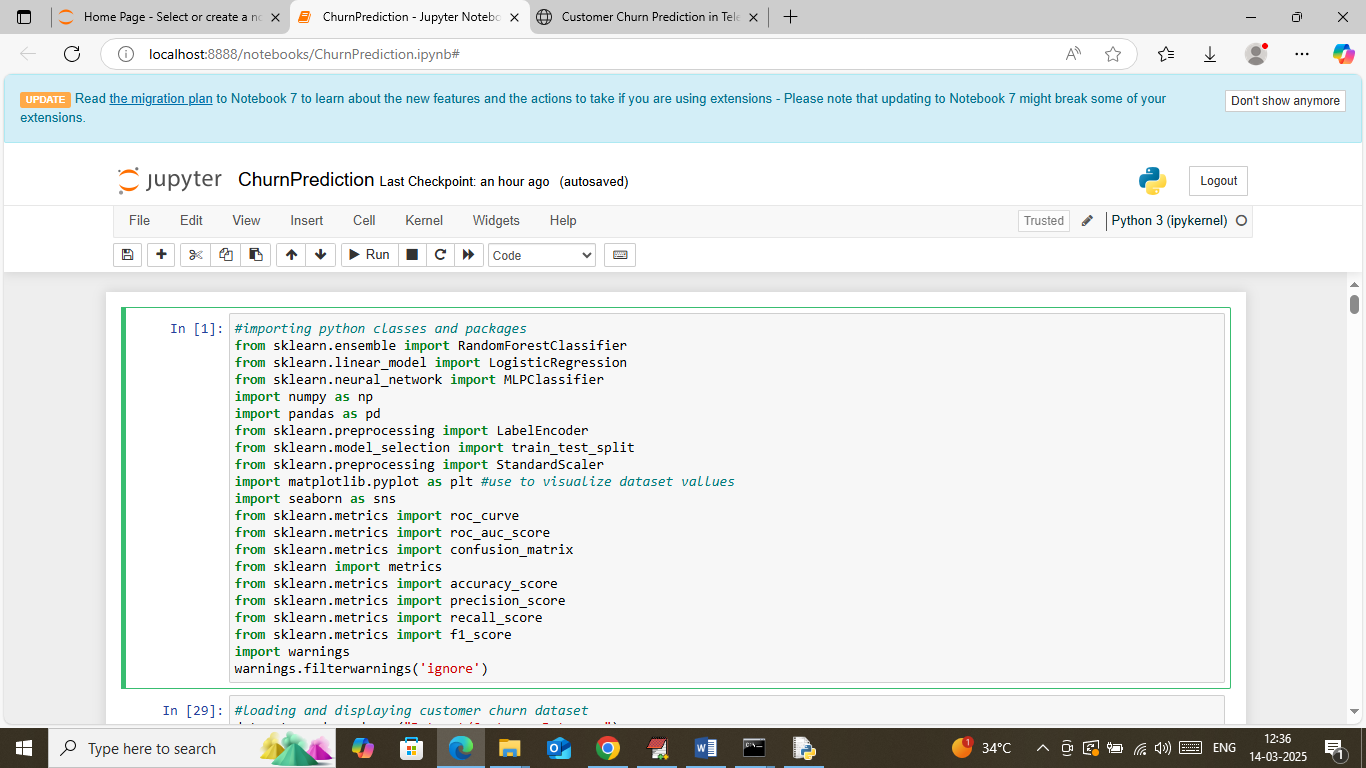
Customer Churn Prediction in Telecom Industry

In this project as per your instructions we have used JUPYTER tool for data visualization, analysis, processing, training and testing. We have used Django web framework for category wise churn visualization and for prediction on sample test data.

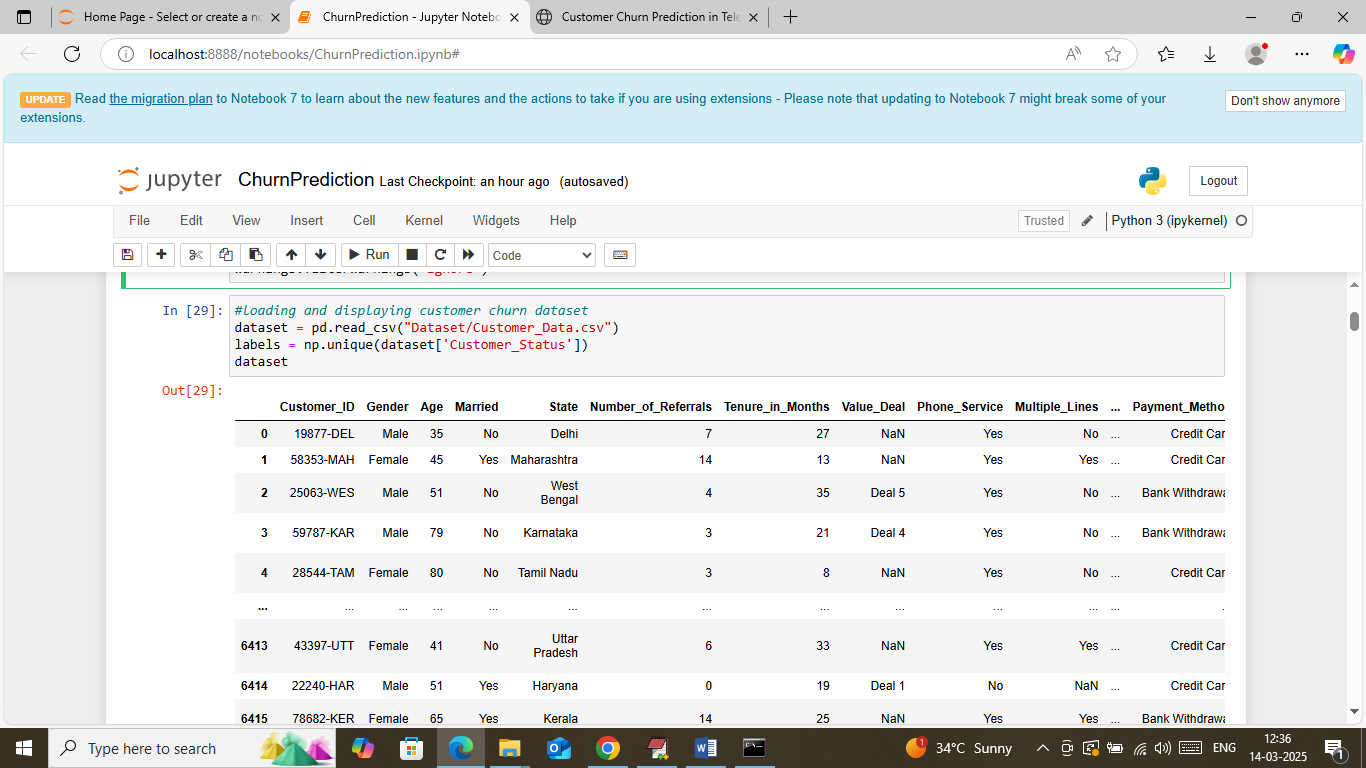
We have evaluated performance of multiple ML algorithms such as Random Forest, Logistic Regression and MLP neural network. Each algorithm performance is evaluated in terms of accuracy, precision, recall and FSCORE. Among all algorithms Random Forest is giving high accuracy so we employed same model for test data prediction.

To train and test above algorithms we have used same dataset given by you and in below screens showing JUPYTER code and output with blue colour comments

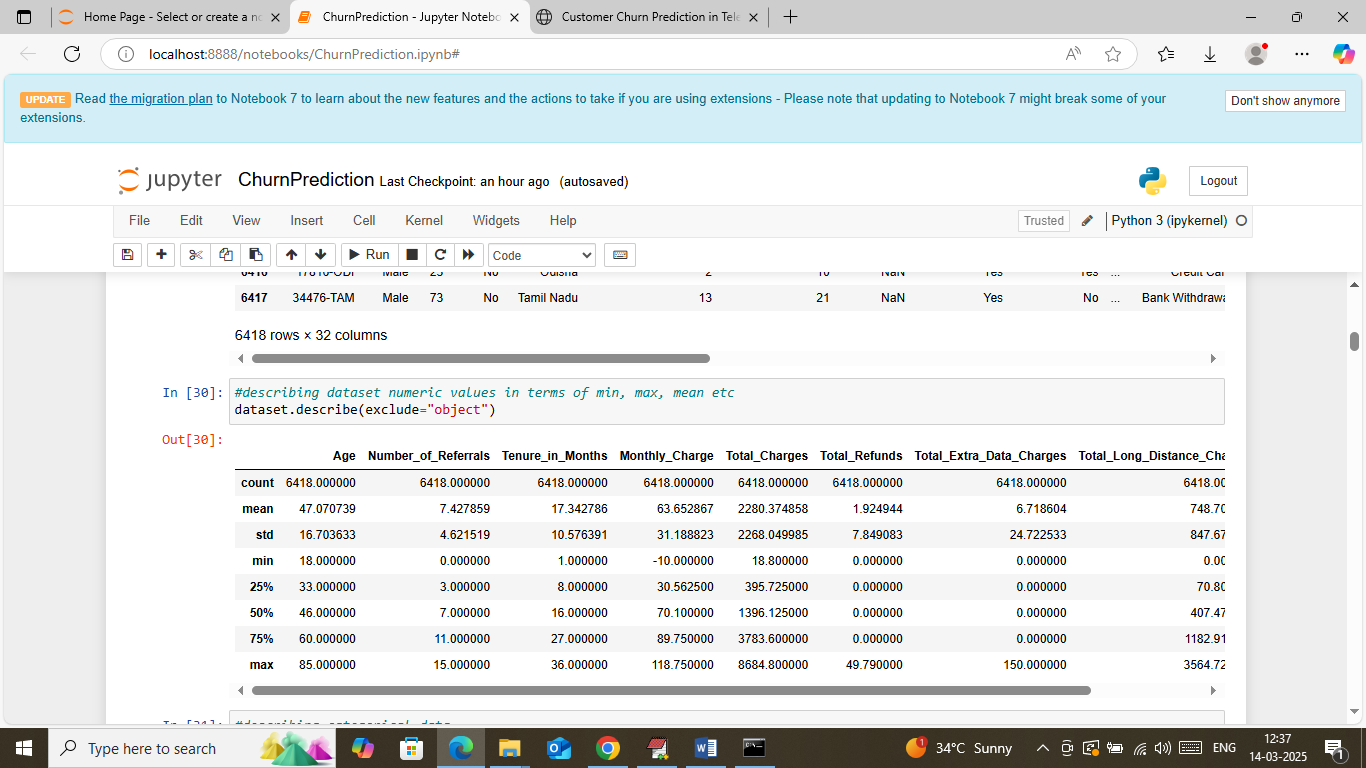
SCREEN SHOTS



In above screen loading and importing python classes and packages



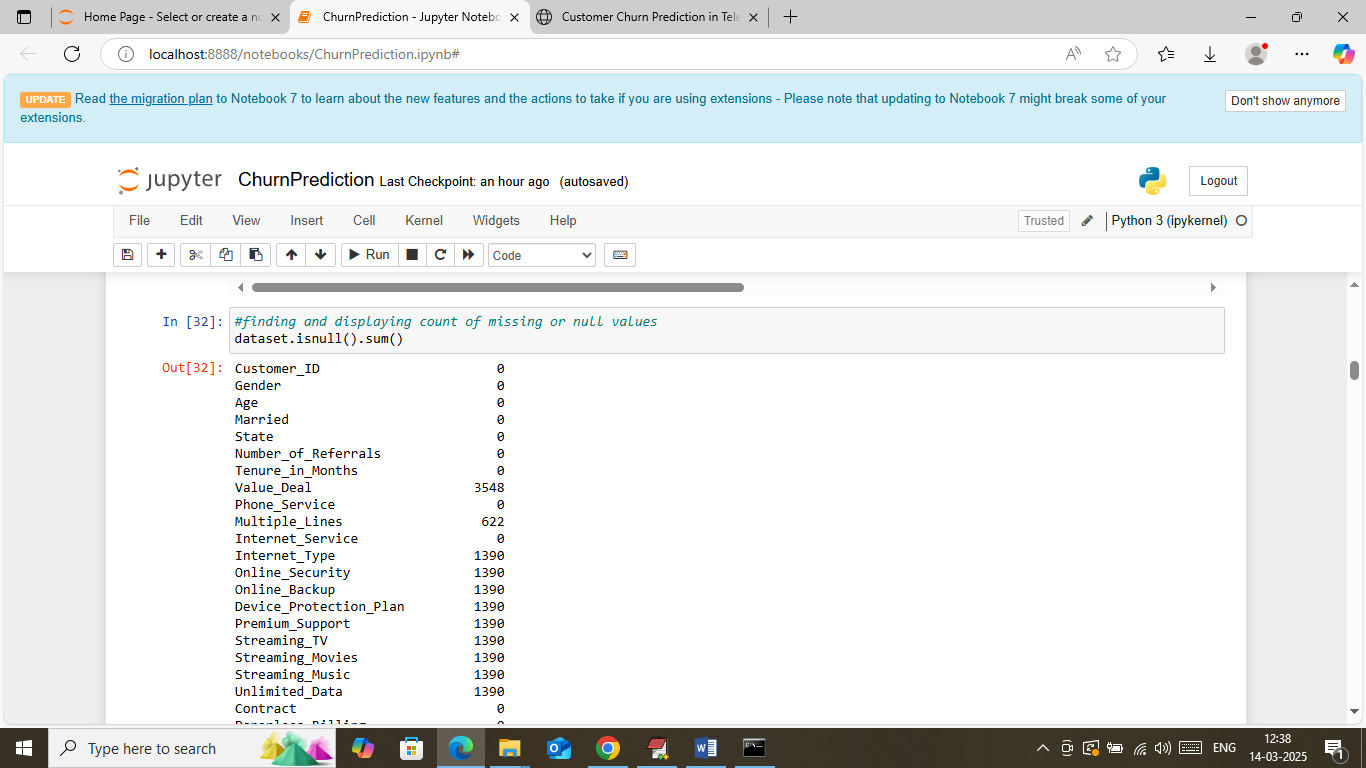
In above screen loading and displaying dataset values



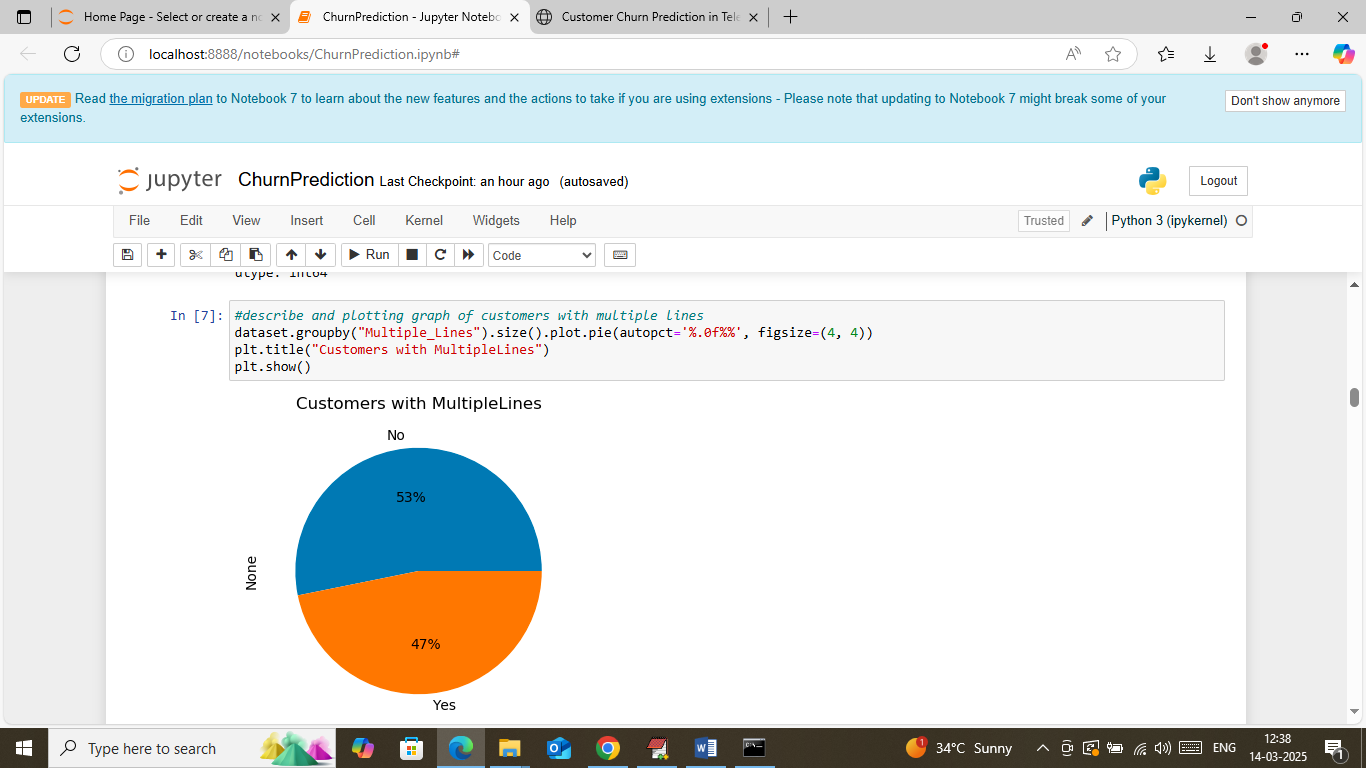
In above screen describing numeric column values in terms of min, max, standard deviation and other metrics



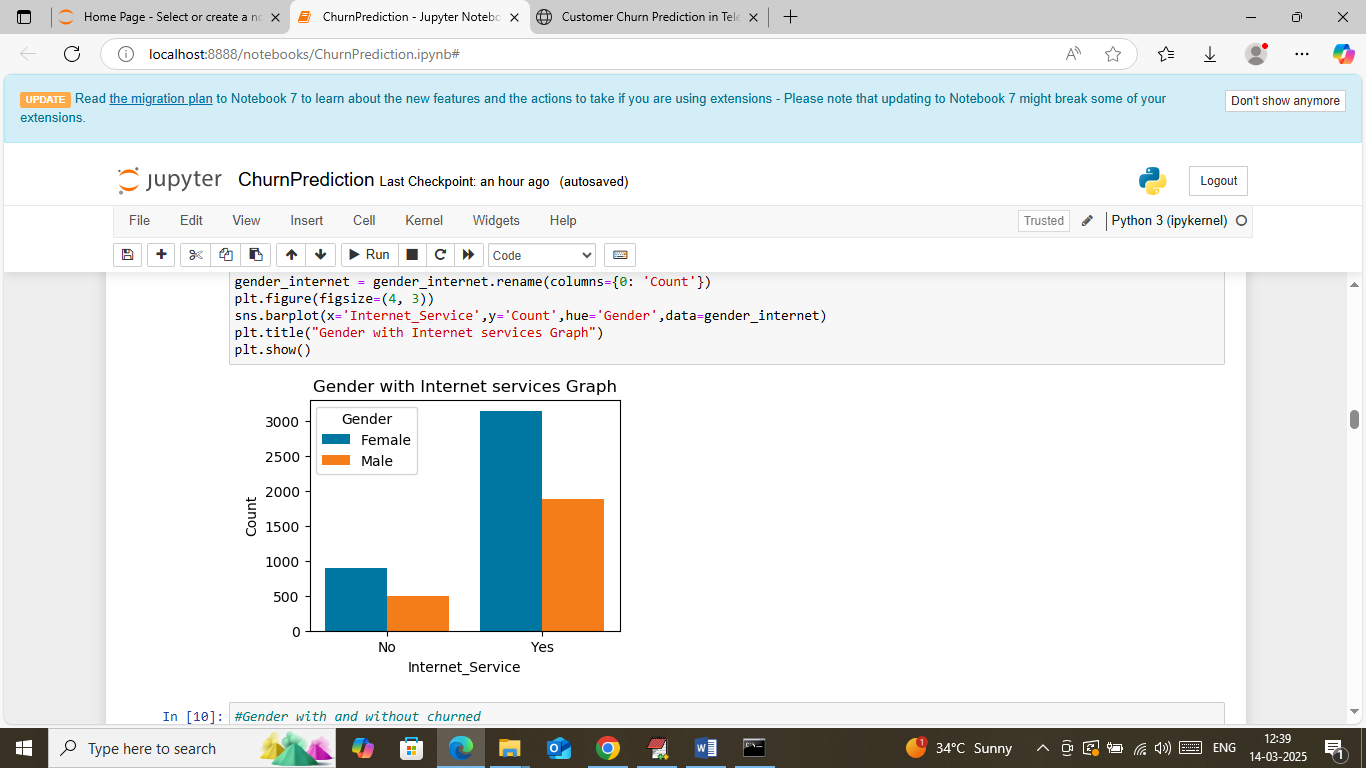
In above screen displaying min, max and other metrics values in terms of categorical data



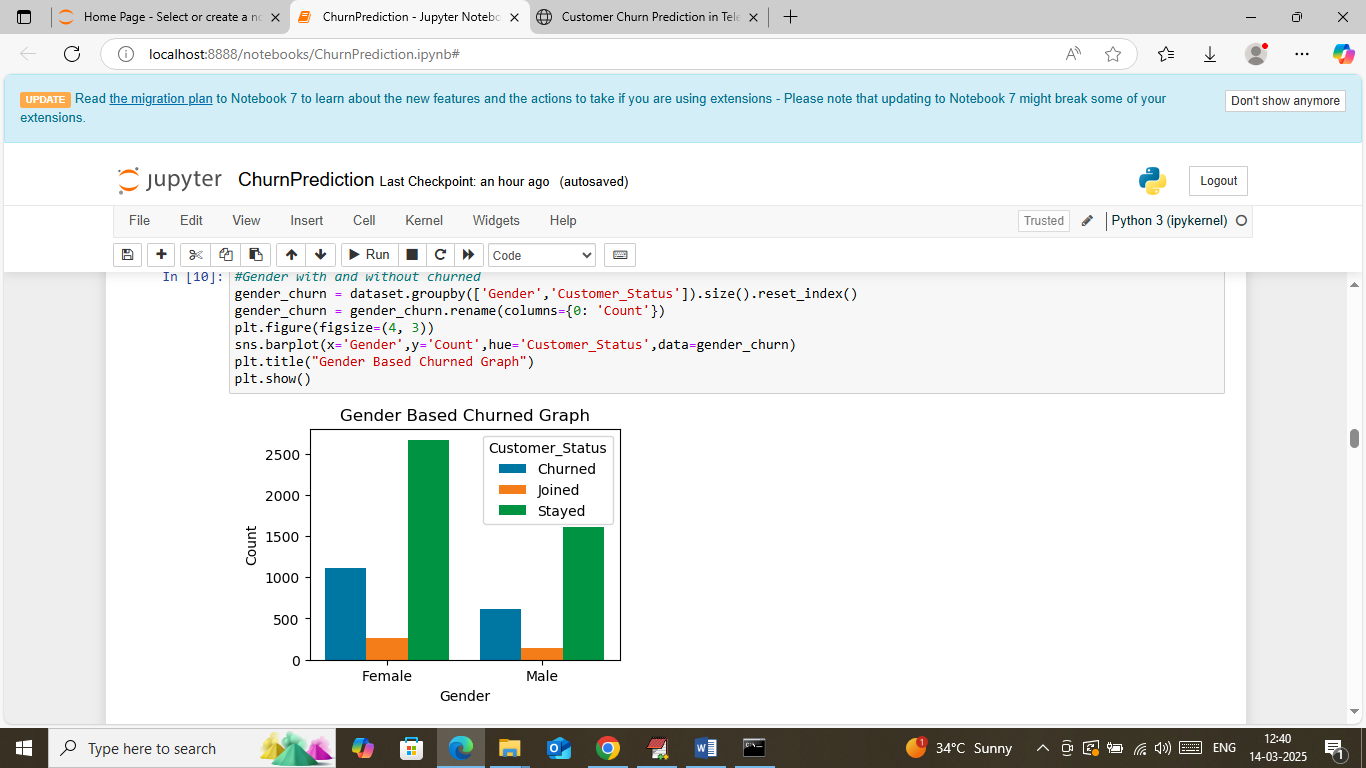
In above screen finding and displaying count of missing values



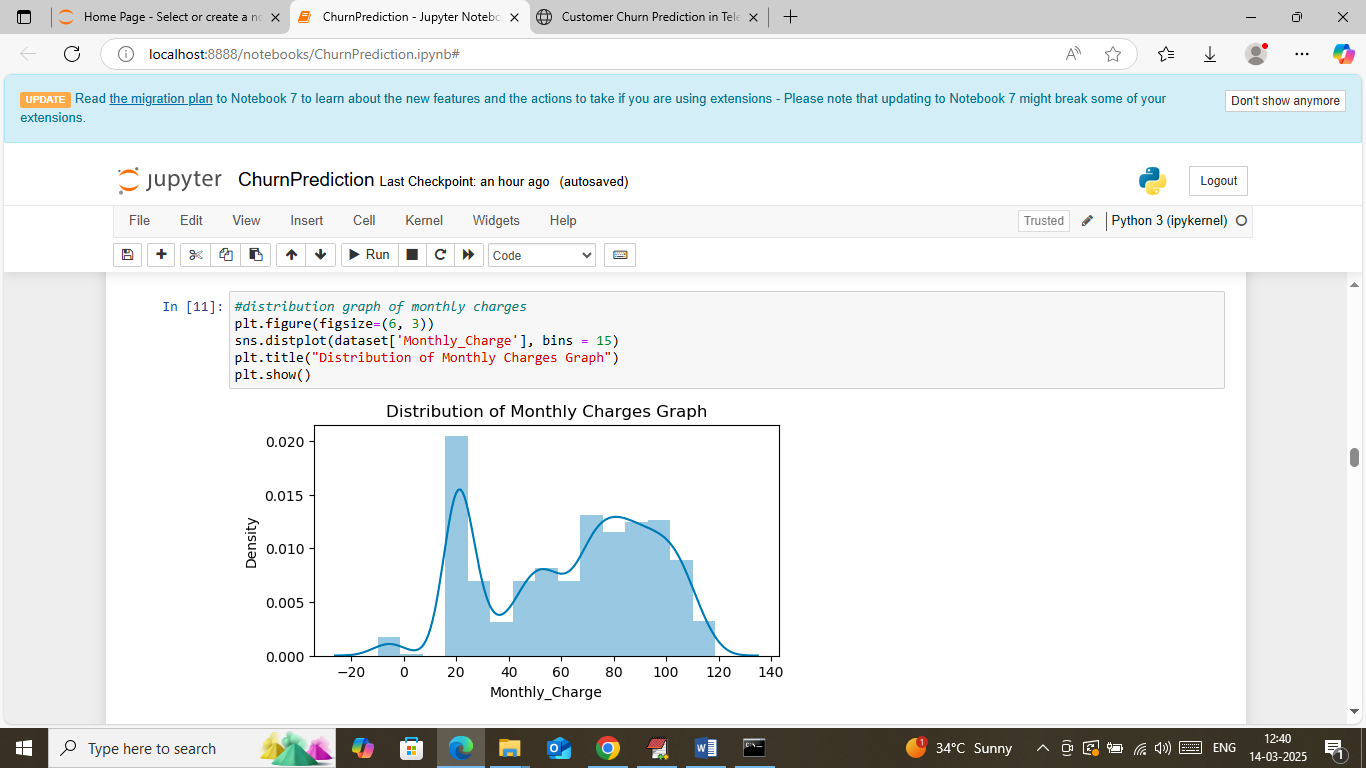
In above screen visualizing graph of customers with multiple connections



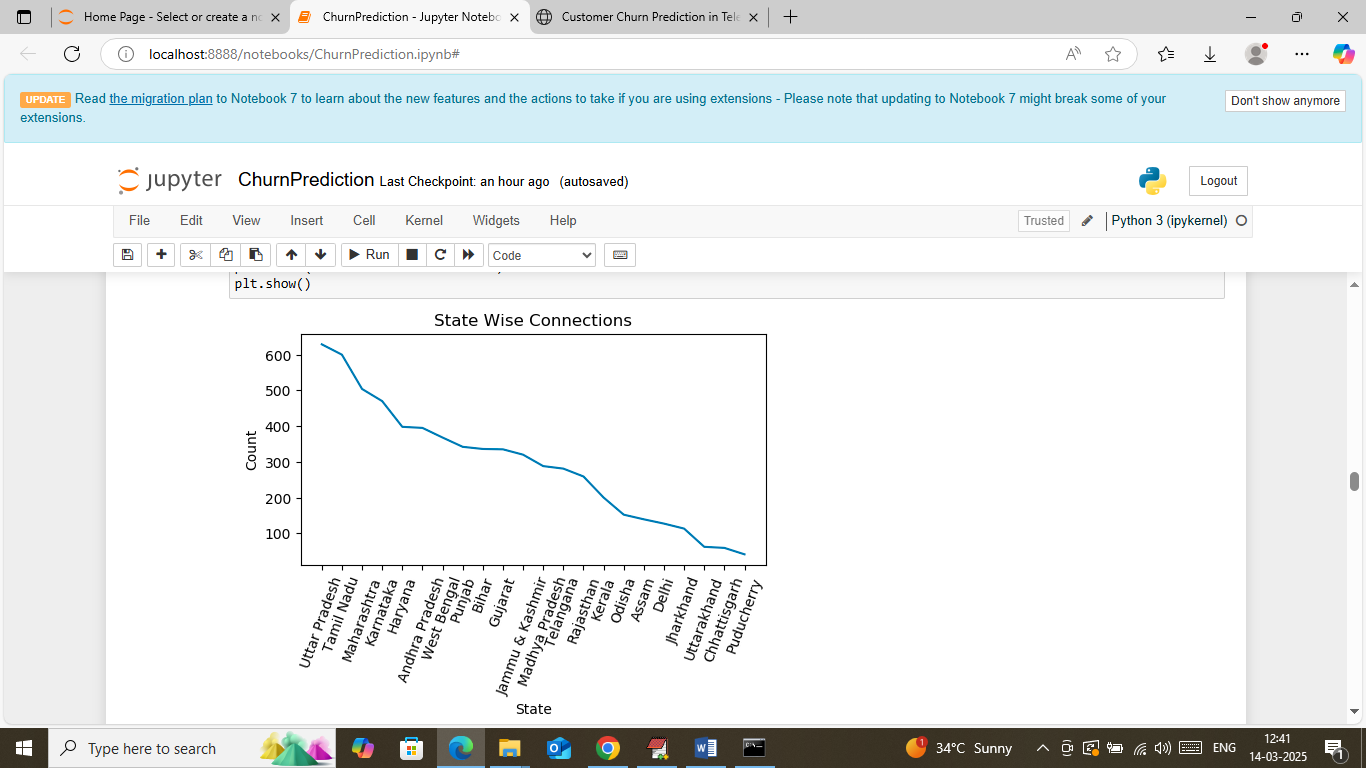
In above screen visualizing gender based graph who has connection with and without internet services



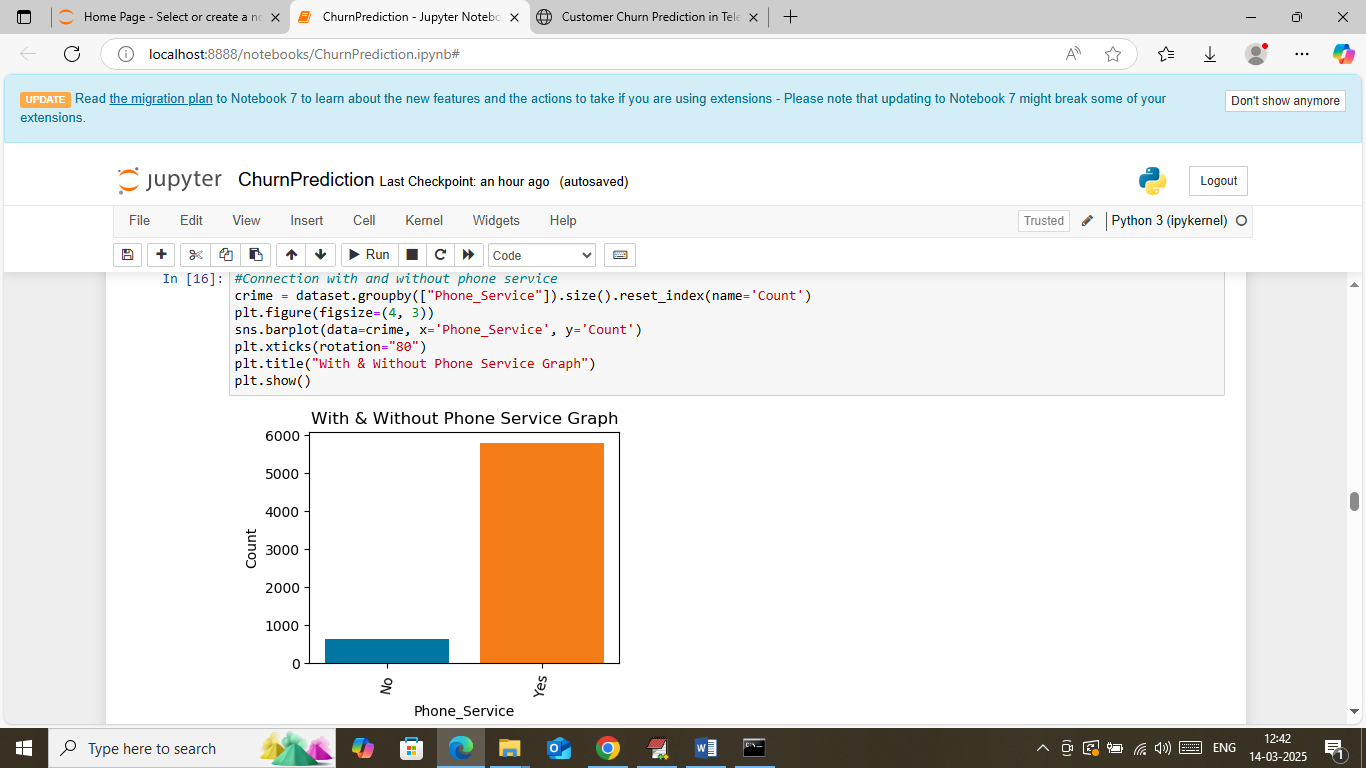
In above screen visualizing gender based customer with different labels such as Churned, Stayed or Joined



In above screen visualizing monthly distribution charges



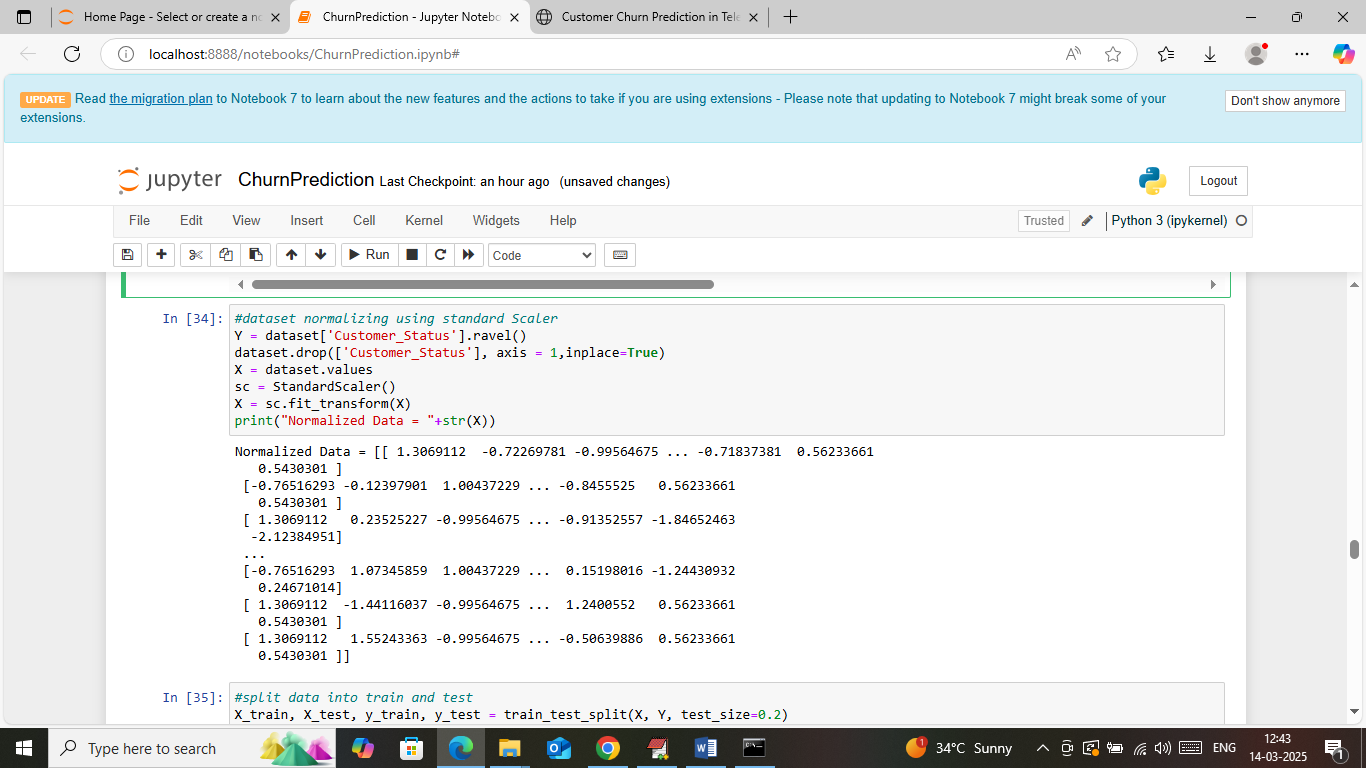
In above screen visualizing geography wise number of connections in descending order where x-axis represents State Name and y-axis represents number of connections



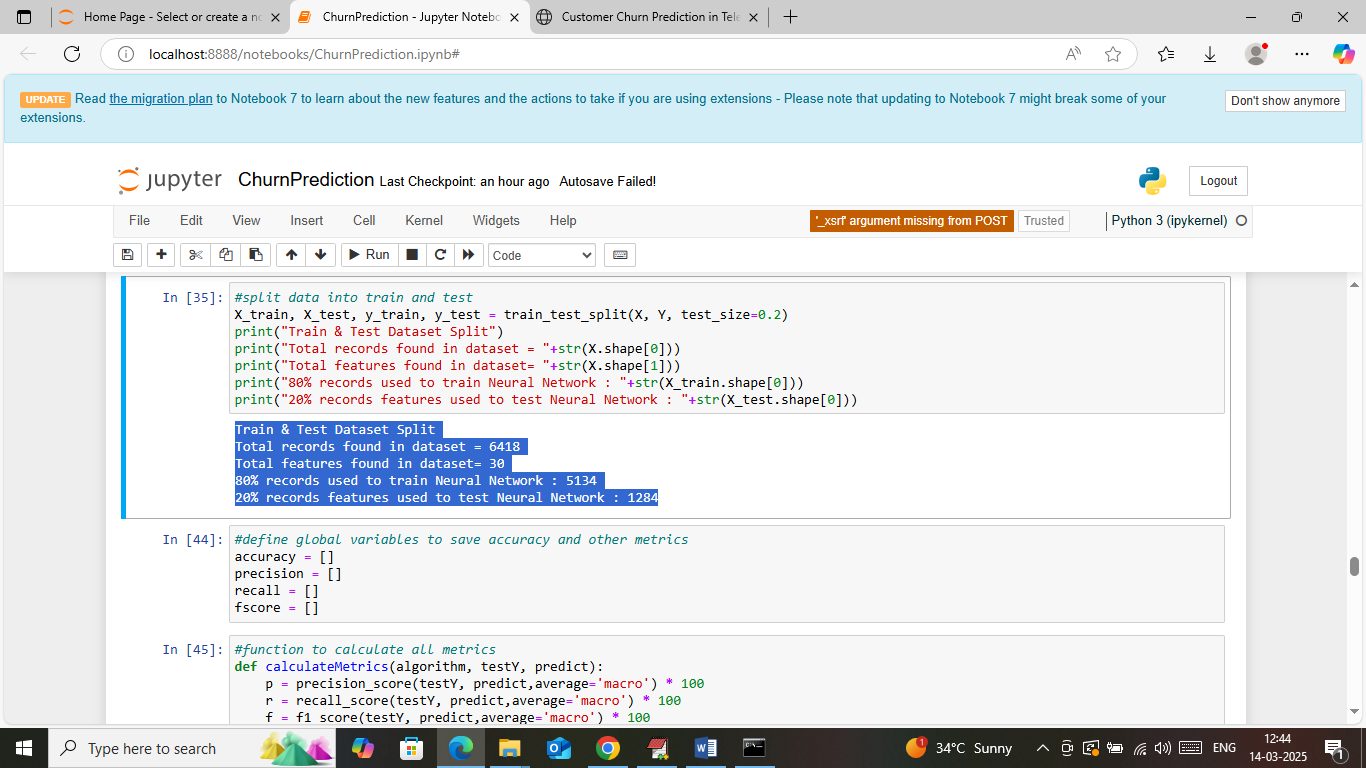
In above screen visualizing graph of number of connection with and without phone service



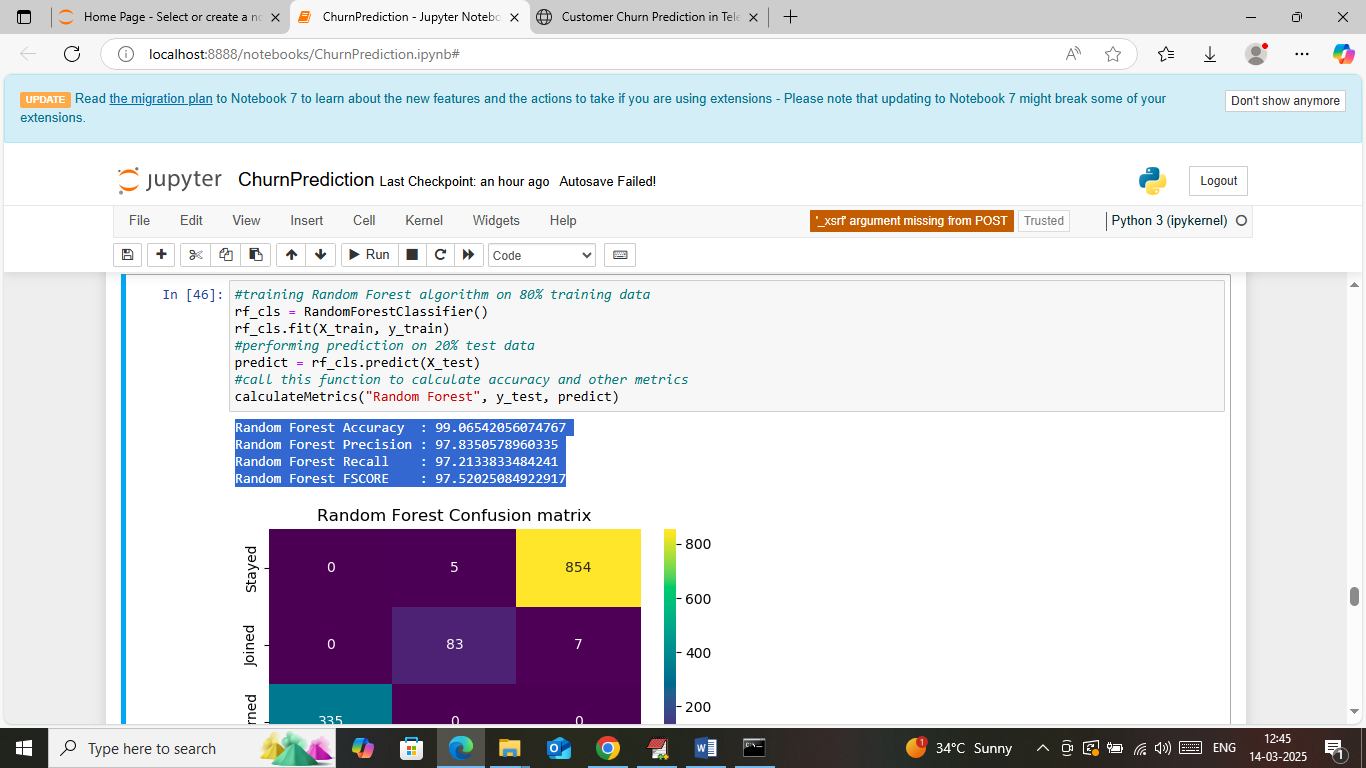
In above screen applying dataset processing to convert all non-numeric data to numeric data and then replacing all missing values with mean and then displaying cleaned and processed data



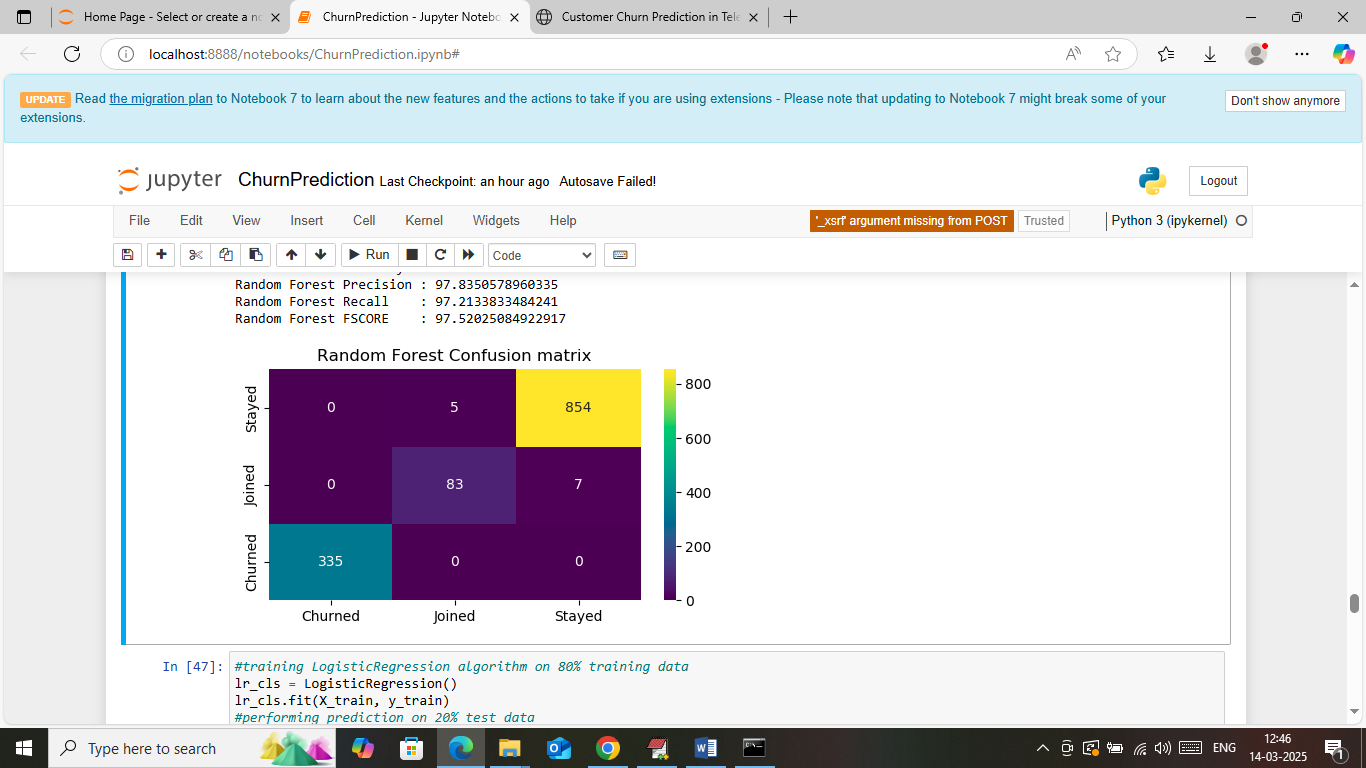
In above screen applying standard scaling function to normalized dataset values



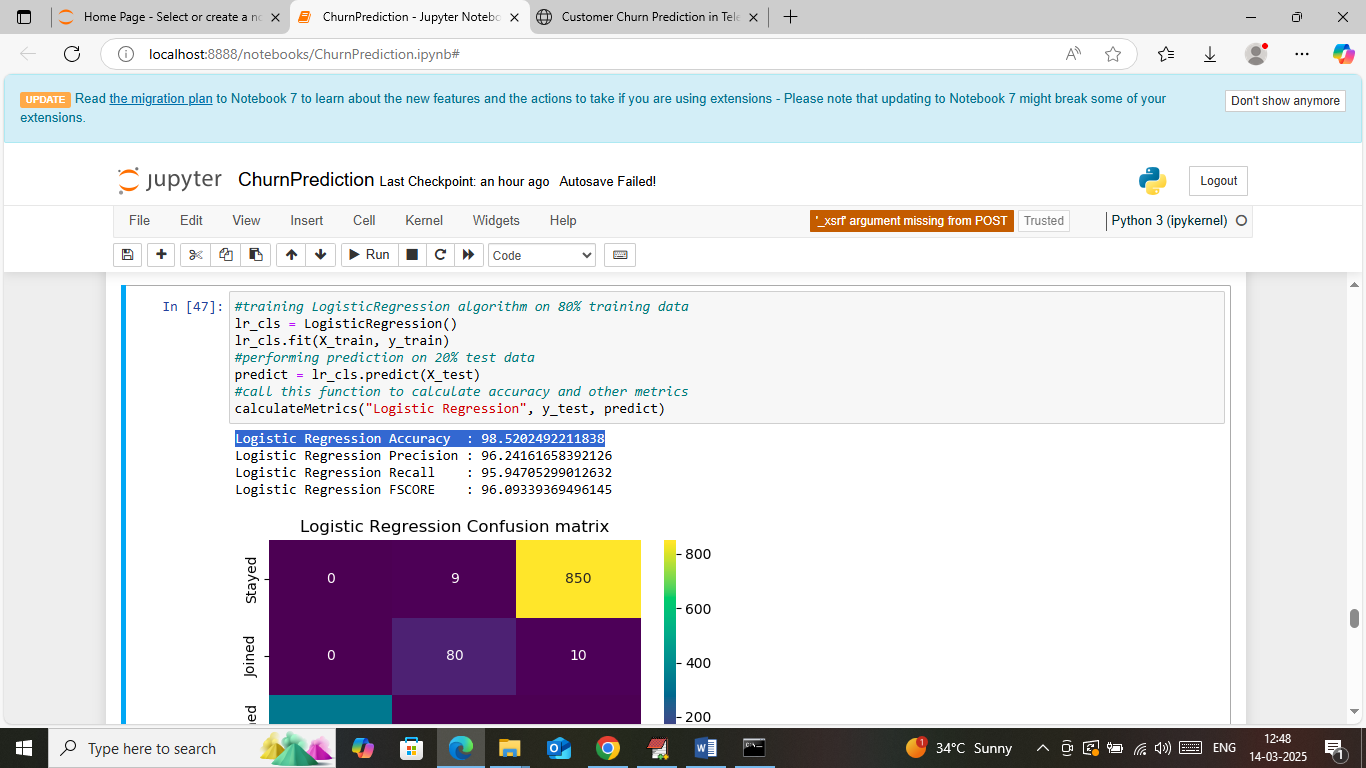
In above screen splitting dataset into train and test where application using 80% dataset for training and 20% for testing and then can see train and test size. In next blocks defining function to calculate accuracy and other metrics



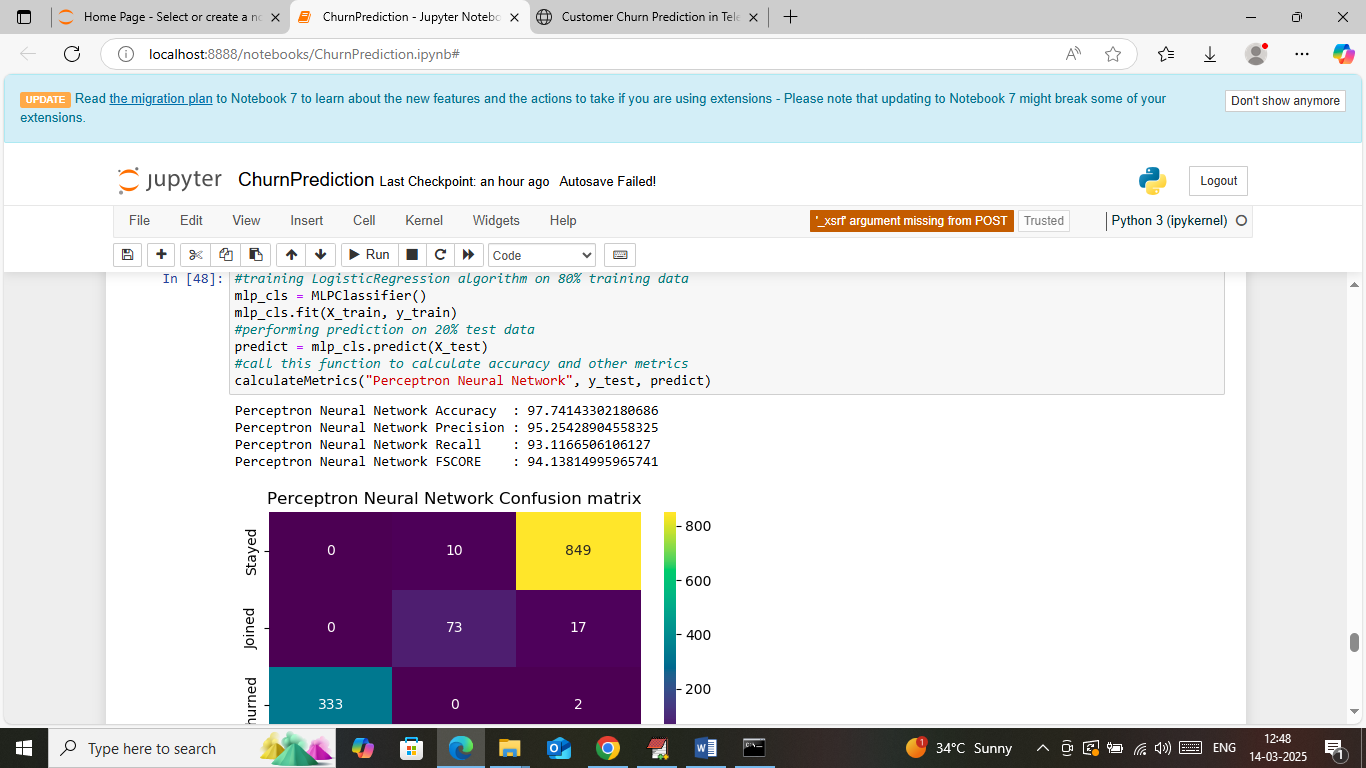
In above screen 80% training data input to random forest algorithm to train a model and this model applied on 20% test data to calculate prediction accuracy. In above screen Random Forest got 99% accuracy and can see other metrics like precision, recall and FSCORE. Below is the confusion matrix classification graph



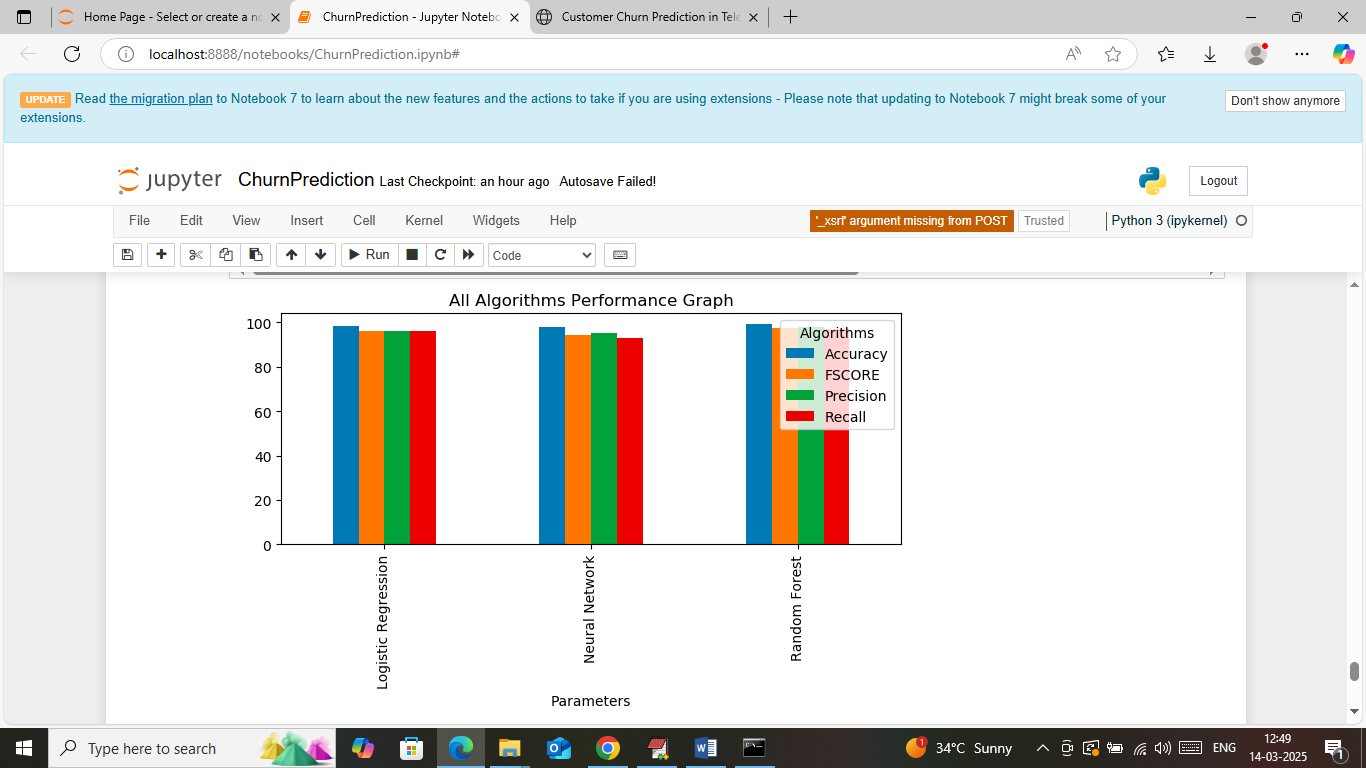
In above Random Forest confusion matrix graph x-axis represents Predicted Labels and y-axis represents True Labels and then all different colour boxes in diagonal represents correct prediction count and remaining boxes represents incorrect prediction count which are very few.



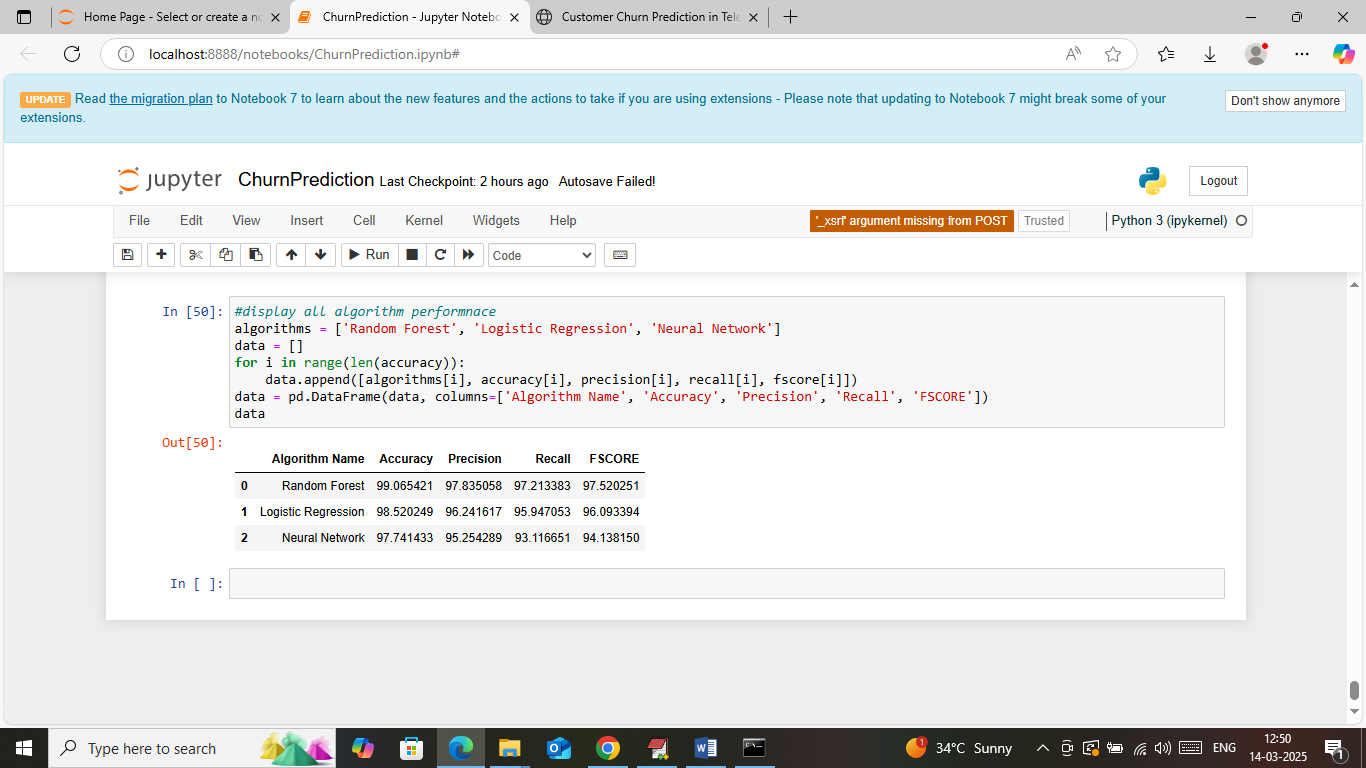
In above screen Logistic Regression got 98% accuracy and can see other metrics also



In above screen MLP neural network got 97% accuracy

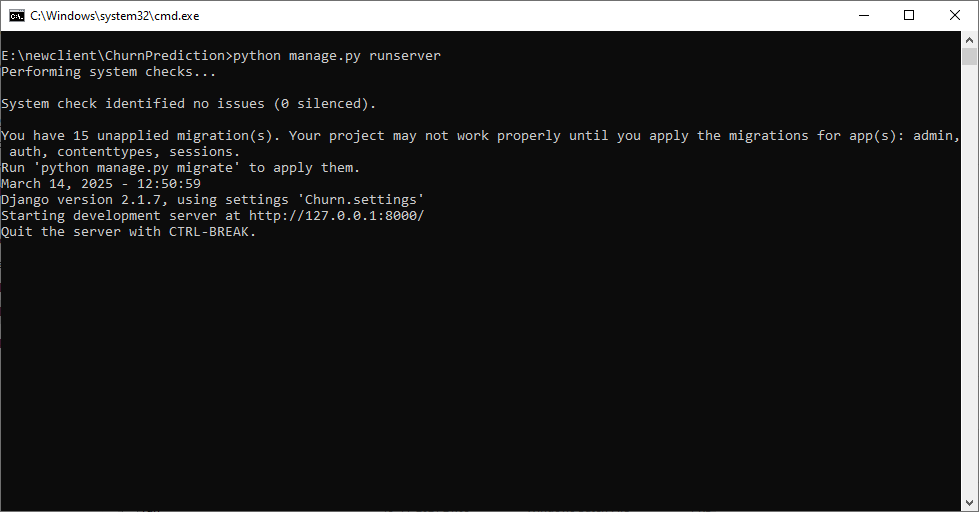


In above screen visualizing comparison graph between all algorithms where x-axis represents algorithm names and y-axis represents accuracy and other metrics in different colour bars

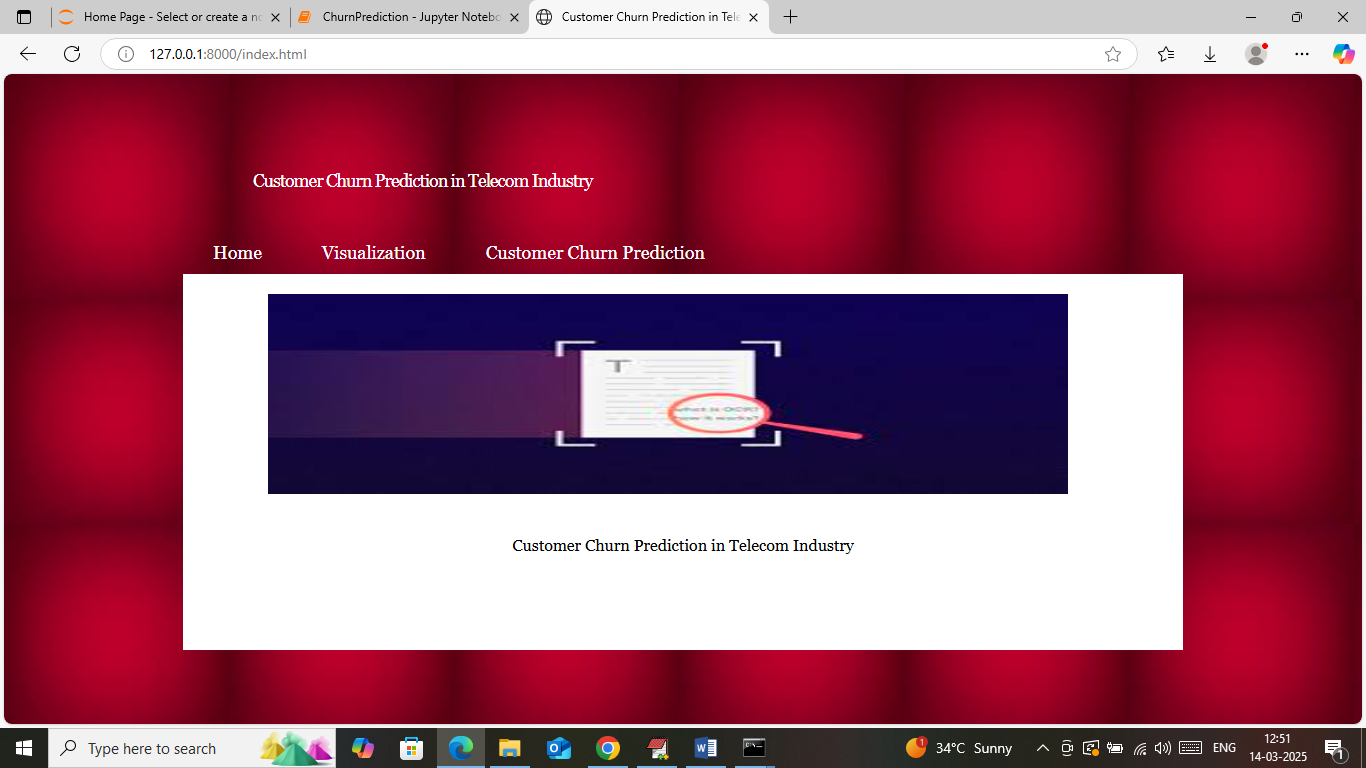


In above screen displaying all algorithms performance in tabular format.

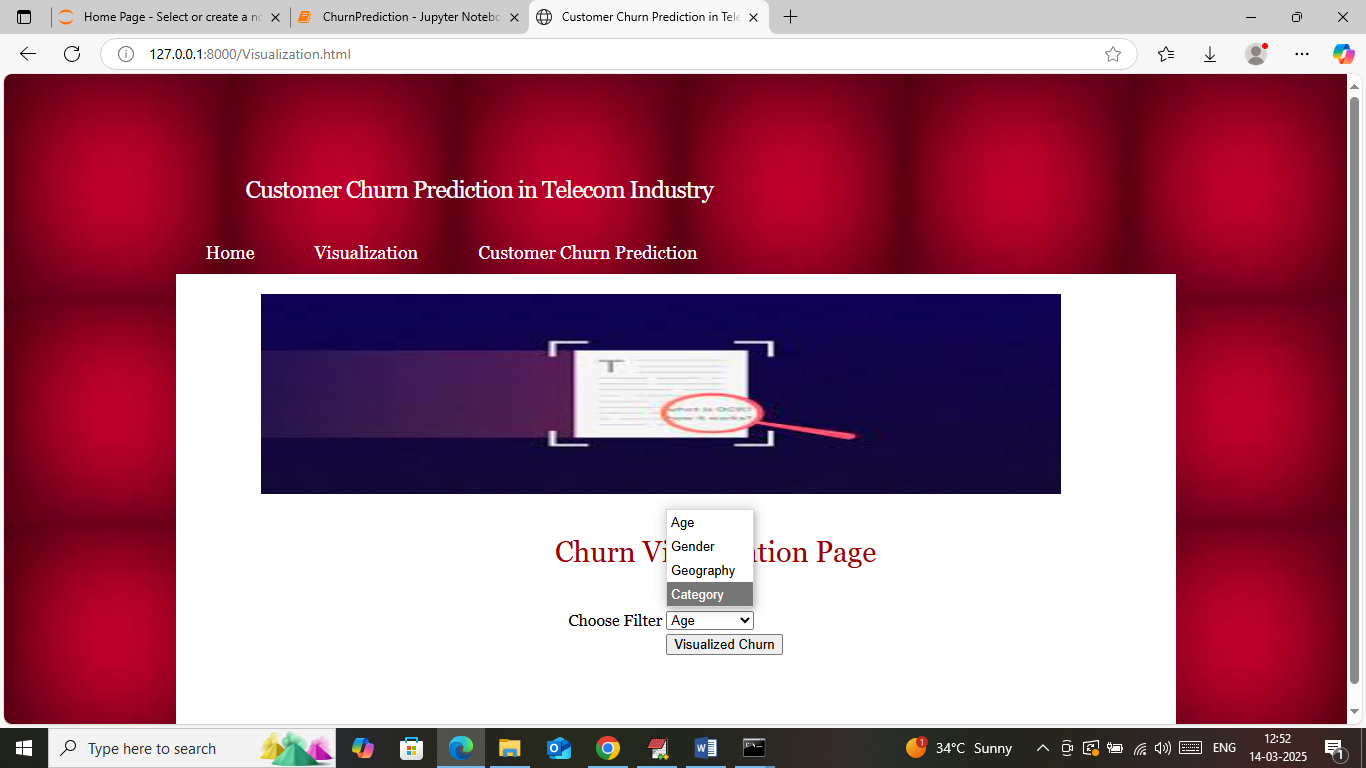
In below screen showing WEB based output and to run web output double click on ‘runWebServer.bat’ file to get below page



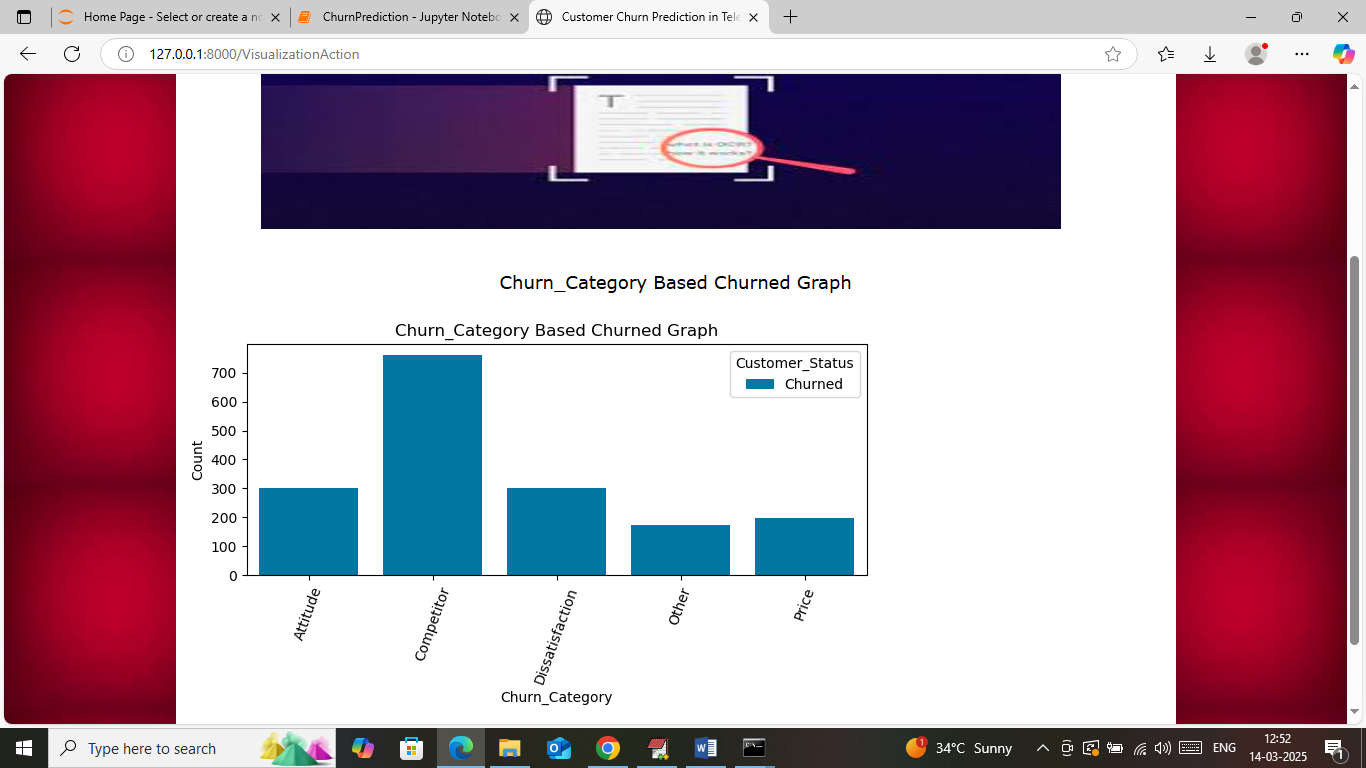
In above screen python server started and now open browser and enter URL as <http://127.0.0.1:8000/index.html> and then press enter key to get below page



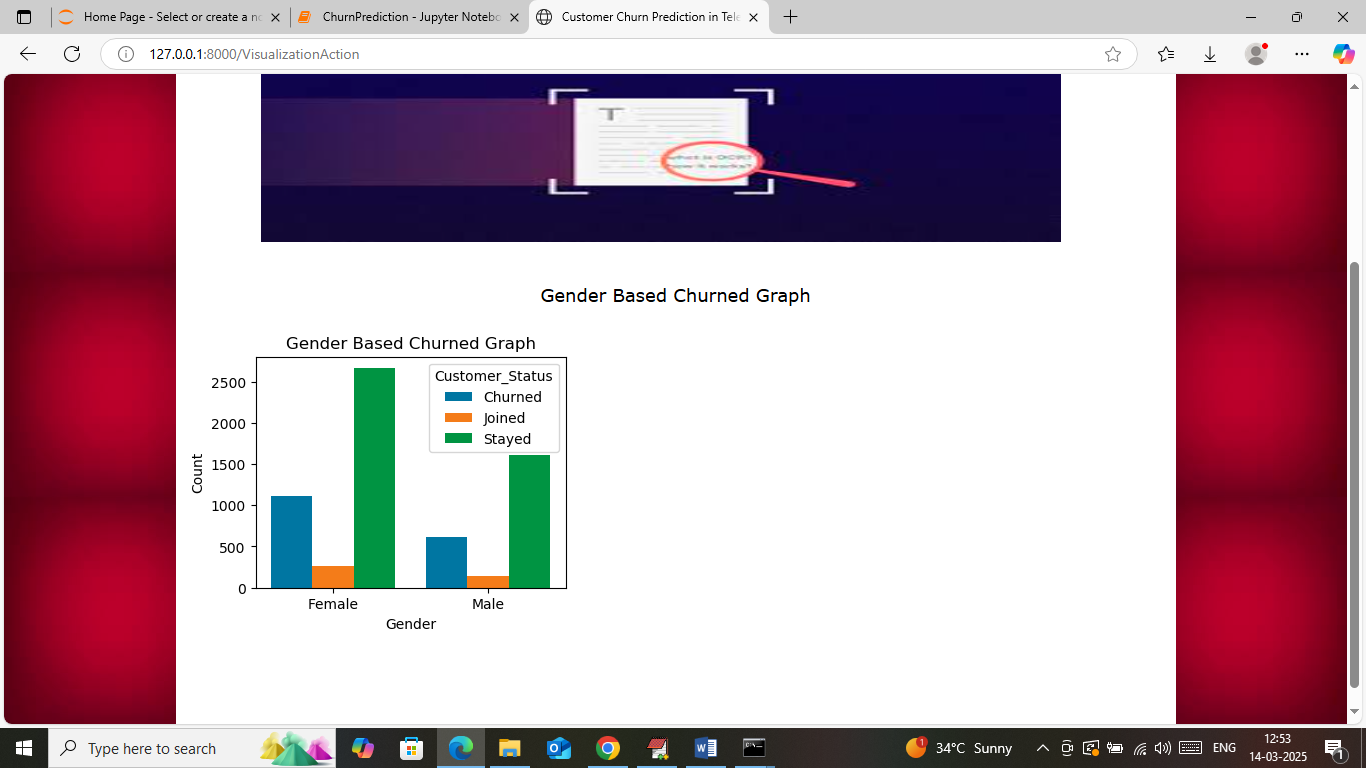
In above screen click on ‘Visualization’ link to get below page



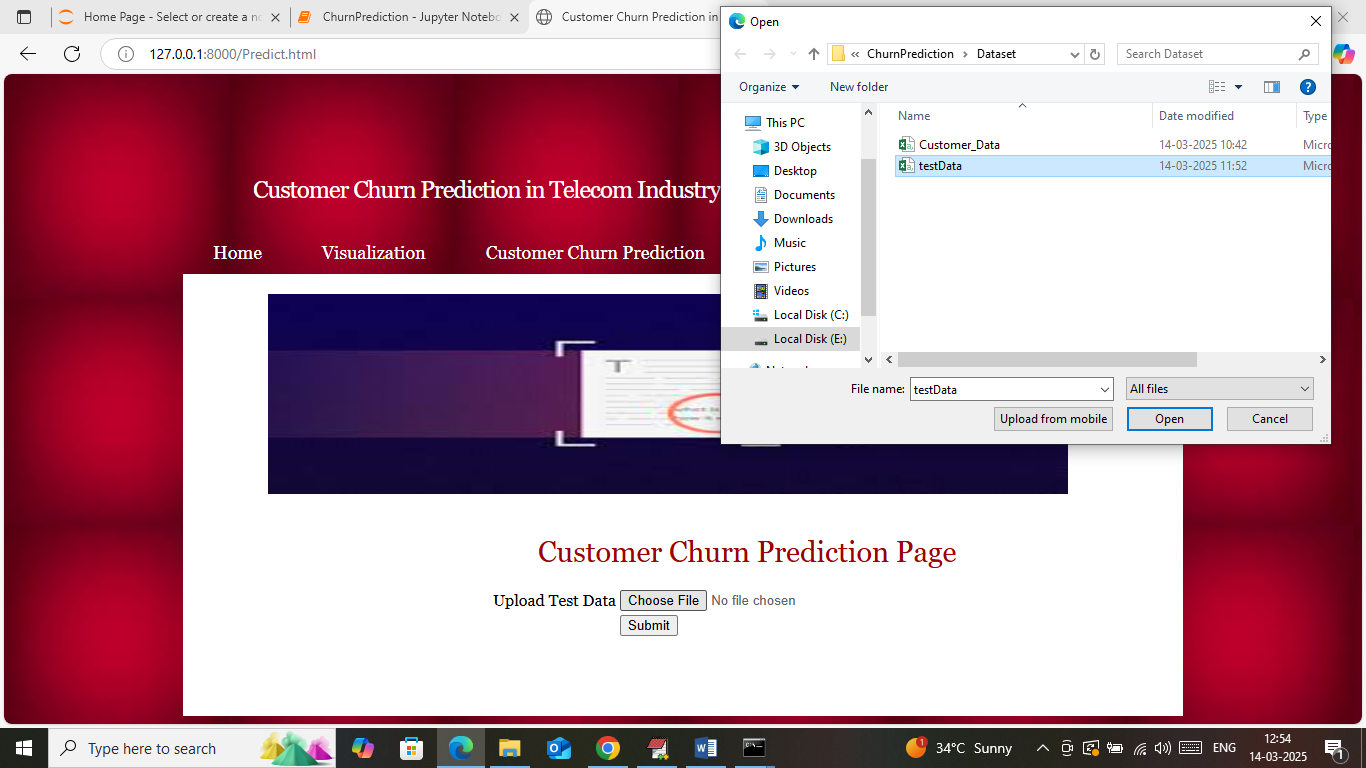
In above screen select any filter and then press button to get below graph



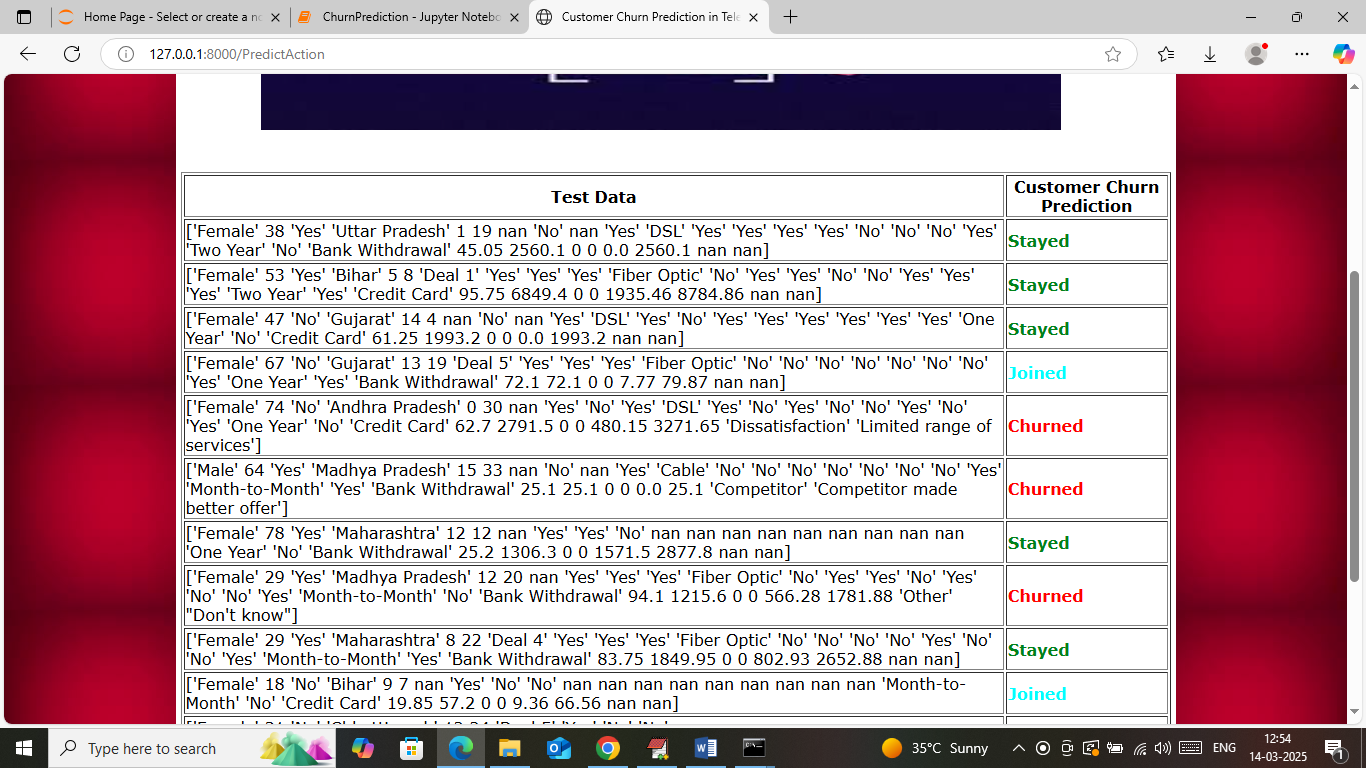
In above graph x-axis represents category and y-axis represents churned and similarly you can select different filter to get different graph



In above screen can see gender wise churn type and now click on ‘Churned Prediction’ link to get below page



In above screen selecting and uploading test data samples and then press button to get below page



In above screen in first column can see Test Data values and in second column can see predicted values as ‘Stayed, Joined or Churned’ showing in different colour.

So above are the JUPYTER and WEB output from your project