HIGH INCOME PREDICTION USING MACHINE LEARNING MODELS.

Objective: -

Kaggle's High Income Dataset. also referred to as the "Census Income" dataset. Based on census statistics, it is used to determine whether a person's income is greater than $50,000 per year. To complete this project, I'll implement some machine learning techniques and create a model that can accurately classify a person's pay class (more than or less than $50k). Age, work class, final weight, education, education-number, married status, occupation, relationship, race, sex, capital gain or loss, hours per week, and native country are the characteristics of everyone provided in the dataset.

Dataset: -

Dataset we are using in this dataset is taken from Kaggle names as High-income dataset. The dataset has 68,378 records and 15 features. It's a binary class classification problem. Our target variable is High Income with value 0 and 1 (where 0 means less than 50k/year and 1 means greater than 50k/year).

Data Pre- Processing: -

Before using the data in the machine learning algorithms, preparing the data is a crucial step. Data preprocessing is a data mining technique used to turn the raw data into a format that is both practical and effective.

For our dataset, we also must perform some data pre-processing. First, we load our dataset using pandas and check shape of our dataset which is 68,378 and 15 columns. Datasets contain both integer and float values. Data has some null values in some columns which we removed by adding average of values and somewhere add 0 in null spaces. After that we dropped some unnecessary column such index id. we are also scaling the dataset using Min Max scaler technique so that each value come into one range.

Train – Test Split: -

The train-test split technique is used to gauge the performance of machine learning algorithms when they are used to generate predictions on data that was not used to train the model. Here, our dataset is divided into a train dataset and a test dataset with a ratio of 70% to 30%, with 70% representing our train dataset and 30% representing our test dataset.

Methodology: -

We are using different machine learning algorithms in this project for blood pressure disease prediction and comparing results of each algorithm to check out which model is fitting and performing well on training data set. We are also using some model enhancement techniques to improve performance of the model. These technique is K-fold cross validation. Our sole purpose is to improve performance of model with fine accuracy.

K-FOLD Cross Validation: -

A resampling technique called cross-validation is used to assess machine learning models on a small data sample. The process contains a single parameter, k, that designates how many groups should be created from a given data sample. As a result, the process is frequently referred to as k-fold cross-validation. When k is set to a specific value, that value may be used in place of k when referring to the model, such as when k=5 is substituted for 5-fold cross-validation.

In applied machine learning, cross-validation is mostly used to gauge how well a machine learning model performs on untrained data. That is, to use a small sample to gauge how the model will generally perform when used to make predictions on data that was not included during the model's training.

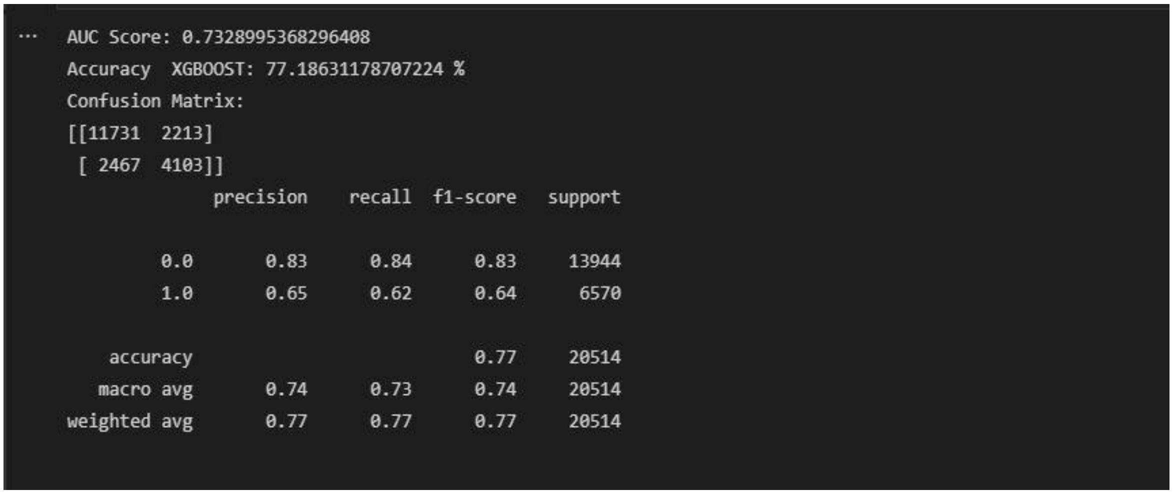
In our project, we are using 5-fold cross validation, in which we are calculating cross validation score on our training data. It is returning us accuracies of 10 folds and mean accuracy. K-fold CV help in decreasing model complexity and in some cases, it also improves model results.

Machine Learning Modelling: -

So, now we are implementing two different machine learning algorithms which are Support vector machine and XGboost classifier to predict high income of a person.

1- XGBoost Classifier: -

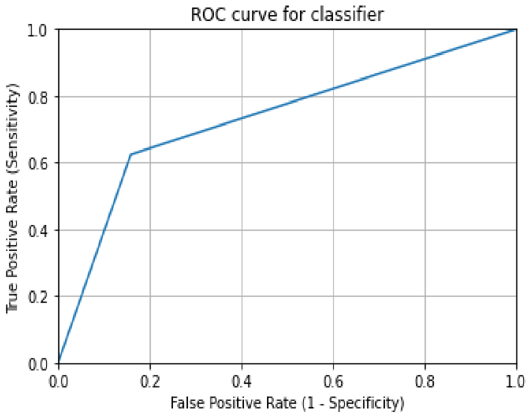
XGBOOST is a decision-tree-based ensemble Machine Learning algorithm that uses a framework. In our case study, XGboost is performing very well and giving us accuracy of 77.05% with good roc, precision and recall values.

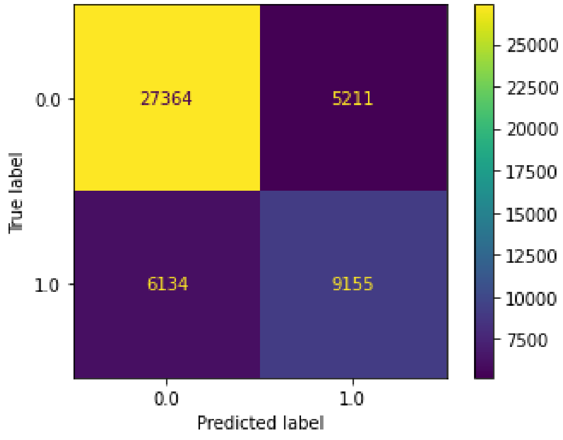


```
...   AUC Score: 0.7328995368296408
      Accuracy  XGBOOST: 77.18631178707224 %
      Confusion Matrix:
      [[11731  2213]
       [ 2467  4103]]
                    precision    recall  f1-score   support

               0.0       0.83      0.84      0.83     13944
               1.0       0.65      0.62      0.64      6570

          accuracy                           0.77     20514
         macro avg       0.74      0.73      0.74     20514
      weighted avg       0.77      0.77      0.77     20514
```

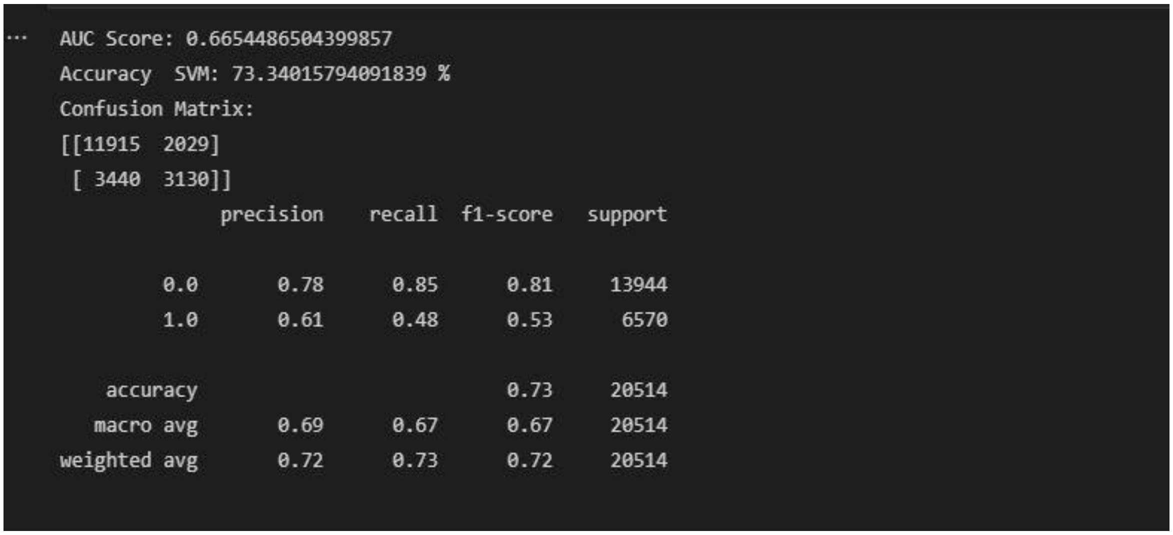




2- Support Vector Machine: -

Support Vector Machine” (SVM) is a supervised that can be used for both classification and regression challenges. However, it is mostly used in classification problems. In the SVM algorithm, we plot each data item as a point in n-dimensional space (where n is several features you have) with the value of each feature being the value of a particular coordinate. Then, we perform classification by finding the hyper-plane that differentiates the two classes very well.

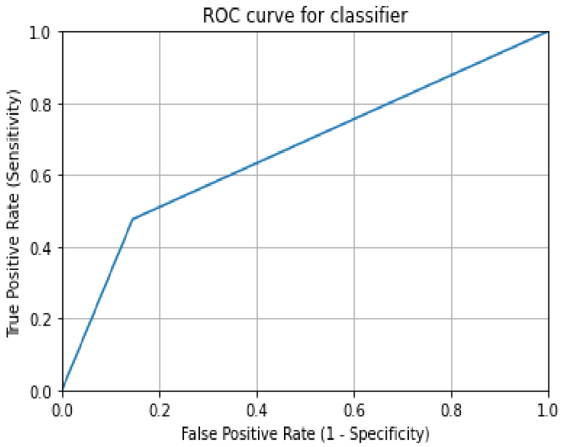
In our project, SVM is performing good in term of accuracy with accuracy score of 73.3% but ROC score is little low of 0.66.

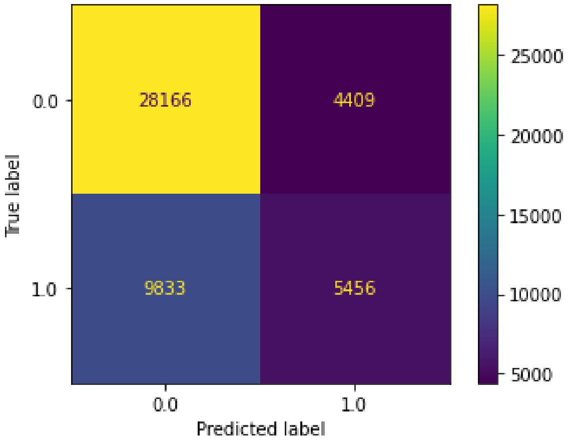


```
AUC Score: 0.6654486504399857
Accuracy  SVM: 73.34015794091839 %
Confusion Matrix:
[[11915  2029]
 [ 3440  3130]]
              precision    recall  f1-score   support

         0.0       0.78      0.85      0.81     13944
         1.0       0.61      0.48      0.53      6570

    accuracy                           0.73     20514
   macro avg       0.69      0.67      0.67     20514
weighted avg       0.72      0.73      0.72     20514
```





Conclusion: -

From above results we concluded that we should implement some other techniques like feature engineering and parameter tuning to increase our model accuracies more. Among these two algorithms, XGboost is performing better than SVM in term of Accuracy, ROC, Specificity, Precision etc.