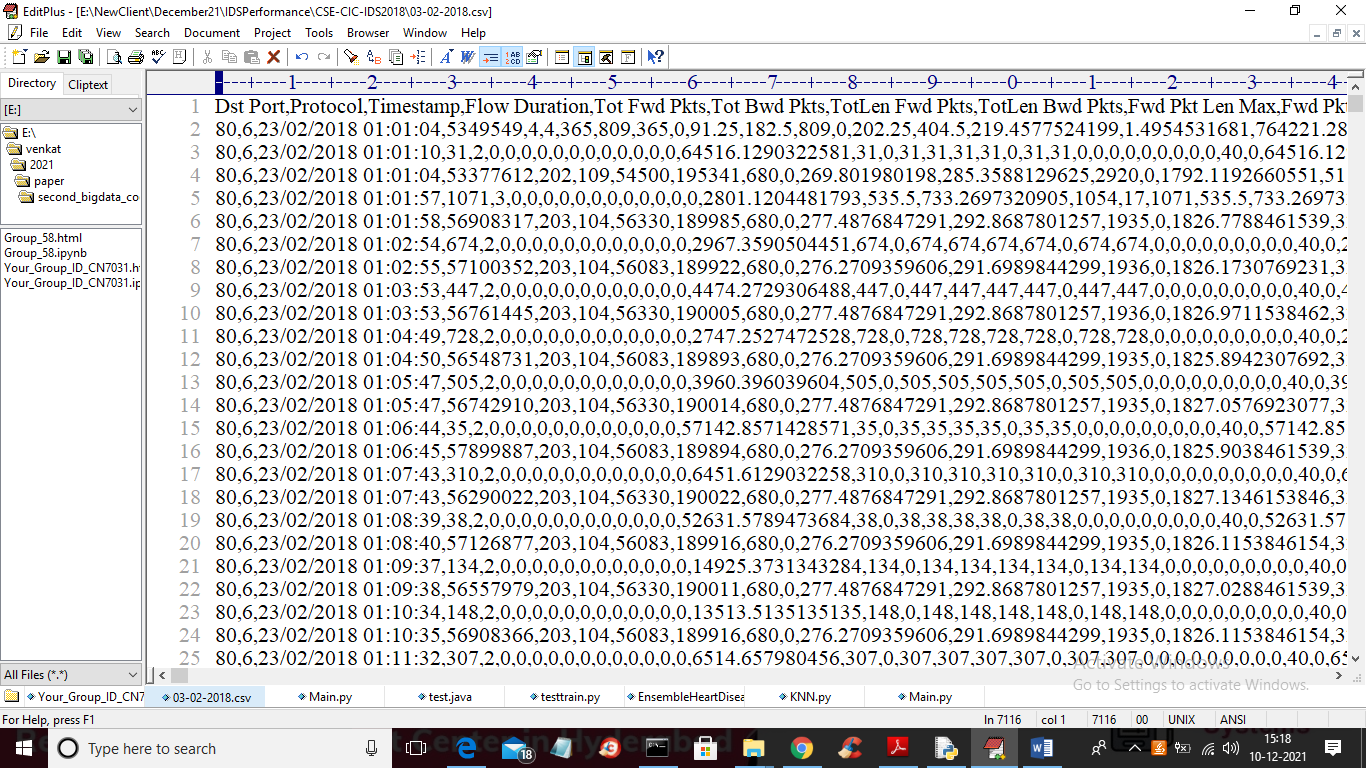
Increasing the Performance of Machine Learning-Based IDSs on an Imbalanced and Up-to-Date Dataset

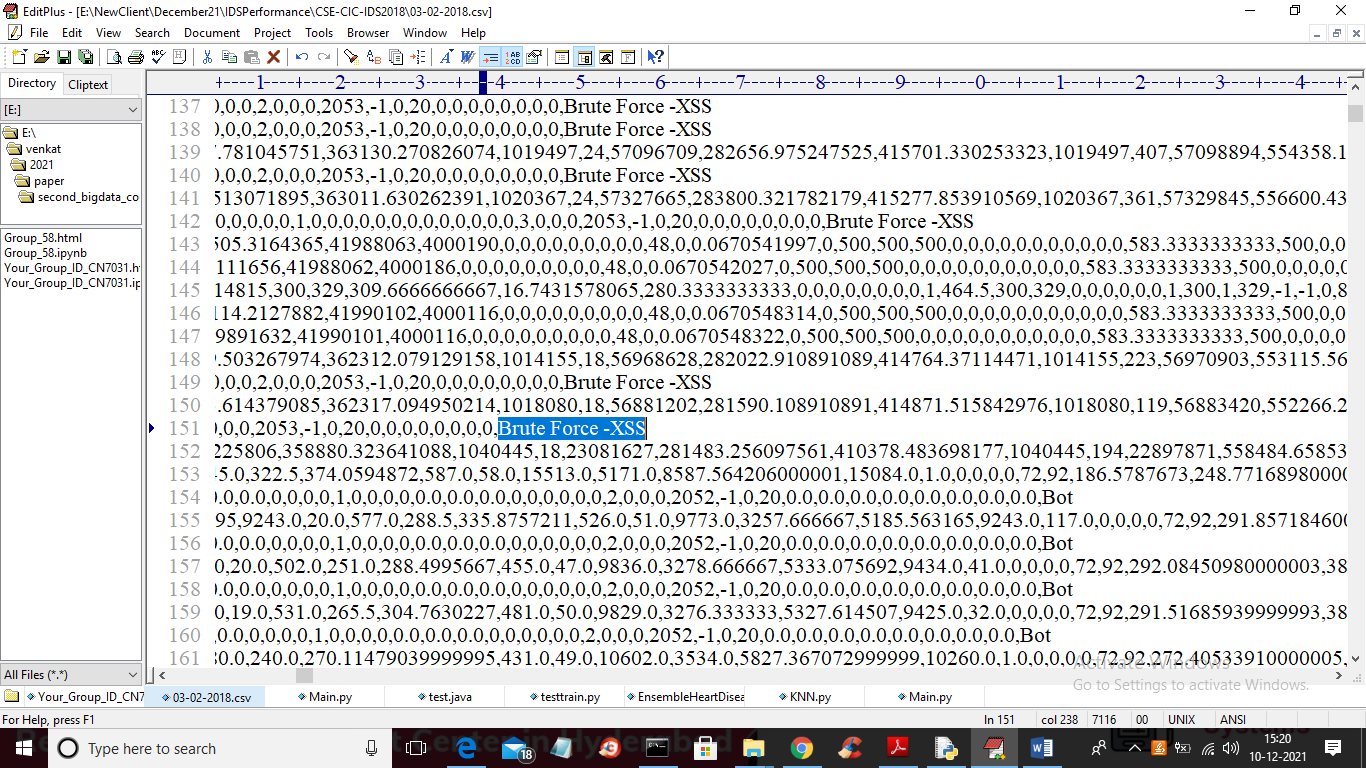
Internet is a collection of servers and clients machines running over a network where client always send request to server to get services such as transferring amount online, checking email messages or posting opinions on social media. This extensive usage of servers allow hackers to identify weak point and by using this they can intrude or hack server and can send enormous request to crash server or alter request or response data. To overcome from this problem many intrusion detection systems have been designed using machine learning algorithms which get trained on past data and then analyse new request to predict it as normal or attack. Hackers always finds new technique to attack servers so all existing IDS may not be equipped with latest data and all are trained on old datasets such KDD or NSL. Machine learning algorithm trained on this data may not be able to predict latest attack.

To overcome from this problem author of this paper has introduce Up-to-Date training dataset to train 6 different machine learning algorithms such as KNN, decision tree, random forest, Adaboost, Gradient Boosting and LDA algorithm. Latest attack can be detected with Up-to-Date dataset such CSE-CIC 2018 but this dataset contains 6 different types of attack with huge instances or records in one attack and fewer records in other attack which raise data imbalance problem and this imbalance will influence machine learning to predict those attacks which has high number of records and ignore predicting low instances attack. To overcome from this author has applied SMOTE algorithm which will equalize all instances in all classes or attacks. SMOTE will increase records size of fewer instances to avoid imbalance problem.

To implement this project author has used CSE-CIC 2018 dataset and we also used same data and this dataset contains network request signature labels with BENIGN (normal request), BOT (attack request). Below screen showing some record from dataset.

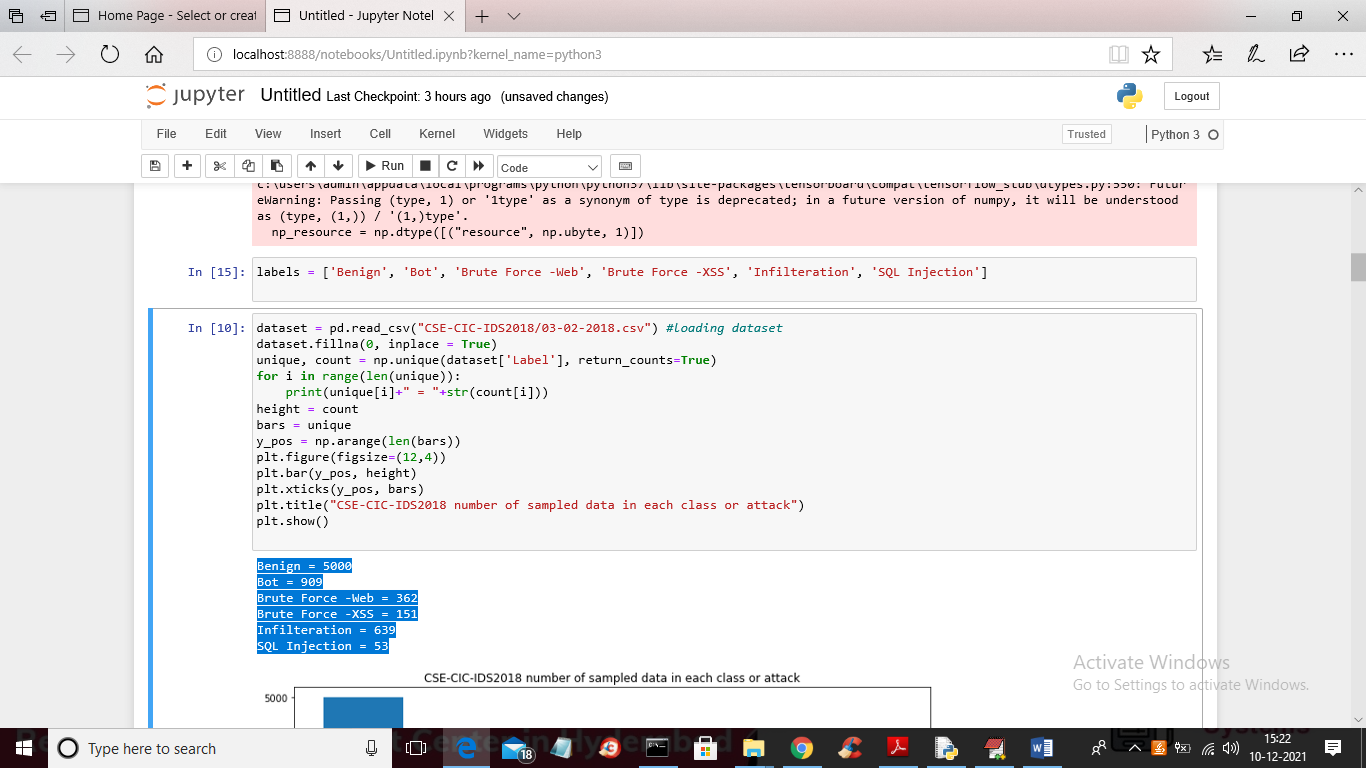


In above screen first row contains dataset column names and remaining rows contains dataset values and in below screen we can see each record is associated with attack class labels.

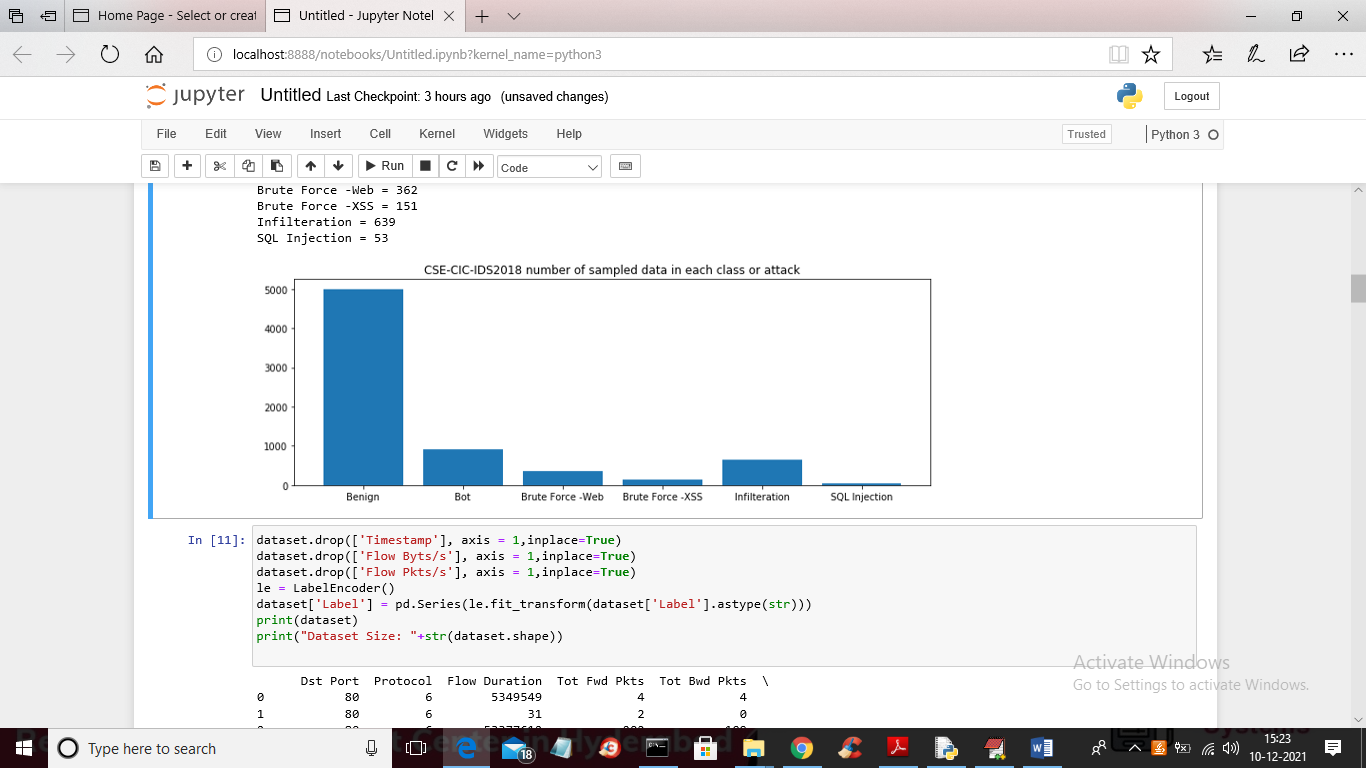


In above screen in last column you can see names of attacks.

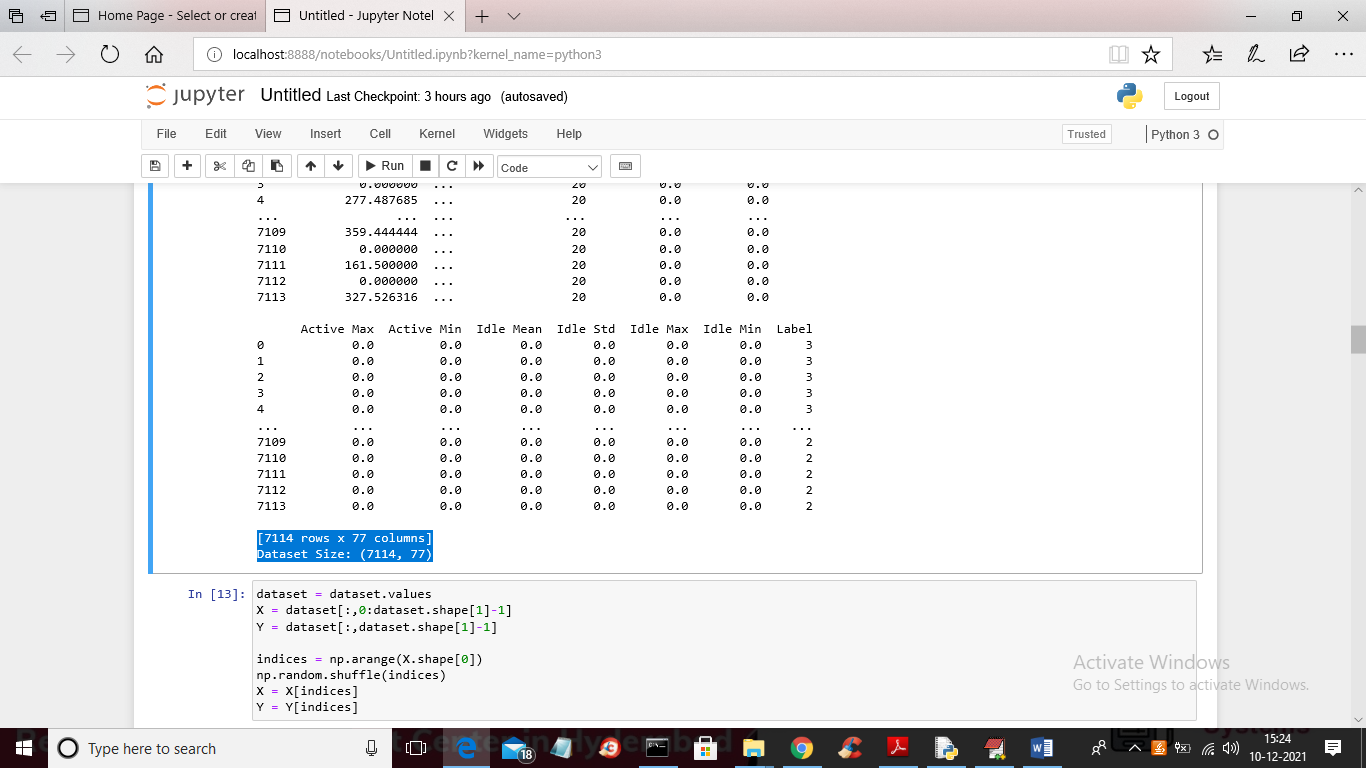
We will used above dataset to train all machine learning algorithms with and without SMOTE and then compare performance between them.



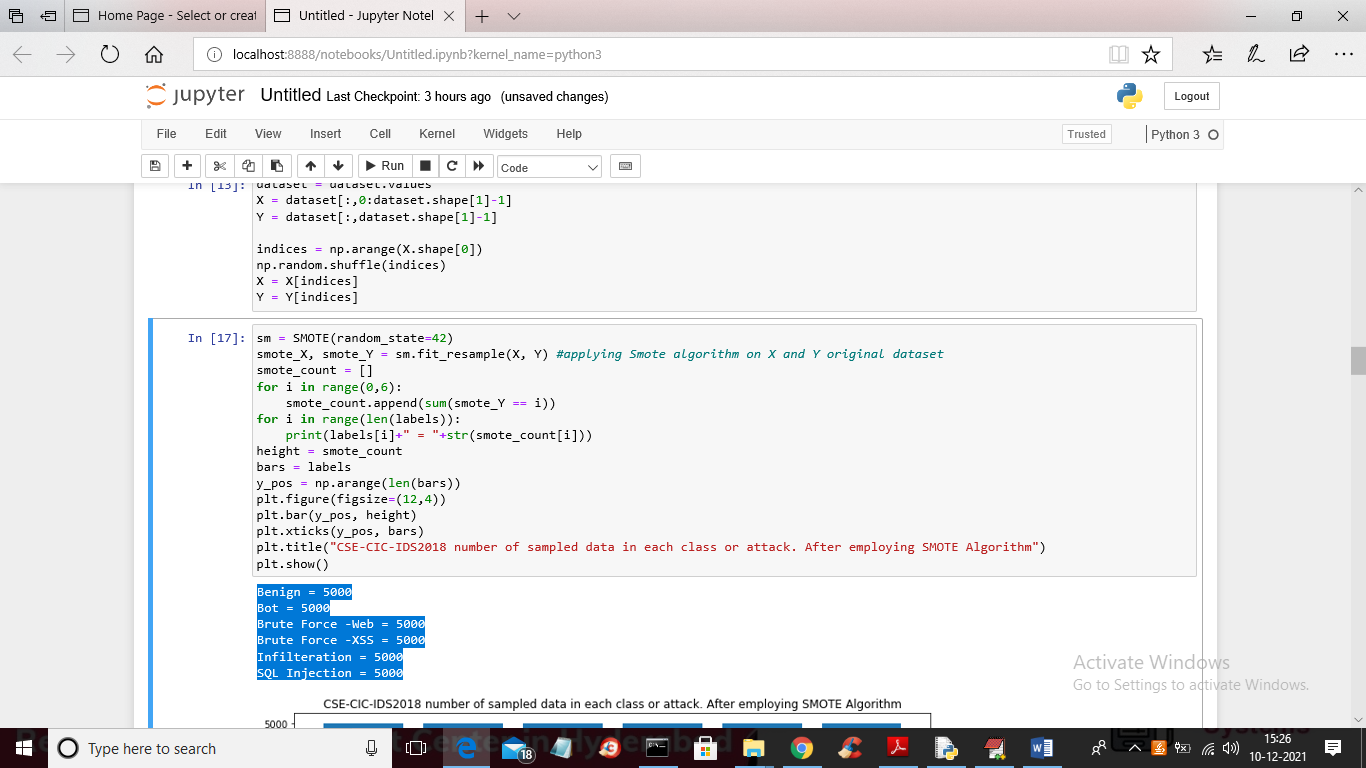
In above screen we are reading dataset and then in blue colour output you can see Benign contains 5000 records in dataset and SQL Injection contains only 53 records and this indicate dataset is highly IMBALANCE. We can see this imbalance in below graph



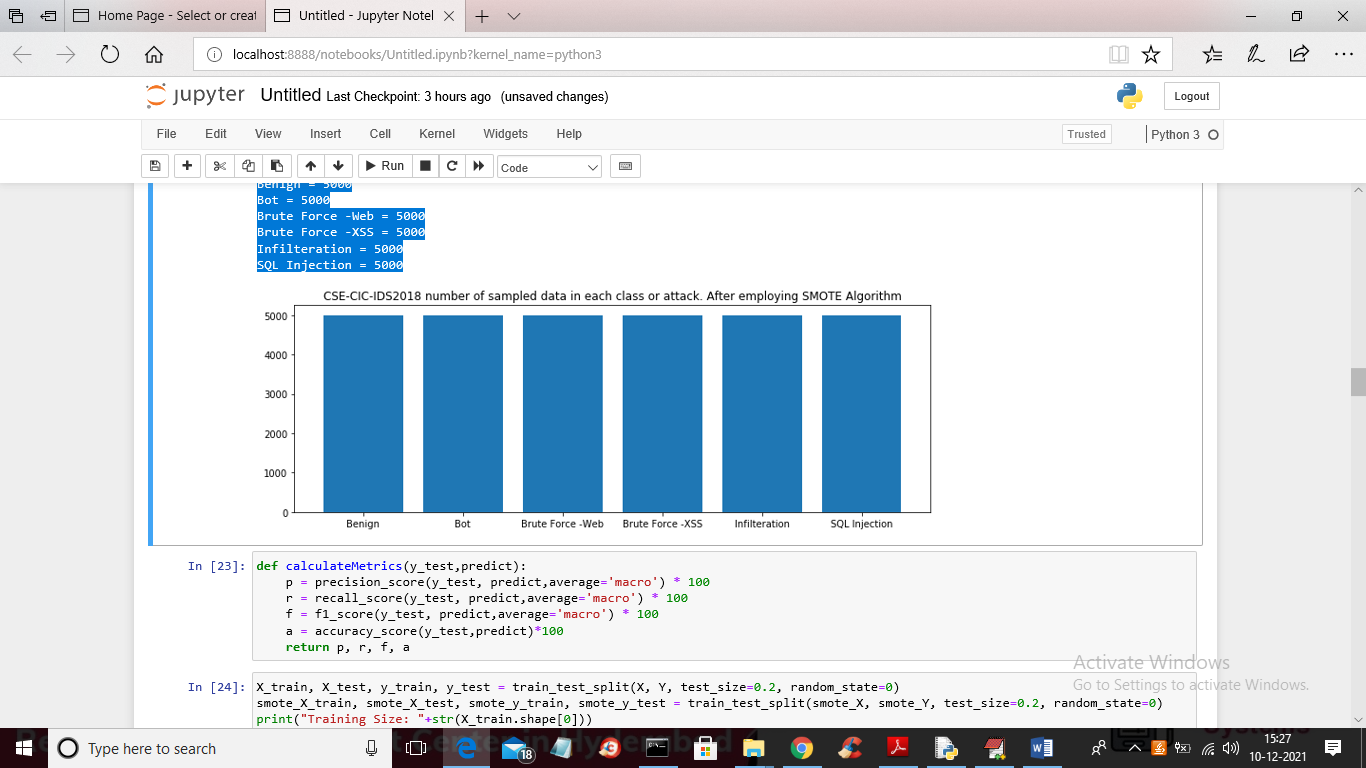
In above graph x-axis represent attack names and y-axis represents number of instances for that attack and in all attacks SQL injection contains very few instances and in below screen I am displaying size of dataset



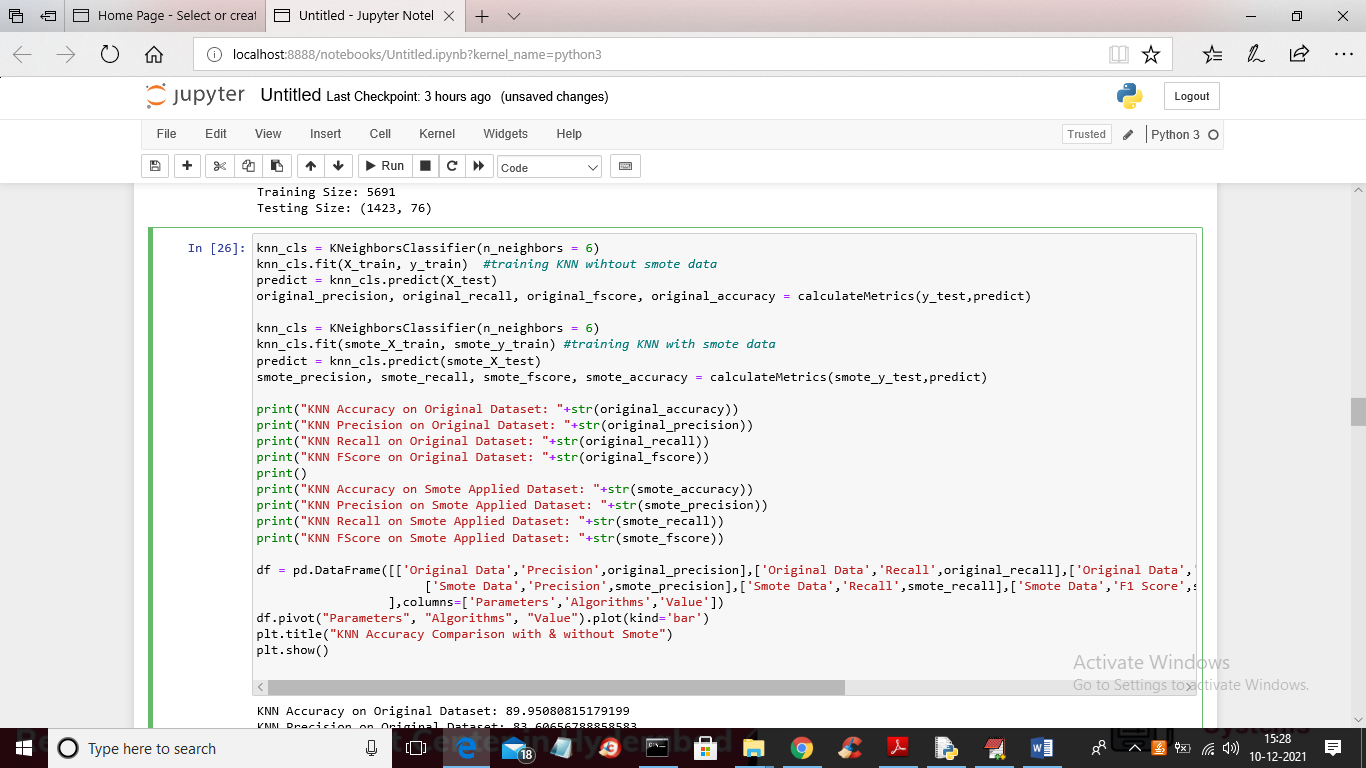
In above screen in blue colour text we can see dataset contains 7114 records and each record contains 77 column values and in screen I am showing code to apply SMOTE algorithm



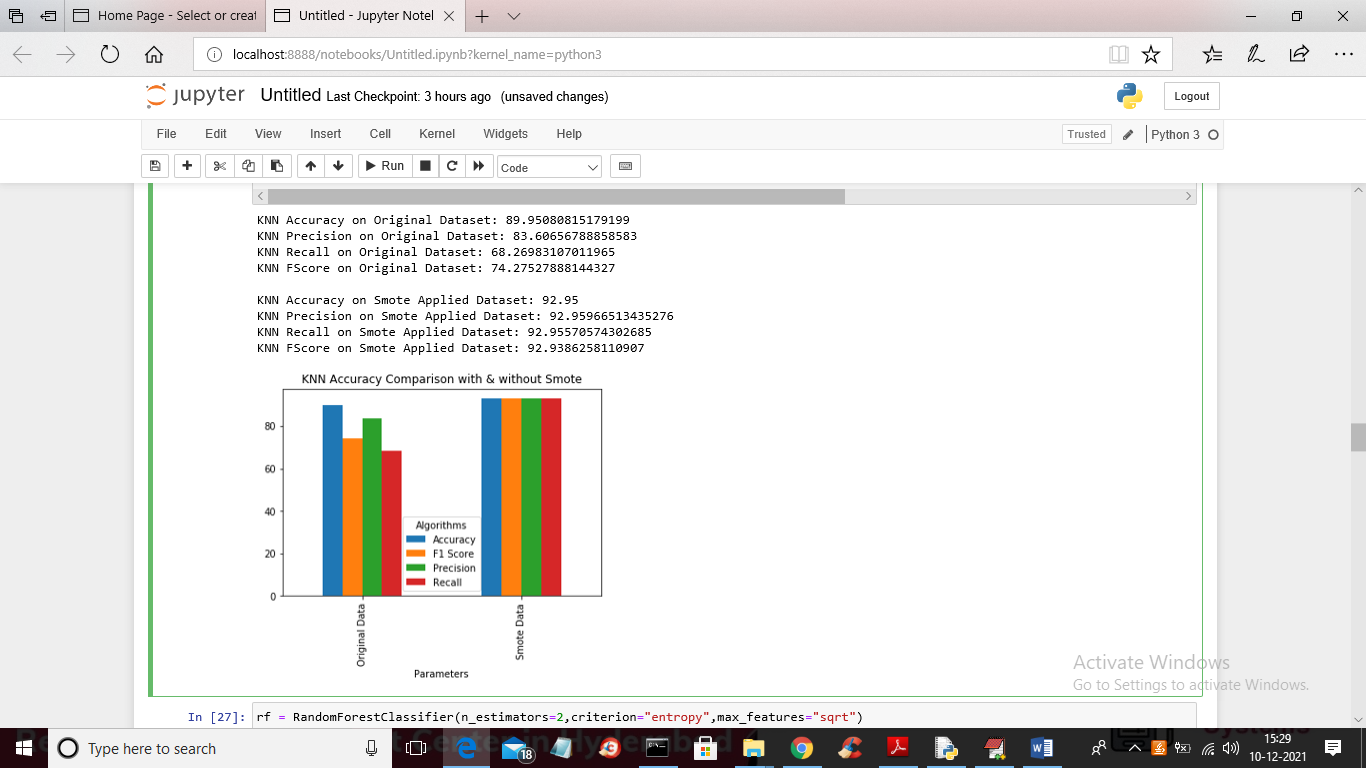
In above screen you can see on original data I applied SMOTE and it increase size of fewer records to equal all instances and after applying SMOTE all attacks contains 5000 equal records and below is the graph after applying SMOTE



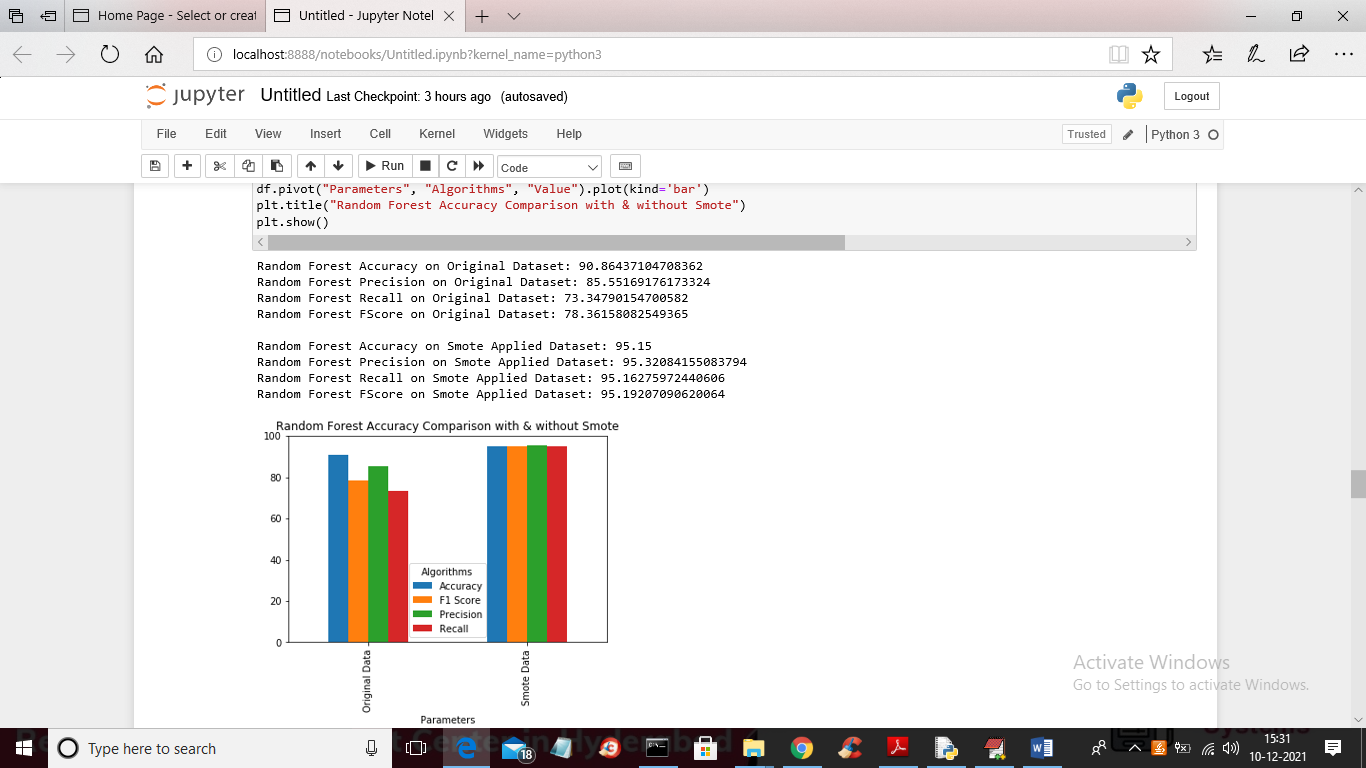
In above graph you can see all attacks has equal instances or records now and in below screen I am training KNN with and without smote



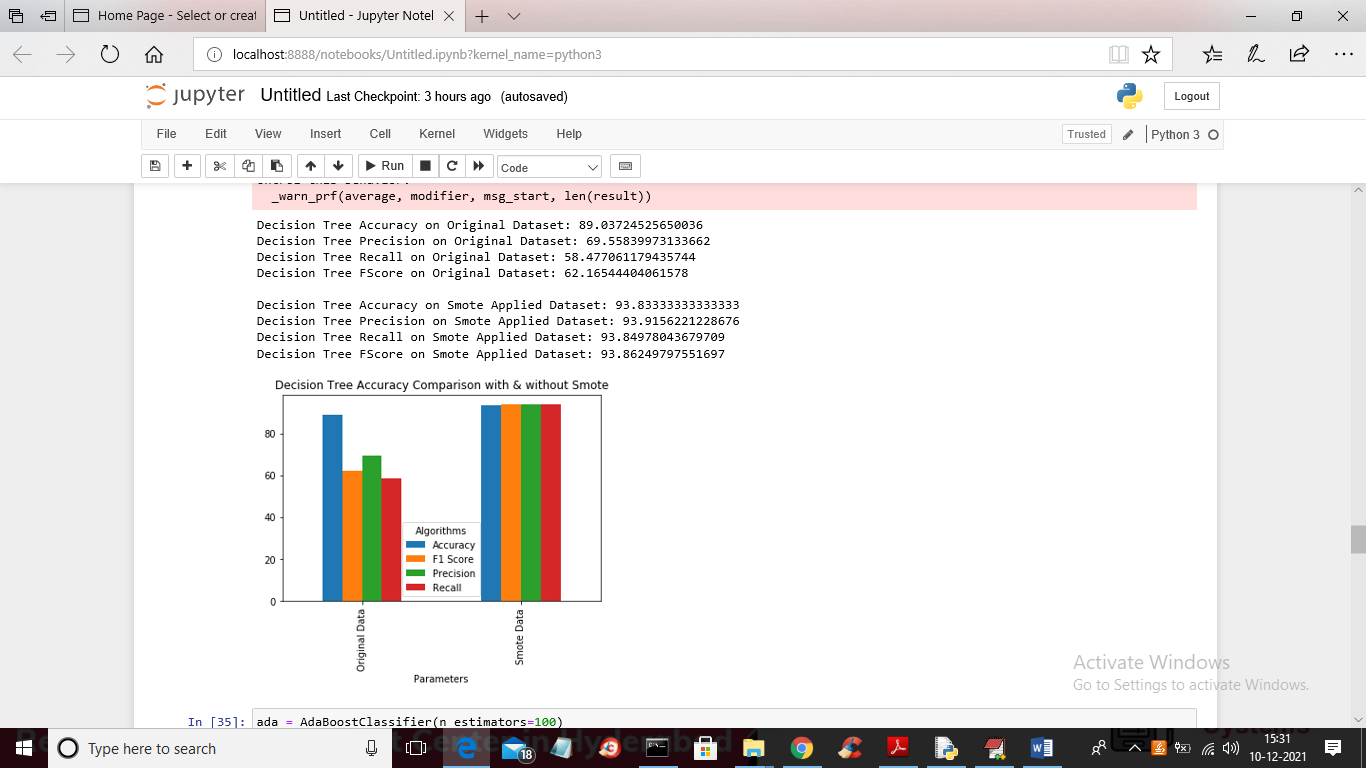
In above screen you can read light blue colour comments starting with # symbol to know that we are training KNN with original and with smote data and below is the accuracy comparison between two algorithms



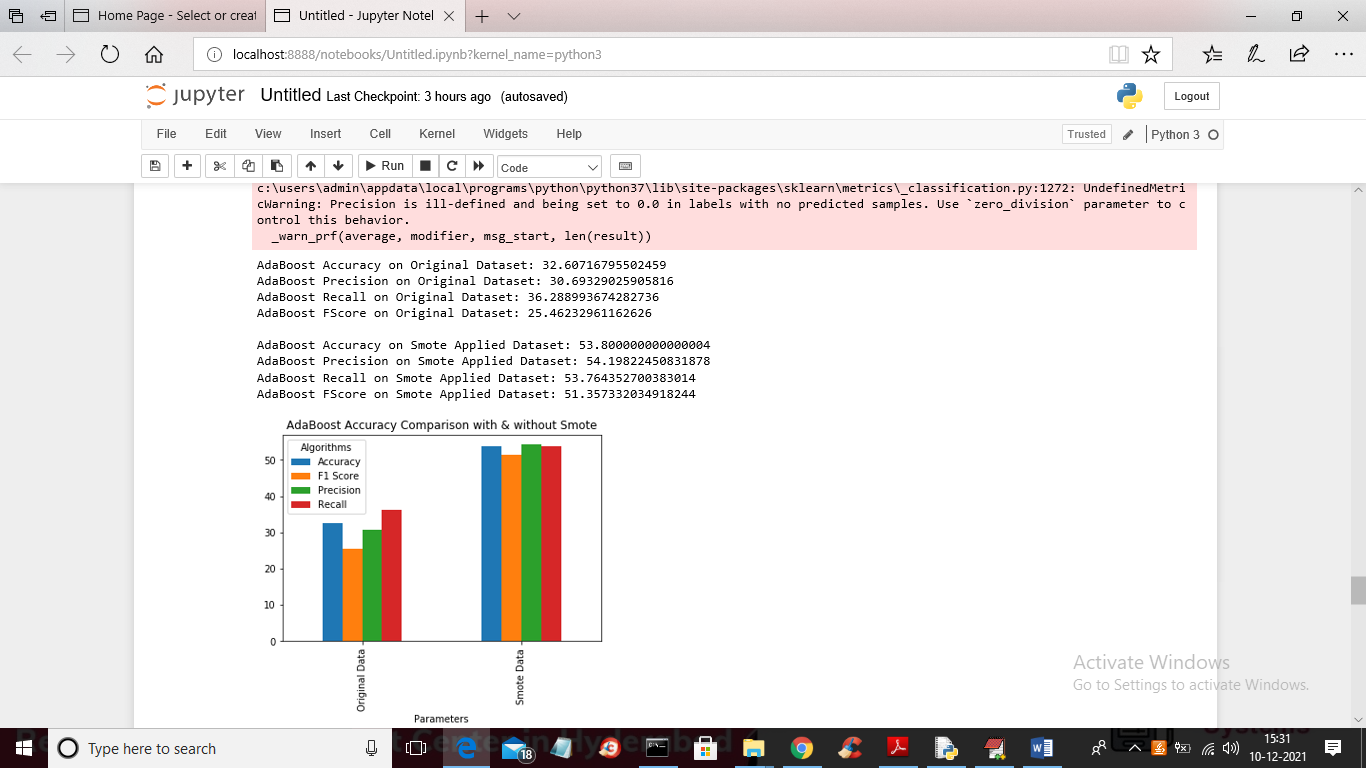
In above screen for KNN we calculated accuracy, precision, recall and FSCORE and then plotting graph for both with and without smote algorithm. In above graph each bar represents one metric for that algorithm. In above graph we can see with smote KNN performance is better. Below is the random forest output



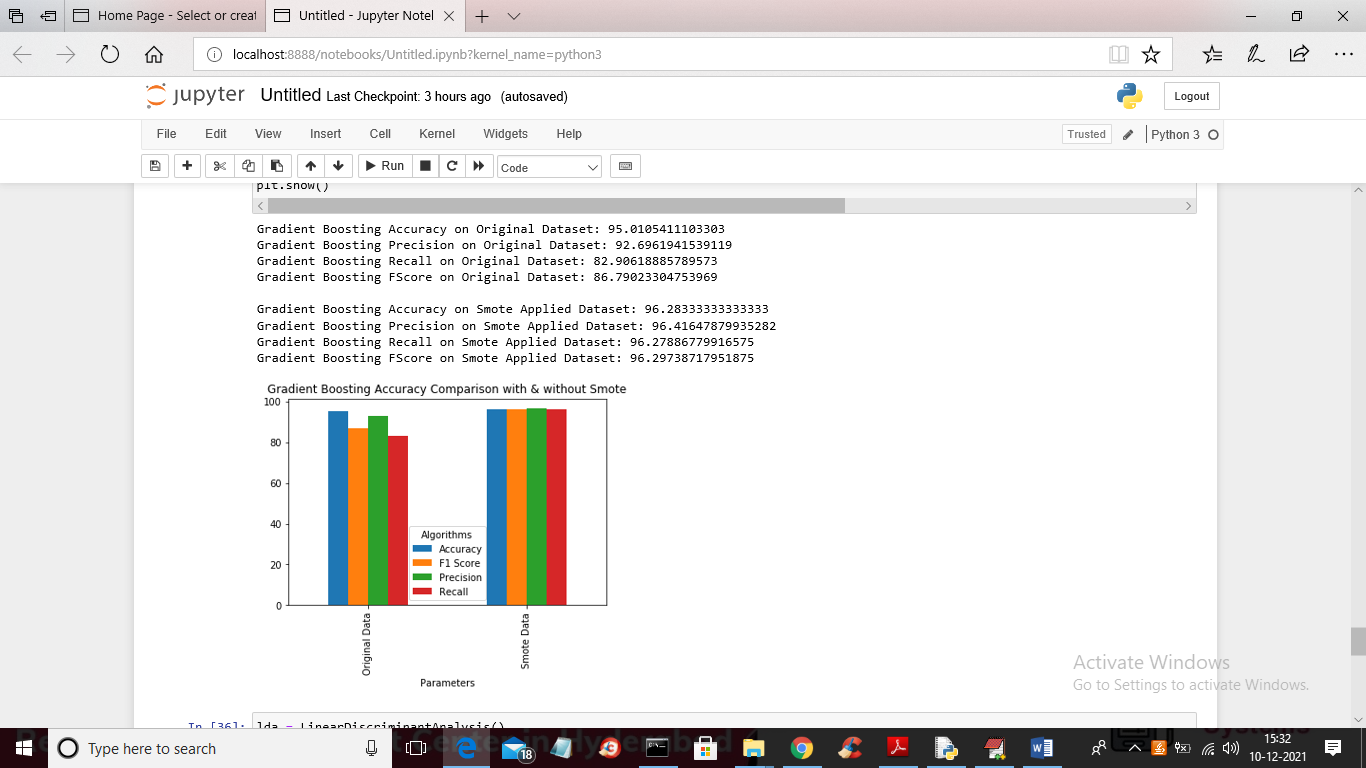
Below is the decision tree output



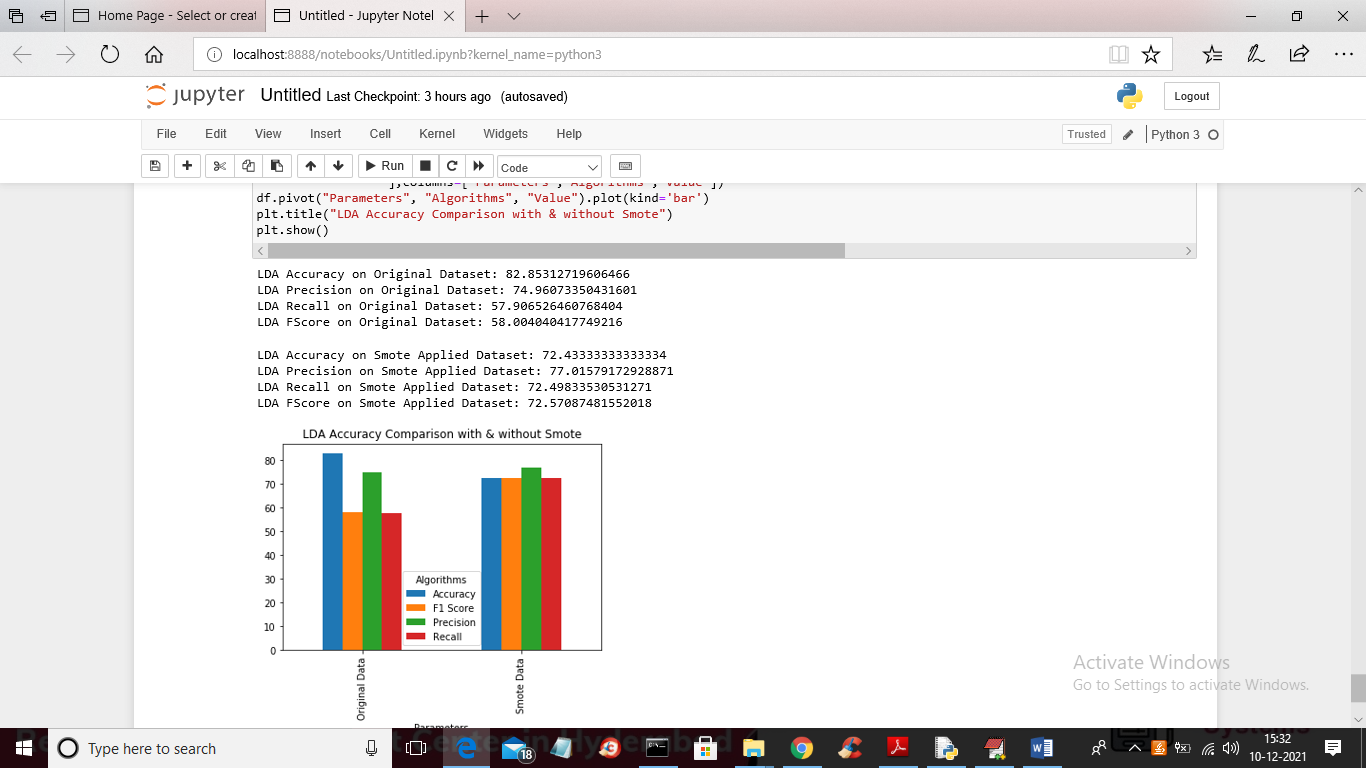
Below is the Adaboost output



Below is the gradient boosting output



Below is the LDA output



From above results we can conclude that in all algorithms SMOTE is giving better result