CNN-LSTM Driving Style Classification Model Based on Driver Operation Time Series Data

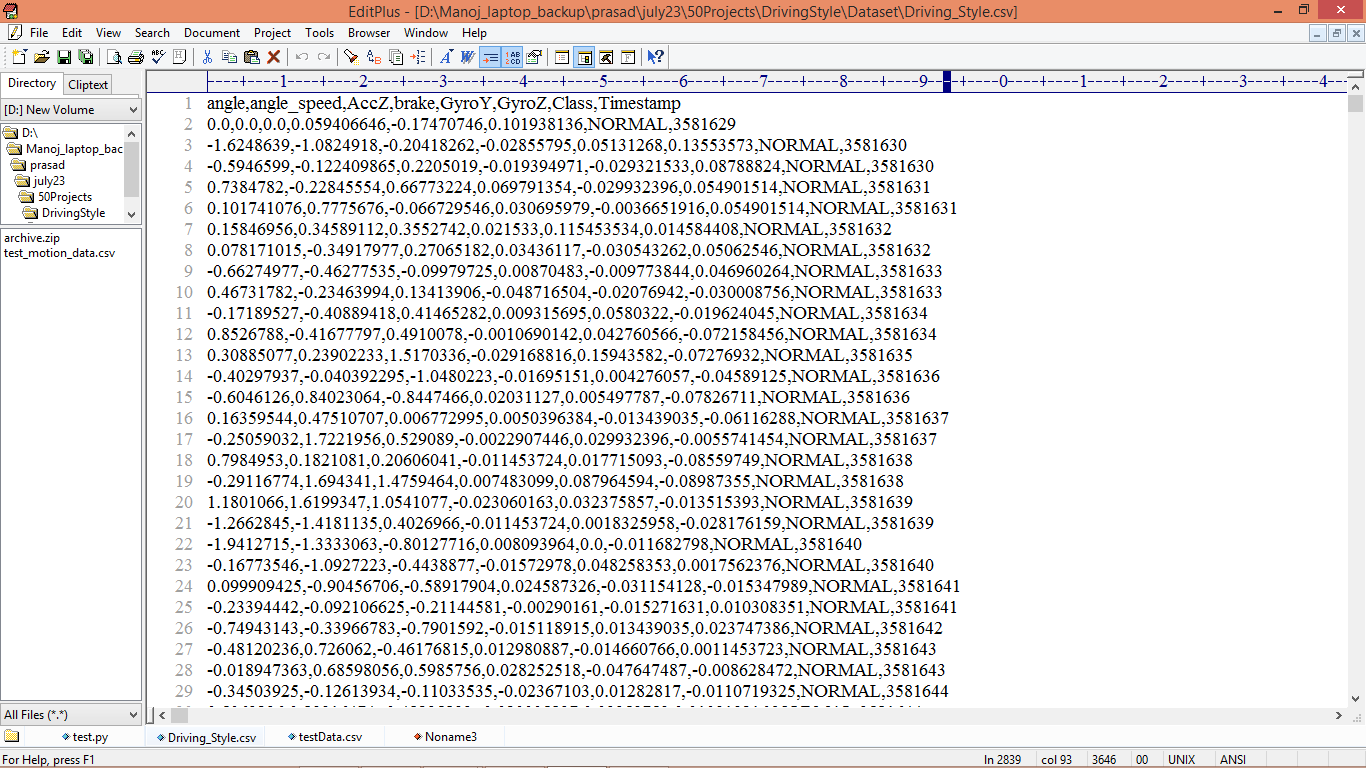
Now modern world enter into the world of self driving vehicles where tiny sensors will sense environment (road condition) or driving behaviour data and then input those data to Artificial Intelligence model which will predict vehicles next motion or will predict driver behaviour.

In past many existing algorithm was introduced such as Question Based Driver behaviour identification where software will ask question to driver for behaviour recognition, image based behaviour identification which will heavily dependent on image lighting and many more but those techniques were not accurate enough.

Later machine learning models was introduced which will take input from sensors and then predict driver behaviour but those techniques prediction accuracy is good but can be further enhance. So author of this paper employing combination of deep learning algorithms called CNN and LSTM where CNN will be used to extract features from dataset and then this extracted features will be input to LSTM to train time series based temporal data and this trained model can be used to classify driving style of the driver.

In propose work author defining two layers of CNN for features extraction and one layer of LSTM for temporal based driver style classification. In propose work author has trained another algorithm called CNN + LSTM + I by inducing sensor signals (where I represents inducing of signals) but we don’t have such signal so we cannot implement this model but we have implemented existing CNN and propose CNN + LSTM.

To train above models author has used simulated and real world driving dataset but not publish this dataset so we have used Driver Style Classification dataset from KAGGLE and below are the dataset details



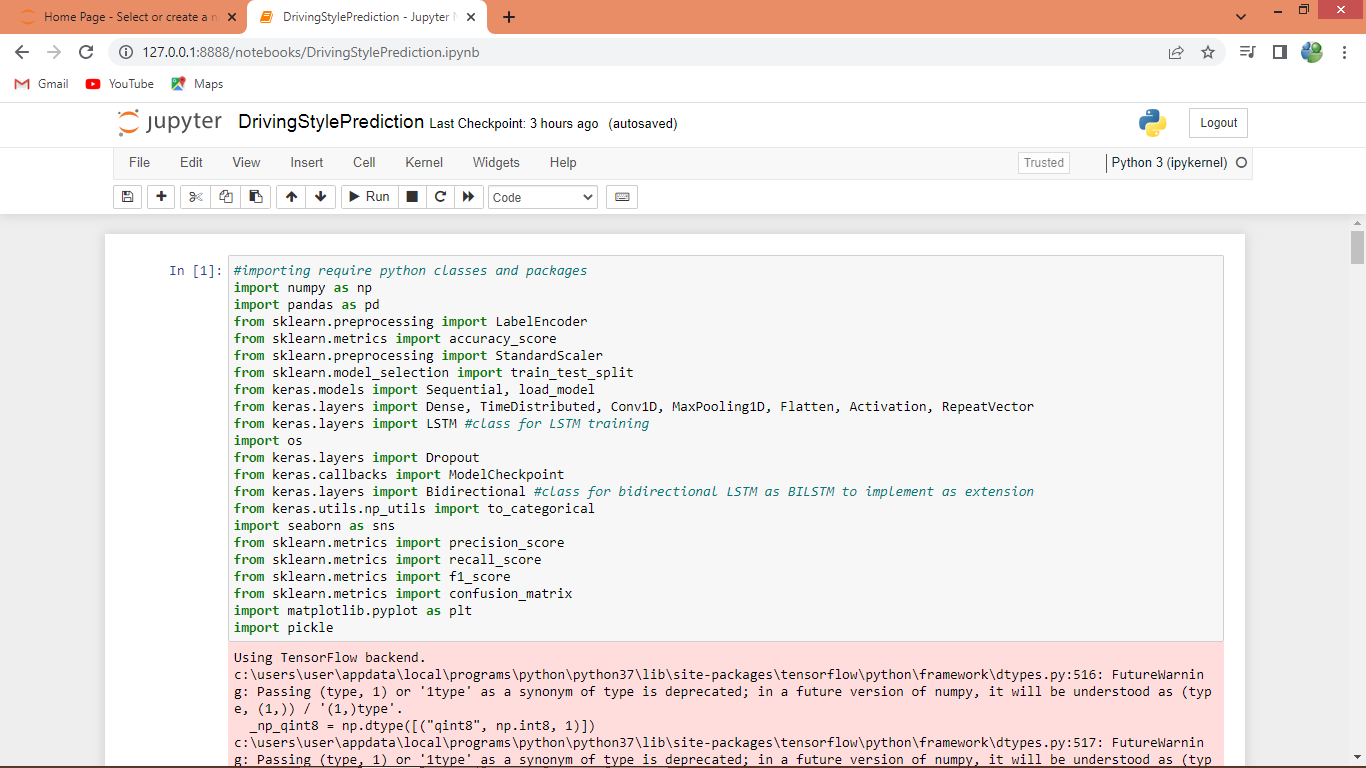
In above dataset screen first row represents dataset column names and remaining rows represents dataset values and this dataset has 3 different class labels called ‘Normal, Aggressive and Conservative’. So by using this dataset we will train and test all algorithms.

Extension Concept

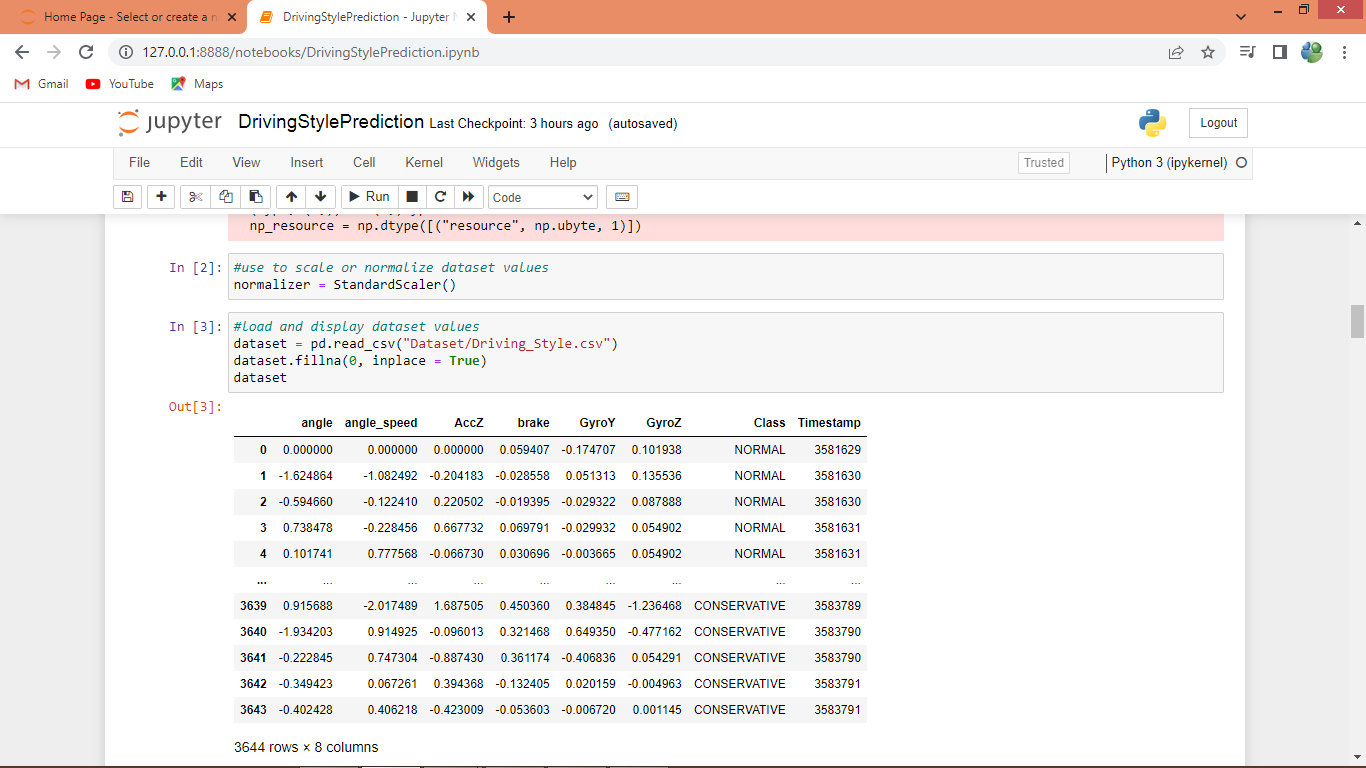
In propose work to manage temporal time series dataset we have another advanced algorithm called BI-Directional LSTM but in propose work author has not used this model so as extension we have added this model with LSTM layer and this model help us in further improving thus this model is called as CNN + LSTM + BI-LSTM.

SCREEN SHOTS

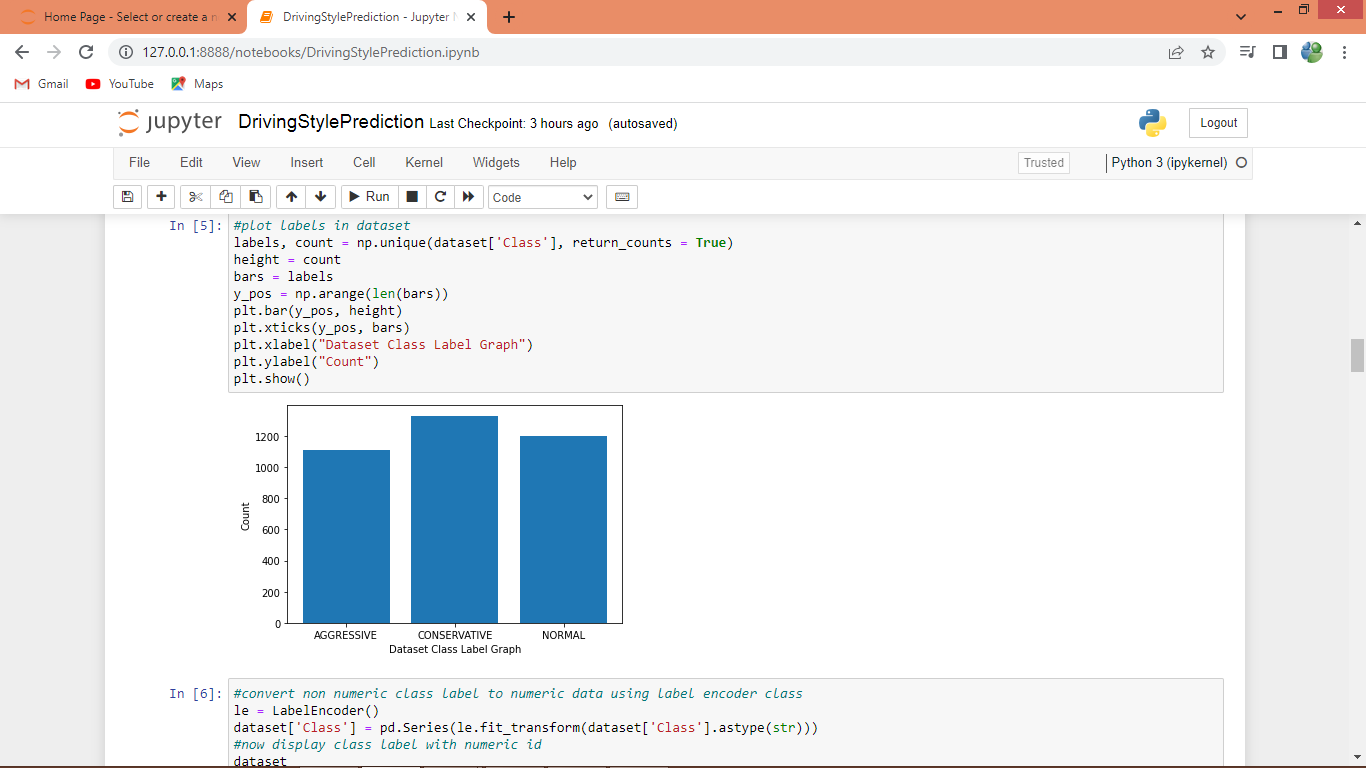
To implemented this project we have used JUPYTER notebook and below is the code and output screen with blue colour comments



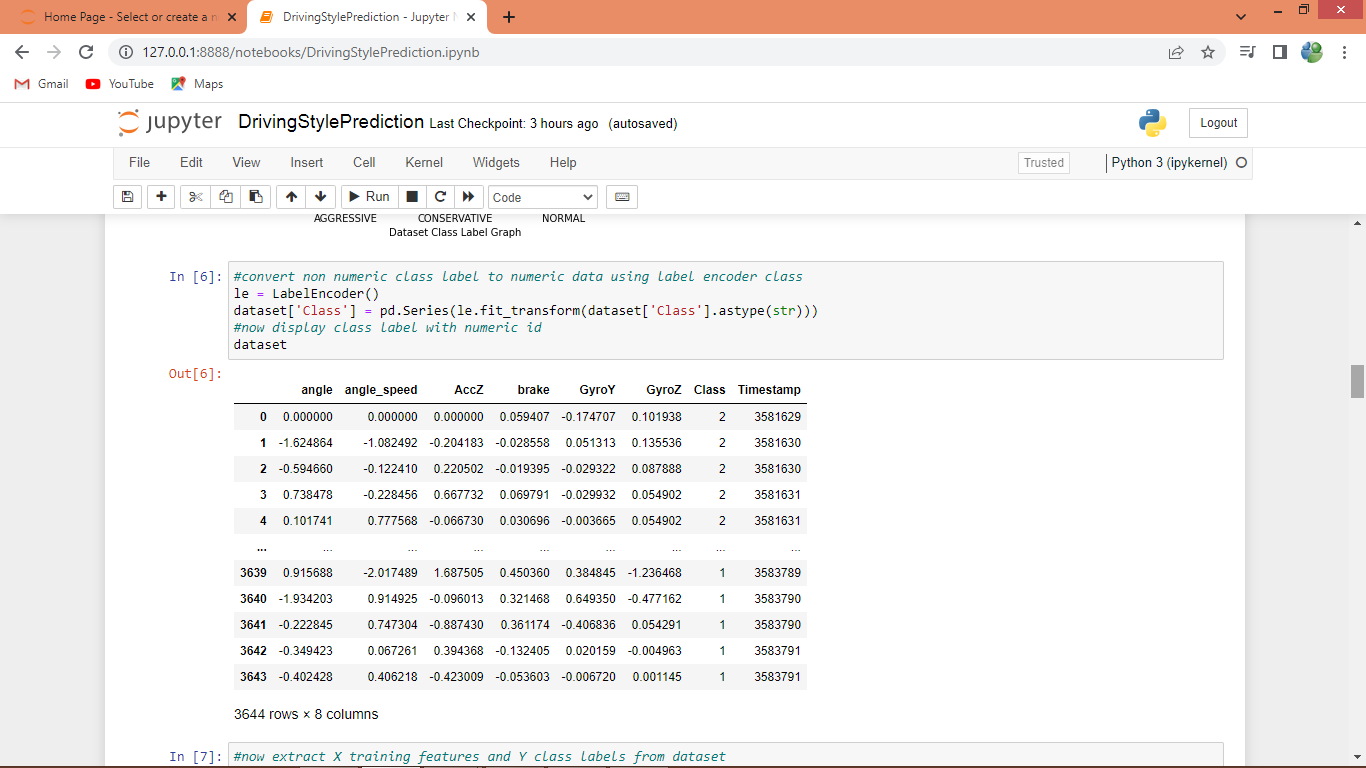
In above screen we are importing required python classes and packages



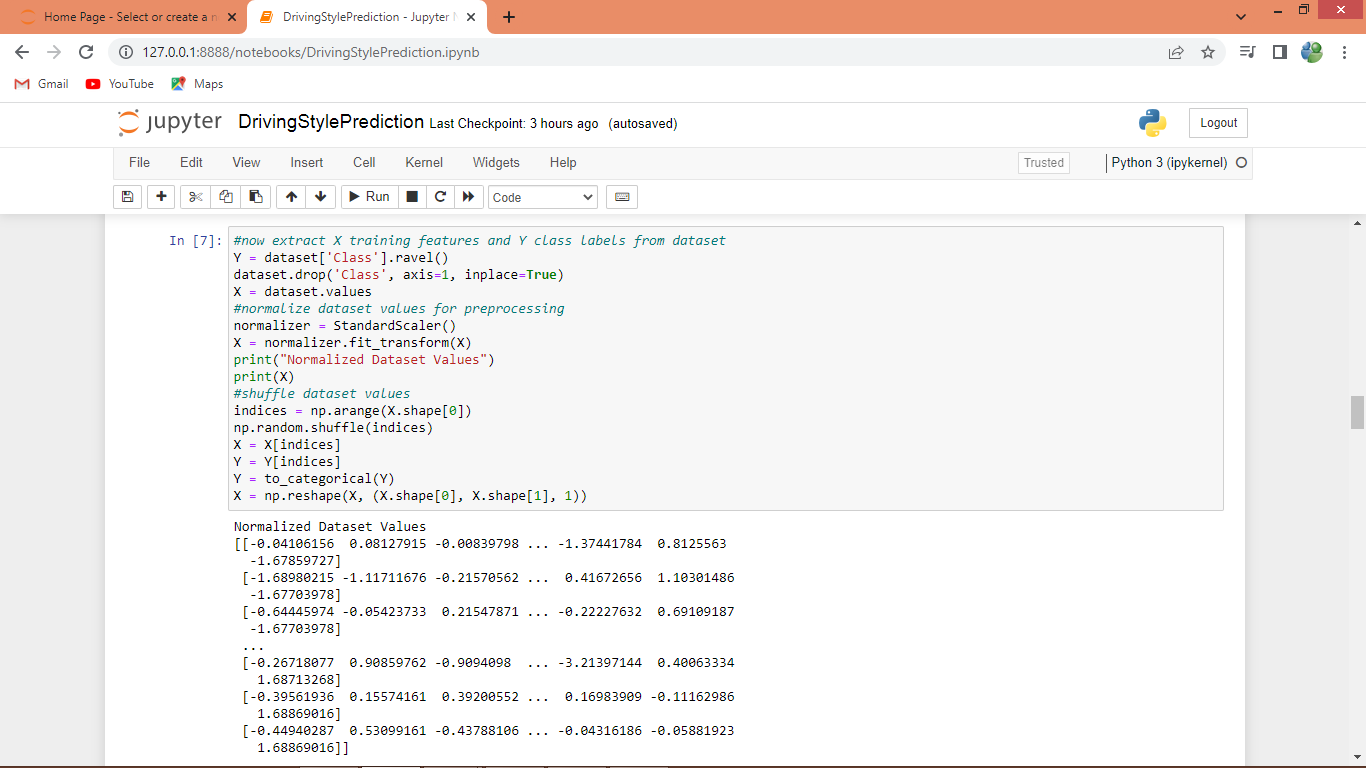
In above screen defining object to normalized dataset and then reading and displaying dataset values



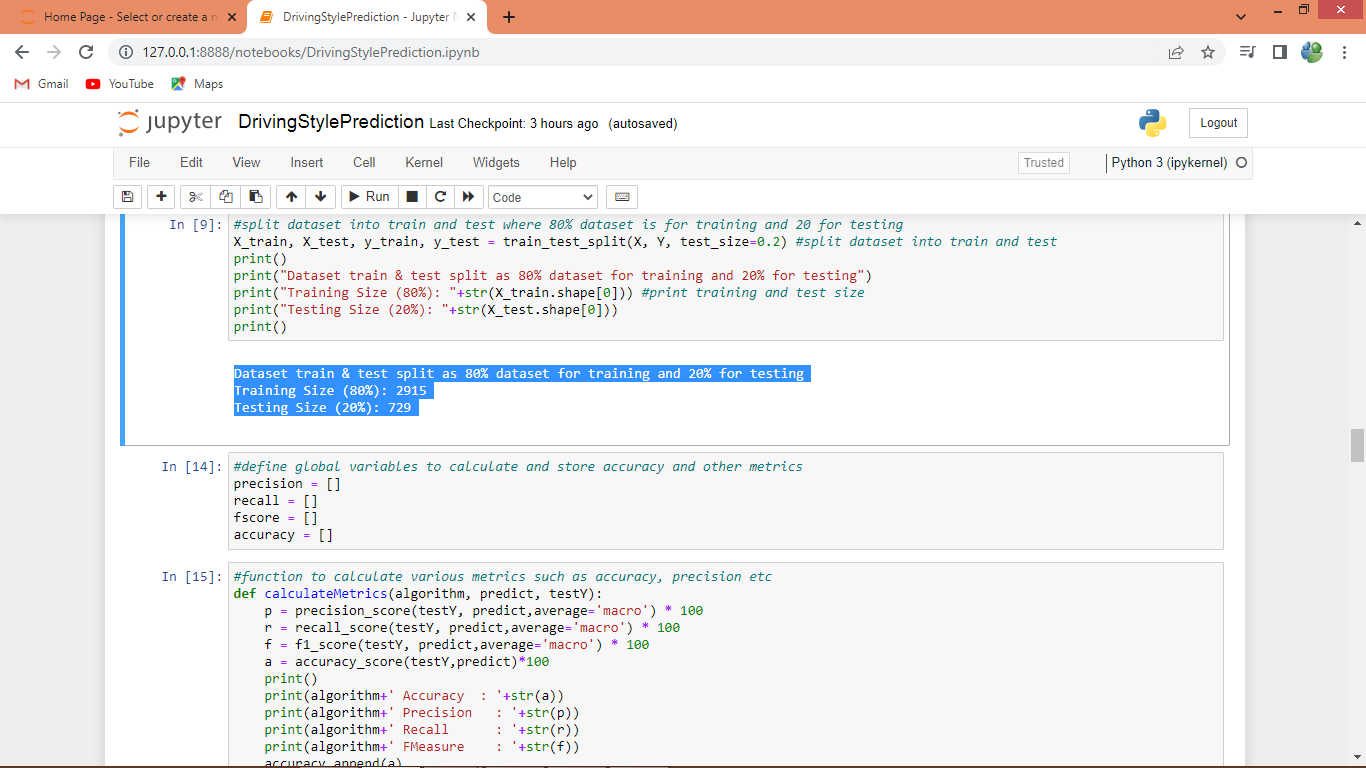
In above screen we are finding and plotting graph of various class labels found in dataset here x-axis represents class label name and y-axis represents count



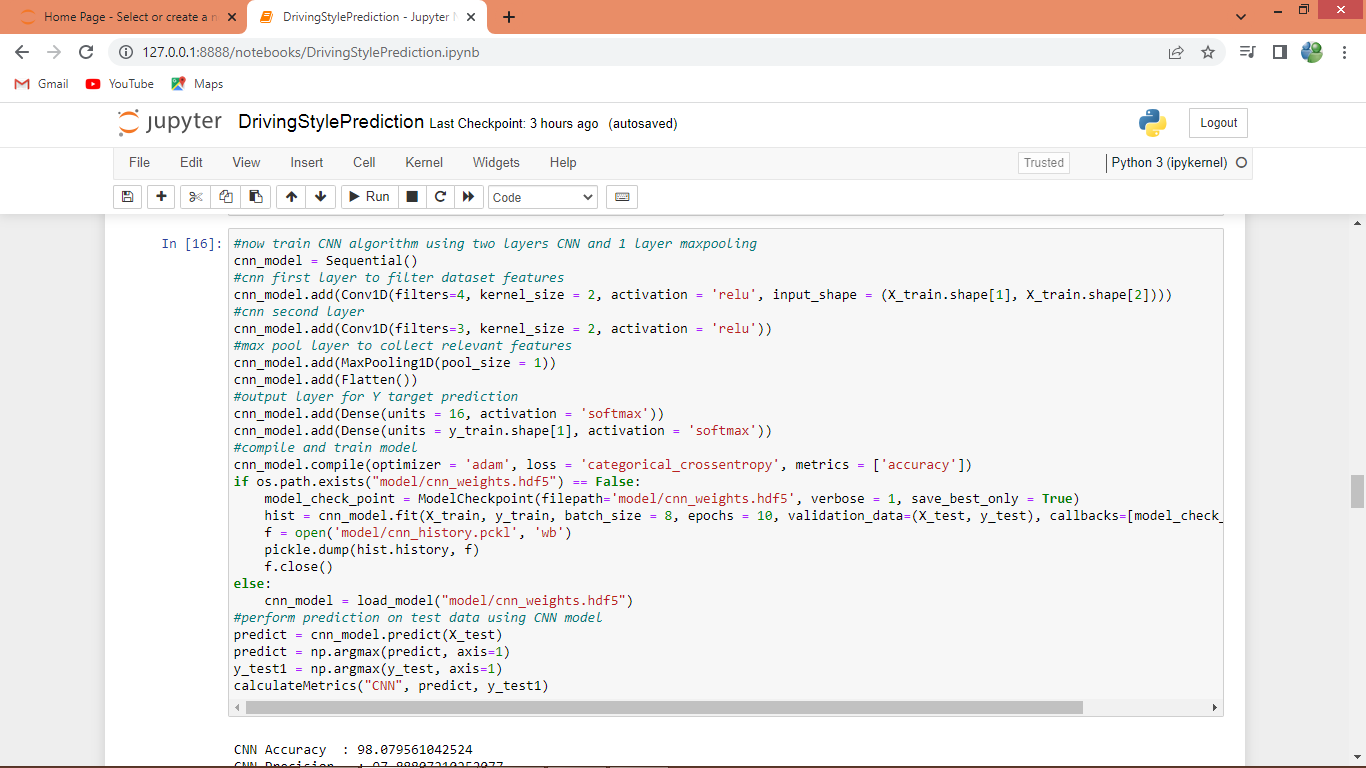
In dataset class labels are not in numeric format so by employing Label Encoder class we have converted non-numeric labels to numeric labels and in above output we can see all values of dataset are in numeric format



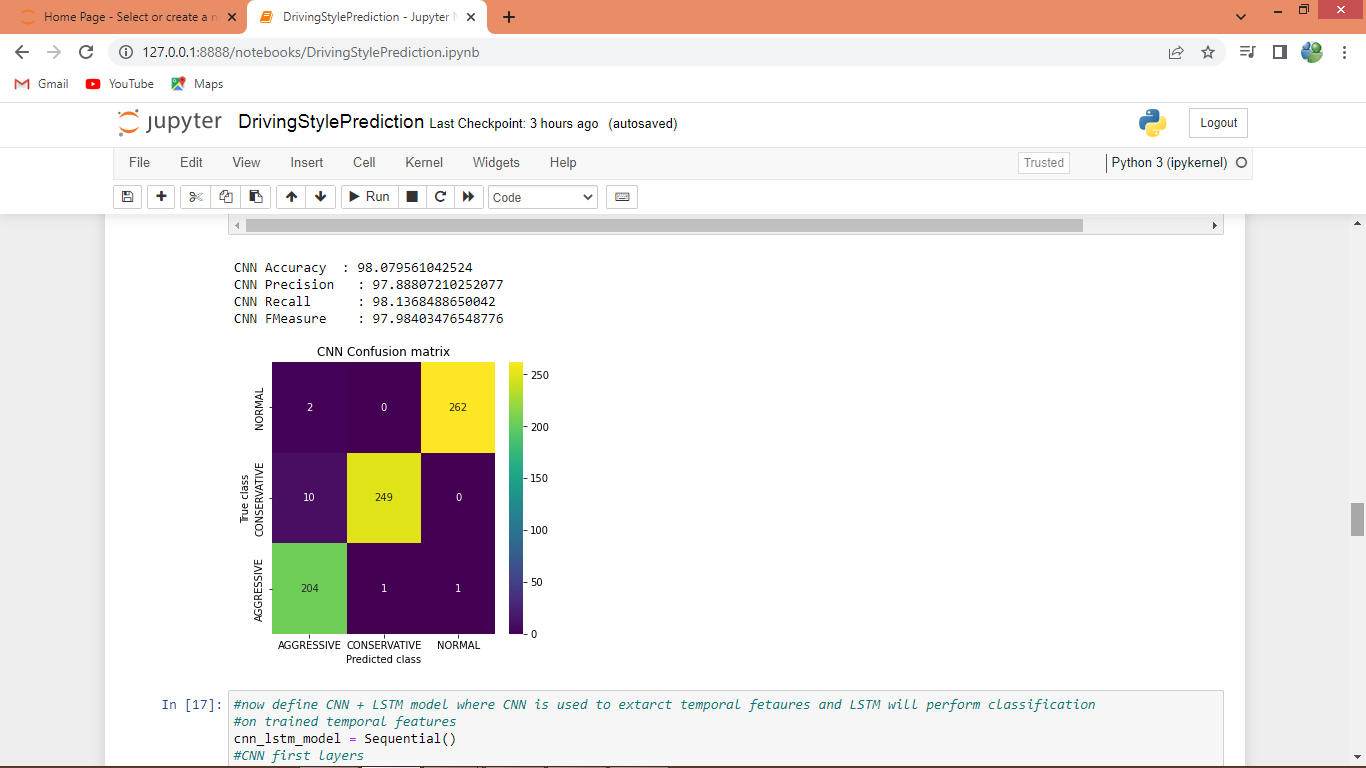
In above screen performing dataset pre-processing such as shuffling, normalizing and displayed normalize dataset values



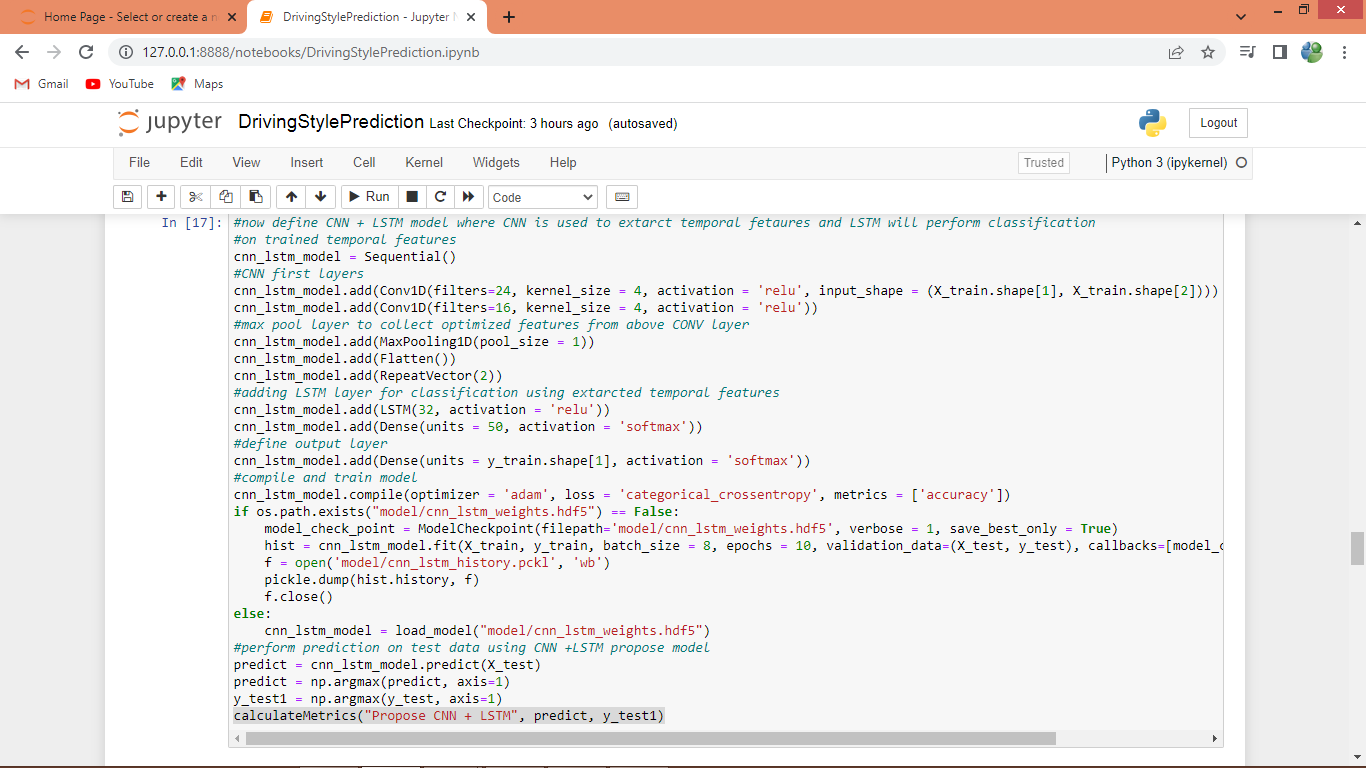
In above screen splitting dataset into train and test where 80% dataset used for training and 20% for testing and then defining function to calculate accuracy and other metrics



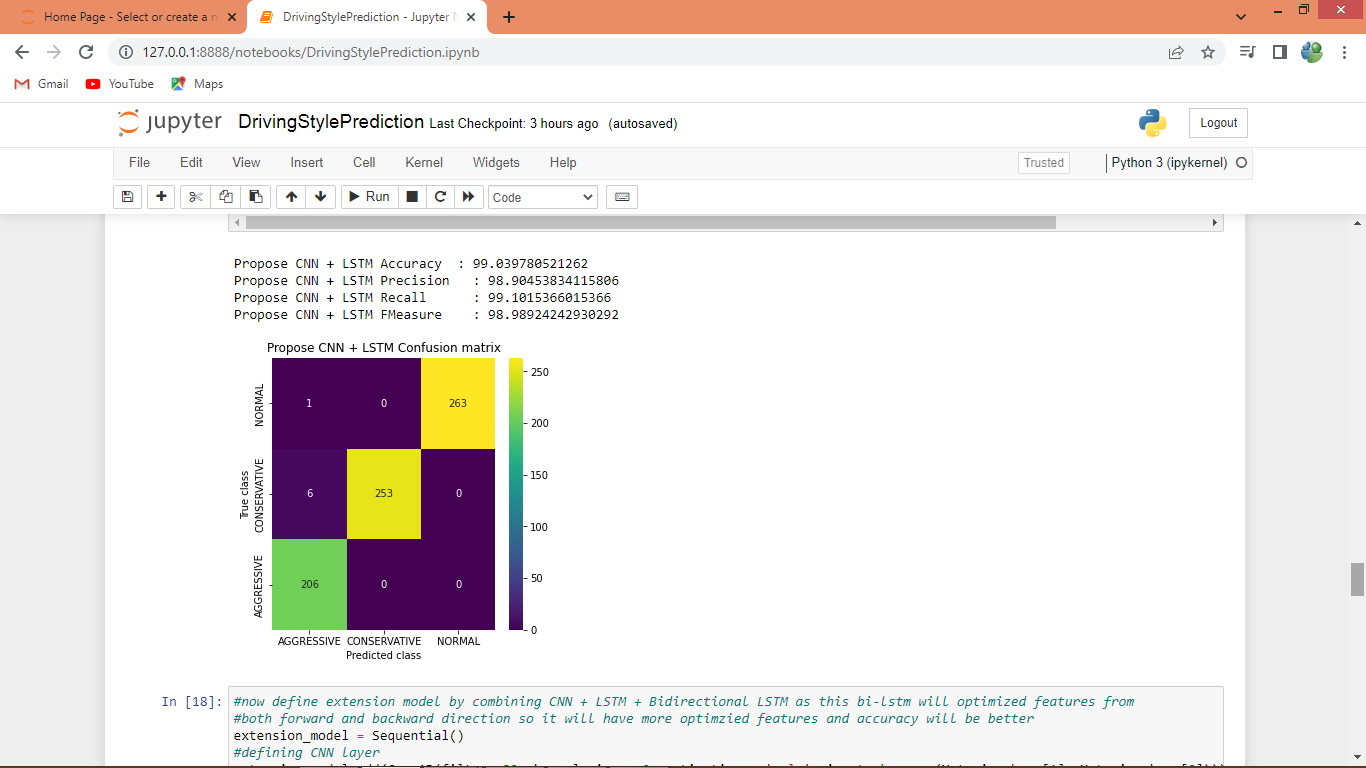
In above screen defining CNN model and its layers and after executing above block will get below output



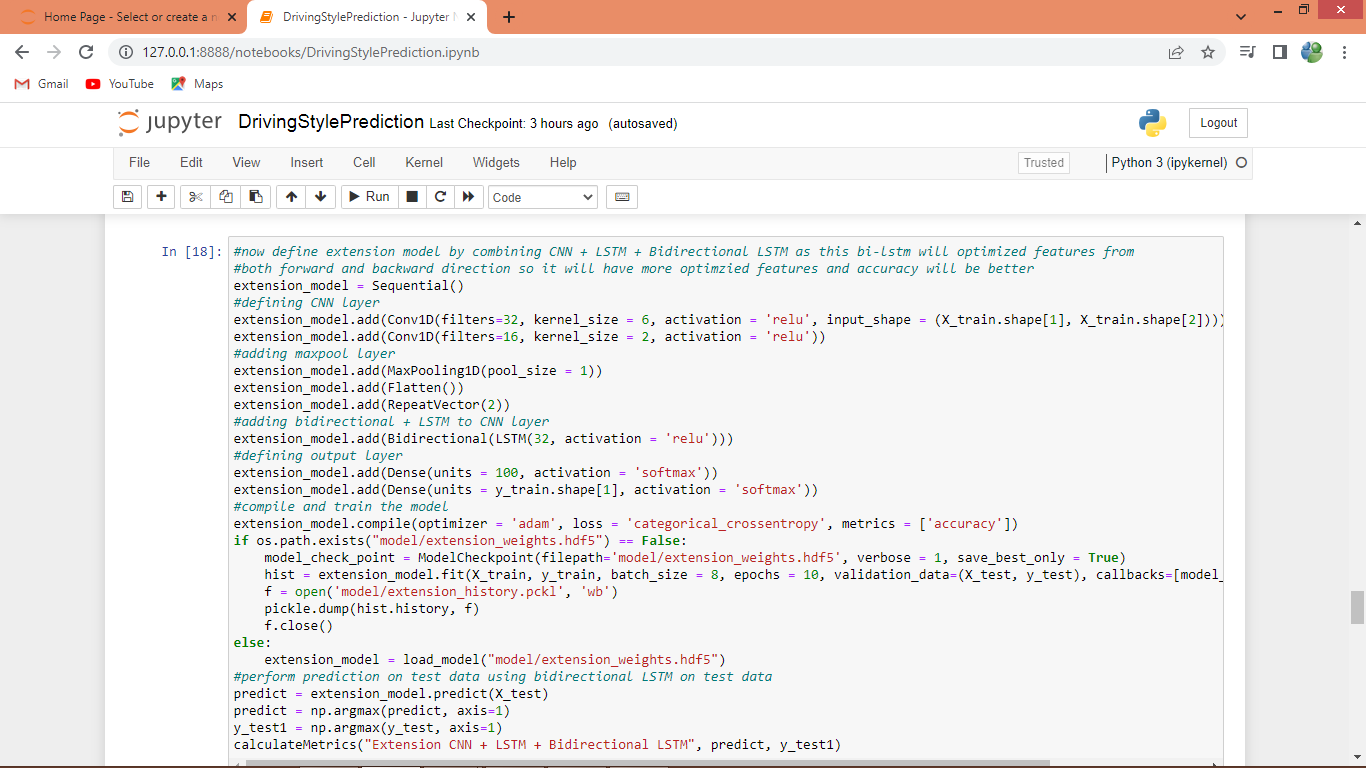
In above screen CNN got 98% accuracy and we can see other metrics and in confusion matrix graph x-axis represents Predicted Labels and y-axis represents True Labels. All different colour boxes in diagnol represent correct prediction count and remaining blue boxes represents incorrect prediction count which is very few.



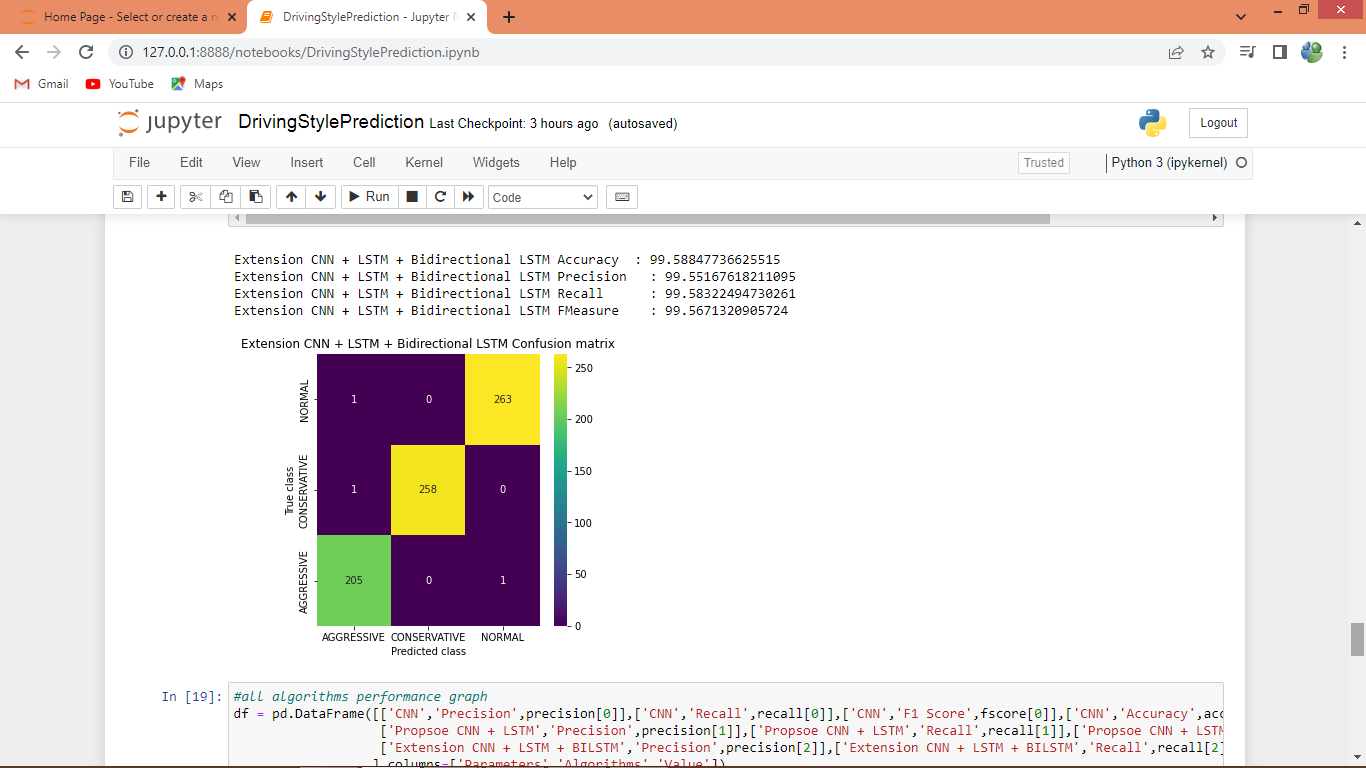
In above screen we are training propose CNN with LSTM layer and after executing above block will get below output



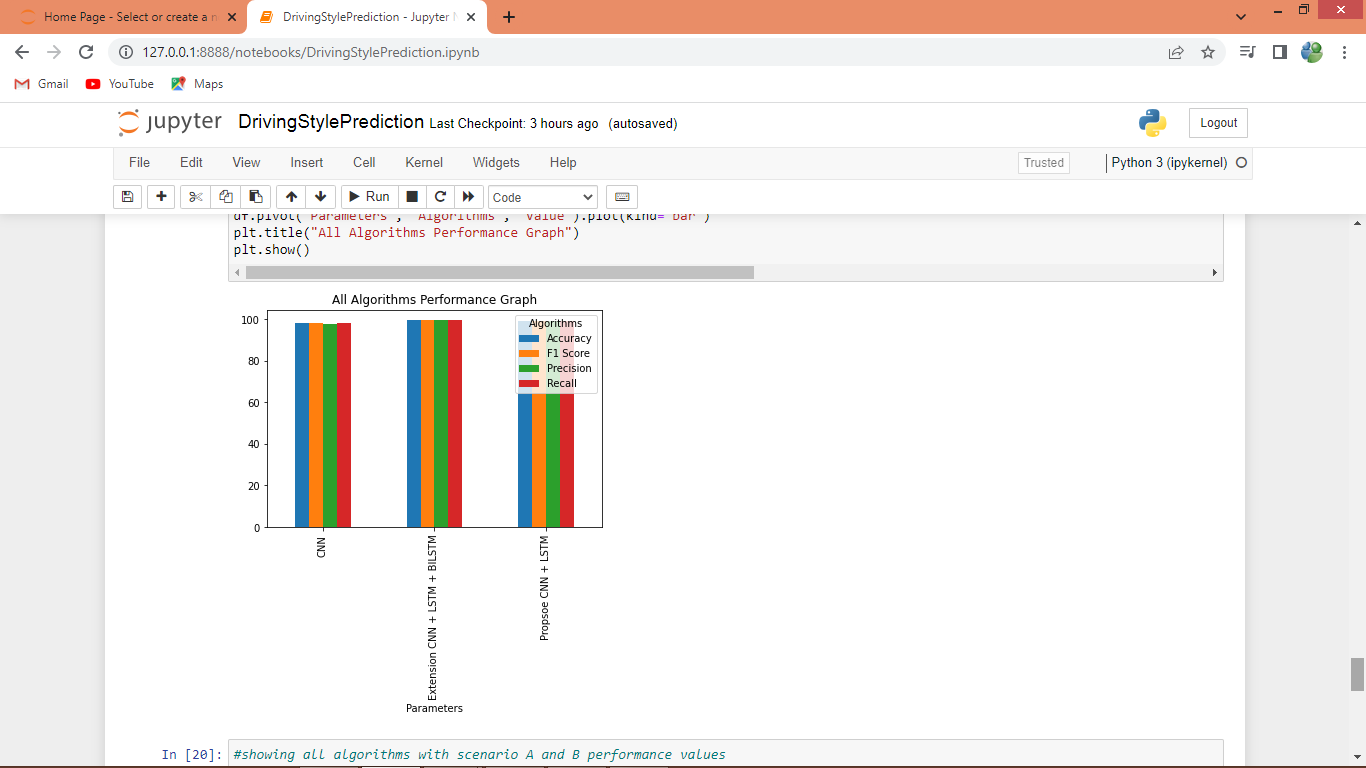
In above screen propose CNN +LSTM got 99.03% accuracy



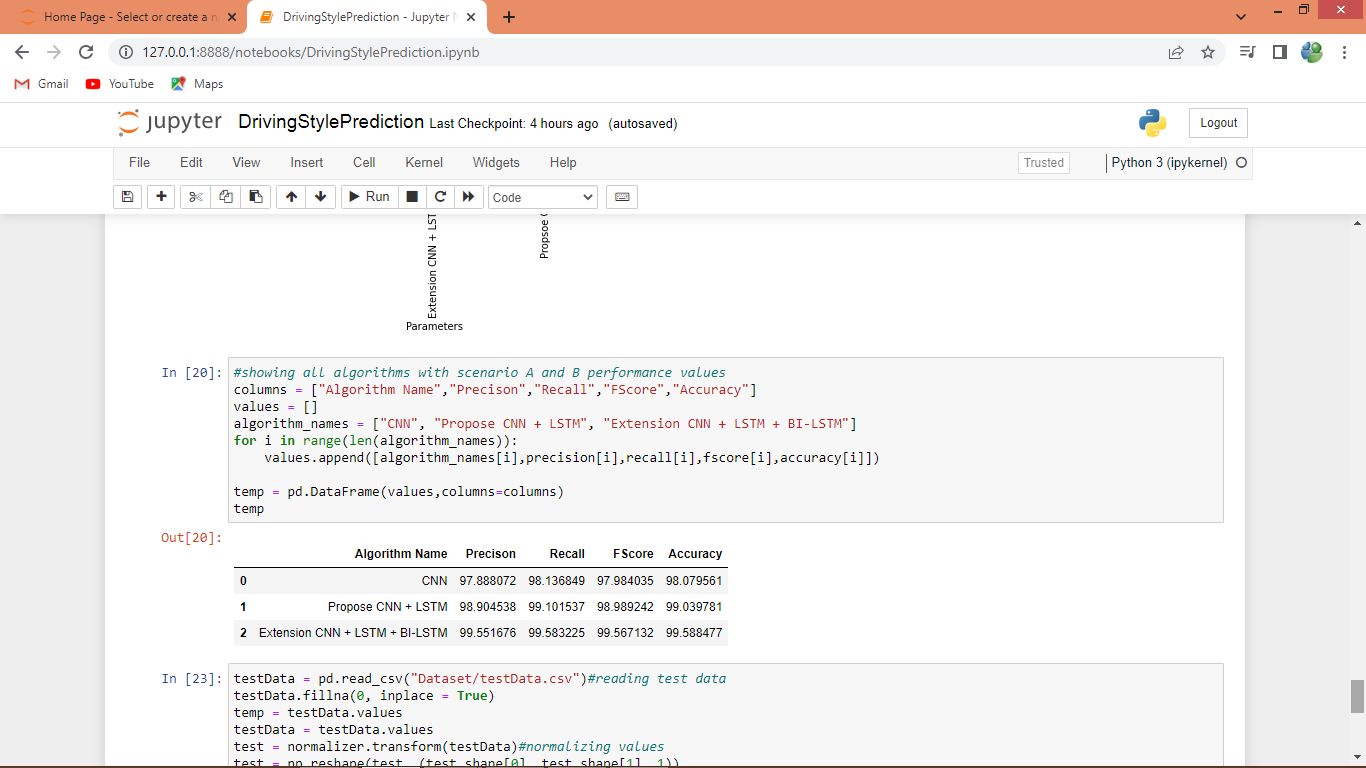
In above screen training with extension CNN + LSTM + Bidirectional LSTM and after executing above block will get below output



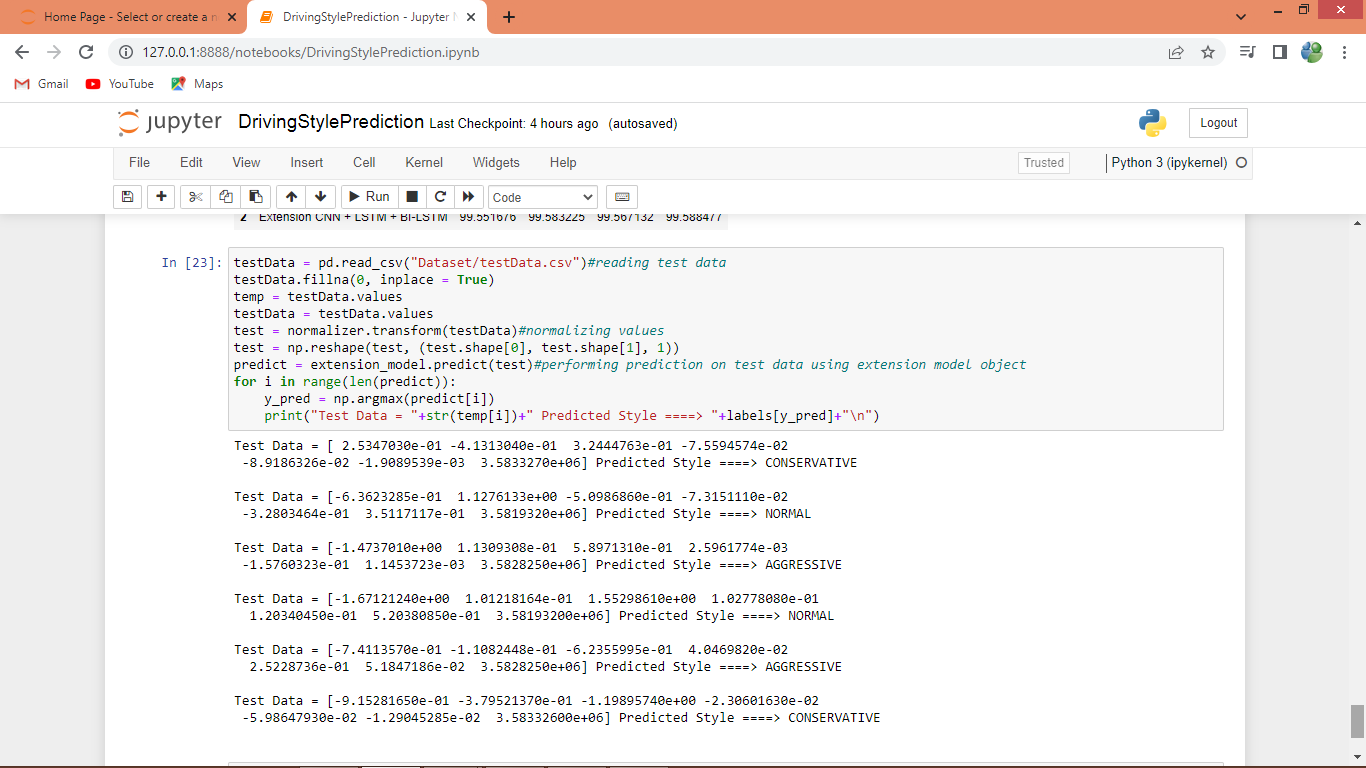
In above screen extension algorithm got 99.58% accuracy which is higher than all algorithms



In above graph x-axis represents algorithm names and y-axis represents accuracy and other metrics in different colour bars and in all algorithms extension got high accuracy



In above screen showing all algorithms performance in tabular format



In above screen reading test data and then predicting with extension model and in prediction output before arrow symbol we can see test data and after arrow symbol we can see predicted Driving Style as Normal, Aggressive or Conservative