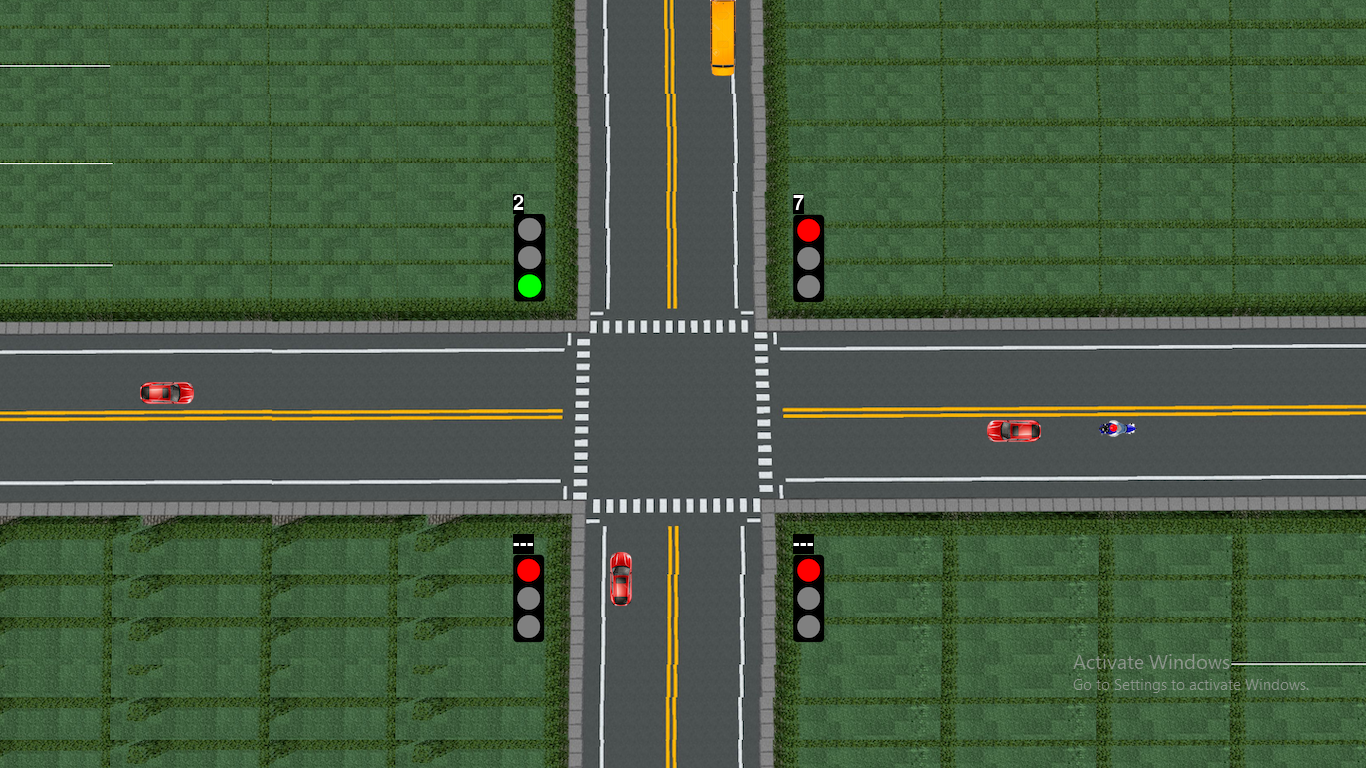
Traffic Congestion and Accident Prevention Analysis for Connectivity in Vehicular Ad-hoc Network

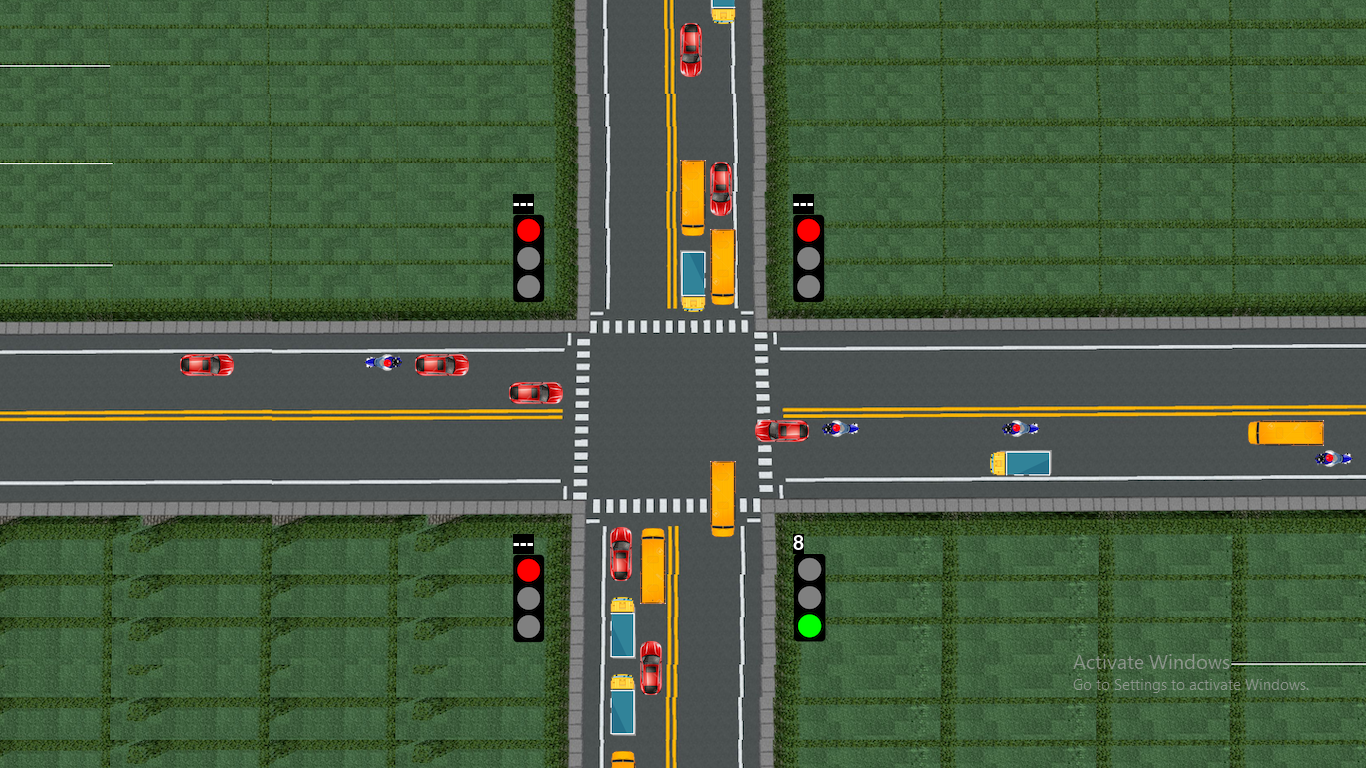
As propose work you combine multiple works like predicting traffic route based on weather condition, congestion and then mention about adjusting traffic signal based on number of vehicles, detecting accident severity and adjusting traffic lane by widening and narrowing.

From above topics all points are possible accept lane narrowing and widening as this depend on hardware not software. Rest of the implementation which is describing below

1. We have trained machine learning model by using weather and traffic congestion data so ML can predict or suggest new route to driver based on traffic congestion and weather condition
2. By using another accident dataset we have trained machine learning algorithm to predict accident severity and to train model we have used ‘US Accident Severity dataset’
3. To adjust traffic signal time we have utilized PYGAME simulator which will monitor number of vehicles arrived at the signal and based on number of vehicles it will adjust signal time and based on number of vehicles it will give priority to only that route which has more vehicles.
4. Road narrowing and widening is not possible via software.

In below screen we are showing simulation output of traffic signal and this simulation you can run by double click on ‘runSimulation.py’ file

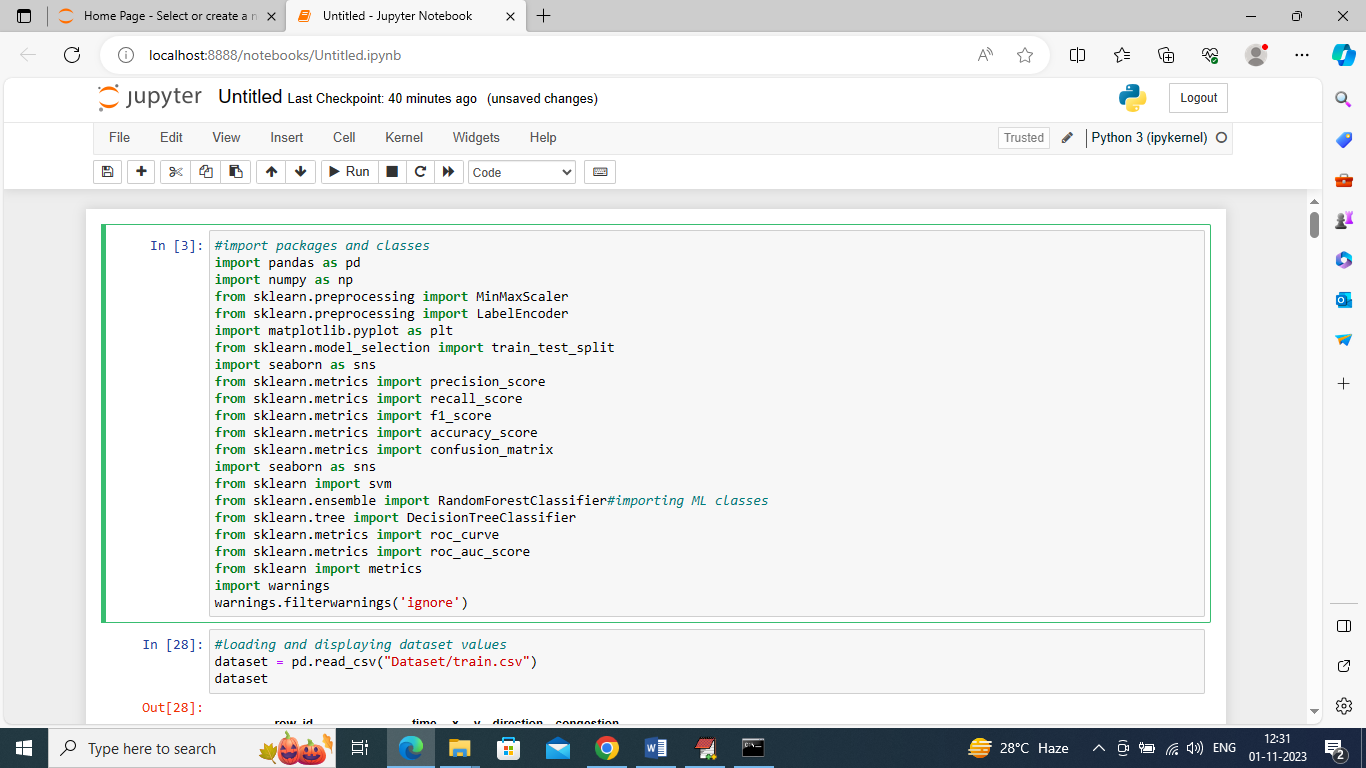




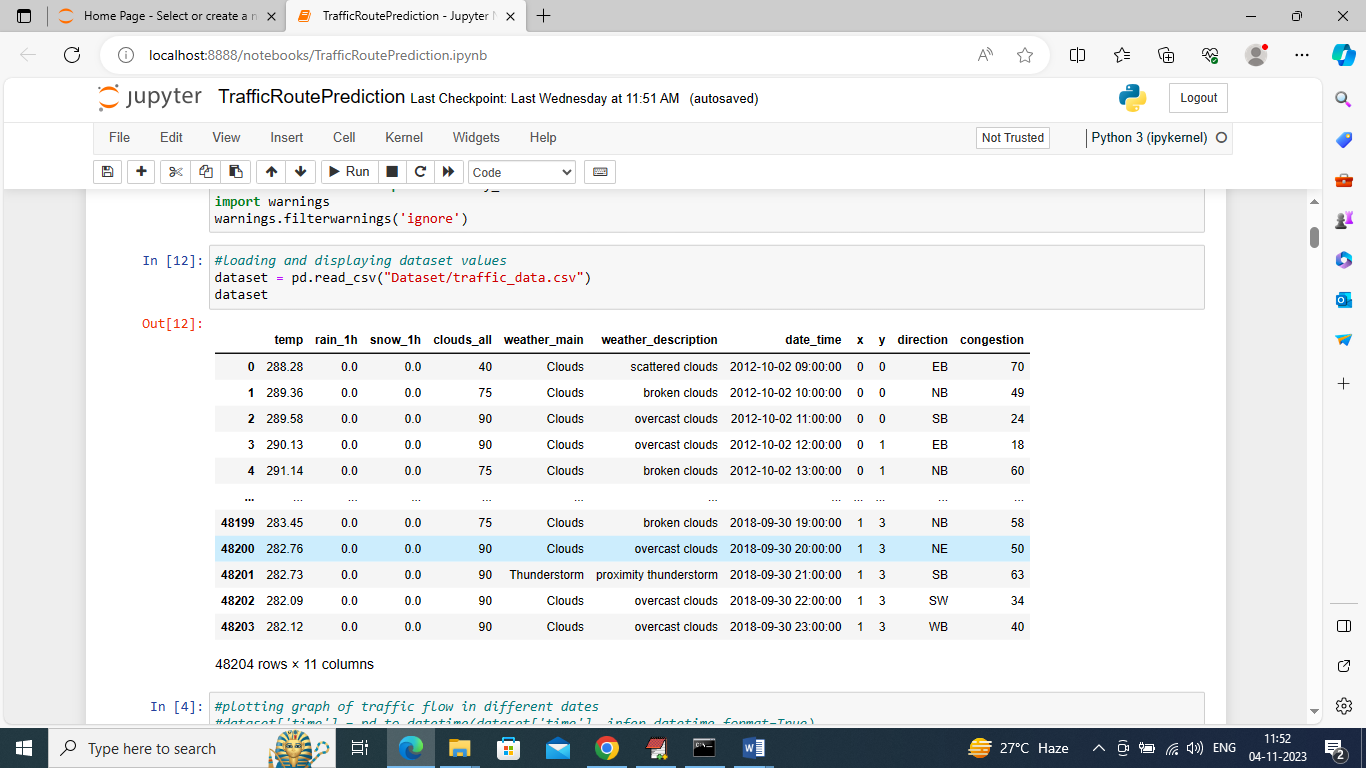
In above screen you can see PYGAME traffic signal simulation output and at each lane traffic density is calculated and then adjust green and red signal. This simulation run in INFINITE loop and for run it will adjust signal based on number of vehicles arrived and to stop simulation you can press ‘windows’ key from keyboard and then close application.

In below screen we are showing output for route prediction which we done using ML algorithm and through JUPYTER notebook

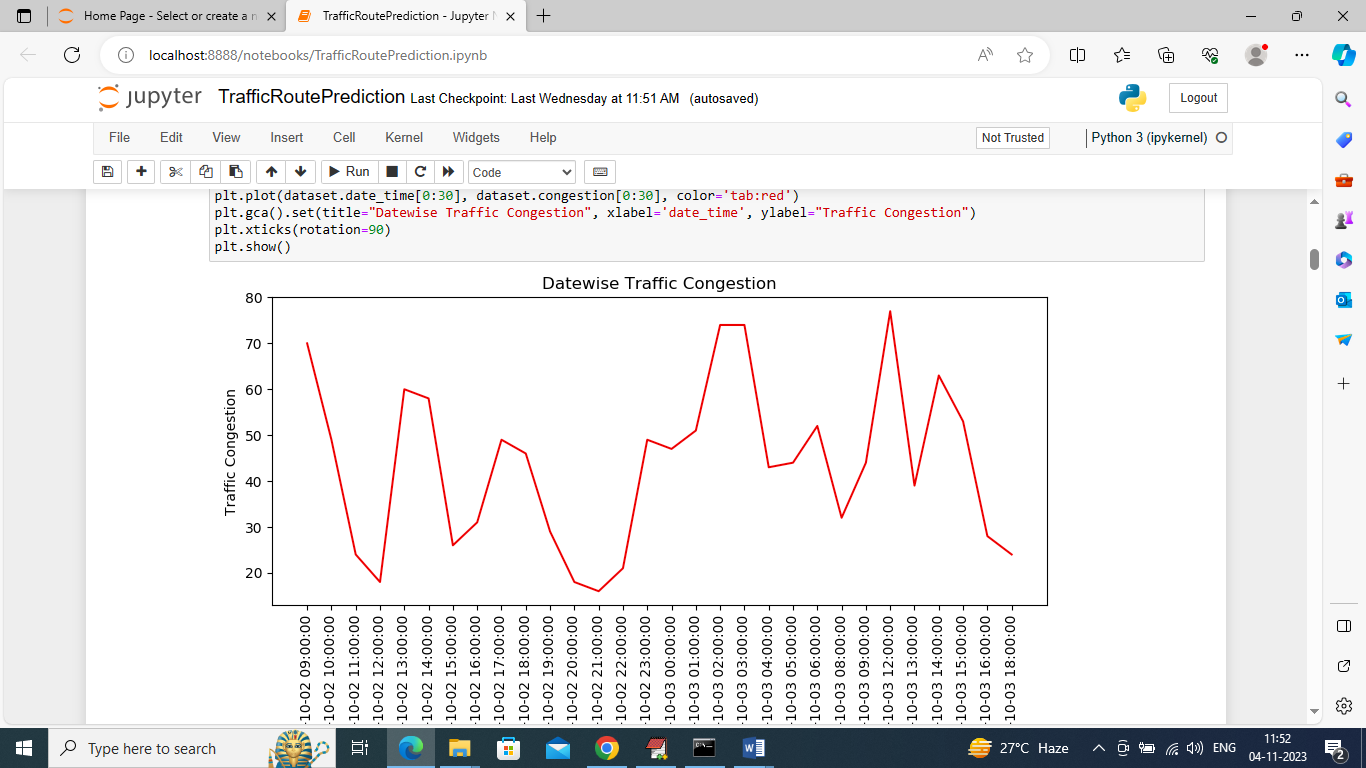
We have coded this project using JUPYTER notebook and below are the code and output screens with blue color comments



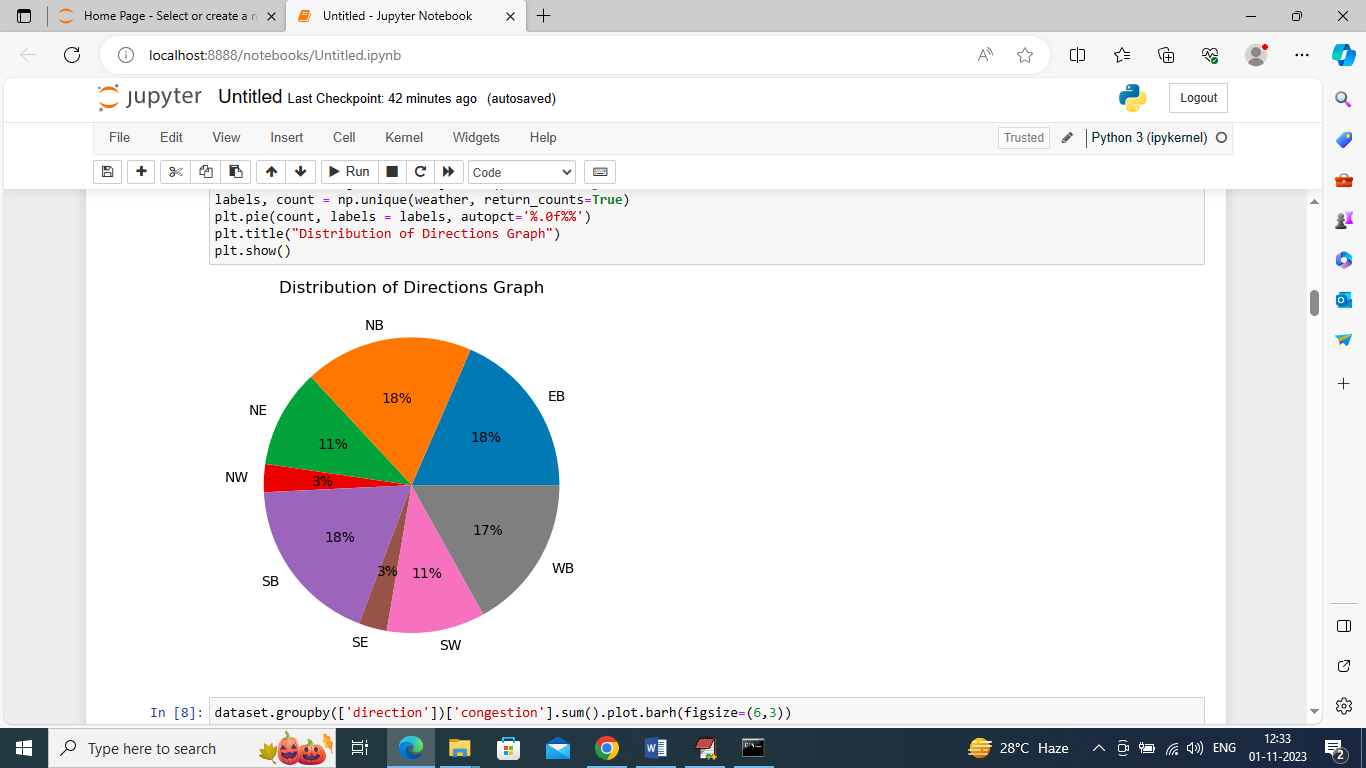
In above screen we are importing require python classes and packages



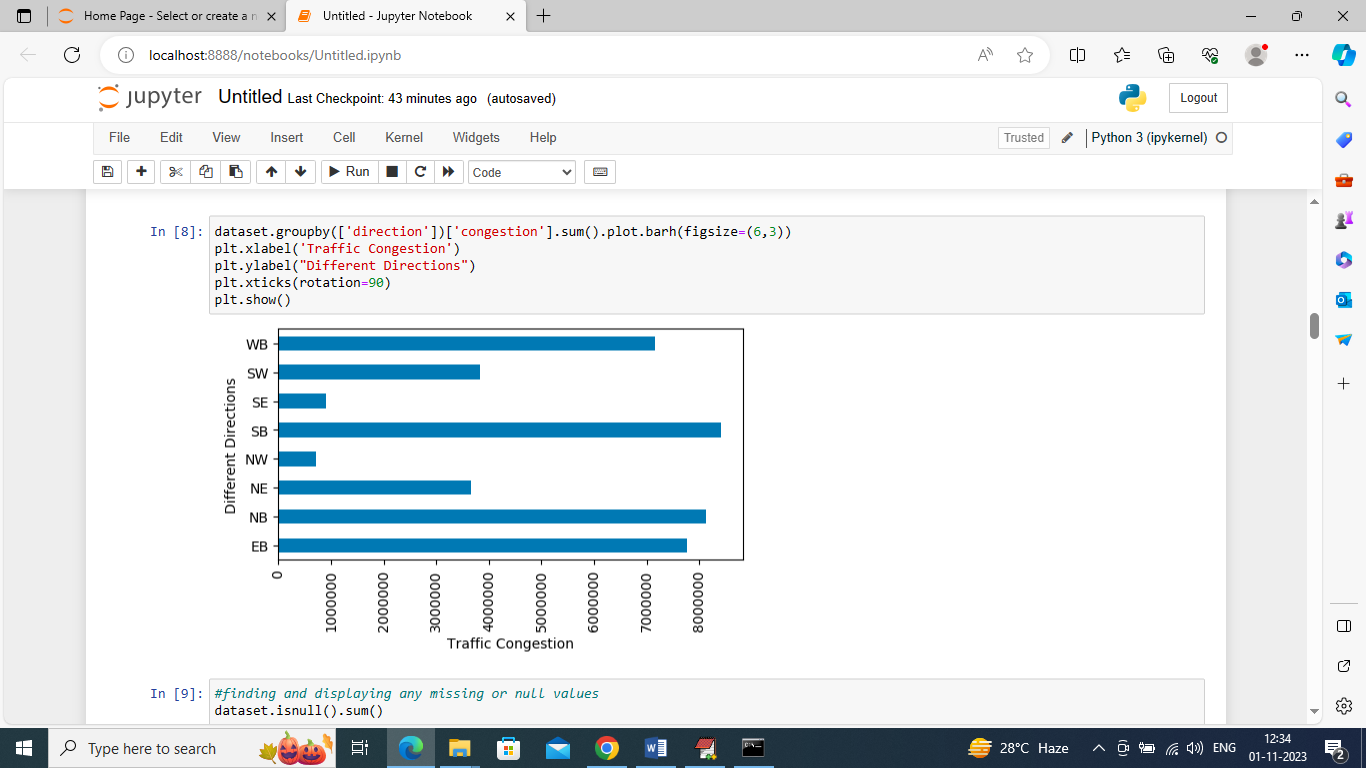
In above screen loading and displaying dataset values



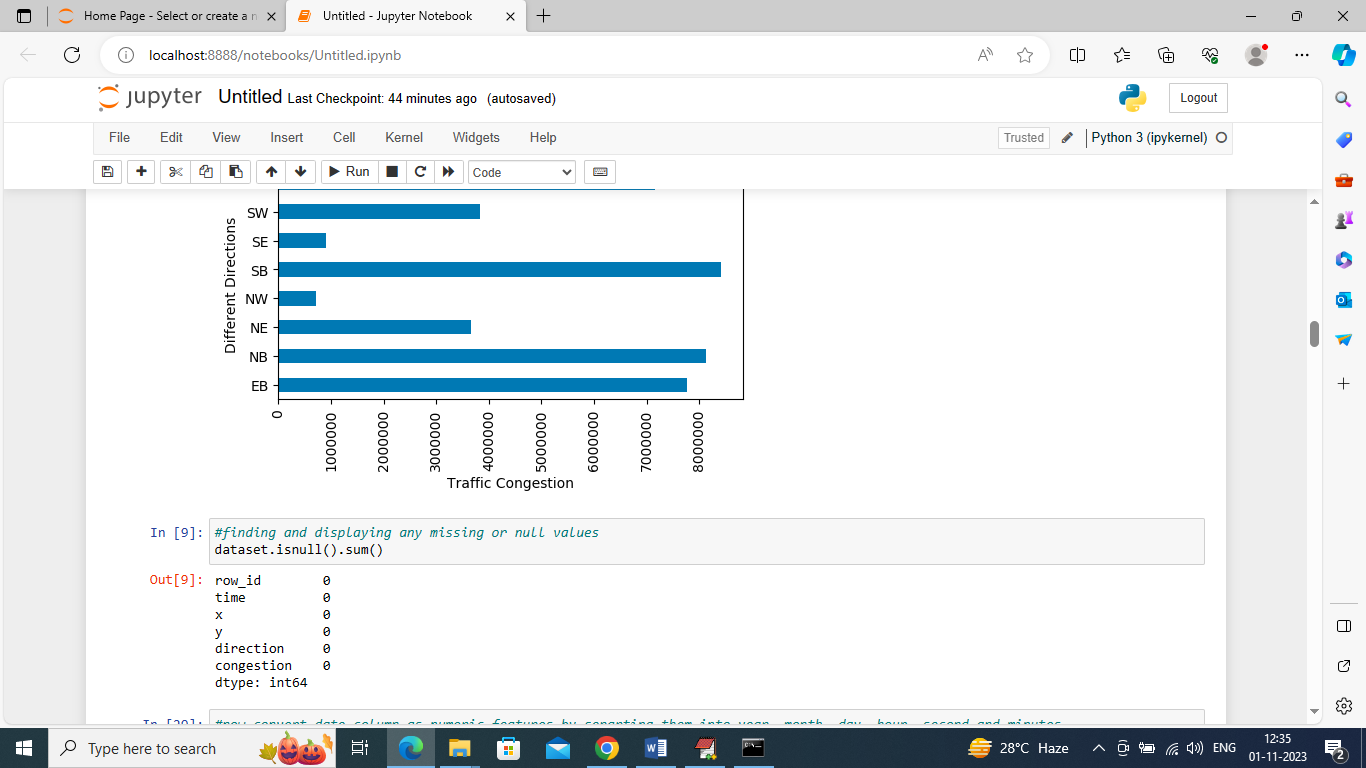
In above graph displaying traffic congestion on different dates where x-axis represents Date and y-axis represents traffic congestion



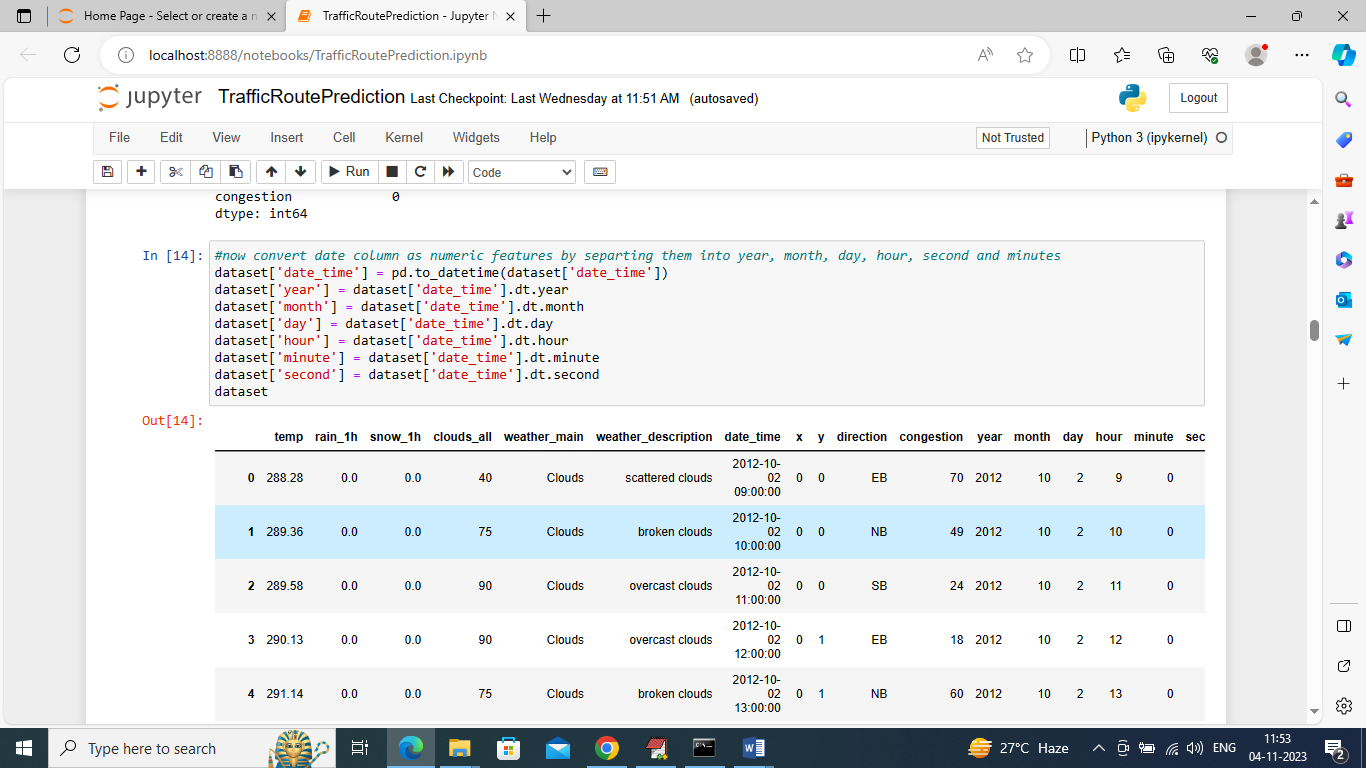
In above graph we are finding percentage of different directions or route exists in the dataset



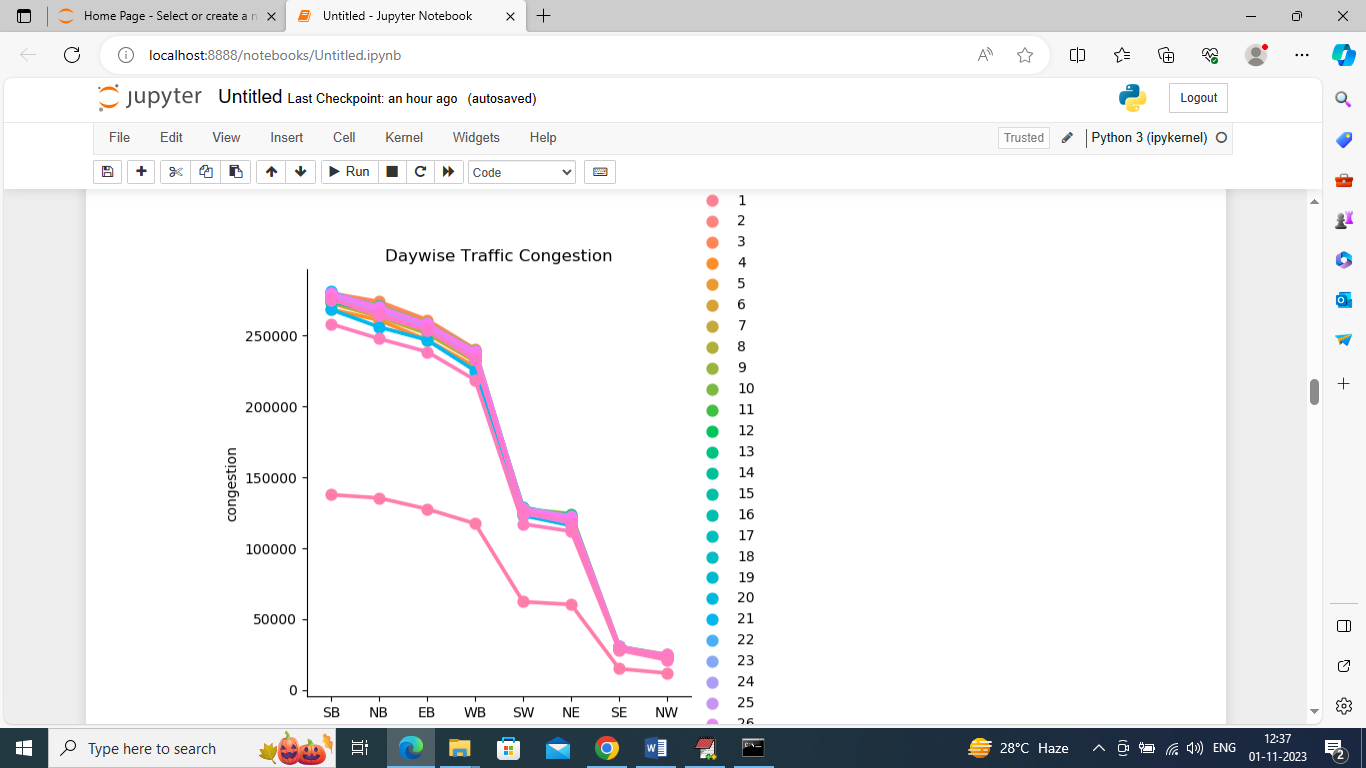
In above graph we are finding sum of traffic exists in each direction where x-axis represents traffic count and y-axis represents direction



In above screen we are finding weather dataset contains any missing or null values but this dataset has no missing values



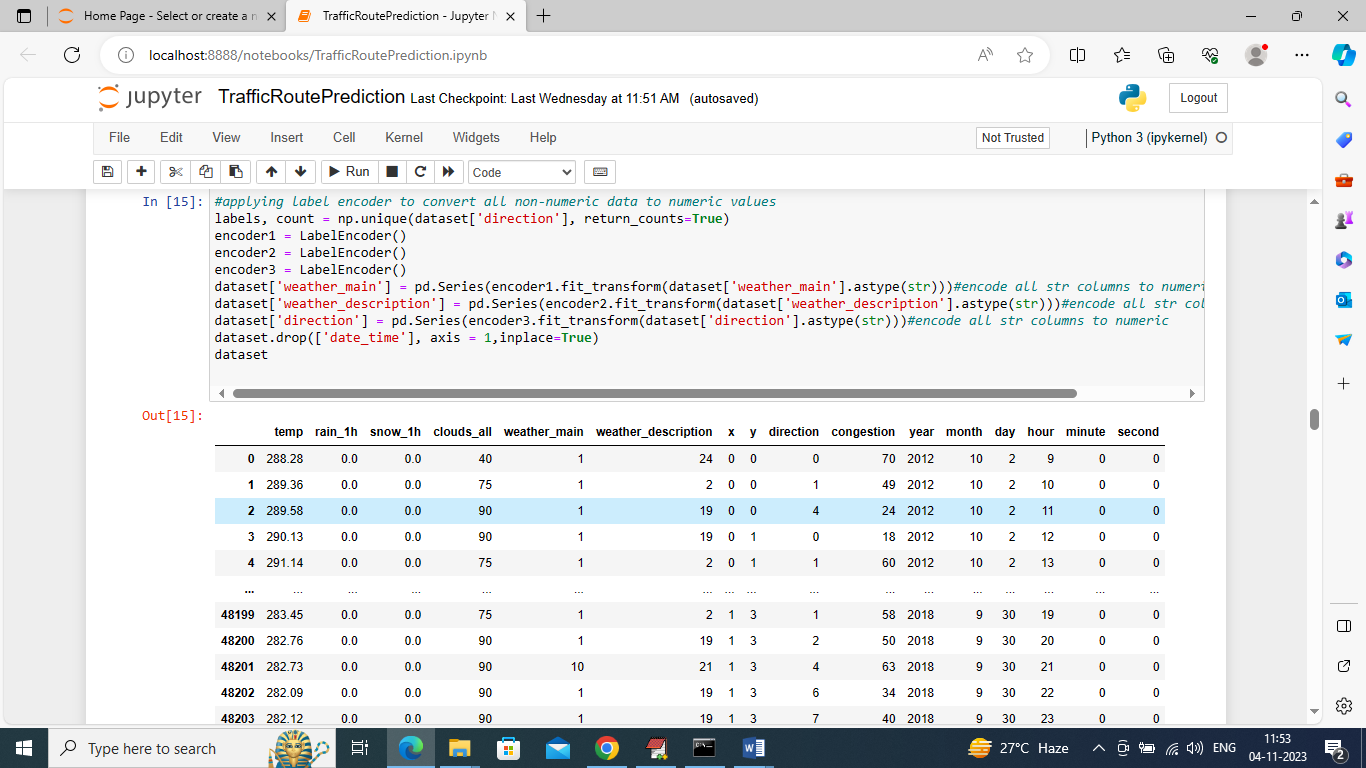
In above screen we are processing dataset to convert date into Day, Month and Year format so we can analyze traffic day or month wise and in above output we can see now dataset has day, year and month columns



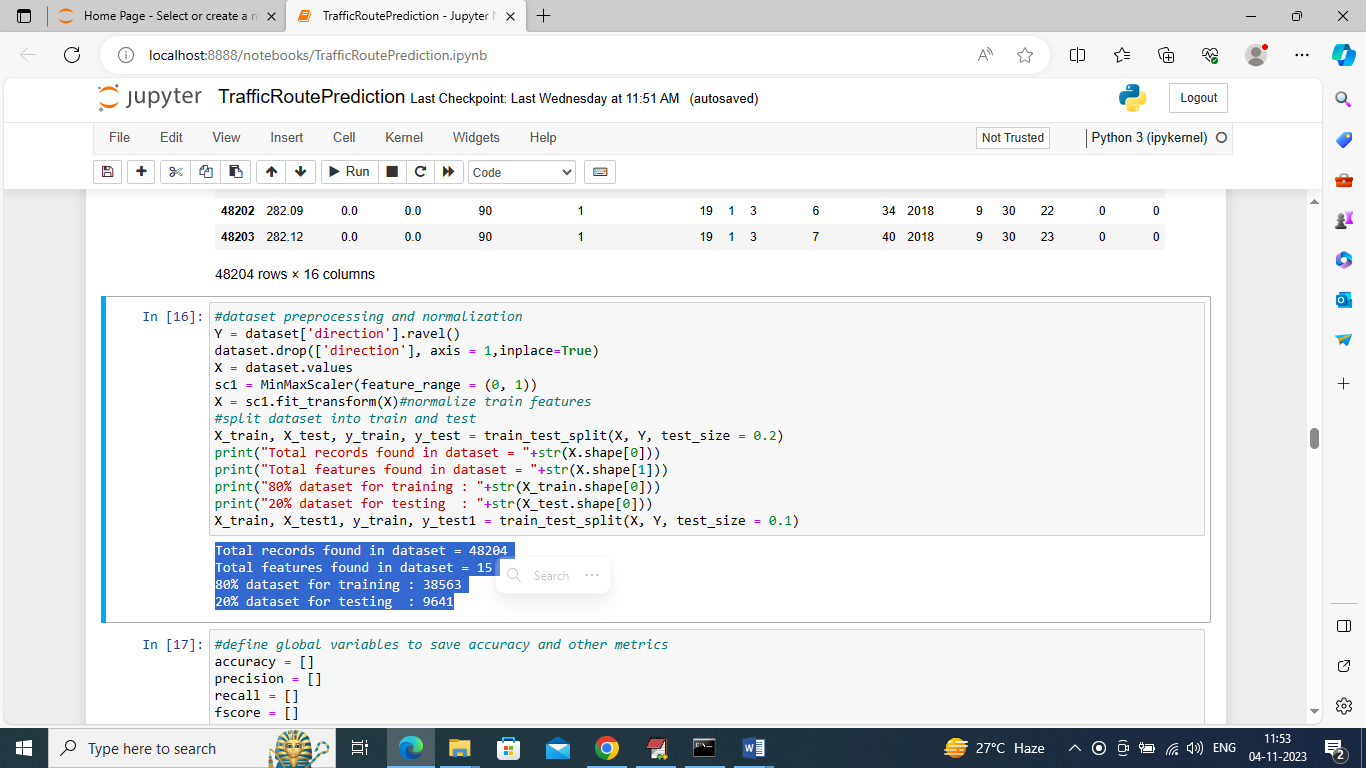
In above graph we are finding traffic day wise and each different color line represents different days traffic where x-axis is the direction and y-axis is the traffic congestion count



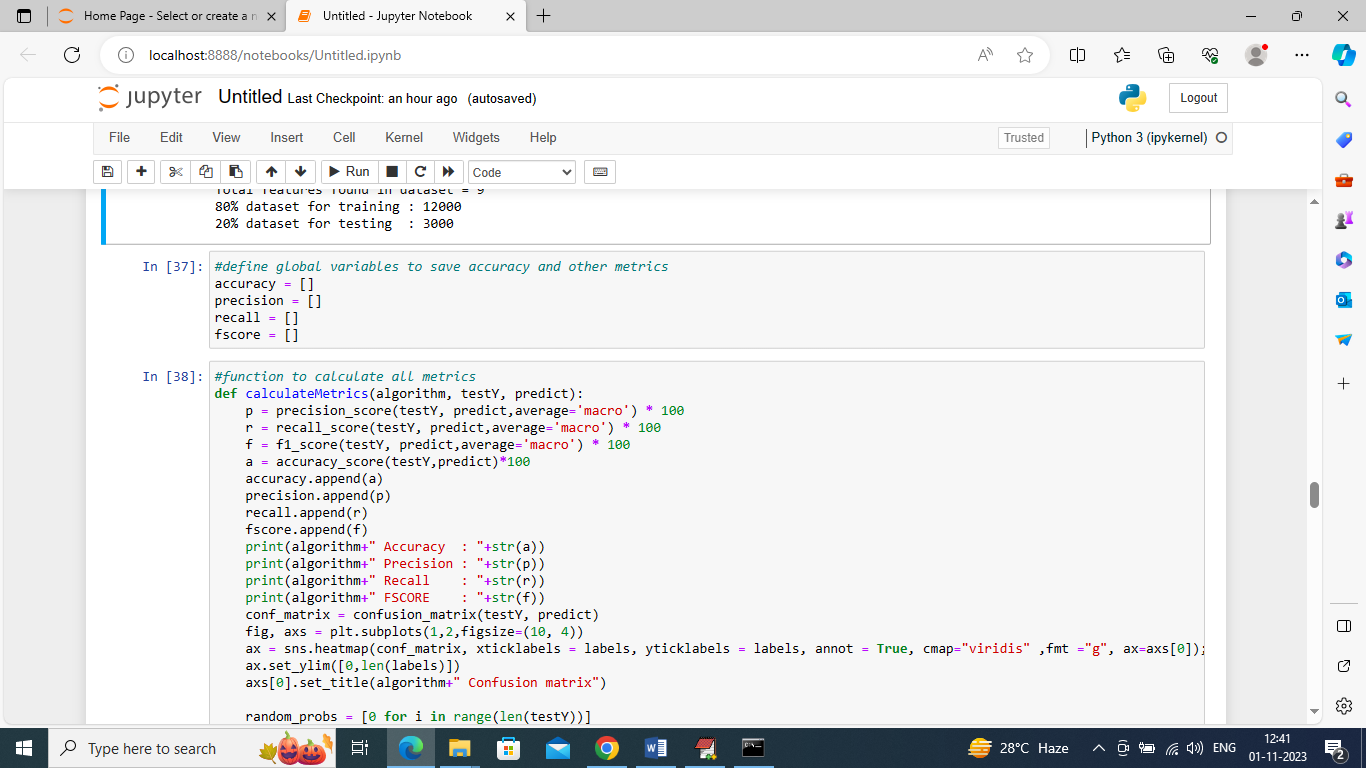
In above graph we are finding month wise traffic in different directions where x-axis represents direction and y-axis represents traffic congestion and different color dots represents months.



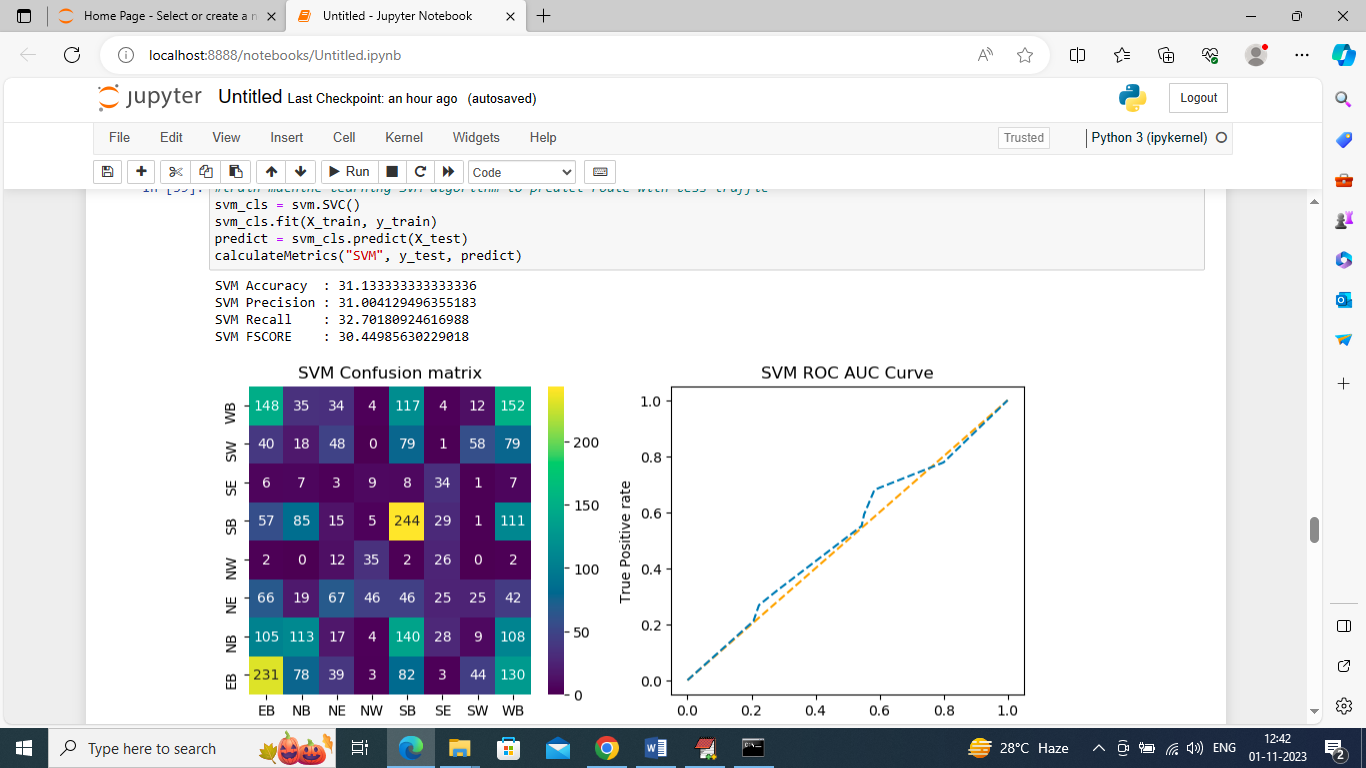
In above screen different directions are in string format so we have converted them into numeric format as ML will take all labels are numeric format so we have converted them into numeric labels



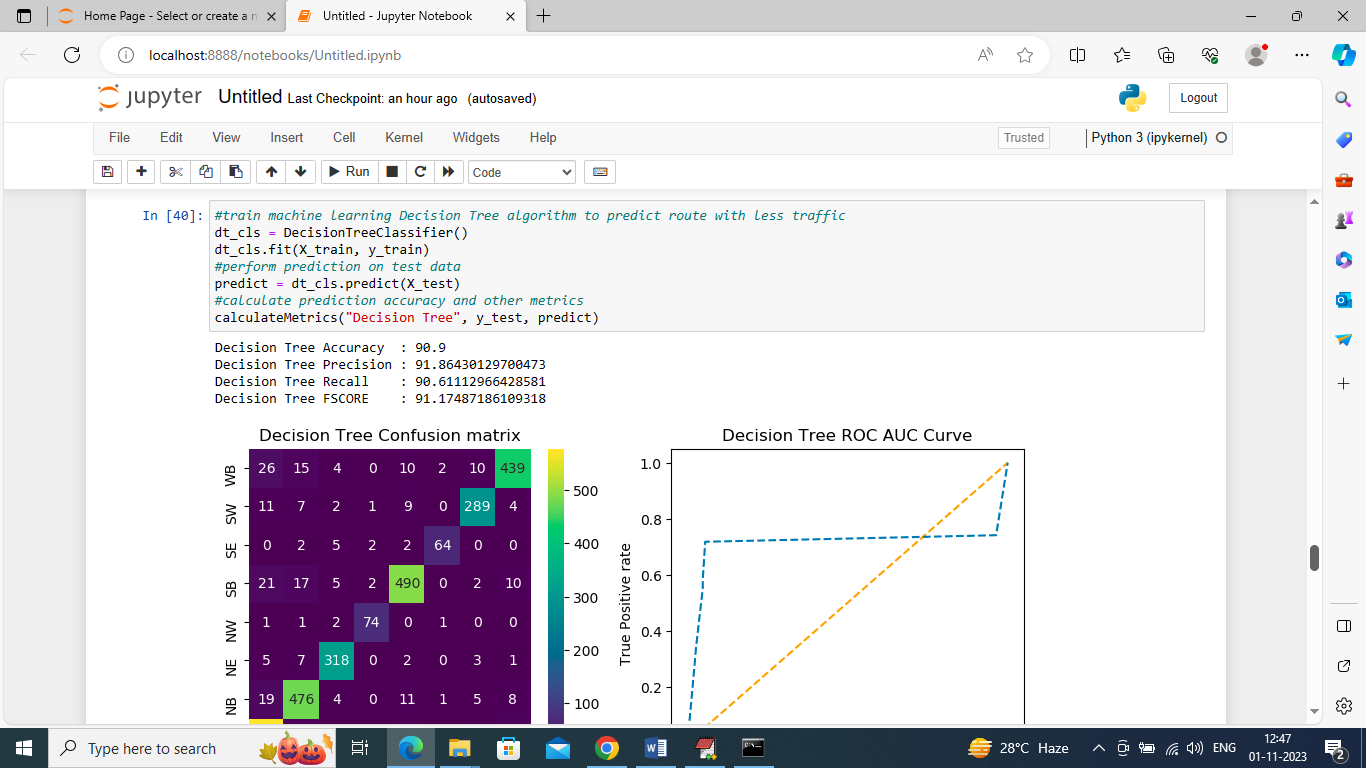
In above screen we are processing dataset such as normalization and then splitting into train and test where application using 80% dataset for training and 20% for testing and in blue color we can see train and test size



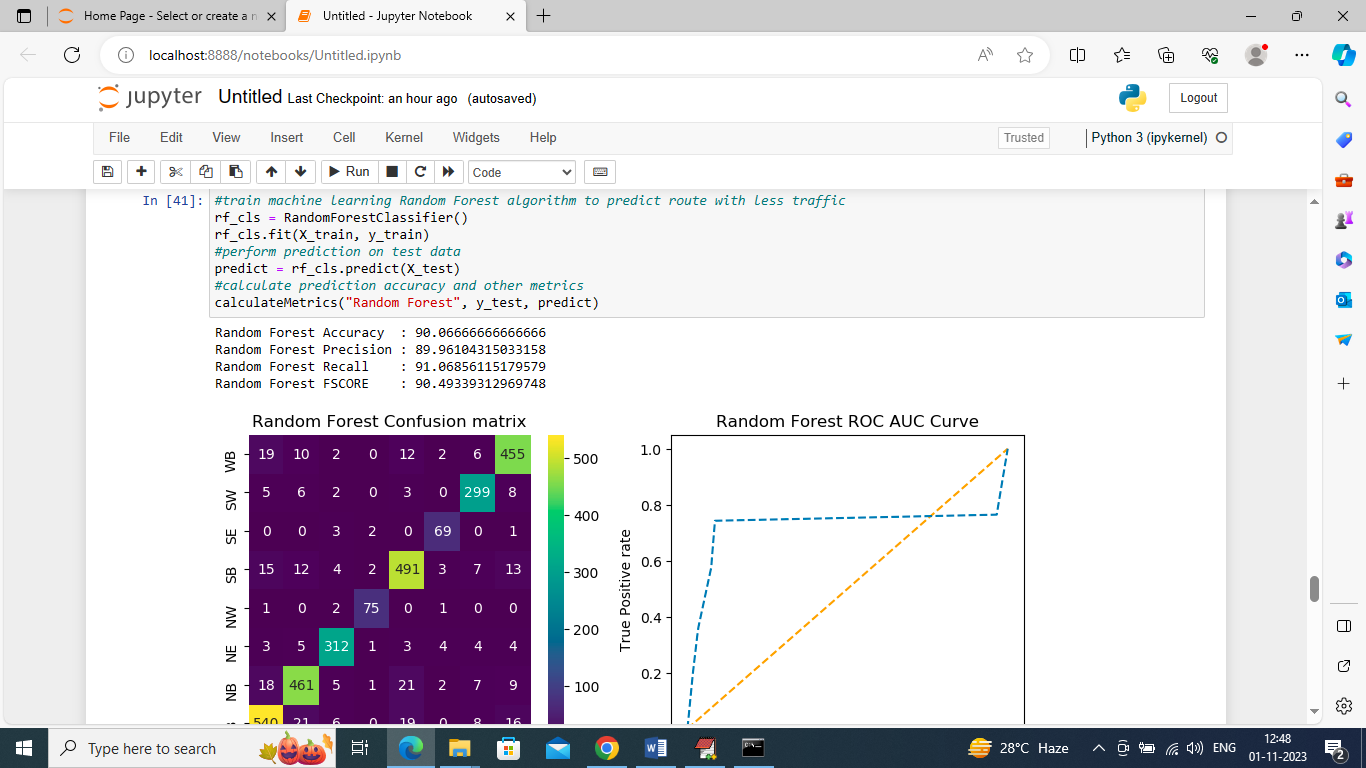
In above screen defining function to calculate accuracy and other metrics



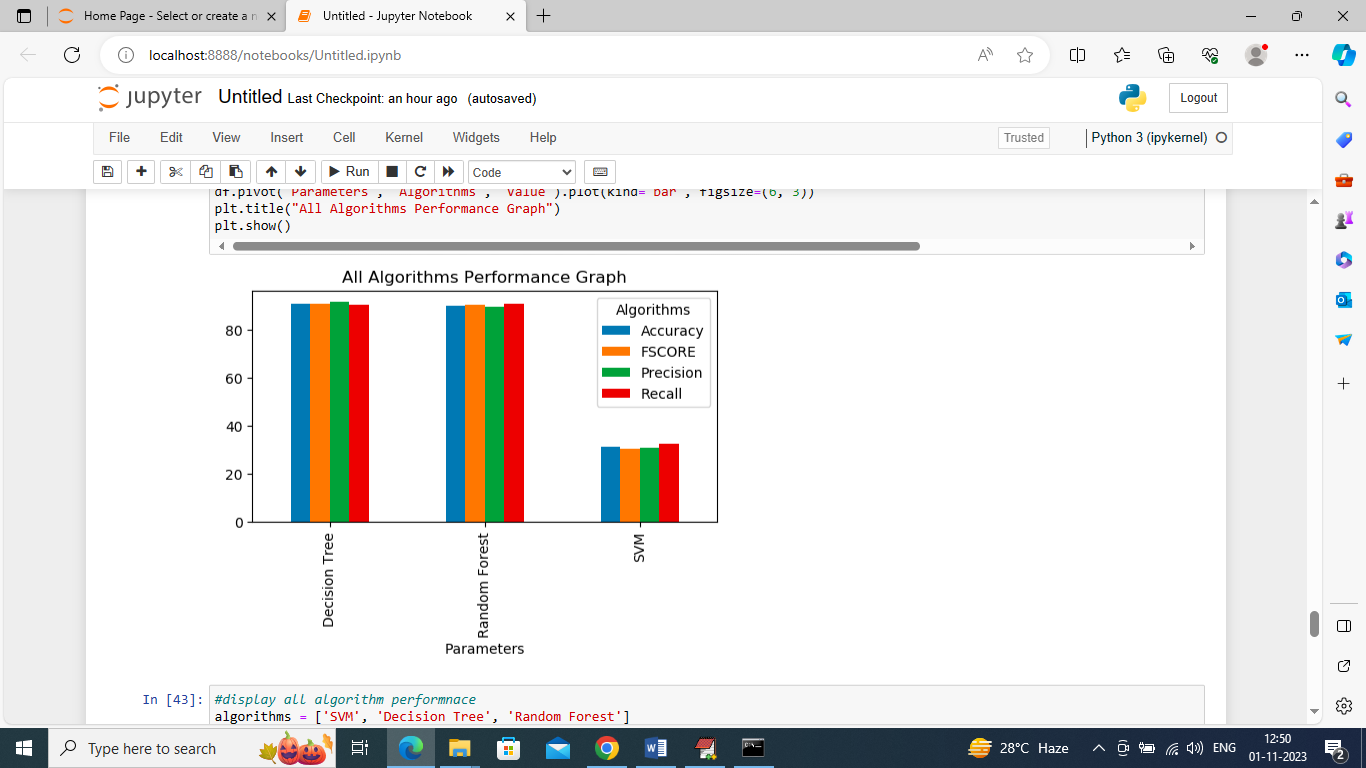
In above screen training SVM algorithm and after training SVM got 30% accuracy and can see other metrics also. In confusion matrix graph x-axis represents Predicted Labels and y-axis represents True Labels and all boxes in diagnol represents correct prediction count and remaining boxes represents incorrect prediction counts and from above graph we can notice SVM predicted many records incorrectly. In ROC curve graph x-axis represents False Prediction and y-axis represents True Predictions and if blue line comes on top of orange line then predictions are correct and if goes below orange line then predictions are incorrect and in above graph we can see only few predictions are correct



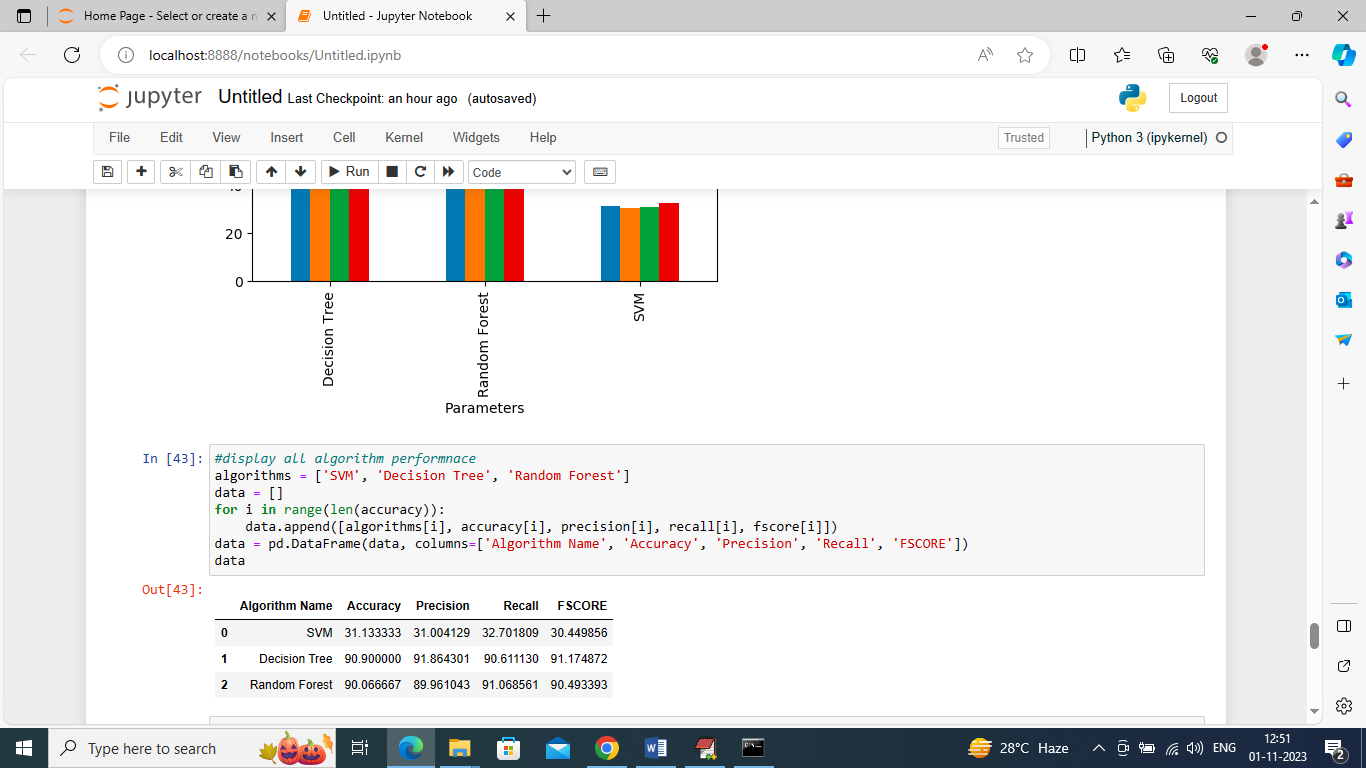
In above sceen training decision tree and it got 90.9% accuracy and can see other metrics and results graph



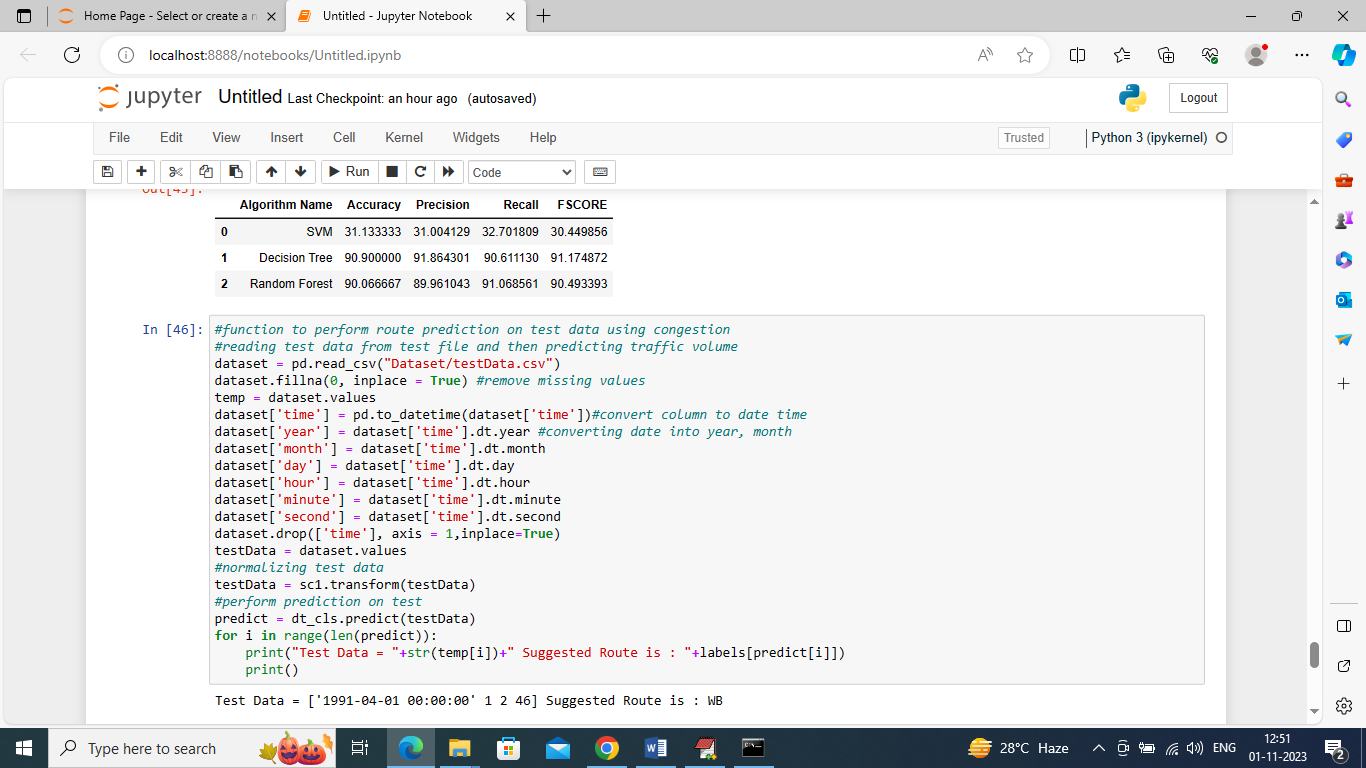
In above screen training Random Forest and it got 90.06% accuracy and can see other metrics also and in above confusion matrix graph in diagnol we can see many records are correctly predicted and in all blue boxes only few are incorrectly prediction. In ROC graph also we can see only few predictions are incorrect



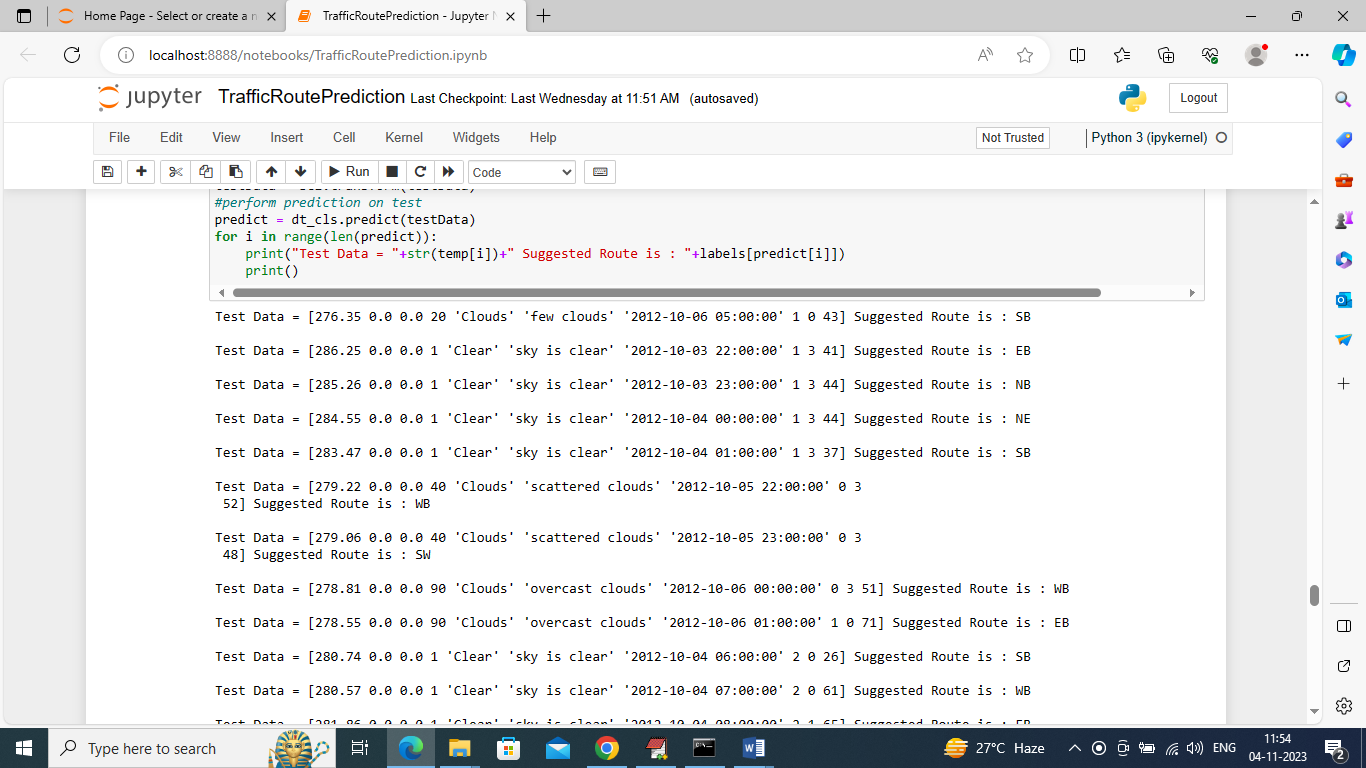
In above graph x-axis represents algorithm names and y-axis represents accuracy and other metrics in different color bars and in all algorithms Random Forest and Decision Tree work best



In above screen can see all algorithm performance in tabular format



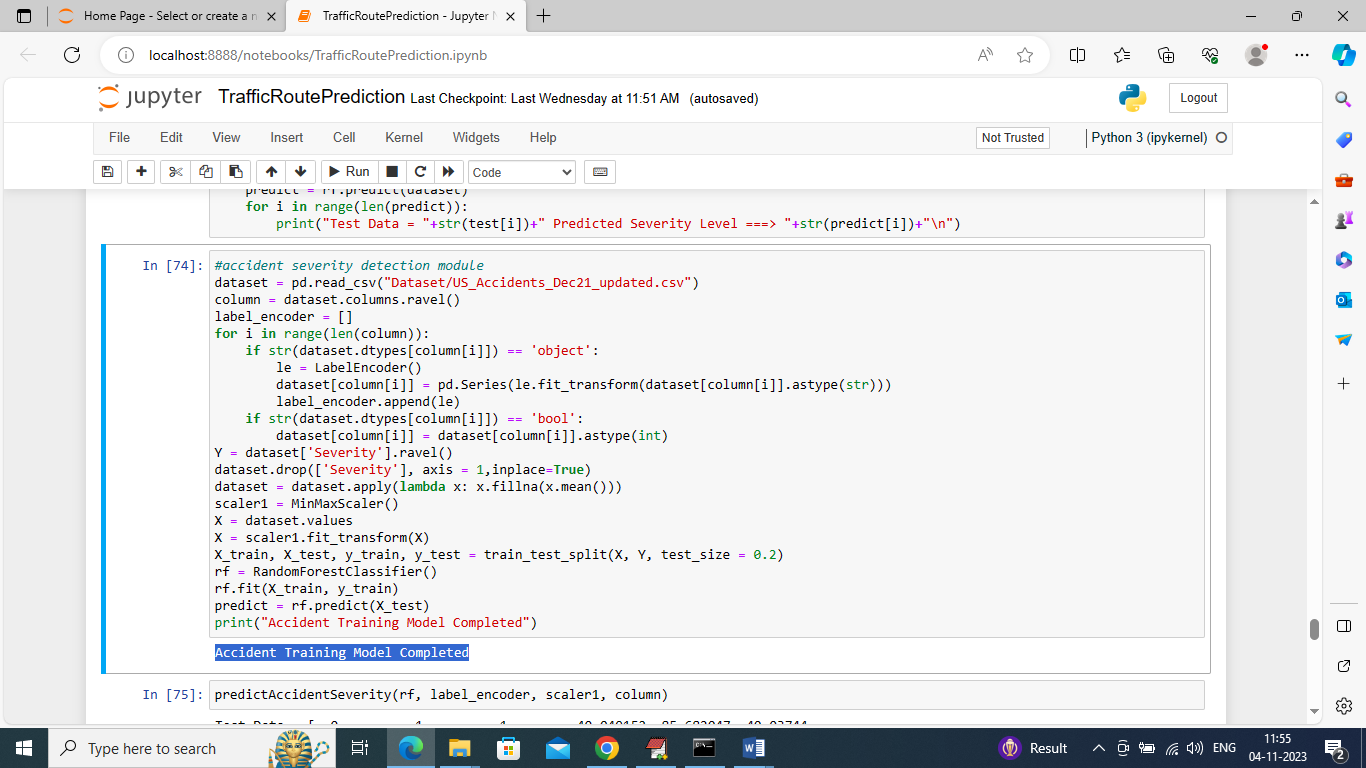
In above screen we are defining function to read TEST data and then perform route prediction on test and after execution above block will get below output



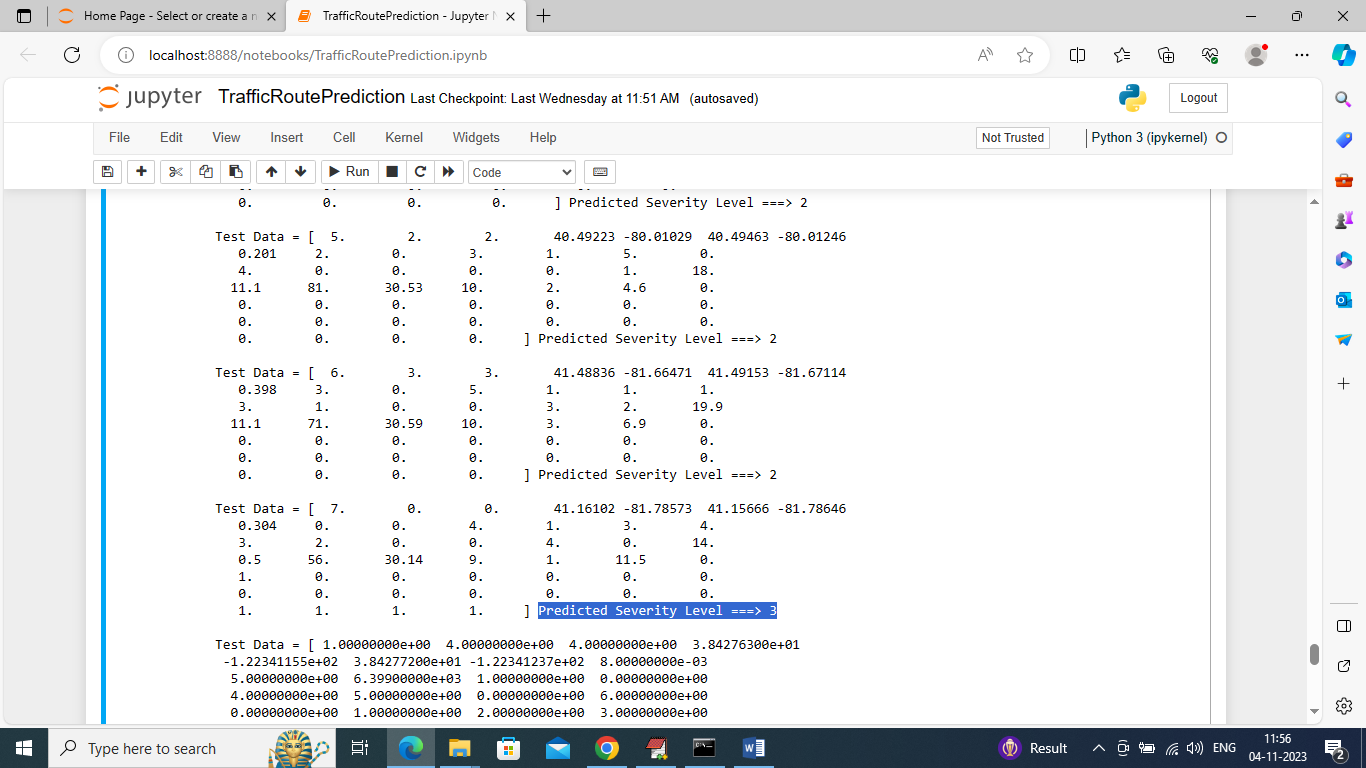
In above predictions in square bracket we can see the TEST data where last value is traffic congestion and based on that congestion we can see suggested Route as WB or NB or SE etc.

Note: the direction of travel of the roadway. EB indicates "eastbound" travel, for example, while SW indicates a "southwest" direction of travel.

In below screen we are reading Accident Dataset and then by using ML we are predicting accident severity and this severity is in range between 2, 3 and 4 and below are the accident severity prediction model training code



In above screen we are reading accident dataset and then training Random Forest model and in blue color text can see model training completed and when we apply this model on test data then will get below severity prediction output



In above screen in square bracket we can see accident Test Data and after =🡺 symbol we can see predicted severity as 2 or 3 or 4