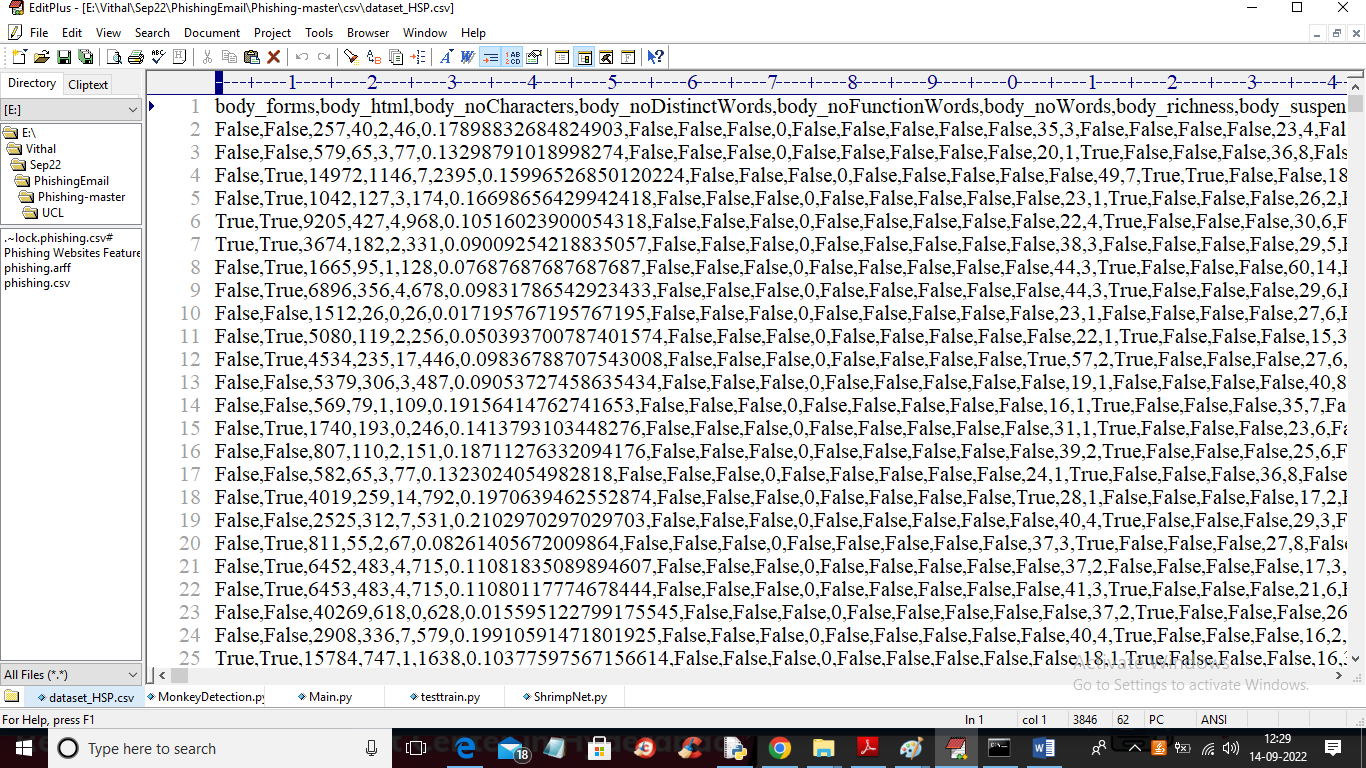
Efficient Spam and Phishing Emails Filtering Based on Deep Learning

In this paper author is using Deep learning algorithm called ANN (artificial neural network) to detect SPAM, HAM and Phishing emails by applying features selection algorithm called PCA (principal component analysis). All existing algorithms were detecting only SPAM and HAM emails but propose algorithm designed to detect 3 different classes called SPAM, HAM and Phishing.

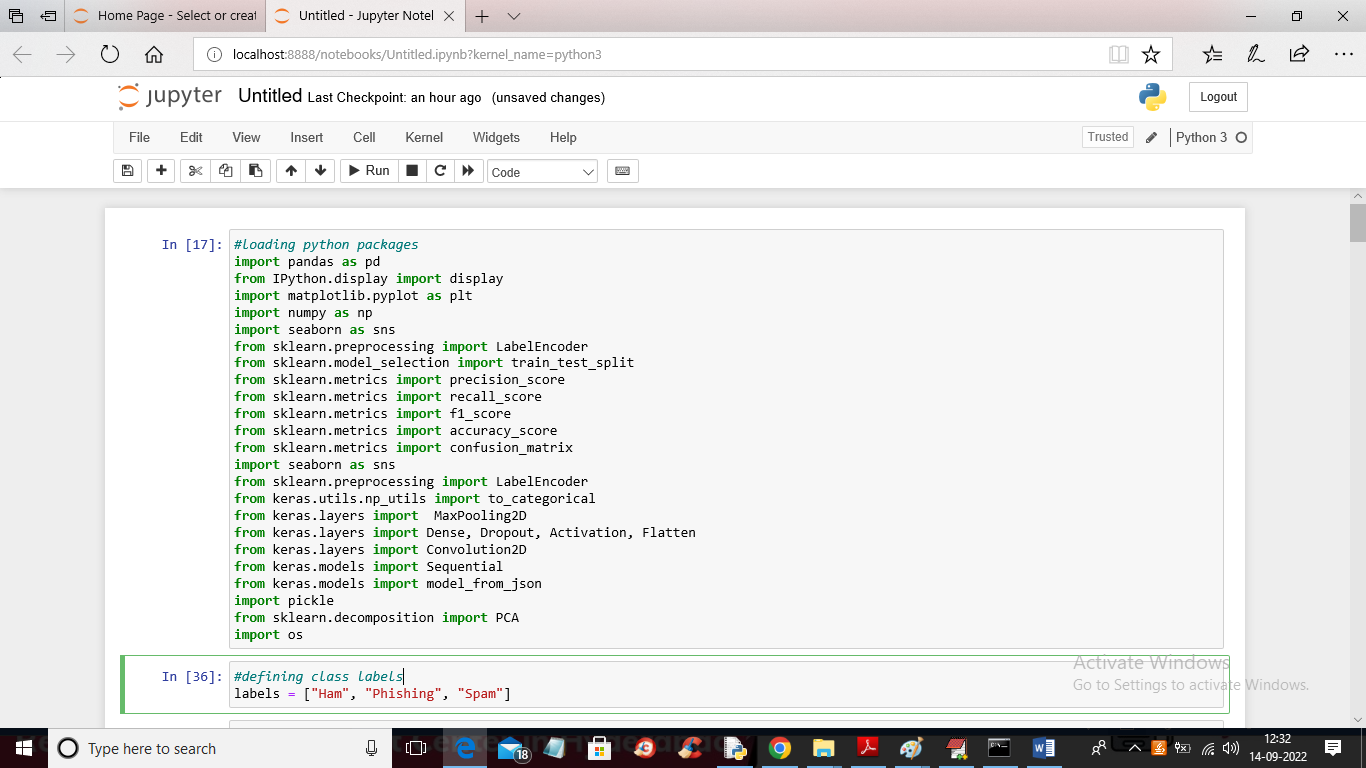
To implement this project we have combined 3 different datasets called UCI, CSDMC and SPAM ASSASSIN dataset. UCI and CSDMC datasets provided SPAM and HAM emails and Spam Assassin dataset provided Phishing emails. All this emails were processed to extract important features used in spam and phishing emails such as JAVA SCRIPTS, HTML tags and other alluring URLS to attract users.

Below screen showing dataset used in this project

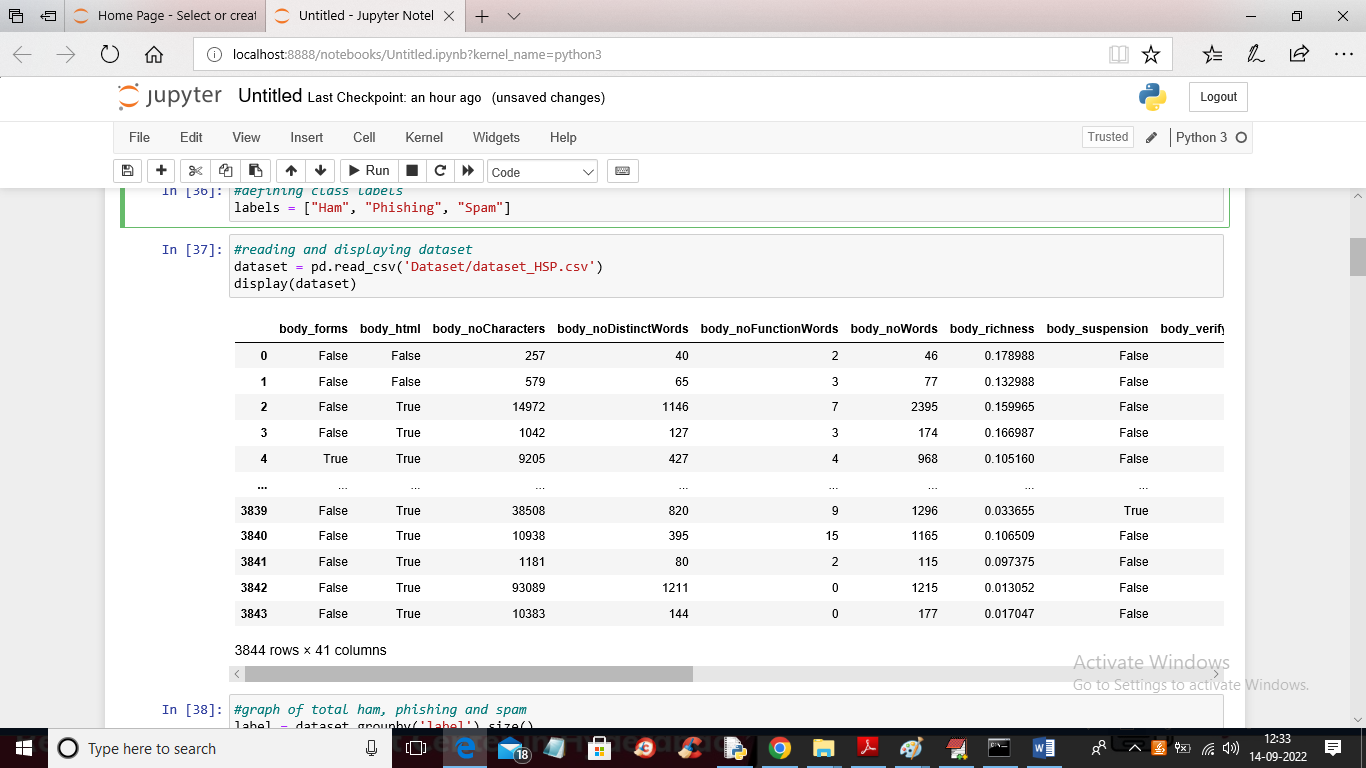


In above screen first row contains dataset column names and remaining rows contains dataset values and in each column values if HTML BODY or java scripts tags are present then TRUE will be applied otherwise false and all numeric values represents average occurrence of column name in EMAIL. For example if word ‘XYX’ occurs 3 times with total email length words 30 then average will be 3/30.

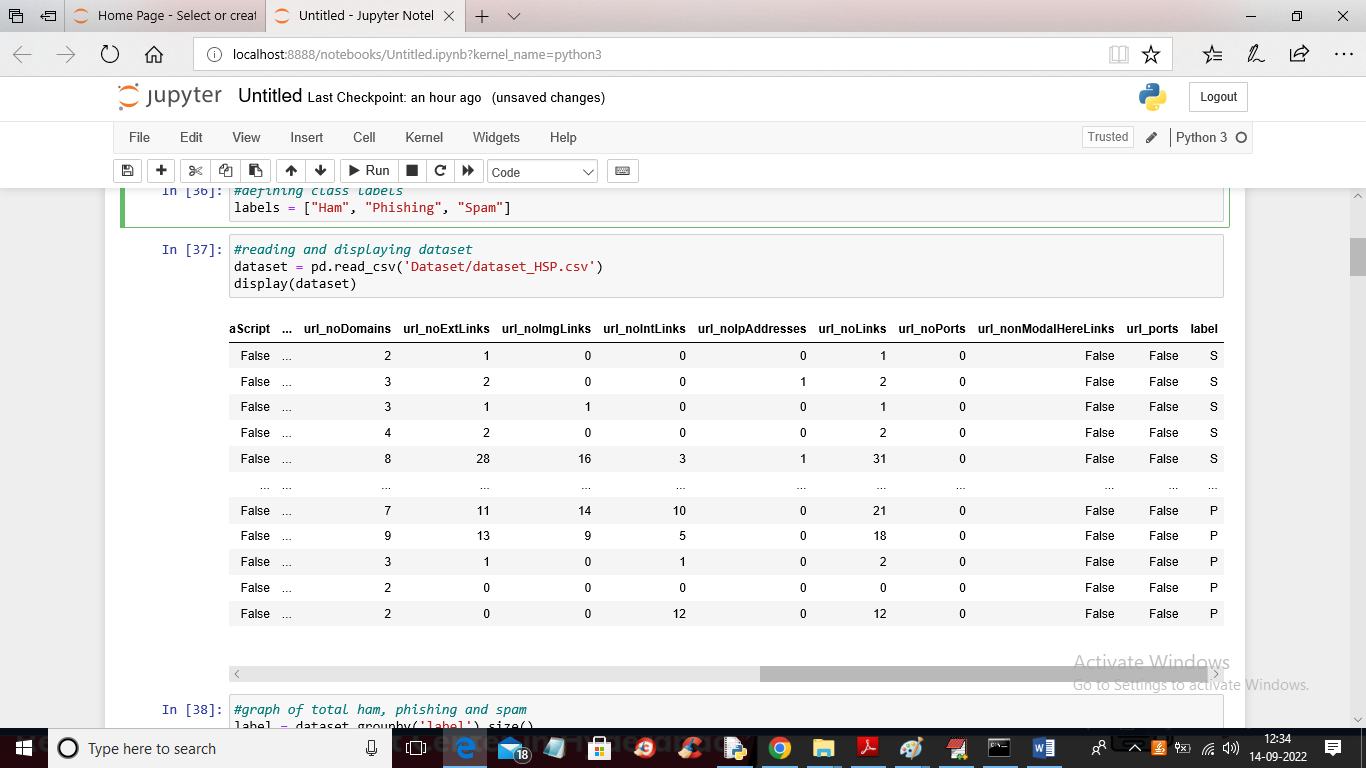
We have coded above project using PCA and deep learning algorithms with JUPYTER notebook and below are the code and output screens



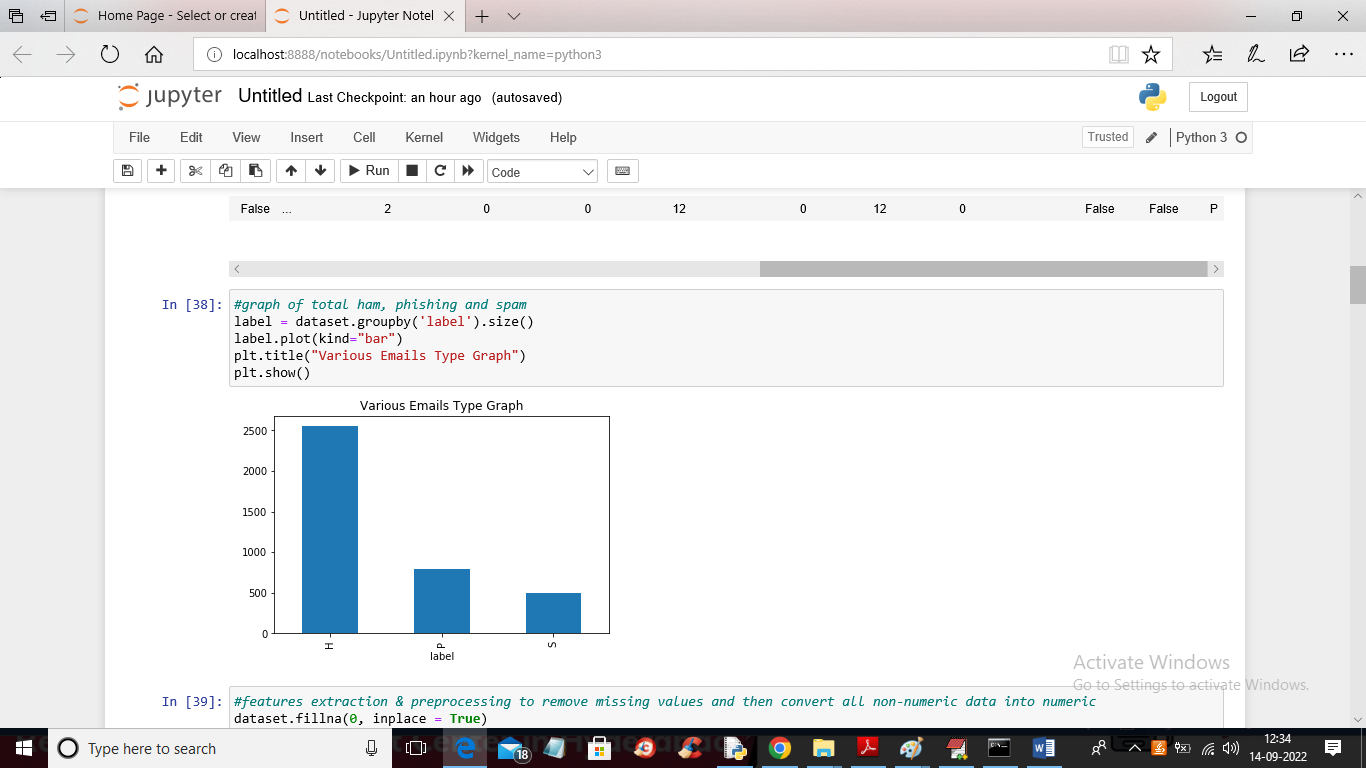
In above screen we are importing all require python packages



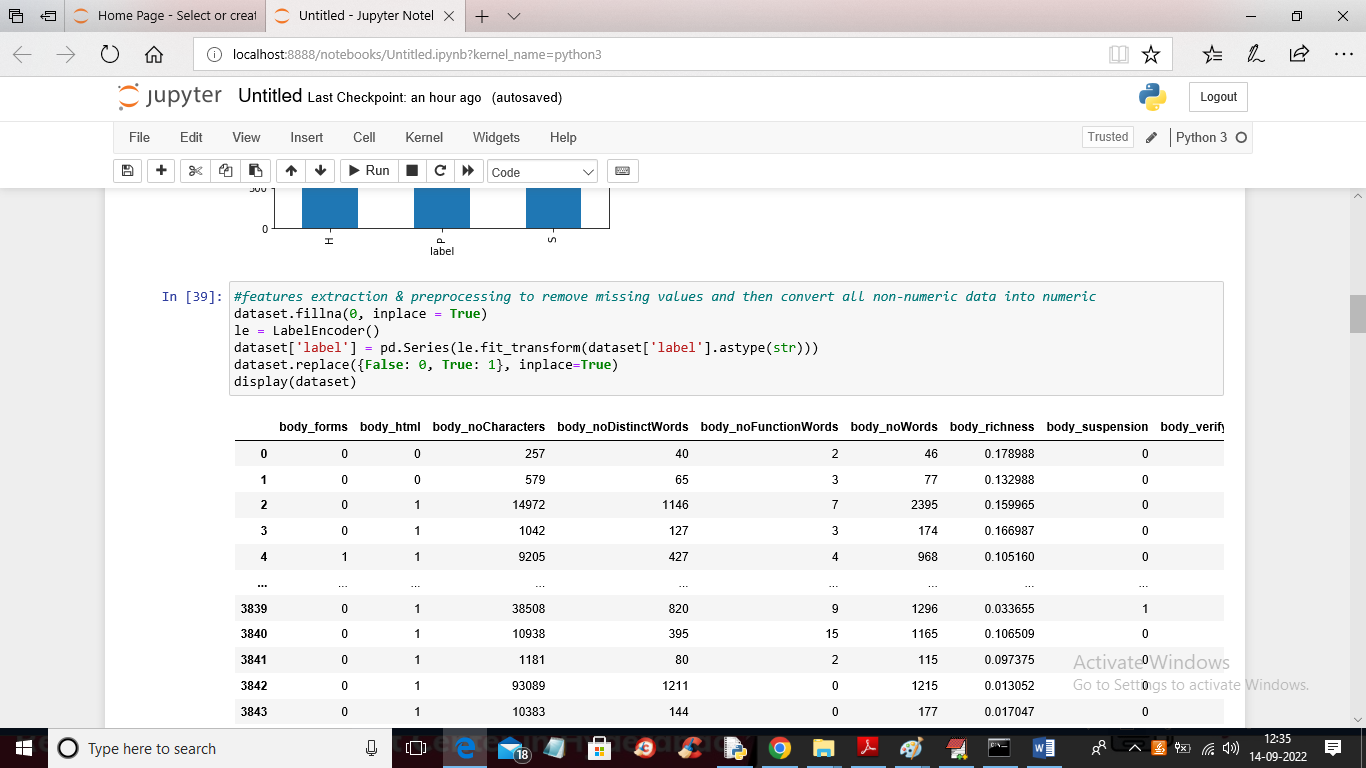
In above screen we are reading and displaying dataset values and last column we have class label as S or P or H where S represents SPAM and P represents PHISHING and H represents HAM and in below screen you can see those values



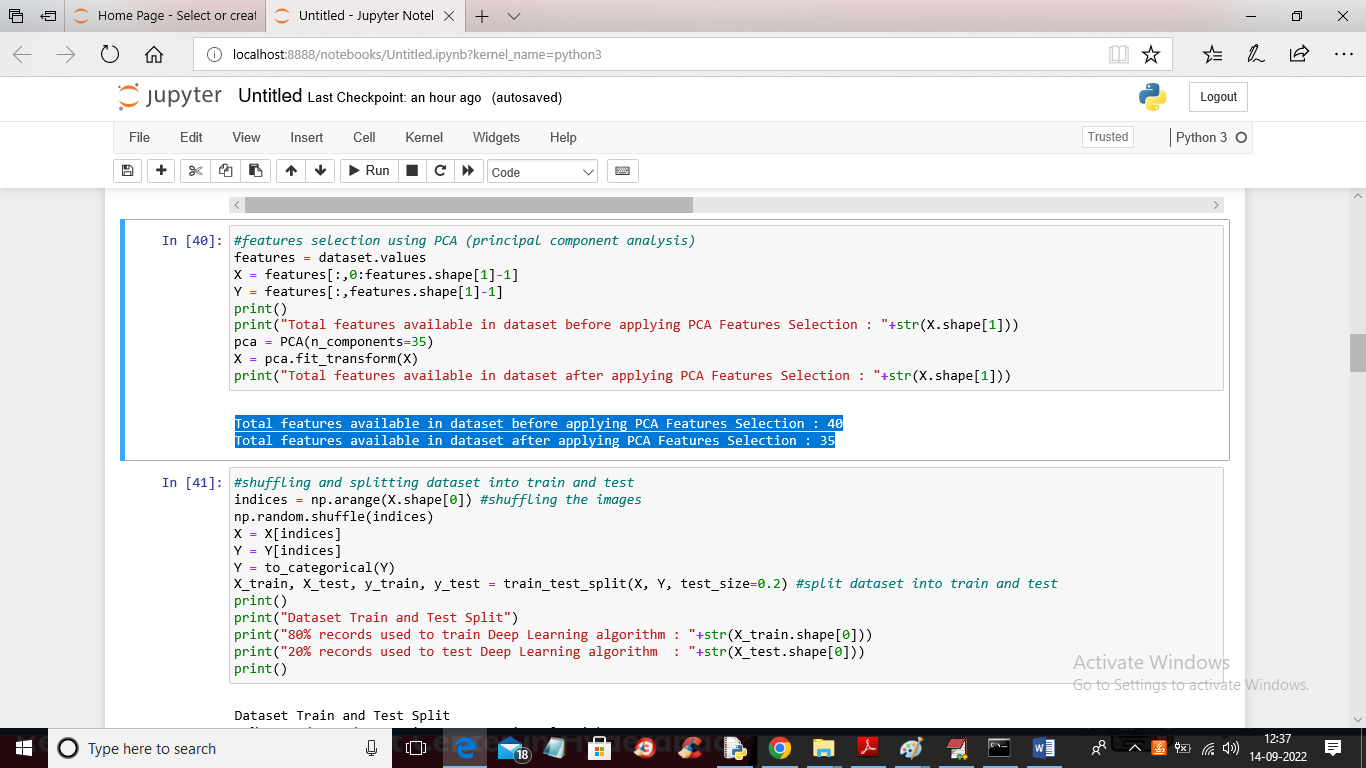
In above screen in last column we can see S, P or H values.



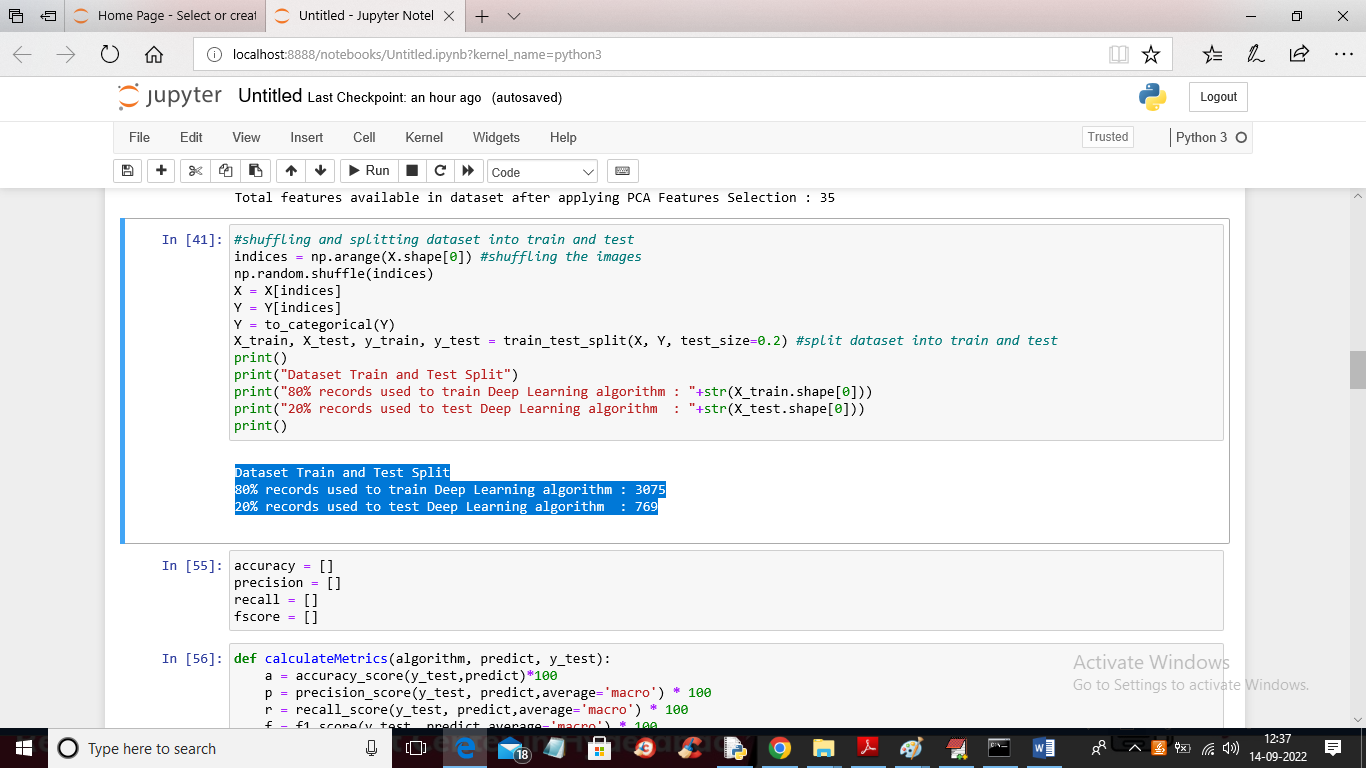
In above screen we are plotting graph with total number of SPAM, HAM and phishing emails. In above graph x-axis represents email type as SPAM, HAM or Phishing and y-axis represents count of those columns.



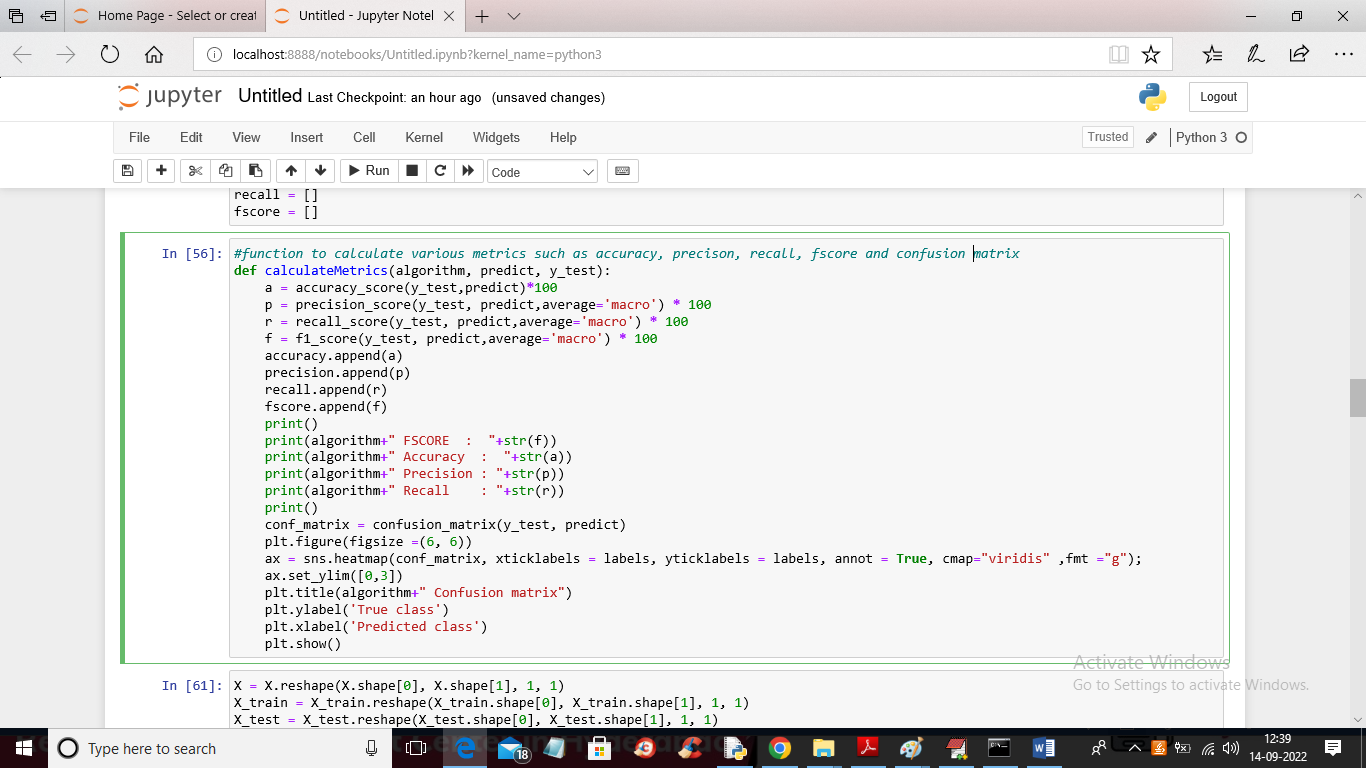
In above screen we are extracting features from dataset and then applying Preprocessing technique to replace missing values with 0 and then replace all non-numeric values with numeric as deep learning algorithm accept only numeric data.



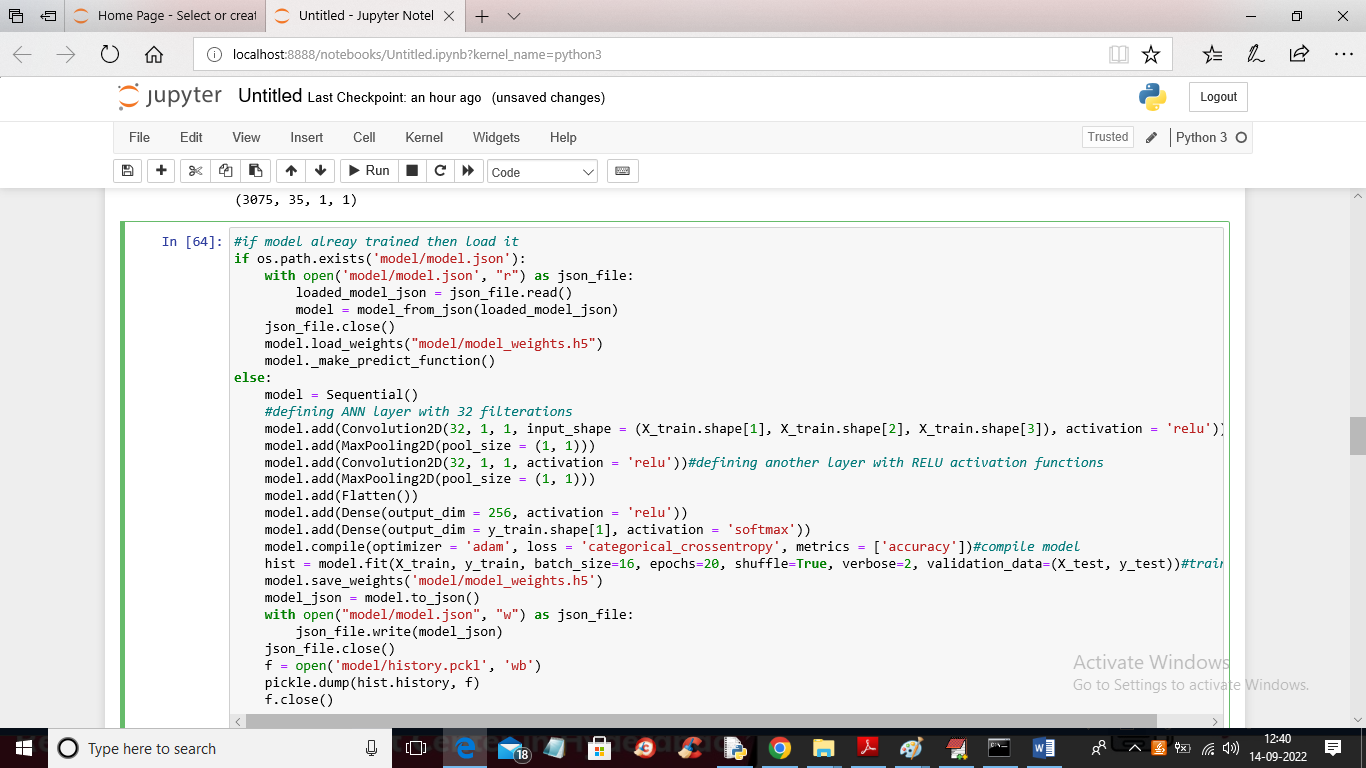
In above screen we are applying PCA features selection algorithm and then we are displaying total features count in blue colour text before and after applying PCA.



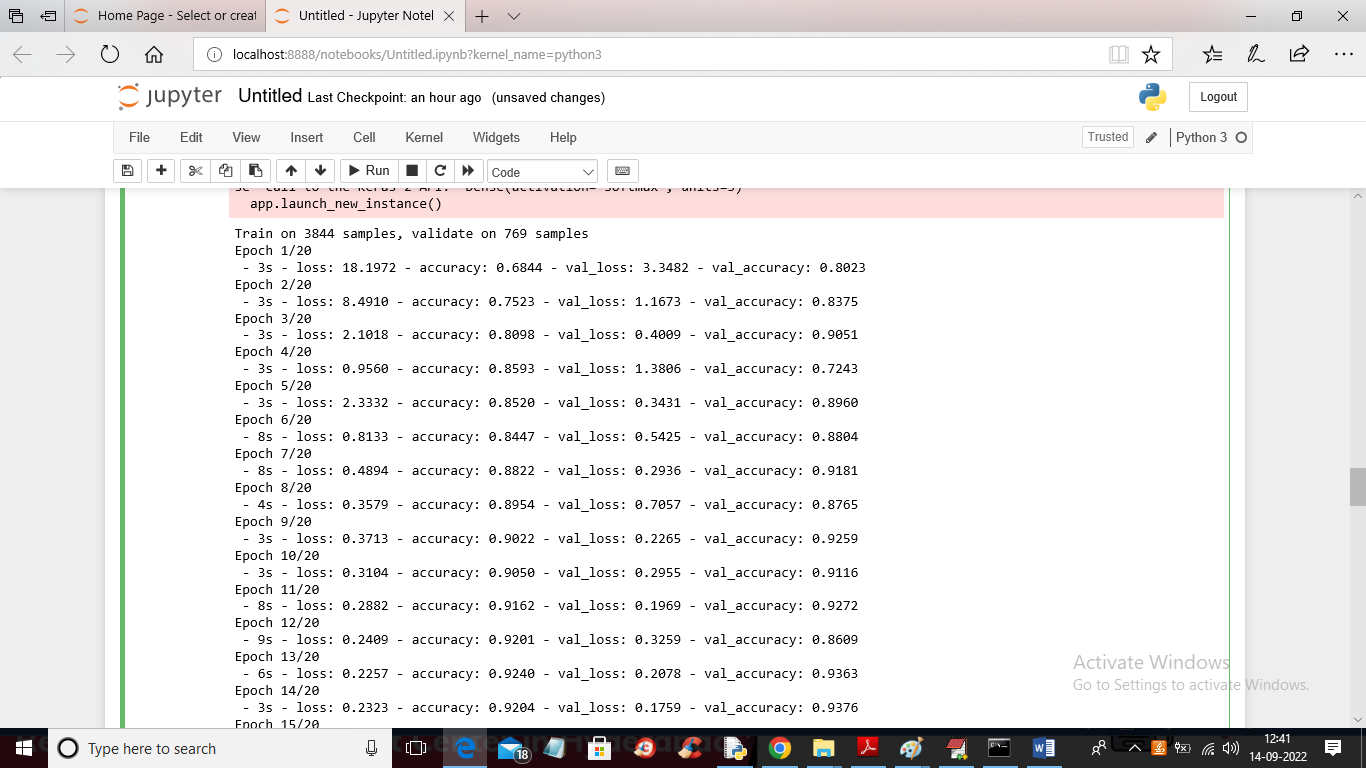
In above screen we are shuffling the dataset and then splitting dataset into train and test



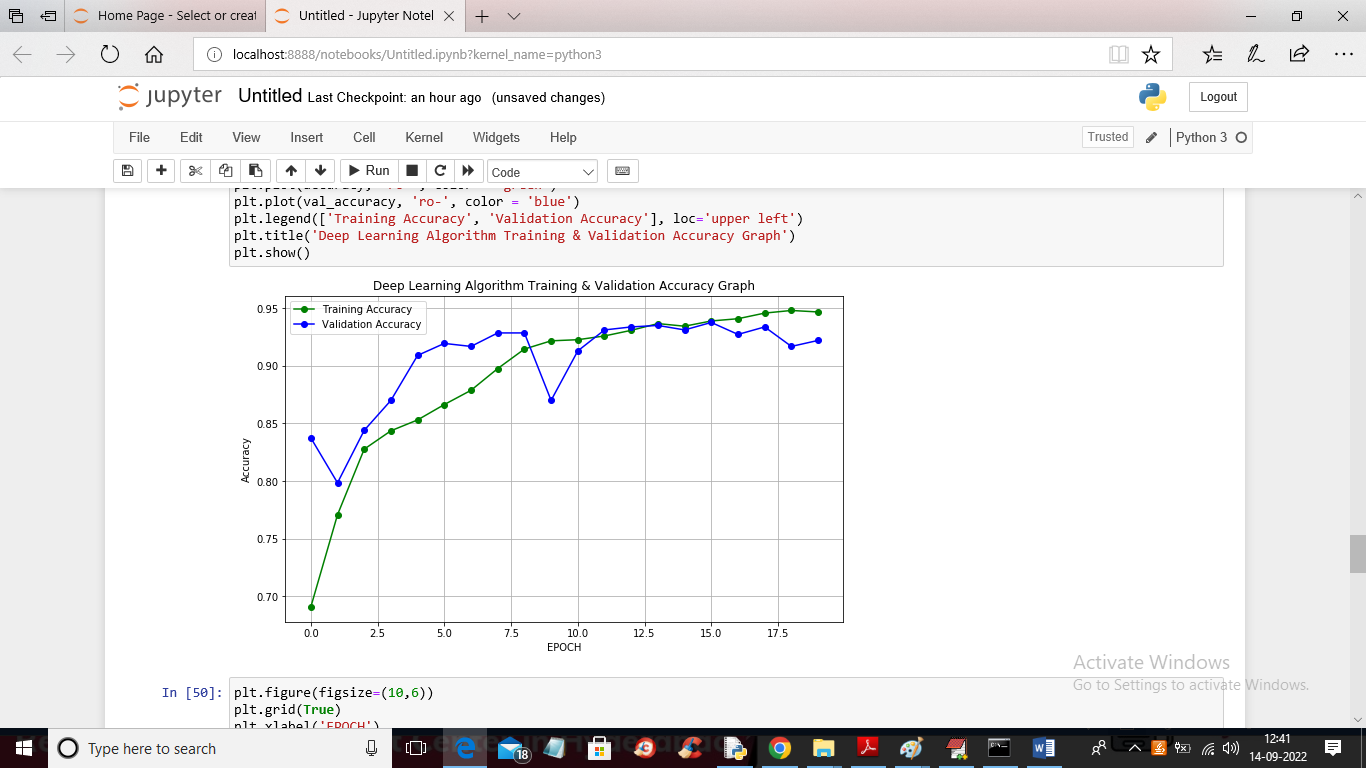
In above screen we have defined function to calculate accuracy, precision and other metric values



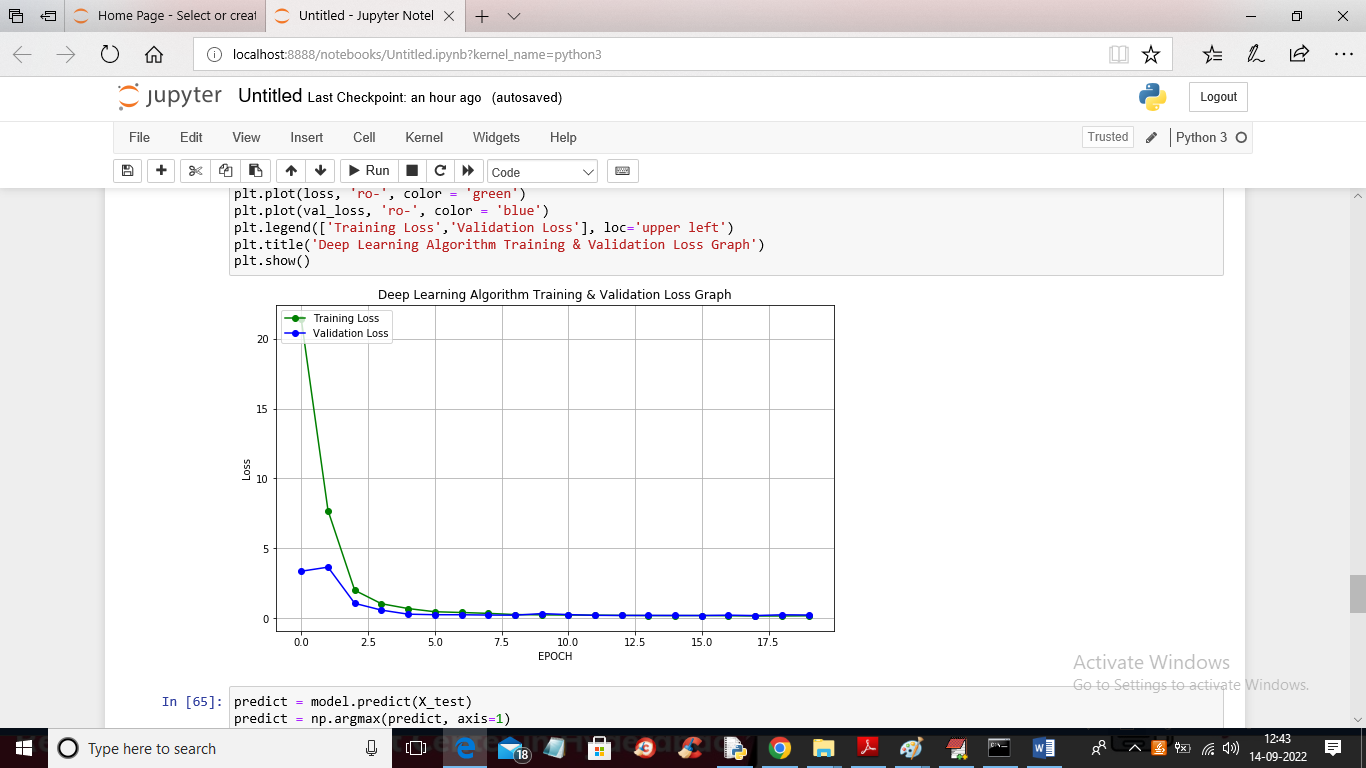
In above screen we are defining and training ANN model and after execution will get below output



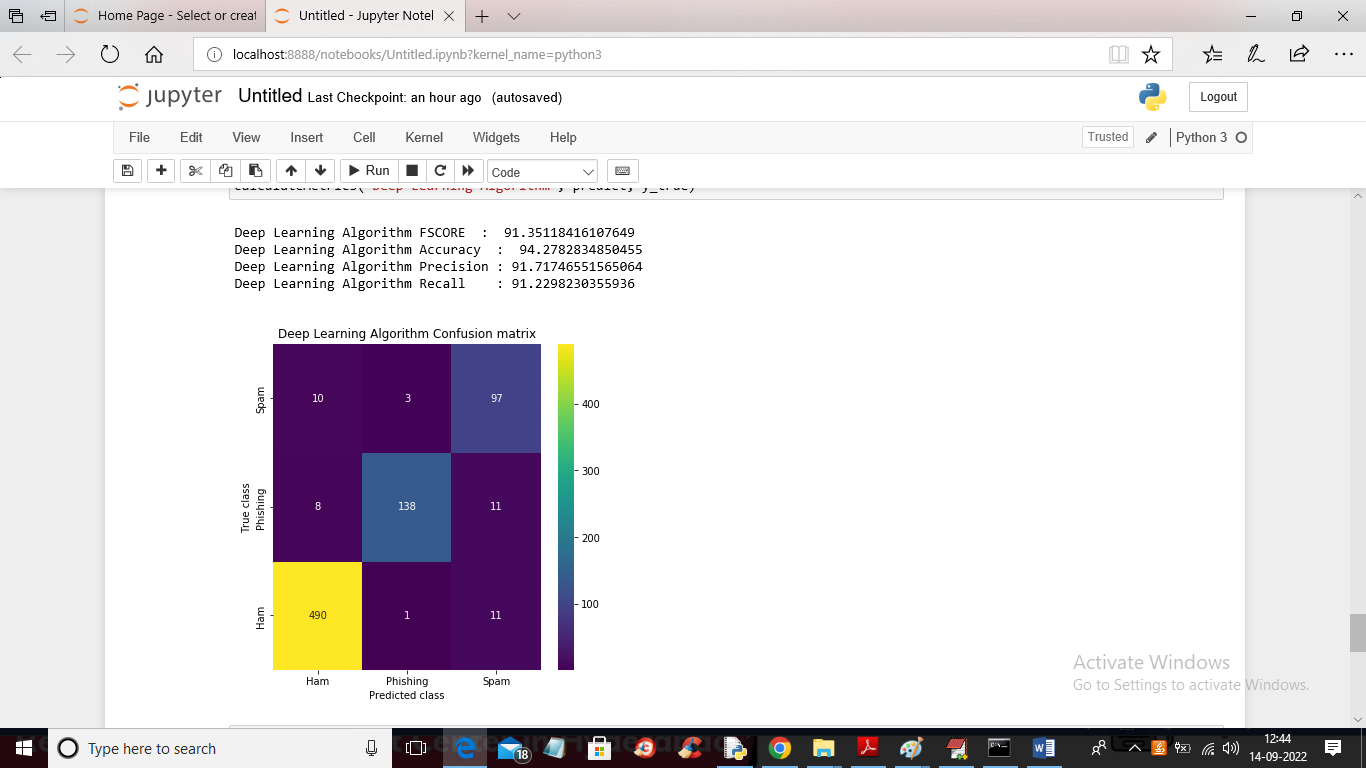
In above screen we can see ANN training started and for each EPOCH we can see training and validation accuracy



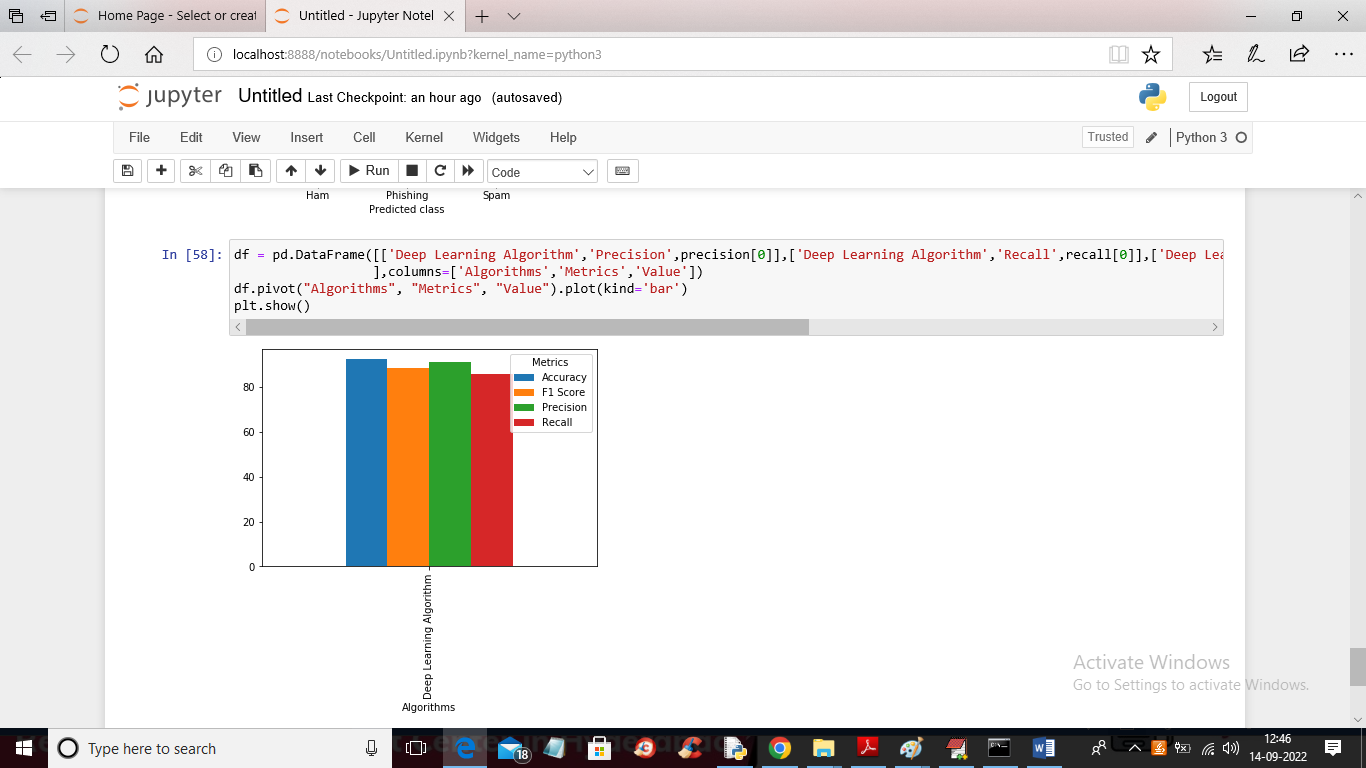
In above screen we are displaying training and validation accuracy graph where x-axis represents EPOCH and Y-axis represents ACCURACY and green line represents training accuracy and blue line represents Validation accuracy and we can see with each increasing epoch accuracy got increase



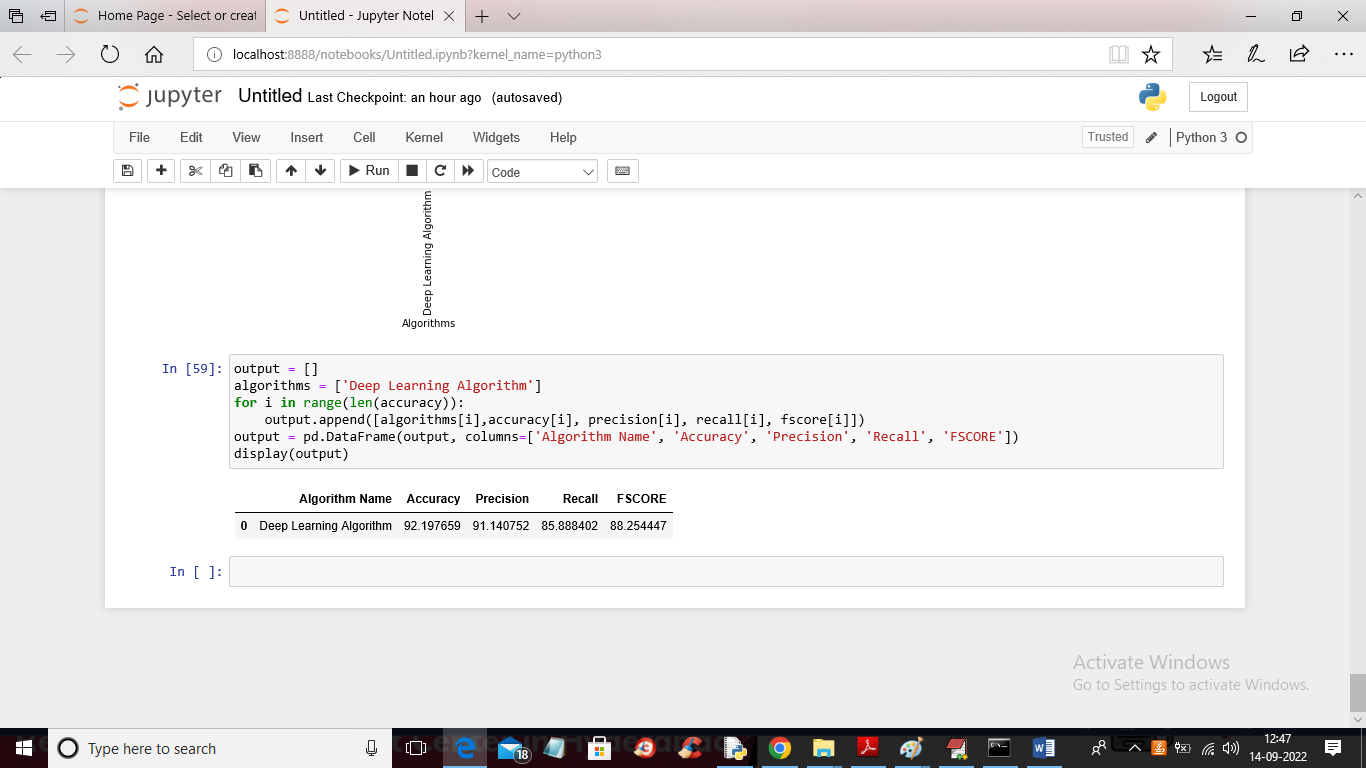
In above screen we are showing training and validation loss graph and with each increasing EPOCH loss got decrease



In above screen we calculated Deep learning accuracy, precision and other metrics and we got accuracy as 94% and in confusion matrix graph x-axis represents Predicted classes and y-axis represents TRUE classes and then COUNT in diagnol represents correct prediction count and count in other boxes are the incorrect prediction count and we can see very few incorrect prediction counts are there in other boxes so we can say deep learning algorithm performance is accurate.



In above graph different colour bar represents different metrics such as accuracy, precision and all metrics are more than 90%



In above screen we are displaying Deep learning algorithm values in TABULAR format