The above illustration depicts that the 'educationnum' variable varies for income levels <=50k and >50k, thus proving that it is a significant variable for predicting the outcome.

Exploring capital-gain and capital-loss variable

After studying the summary of the capital-gain and capital-loss variable for each income level, it is clear that their means vary significantly, thus indicating that they are suitable variables for predicting the income level of an individual.

Exploring hours/week variable

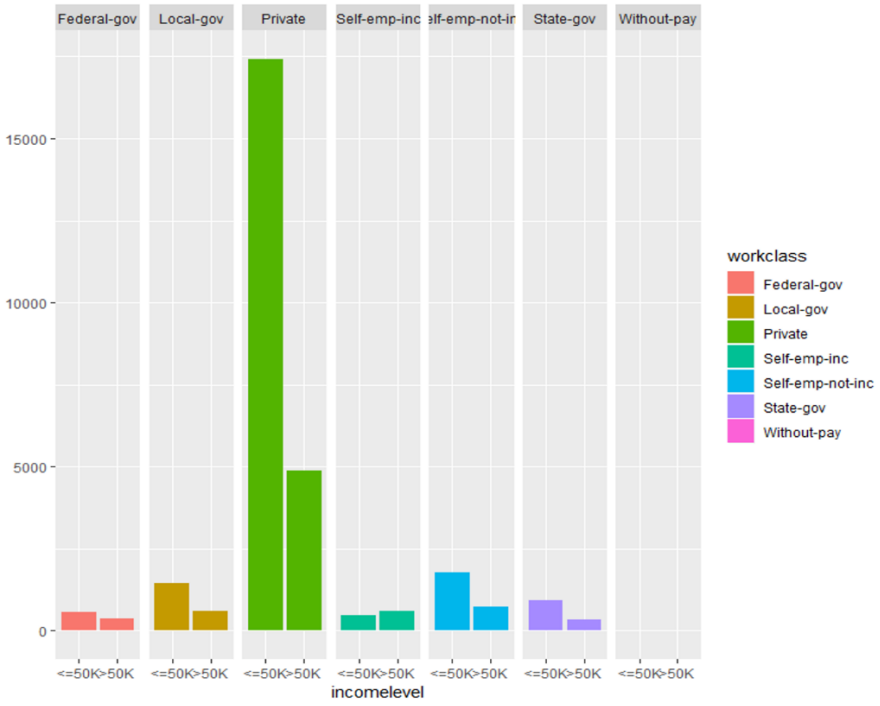
Similarly, the 'hoursperweek' variable is evaluated to check if it is a significant predictor variable.

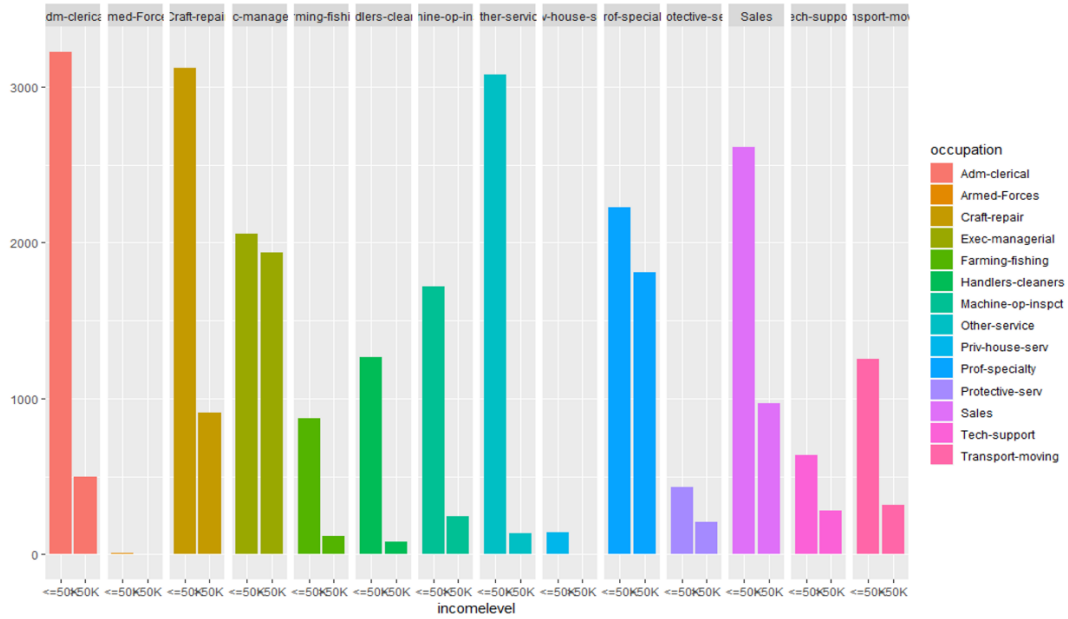


The boxplot shows a clear variation for different income levels which makes it an important variable for predicting the outcome.

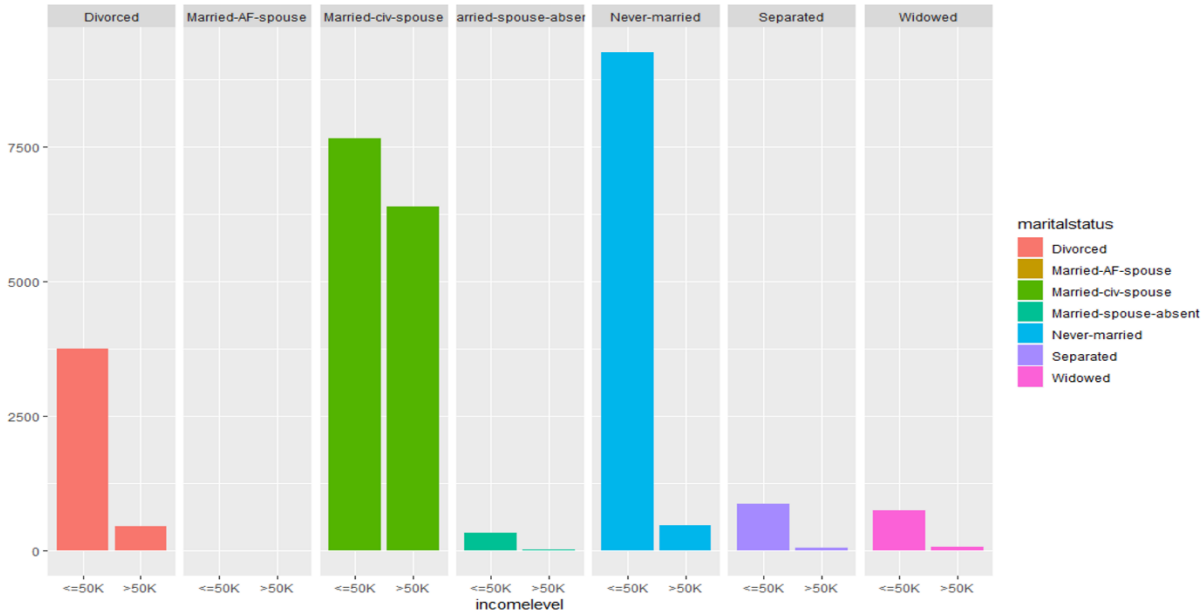
Similarly, we'll be evaluating categorical variables as well. In the below section I've created qplots for each variable and after evaluating the plots, it is clear that these variables are essential for predicting the income level of an individual.

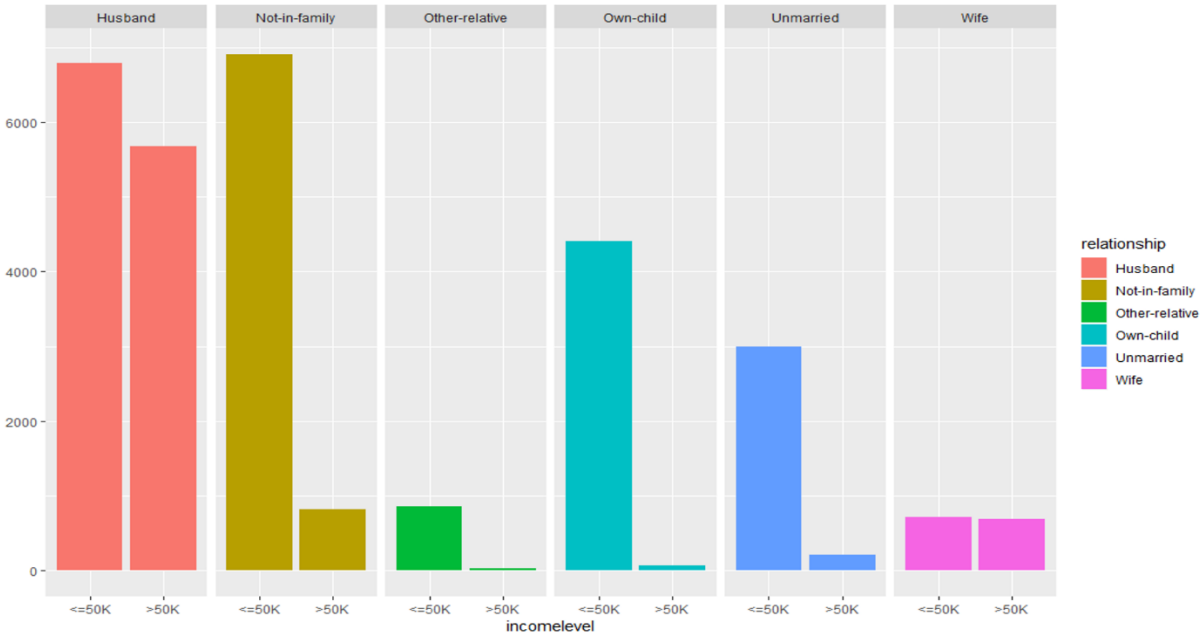
Exploring work-class variable and occupation





Exploring maritalstatus and relationship





All these graphs show that these set of predictor variables are significant for building our predictive model.

Step 4: Building A Model

So, after evaluating all our predictor variables, it is finally time to perform Predictive analytics. In this stage, we'll build a predictive model that will predict whether an individual earns above USD 50,000 or not based on the predictor variables that we evaluated in the previous section.

To build this model I've made use of the boosting algorithm since we have to classify an individual into either of the two classes, i.e:

1. Income level <= USD 50,000
2. Income level > USD 50,000

```
#Building the model

set.seed (32323)


trCtrl = trainControl(method = "cv", number = 10)


boostFit = train (incomelevel ~ age + workclass + education + educationnum +
                  maritalstatus + occupation + relationship +
                  race + capitalgain + capitalloss + hoursperweek +
                  nativecountry, trControl = trCtrl,
                 method = "gbm", data = TrainSet, verbose = FALSE)
```

Since we're using an ensemble classification algorithm, I've also implemented the Cross-Validation technique to prevent overfitting of the model.

Step 5: Checking the accuracy of the model

To evaluate the accuracy of the model, we're going to use a confusion matrix:

```
#Checking the accuracy of the model

> confusionMatrix (TrainSet$incomelevel, predict (boostFit, TrainSet))

Confusion Matrix and Statistics


Reference

Prediction <=50K >50K

<=50K 21404 1250 >50K 2927 4581

Accuracy : 0.8615

95% CI : (0.8576, 0.8654)

No Information Rate : 0.8067

P-Value [Acc > NIR] : < 2.2e-16
```

Kappa : 0.5998

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

Sensitivity : 0.8797

Specificity : 0.7856

Pos Pred Value : 0.9448

Neg Pred Value : 0.6101

Prevalence : 0.8067

Detection Rate : 0.7096

Detection Prevalence : 0.7511

Balanced Accuracy : 0.8327

'Positive' Class : <=50K

The output shows that our model calculates the income level of an individual with an accuracy of approximately 86%, which is a good number.