INCOME CLASSIFICATION

Aravindh Boya

1. INTRODUCTION

The goal of this project to predict if the persons income will be over 50k$ a year or under 50k$. This data was collected by Barry Becker from the census database of the year 1994. The dataset is available here https://www.kaggle.com/lodetomasi1995/income-classification. This is a medium sized dataset with 32561 observations and 15 attributes. Of these 15 attributes 9 are categorical and 6 are numerical.

The dataset has various information about the person like the age, sex, gender, occupation, education level, hours per week, race, native country etc, these are used to predict if the person makes above or under 50k a year.

The dataset was found at kaggle.com, a data science and machine learning competitions platform. The dataset has reasonably clean records with minimal data pre processing needed.

Nature of data:

- Data Set Characteristics: Multivariate
- Attribute Characteristics: Categorical and Numerical
- Number of Records: 32561
- Number of Attributes: 15

Attributes Info:

Attribute	Information
income	The income of the person.
age	continuous.
workclass	Private, Self-emp-not-inc, Self-emp-inc, Federal-gov, Local-gov, State-gov, Without-pay, Never-worked.
fnlwgt	continuous.
education	Bachelors, Some-college, 11th, HS-grad, Prof-school, Assoc-acdm, Assoc-voc, 9th, 7th-8th, 12th, Masters, 1st-4th, 10th, Doctorate, 5th-6th, Preschool.
education-num	continuous.
marital-status	Married-civ-spouse, Divorced, Never-married, Separated, Widowed, Married-spouse-absent, Married-AF-spouse.

occupation	Tech-support, Craft-repair, Other-service, Sales, Exec-managerial, Prof-specialty, Handlers-cleaners, Machine-op-inspct, Adm-clerical, Farming-fishing, Transport-moving, Priv-house-serv, Protective-serv, Armed-Forces.
relationship	Wife, Own-child, Husband, Not-in-family, Other-relative, Unmarried.
race	White, Asian-Pac-Islander, Amer-Indian-Eskimo, Other, Black.
sex	Female, Male.
capital-gain	continuous.
capital-loss	continuous.
hours-per-week	continuous.
native-country	United-States, Cambodia, England, Puerto-Rico, Canada, Germany, Outlying-US(Guam-USVI-etc), India, Japan, Greece, South, China, Cuba, Iran, Honduras, Philippines, Italy, Poland, Jamaica, Vietnam, Mexico, Portugal, Ireland, France, Dominican-Republic, Laos, Ecuador, Taiwan, Haiti, Columbia, Hungary, Guatemala, Nicaragua, Scotland, Thailand, Yugoslavia, El-Salvador, Trinadad&Tobago, Peru, Hong, Holand-Netherlands.

```
library(caret)
```

```
## Loading required package: lattice
```

```
## Loading required package: ggplot2
```

```
library(smotefamily)
```

```
## Warning: package 'smotefamily' was built under R version 4.0.2
```

```
library("RColorBrewer")
library(dplyr)
```

```
##
## Attaching package: 'dplyr'
```

```
## The following objects are masked from 'package:stats':
##
##     filter, lag
```

```
## The following objects are masked from 'package:base':
##
##     intersect, setdiff, setequal, union
```

```
library(class)
```

```
df = read.csv('C:/Users/aravi/OneDrive/Documents/special_topics/Datasets/income-evaluation.xls')
```

```
cat("The number of Null values in dataset is", sum(is.na(df)))
```

```
## The number of Null values in dataset is 0
```

```
summary(df)
```

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

```
df$income = as.factor(df$income)
df$income = as.numeric(df$income)

df$income = df$income - 1

table(df$income)
```

```
##
##     0     1
## 24720  7841
```

2. METHODS

The dataset is reasonably clean so I really did not need to do much data pre-processing. This is a classification task where the goal is to predict if the income is over 50K or under. I ran 3 machine learning models namely

- Logistic Regression
- Cross Validation
- Linear Discriminant Analysis
- K-Nearest Neighbors

```
###### set seed to ensure you always have same random numbers generated
set.seed(22)

train_ind = sample(seq_len(nrow(df)),size = 0.7 * nrow(df))
###### creates the training dataset with row numbers stored in train_ind
train =df[train_ind,]
test=df[-train_ind,]

table(train$income)

##
##     0     1
## 17253  5539
```

- **Split the data:** To address our problem, Above I have divided the data set into two samples, Training(70%) and Testing(30%). In our case we use for example 200 observations in the train data, 86 in the test data.

```
chisq.test(df$income, df$age)
chisq.test(df$income, df$workclass)
chisq.test(df$income, df$fnlwgt)
chisq.test(df$income, df$education)
chisq.test(df$income, df$education.num)
chisq.test(df$income, df$marital.status)
chisq.test(df$income, df$occupation)
chisq.test(df$income, df$relationship)
chisq.test(df$income, df$race)
chisq.test(df$income, df$sex)
chisq.test(df$income, df$capital.gain)
chisq.test(df$income, df$capital.loss)
chisq.test(df$income, df$native.country)
chisq.test(df$income, df$hours.per.week)
```

- **Dependent Variables:**

By using chi-squared test we came to know that dependent variables of income are all the other variables in the dataset except fnlwgt.The fnlwgt which is the final weight determined by the Census Organization is of no use in any of the analysis that we are doing henceforth and is removed. The educationnum if a repetitive variable which recodes the

categorical variable education as a numeric variable but will be used in the analysis for decision trees, hence is not being removed.

- **Analysis:**

All the above classifiers are modeling the probability that an individual makes more than 50K annually. In another word, a response closer to 1 indicates higher chance of making over 50K while a response closer to 0 indicates a higher chance of making less than 50K. Thus, a threshold of 0.5 is used to determine whether an individual is predicted to make more than 50K annually or not. A confusion matrix is presented to evaluate how well the model predicts income.

Classification Techniques

Logistic Regression

A logistic regression model will first be developed to compare the performance of various machine learning algorithms and is the most common classification model, though it has regression in its name. It is a statistical model which uses the logistic(sigmoid) function to model the binary dependent variable. It is used to describe the data and to understand the relationships between one dependent binary variable to the one or more independent variables.

```
logit_model = glm(
  as.factor(income) ~ workclass + education + marital.status  + occupation +
relationship + race,
  data = train,
  family = "binomial"
)



logit_pred = predict(logit_model,newdata = test, type = "response")

logit_pred = ifelse(logit_pred > 0.5, 1, 0)
```

Cross Validation

Cross-validation refers to a set of methods for measuring the performance of a given predictive model on new test data sets.The basic idea, behind cross-validation techniques, consists of dividing the data into two sets: The training set, used to train (i.e. build) the model; and the testing set (or validation set), used to test (i.e. validate) the model by estimating the prediction error. Cross-validation is also known as a resampling method because it involves fitting the same statistical method multiple times using different subsets of the data. The most commonly used statistical metrics for measuring the performance of a regression model in predicting the outcome of new test data.though it is said as regression model but can used for the classification as well.

```
train_control_cv = trainControl(method = "cv", number = 3)

logit_model_cv = train(
  as.factor(income) ~  workclass + education + marital.status  + occupation +
relationship + race,
  data = train,
  trControl = train_control_cv,
  method = "glm",
  family = binomial
)

logit_pred_cv = predict(logit_model_cv, newdata = test)
```

Linear Discriminant Analysis

Linear Discriminant Analysis is a dimensionality reduction and classification technique. It consists of statistical properties of your data like the mean and variance of each class. A discriminant rule tries to separate the data into N regions which are disjoint, N being the number of classes. LDA is often used as benchmark model and it does address some of the limitations of the logistic regression model.

```
library(MASS)

lda_model <- lda(income ~  workclass + education + marital.status  + occupati
on + relationship + race, data = train)

lda_predictions = predict(lda_model, newdata = test)

lda_predictions = lda_predictions$class
```

K-Nearest Neighbors

K-NN is a memory based model used for both classification and regression. It assumes that similar things exist in the close proximity. It is a memory based algorithm because it stores all the training data. The new data is classified based on the similarity measure mainly euclidean distance. The value of K is choose arbitarily but the general rule is to take the root of n, and it's always preferred to take the odd number.

```
library(class)

dummy_data <- dummyVars(" ~ .", data = train)

train_oh <- data.frame(predict(dummy_data, newdata = train))

test_oh <- data.frame(predict(dummy_data, newdata = test))

train_cols <- names(train_oh)
test_cols <- names(test_oh)
com_cols <- intersect(train_cols, test_cols)
```

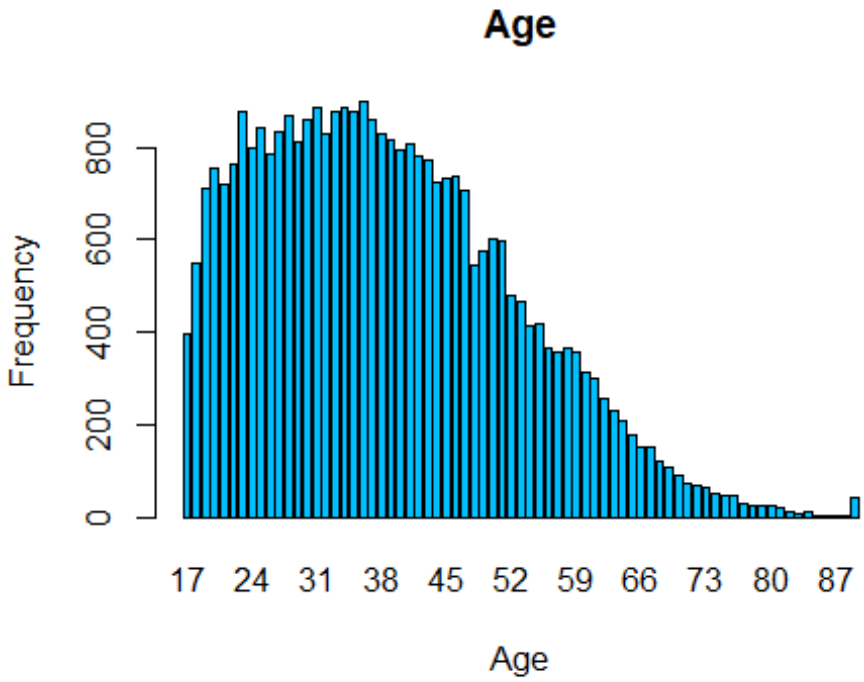
```
train_knn = train_oh[com_cols]
test_knn = test_oh[com_cols]

cols_exc <- names(train_knn) %in% c("income")

train_knn <- train_knn[!cols_exc]

test_knn <- test_knn[!cols_exc]
train_labels <- train_oh$income
test_labels <- test_oh$income

knn_preds = knn(train_knn,test_knn,cl=as.factor(train_labels),k=9)
```

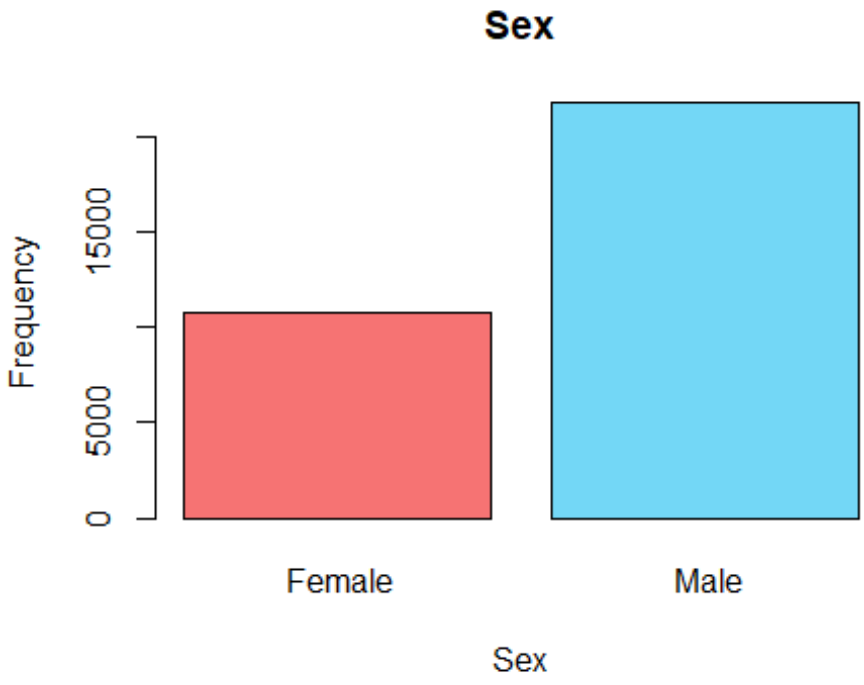
Data Visualizations

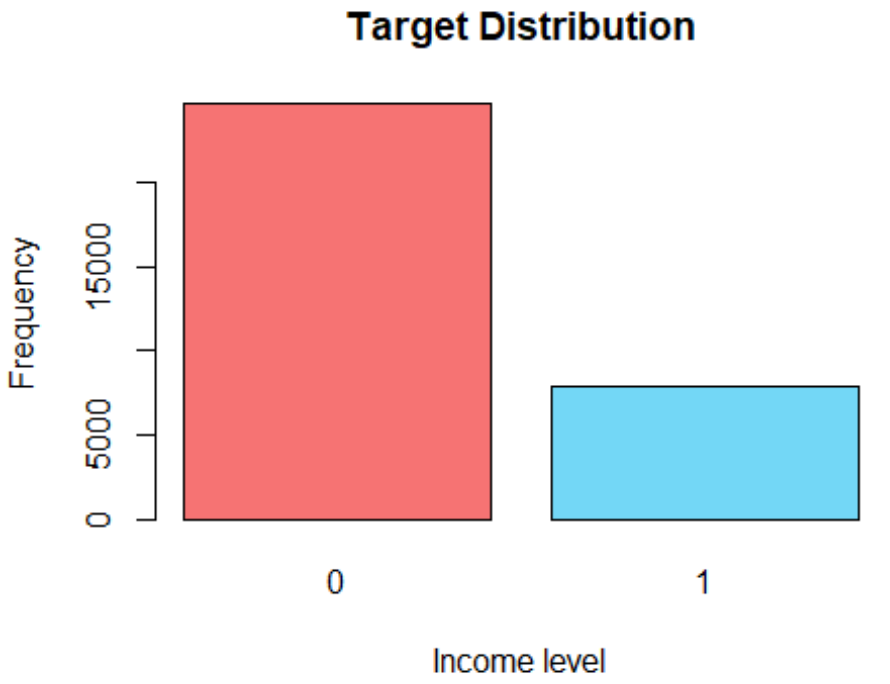
Plots for the Data set

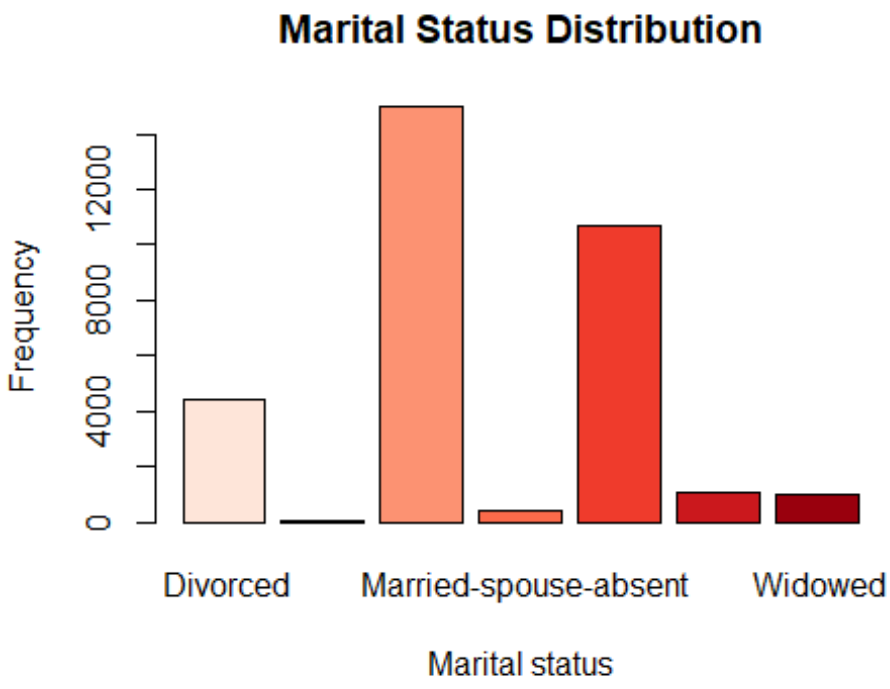
```
barplot(table(df$age), xlab = 'Age', ylab = 'Frequency', main = 'Age', col= "
deepskyblue1")

barplot(table(as.factor(df$sex)), xlab = 'Sex', ylab = 'Frequency', main = 'S
ex', col = c("#F67373", "#73D7F6"))

barplot(table(as.factor(df$income)), xlab = 'Income level', ylab = 'Frequency
', main = 'Target Distribution', col = c("#F67373", "#73D7F6"))

barplot(table(as.factor(df$marital.status)), xlab = 'Marital status', ylab =
'Frequency', main = 'Marital Status Distribution', col =  brewer.pal(n = 7, n
ame = "Reds"))

barplot(table(as.factor(df$race)), xlab = 'Race', ylab = 'Frequency', main =
'Race Distribution', col =  brewer.pal(n = 5, name = "PuBu"))

barplot(table(as.factor(df$education)), xlab = 'Education', ylab = 'Frequency
', main = 'Education Distribution', col =  brewer.pal(n = 7, name = "Set3"))

hist(df$capital.gain, xlab = 'Capital gain', ylab = 'Frequency', main = 'Capi
tal Gain histogram', col = 'deepskyblue')

hist(df$capital.loss, xlab = 'Capital loss', ylab = 'Frequency', main = 'Capi
tal Loss histogram', col = 'deepskyblue3')

barplot(table(as.factor(df$relationship)), xlab = 'relationship', ylab = 'Fre
quency', main = 'relationship Distribution', col =  brewer.pal(n = 7, name =
"Dark2"))

hist(df$hours.per.week, xlab = 'Hours per week', ylab = 'Frequency', main = '
Hours per week histogram', col = 'tomato2')
```

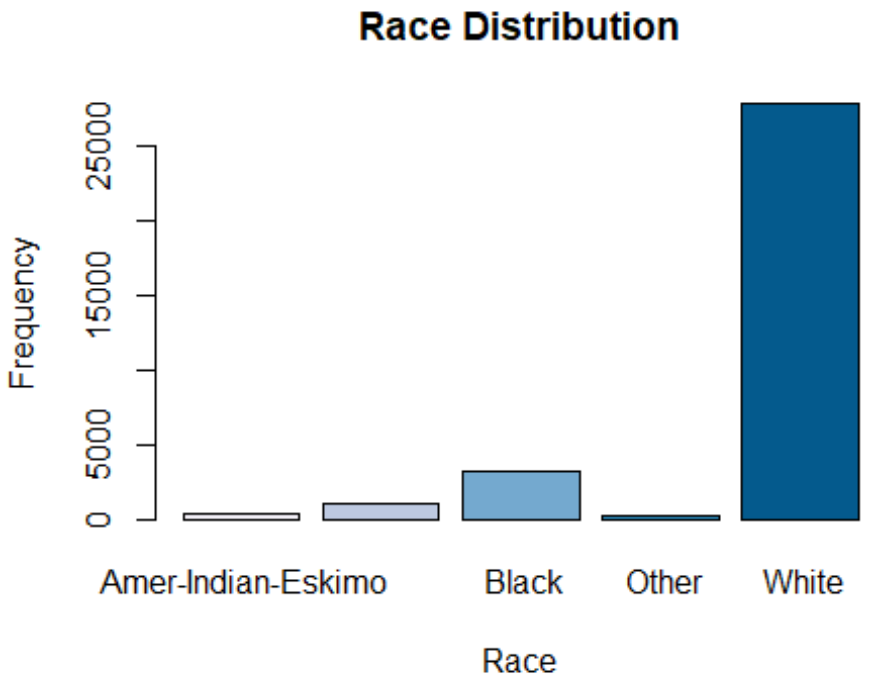
```r
country_names = c('USA','Non-USA')
country_freq = c(dim(df %>%filter(as.integer(as.factor(native.country)) %in%
c(40)))[1], dim(df %>%filter(!as.integer(as.factor(native.country)) %in% c(40
)))[1])

barplot(country_freq, main = "Country Frequency", xlab = "Country", ylab = 'F
requency', names.arg = country_names, col = c("#F63C6E", "#D0F989"))
```

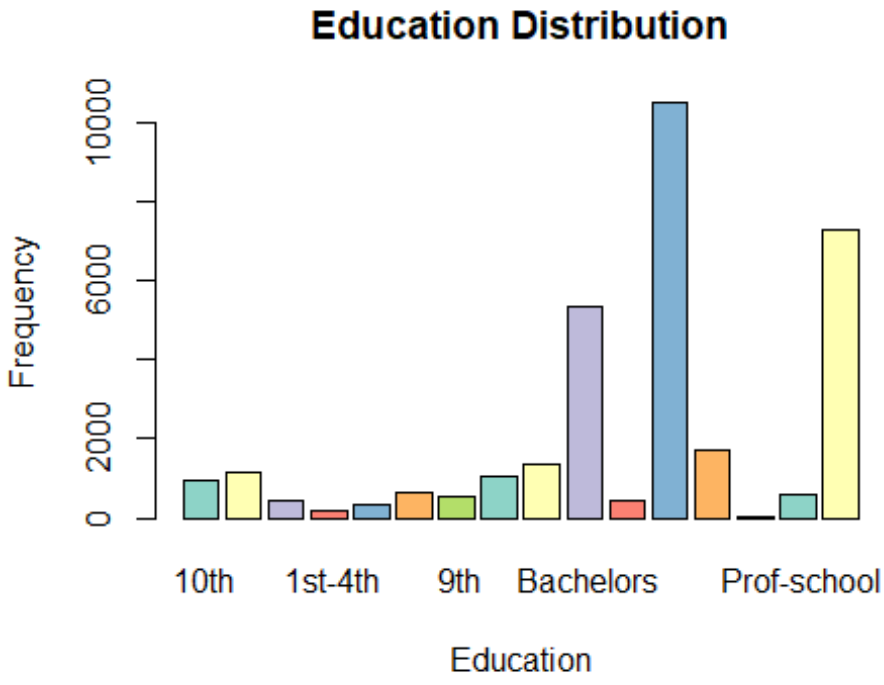

Age
Frequency
0
200
400
600
800
17
24
31
38
45
52
59
66
73
80
87
Age

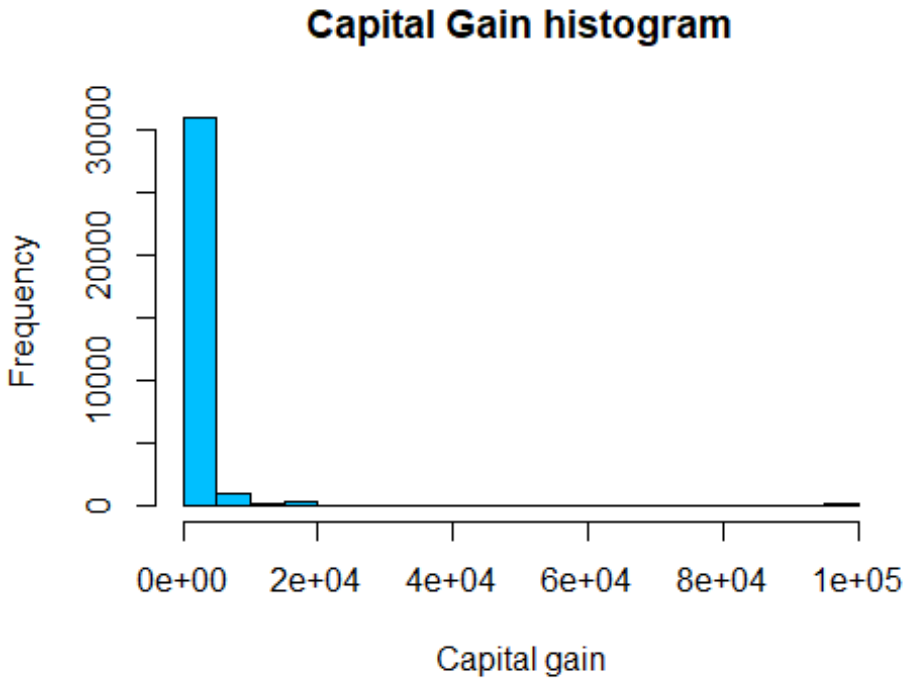

Sex
Frequency
0
5000
15000
Female
Male
Sex

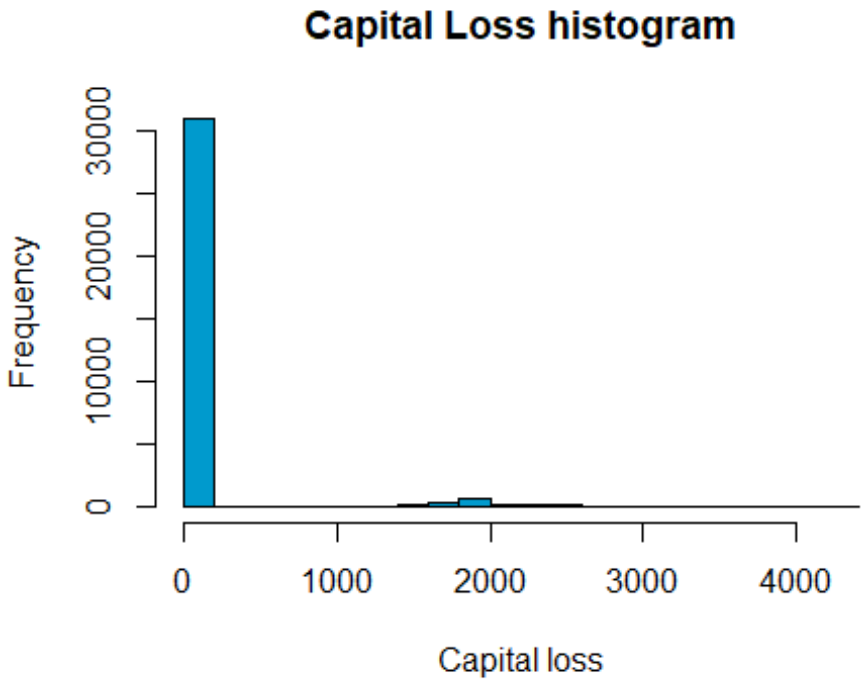

Target Distribution
Frequency
0
5000
15000
0
1
Income level

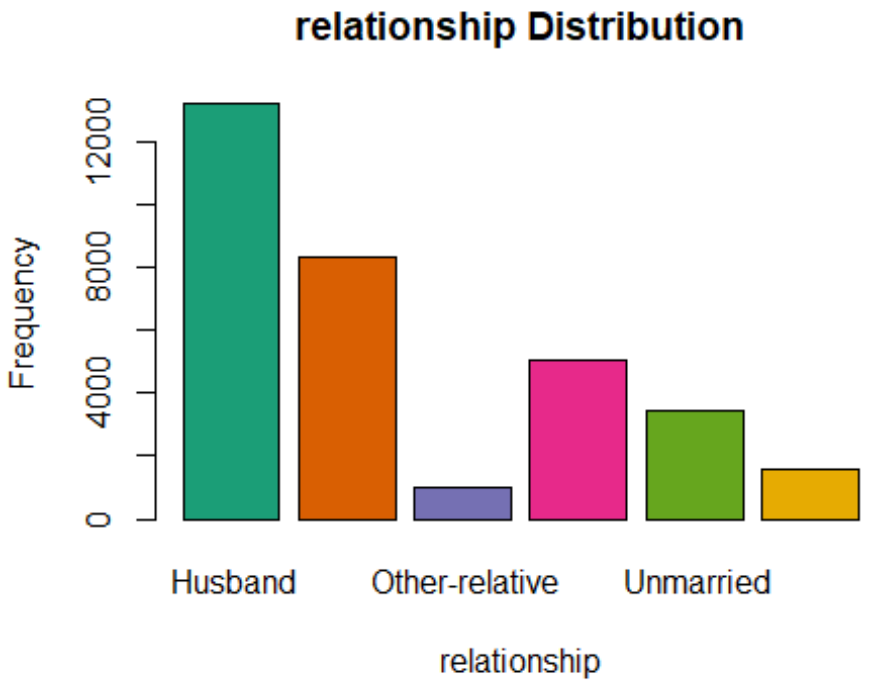

Marital Status Distribution
Frequency
0
4000
8000
12000
Divorced
Married-spouse-absent
Widowed
Marital status

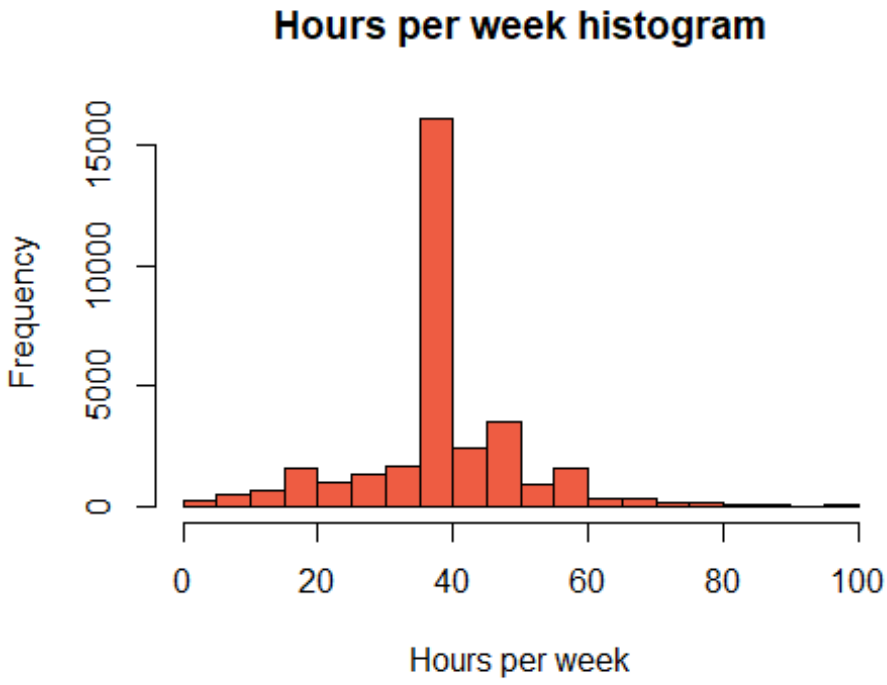

Race Distribution
Frequency
0
5000
15000
25000
Amer-Indian-Eskimo
Black
Other
White
Race

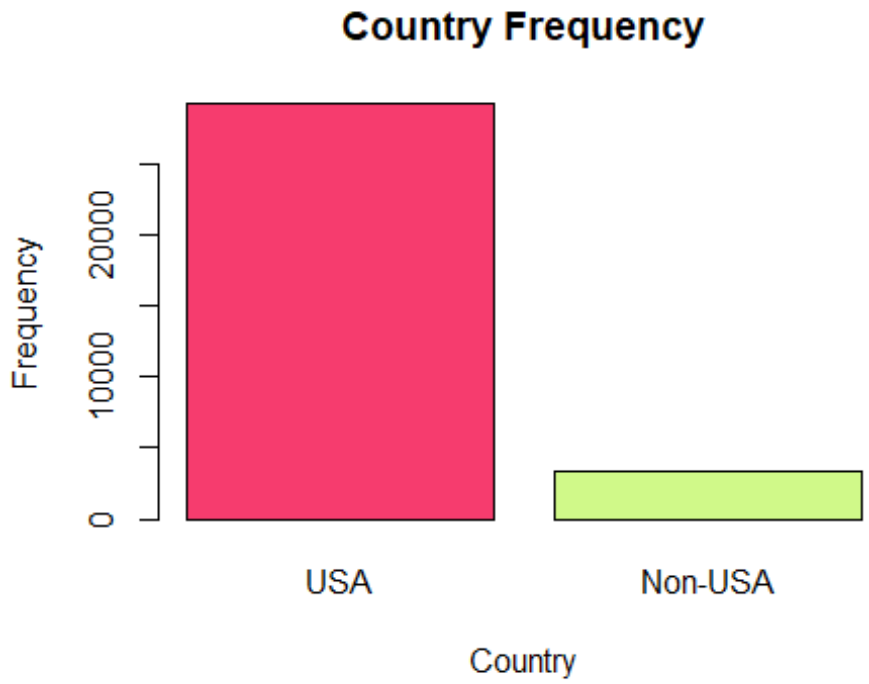

Education Distribution
Frequency
0
2000
6000
10000
10th
1st-4th
9th
Bachelors
Prof-school
Education


Capital Gain histogram
Frequency
0
10000
20000
30000
0e+00
2e+04
4e+04
6e+04
8e+04
1e+05
Capital gain


Capital Loss histogram
Frequency
0
10000
20000
30000
0
1000
2000
3000
4000
Capital loss


relationship Distribution
Frequency
0
4000
8000
12000
Husband
Other-relative
Unmarried
relationship


Hours per week histogram
Frequency
0
5000
10000
15000
0
20
40
60
80
100
Hours per week



3.RESULTS

Accuracy

Accuracy is one of the most common metrics in measuring the classification models performance. It is defined as the ratio of number of correct predictions over the total predictions made. So, obviously the more the accuracy is the better the model. We should always evaluate our models performance on the test data but on the train data since the model is built on the train data it usually performs better on the data it has seen, but test data is something which the model has not seen so we can trust the test data's metrics.

Accuracy = Number of correct predictions / Total number of predictions made

Confusion Matrix:

We will use confusion matrix as our second metric as just in case if the accuracies of two models are relatively same we look at confusion matrix to identify the false negatives and false positives. Here in our case the primary goal is to predict if the person makes above 50k false positives and false negatives helps us in deciding the best model.

Confusion matrix as the name says can be really confusing to understand, it is the summary of the predictions where the correct and incorrect classifications are summarized. It is has 4 elements

- True positive - Observation is positive, and is predicted to be positive.

- True Negative - Observation is negative, and is predicted to be negative.
- False Positve - Observation is negative, but is predicted positive.
- False Negative - Observation is positive, but is predicted negative.

Here the goal for our project is to show that most number false positive through confusion matrix that means people has the income more than 50K$.

Logistic Regression Analysis

The following are the results of the logistic regression. As you can see we achieved an accuracy of **83.13%** using the logistic regression without any hyper parameter tuning.

As we look at the results it's shows that accuracy is better than the 'NIR' (No Information rate).So, there is no class imbalance in our model.The '95% CI' (confidence interval) for this model for both 0 and 1 is greater than **82%** and the p-value is in negative .So we can reject the null hypothesis. Our model has the "Pos Pred Value" more than **86%**. So, we can say that this model fits best to the dataset.

```
logit_reg_cm = confusionMatrix(as.factor(logit_pred), as.factor(test$income))
logit_reg_cm
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 6910 1091
##          1  557 1211
##
##                Accuracy : 0.8313
##                  95% CI : (0.8237, 0.8387)
##     No Information Rate : 0.7644
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4908
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9254
##             Specificity : 0.5261
##          Pos Pred Value : 0.8636
##          Neg Pred Value : 0.6850
##              Prevalence : 0.7644
##          Detection Rate : 0.7073
##    Detection Prevalence : 0.8190
##       Balanced Accuracy : 0.7257
##
##        'Positive' Class : 0
##
```

```
logit_accuracy <- logit_reg_cm$overall[1]

cat("The logistic Regression accuracy is", logit_accuracy)

## The logistic Regression accuracy is 0.8313031
```

Cross validation

The following are the results of the cross validation on logistic regression. The accuracy using this model is **83.13%**

If we look at the results of this particular model the accuracy is same as the logistic regression.when it comes to the no -information rate) the classes are not imbalanced as classes are splitted into 70 and 30 each.Null hypothesis can be rejected. This model prediction rate for the positive value is 86%. So, this model also suits very well to the dataset.

```
logit_reg_cv_cm = confusionMatrix(as.factor(logit_pred_cv), as.factor(test$in
come))
logit_reg_cv_cm

## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 6910 1091
##          1  557 1211
##
##                Accuracy : 0.8313
##                  95% CI : (0.8237, 0.8387)
##     No Information Rate : 0.7644
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4908
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9254
##             Specificity : 0.5261
##          Pos Pred Value : 0.8636
##          Neg Pred Value : 0.6850
##              Prevalence : 0.7644
##          Detection Rate : 0.7073
##    Detection Prevalence : 0.8190
##       Balanced Accuracy : 0.7257
##
##        'Positive' Class : 0
##
```

```
logit_cv_accuracy <- logit_reg_cv_cm$overall[1]

cat("The Cross validated logistic Regression accuracy is", logit_cv_accuracy)

## The Cross validated logistic Regression accuracy is 0.8313031
```

Linear Discriminant Analysis

The accuracy using the LDA model is **82.83%**

The Accuracy of this model is little lower than the above two models but can say one of the suitable models for this type of dataset by looking at the output particular and they does not seems to be bad.It has good accuracy with decent confidence interval. No class imbalance and can reject the Null hypothesis.

```
lda_cm = confusionMatrix(as.factor(lda_predictions), as.factor(test$income))
lda_cm

## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 6889 1099
##          1  578 1203
##
##                Accuracy : 0.8283
##                  95% CI : (0.8207, 0.8358)
##     No Information Rate : 0.7644
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.483
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9226
##             Specificity : 0.5226
##          Pos Pred Value : 0.8624
##          Neg Pred Value : 0.6755
##              Prevalence : 0.7644
##          Detection Rate : 0.7052
##    Detection Prevalence : 0.8177
##       Balanced Accuracy : 0.7226
##
##        'Positive' Class : 0
##

lda_accuracy <- lda_cm$overall[1]

cat("The Linear Discriminant Analysis accuracy is", lda_accuracy)

## The Linear Discriminant Analysis accuracy is 0.8283345
```

KNN Analysis

The accuracy using KNN model is **79.35%**

The Accuracy of this model is little lower than the above two models but can say one of the suitable models for this type of dataset by looking at the output particular and they does not seems to be bad.It has good accuracy with decent confidence interval. No class imbalance and can reject the Null hypothesis. It has got the low balance accuracy level.

```
knn_cm = confusionMatrix(as.factor(knn_preds), as.factor(test_labels))
knn_cm
```

```
## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 7166 1718
##          1  301  584
##
##                Accuracy : 0.7933
##                  95% CI : (0.7852, 0.8013)
##     No Information Rate : 0.7644
##     P-Value [Acc > NIR] : 4.124e-12
##
##                   Kappa : 0.2711
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9597
##             Specificity : 0.2537
##          Pos Pred Value : 0.8066
##          Neg Pred Value : 0.6599
##              Prevalence : 0.7644
##          Detection Rate : 0.7335
##    Detection Prevalence : 0.9094
##       Balanced Accuracy : 0.6067
##
##        'Positive' Class : 0
##
```

```
knn_accuracy <- knn_cm$overall[1]

cat("The Knn accuracy is", knn_accuracy)
```

```
## The Knn accuracy is 0.7933258
```

To conclude, we are encouraged by our results. Our classifiers are extrapolating patterns from the data, and this shows promise to be successfully can predict income based on Census information. it is clear that **Logistic Regression** model performs well over the other models we tried and also since the accuracy difference is significant we can rule out

the confusion matrix metric. Though cross validation has got the same accuracy as logistic regression but the cross validation has more time complexity as compared to the logistic regression because it undergoes the validation of the data 3 times.

Challenges faced:

Through the course of the project, there were a few challenges that I faced. Since the dataset is reasonably clean I took the least amount of time in cleaning and preprocessing the data. In the modeling phase, the real challenge I faced is with the K-NN, as unlike other models like Logistic Regression and LDA, K-NN doesn't accept the categorical columns. So, I did one-hot encoding on the categorical columns like occupation, workclass, etc, and then fed it into the K-NN model. It was smooth running the other models as they are fine accepting the categorical columns I really didn't face any trouble while running those models.

4. DISCUSSION

Though their motivations differ, the logistic regression and LDA methods are closely connected. Both logistic regression and LDA produce linear decision boundaries. The only difference between the two approaches lies in the fact that ??0 and ??1(In logistic regression) are estimated using maximum likelihood, whereas c0 and c1 (In LDA) are computed using the estimated mean and variance from a normal distribution. This same connection between LDA and logistic regression also holds for multidimensional data with p > 1. Since logistic regression and LDA differ only in their fitting procedures, one might expect the two approaches to give similar results. KNN takes a completely different approach from the logistic regression, LDA classifiers.In order to make a prediction for an observation X = x, the K training observations that are closest to x are identified. Then X is assigned to the class to which the plurality of these observations belong. Hence KNN is a completely non-parametric approach: no assumptions are made about the shape of the decision boundary. Therefore, we can expect this approach to dominate LDA and logistic regression when the decision boundary is highly non-linear. On the other hand, KNN does not tell us which predictors are important.

Since **logistic regression analysis** model and **Cross Validated logistic regression analysis** gives the best accuracy i.e., **83.13%** , among all other models, we can conclude that, **logistic regression analysis** is the best fits to my data set and highly recommended.

Decision Tree

The one other method that might give the better performance is the decision tree and it's interpretability goes as follows A decision tree classifies by choosing a threshold on a feature and splits the data according to a 'splitting rule'. Since the features need to be numerical, we had to discard certain features and change how we represented others. For example, we could not convert native.country into numerical values since this would cause an implicit feature ranking skewing our results.However, education is a feature that can be converted into a numerical value, as a certain level of education can be higher or lower than others in rank. For this reason, we chose only to consider limited attributes (This is

represented as a binary feature with 1 being male and 0 being female). The tree is then built on the training set and used to predict the binary value of the label (whether or not an individual makes more that $50,000) on the test set.

```
library(rpart)

dt <- rpart(income ~  workclass + education + marital.status  + occupation +
relationship + race, data = train)

dt_preds = predict(dt, test)

dt_preds = round(dt_preds)
```

Decision Tree Analysis

The following are the results of the Decision Tree Analysis. The accuracy using this model is **81.94%**

```
dt_cm = confusionMatrix(as.factor(dt_preds), as.factor(test_labels))
dt_cm

## Confusion Matrix and Statistics
##
##           Reference
## Prediction    0    1
##          0 7077 1374
##          1  390  928
##
##                Accuracy : 0.8194
##                  95% CI : (0.8117, 0.827)
##     No Information Rate : 0.7644
##     P-Value [Acc > NIR] : < 2.2e-16
##
##                   Kappa : 0.4118
##
##  Mcnemar's Test P-Value : < 2.2e-16
##
##             Sensitivity : 0.9478
##             Specificity : 0.4031
##          Pos Pred Value : 0.8374
##          Neg Pred Value : 0.7041
##              Prevalence : 0.7644
##          Detection Rate : 0.7244
##    Detection Prevalence : 0.8651
##       Balanced Accuracy : 0.6754
##
##        'Positive' Class : 0
##
```

```r
dt_accuracy <- dt_cm$overall[1]

cat("The decision tree accuracy is", dt_accuracy)

## The decision tree accuracy is 0.8194288
```

5. REFERENCES CITED

- Ron Kohavi, “Scaling Up the Accuracy of Naive-Bayes Classifiers: A Decision-Tree Hybrid”, Proceedings of the Second International Conference on Knowledge Discovery and Data Mining, 1996
- Rich Caruana and Alexandru Niculescu-Mizil. An Empirical Evaluation of Supervised Learning for ROC Area. ROCAI. 2004.
- https://www.kaggle.com/jiashenliu/d/uciml/adult-census-income/who-can-earn-more-than-50k-peryear
- https://www.kaggle.com/jiashenliu/d/uciml/adult-census-income/who-can-earn-more-than-50k-peryear
- https://www.kaggle.com/bananuhbeatdown/d/uciml/adult-census-income/multiple-ml-techniquesand-analysis-of-dataset