Name: Malay Dhami
Email: malaydhami99@gmail.com

Problem Statement

Given a dataset of phishing & normal emails, our task is to detect if a given email is a phishing email or not using a ML-led solution.

Dataset

- There are 2 folders: 'Normal' and 'Phishing'.
- 'Normal' contains all the legitimate emails. There are 2551 normal mails.
- 'Phishing' contains phishing emails. There are 2274 such emails.

Approach

- Our dataset has .eml files. To extract body part of emails I have used email parser. This parser can understand email document structure. We can get much information from this parser but I have extract only body part.

  Installation: pip install email

- To parse only text from body part I have used beautifulsoup library. This library can help us to get text from html document.

  Installation: pip install bs4

- Used some pre-processing techniques using nltk library. Main purpose of performing this step was to get most frequent words. I have performed lowering text, removal of punctuation, removal of stop words and lemmatization.

  Installation: pip install nltk

- By completing previous step, I got to know some of the features which might help us to predict our target variable. So, I have extracted these features from email's body part.

- I have performed previously mentioned steps on both of the folders ('normal' & 'phishing') to generate dataset for each folder.

- Merged both of the datasets for visualisation purpose and some pre-processing tasks. Then applied various machine learning algorithms to predict output class.

- Here are algorithms which I have used:

  1) Logistic Regression

  2) Support Vector Machine

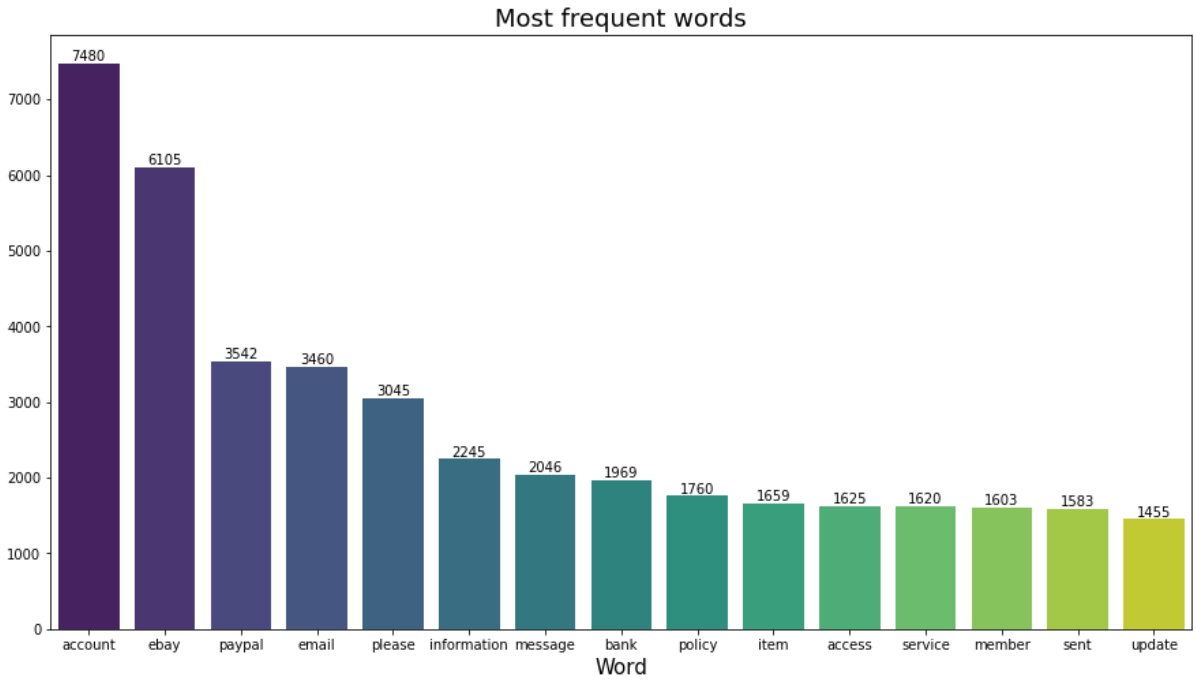
  3) Decision Tree Classifier

  4) Random Forest Classifier

  5) Bernoulli Naïve Bayes

- Split data into 80:20 ratio. Then used gridsearchcv to find the optimal hyperparameters of a model which results in the most accurate predictions. Then chose best model.

Extracted Features

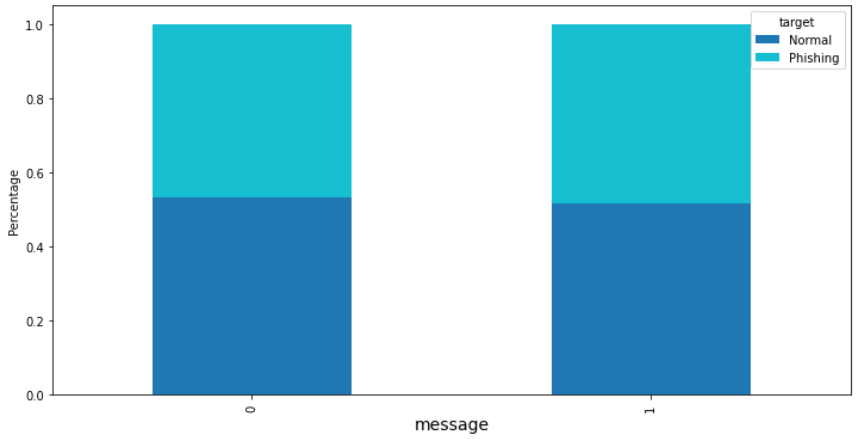


After looking at this plot I have selected 16 features for prediction:

1. HTML tag: Number of HTML tags in body part
2. Image tag: Checks how many image tags are present. Basically, it gives numbers of images in email
3. Dots count: Maximum number of dots in a URL of body part
4. URLs: Number of URLs present in email
5. Account: Word ‘account’ is present in body or not
6. Ebay: Word ‘ebay’ is present in body or not
7. Paypal: Word ‘paypal’ is present in body or not
8. Please: Word ‘please’ is present in body or not
9. Information: Word ‘information’ is present in body or not
10. Message: Word ‘message’ is present in body or not
11. Bank: Word ‘bank’ is present in body or not
12. Policy: Word ‘policy’ is present in body or not
13. Access: Word ‘access’ is present in body or not
14. Member: Word ‘member’ is present in body or not
15. Update: Word ‘update’ is present in body or not

Observations

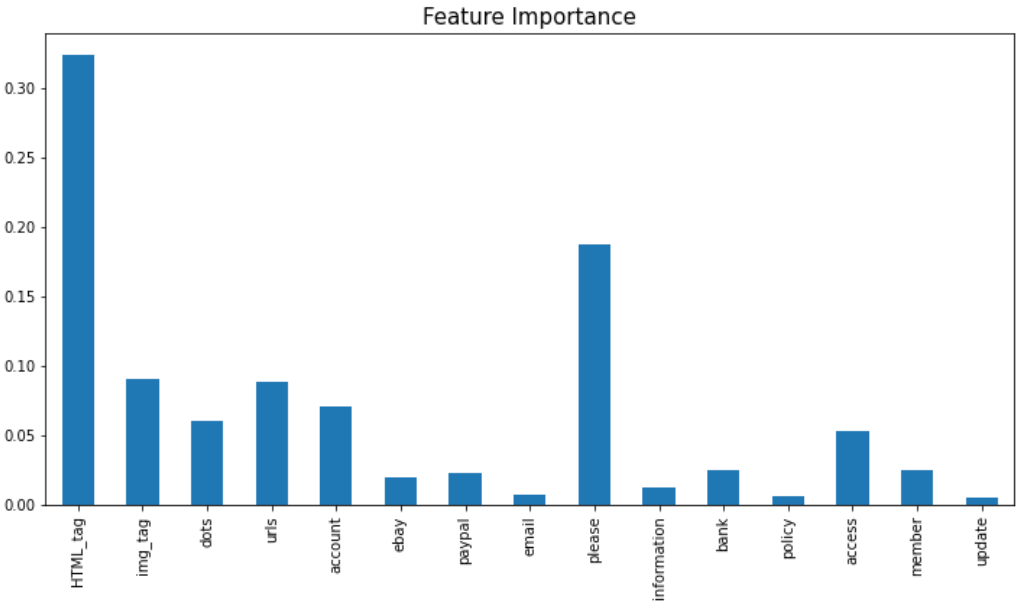
- As this plot shows ‘message’ feature doesn’t seem important. So I have just dropped this feature.



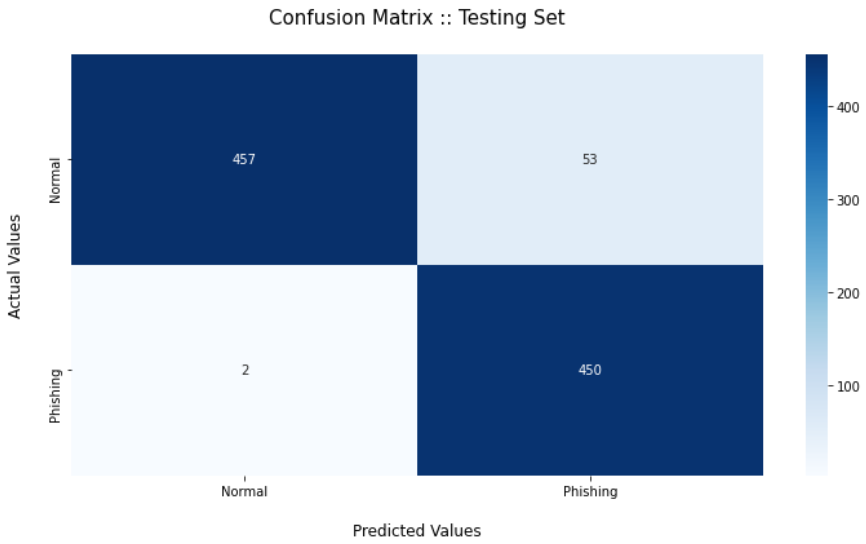
- Below table shows results of different algorithms:

Model	Best score	Training score	Testing score
Logistic Regression	0.9108	0.9111	0.9200
SVC	0.9350	0.9436	0.9241
Decision Tree	0.9439	0.9465	0.9356
Random Forest Classifier	0.9480	0.9514	0.9428
Bernoulli Naïve Bayes	0.9030	0.9041	0.9085

- Feature Importance:



- Here we can clearly see Radom Forest Classifier is giving us highest accuracy. Below is a confusion matrix of the best model.



- Indeed, our model is performing good. As for given problem it is important to predict phishing emails correctly. Out of all 452 phishing emails of testing data our model is predicting 450 records correctly only 2 records are misclassified.

- Here,
  Positive class = 'Phishing'
  Negative class = 'Normal'

  TN = 457
  TP = 450
  FN = 2
  FP = 53

  1. True Positive Rate: The percentage of phishing emails in the training data set that were correctly classified by the algorithm.

     TPR = 99.55

  2. True Negative Rate: The percentage of legitimate emails that were correctly classified as legitimate by the algorithm.

     TNR = 89.60

3. False Positive Rate: It is the percentage of legitimate emails that were incorrectly classified by the algorithm as phishing emails.

   FPR = 10.39

4. False Negative Rate: The number of phishing emails that were incorrectly classified as legitimate by the algorithm.

   FNR = 0.79

5. Precision: Measures the exactness of the classifier. i.e., what percentage of emails that the classifier labelled as phishing are actually phishing emails.

   Precision = TP / (TP + FP) = 89.46

6. Recall: Measures the completeness of the classifier results. i.e., what percentage of phishing emails did the classifier label as phishing.

   Recall = TP / (TP + FN) = 99.55