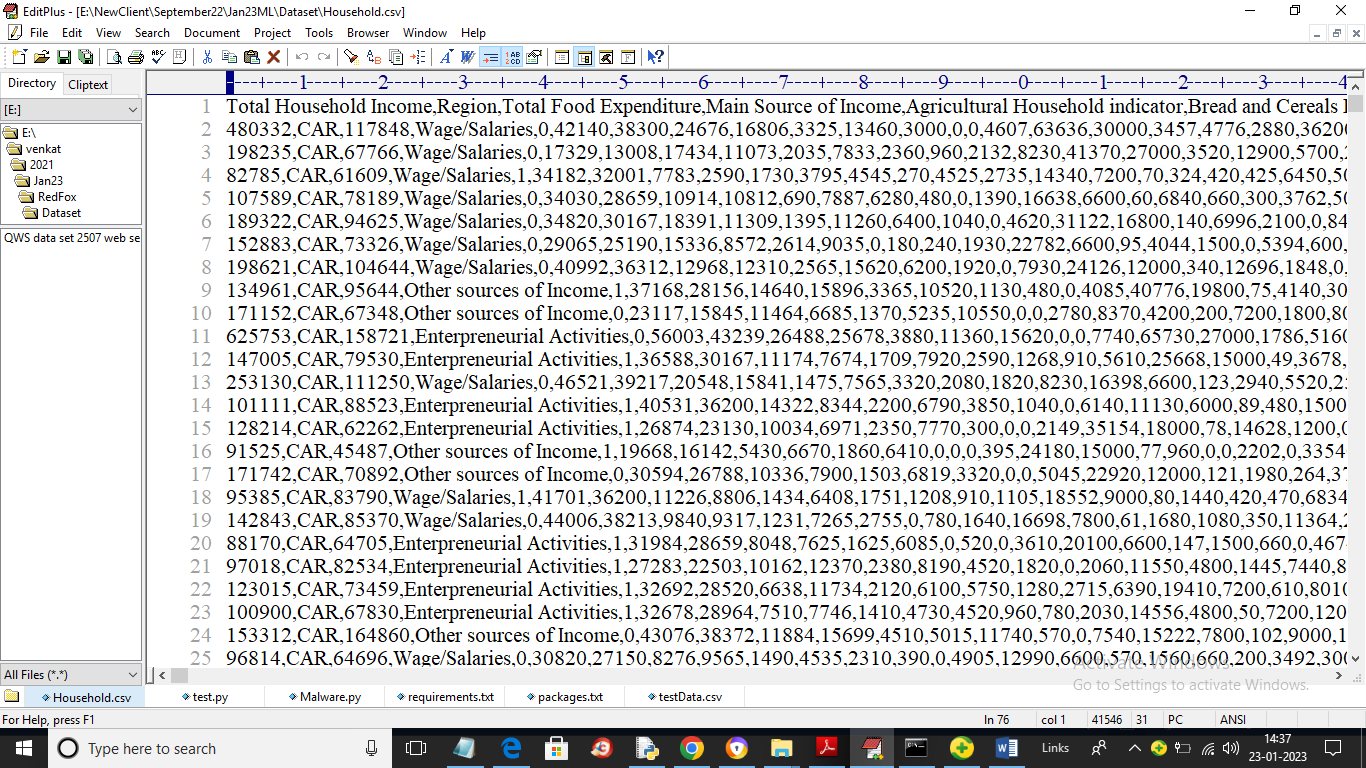
Machine Learning Models for Predicting Financially Vigilant Low-Income Households

Recently world has face Covid19 Pandemic which causes death and financial problems to billions of peoples. Some countries has arrange relive packages for their poor or low income citizens but due to insufficient time and improper information relief work was not sufficient and many poor people’s struggle a lot. To overcome from above issue author of this paper employing machine learning algorithms to identify peoples with less or no savings so in future proper packages will be provided to them.

In propose paper author is employing and evaluating performance of 5 different classifiers such as Decision Tree with GINI and Info Gain parameters, Random Forest with GINI and Info Gain Parameters, Naïve Bayes, Neural Network MLP and Gradient Boosting. All algorithms performance was evaluated in terms of accuracy and FSCORE and among all algorithms Gradient Boosting was giving high accuracy.

In propose work author has used House Hold income of Malaysian peoples but this dataset is not available on Internet so we have used House Hold Income dataset from KAGGLE and below screen showing dataset details

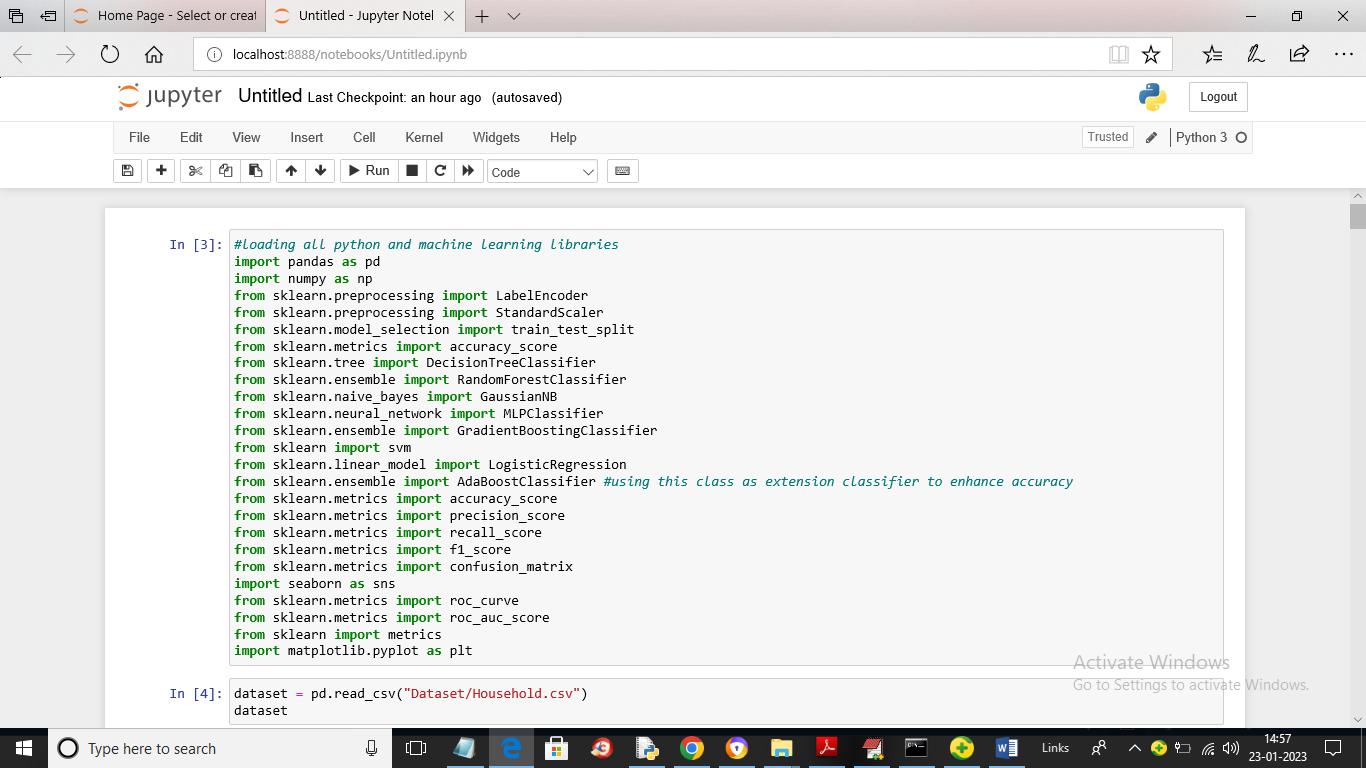


In above dataset screen first row contains dataset column names and remaining rows contains dataset values and in above dataset we can see person total incomes and house hold expenditure and in below screen we can see last column as SAVINGS with values as 0 and 1 where 0 means SAVINGS and 1 means ‘No Savings’.

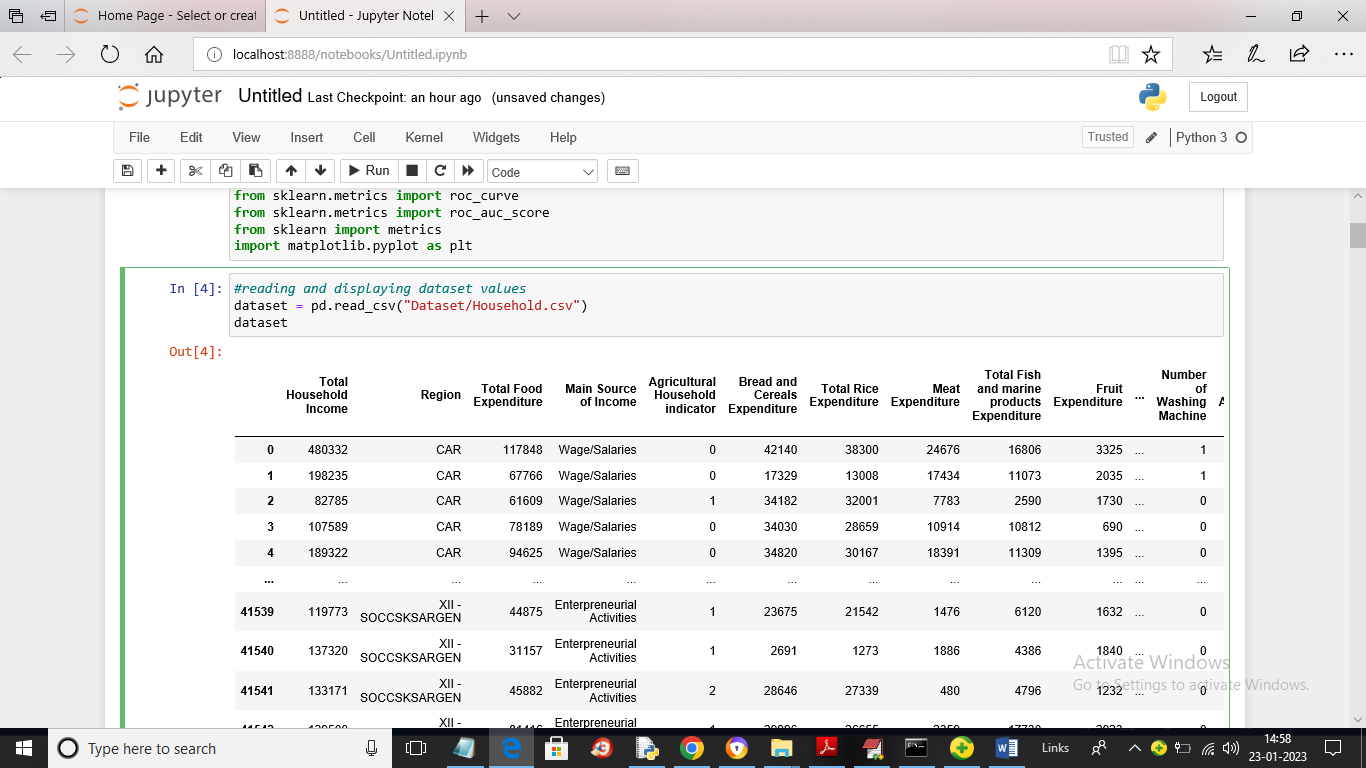
So by using above dataset we will trained all algorithms and test their performance.

Extension Concept: in propose paper author has used only traditional old algorithms but not used any advance ML algorithm like XGBOOST, Extreme Learning Machine or ADABOOST. So as extension we have experiment with all the above advance ML algorithms and in all algorithms ADABOOST is giving 100% accuracy on same dataset.

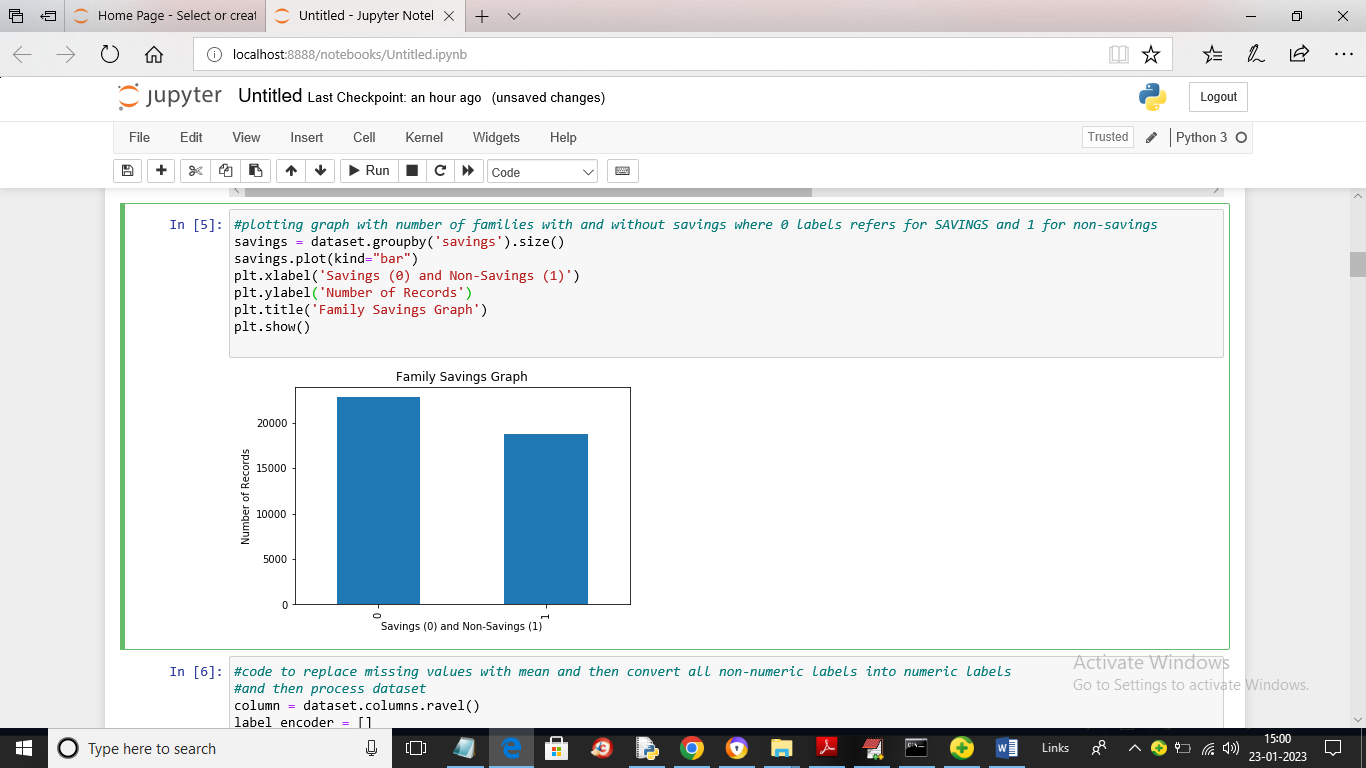
We have coded this algorithms using JUPYTER notebook and below are the code and output screens with blue colour comments



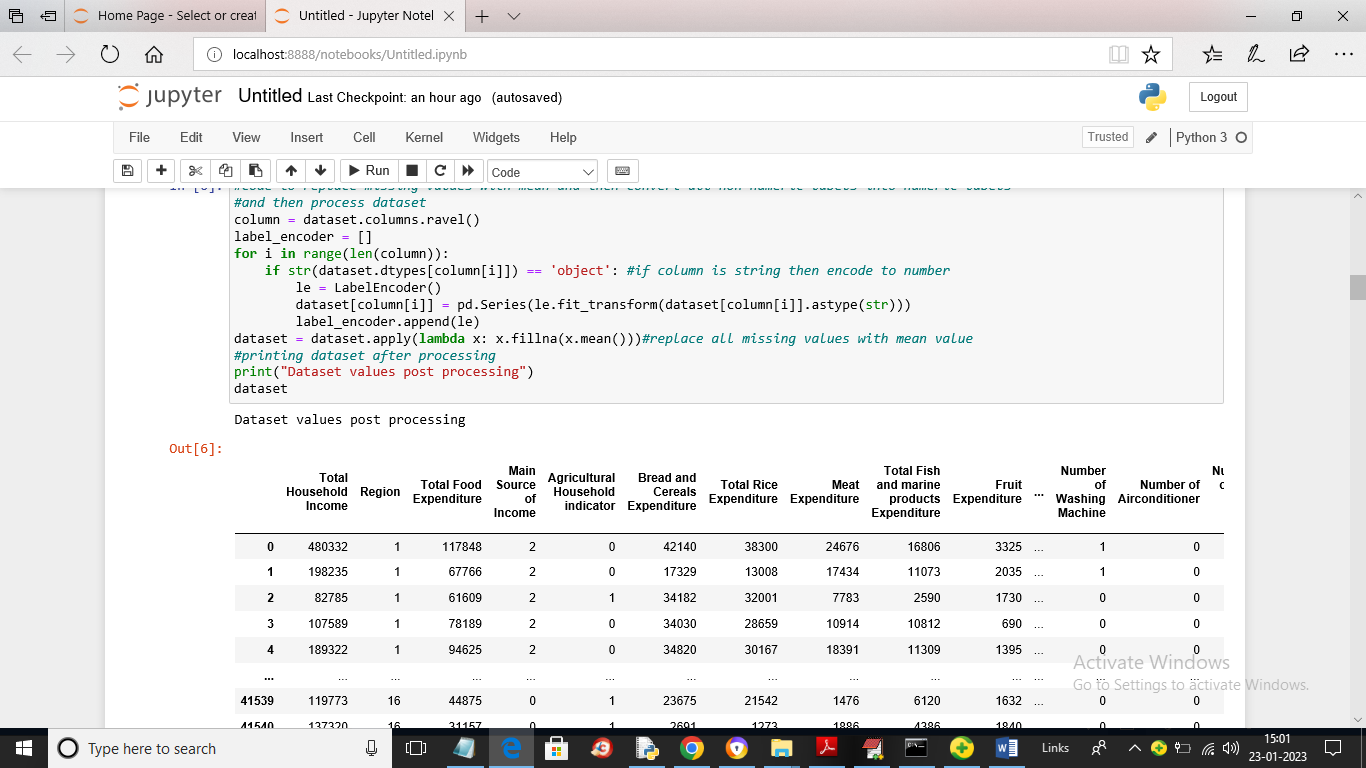
In above screen loading all python and machine learning libraries and you can read blue colour comments to know about coding



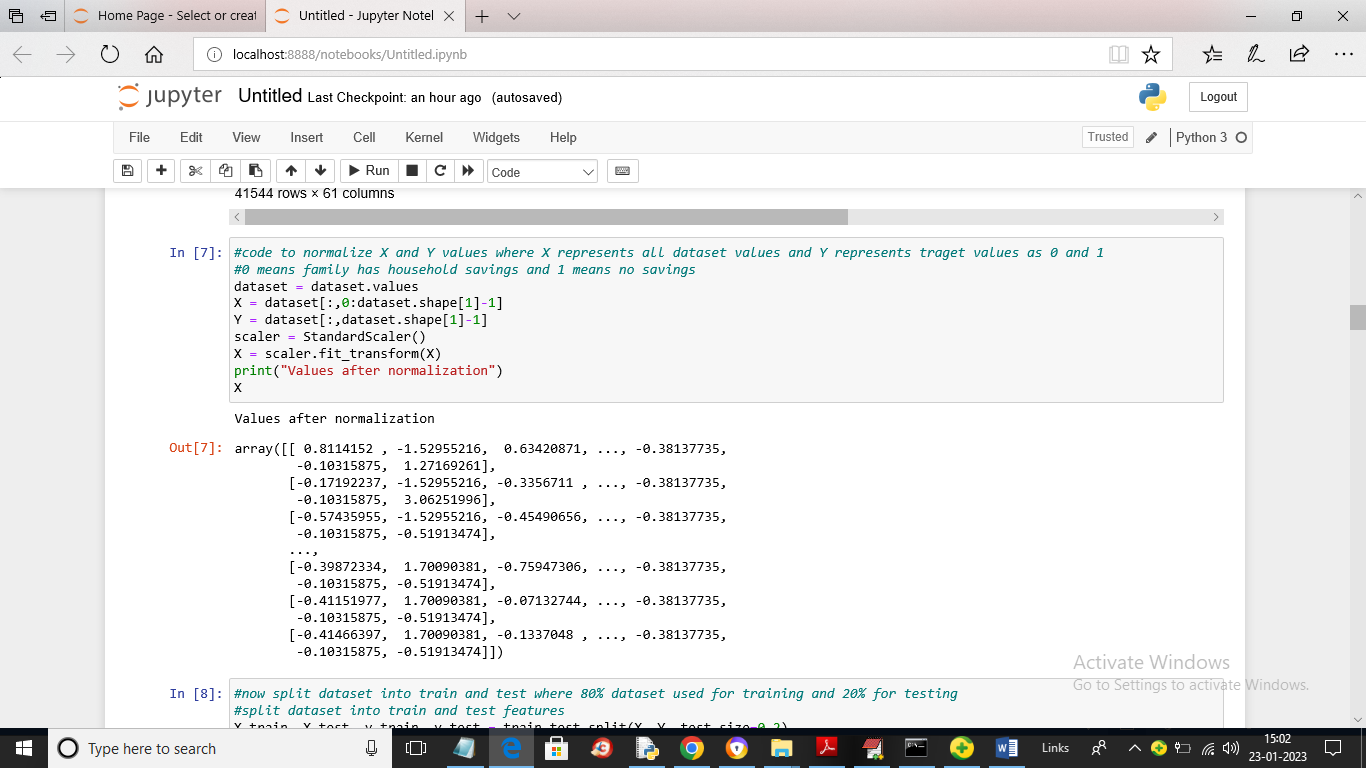
In above screen reading and displaying dataset values and in above dataset we can see some values are numeric and some are non-numeric and ML algorithms take only numeric values so we need to encode those values to numeric by using label encoder class



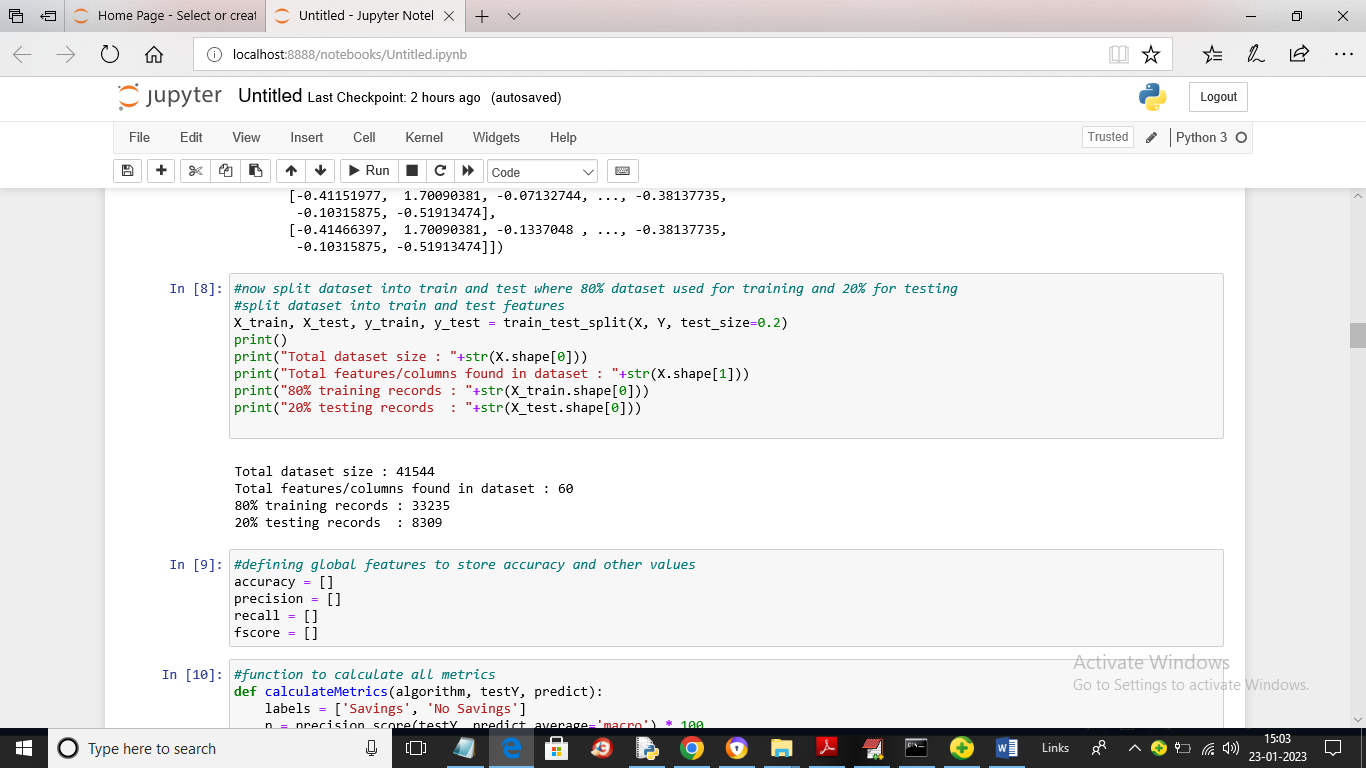
In above screen we are finding total families with saving and non-savings and then plotting in graph where x-axis represents labels as 0 (savings) and 1 (non-savings) and y-axis represents count



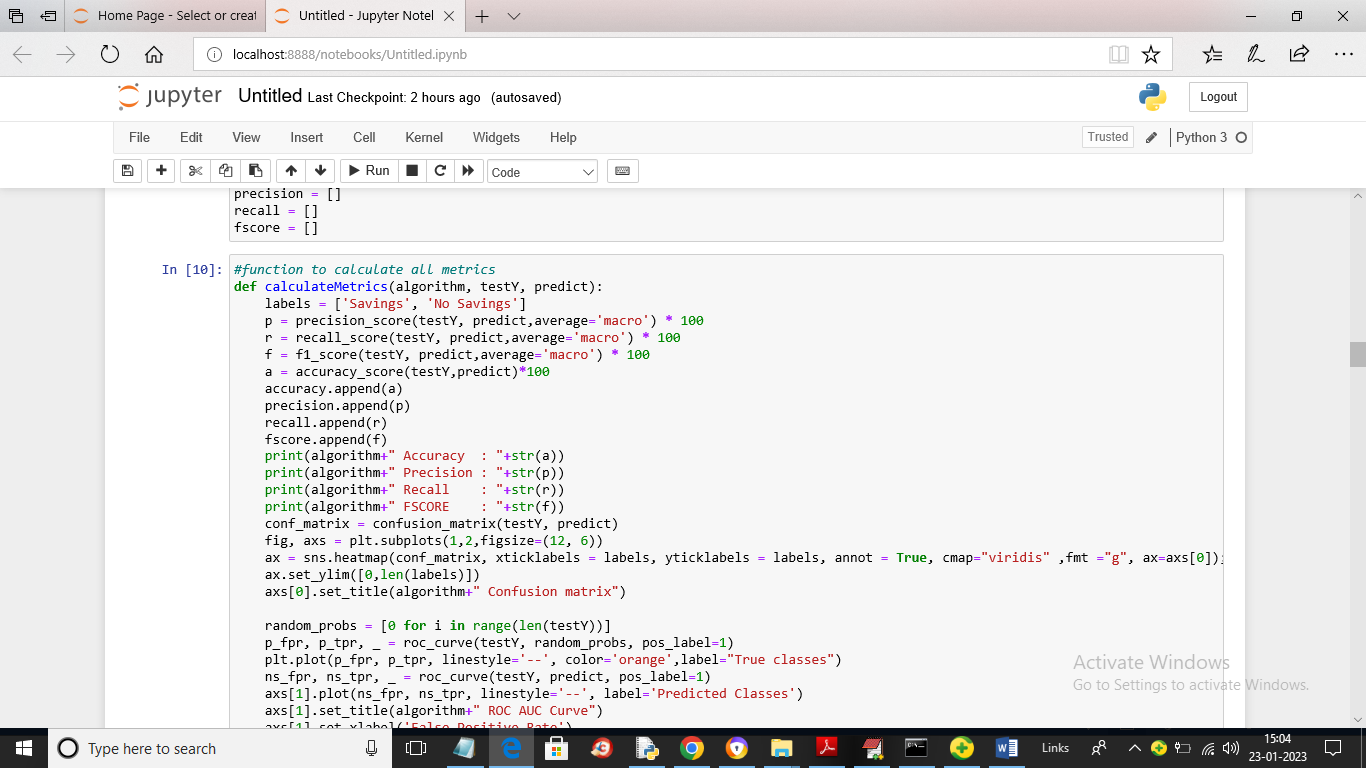
In above screen we are processing dataset values to convert non-numeric data to numeric data and then replacing all missing values with MEAN value and after encoding we can see all values are converted to numeric only



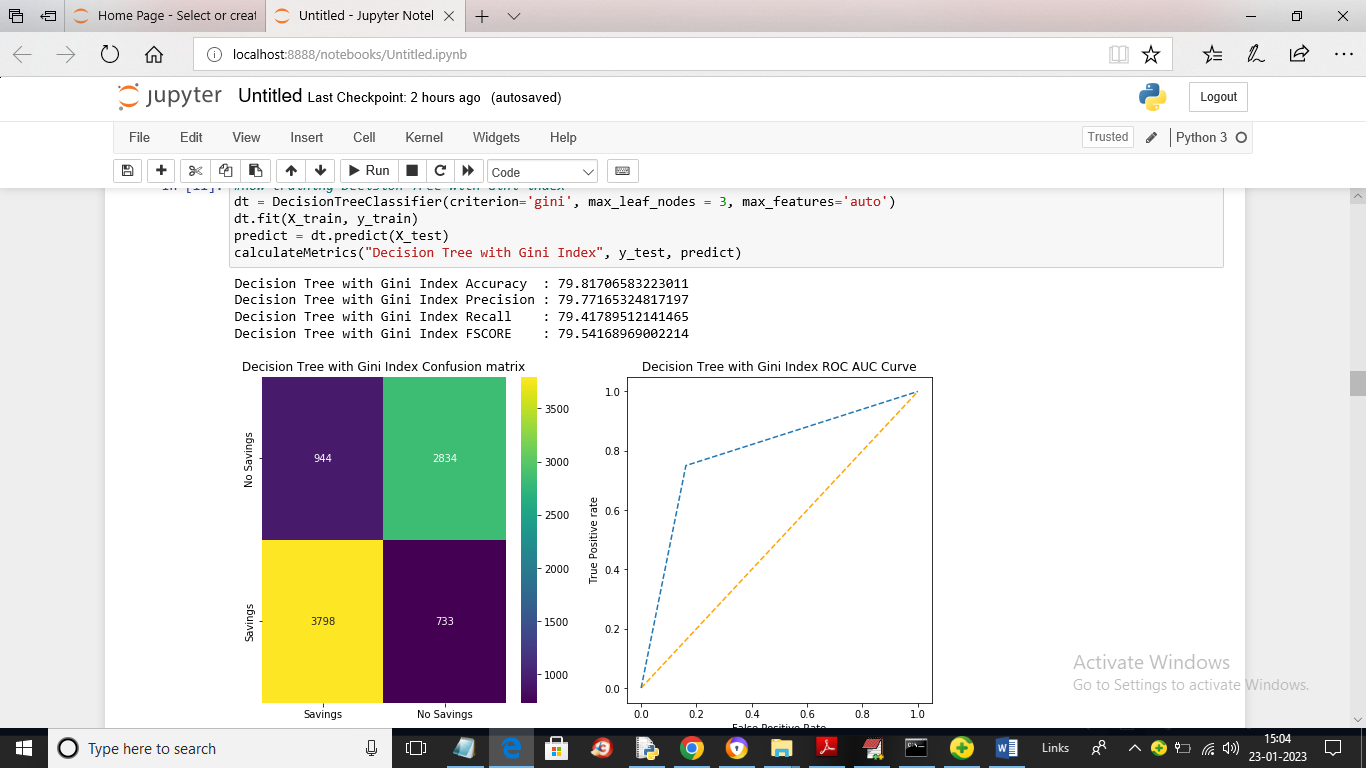
In above screen we are normalizing the dataset values using Standard Scaler class to convert values to MIN and MAX between 0 and 1



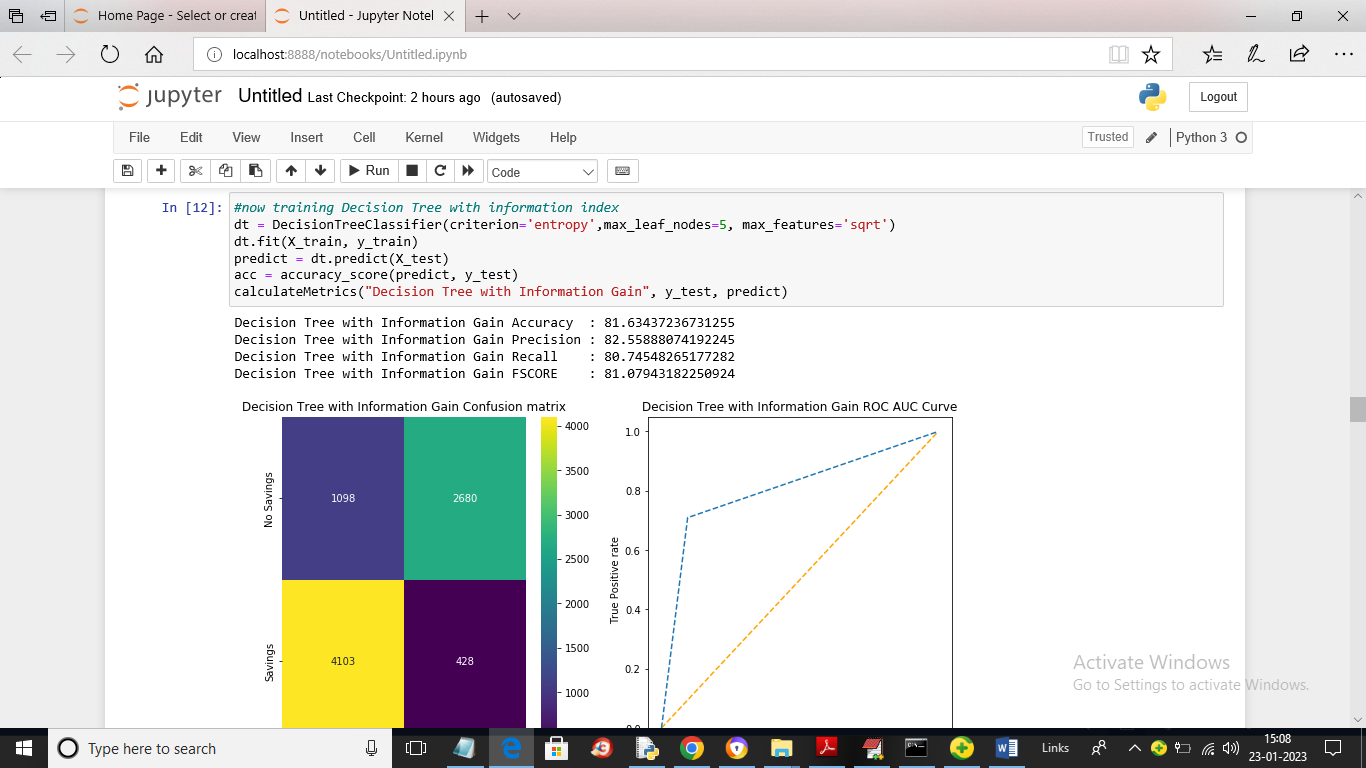
In above screen we are splitting dataset into train and test where application using 80% dataset for training and 20% for testing and then defining global array variables to store accuracy and other metrics



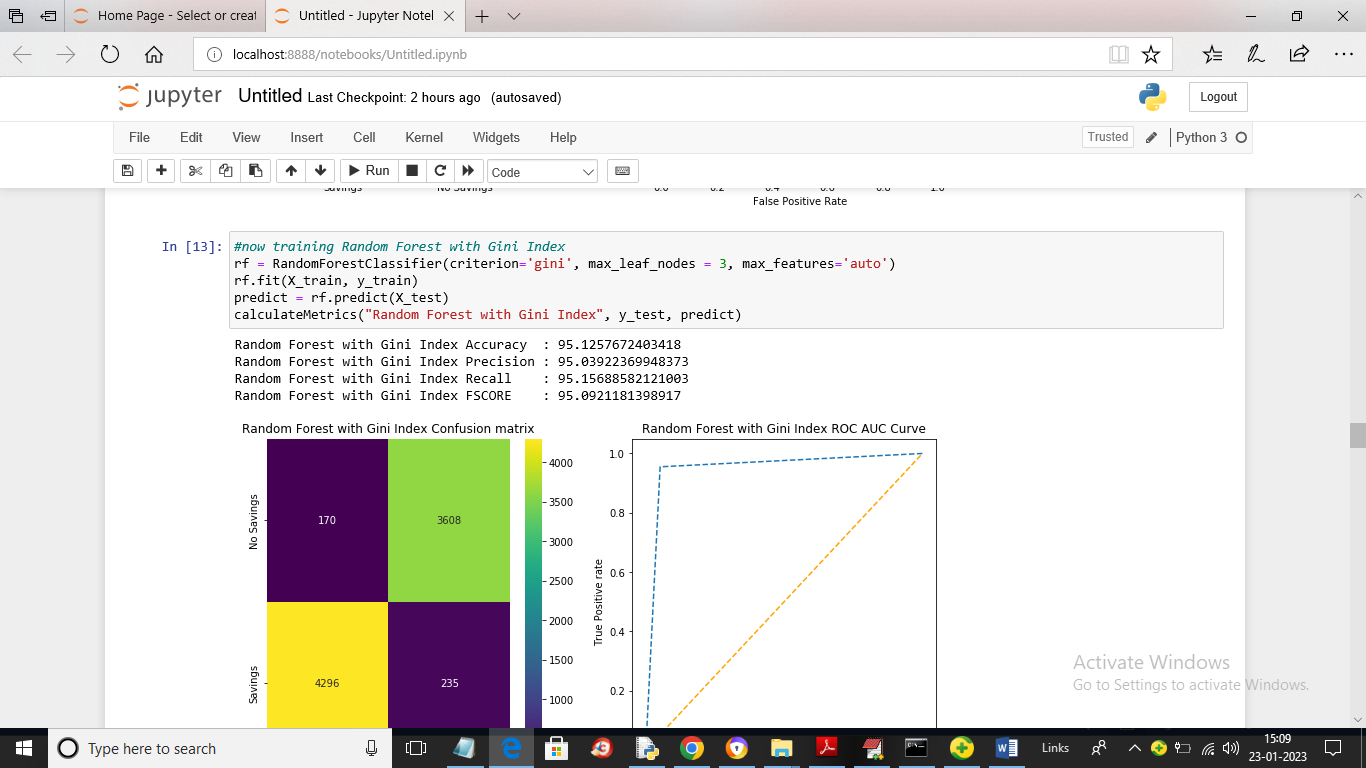
In above screen defining function to calculate accuracy and other metrics



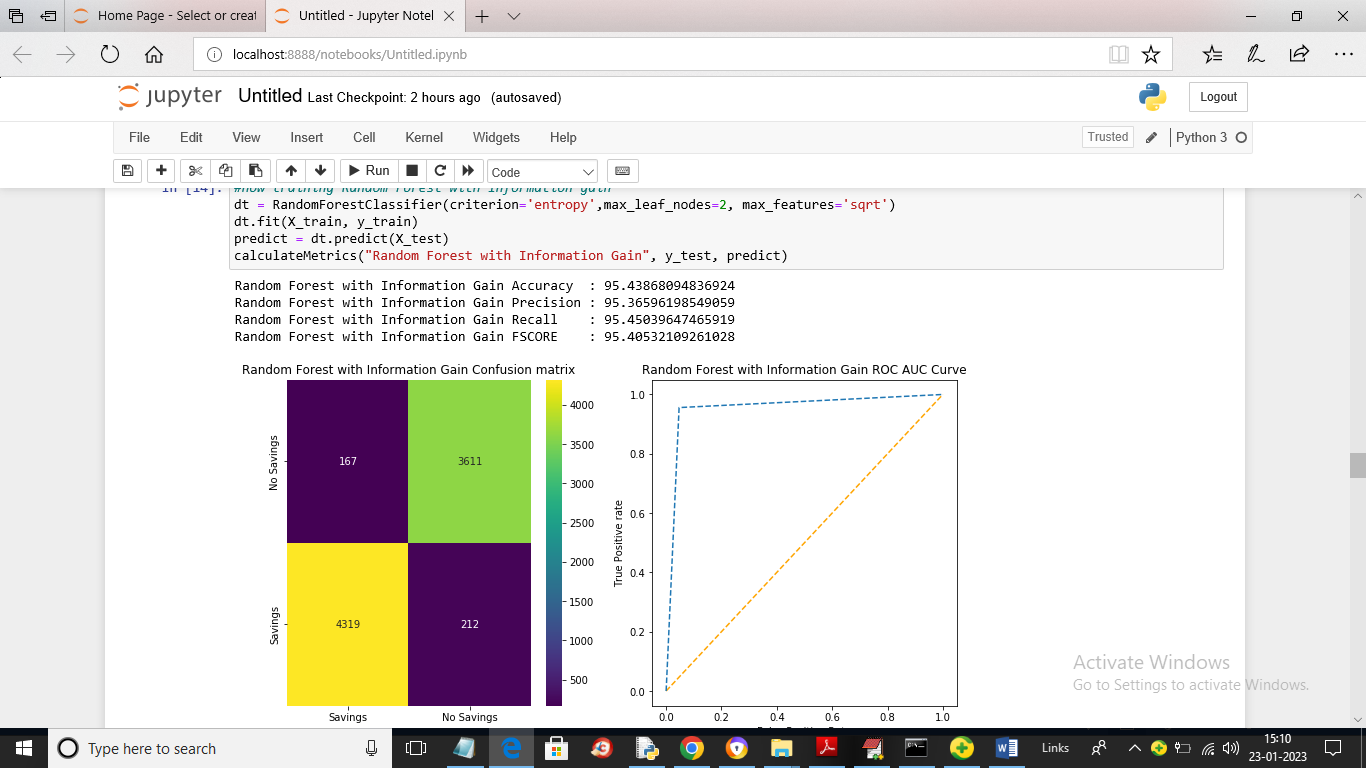
In above screen we are training Decision Tree with GINI index and we got its accuracy as 79% and we can see other metrics also. In above confusion matrix graph x-axis represents Predicted Labels and y-axis represents True Labels and blue colour boxes contains incorrect prediction count and yellow and green boxes contains CORRECT prediction count. In above ROC graph x-axis represents False Positive Rate and y-axis represents True Positive Rate and in blue lines comes on top of orange line then predictions are TRUE and if comes below orange line then predictions are false.



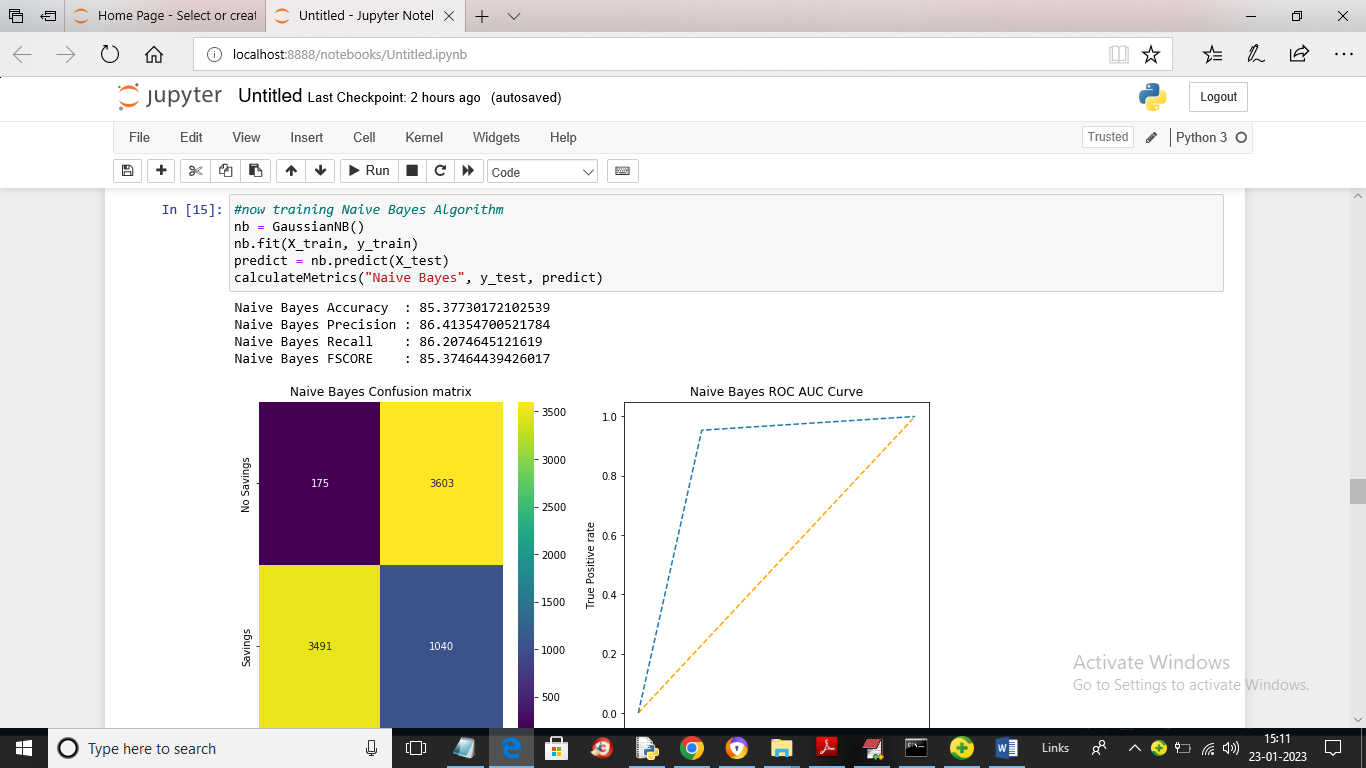
In above screen we are training Decision Tree with Info Gain and we got its accuracy as 81%



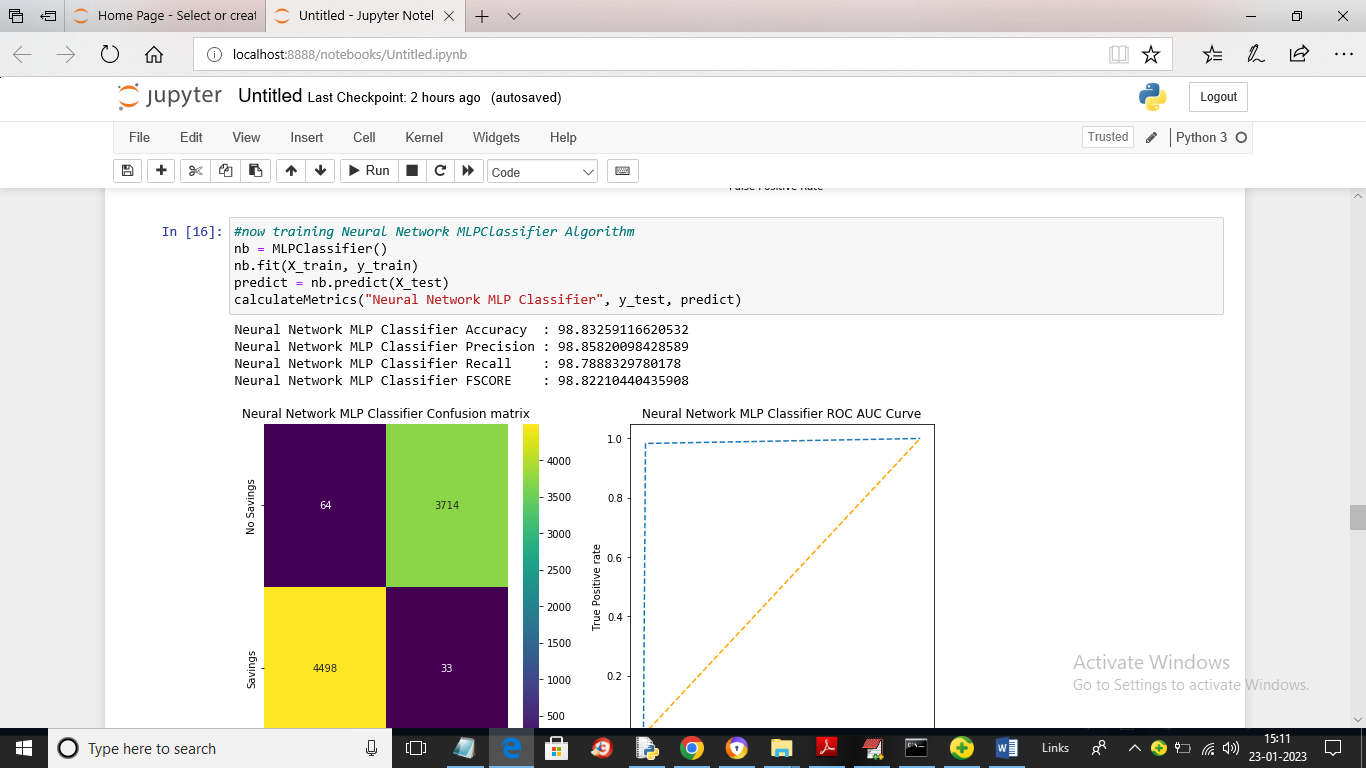
In above screen training Random Forest with GINI parameter and we got its accuracy as 95%



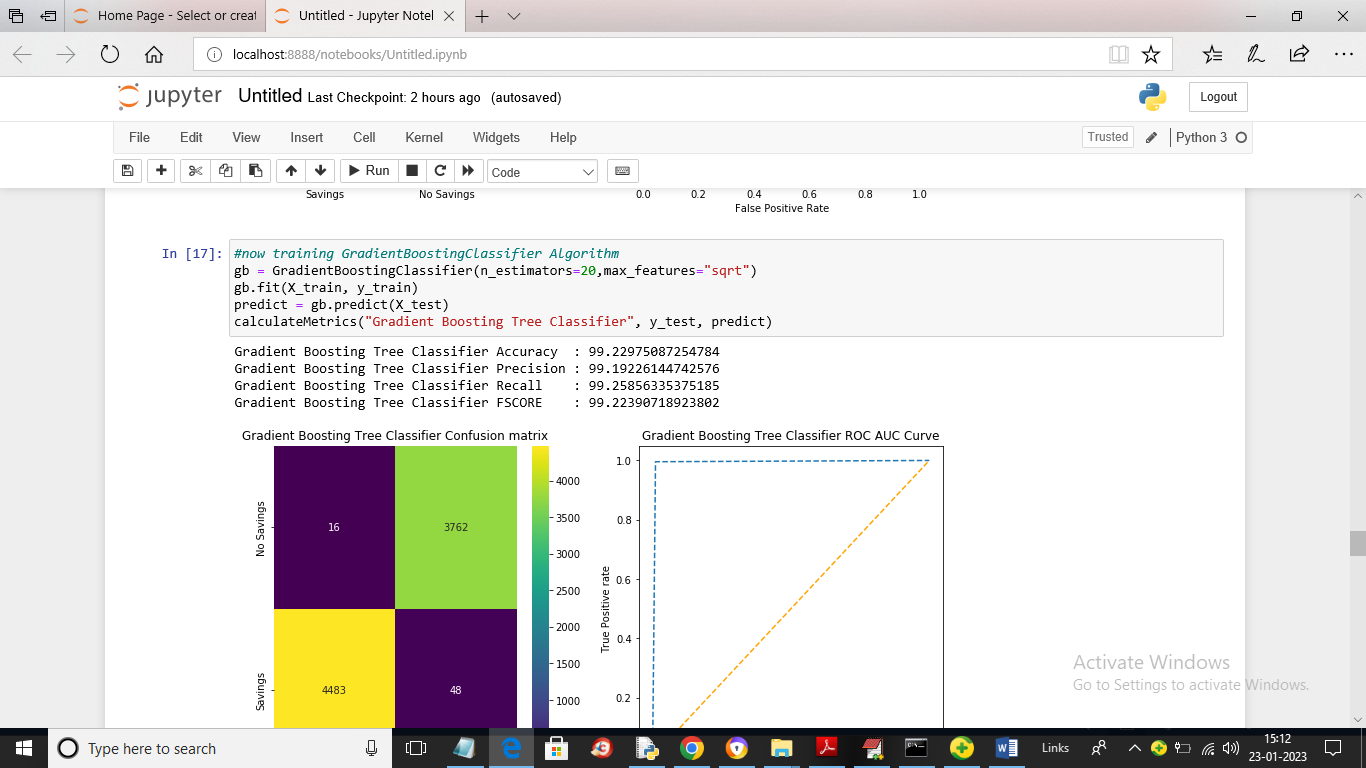
In above screen training Random Forest with Info Gain and we got its accuracy as 95.43%



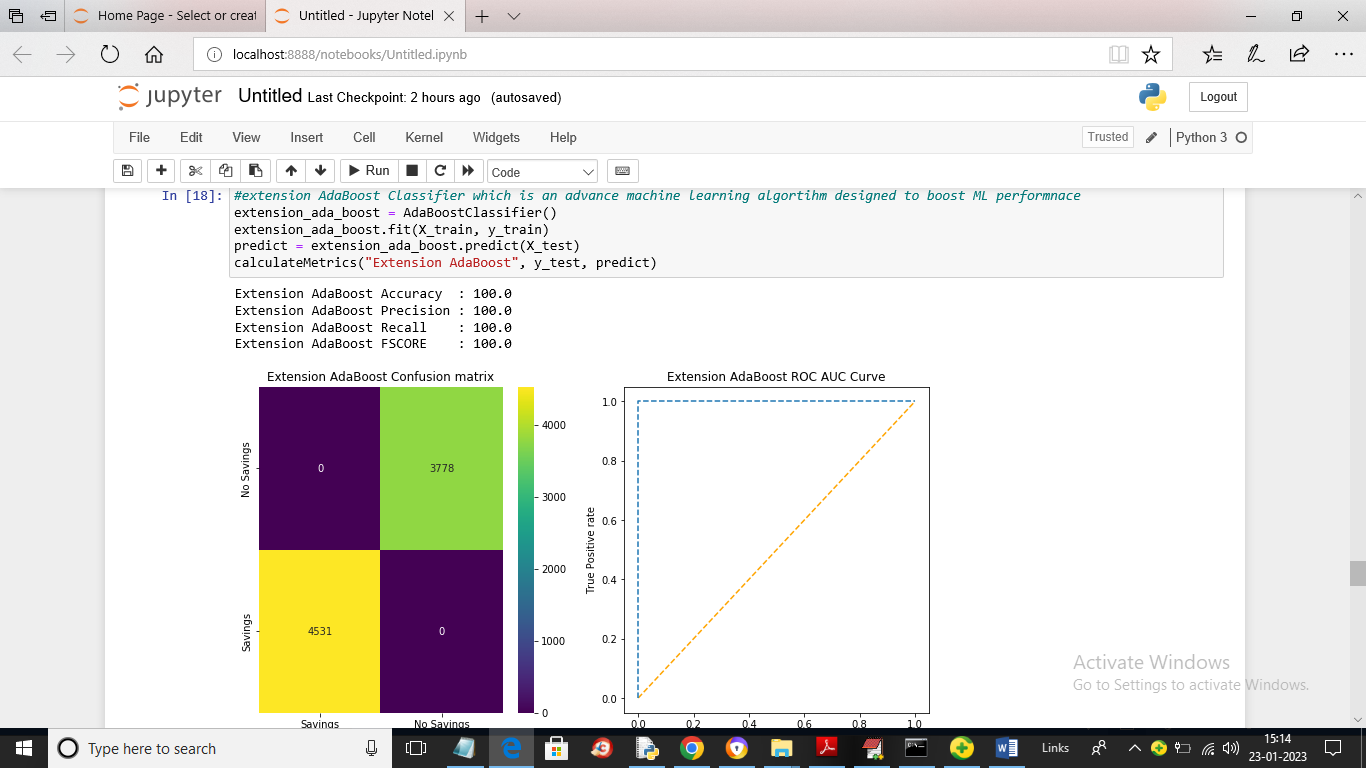
In above screen training Naïve Bayes and we got its accuracy as 85%



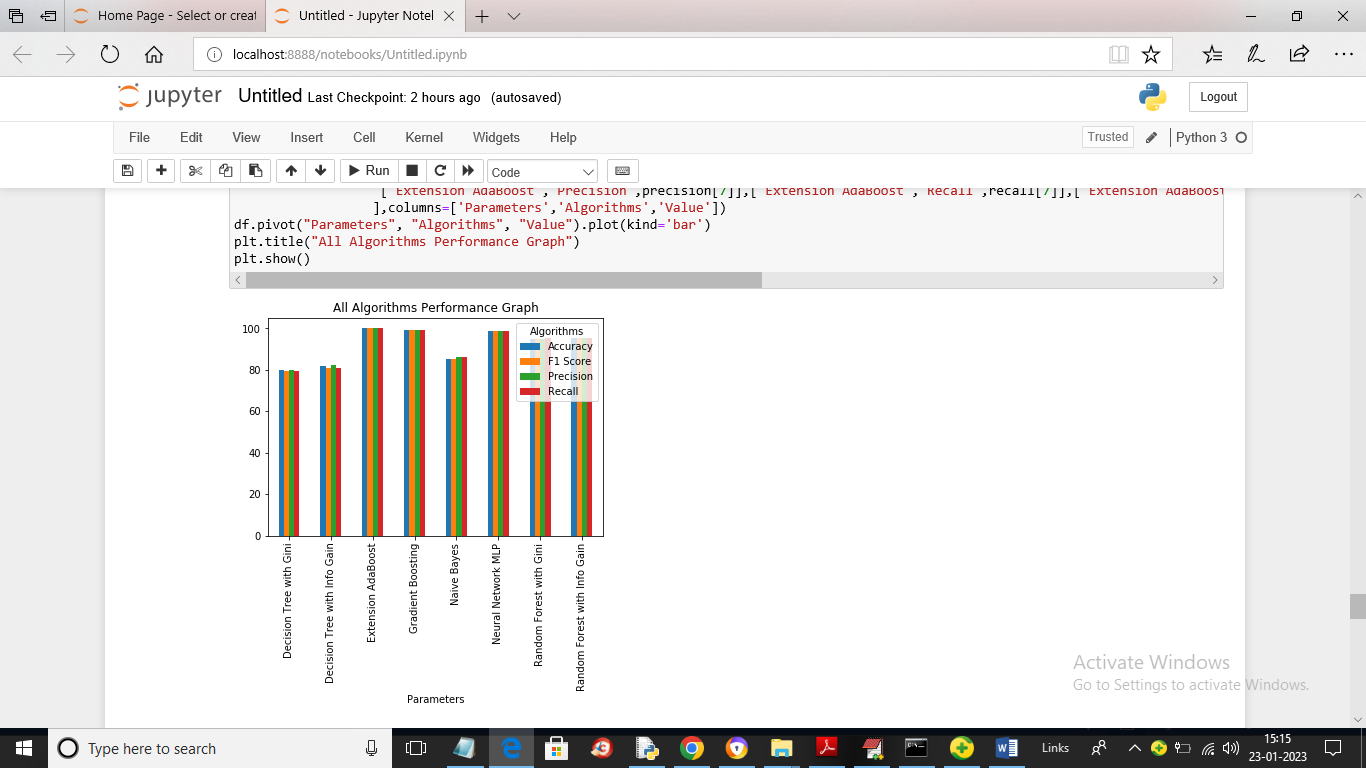
In above screen we are training Neural Network MLP classifier and we got its accuracy as 98.83%



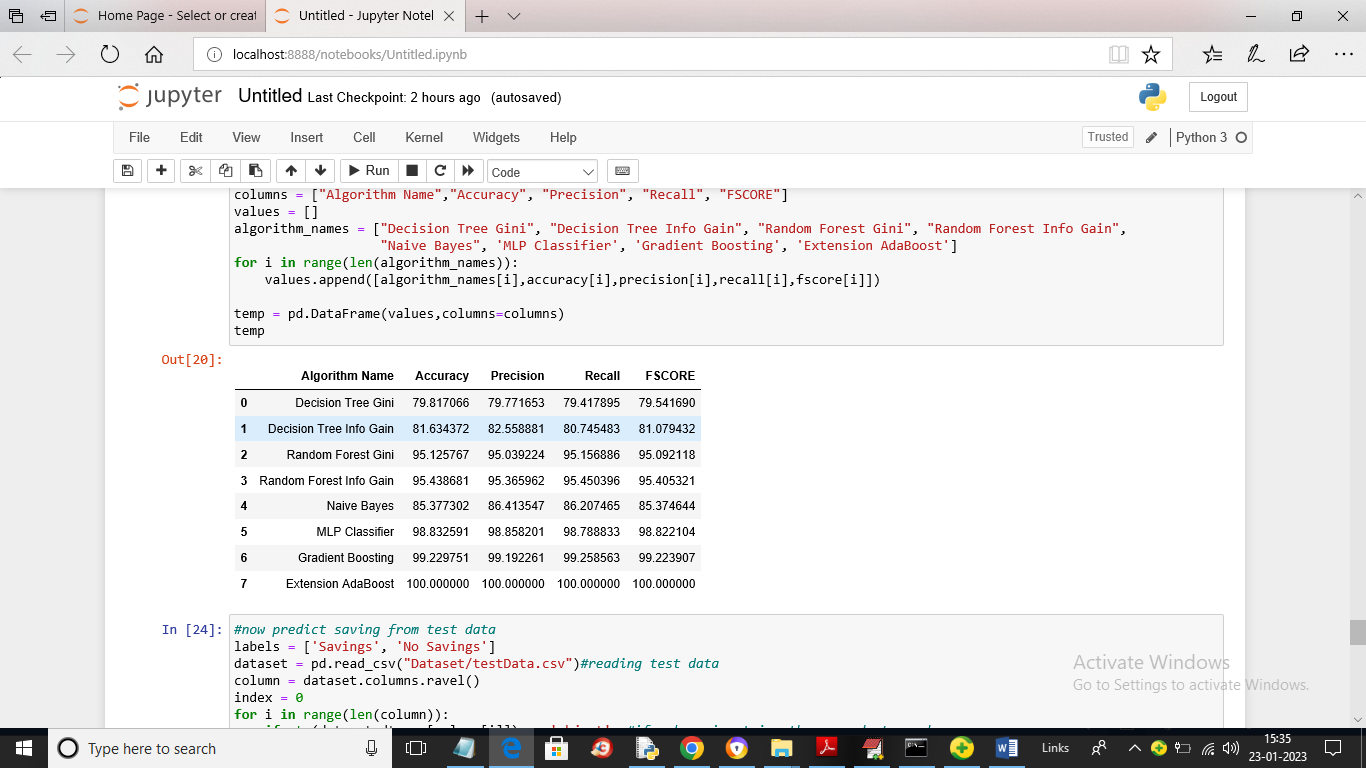
In above screen we are training Gradient Boosting and we got its accuracy as 99%



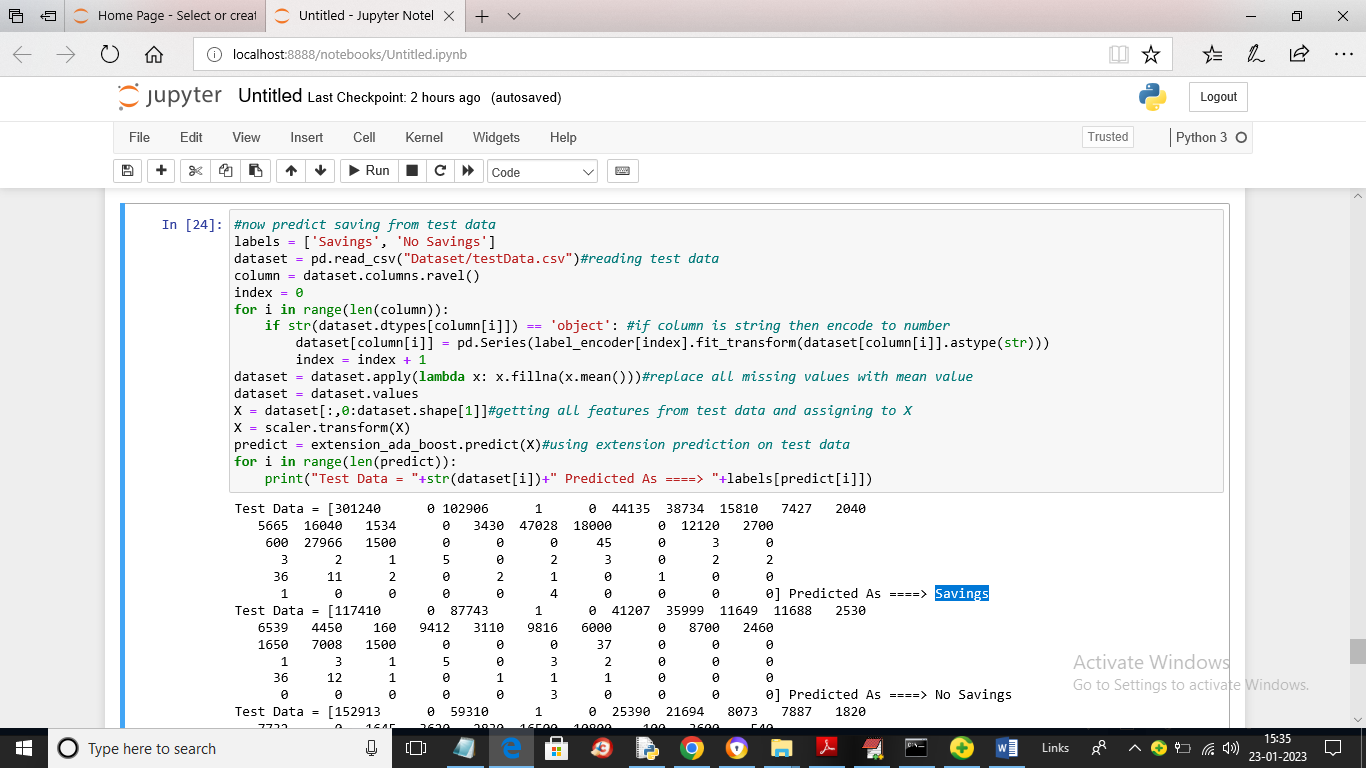
In above screen we are training Extension ADABOOST classifier and we got its accuracy as 100%



In above graph x-axis represents algorithm names and y-axis represents accuracy, precision and other metric in different colour bars. In above graph we can see extension ADABOOST got high accuracy



In above screen we can see performance of all algorithms in tabular format



In above screen we are reading test data and then performing prediction and in above screen in square bracket we can see test data and after arrow symbol =🡺 we can see prediction output as ‘Savings’ or ‘Non-Savings’;

