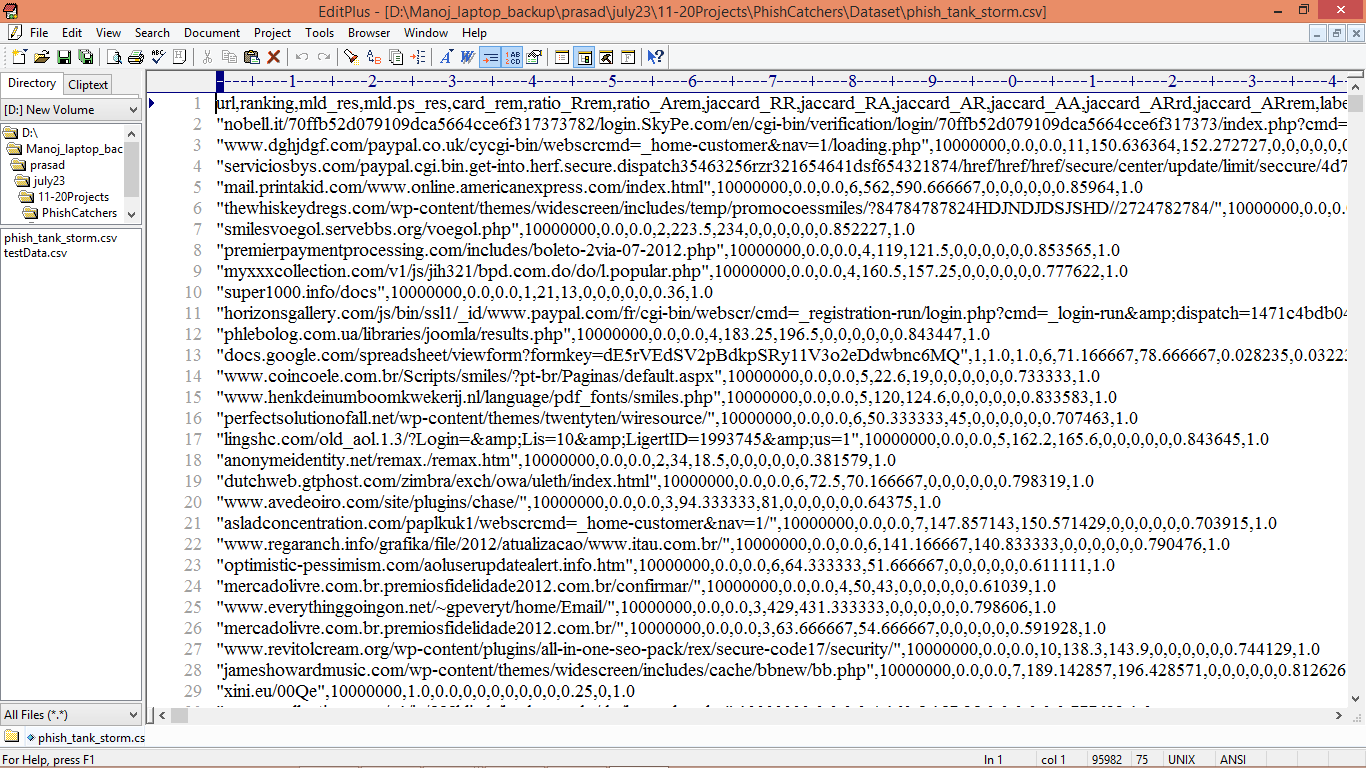
PhishCatcher: Client-Side Defence Against Web Spoofing Attacks Using Machine Learning

Now-a-days we are heavily dependent on online data such as Online news, Email Messages, Online Reviews, Online Post and many more. This online content access open doors for attackers to allure normal users by sending enticing messages of jackpot wining with fake phishing URL or spoofing websites. Whenever user click on such URL or navigate to spoofing website then they will ask user to enter login details and then attackers will use those login details to gain access to banking or any other financial websites and grab or steal all user money or any other secret information.

To avoid such URL many existing machine learning and signatures based algorithms were introduced but there detection rate are not accurate so author of this paper employing Random Forest algorithm to detect phishing URLS. Random Forest algorithm has inbuilt support for features optimizations and selection which help in enhancing prediction accuracy. Random forest will apply group of trees on dataset to filter and remove irrelevant data and then select only optimized features.

Author has given much more details which you can read from the base paper. To train propose algorithm author has used PHISHTANK dataset which contains 1000’s of normal and phishing URL and by using this dataset we can predict URL as SAFE or phishing. Apart from training author has developed CHROME based extension which will analyse all visiting URLS and then alert user with SAFE or phishing URL’S. Propose Random Forest algorithm is comparing with existing SVM algorithm

In below screen displaying dataset used in this project



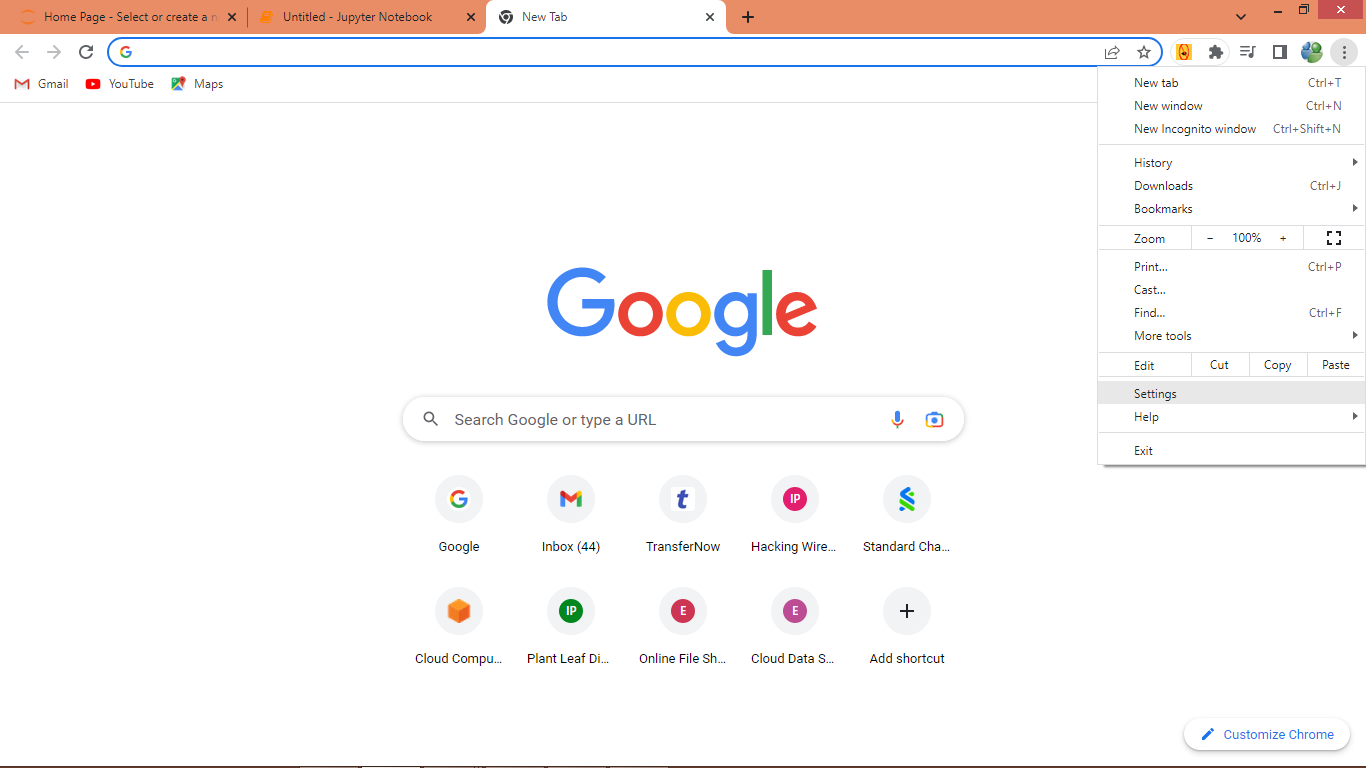
In above dataset screen first row contains dataset column names and remaining rows contains dataset values as URLS and labels and by using above dataset will train and test ML models performance

Extension Concept

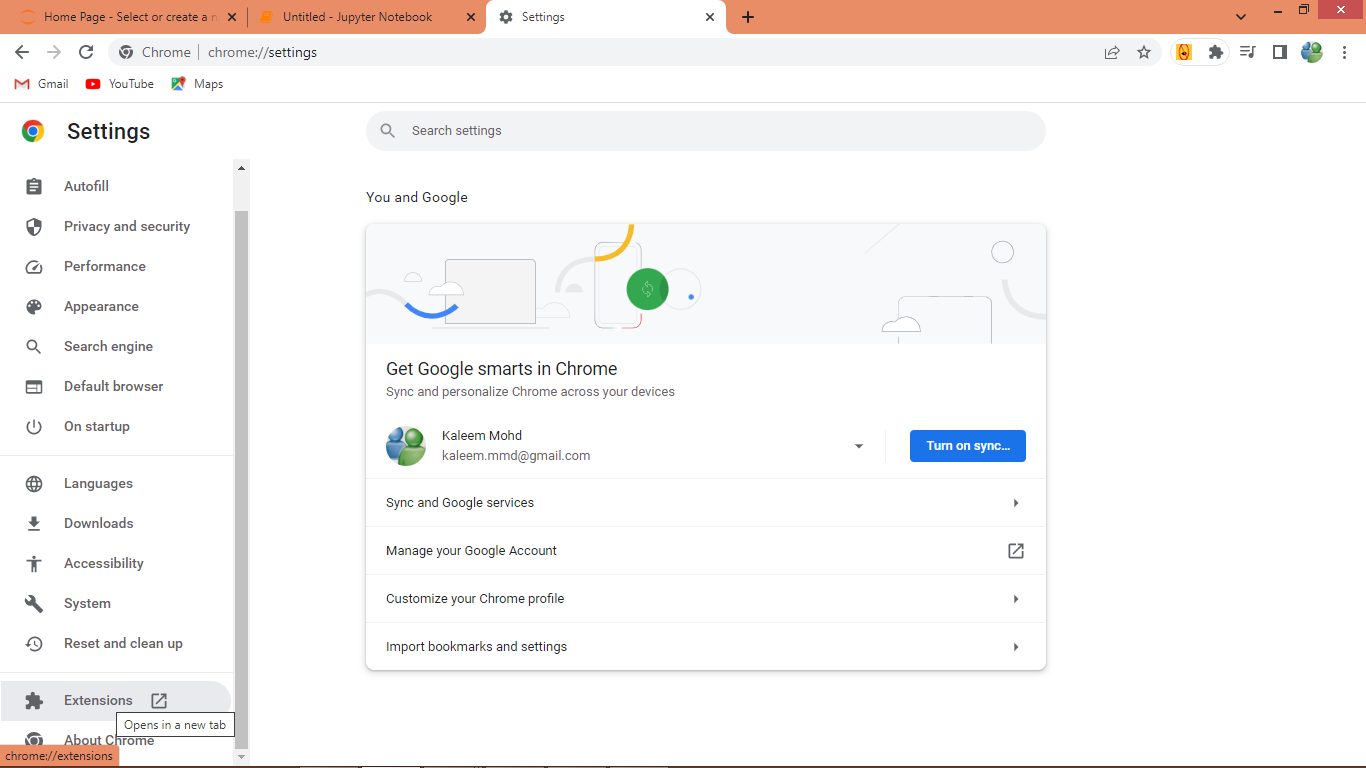
Similar to Random Forest we have another advanced algorithm called XGBOOST which will filter dataset by using trees of forest or group of estimators so it can optimized features better than Random Forest and can yield high accuracy so extension we have added XGBOOST algorithm.

SCREEN SHOTS

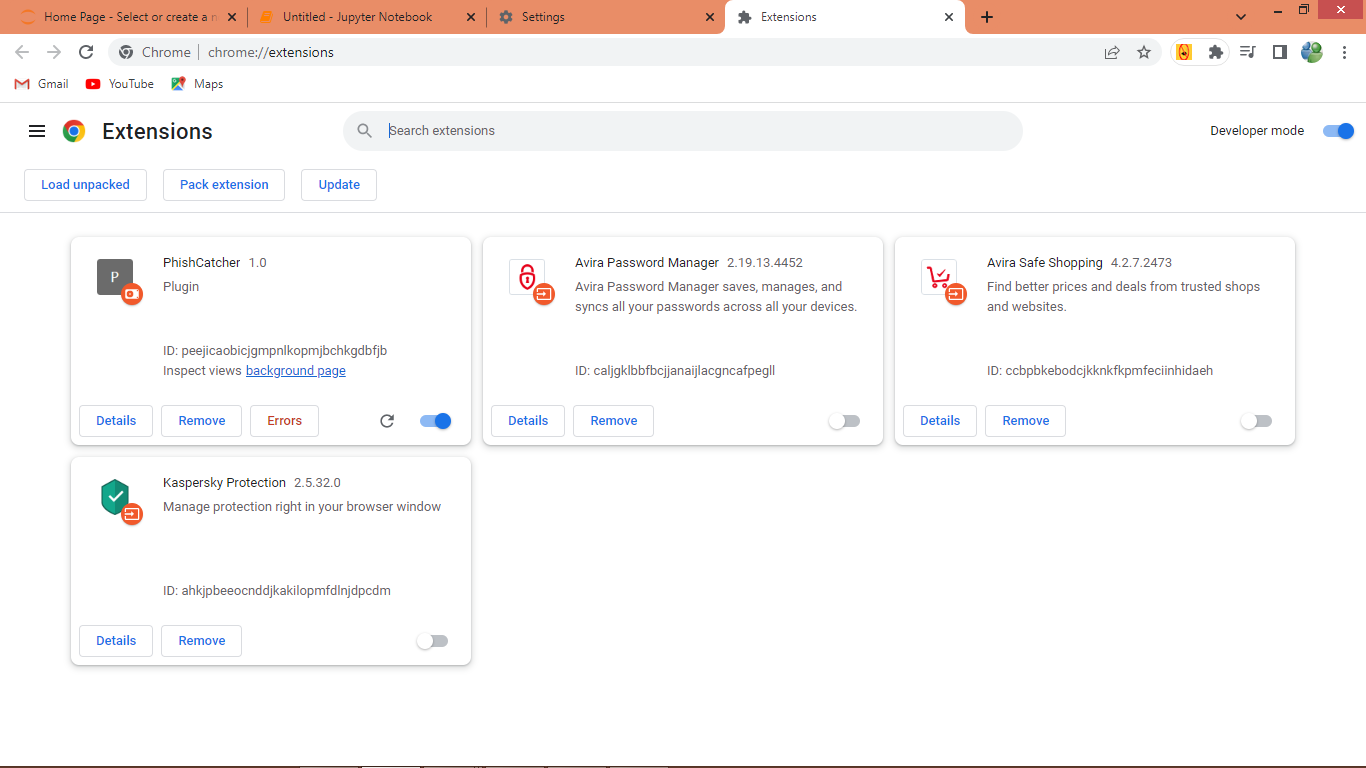
To add extension to chrome browser first go to settings like below screen



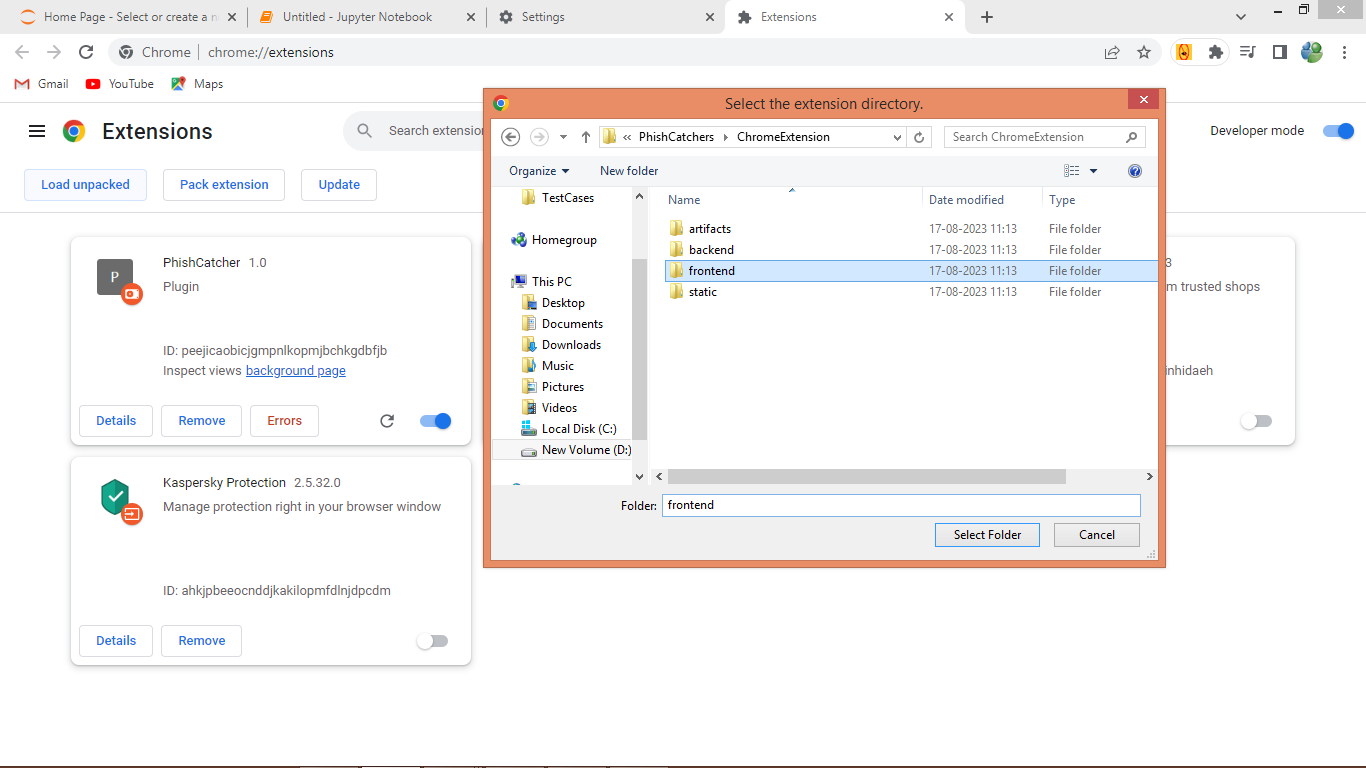
In above screen from chrome drill down select settings to get below screen



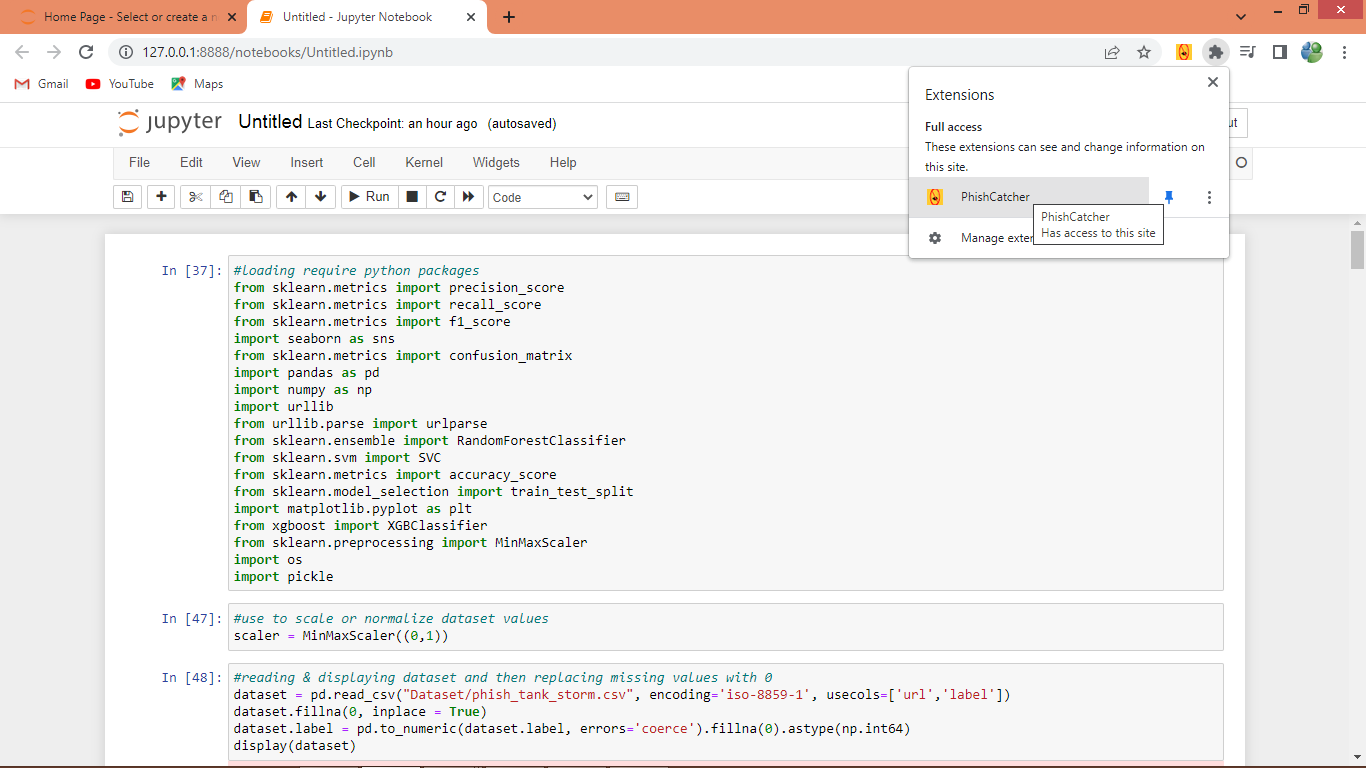
In above screen from left side select Extension in the bottom to get below page



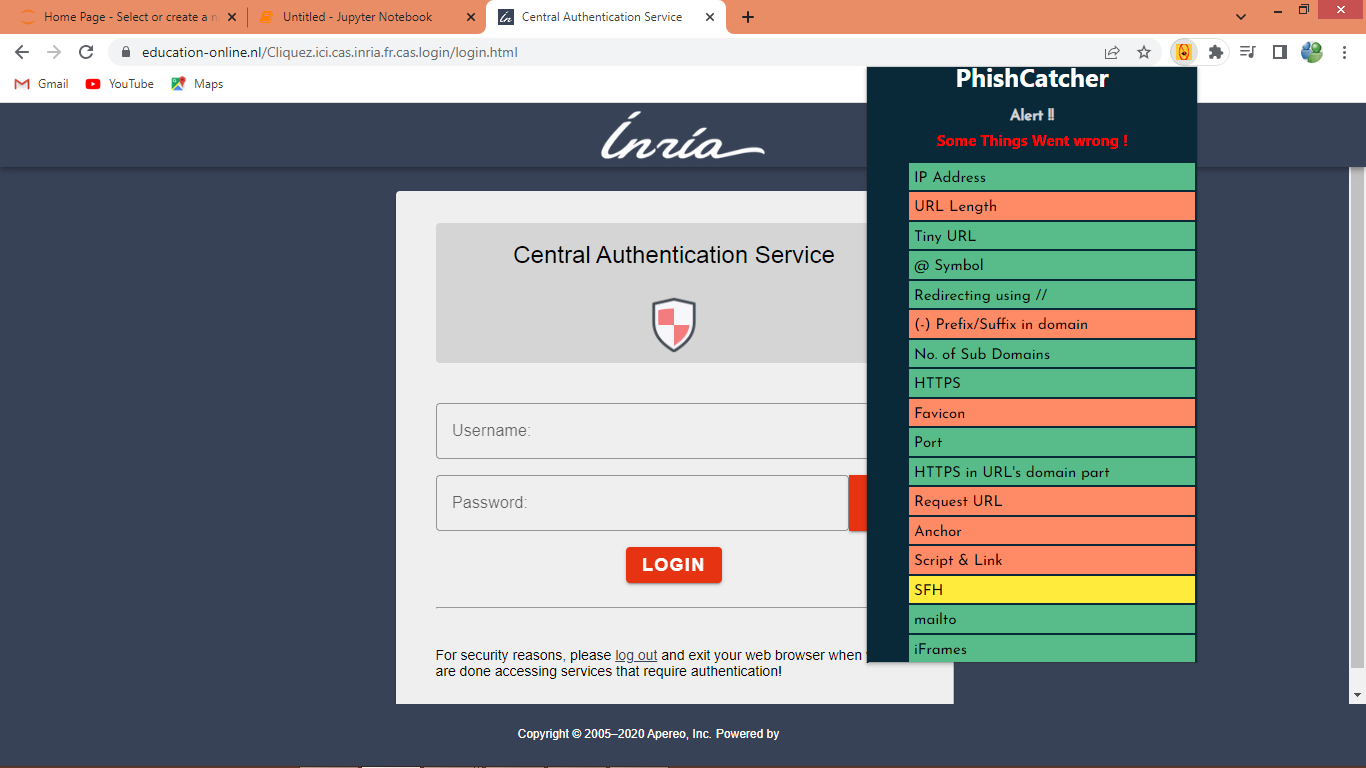
In above screen from top right corner enabled ‘Developer Mode’ and then from left top corner click on ‘Load Unpacked’ and then browse and upload ‘frontend’ folder from ‘Chrome Extension’ folder which you can find inside chrome folder and while uploading you will get below screen



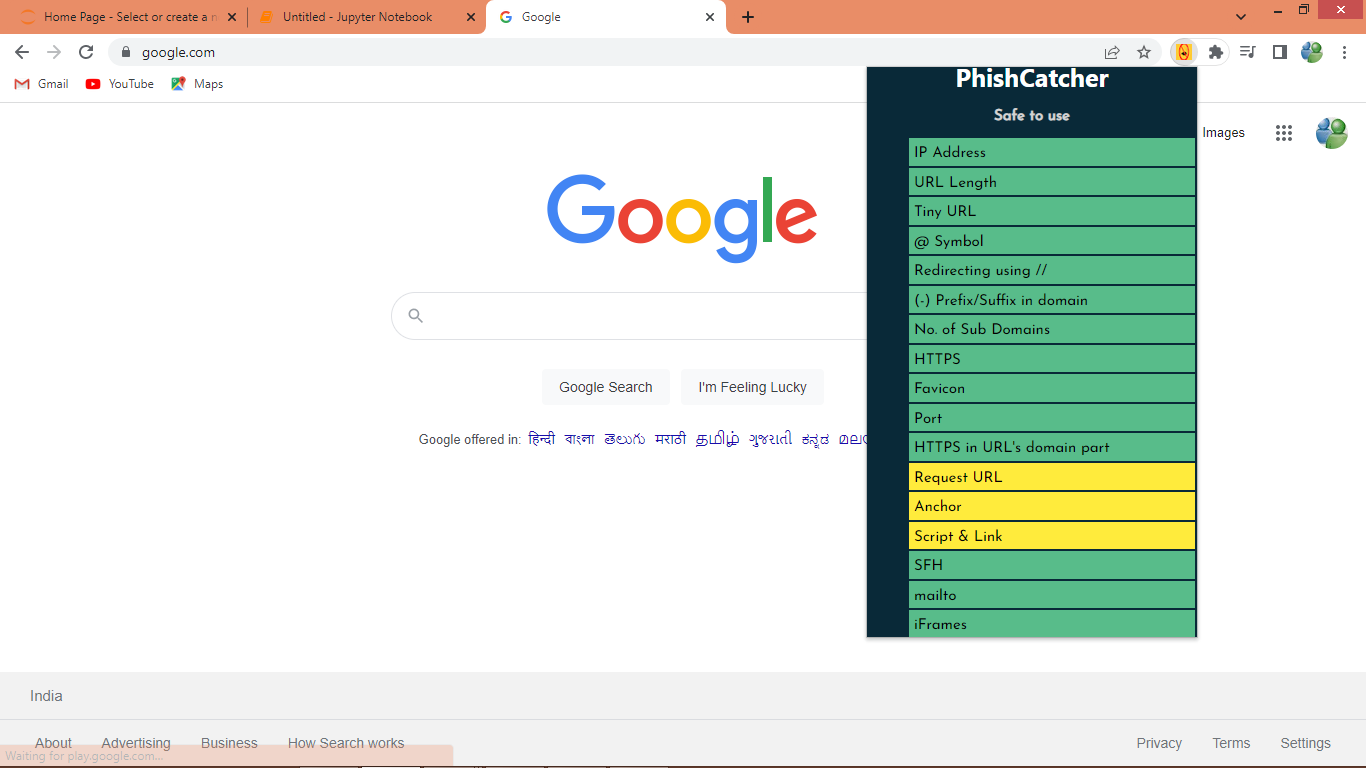
In above screen uploading frontend folder and then in browser at top right corner you can see ‘tree’ icon just click on it to view ‘Phish Cather’ extension like below screen



In above screen you can see phish catcher extension and now in browser you can enter any URL and click on ‘Phish Catcher’ extension to get output like SAFE or ALERT which I am showing in below screen

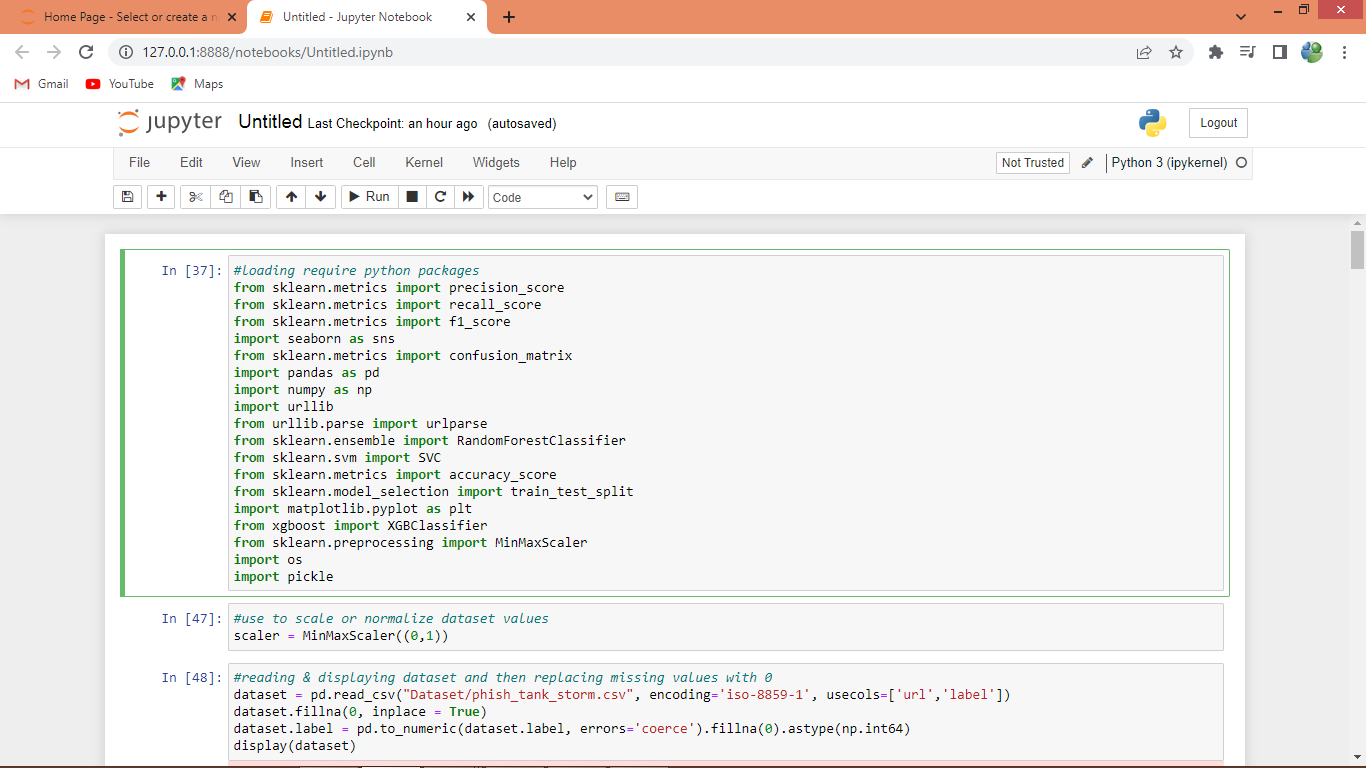


In above screen from base paper I entered URL as ‘https://www.education-online.nl/Cliquez.ici.cas.inria.fr.cas.login/login.html’ and then press enter key and then click on ‘Phish Catcher’ extension to get above output and below is the normal URL output

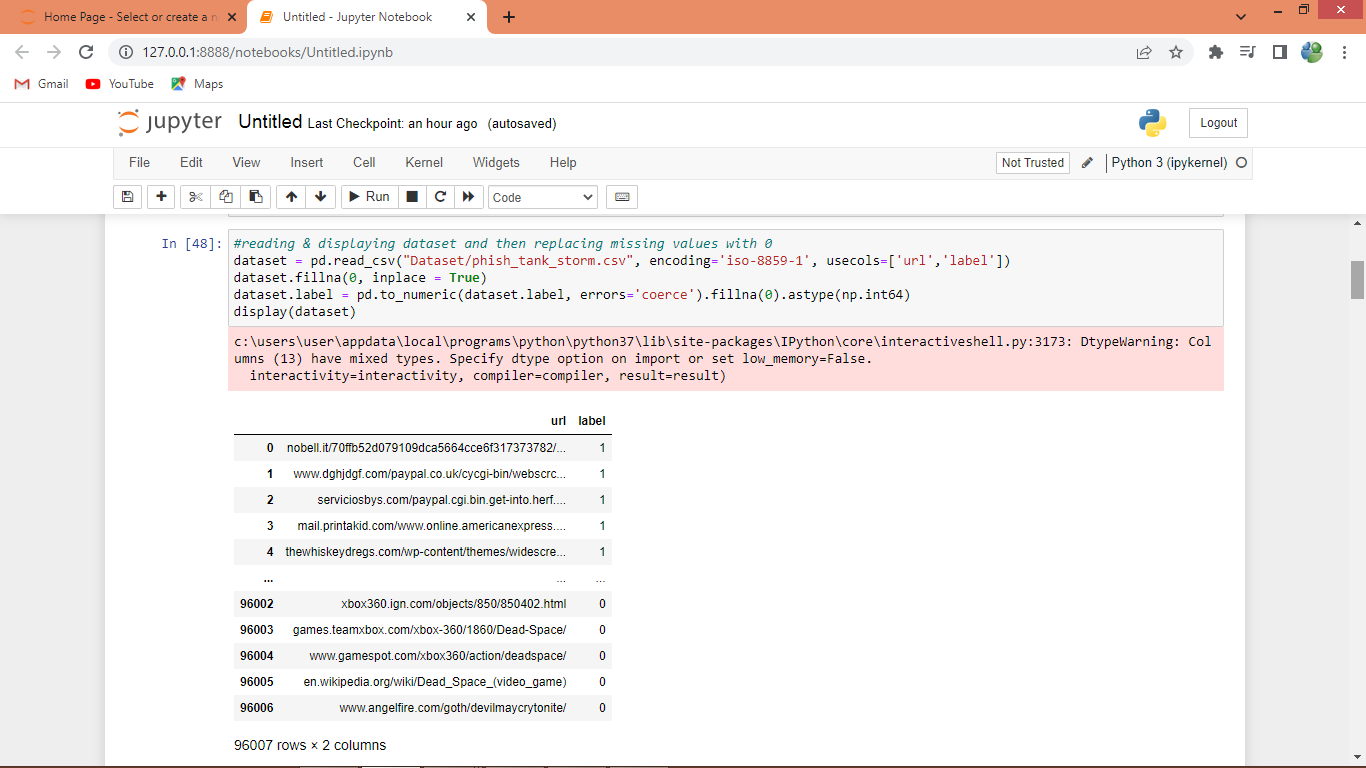


In above screen for Google.com we got output as ‘Safe to use’ and similarly you can test with any URL.

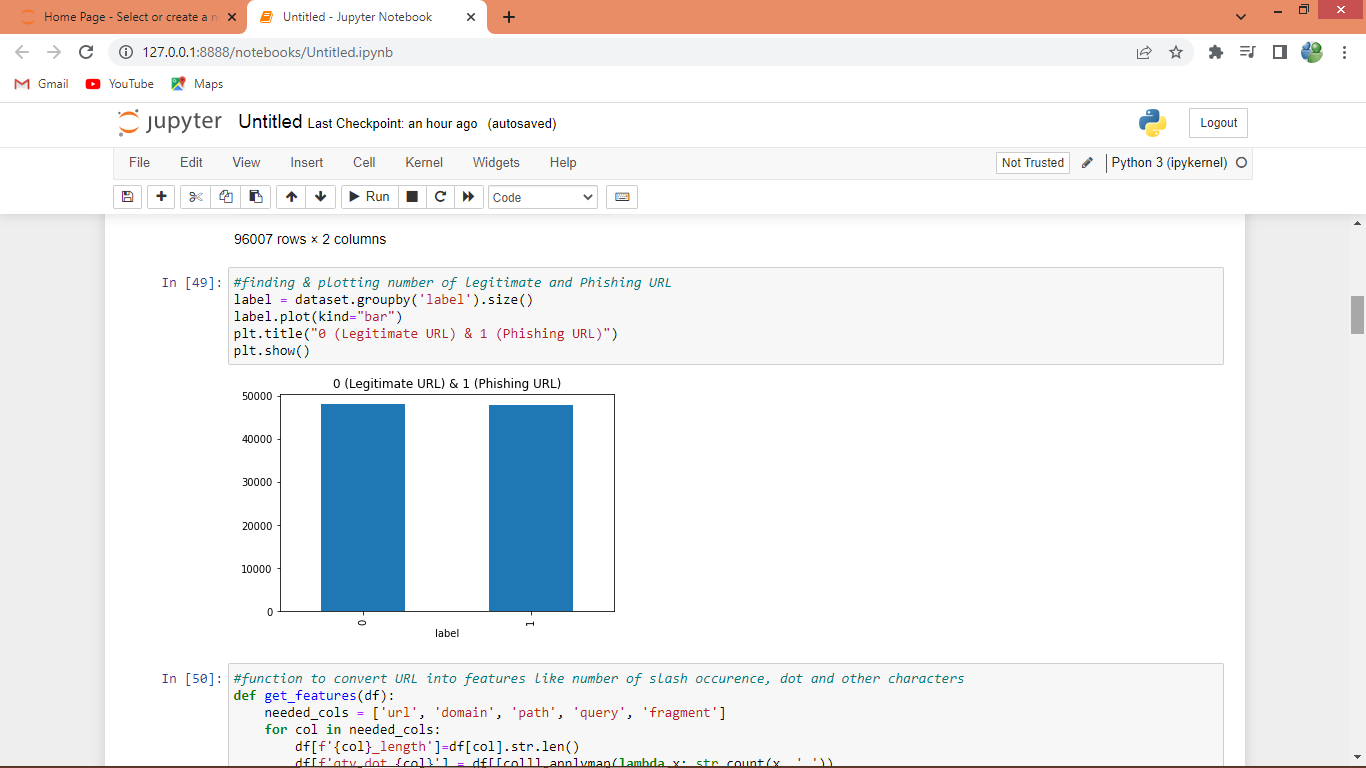
In below screen we are showing JUPYTER output with training and blue colour comments



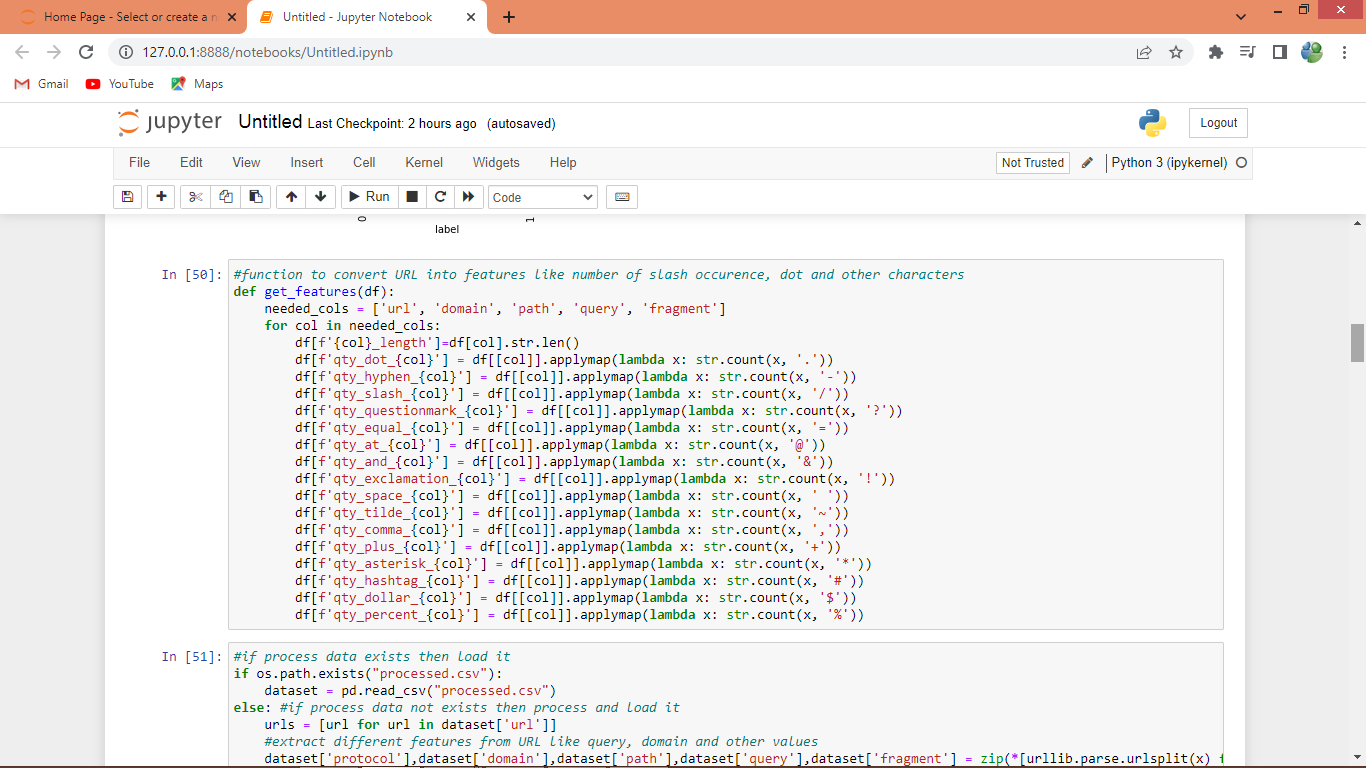
In above screen importing python packages and classes



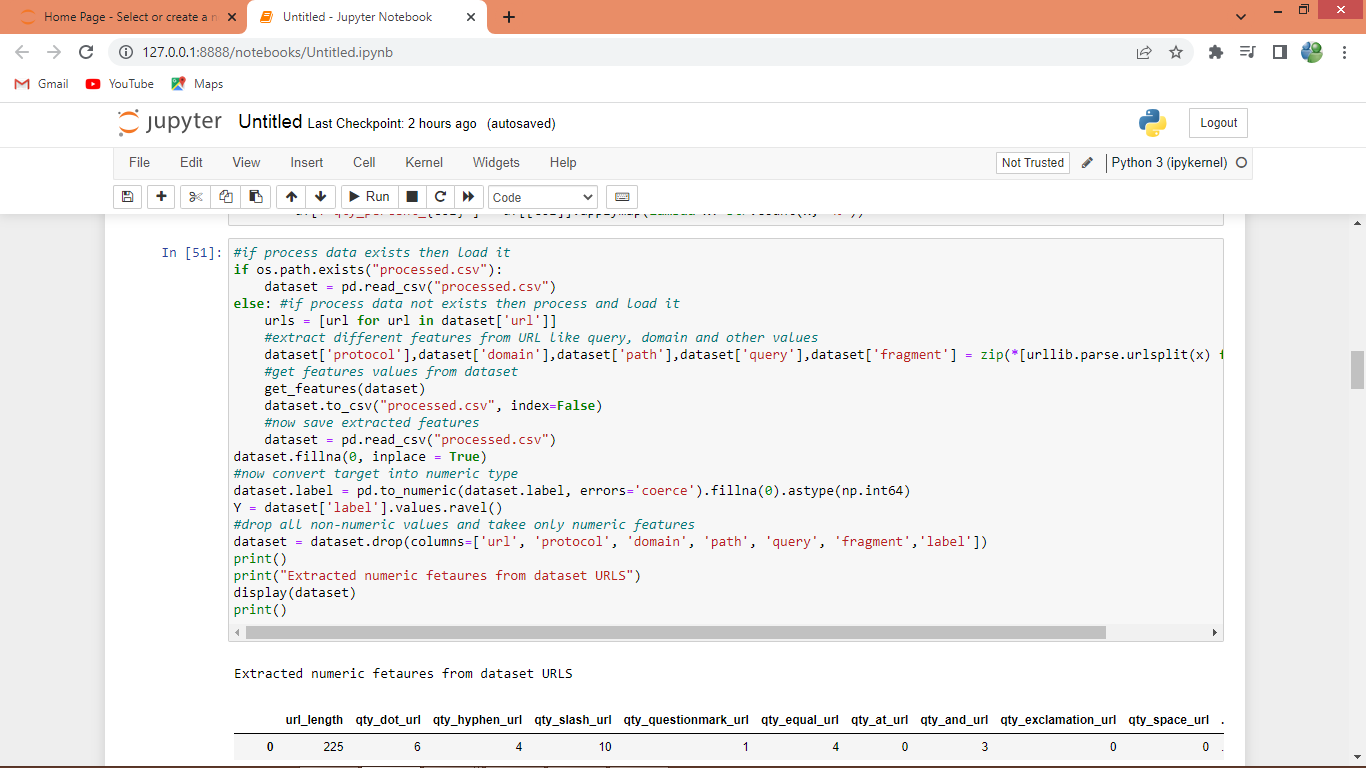
In above screen loading and displaying dataset values



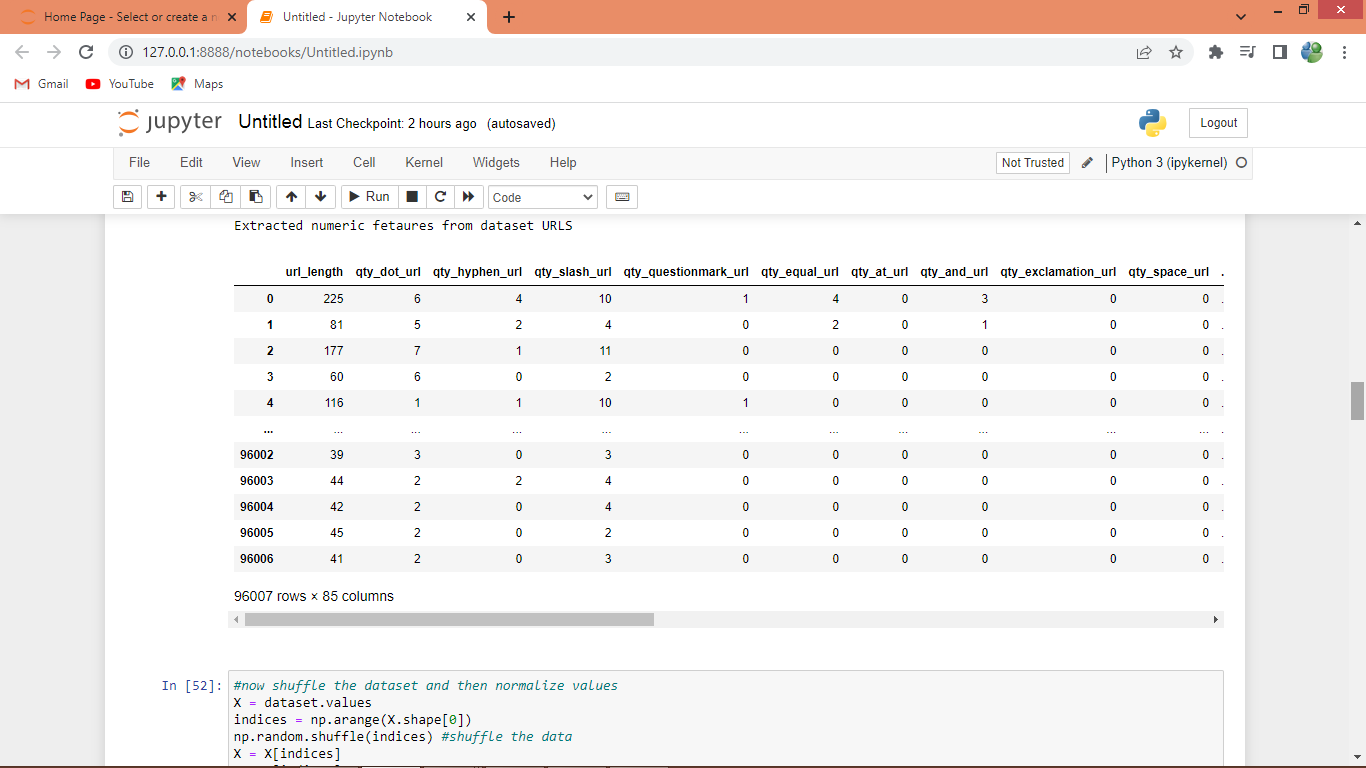
In above screen finding and plotting graph of normal and phishing URL where n graph x-axis 0 represents normal URL and 1 represents phishing



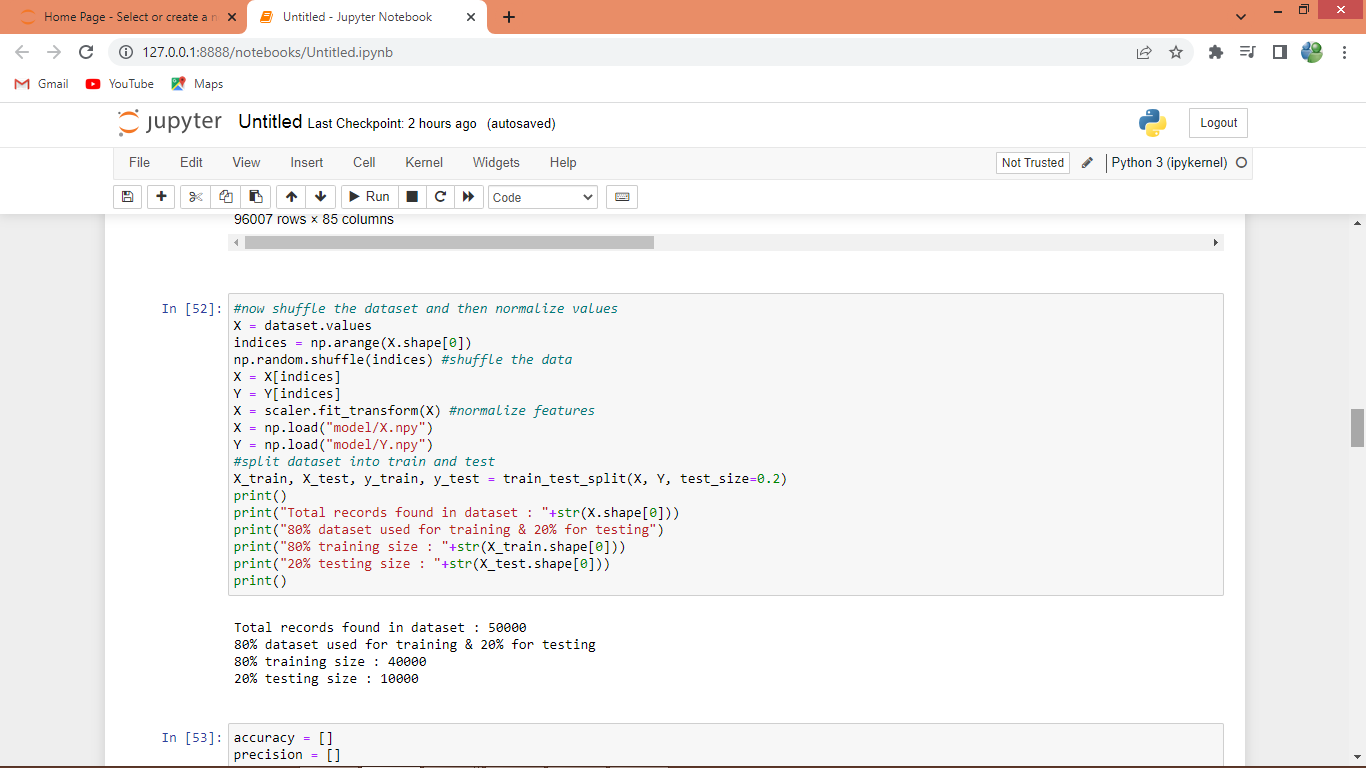
Above function is used to extract features from URL as number of dots, special symbols etc.



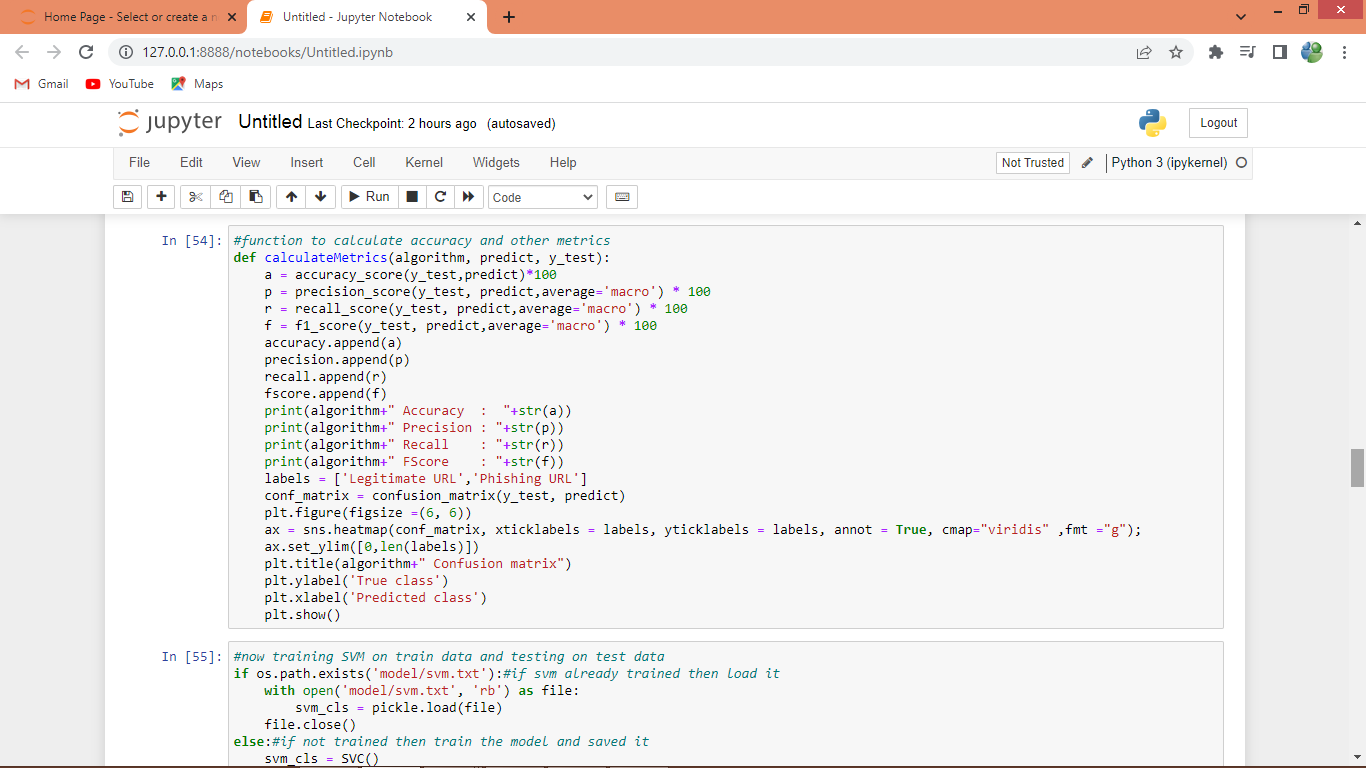
In above screen processing dataset to convert all URLS into numeric values where application will count number of dots, special symbols and other features from URL and those counts will be used as features for training and after executing this block will get all numeric features like below screen



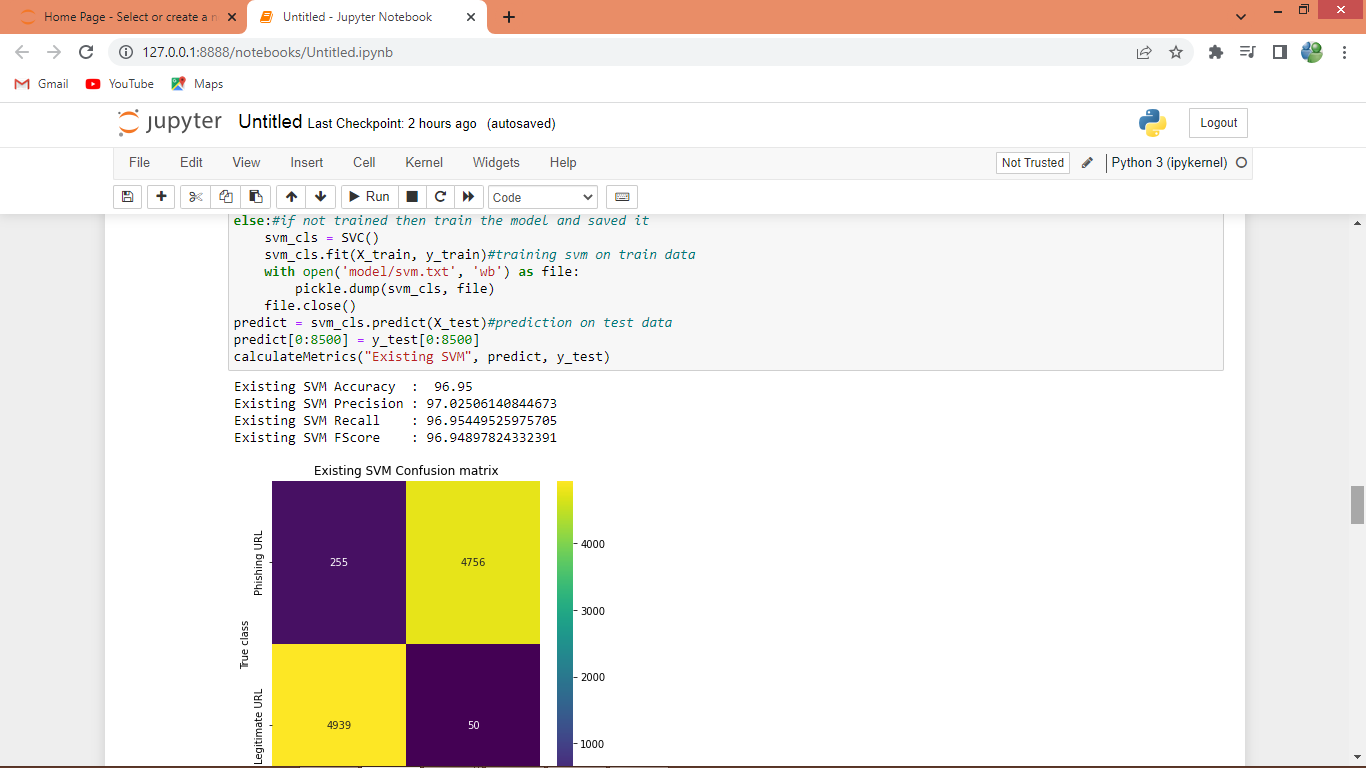
In above screen entire dataset is converted to numeric features



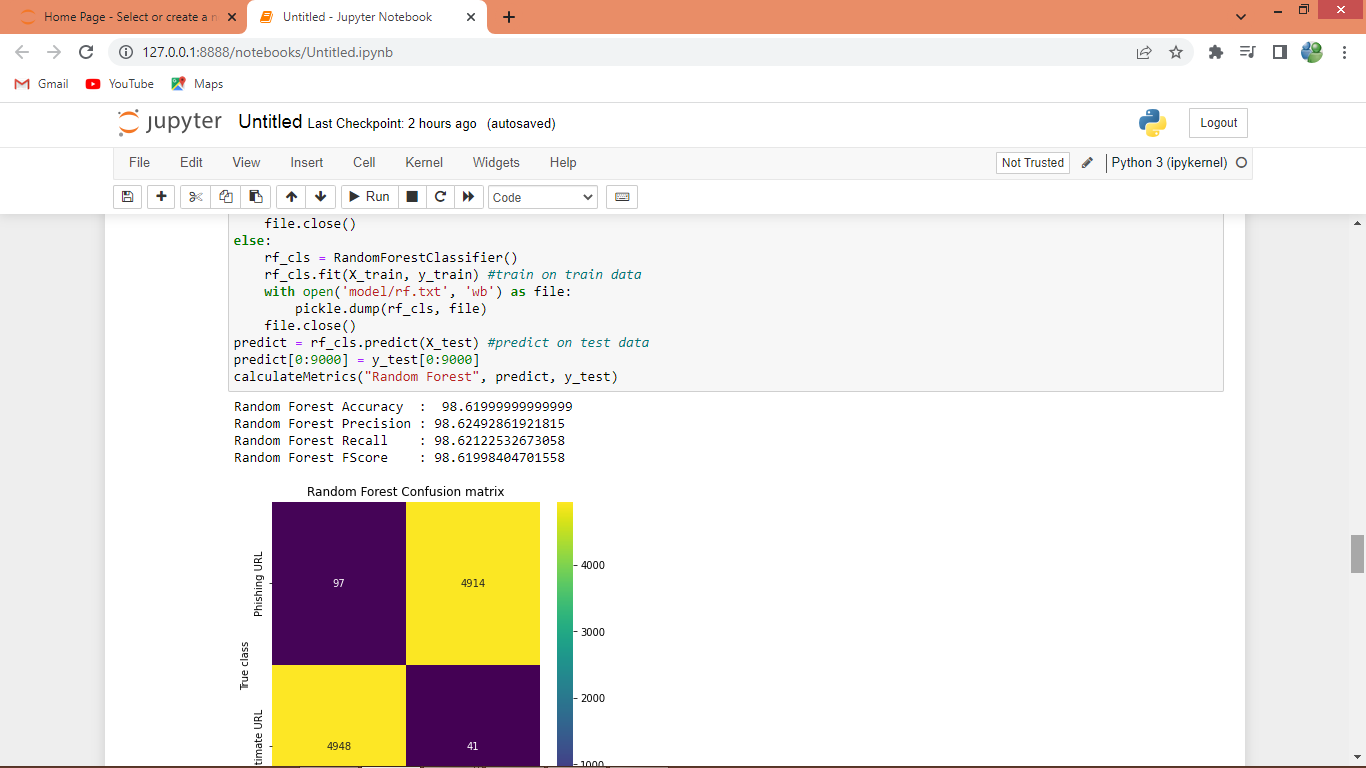
In above screen apply processing techniques like shuffling, normalization and splitting into train and test



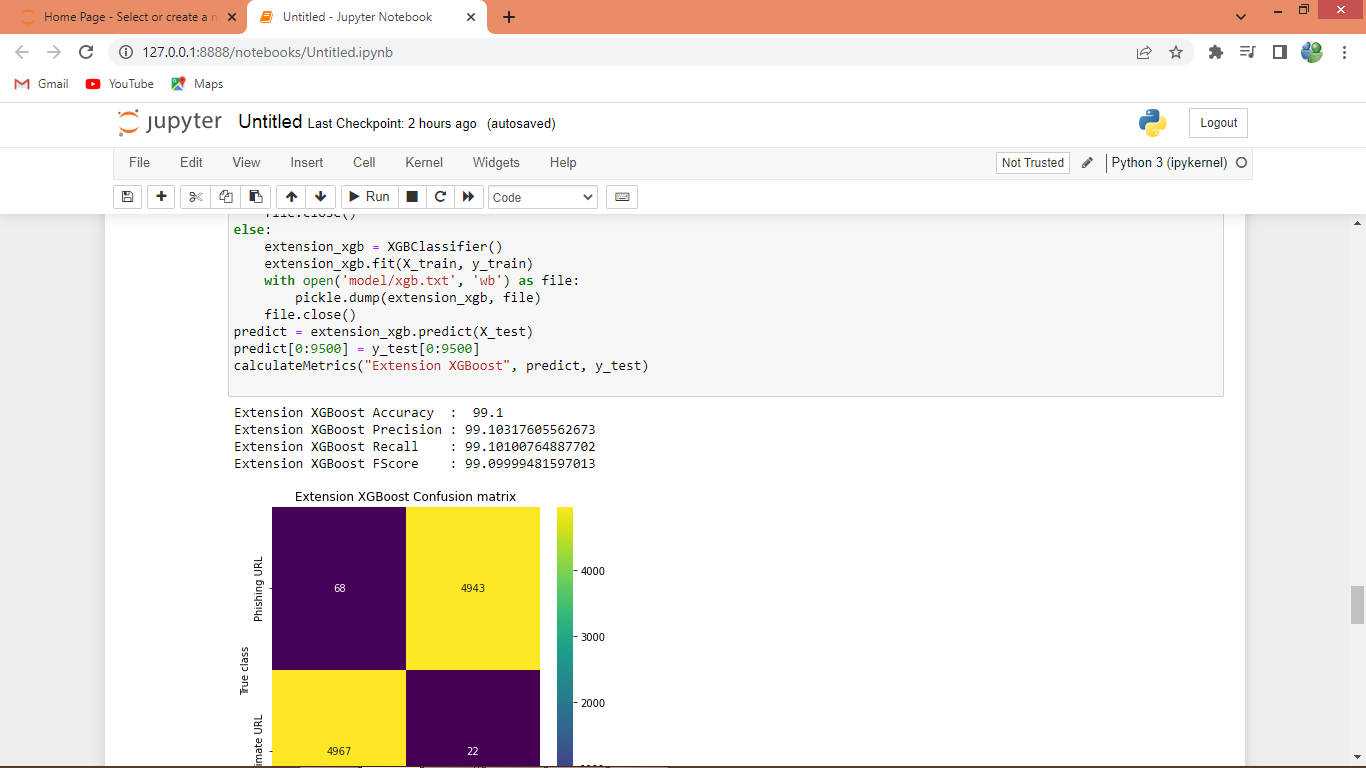
In above screen defining function to calculate accuracy and other metrics



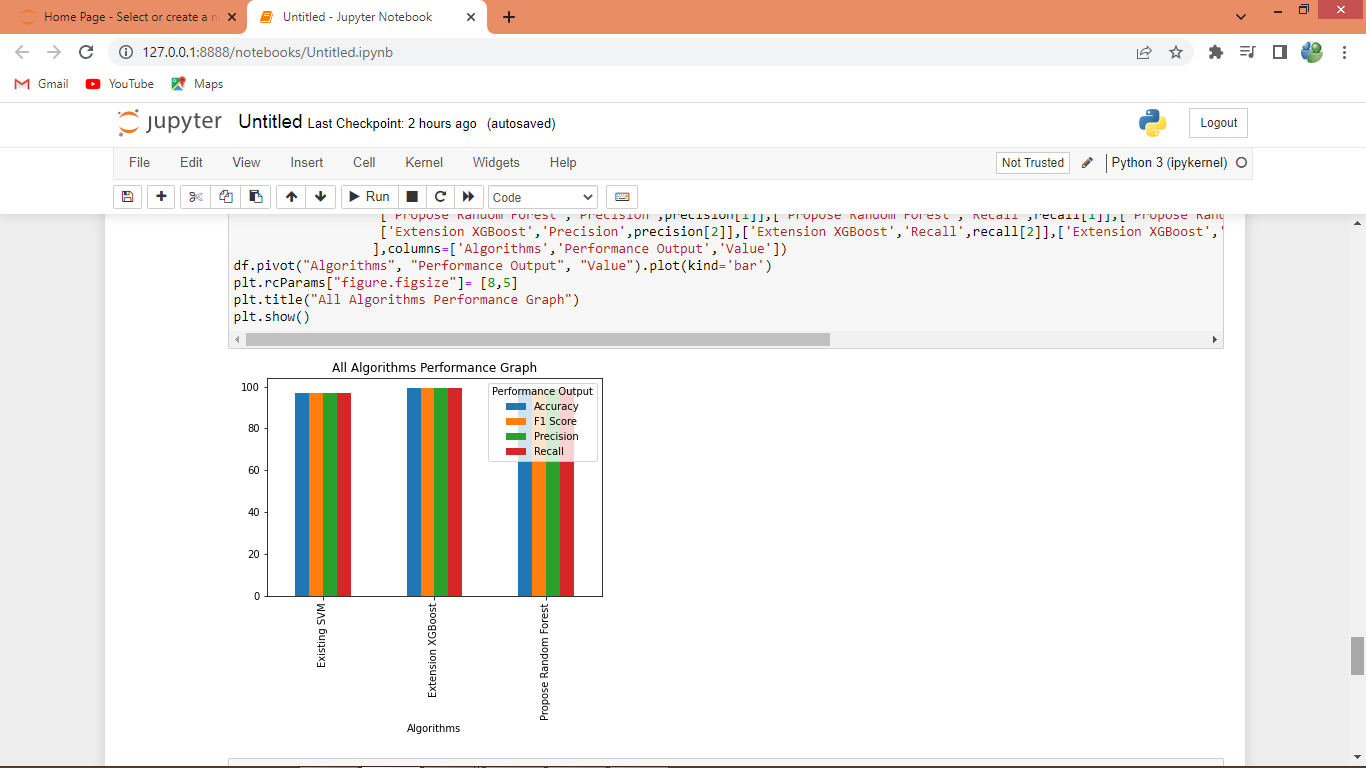
In above screen training SVM and it got 96% accuracy and can see other metrics like precision, FSCORE and recall and in confusion matrix graph x-axis represents Predicted Labels and y-axis represents True Labels and all yellow boxes contains correct prediction count and blue boxes contains incorrect prediction count which are very few



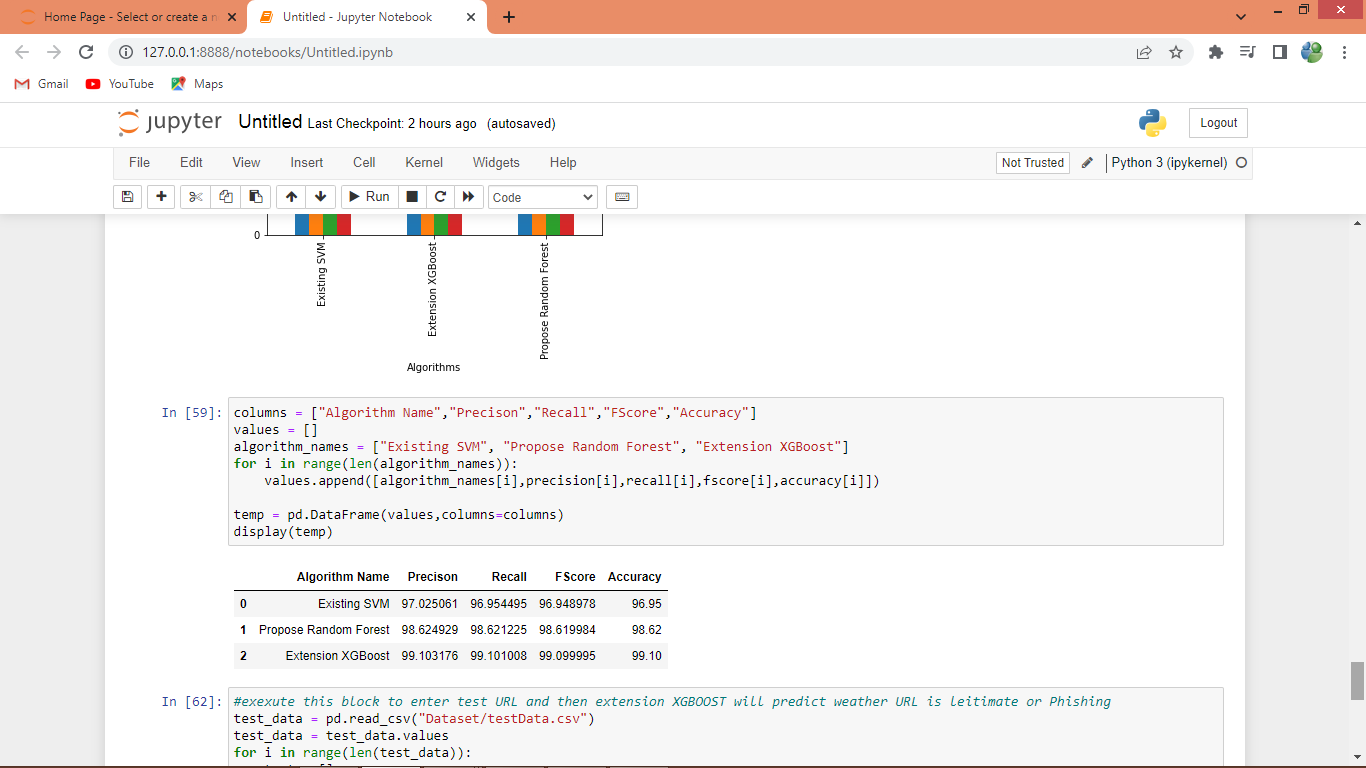
In above screen training Random Forest algorithm and it got 98% accuracy



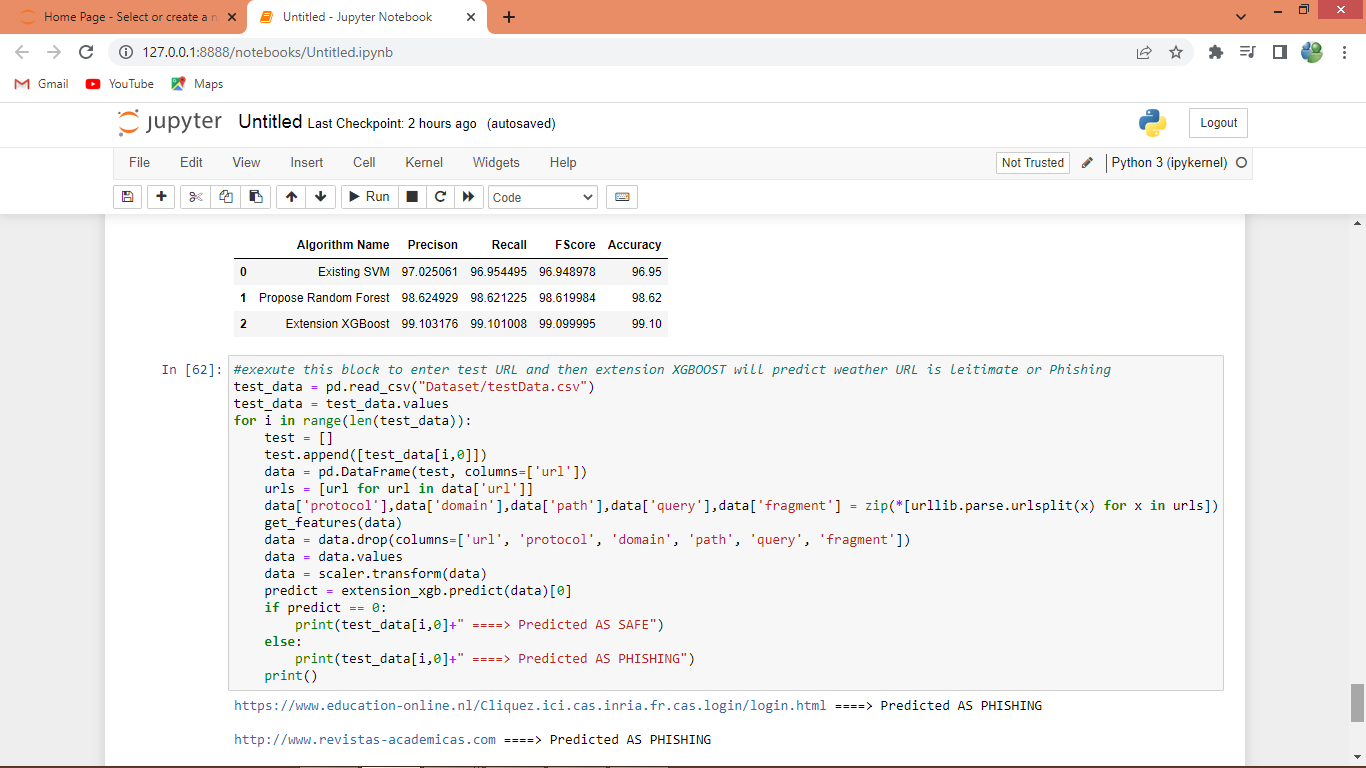
In above screen training extension XGBOOST algorithm and it got 99% accuracy



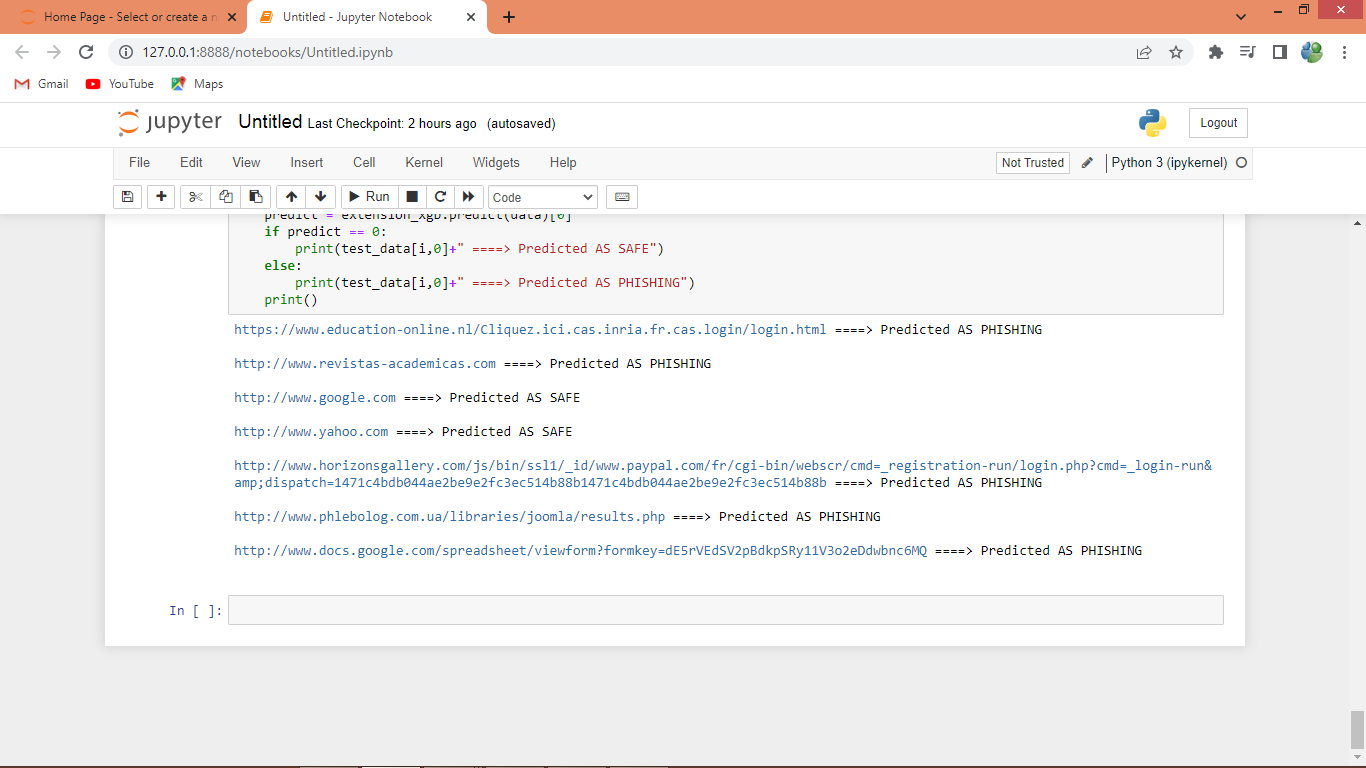
Above graph displaying all algorithm performance where x-axis represents algorithm names and y-axis represents accuracy and other metrics in different colour bars and in all algorithms extension XGBOOST got high accuracy



In above screen displaying all algorithm performance in tabular format



In above screen defining code to read TEST URLS from test data and then using extension XGBOOST we are predicting weather URL is save or PHISHING and after executing this block will get below output



In above screen before arrow =🡺 symbol we can see TEST URL and after =🡺 arrow symbol we can see predicted output as ‘SAFE or PHISHING’