Darknet Traffic Analysis: Investigating the Impact of Modified Tor Traffic on Onion Service Traffic Classification

Computer network consists of many protocols for data transmission and this protocols include HTTP, Voice Protocol, Email Protocol and many more. This protocols often vulnerable of attacks for data steal and to overcome from this issue TOR protocol was introduced which provide anonymity or security to user’s data. Tor which provide anonymity is called as ONION Services. Often Onion services was used by illegal websites to avoid detection and perform malicious activities. Therefor in propose paper author applying machine learning algorithms to classify Tor Services as Onion Services or normal Tor services.

In propose work author has evaluated performance of various machine learning algorithms such as SVM, KNN and Random Forest. Each algorithm performance is evaluated in terms of accuracy, precision, recall and FSCORE.

To train above algorithm author has used TOR (No Defence) dataset and OS dataset where TOR data contains TOR services and OS data contains Onion services. TOR and OS dataset further modified to form new dataset called WTFPAD and Traffic Silver. Traffic Silver dataset is not available so we have train all algorithms using TOR Original dataset and WTFPAD modified dataset.

All algorithms get trained on Original dataset and then answer QUESTION 1 that, it’s possible to classify Tor and Onion services, QUESTION 2 answer that after modifying or padding dataset we can still classify Tor or Onion services but Modified TOR dataset accuracy is getting affected little.

As Question 3 Answer author has selected relevant 50 features from dataset by applying Information Gain, Correlation Coefficient and Fisher Score. Author has used various number of features such as 8, 13, 6 and many more but top selected 6 features are giving better performance so we have experimented with top 6 features.

In propose work we have trained SVM, KNN and Random Forest on Original TOR dataset, Modified WTFPAD dataset and OS Onion Services dataset. First algorithms get trained on Original TOR data, second time algorithm get trained on WTFPAD dataset and then for third time we combine OS and WTFPAD dataset to classify network as Tor or Onion. Each algorithm is tuned using Hyper parameters.

Tor (No Defence (Original) and WTFPAD (modified dataset)) dataset is available in PICKLE format and can be downloaded from below website

<https://github.com/deep-fingerprinting/df>

OS dataset can be downloaded from below website

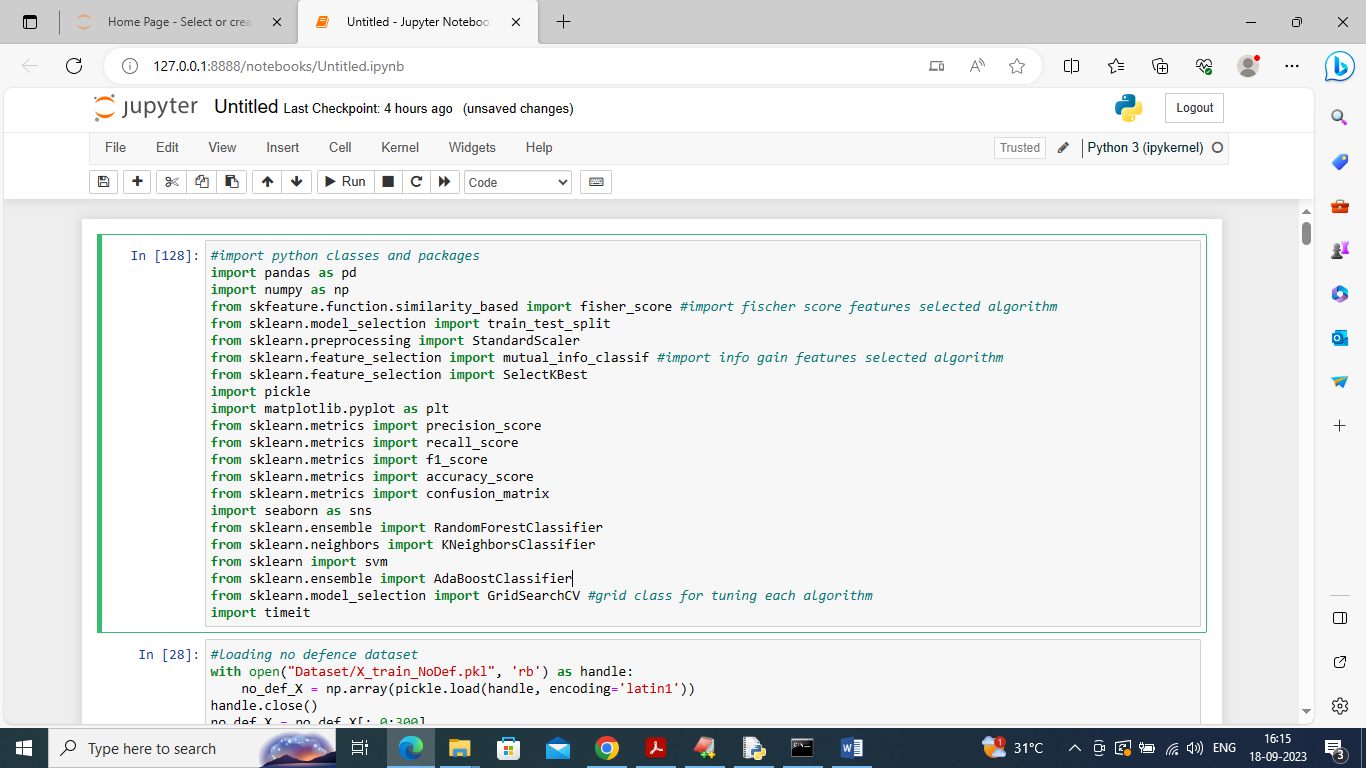
<https://github.com/fingerprintability/fingerprintability/tree/master>

Extension Concept

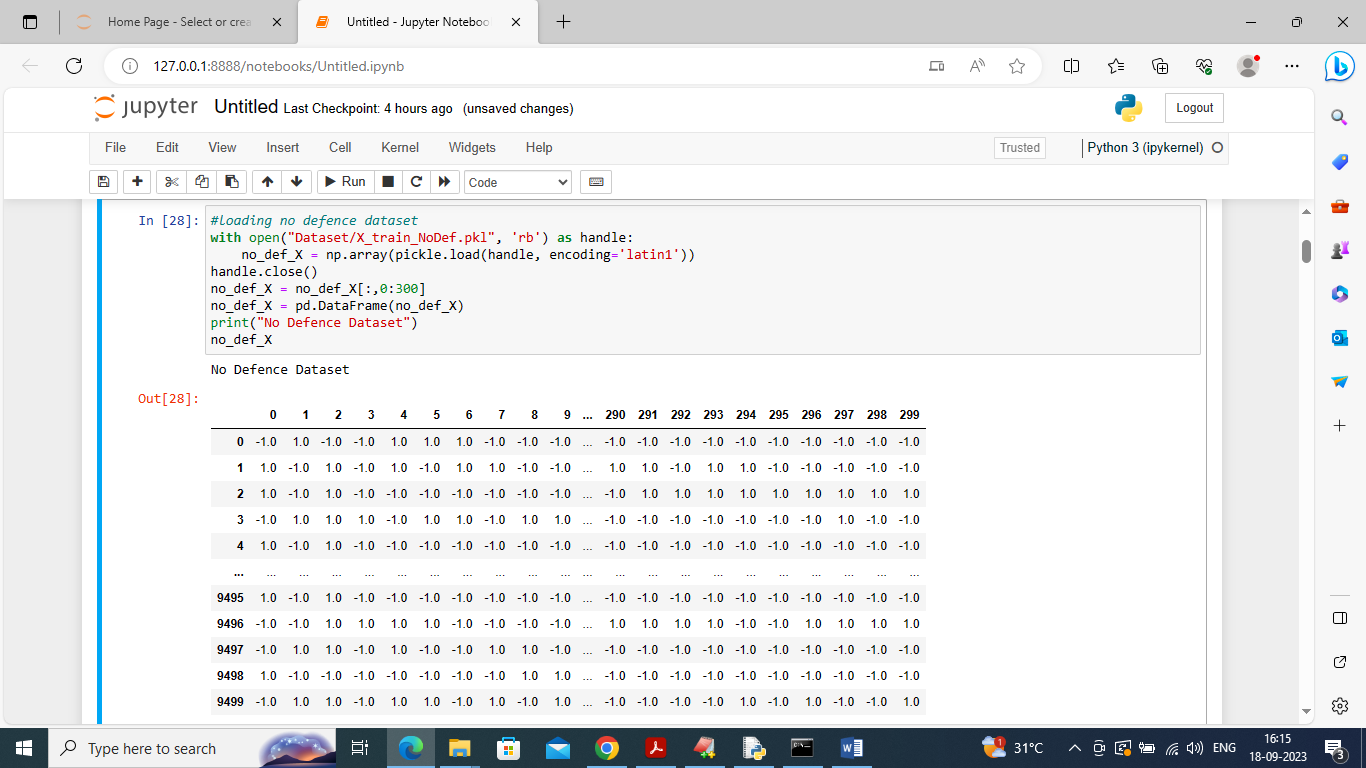
In propose work author has used traditional machine learning algorithms and have not used any advance machine learning algorithms like XGBOOST or ADABOOST so as extension we have experimented with ADABOOST algorithm which will use multiple estimator for training and then used estimator with best accuracy. Extension ADABOOST able to get 100% accuracy for TOR or Onion service classification.

SCREEN SHOTS

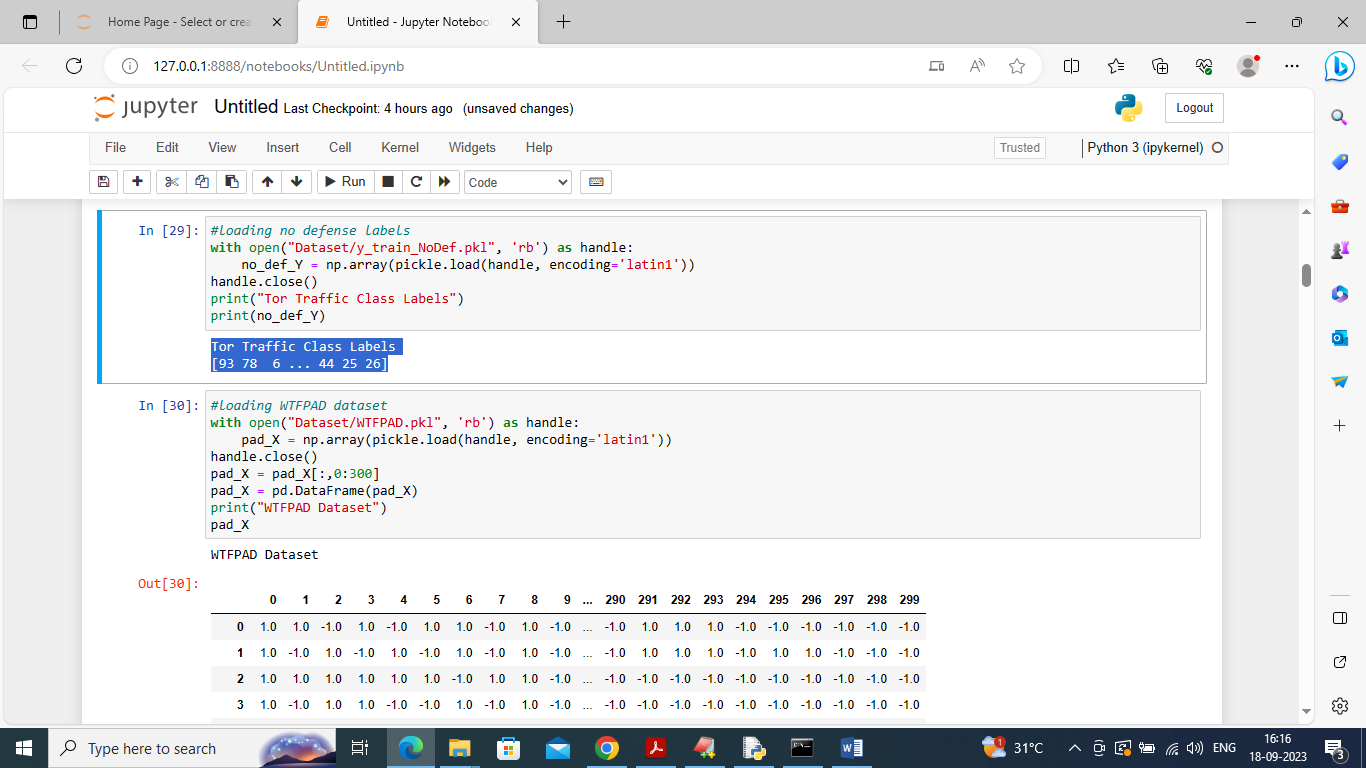
We have coded this project using JUPYTER notebook and below are the code and output screens with blue colour comments



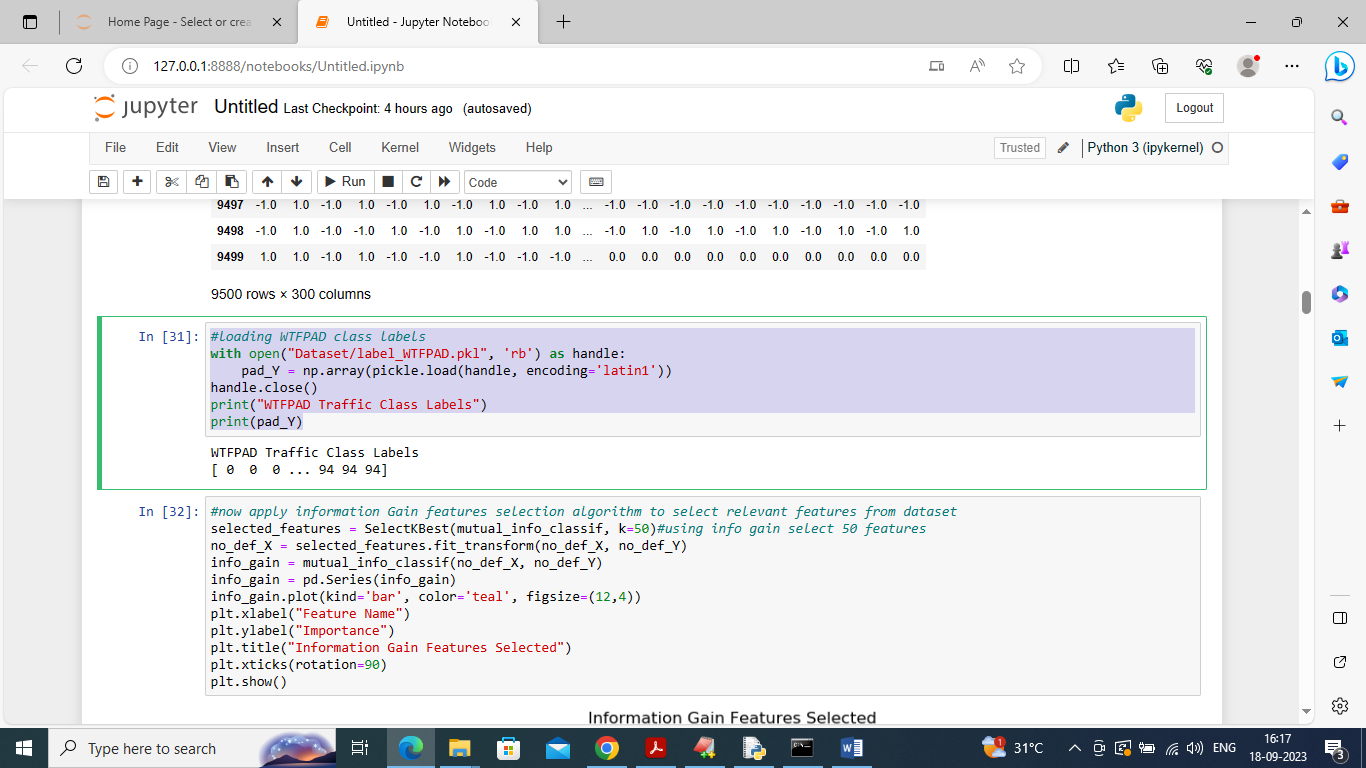
In above screen importing require python classes and packages



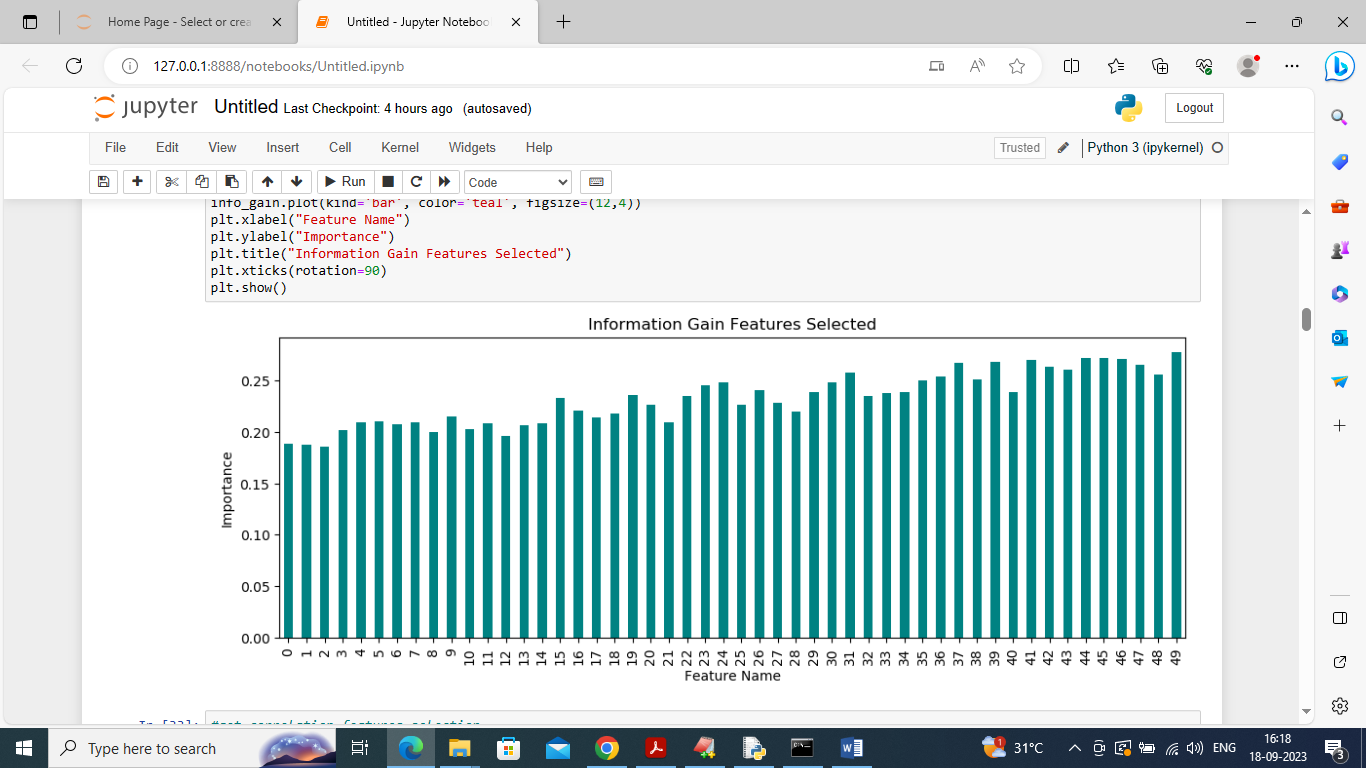
In above screen loading and displaying NO DEFENCE original TOR dataset



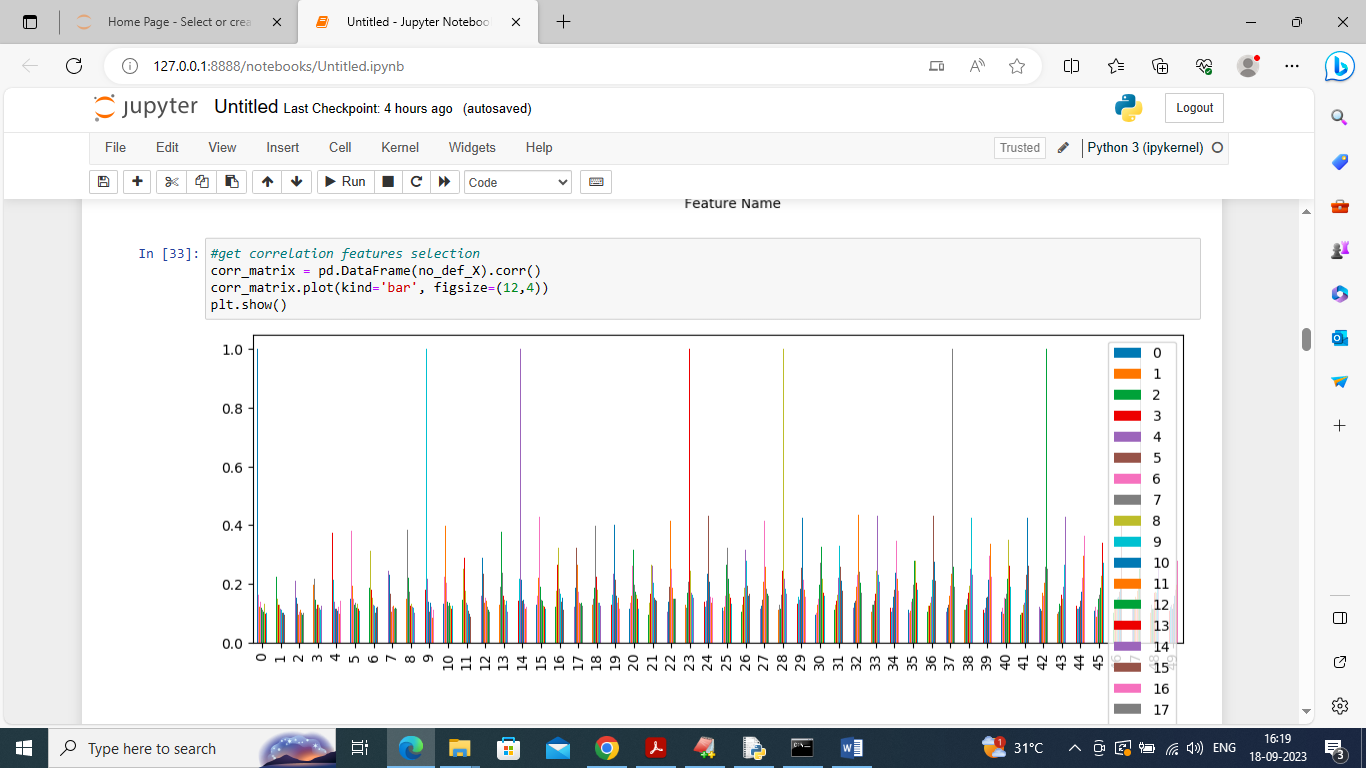
In above screen loading Original TOR class labels dataset and then loading and displaying WTFPAD dataset



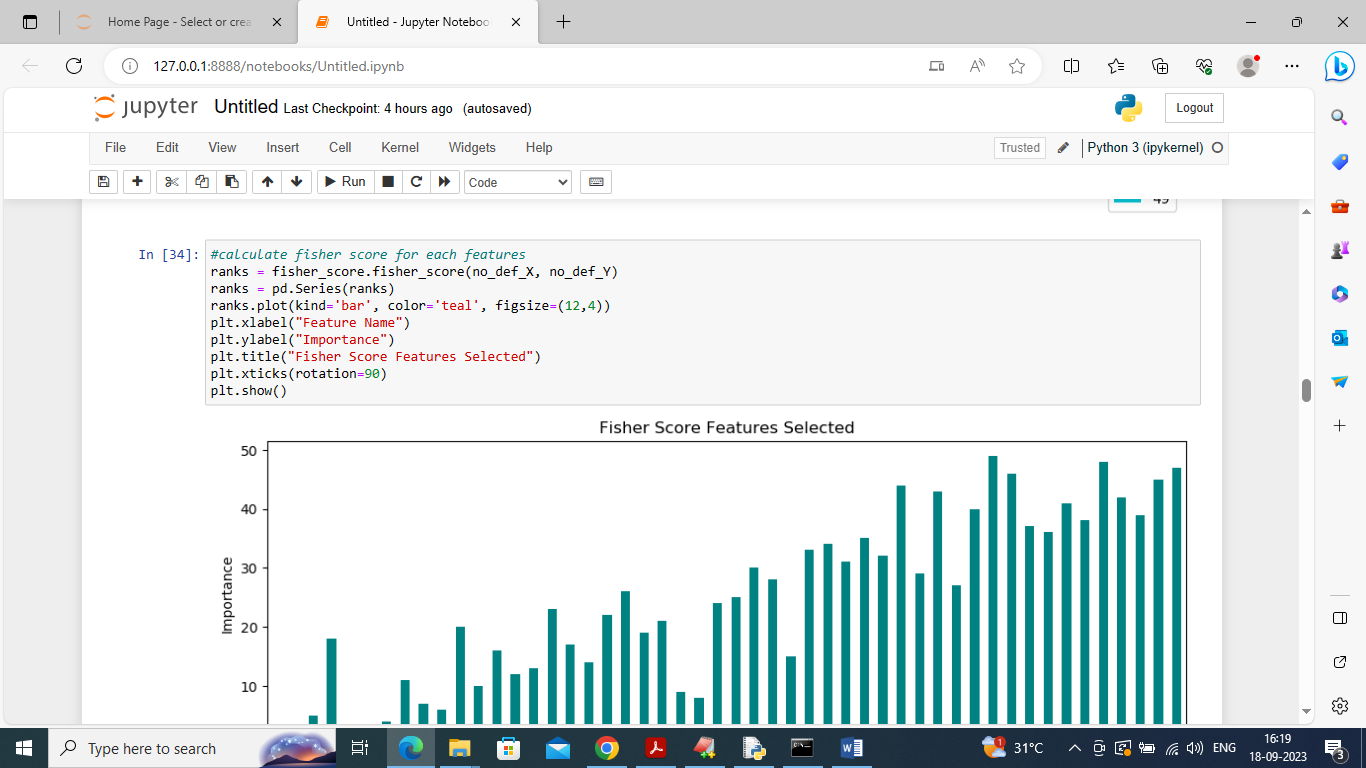
In above screen loading WTFPAD class labels and then defining code to select relevant 50 features from dataset using Information Gain algorithm and below is the selected features importance graph



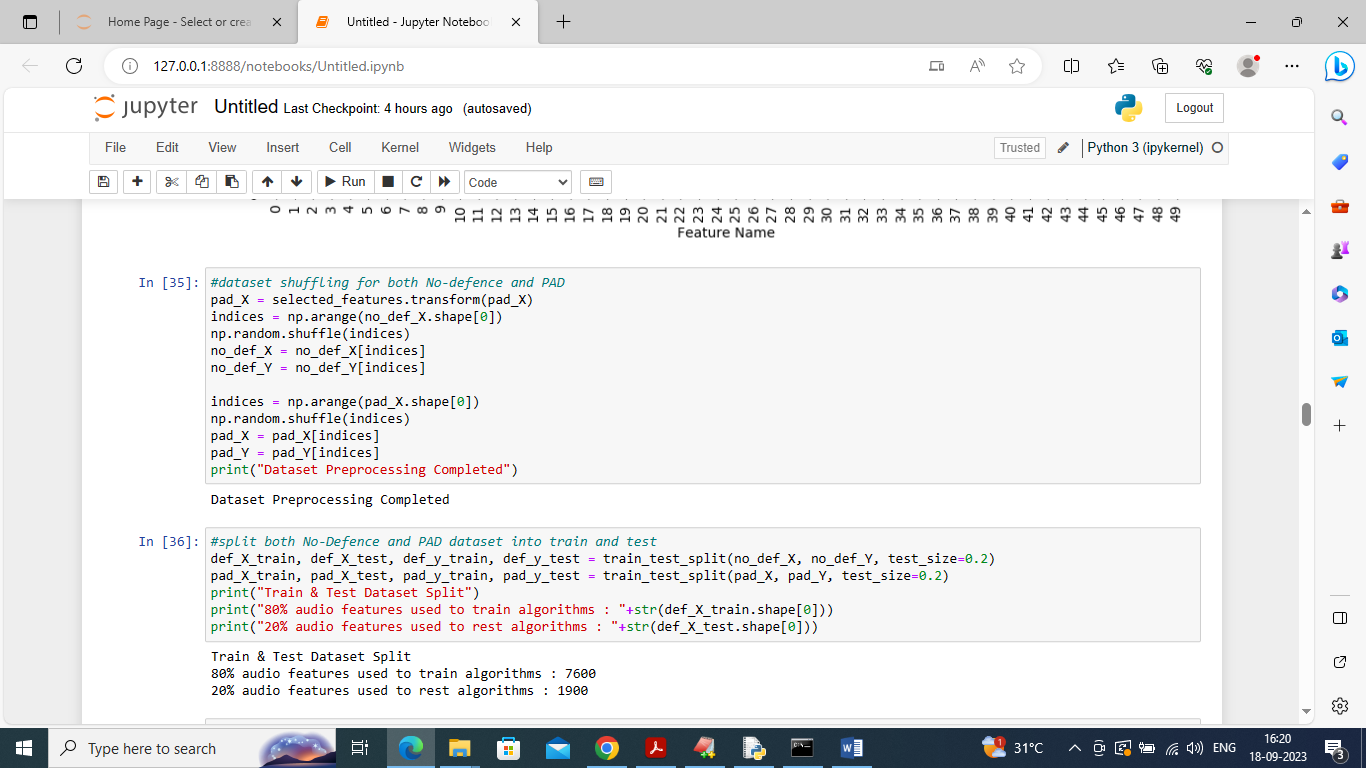
In above graph x-axis represents features names and y-axis represents importance or relevant score value



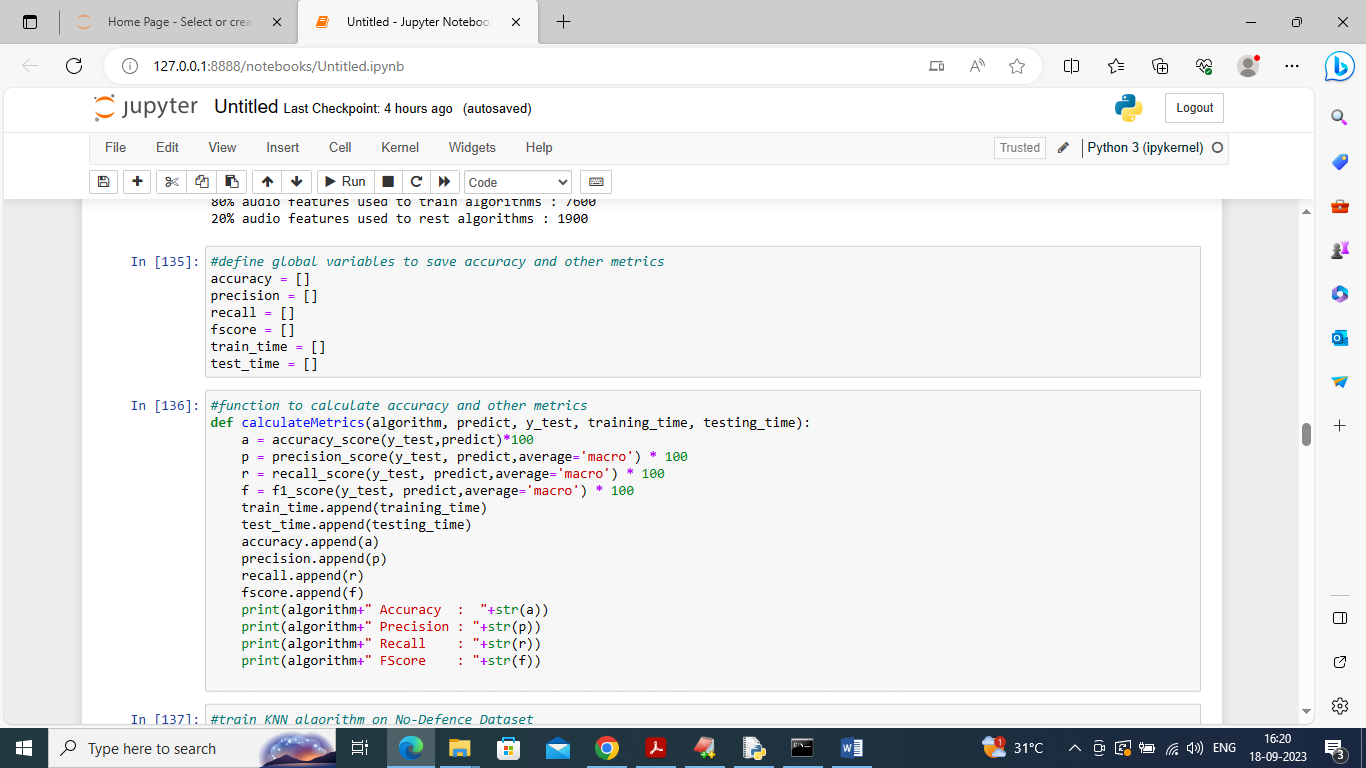
In above screen calculating features importance using Correlation Coefficient



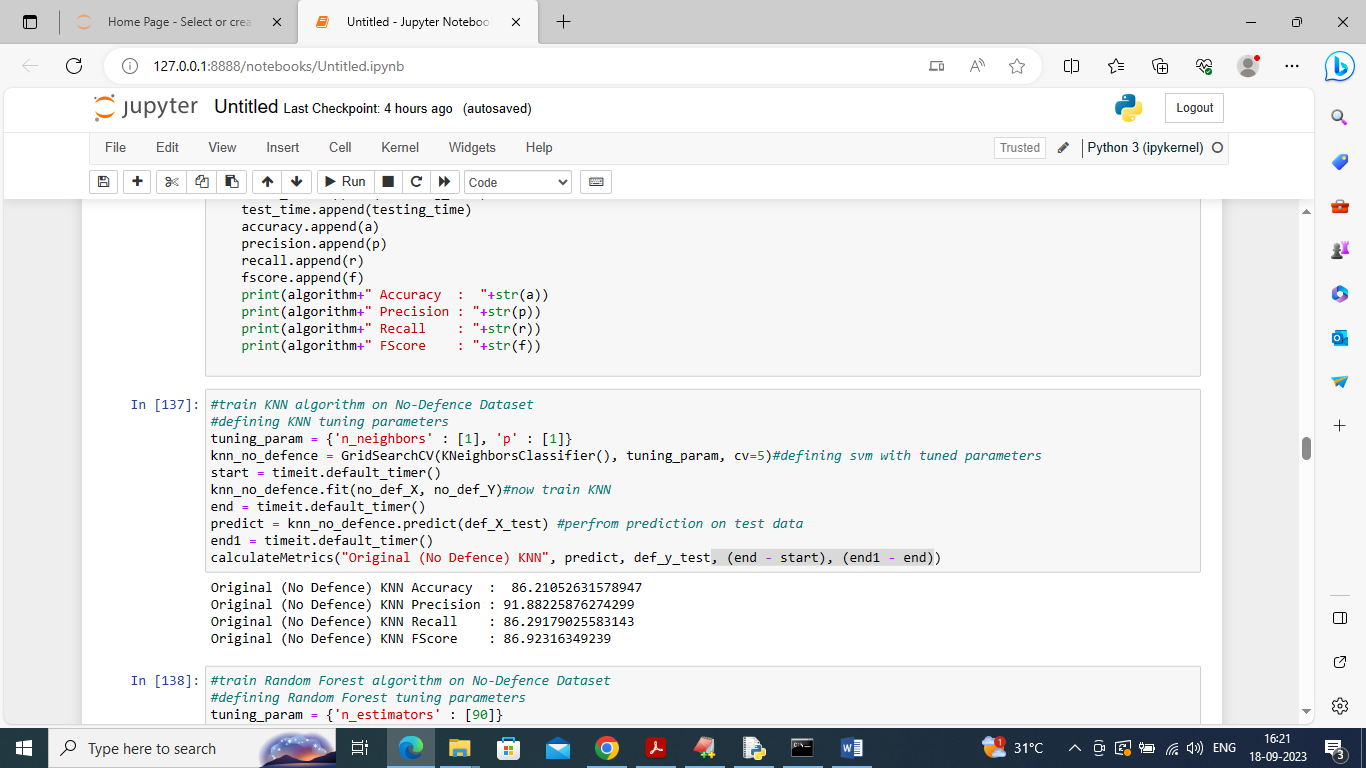
In above screen calculating features importance using Fisher Score



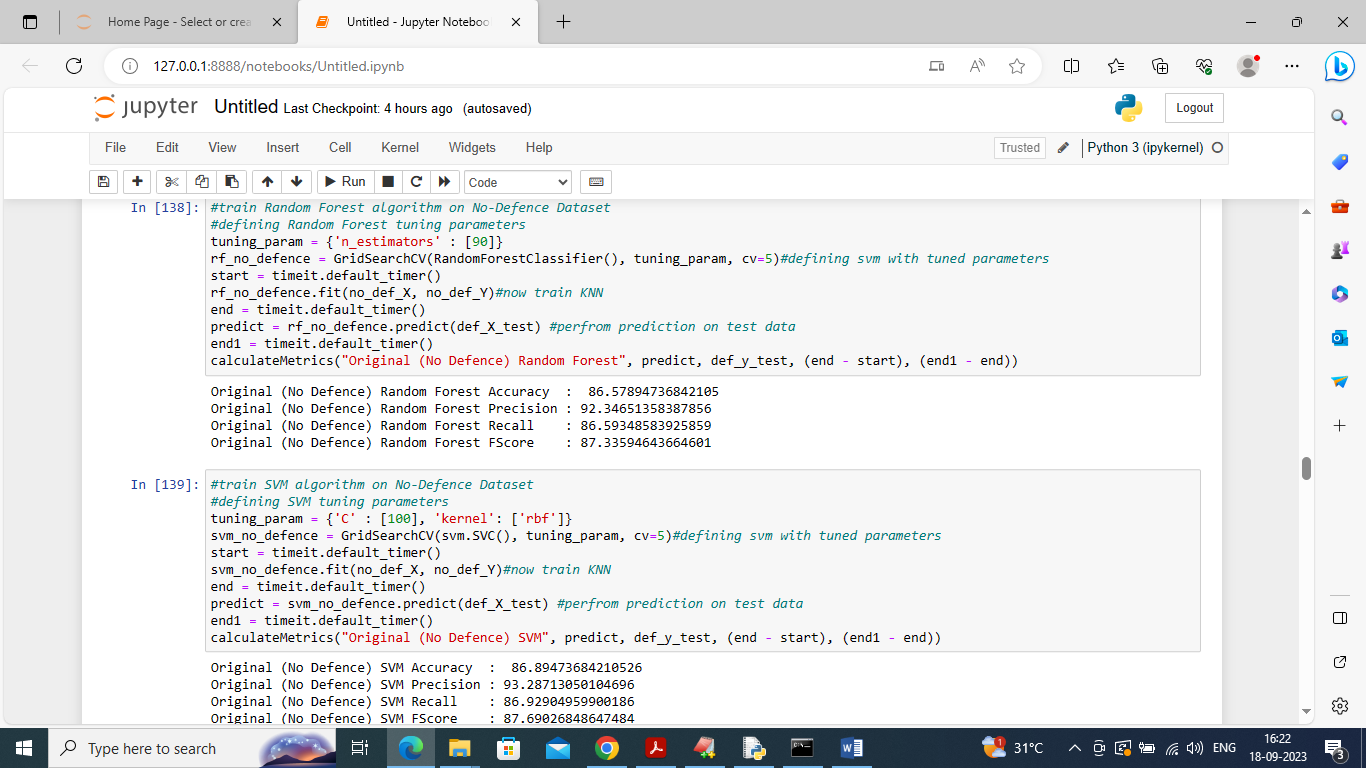
In above screen applying dataset processing like shuffling and splitting dataset into train and test



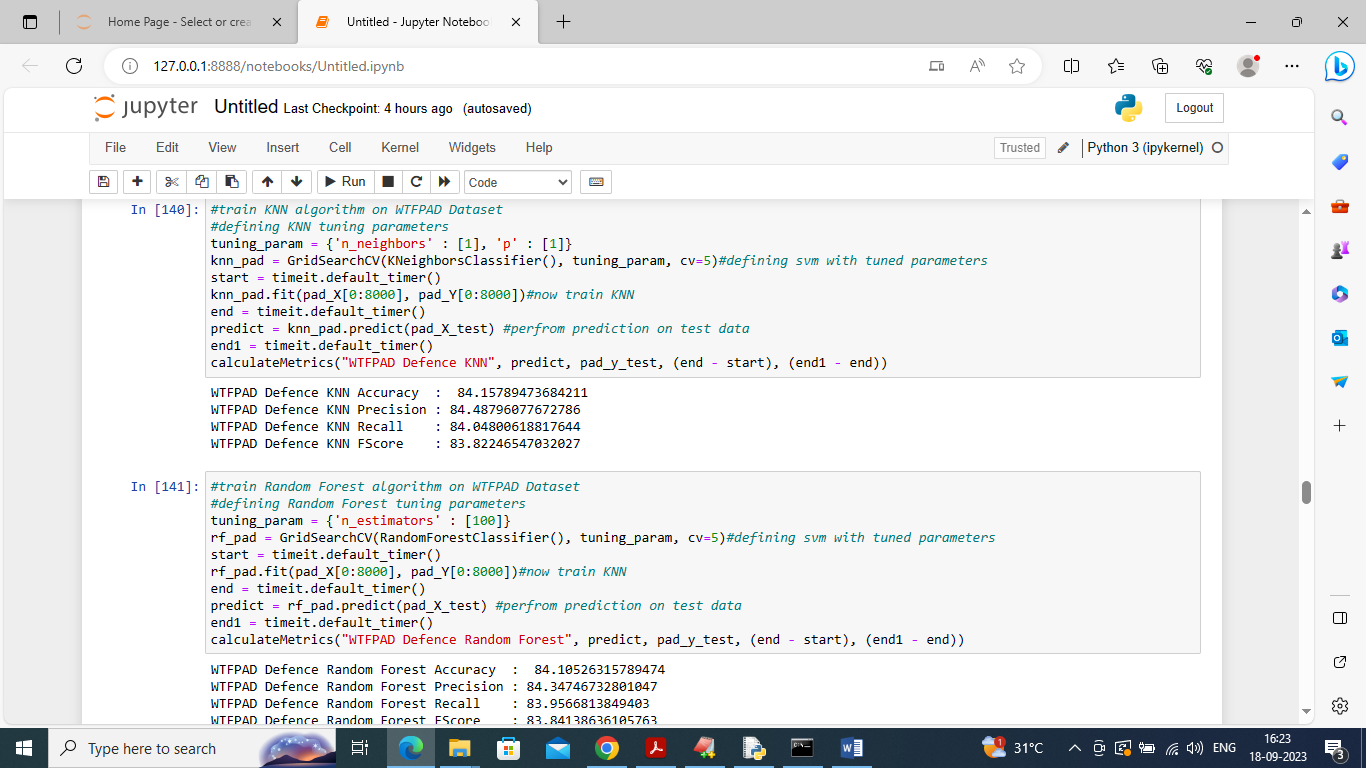
In above screen defining function to calculate accuracy and other metrics



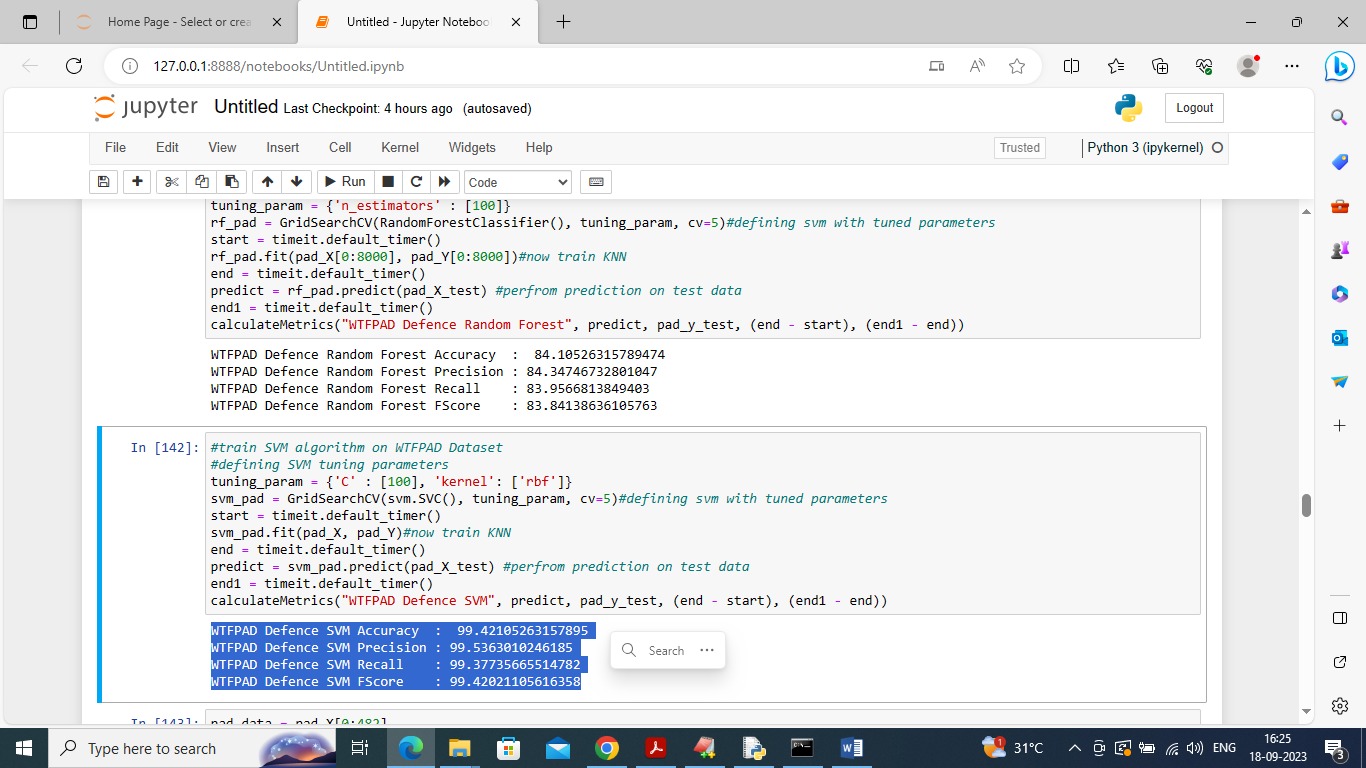
In above screen training KNN on original No Defence dataset and then KNN got 86% accuracy and in above screen KNN optimize by using hyper parameters



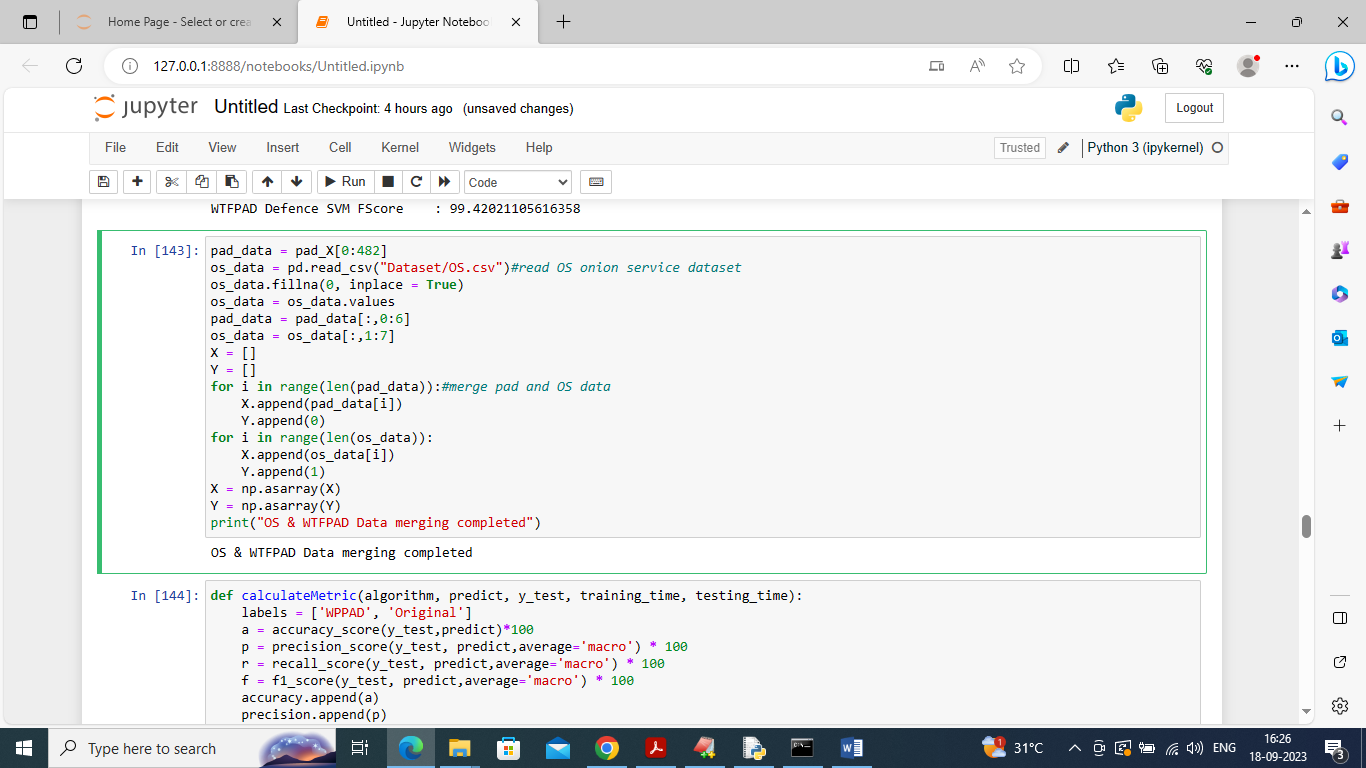
In above screen Random Forest got 86.57% accuracy on original no defence dataset and then SVM 86.89% which is higher than KNN and Random Forest on Original No Defence dataset. In below screen train above same algorithms on modified WTFPAD dataset



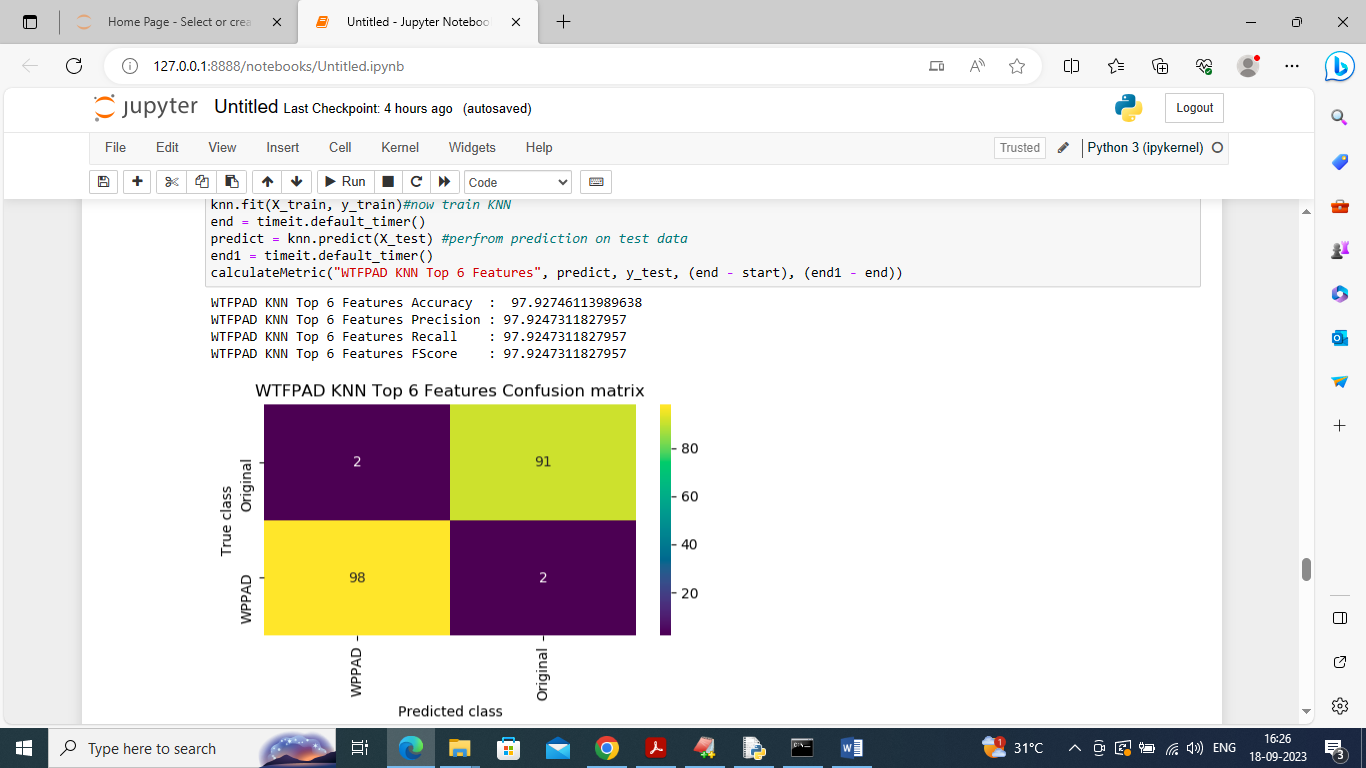
In above screen KNN got 84.15% accuracy on WTFPAD dataset and Random got 84.10% accuracy. In WTFPAD accuracy got little affected compare to original dataset

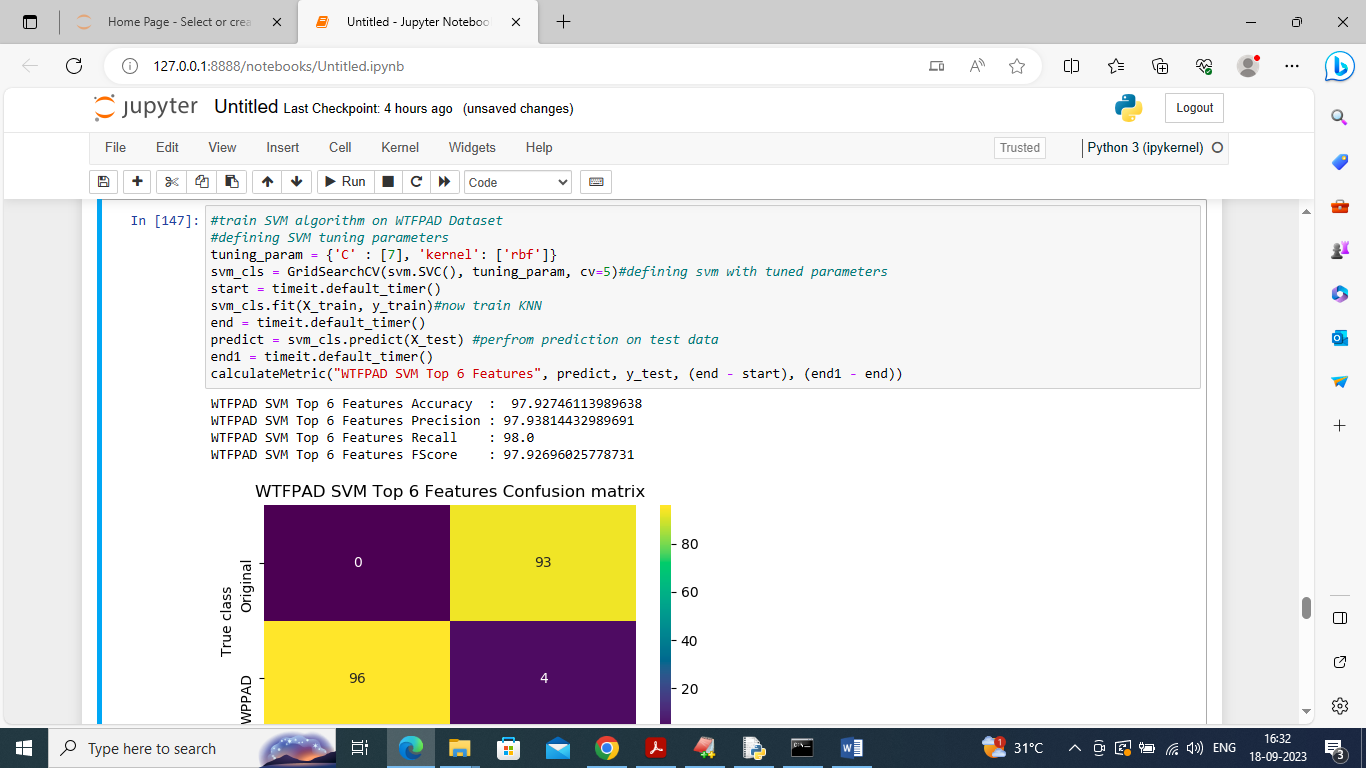


In above screen SVM got 99% accuracy on WTFPAD dataset which is higher than KNN and random Forest

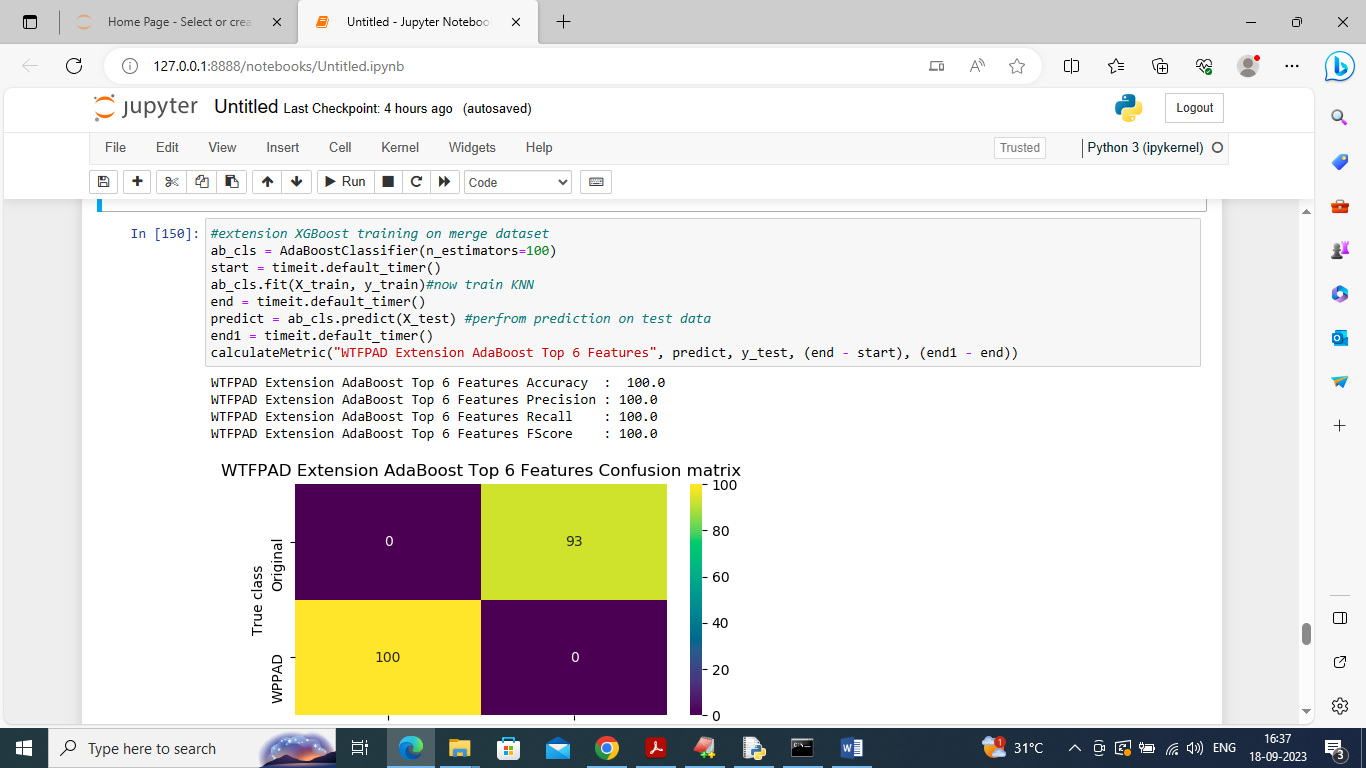


In above screen merging WTFPAD and OS Onion services dataset for TOR or onion classification

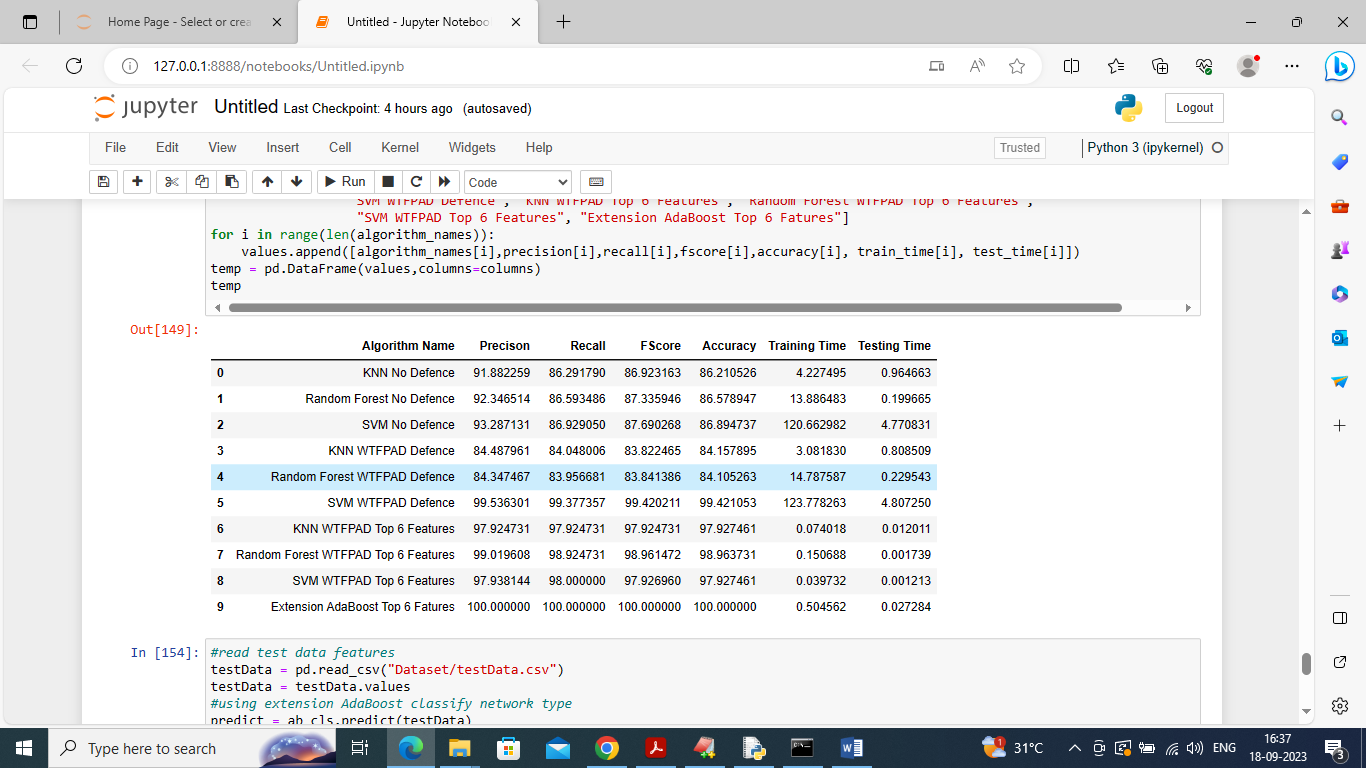


In above screen training KNN on merged TOR and OS TOP 6 features selected dataset and then after training KNN got 97.92% accuracy. In confusion matrix graph x-axis represents predicted Original OS service label and WTFPAD service label and y-axis represents true labels. Yellow and light green boxes contains correct prediction count and blue boxes contains incorrect prediction  
count. 

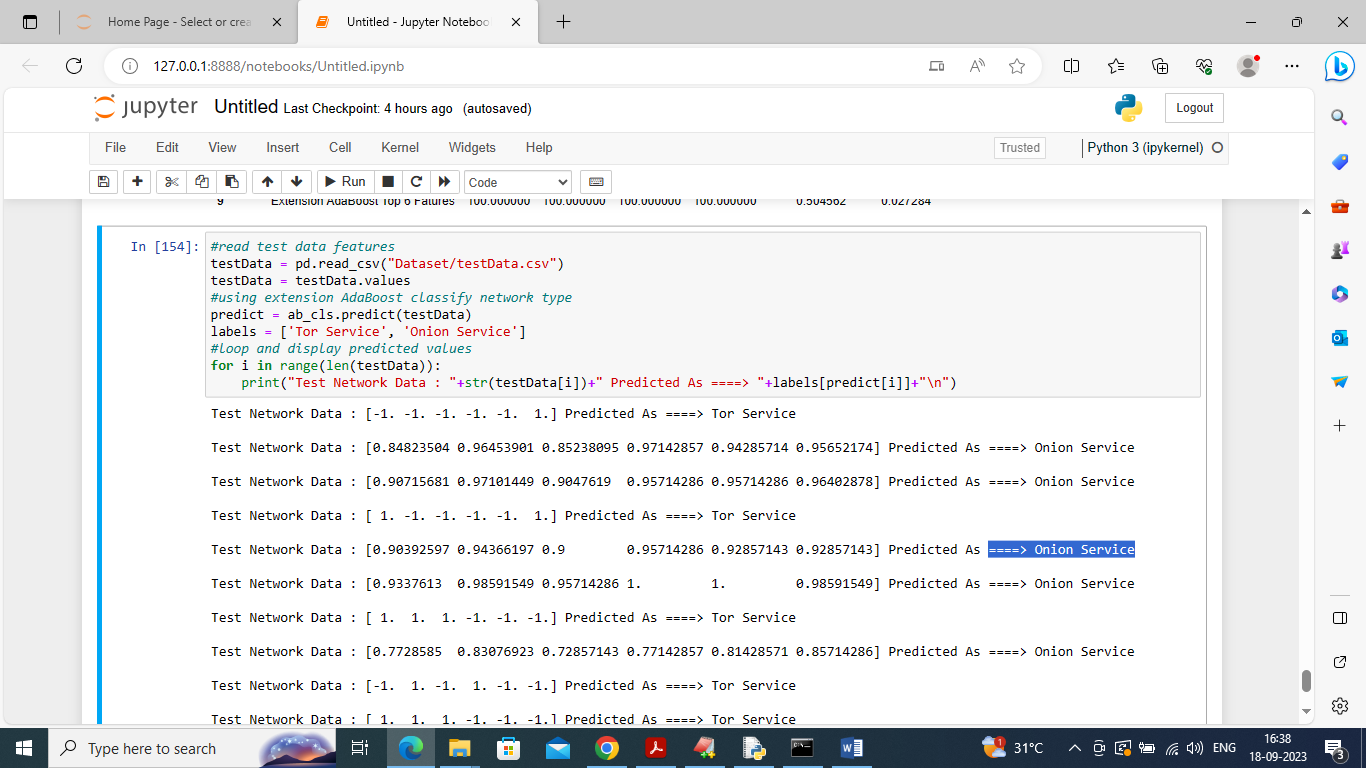
In above screen SVM got 97.92% accuracy



In above screen training extension ADABOOST algorithm and then after training extension got 100% accuracy



In above screen displaying all algorithm performance in tabular format and some algorithms can able to classify Tor and Onion Services with more than 95-99% accuracy



In above screen reading test Network data and then using extension ADABOOST algorithm classifying network traffic as Tor or Onion services. in square bracket we can see test data and after arrow symbol can see predicted service type