# Predict Census Income

Kaggle Project

# Content

# Introduction

## Objective:

1. To predict a model to determine income level of people.
2. Income levels were determined at below 50,000 and above 50,000.

## Data/Methodology:

Data consists of 199523 values taken from a census survey. Data consists of 40+1 variables out of which 7 are continuous variables and 33 are nominal variables. The target variable is income which is binned at below 50,000 and above 50,000.

Methodology used was Machine Learning Pipeline: Data cleaning, data preprocessing, feature engineering and model training.

| | age | classworker | industrycode | occupationcode | education | wagehour | enrollineduclstweek | maritalstat | majorindcode | majoroccccode | race | origin | sex | memberoflabourunion | reasonofunemploy | fullorparttimeemploy |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| 0 | 58 | Self-employed-not incorporated | 4 | 34 | Some college but no degree | 0 | Not in universe | Divorced | Construction | Precision production craft & repair | White | All other | Male | Not in universe | Not in universe | Children or Armed Forces |
| 1 | 18 | Not in universe | 0 | 0 | 10th grade | 0 | High school | Never married | Not in universe or children | Not in universe | Asian or Pacific Islander | All other | Female | Not in universe | Not in universe | Not in labor force |
| 2 | 9 | Not in universe | 0 | 0 | Children | 0 | Not in universe | Never married | Not in universe or children | Not in universe | White | All other | Female | Not in universe | Not in universe | Children or Armed Forces |
| 3 | 10 | Not in universe | 0 | 0 | Children | 0 | Not in universe | Never married | Not in universe or children | Not in universe | White | All other | Female | Not in universe | Not in universe | Children or Armed Forces |
| 4 | 48 | Private | 40 | 10 | Some college but no degree | 1200 | Not in universe | Married-civilian spouse present | Entertainment | Professional specialty | Amer Indian Aleut or Eskimo | All other | Female | No | Not in universe | Full-time schedules |

*Fig 1: A snapshot of the data*

# Data Preparation

- Outliers were identified using IQR because there were some extreme values in the data. Around 10% of outliers were removed from data.

Dealing with Outliers

- Data was transformed to normal distribution using Box - Cox transformation method. The Box - Cox transformation method was used because there were some 0 values in the data set and Box - Cox transformation method can deal with the 0 values.

Data Transformation

Label Encoding

- Label Encoding technique was used to deal with categorical variables and assign them numerical values.

Data Cleaning

- Column names were added to the dataset.
- Null values were present in the form of '?'
- Some values in dataset were also in the form of 'Not in universe'. These values were also treated as Null.
- The null values was replaced by the mode.
- Duplicate values were also removed from data set.

# Visualization of Box Plot and Distribution Plot

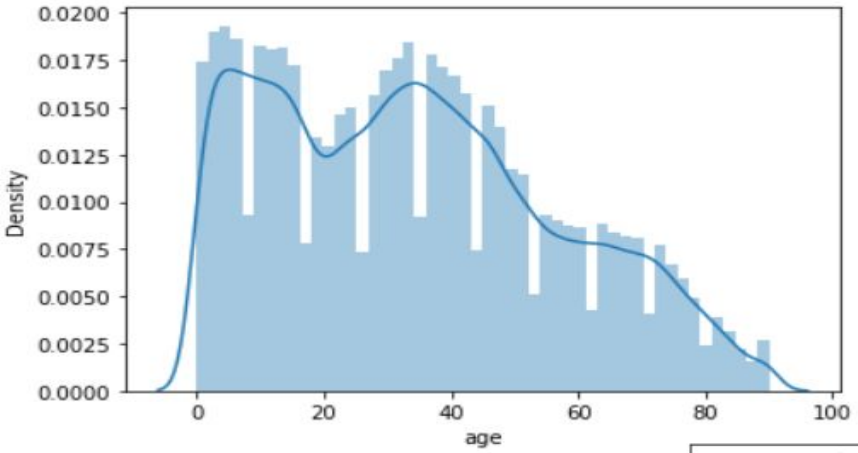



*Fig 1: Distribution Plot of age variable*

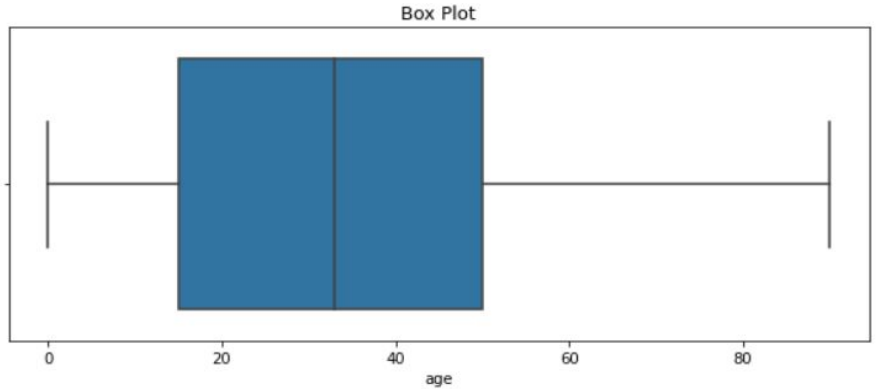



*Fig 2: Box Plot Plot of age variable*

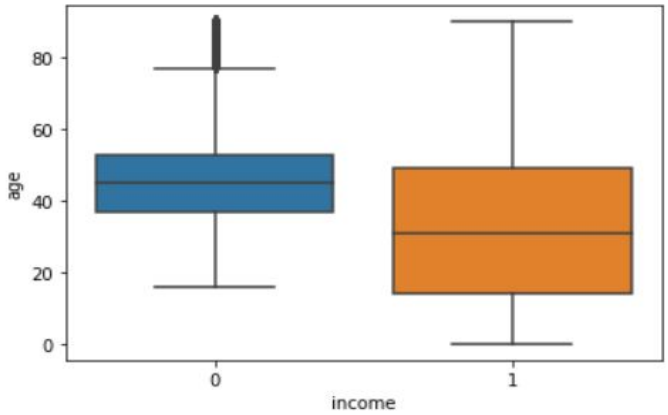


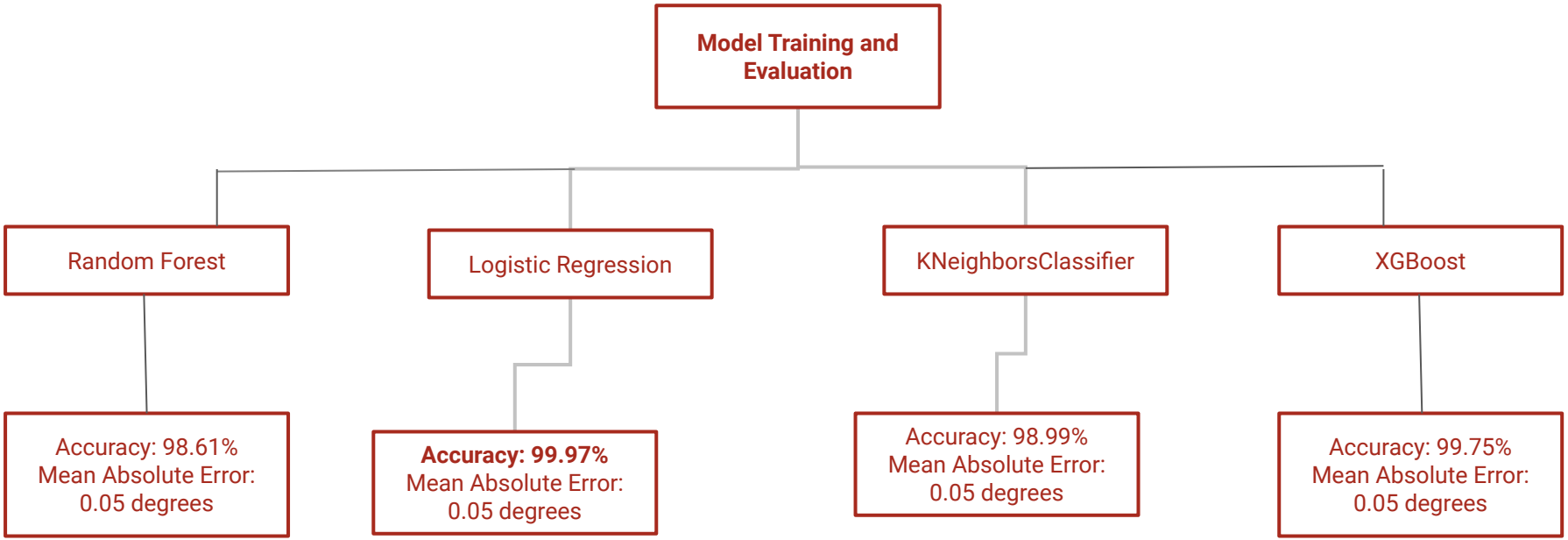

*Fig 3: Box Plot of age variable with respect to target variable (income)*

# Feature Engineering

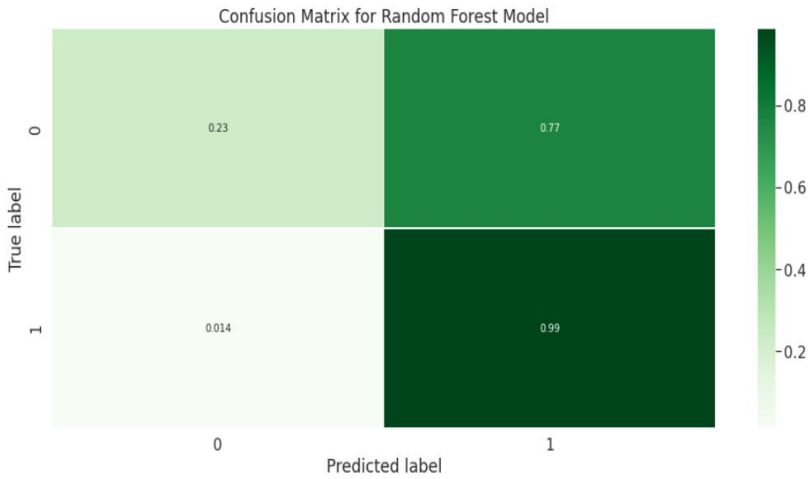
- Multivariate Analysis was conducted to analyse the relationship between different variables.
- Correlation heatmap was prepared to have better understanding the relevant features.
- Chi Squared Method was used to understand correlation between categorical variables and categorical variables.
- Multicollinearity between the variables was identified using VIF. Variables that were having high VIF were removed from model.

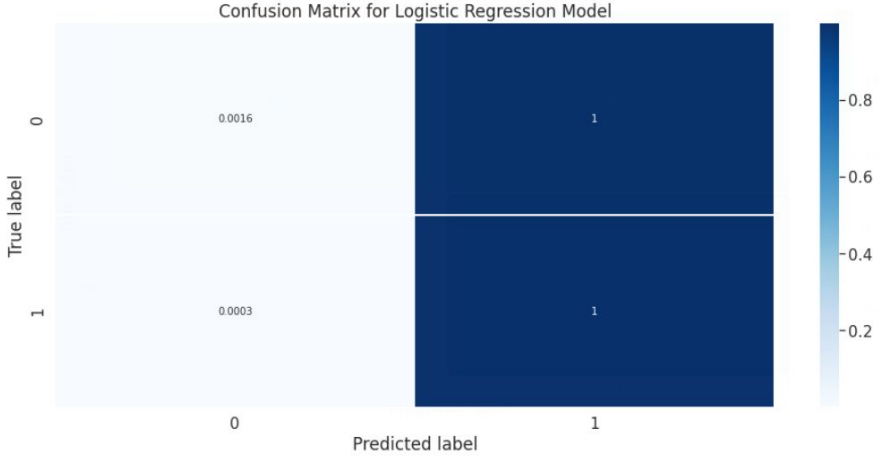
https://github.com/VibhutiChitkara/Census_Income/blob/main/CensusIncome.ipynb

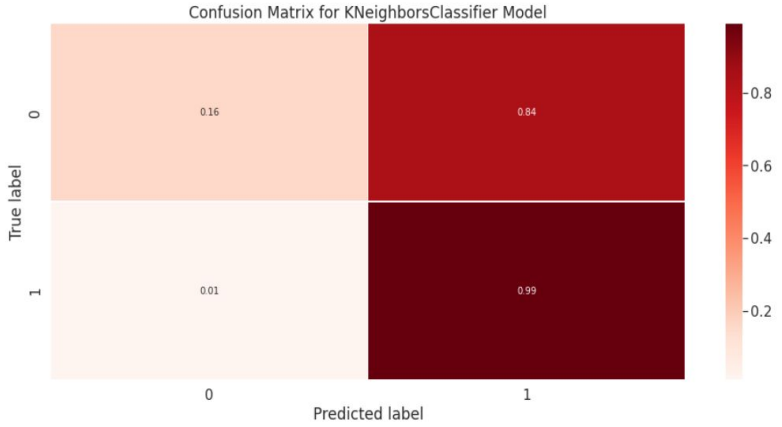
# Model Training and Evaluation

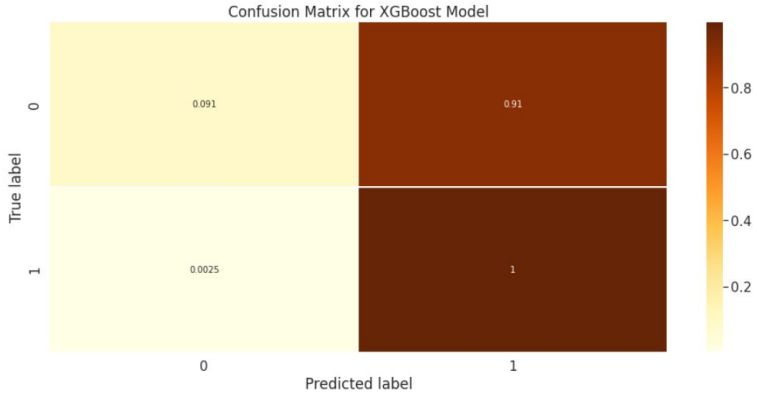


# Confusion Matrix



# Classification Report

## Classification Report of Random Forest

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.49 | 0.23 | 0.31 | 1906 |
| 1 | 0.96 | 0.99 | 0.97 | 33689 |
| accuracy | | | 0.95 | 35595 |
| macro avg | 0.72 | 0.61 | 0.64 | 35595 |
| weighted avg | 0.93 | 0.95 | 0.94 | 35595 |

## Classification Report of Logistic Regression

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.23 | 0.00 | 0.00 | 1906 |
| 1 | 0.95 | 1.00 | 0.97 | 33689 |
| accuracy | | | 0.95 | 35595 |
| macro avg | 0.59 | 0.50 | 0.49 | 35595 |
| weighted avg | 0.91 | 0.95 | 0.92 | 35595 |

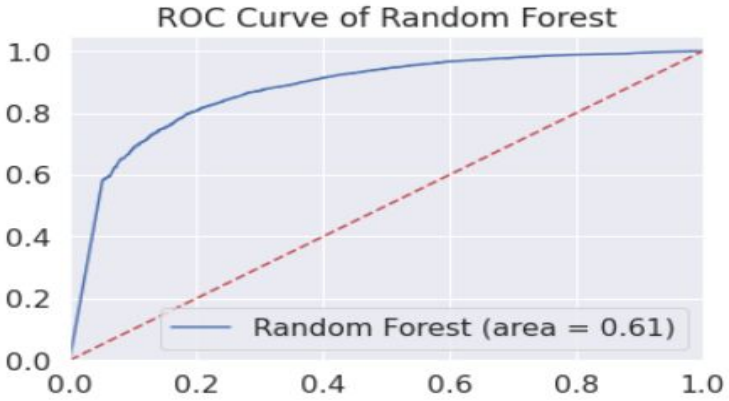
## Classification Report of KNN Classifier

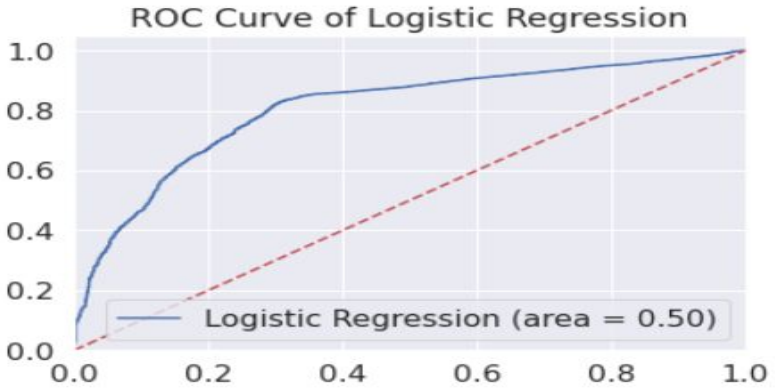
| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.47 | 0.16 | 0.24 | 1906 |
| 1 | 0.95 | 0.99 | 0.97 | 33689 |
| accuracy | | | 0.95 | 35595 |
| macro avg | 0.71 | 0.57 | 0.60 | 35595 |
| weighted avg | 0.93 | 0.95 | 0.93 | 35595 |

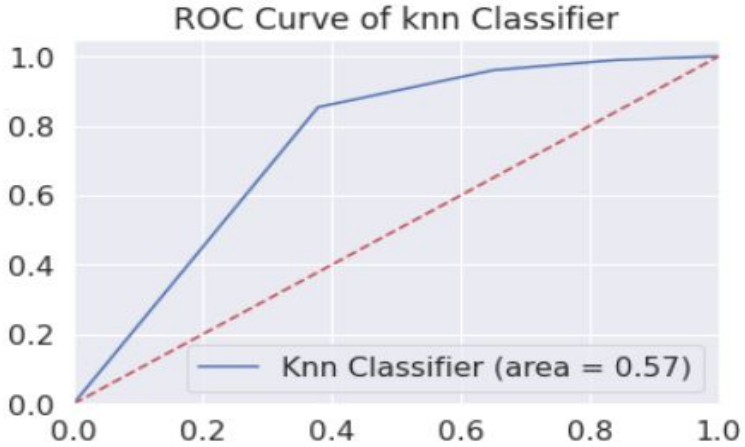
## Classification Report of XGBoost Classifier

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.67 | 0.09 | 0.16 | 1906 |
| 1 | 0.95 | 1.00 | 0.97 | 33689 |
| accuracy | | | 0.95 | 35595 |
| macro avg | 0.81 | 0.54 | 0.57 | 35595 |
| weighted avg | 0.94 | 0.95 | 0.93 | 35595 |

# ROC Curve


ROC Curve of Random Forest
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
Random Forest (area = 0.61)


ROC Curve of Logistic Regression
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
Logistic Regression (area = 0.50)


ROC Curve of knn Classifier
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
Knn Classifier (area = 0.57)


ROC Curve of XGB Classifier
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0
XGB Classifier (area = 0.54)

# Classification Report After Balanced Data

## Classification Report of Random Forest

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.92 | 0.98 | 0.95 | 134794 |
| 1 | 0.98 | 0.92 | 0.95 | 134794 |
| accuracy | | | 0.95 | 269588 |
| macro avg | 0.95 | 0.95 | 0.95 | 269588 |
| weighted avg | 0.95 | 0.95 | 0.95 | 269588 |

## Classification Report of Logistic Regression

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.71 | 0.79 | 0.75 | 134794 |
| 1 | 0.77 | 0.68 | 0.72 | 134794 |
| accuracy | | | 0.74 | 269588 |
| macro avg | 0.74 | 0.74 | 0.74 | 269588 |
| weighted avg | 0.74 | 0.74 | 0.74 | 269588 |

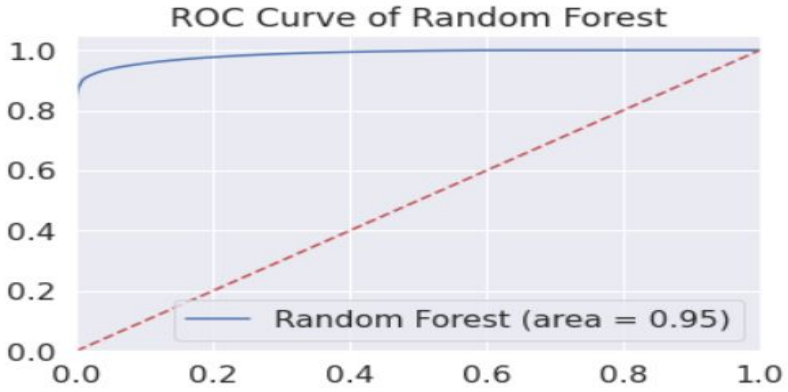
## Classification Report of KNN Classifier

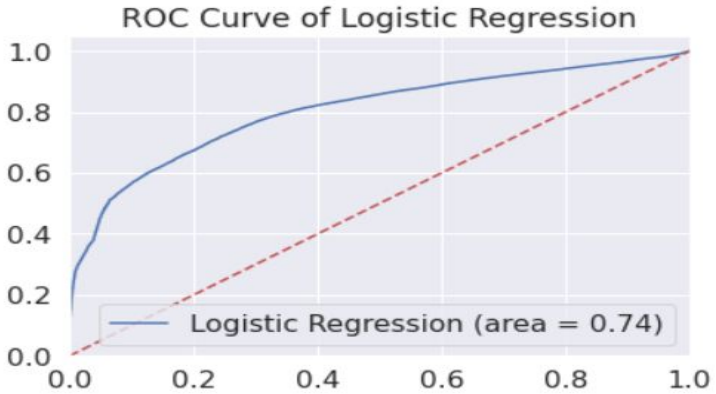
| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.88 | 0.96 | 0.92 | 134794 |
| 1 | 0.96 | 0.87 | 0.91 | 134794 |
| accuracy | | | 0.92 | 269588 |
| macro avg | 0.92 | 0.92 | 0.92 | 269588 |
| weighted avg | 0.92 | 0.92 | 0.92 | 269588 |

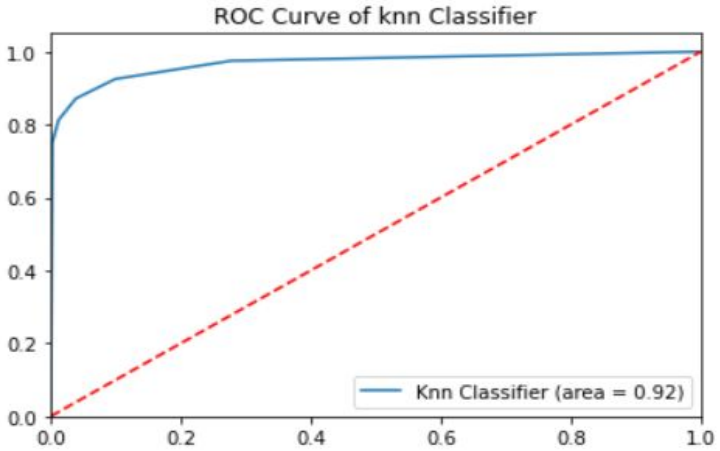
## Classification Report of XGBoost Classifier

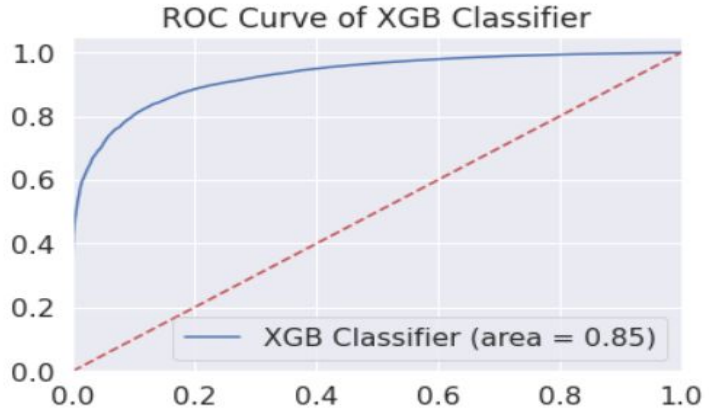
| | precision | recall | f1-score | support |
|---|---|---|---|---|
| 0 | 0.82 | 0.90 | 0.86 | 134794 |
| 1 | 0.89 | 0.80 | 0.84 | 134794 |
| accuracy | | | 0.85 | 269588 |
| macro avg | 0.86 | 0.85 | 0.85 | 269588 |
| weighted avg | 0.86 | 0.85 | 0.85 | 269588 |

# ROC Curve after Balanced Data Set



# Conclusion

- Random Forest and KNN Classifier gave the best f1-score after balancing classes.
- We observed that accuracy is not a good measure of evaluating model performance.
- Jaccard Score of random forest models is around 0.84 which is close to 1 means that the two data sets are similar.
- After balanced data set, true positive and true negative values of the model were improved and false positive and false negatives values of the model were reduced.
- From confusion matrix, True Positive values of Random Forest and KNN Classifier gave the best results. .
- 94.6% of people have less than 50,000 income. The data set was balanced using SMOTE technique.
- 18 features were selected for model training out of 40 features as they were having high correlation values with the target variables.

# Thank You