Phishing Email Detection

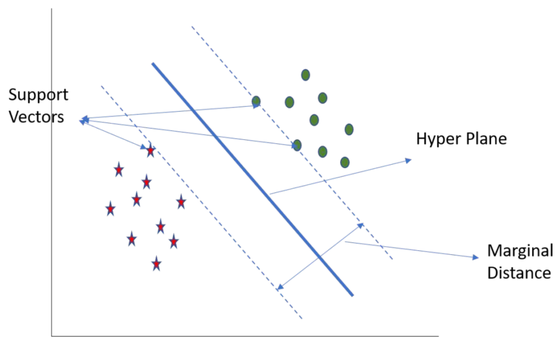
Pat Smurla
Department of Computer Science, Muhlenberg College
Guided by: Dr. Joseph Helsing

Introduction

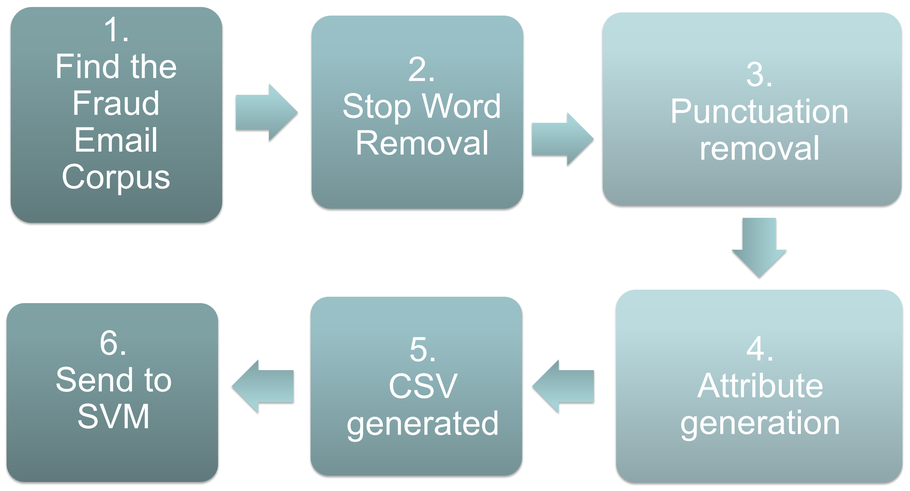
- In a study involving more than 50 million emails, one out of every 99 emails was classified as spam.
- Email spam is not the only way hackers try to steal your personal information…
  - Spear phishing - Targets a specific individual or company to steal credentials.
  - Pop-up phishing - Scams that will pop up on your screen as “advertisements”.
  - Vishing - Voice phishing. Receiving a call in order to try and have the victim say their credentials to the hacker.
- I believe this research is important because my contribution is seeing what attributes work better than others. Finding the ideal attributes is something that can make scam detection better.

Background

- SVM(Support Vector Machine)
  - Algorithm that has a group of supervised learning methods used to find classification, regression, and outliers.
  - Hyper Plane - Decision boundary that differentiates the two classes.
  - Decision Boundary - Point that is maximally far away from the data points
- Stop word implementation
  - Eliminating meaningless words
  - Python library, NLTK, has a list that includes a list of 40 stop words put together.
  - A, an, the, etc.
- Adding different attributes
  - Attributes - A characteristic of something, in this case of the emails.
  - Example:
    - A person’s attributes can be funny, nice, etc.
- Analyzing the outcome
  - How did the machine react to selected attributes?
  - Was it successful?



Methodology



1. Find a data set to use. Supplied by Kaggle, I found a large data set with only two attributes, the content of the email and the classification of it (phishing or not).

```
6    Not a surprising assessment from Embassy.,0
```

2. Next, stop words needed to be removed from the corpus. Using multiple loops to parse through every word in every email and it would remove meaningless words (a, an, the, etc.). Python has a library that included a list of all the stop words.
3. The same thing had to be done in order to remove the punctuation marks (Note: Unique IDs were added in this step as well).

```
6    4,Not surprising assessment ,0
```

4. In order to teach the machine all of this information, specific attributes needed to be generated. The 4 attributes used were the percent of the email that was misspelled, if the email contained a URL, the percent of the email that was fully uppercase, and the length of the email.

```
ID,Text,Class,Percent Misspelled,Has URL,Fully Uppercase,Total Words
6    4,Not surprising assessment ,0,0.0,0,0.0,6
```

5. With the class and all of the attributes put together, there is now a file that can be fed to the SVM algorithm.

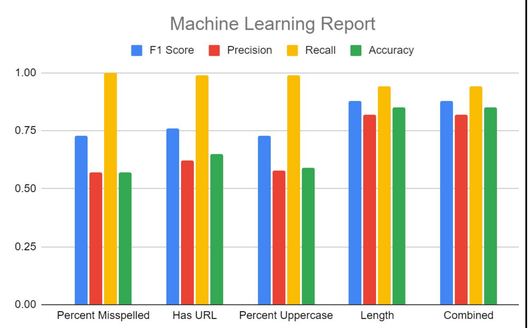
```
6    0,0.0,0,0.0,6
```

6. Final step, we feed this to the SVM algorithm.

```
=== RESTART: C:\Users\Pat\AppData\Local\Programs\Python\Python310\cleantxt.py ==
Files successfully cleaned.
>>>
==== RESTART: C:\Users\Pat\AppData\Local\Programs\Python\Python310\svmv2.py ====
Reading the dataset...
Dividing the dataset into testing and training sets...
Training started...
Training Completed.
```

Conclusions

The algorithm was very successful when using the selected attributes. To see what attributes were more impactful than others, I taught the algorithm each attribute individually and then once with all attributes together. Overall, the algorithm performed very well with all the attributes. The most successful attributes were the length, with an accuracy of 85%, followed by has URL with 65% accuracy. One thing that could make this algorithm perform even better would be adding more attributes into the CSV file. More specifically, more attributes that would make the algorithm think a particular email is not phishing. Most attributes selected were based on phishing. This is important for the field of ML because we can make email protection safer based on the attributes that are selected for the machine learning algorithm.



Confusion Matrix

	Positive	Negative
Positive	1268	81
Negative	271	766

References

1. Abbassi, Qandeel. “SVM from Scratch-Python.” *Medium*, Towards Data Science, 1 Apr. 2020, https://towardsdatascience.com/svm-implementation-from-scratch-python-2db2fc52e5c2.
2. Brad. “How Machine Learning Helps in Fighting Phishing Attacks.” *PhishProtection.com*, 6 Oct. 2021, https://www.phishprotection.com/blog/machine-learning-helps-fighting-phishing-attacks/.
3. Gatefy. “How Artificial Intelligence and Machine Learning Fight Phishing.” *Gatefy*, 22 Mar. 2021, https://gatefy.com/blog/how-ai-and-ml-fight-phishing/.
4. “R/Smitegodconcepts - Oshun, Goddess of Love.” *Reddit*, https://www.reddit.com/r/SMITEGODCONCEPTS/comments/svc4qe/oshun_goddess_of_love/.
5. “Removing Stop Words with NLTK in Python.” *GeeksforGeeks*, 31 May 2021, https://www.geeksforgeeks.org/removing-stop-words-nltk-python/.
6. Tatman, Rachael. “Fraudulent E-Mail Corpus.” *Kaggle*, 25 July 2017, https://www.kaggle.com/datasets/rtatman/fraudulent-email-corpus.
7. Chauhan, G. (2021, March 22). *Sklearn SVM - Starter guide*. Machine Learning HD. Retrieved May 6, 2022, from https://machinelearninghd.com/sklearn-svm-starter-guide/