Short-Term Arrival Delay Time Prediction in Freight Rail Operations Using Data-Driven Models

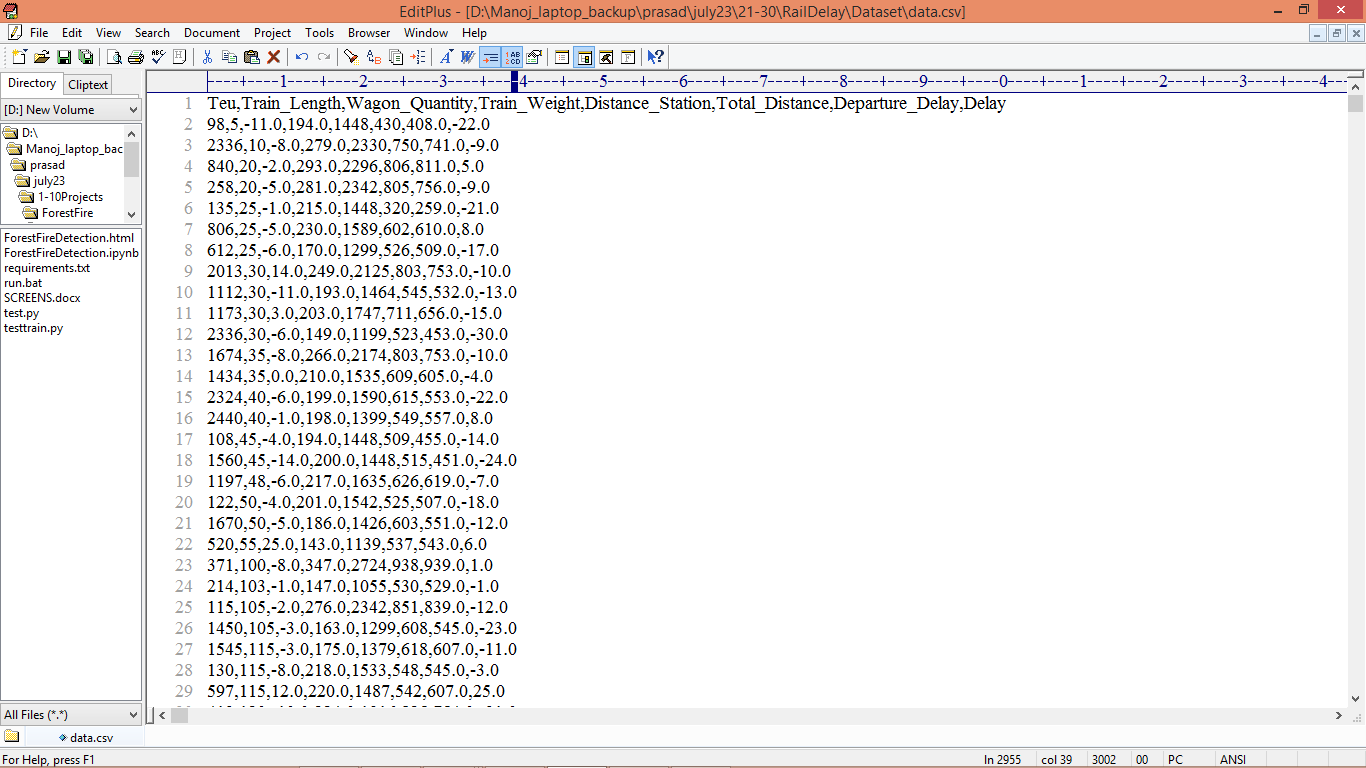
Freight Rail means good transportation using train or any other vehicle from one place to other place. All daily usage goods will manufacture in different places and need to shift from one place to other place and biggest available model of transportation is Rail or Train. While transportation operators must have accurate information on travel and departure time to know about arrival time. Earlier arrival time was calculated using distance and travel time but this calculation process is not accurate. Later machine learning models was introduced but they lack of Short Term Arrival delay. Short term arrival delay will give us accurate arrival time as this short term time will be calculated on each station which helps in knowing arrival time of next station. Short term will be subtracted from Actual arrival, scheduled arrival, schedule departure and departure delay.

Author calculating short term arrival from live dataset obtained from ‘National Rail Company of Luxembourg’ and then used this dataset to train with various machine learning algorithms such as LIGHTGBM, Random Forest, KNN and Linear Regression and each algorithm performance is evaluated in terms of RMSE, MAPE, MAE and R2. RMSE, MAPE and MAE refers to difference between original values and predicted values so the lower the difference the better is the model. R2 will be considering as accuracy of the model so the higher the R2 the better is the model. In propose work LIGHTGBM is giving high R2.

Apart from Short Term prediction author applying SHAPLEY technique for model explanation which will explain about what features help model in getting high R2 score.

Author also applying various dataset processing techniques such as features selection using CORRELATION and then finding features importance using LIGHTGBM algorithm. Other processing techniques are removing missing values and features normalization.

To train all algorithms author has not publish dataset so we downloaded available data from KAGGLE and below are the dataset details



In above dataset screen first row contains dataset column names and remaining rows contains dataset values and in last column we have Short Time Delay as target value which is in minutes from junction to junction. So by using above dataset we will train and test all algorithm performance.

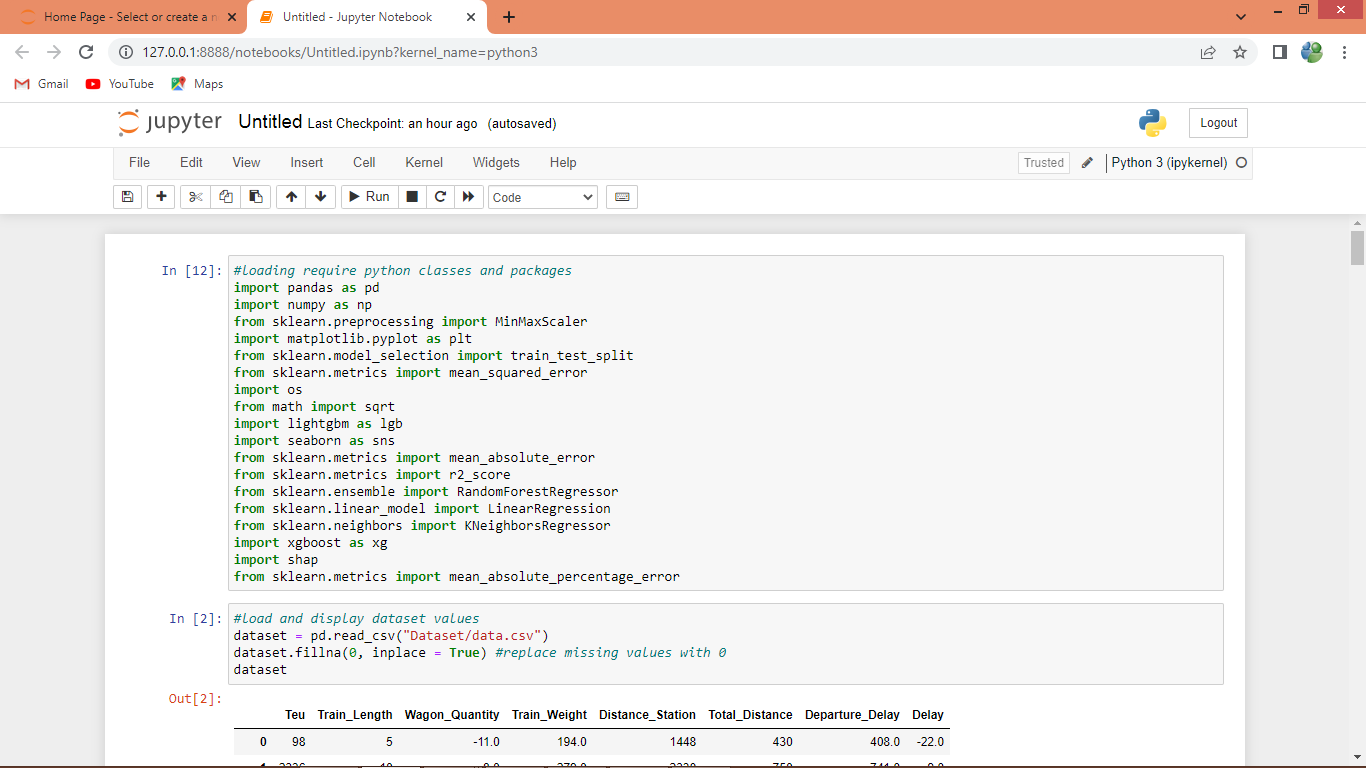
Extension Concept

In propose work author has used all traditional algorithms like Random Forest, KNN, Linear Regression and LIGHTGBM so as extension we have experimented with advance XGBOOST algorithm which will optimize features using 100 estimators or group of forest and this optimizations help XGBOOST in getting enhanced R2 score.

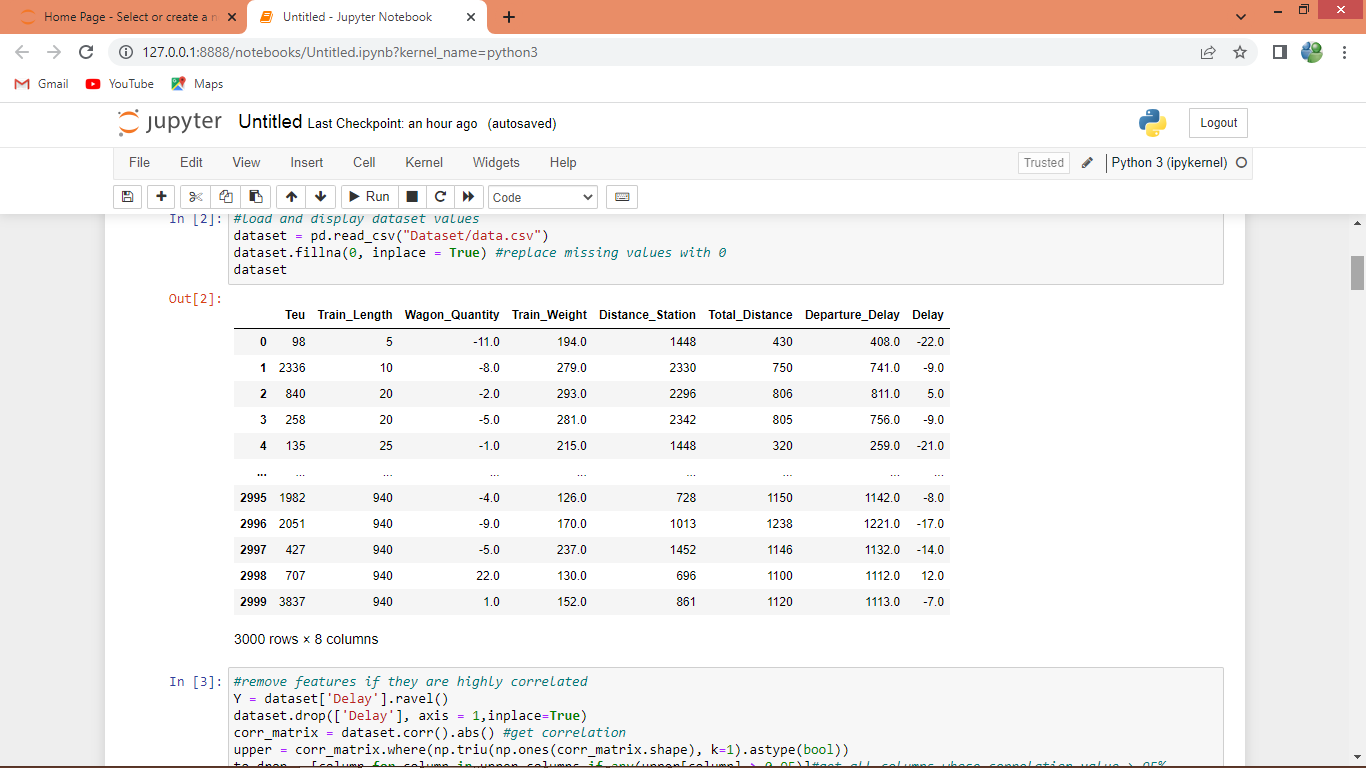
Note: from base paper our results will be vary from 2 to 15% as author has not publish dataset so we trained algorithms on custom dataset obtained from KAGGLE.

SCREEN SHOTS

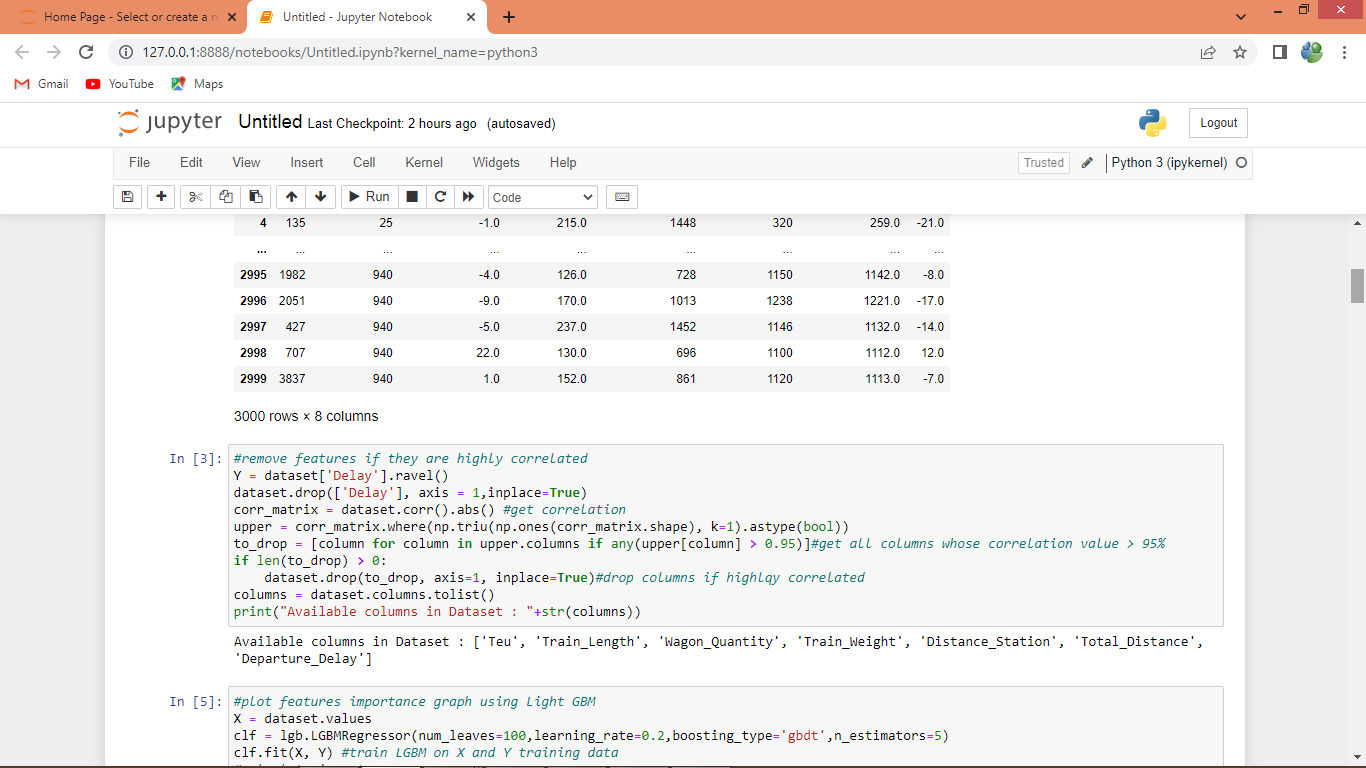
We have coded this project using JUYTER notebook and below are the code and output screens with blue colour comments



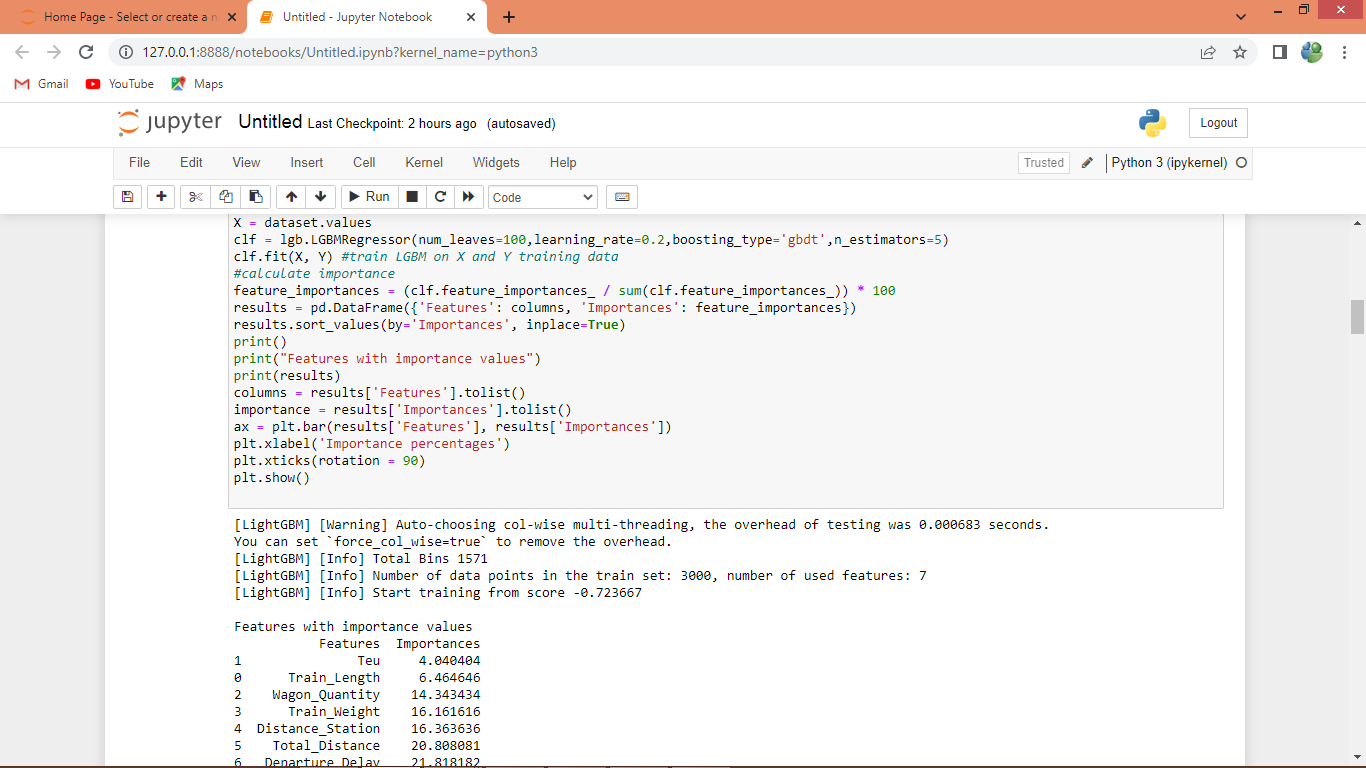
In above screen importing require python classes and packages



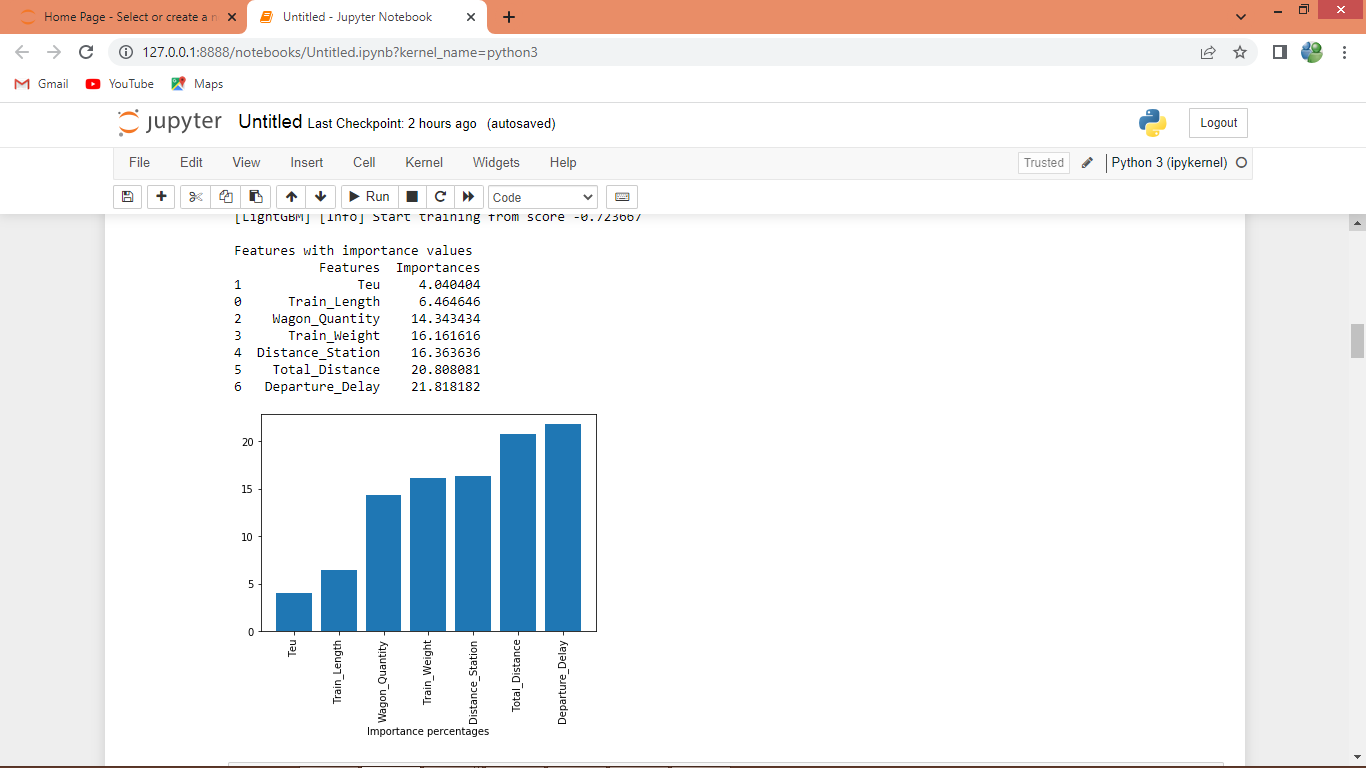
In above screen reading and displaying dataset values



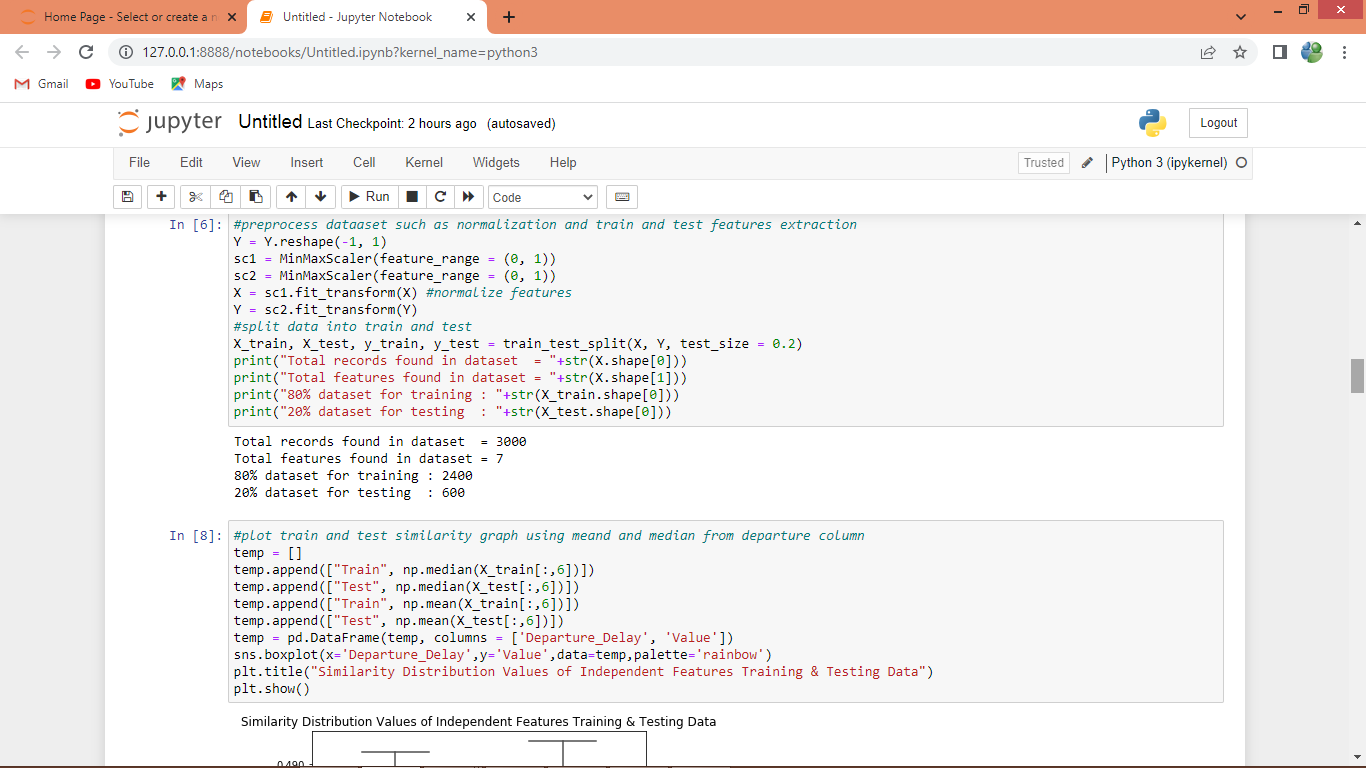
In above screen finding correlation for each column and then removing those columns which have high score or highly correlated



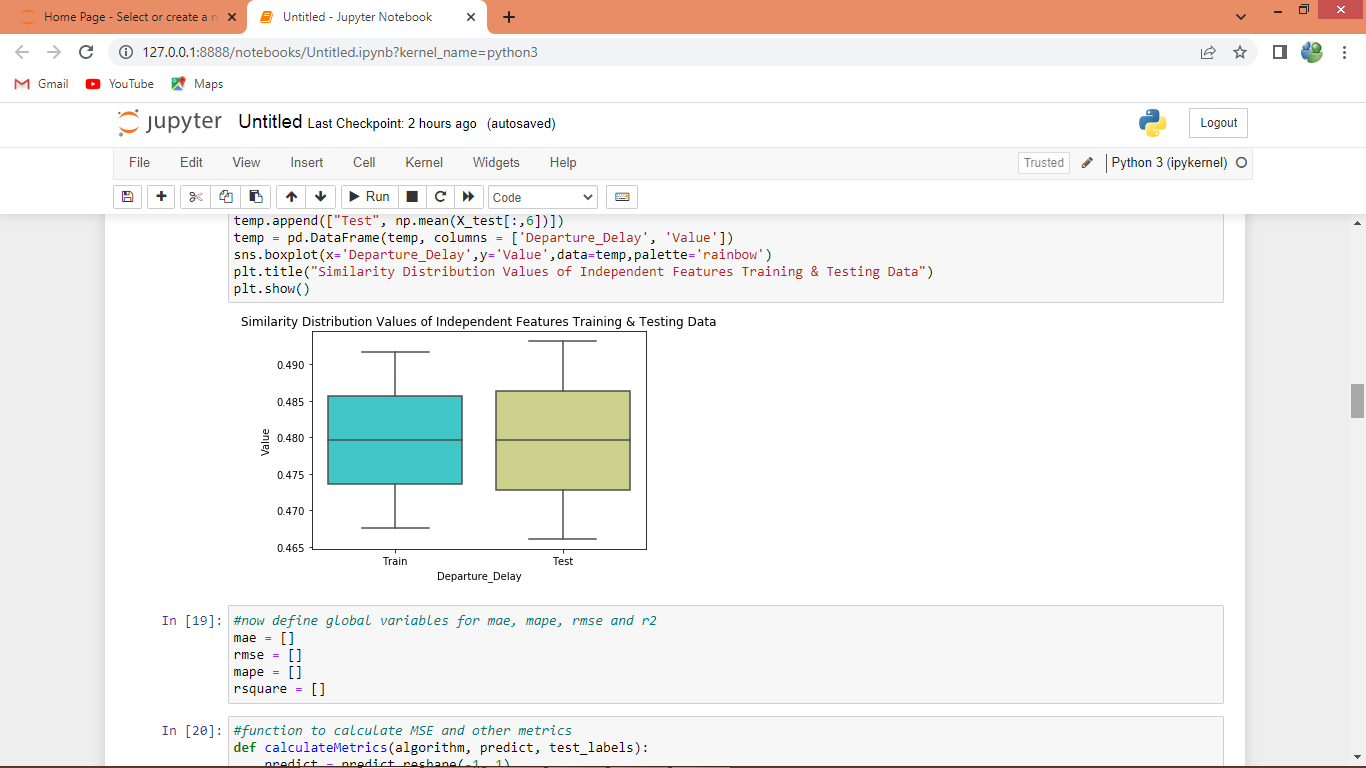
In above screen using LIGHTGBM finding importance of each features or columns and after executing above block will get below output



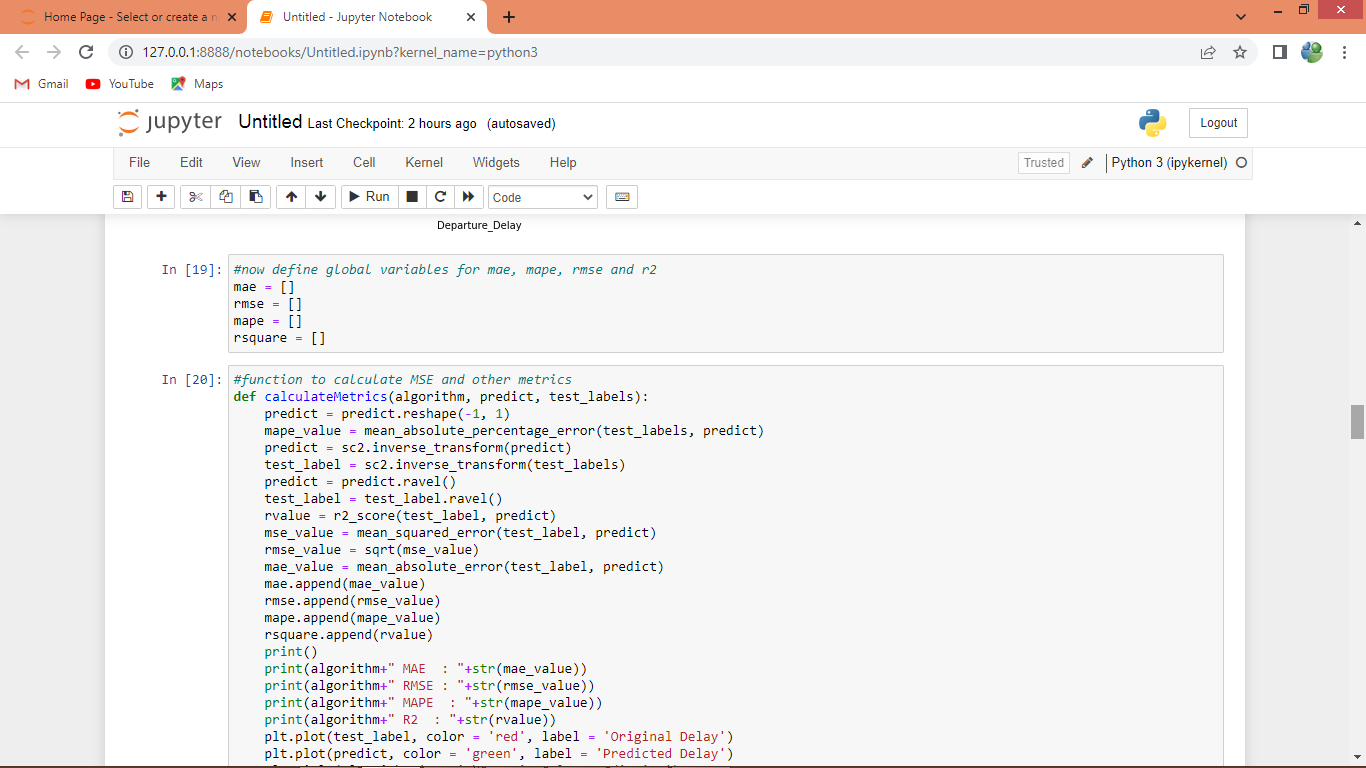
In above screen displaying feature names and its importance values and then plotting same in graph where x-axis represents feature name and y-axis represents importance values



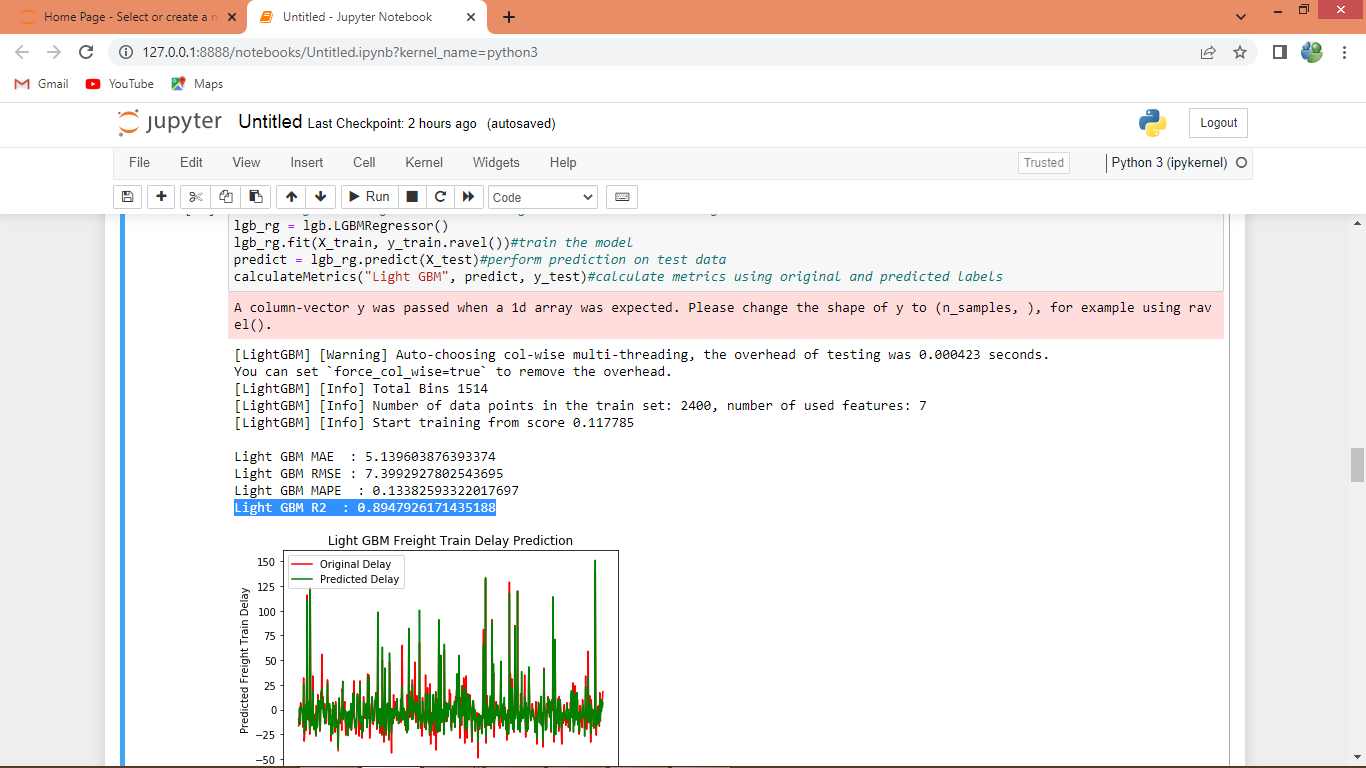
In above screen applying dataset pre-processing techniques such as X and Y features extraction, normalization and splitting dataset into train and test and then defining code to find similarity between train and test features and below is the similarity graph



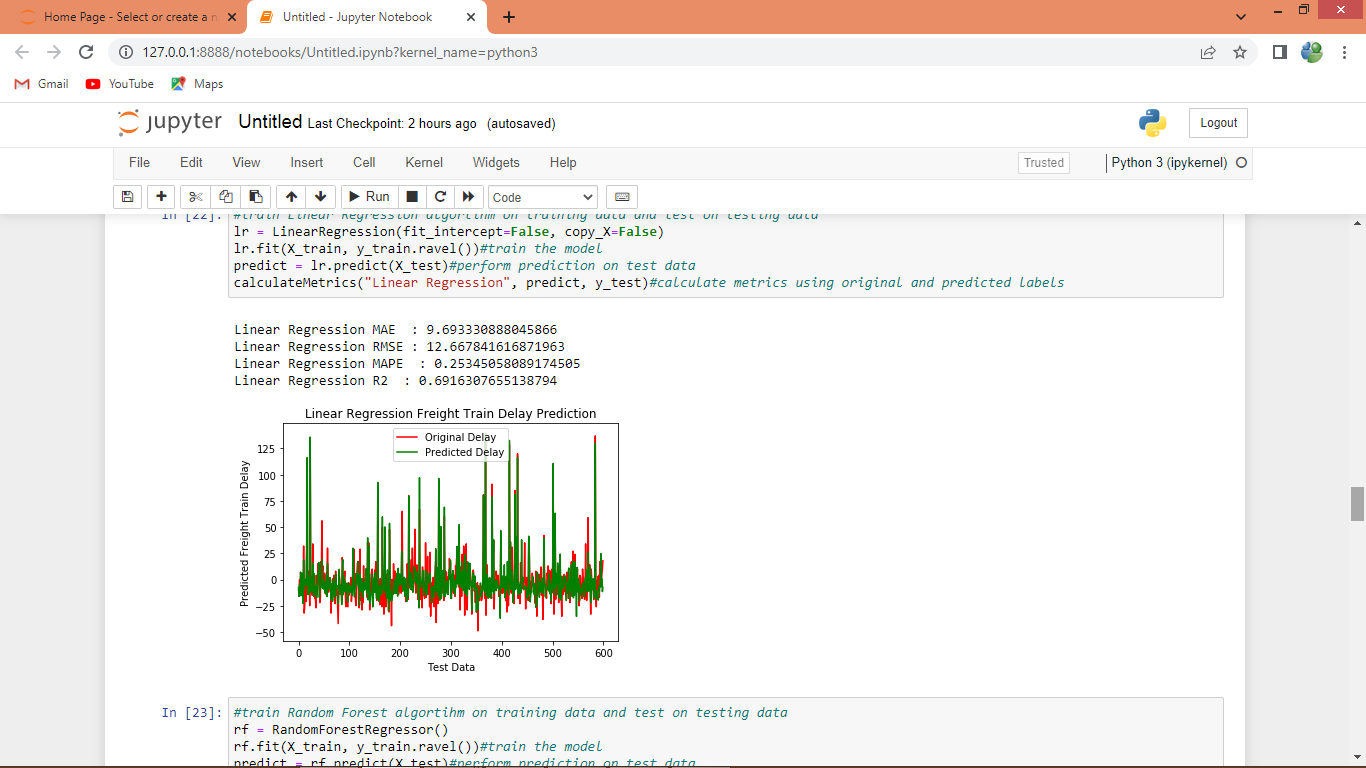
In above graph x-axis represents train and test data and y-axis represents similarity between them and we can see both are having very close similar data and both boxes are in same height and position



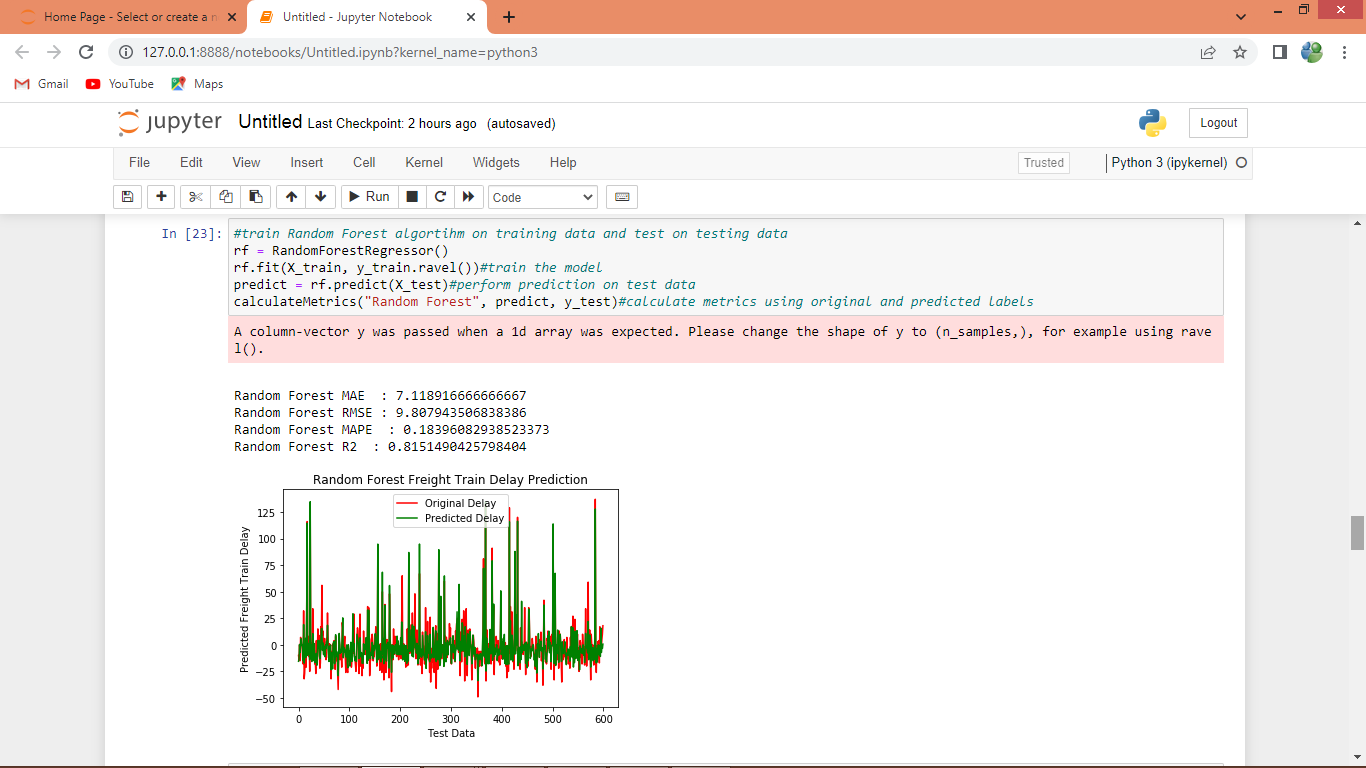
In above screen defining function to calculate MAE, MAPE, RMSE and R2.



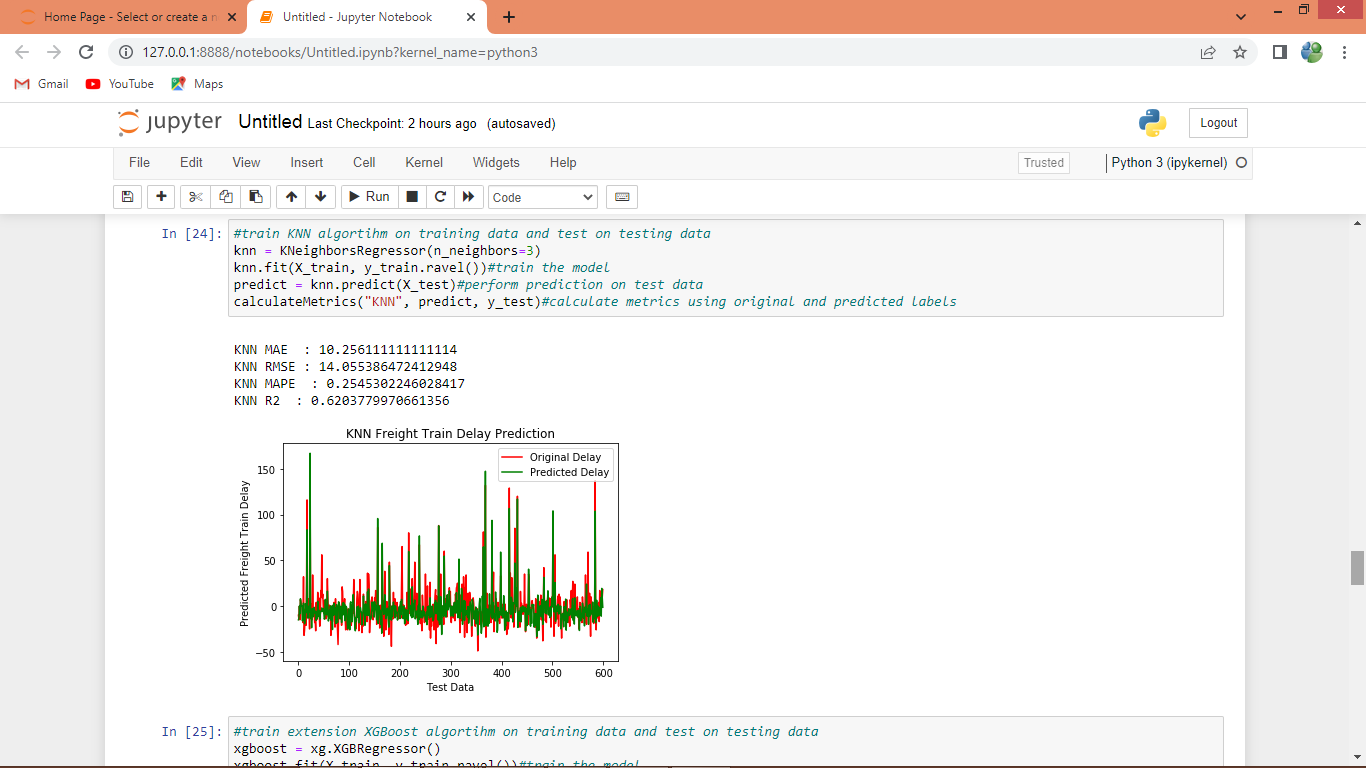
In above screen training LIGHTGBM algorithm on training data and then performing prediction on test data and then in blue colour line we can see R2 value as 89% and can see other metrics like MAE, MAPE and RMSE. In graph x-axis represents Test data index and y-axis represents DELAY where red line represents original Delay and green line represents Predicted Delay and we can see both lines are fully overlapping so predicted delay is closed to original test data delay



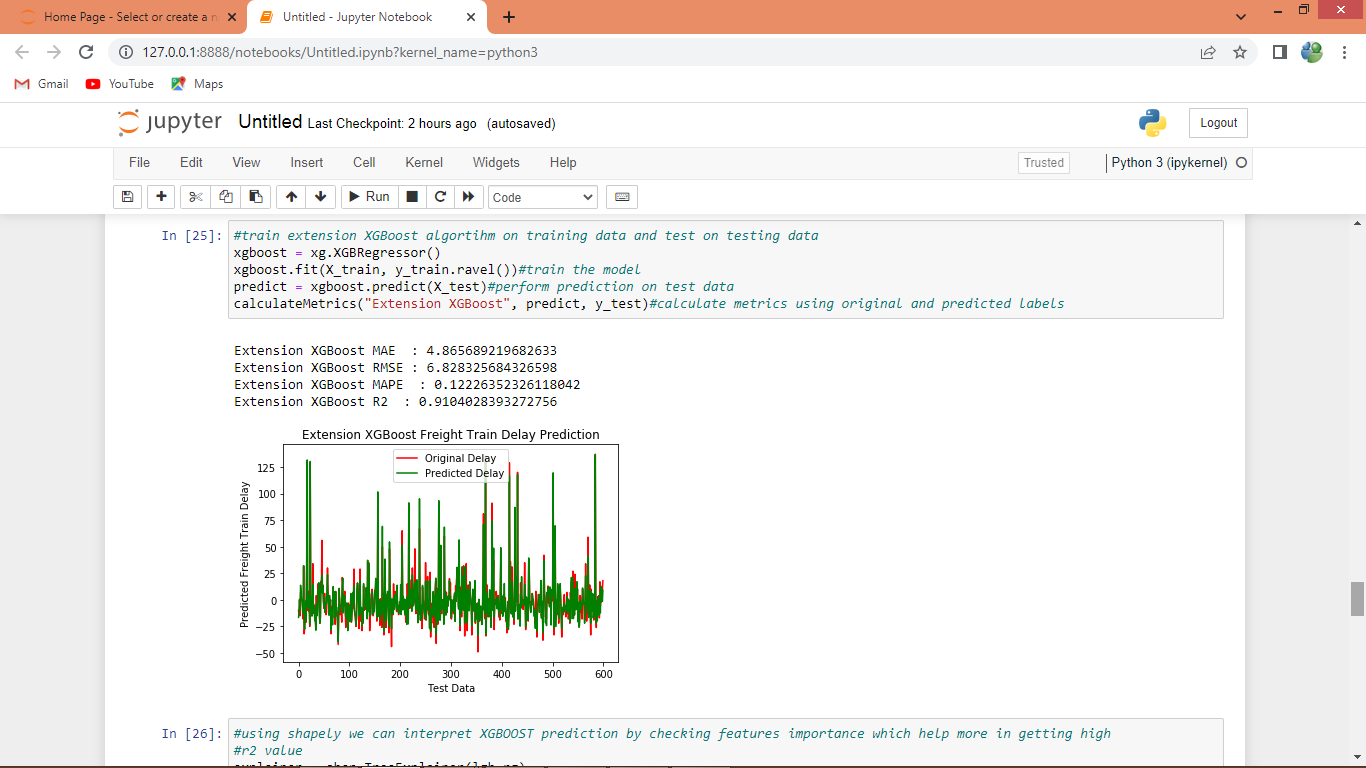
In above screen training Linear Regression and its R2 values is 69%



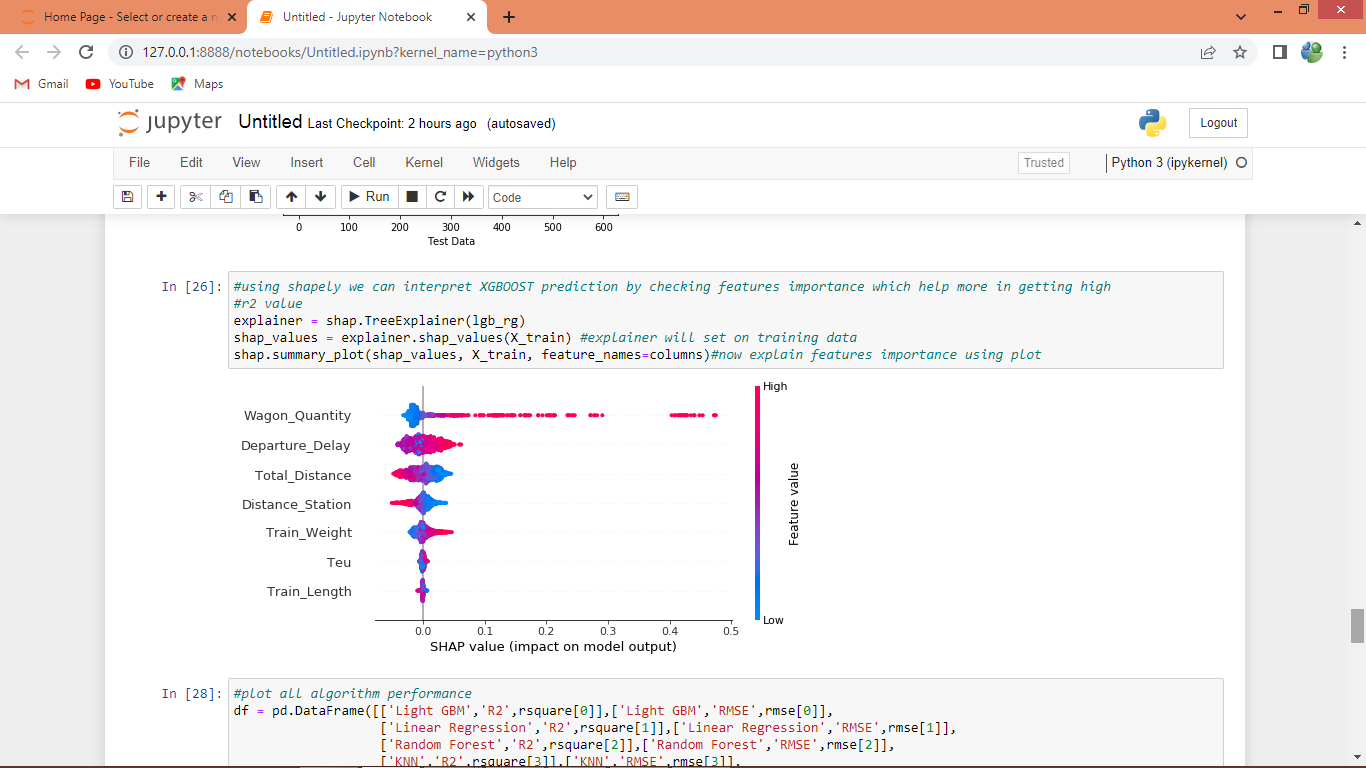
In above screen training Random Forest algorithm and its R2 values is 81%



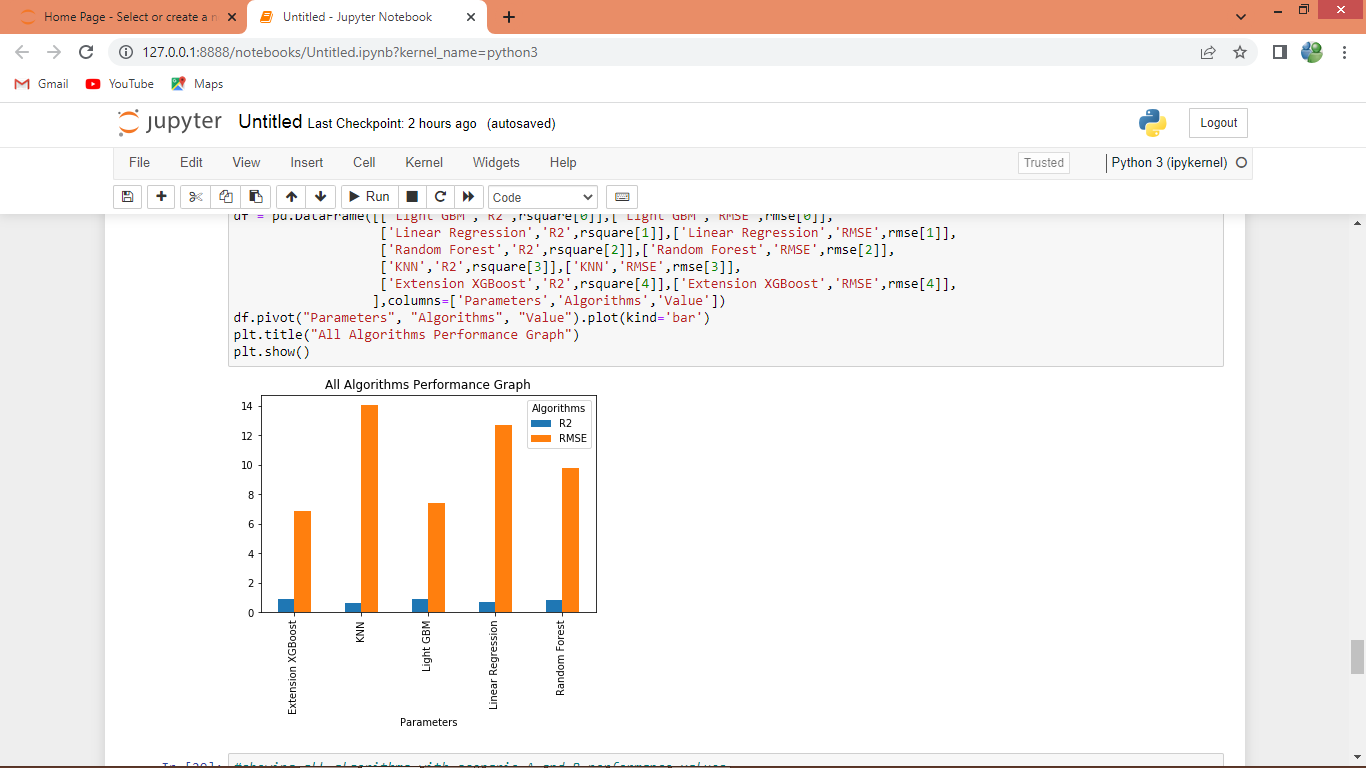
In above screen training KNN and its R2 value is 62%



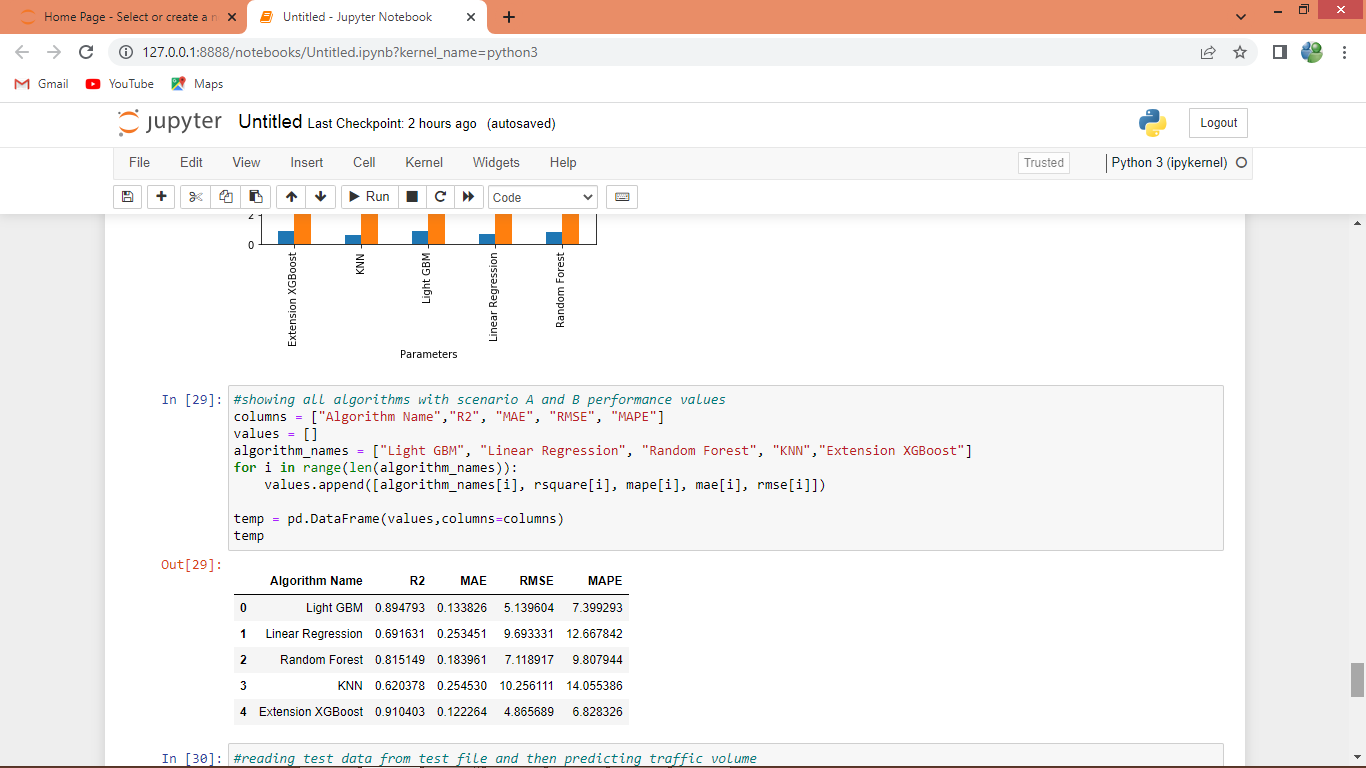
In above screen training extension XGBOOST and its R2 value is 91%



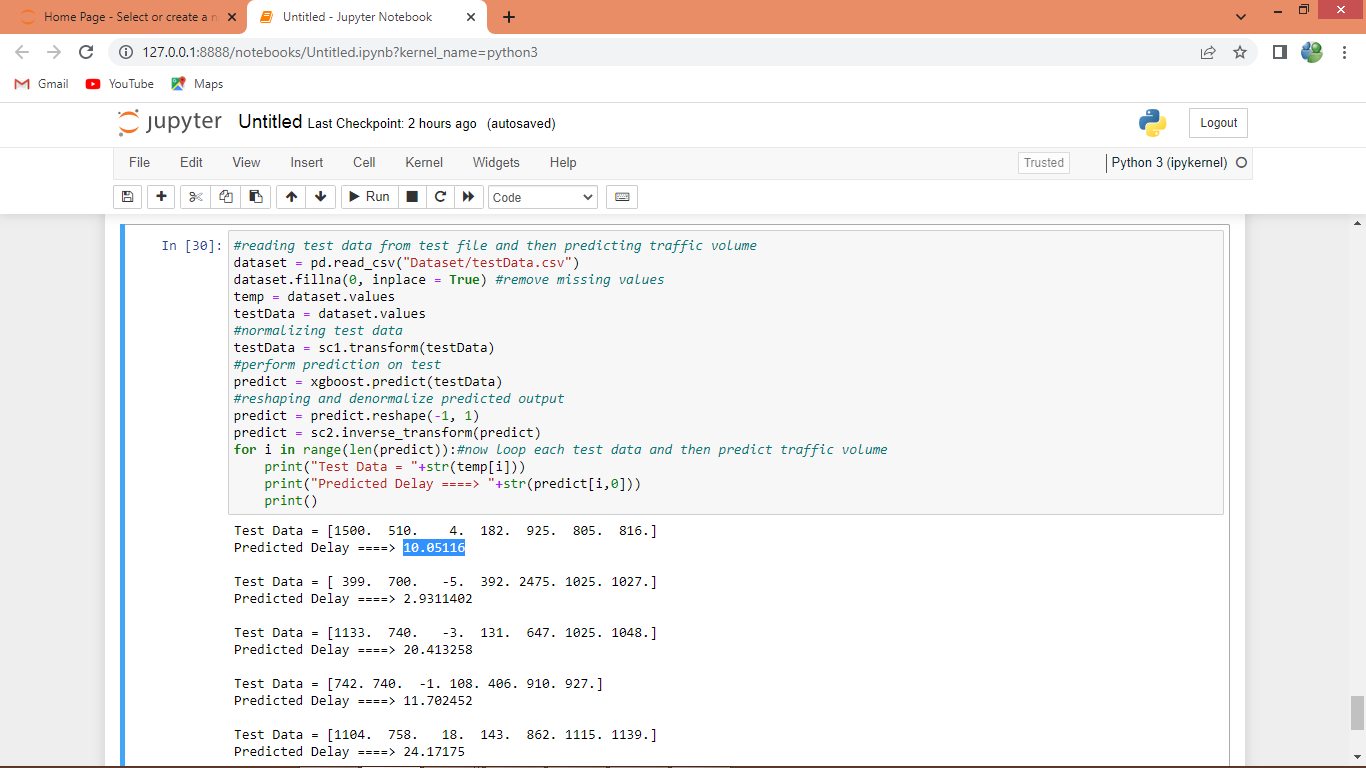
In above screen we are explaining about those features which help XGBOOST most in getting high R2 value. Above explanation we got using Shapely and this shapely will take algorithm object and training data and then find all those features which help algorithm in getting high R2. In above graph features which are reaching closer to HIGH boundary is contributing most for the algorithm to get high result



In above graph x-axis represents algorithm names and y-axis represents R2 and RMSE error and in all algorithms Extension XGBOOST and Light GBM got high R2 and less RMSE error compare to other algorithms



In above screen displaying all algorithm performance in tabular format



In above screen we are reading test data from test file and then predicting delay using XGBOOST extension object and in output we can see test data in square bracket and after arrow =🡺 symbol we can see predicted delay