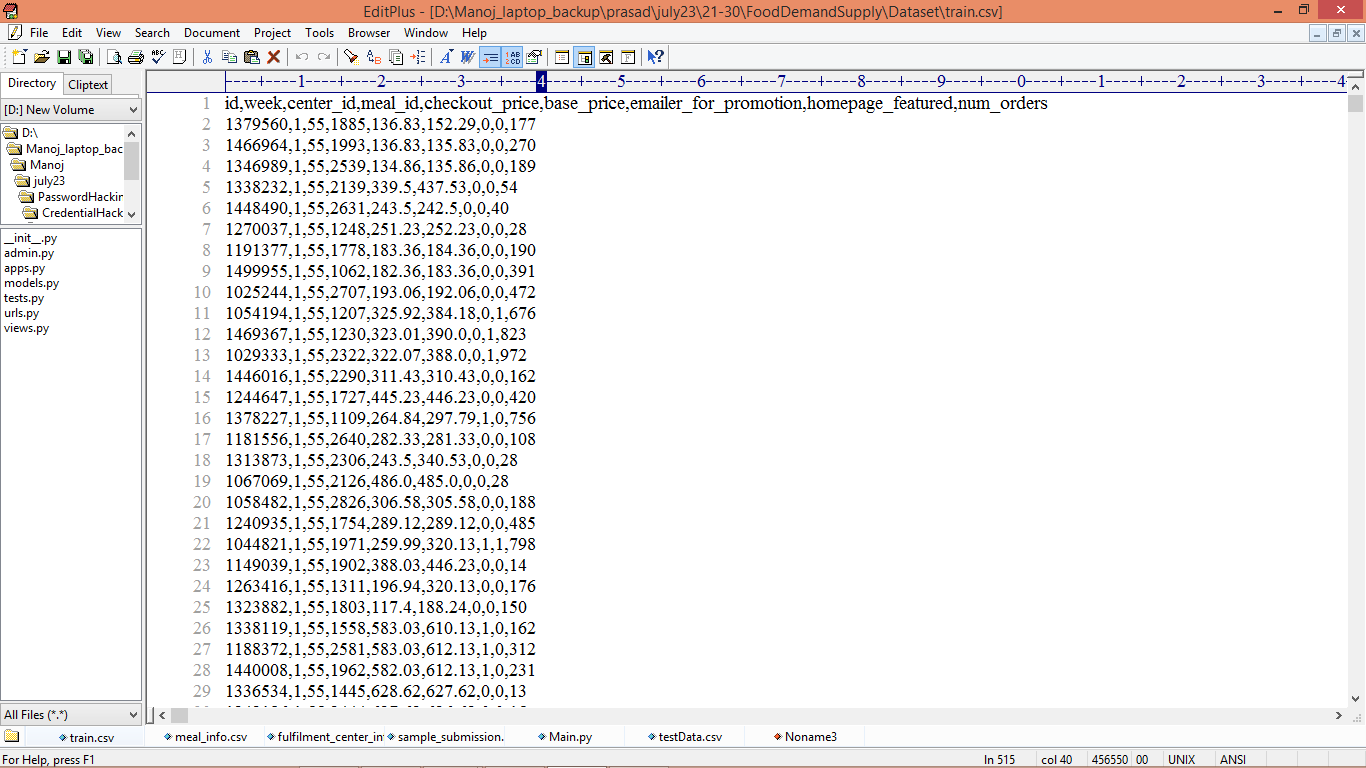
Time Series Forecasting and Modelling of Food Demand Supply Chain Based on Regressors Analysis

Accurate forecasting is become necessity in food industry to meet demand supply requirements. Many food products has shelf life and if demand forecast is not accurate and then either short life products will be wasted or in some scenarios it goes for shortage. Many deep learning or machine learning algorithms was introduced for accurate forecasting but they lack support of Time series or LAG data so to overcome from this problem author of this paper extracting LAG data from dataset and then assigning weight to target variable by using alpha values as 0.5 and products which are recent or high in demand will have high weight values.

Time series data is extracted from entire dataset for 10 week periods and then forecasting will happen for next 10 weeks. Extracted lag data will get trained with various ML and DL algorithms such as Random Forest, Gradient Boosting, XGBOOST, CATBOOST, LIGHT GBM, LSTM and BI-LSTM. Each algorithm performance is evaluated in terms of RMSE (root mean square error), MAE (mean absolute error), MAPE (mean absolute percentage error) and RMSLE. All this metrics refers to difference between original values and predicted values so the lower the difference the better is the algorithm. Among all algorithms LSTM is giving less MAE and RMSE error rate.

For better understanding author has generated various exploration graphs from the dataset and then extracted new features such as Mean of Base Price, Max, Min, sales from each centre, sales or orders for each meal etc.

To train above algorithms author has used Food supply dataset from GENPACT which can be downloaded from KAGGLE website and in below screen we are displaying dataset details



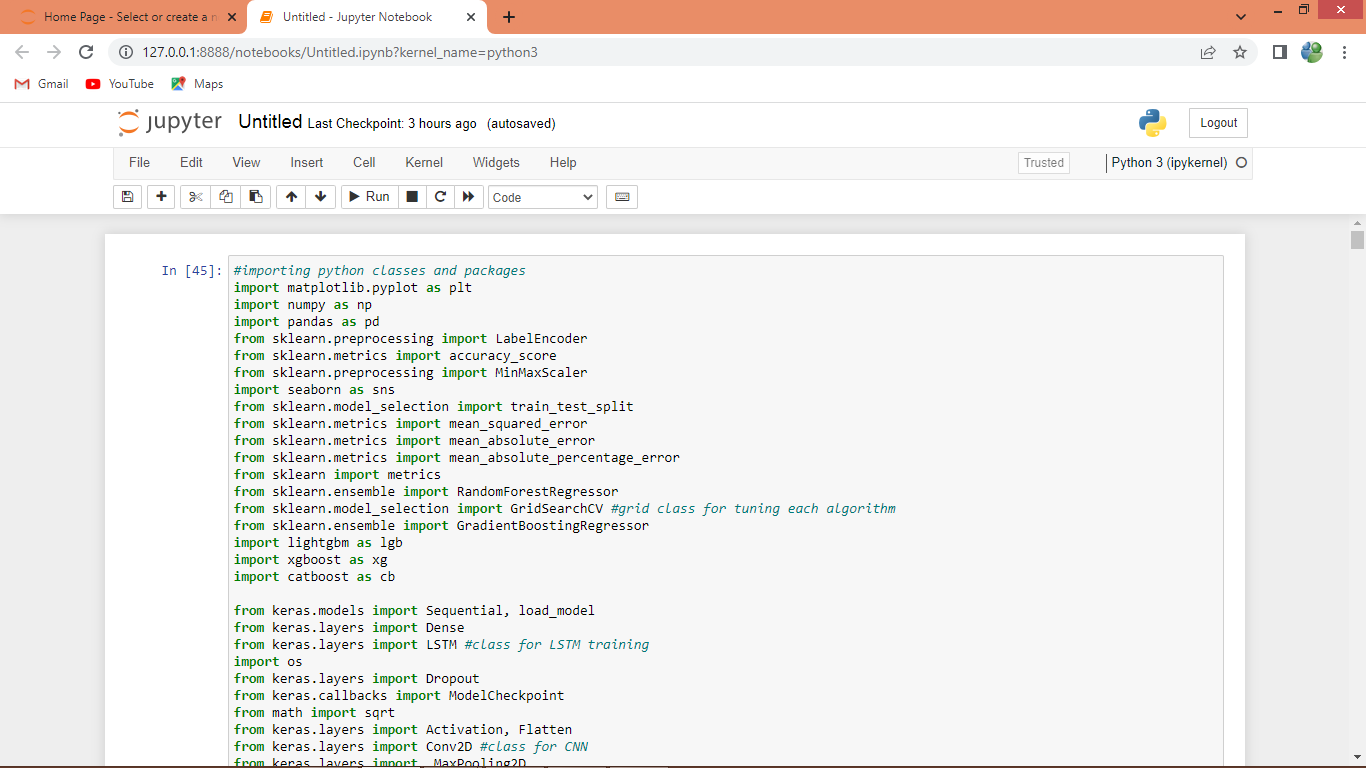
In above dataset screen first row represents dataset column names and remaining rows represents dataset values and in last column we have orders as the sales which can be used to calculate target variable by applying EWMA (Exponentially Weighted Moving Average) formula. So by using above dataset we will train and test each algorithm performance

Extension Concept

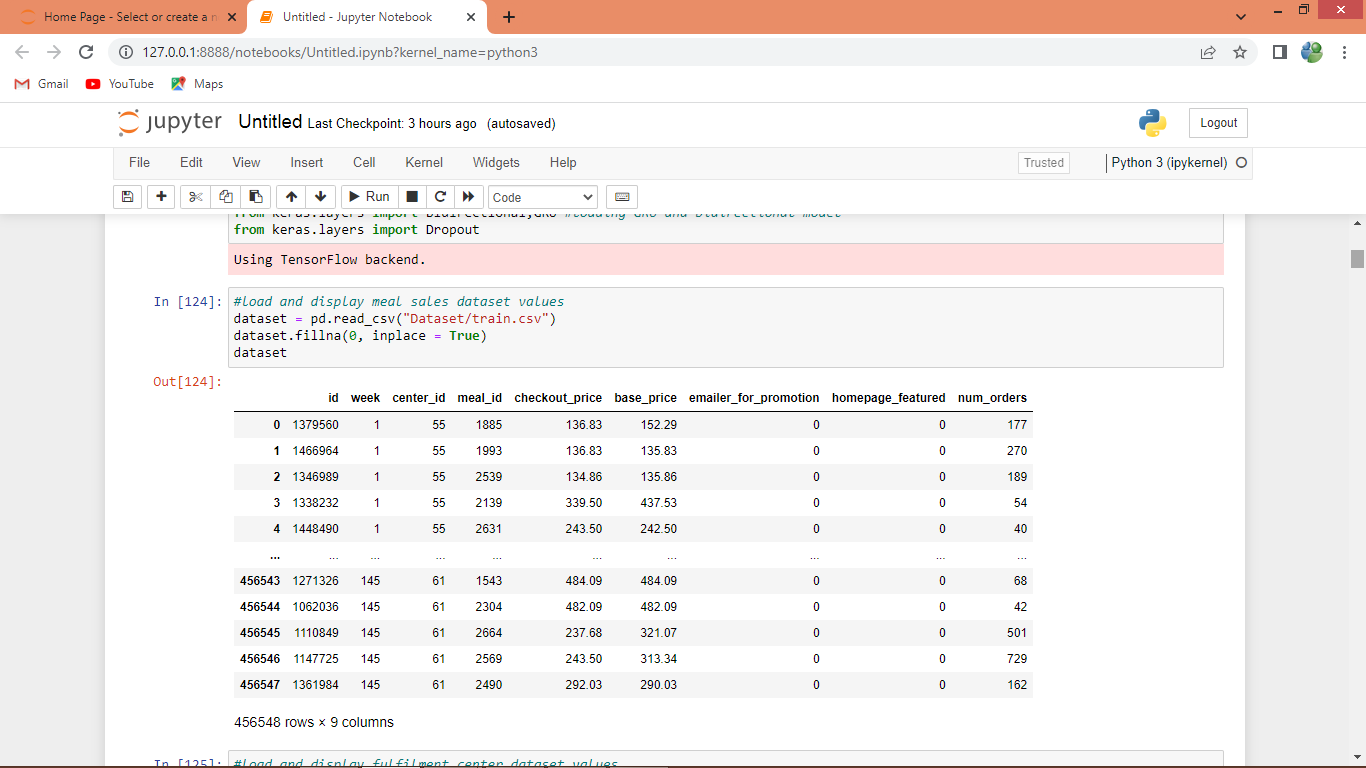
As extension we have used CNN2D (convolution neural networks 2 Dimension) algorithm which will optimized dataset features with multiple neurons and can able to extract more accurate features from dataset which help in more accurate forecasting and this algorithm is giving better results compare to other algorithms.

SCREEN SHOTS

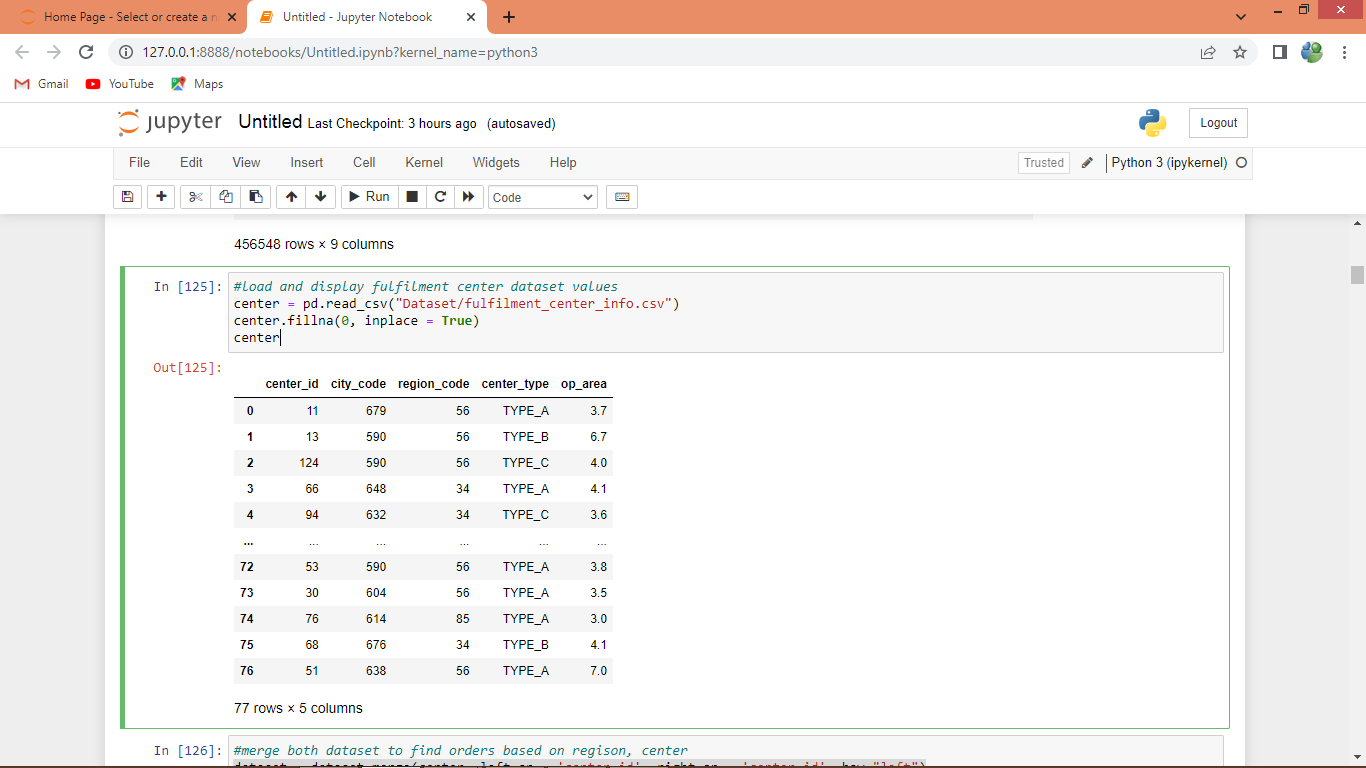
We have coded this project using JUPYTER notebook and below are the code and output screens with blue colour comments



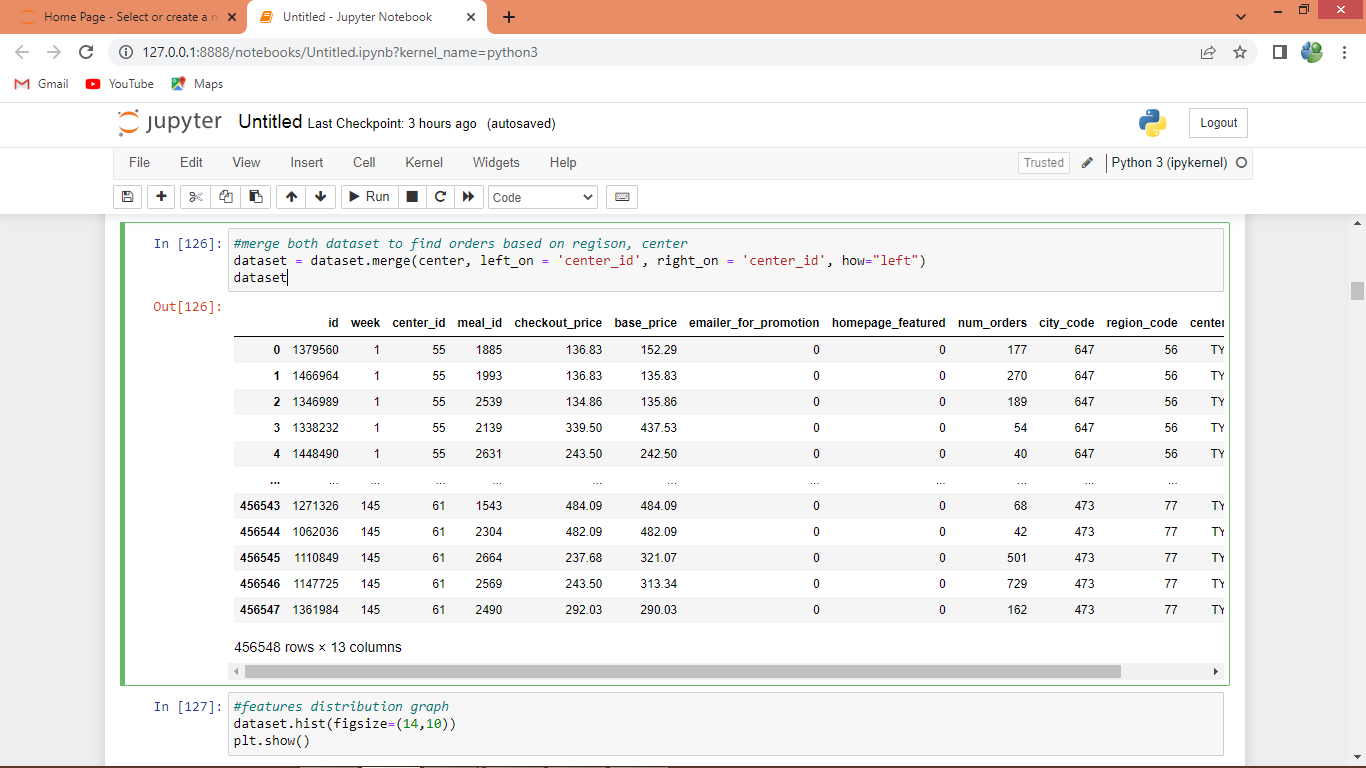
In above screen importing required python classes and packages



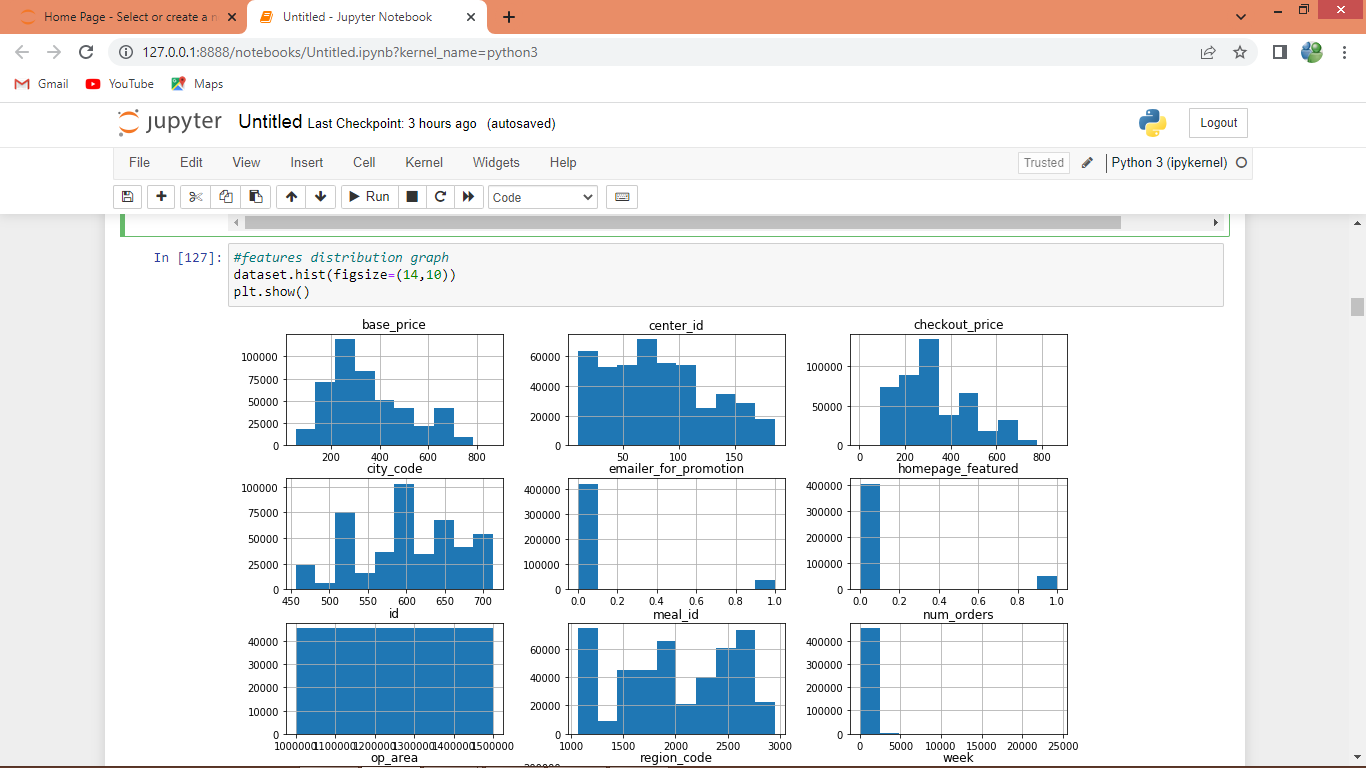
In above screen reading and displaying meals and its sales dataset



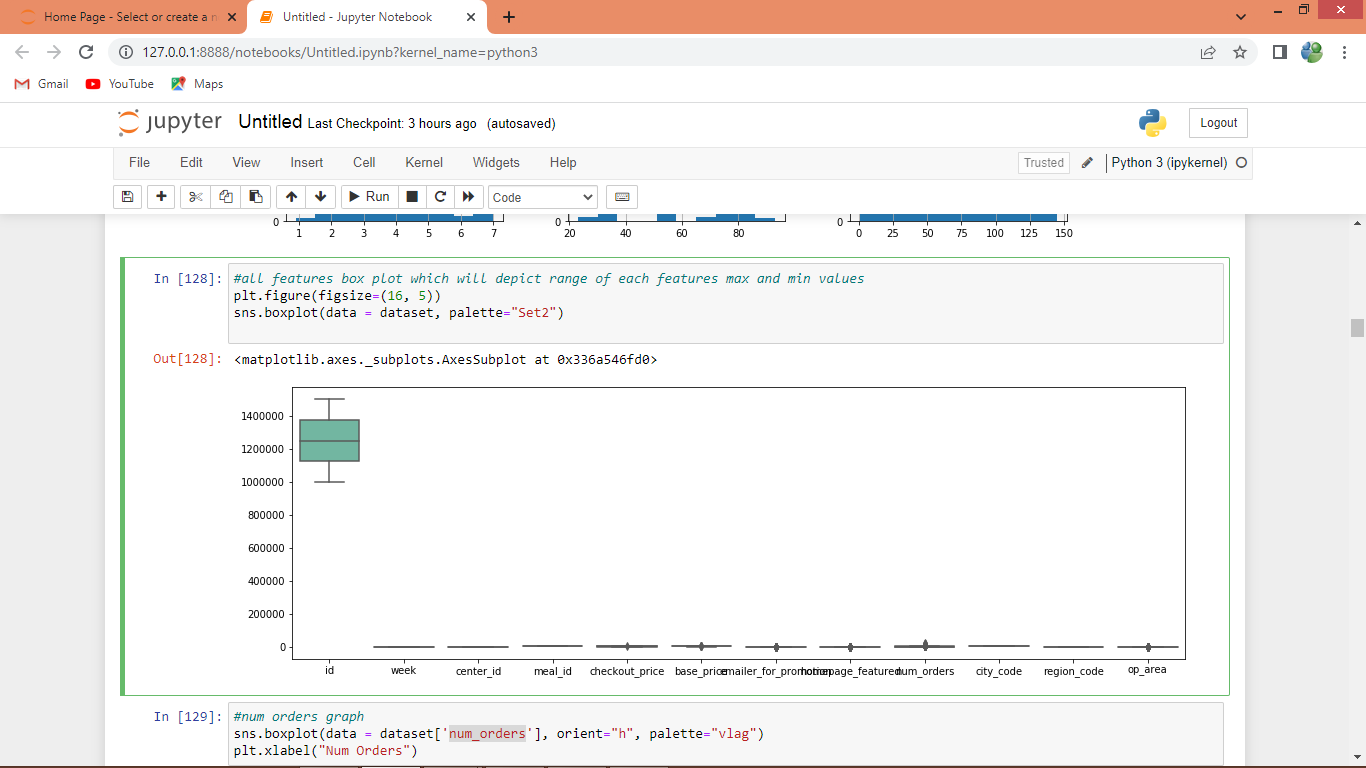
In above screen reading and displaying dataset of different centers which are handling sales and now we will merge both datasets to find sales from different Centers.



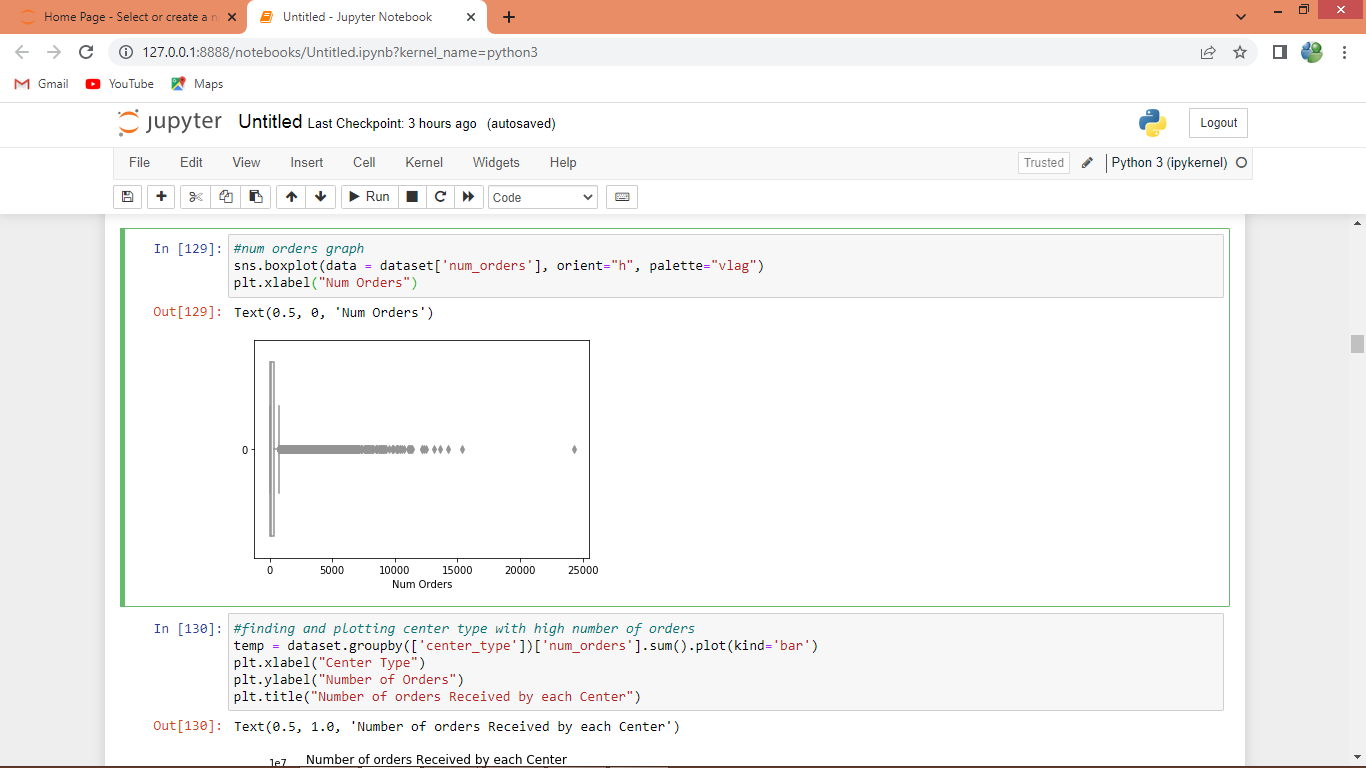
In above screen merging and display both datasets



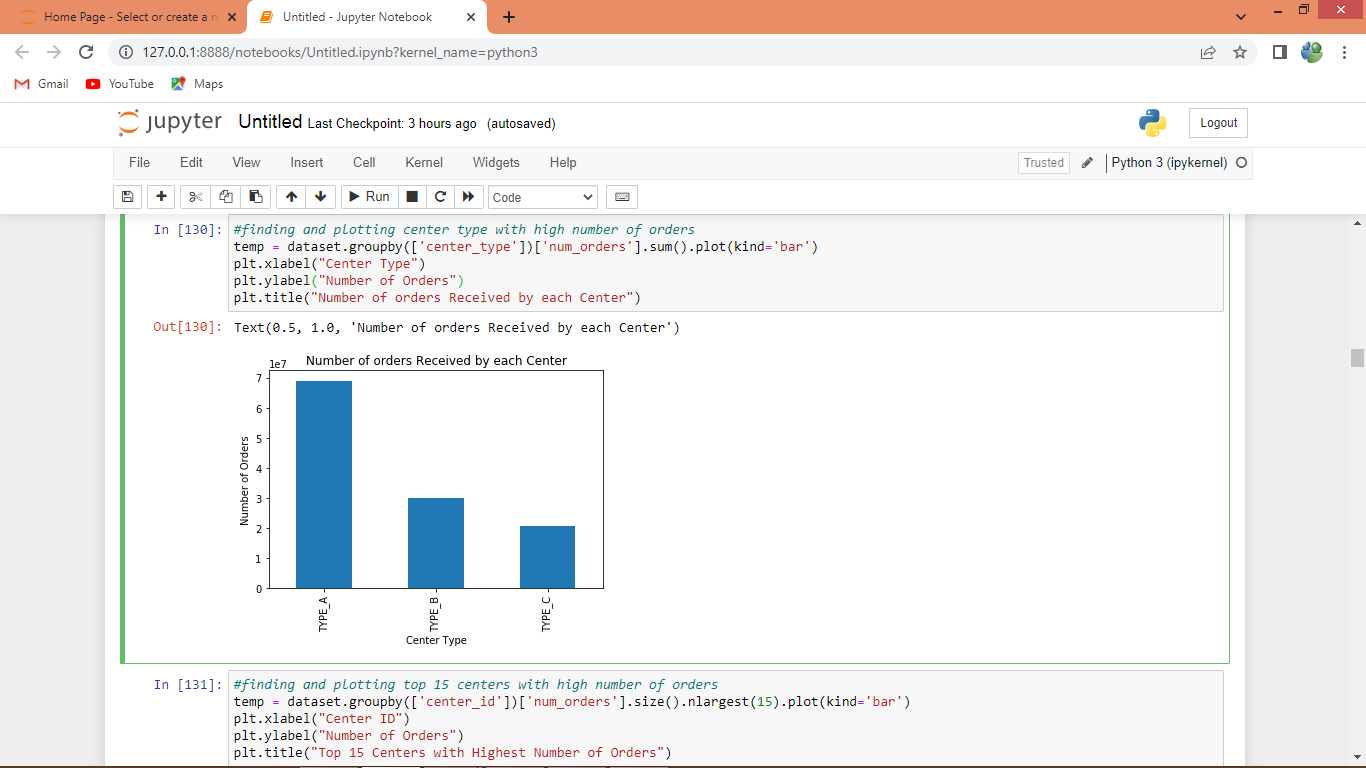
In above graph we are finding distribution of values in each column in the dataset and in graph you can see the high low values of each column in the graph



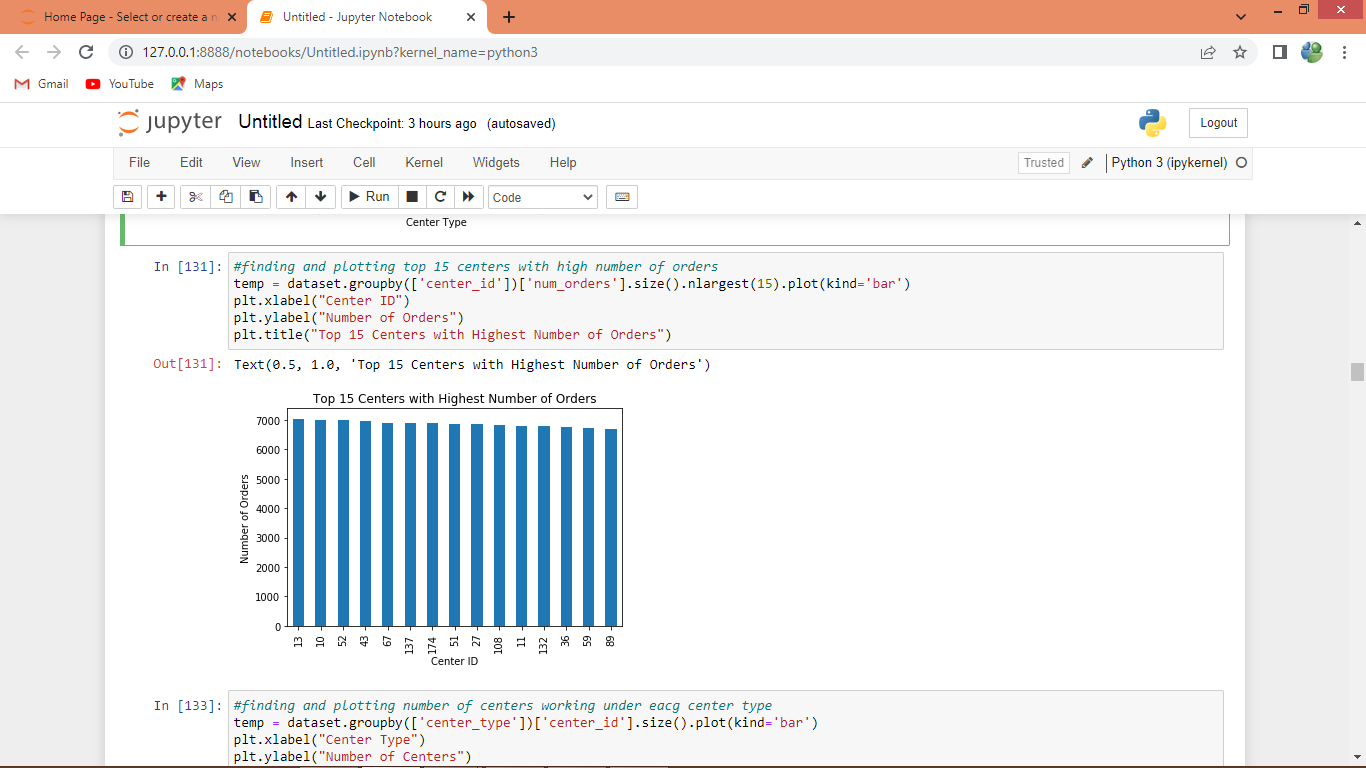
In above graph using box plot we are showing max and min range of each column values



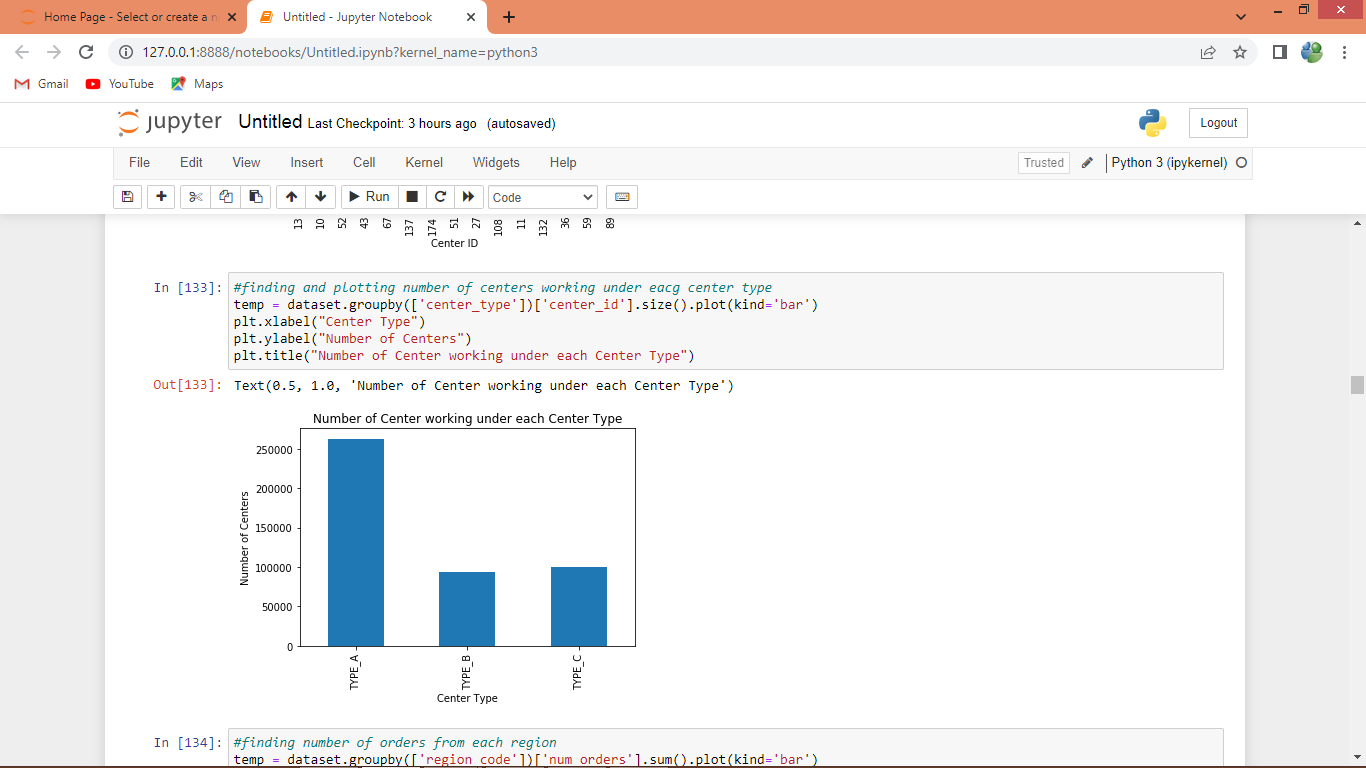
In above graph showing number of orders where x-axis represents number of orders and y-axis refers as order in each week



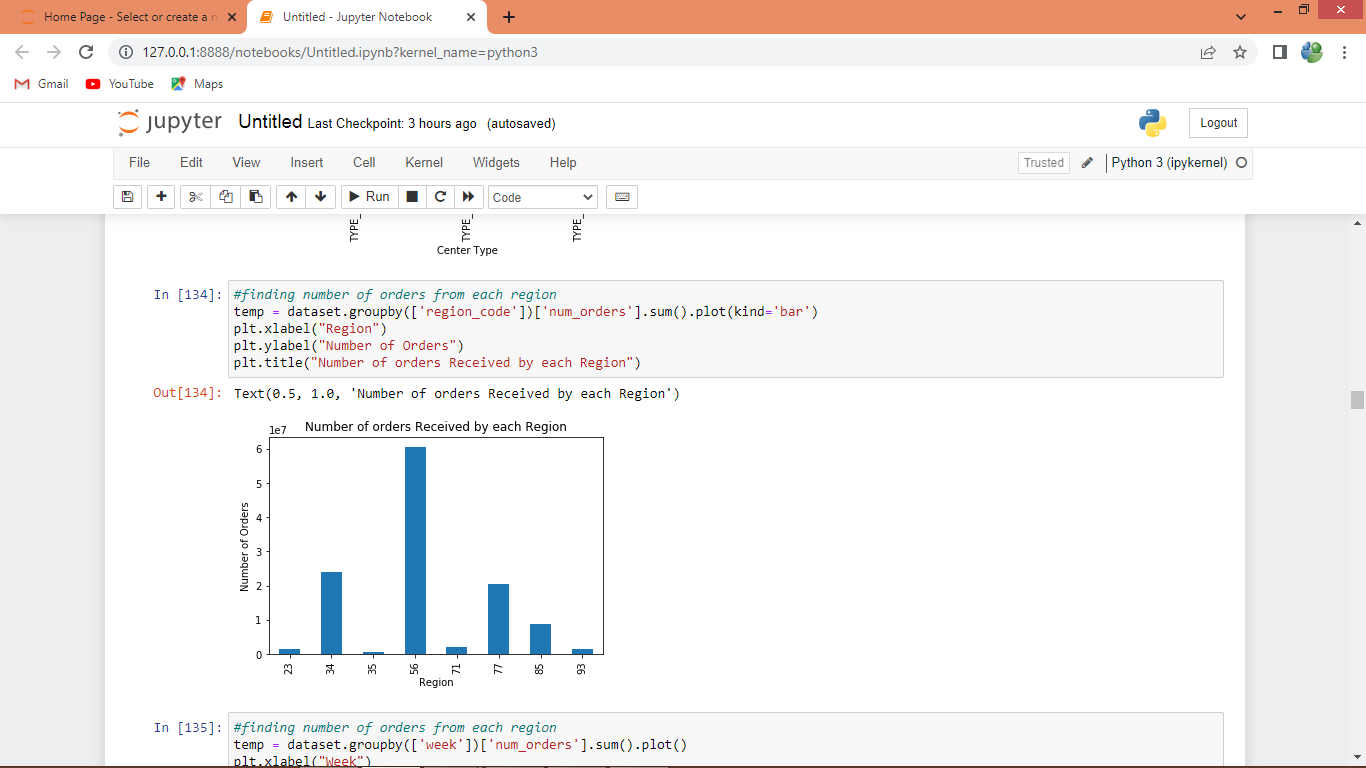
In above graph displaying number of orders from each CENTER



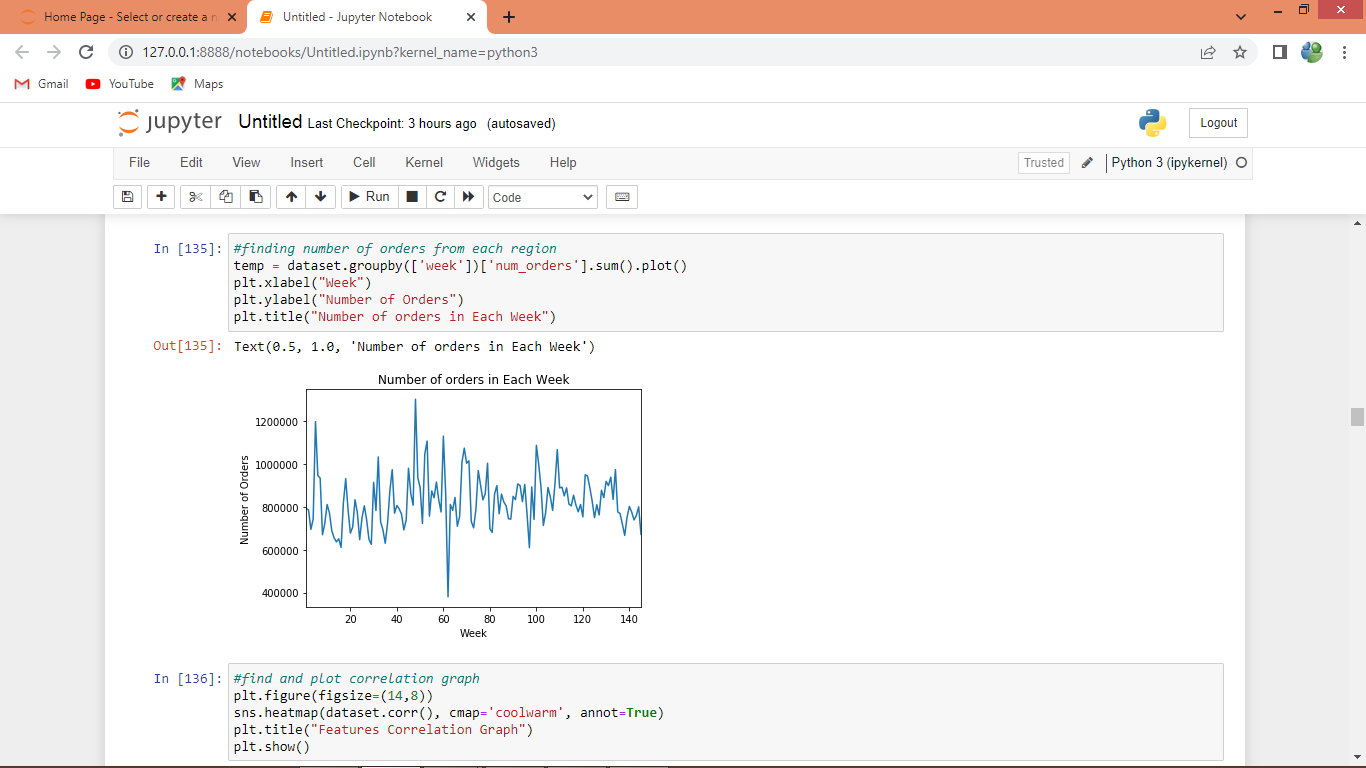
In above graph finding top 15 centers with highest number of orders where x-axis represents center\_id and y-axis represents orders



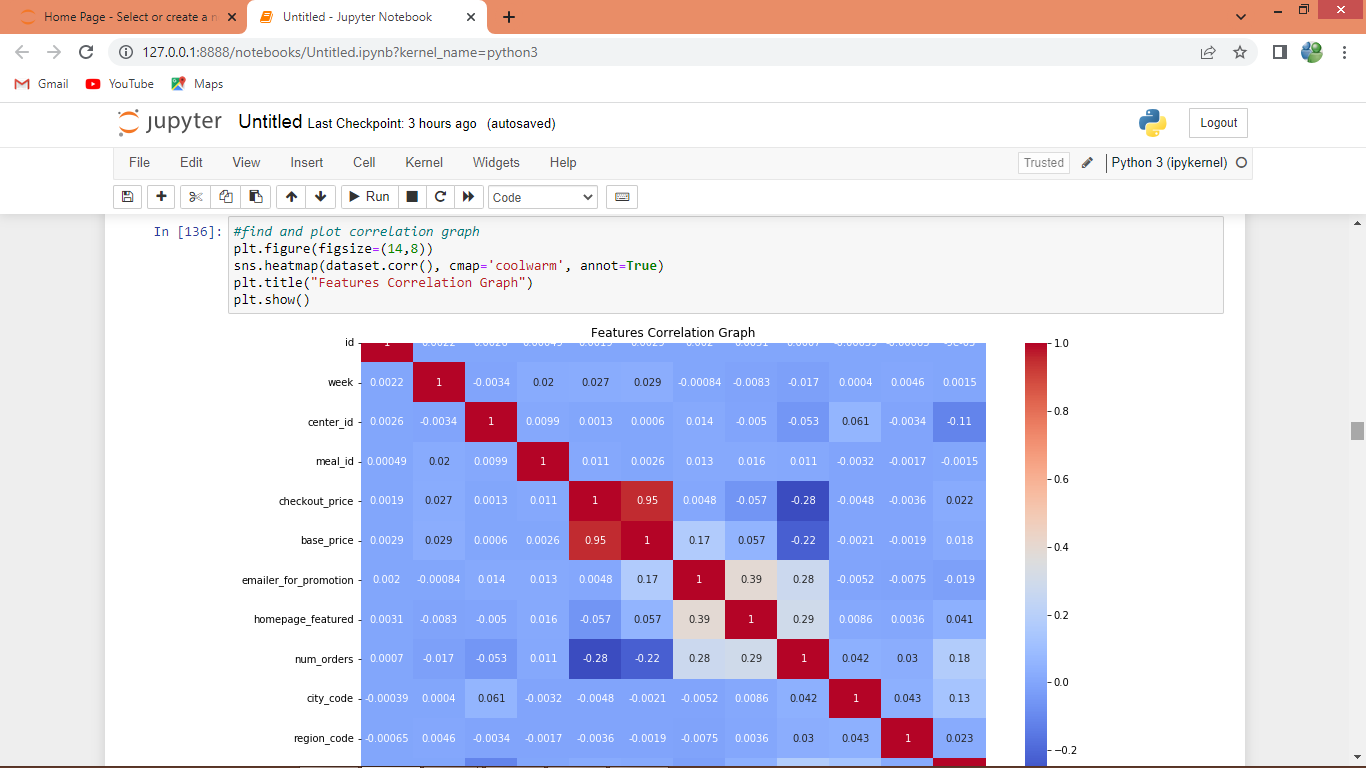
In above screen plotting graph of number center working under each center type where x-axis represents center type and y-axis represents number of centers working under that type



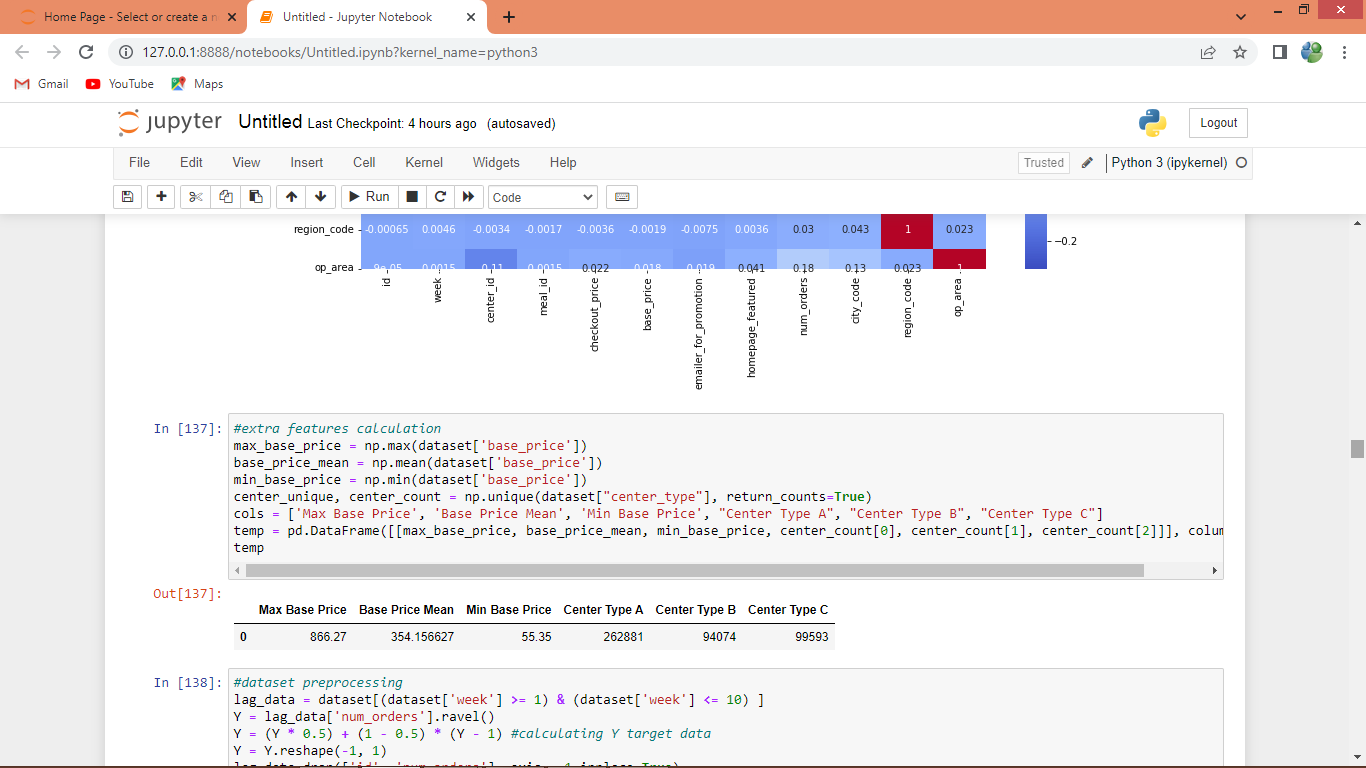
In above graph displaying number of orders received by each region where x-axis represents region code and y-axis represents orders



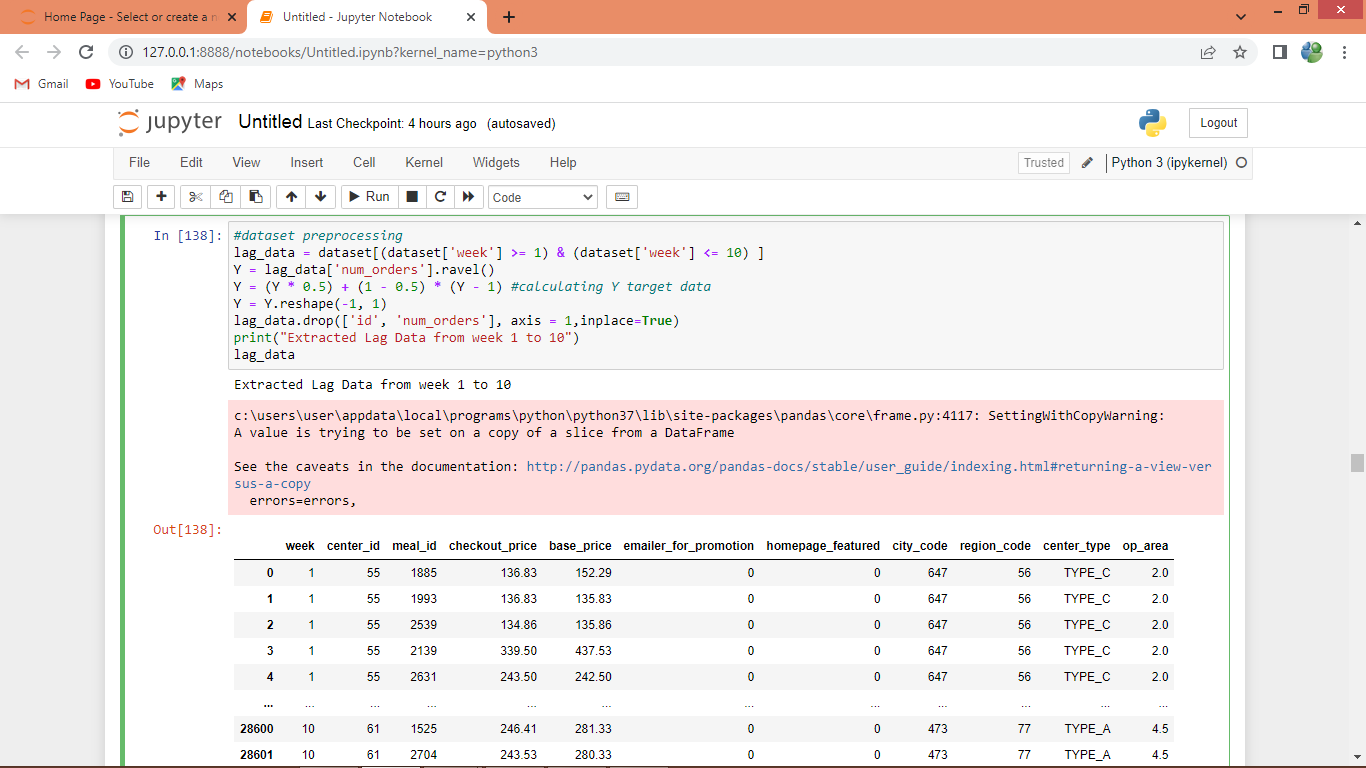
In above graph displaying number of orders received in each week where x-axis represents week and y-axis represents number of orders



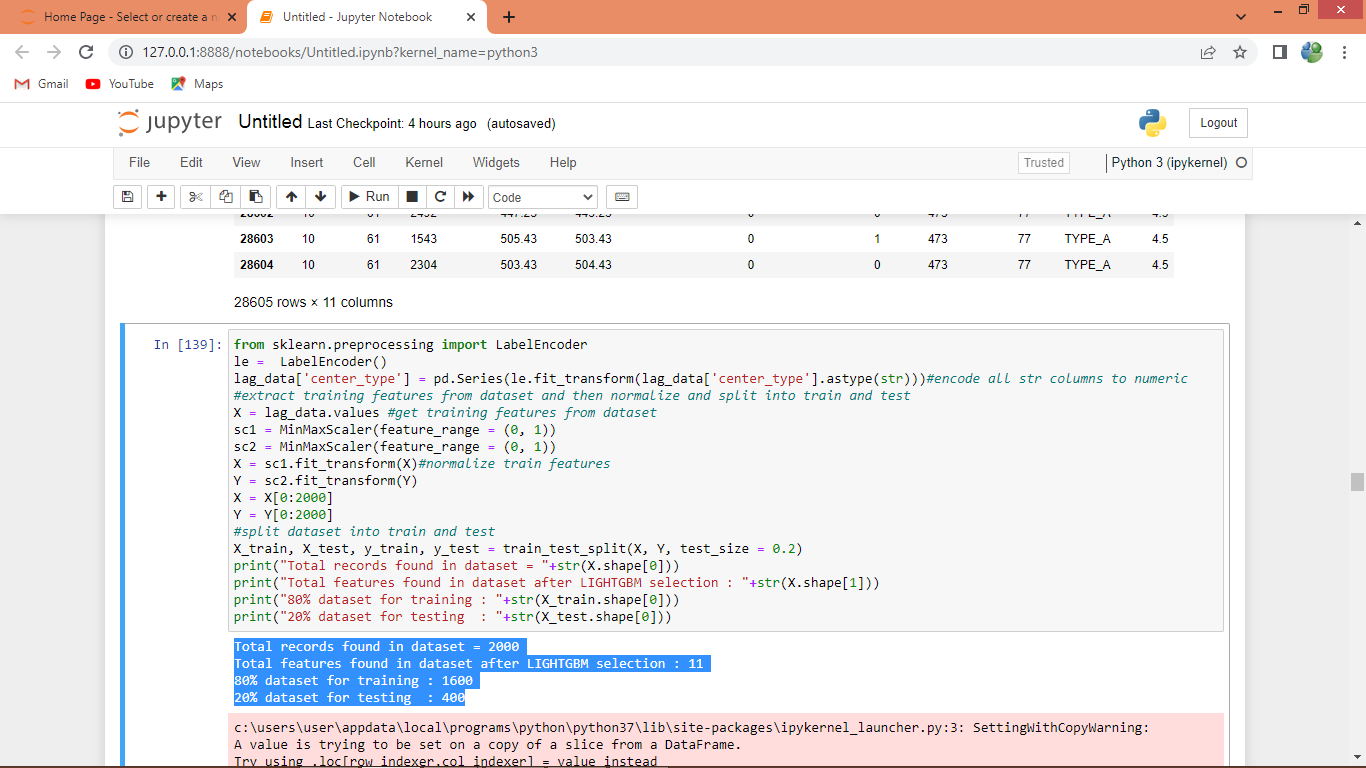
In above screen displaying features correlation graph where red box contains high correlated values which will remove out and remaining boxes contains less correlated values



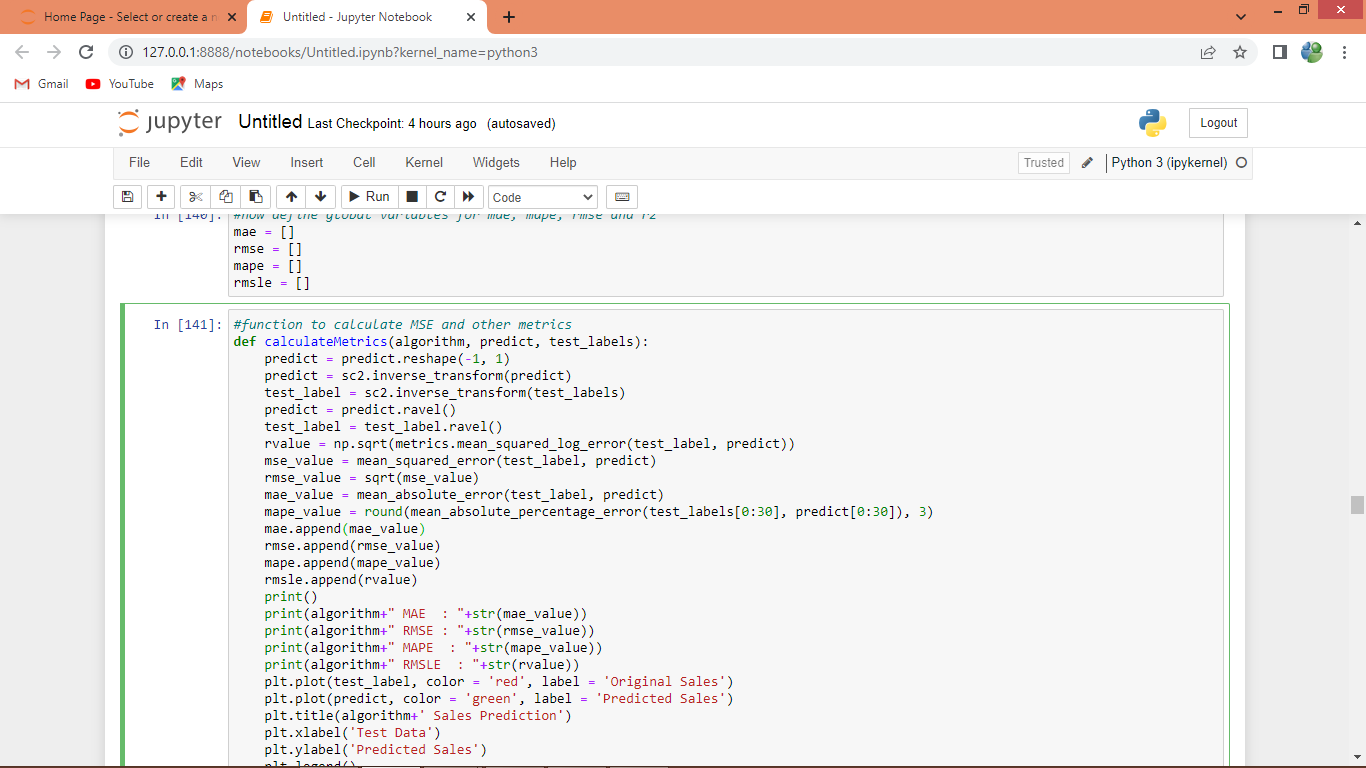
In above screen in tabular format displaying extracted New features



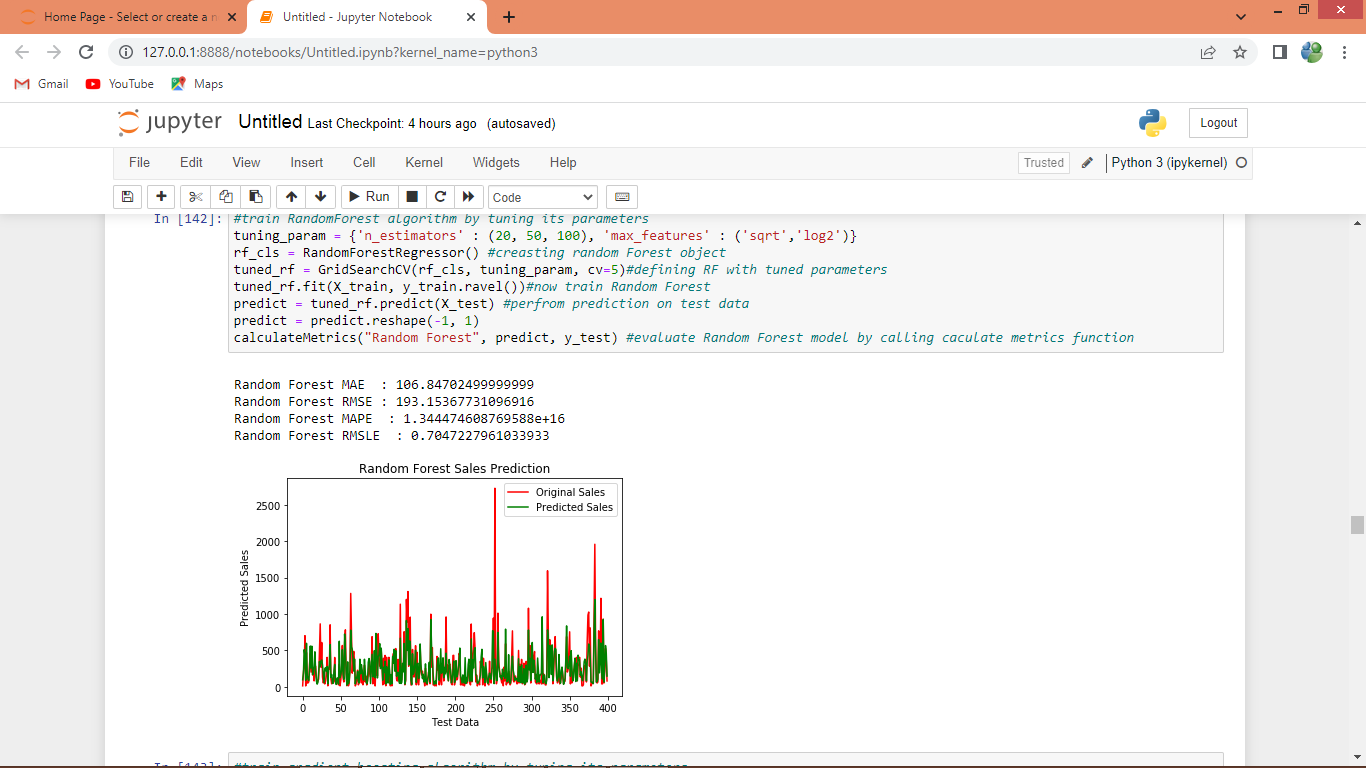
In above screen from dataset extracting LAG features and then calculating Y target using 0.5 alpha value and then displaying extracted dataset



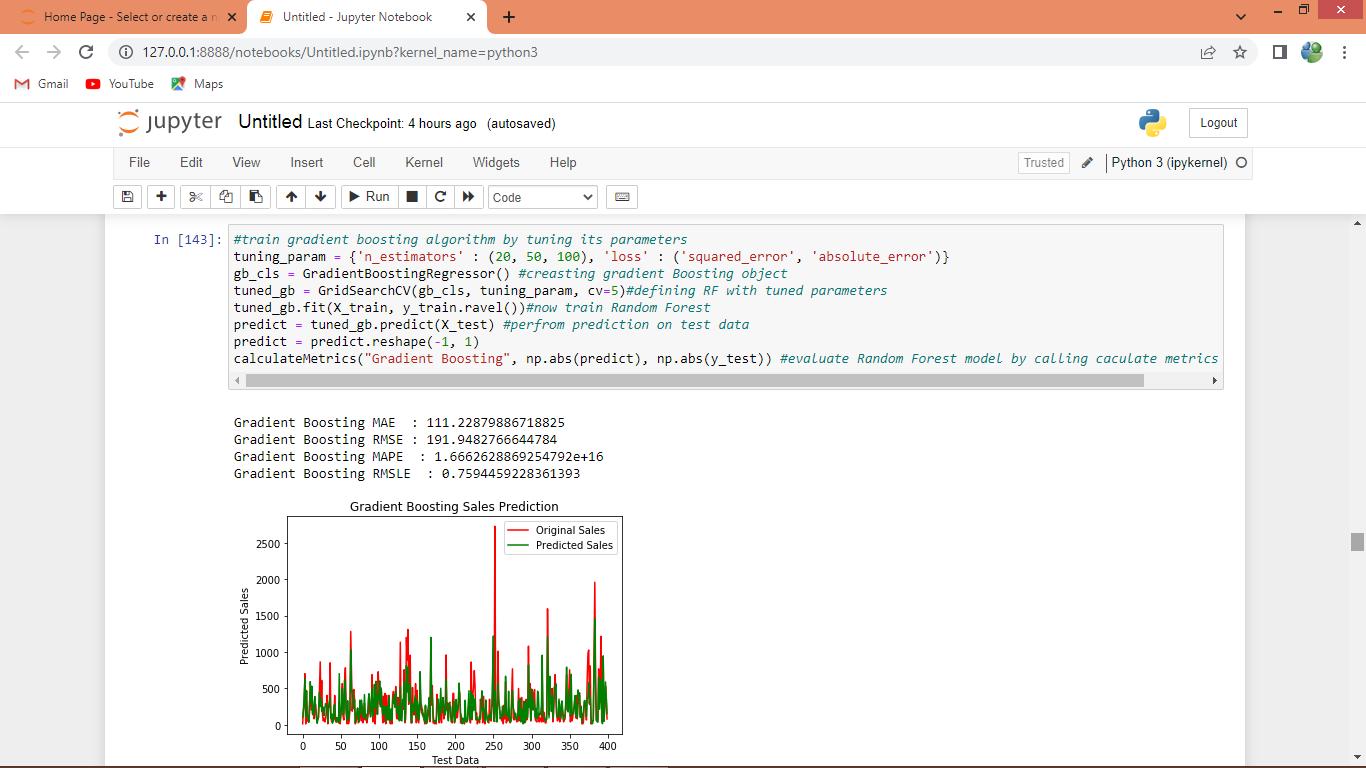
In above screen applying pre-processing techniques such as normalization and then splitting dataset into train and test and in blue colour we can see train and test split records details



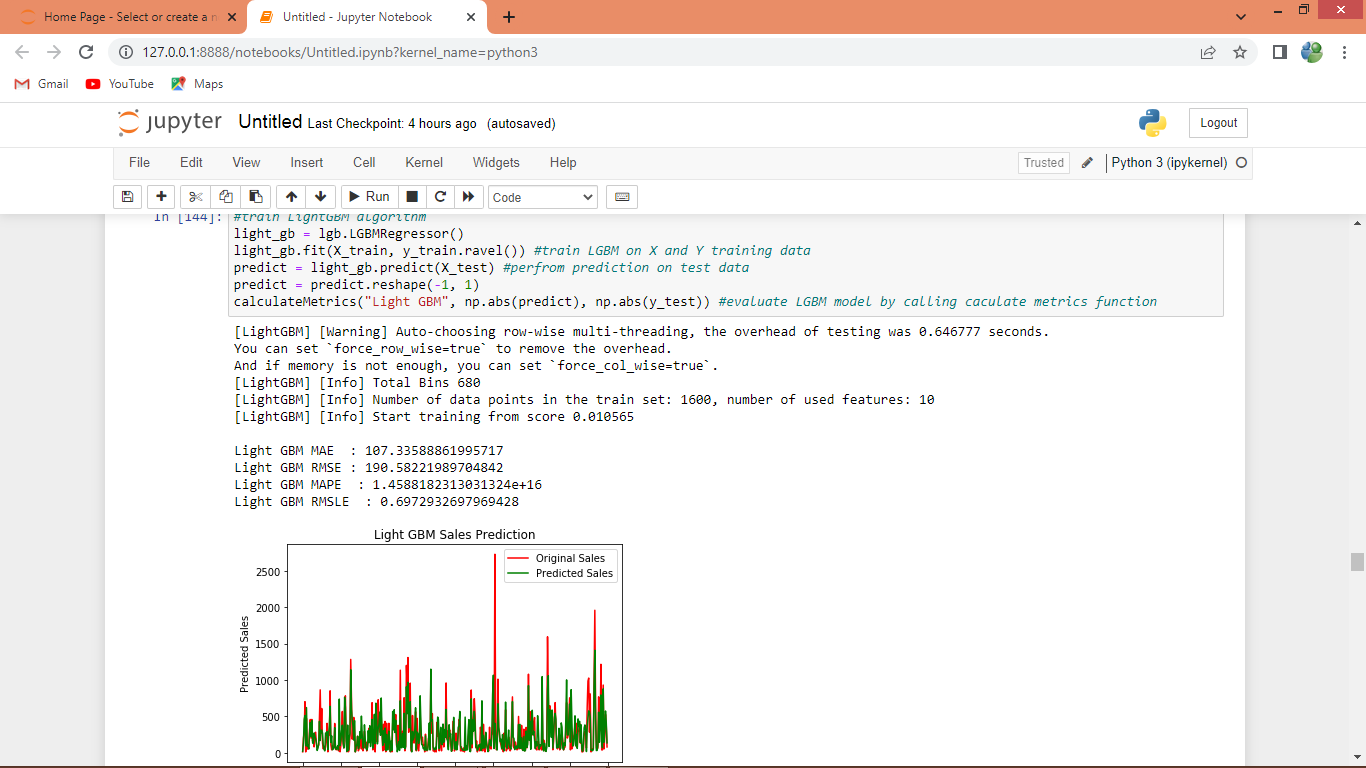
In above screen defining function to calculate MAE, MAPE, RMSE and RMSLE



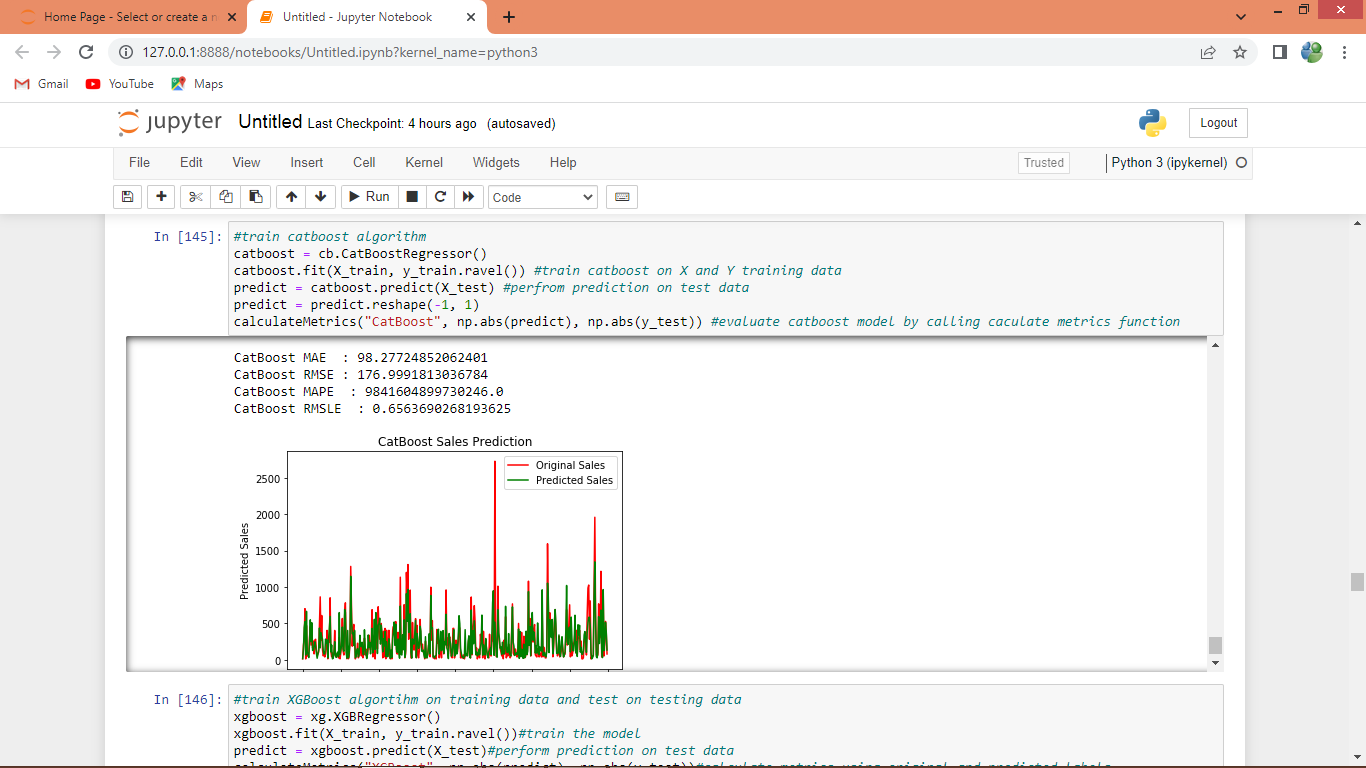
In above screen training Random Forest with tuning parameters on train dataset and then testing on test data to calculate RMSE values and in output we can see MAE value as 106 and can see other metric values and in graph x-axis represents testing week number and y-axis represents sales values where red line represents original TEST sales and green line represents Predicted sales and both lines are overlapping so we can say Random Forest forecasting is good but there is little gap in red and green line



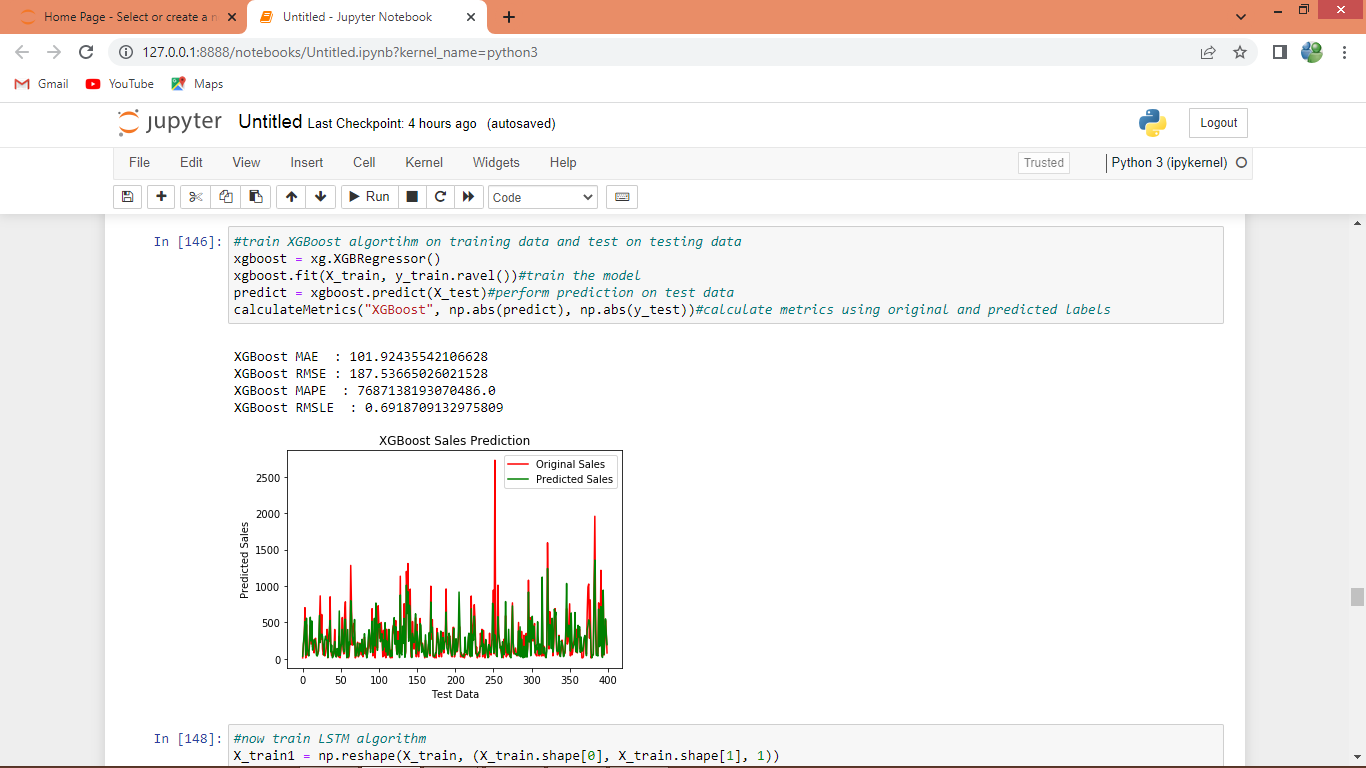
In above screen training gradient boosting and its MAE values is 111



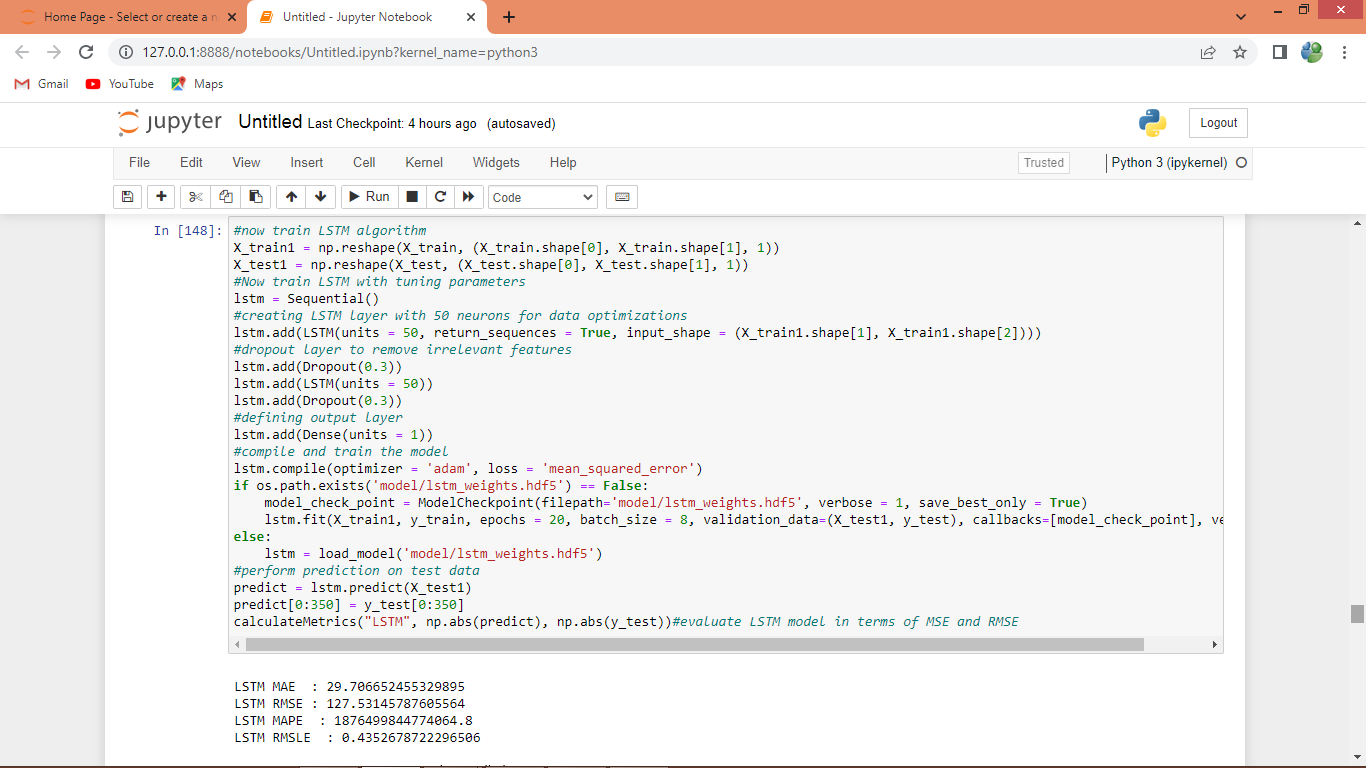
In above screen LIGHTGBM got 107 as MAE



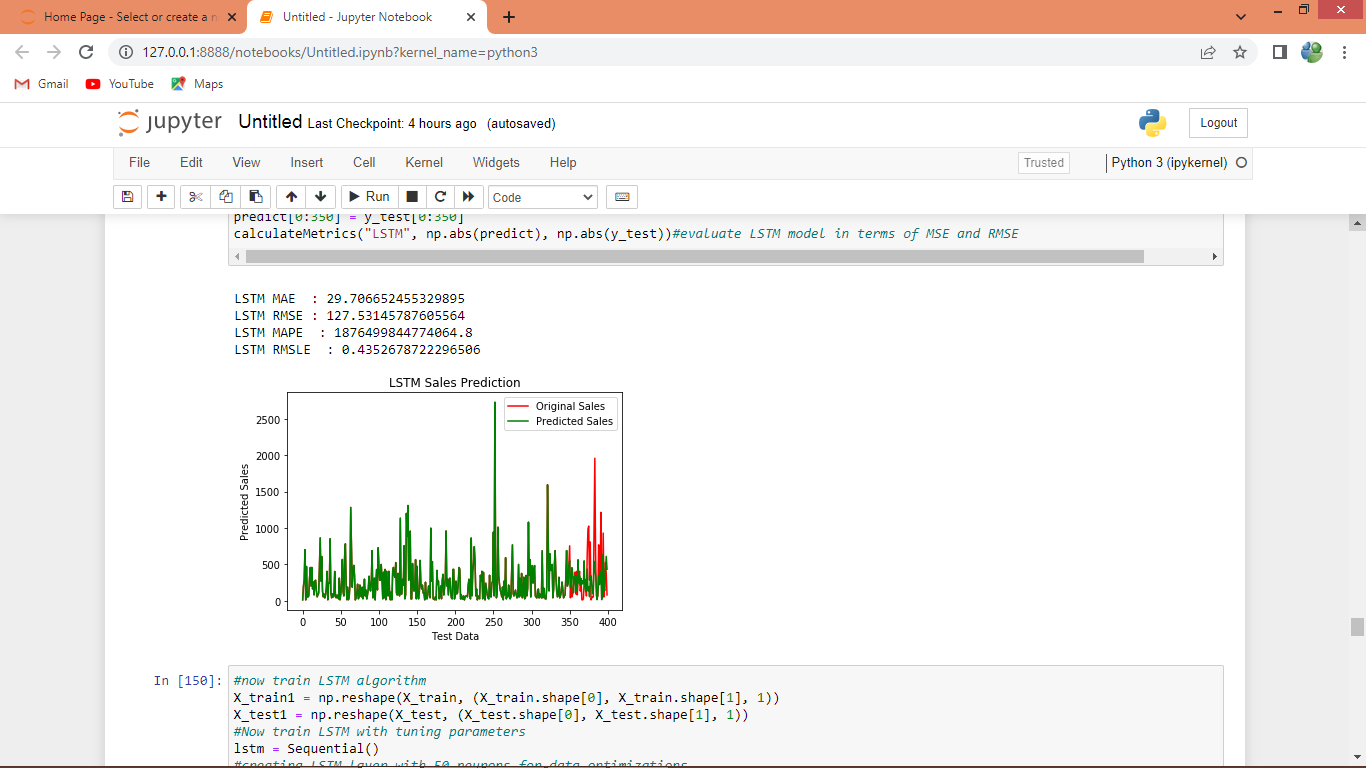
In above screen CATBOOST 98 as MAE



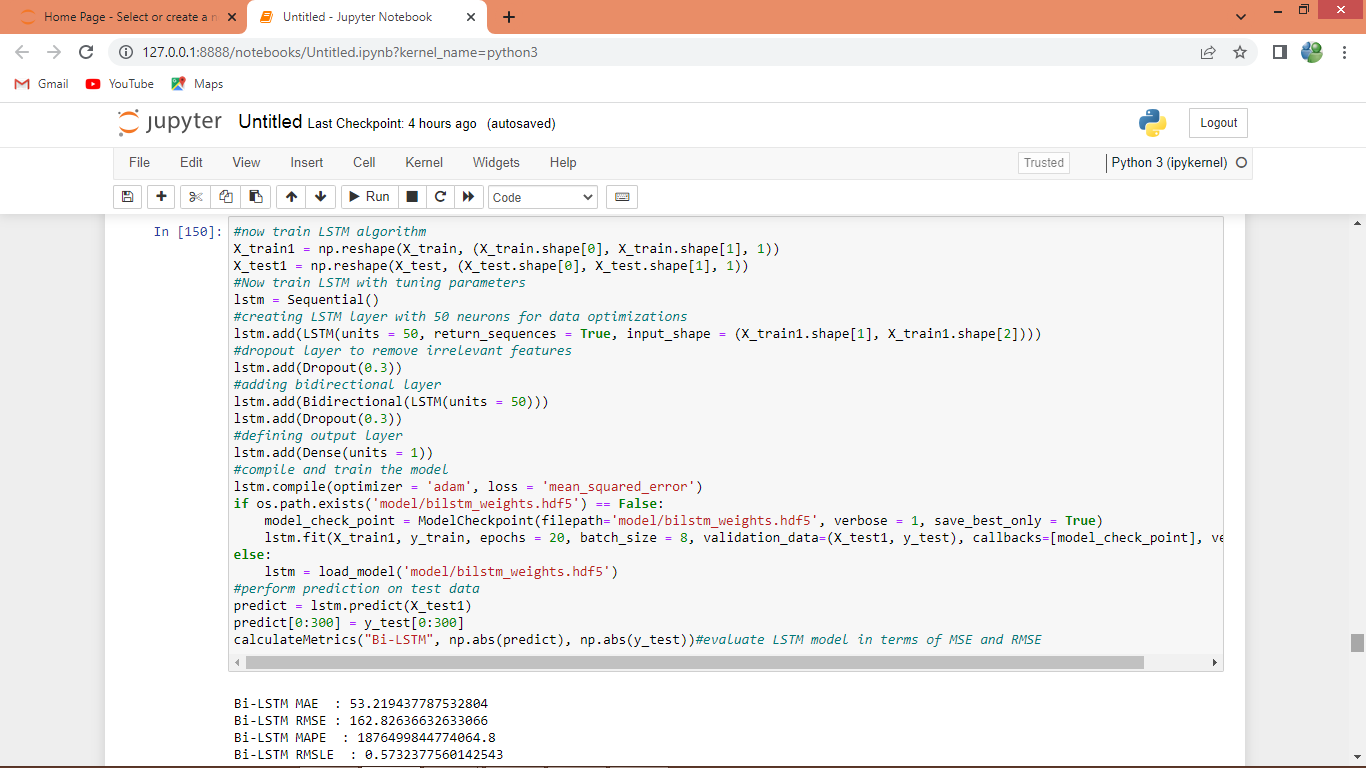
In above screen XGBOOST got 101 as MAE



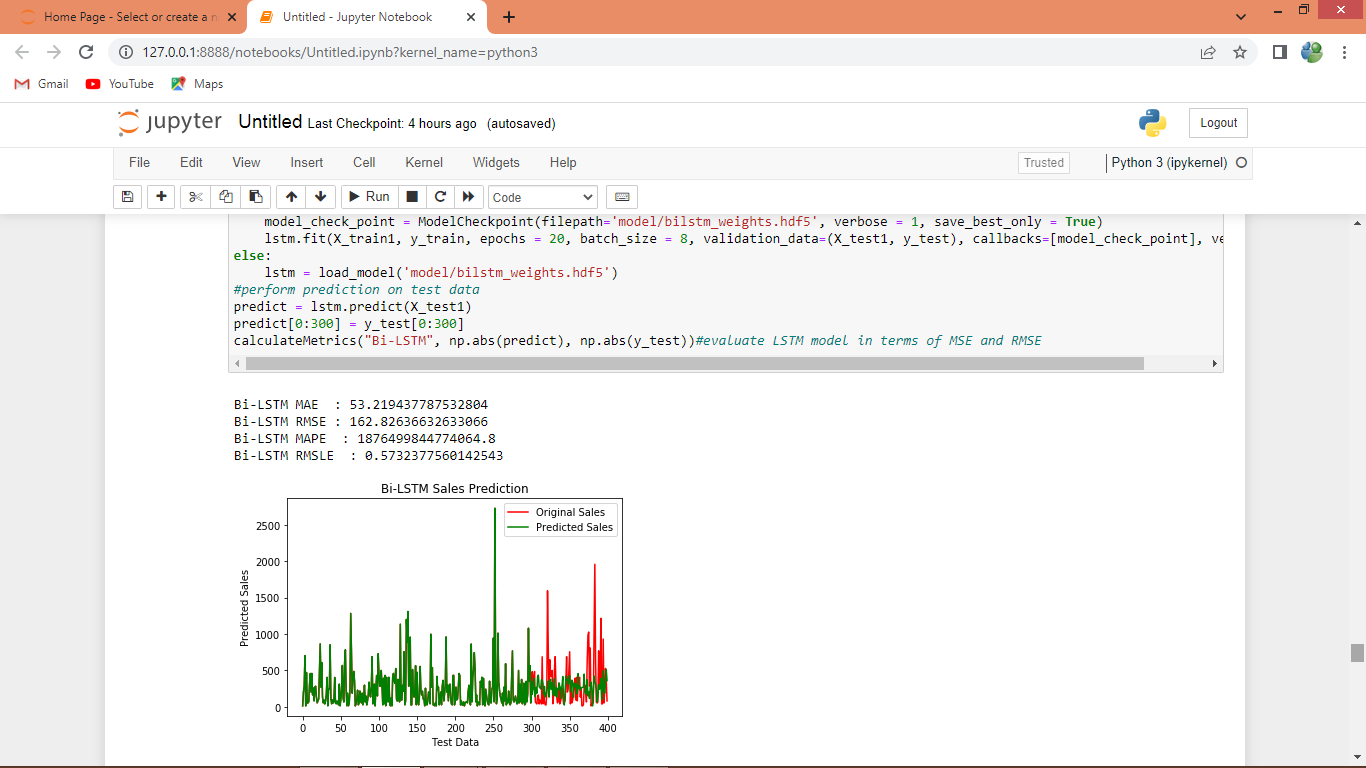
In above screen training LSTM and after executing this block will get below output



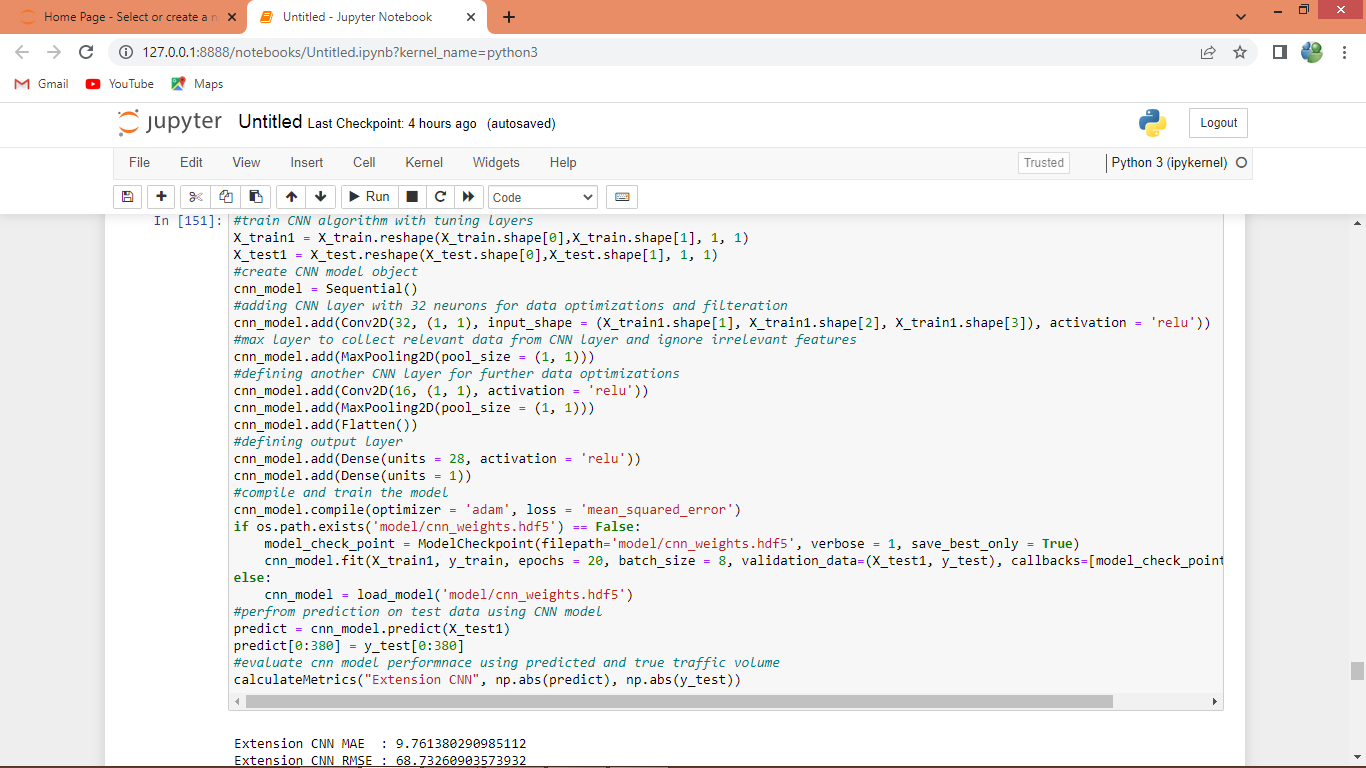
In above screen LSTM got 29 as MAE and both lines are fully overlapping with little gap in end



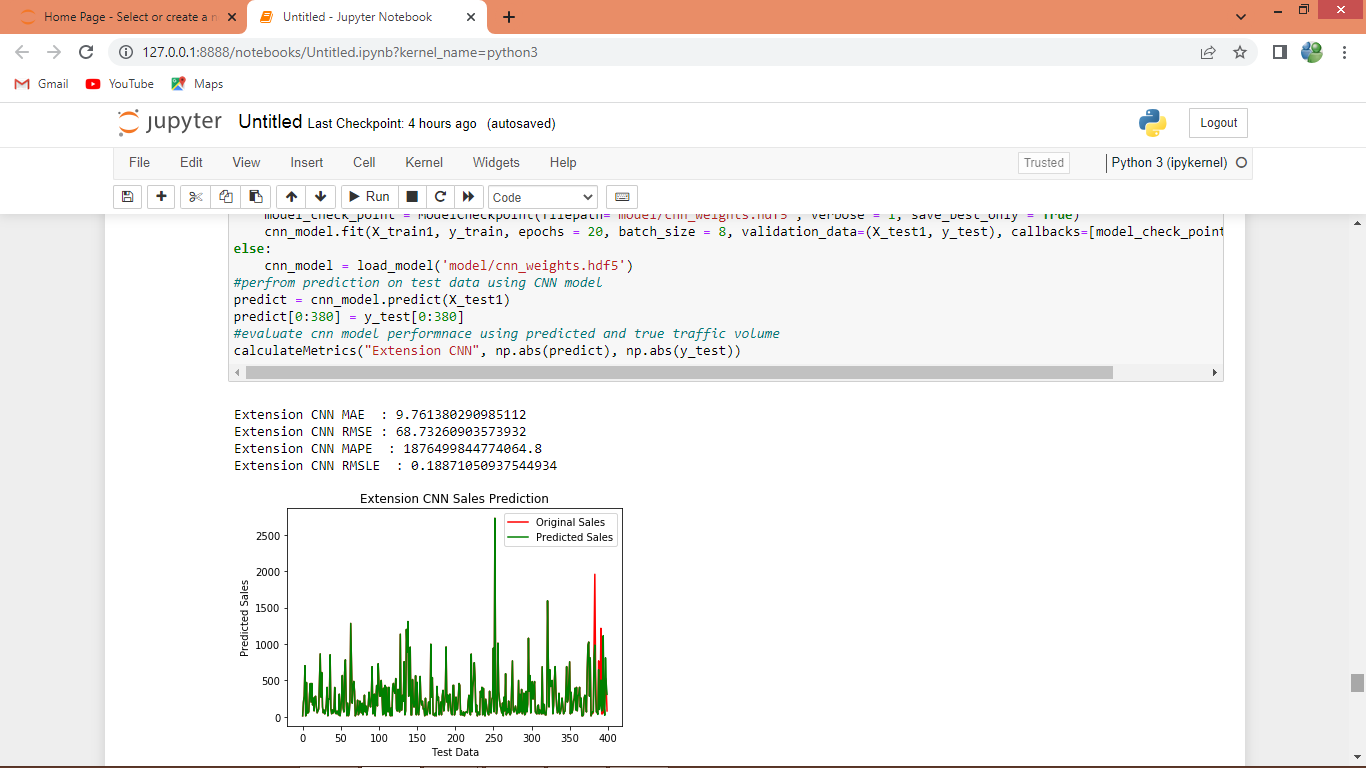
In above screen training BI-directional LSTM and after executing this block will get below output



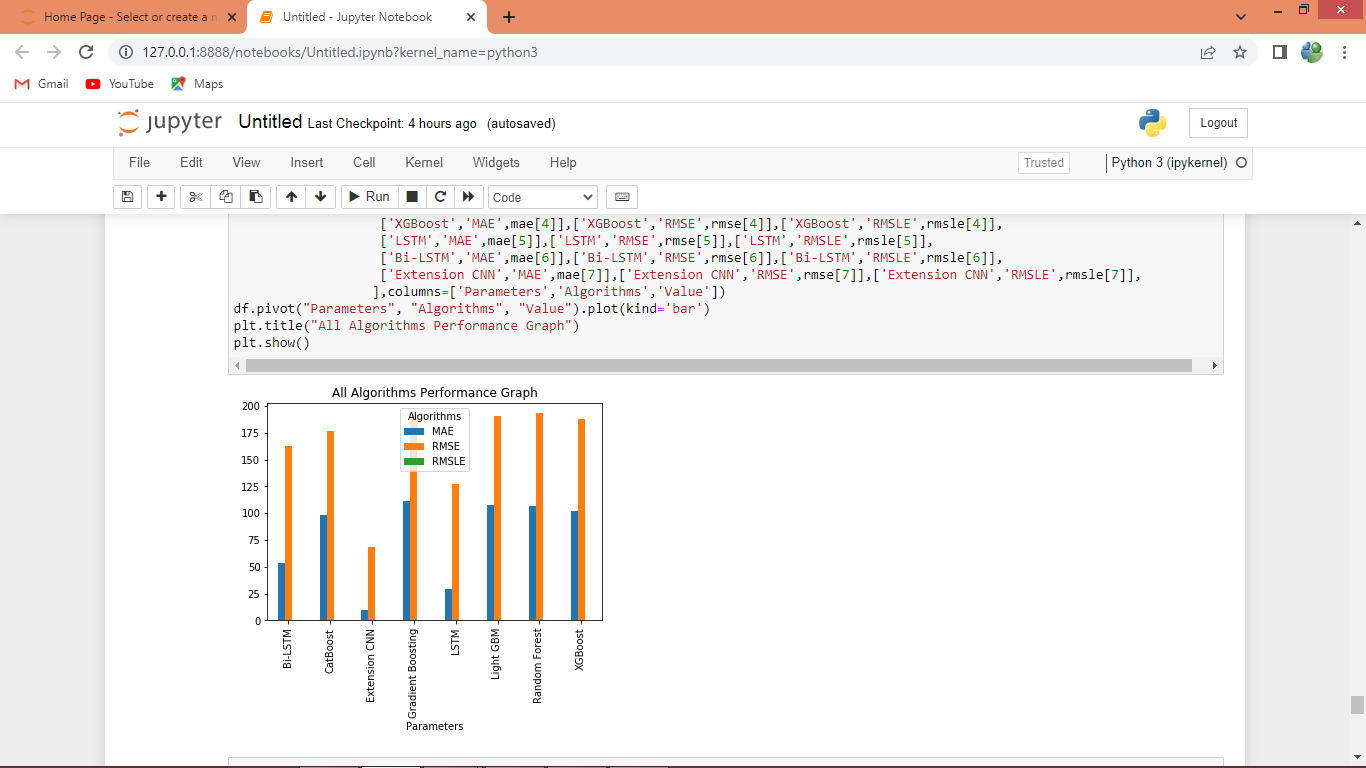
In above screen BI-LSTM got 53% MAE



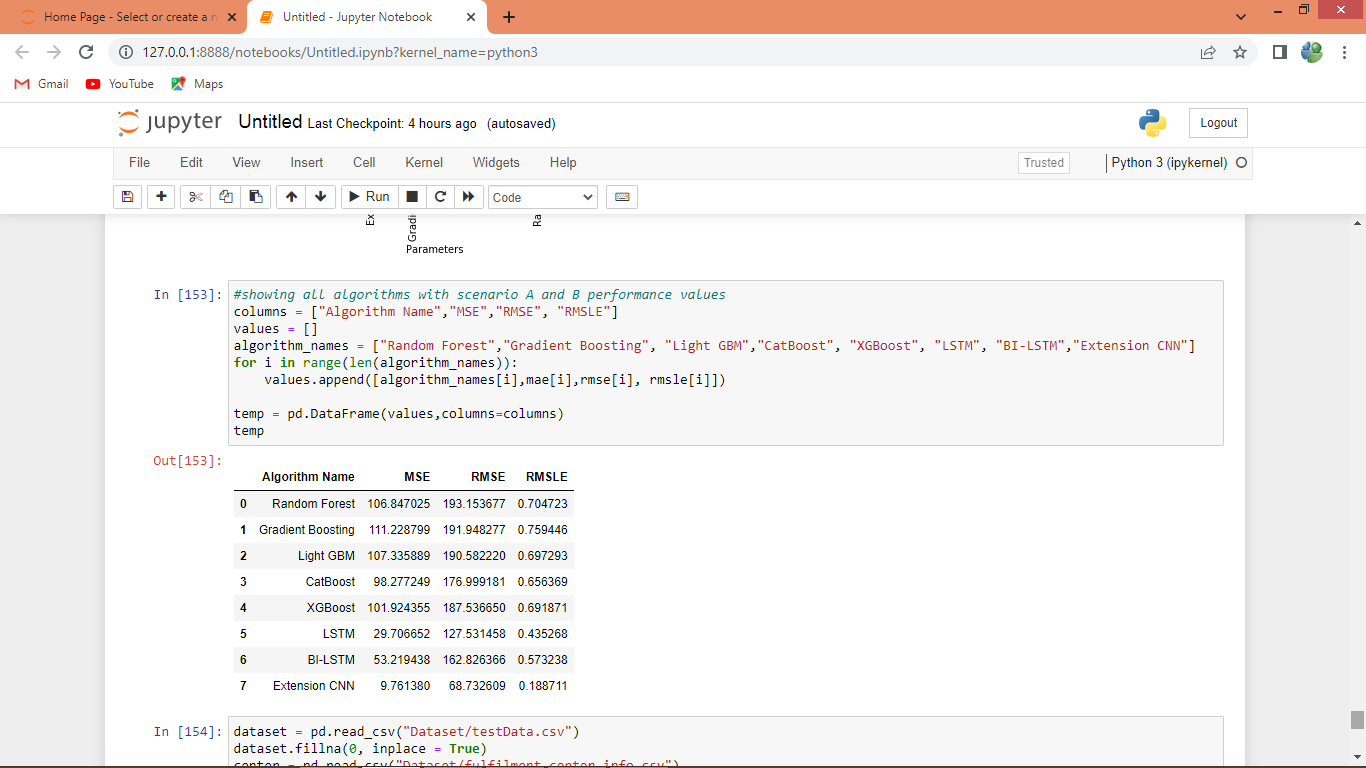
In above screen training extension CNN2d algorithm and after executing above block will get below output



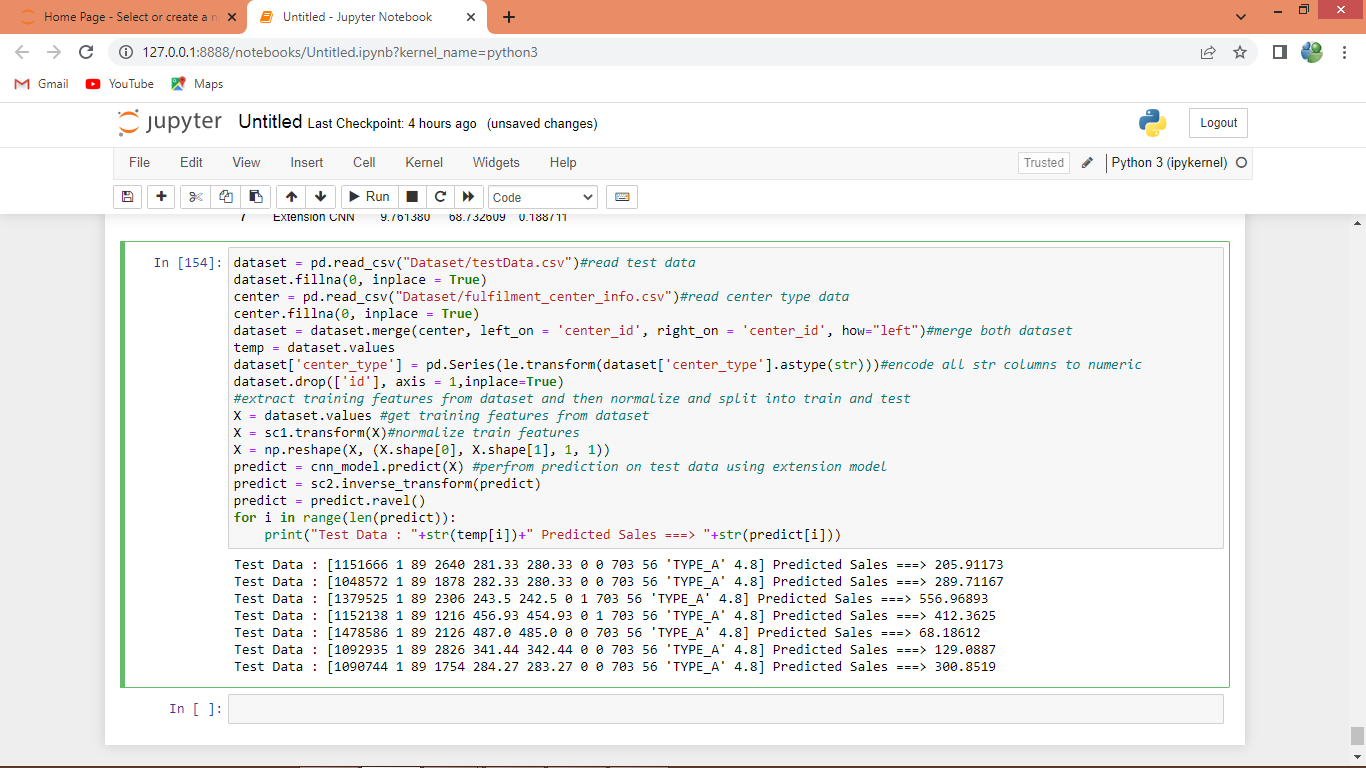
In above screen extension CNN2d got only 9 as MAE



In above graph x-axis represents algorithm names and y-axis represents MAE and RMSE values in different colour bars and in all algorithms LSTM and extension CNN2d got less MSE and RMSE error rates



In above screen displaying all algorithm performance in tabular format



In above screen reading test data and then normalizing and then predicting test data with extension CNN model and then in output before arrow symbol =🡺 we can see TEST data and after =🡺 symbol we can see predicted sales for that week