Lucas Hyatt
Olivia Pannell
CIS 433 W20

Midterm Report - Spam Detection

Introduction:

Entering this project, our team knew we wanted to create something relevant to the modern security risks associated with social media. In many cases, these risks present themselves as malicious emails or text messages, among other forms. Although we would love to create a spam detection mechanism capable of inputting multiple data types (email, text, comments, … etc.), we are reserving that right and focusing on **email detection** for this project.

While some spam can be harmless, there is an overwhelming security risk associated to malicious attackers causing unwanted behavior through hidden links and misleading content. For these reasons, we are passionate about creating an accurate model with the ability to detect these security risks. Our objectives are to create a useful and functional graphical user interface in which the user can detect if a message is spam or ham. In this case, ham is a term used to represent valid emails. To accomplish this, we are planning to use a common machine learning algorithm called K-Nearest-Neighbor. In this approach, emails will be converted to vectors, where each vector will contain the frequency of each word within the email. These frequencies are the weights of our vector, and function as the features of our model. By representing each email as a vector, we can begin to compare the likelihood that an email is spam or ham depending on that vectors *k nearest neighbors*. From here, we are exploring normalization techniques which will enhance the accuracy of this model. Normalization will be included in the "data preparation" phase our timeline (this is happening currently). Our primary objective for this project centers around successfully implementing this training method.

Apart from delivering this model, we plan to continue development of our GUI interface to increase usability of our project. We have a vision of this project not only detecting spam but providing users with important information regarding their email content. The goal is to deliver a system where users can input an "inbox" of emails to not only clean their inbox but also receive critical information about their account. For obvious reasons, the GUI will be a big component of accomplishing this objective. Other deliverables include *instructions for use*, a *final report,* and a *poster* highlighting our project. The final report and poster will depict the usage and limitations of project, results pertaining to the model, and analysis of the final work. We will also have a group of spam emails and non-spam emails ready to be tested as a demonstration of the software. We do not plan to include the data we used to train the model (30 MB of text). Accomplishing these tasks will rely on our team's ability to follow the updated timeline.

Week 2, 3	Week 4	Week 5	Week 6, 7	Week 8	Week 9
Research how to build the model.	Collect spam and ham data.	Data preparation.	Training the model	Testing + Project Refinement	Deliver and Present Work

NOTE: We left time at the end to explore the idea of expanding this project to include different social mediums other than email. This will rely heavily on data availability, but with K-NN nearest neighbor we will have a model which can be used for text messages, Instagram comments, Facebook messages, … etc. The only thing we will need to do is prepare the data!

Work Relating to Our Project:

Considering that spam detection via machine learning is a relatively common topic, there is a large amount of resources online that describe the problem. More specifically, our team has found an abundance of related works that go in depth talking about Python libraries commonly used for machine learning. In many ways, this is was a strong incentive for choosing this research topic, but it also presents us with the challenge of choosing a unique way to approach this problem. For these reasons, we have focused on finding resources regarding the nearest neighbor approach to machine learning. Although this is a common method for creating a prediction model, we haven't found this implementation used very frequently for spam detection. In most cases, related work involves using common open-source Python machine learning libraries such as "scikit-learn". These libraries are often built on NumPy and SciPy, and although they are incredibly relevant to our project, our team feels the need to diversify the way we approach this challenge. Using nearest neighbor will also allow us to easily expand upon our work. Apart from this decision, one key aspect regarding our project is the wide availability of email data. We are interested in datasets that include both ham and spam emails. Formatting these emails is an integral part of our project, so our team has decided to explore common approaches to creating data from text, something which will hopefully make our feature extraction process relatively straight forward.

Here is a popular email dataset we plan on using
Here is a scholarly article on text feature extraction.

What We Have Accomplished So Far:

Preliminary Model
The development of our spam detection interface has taken us past a few major milestones so far. The first of which is training a preliminary model using the "scikit-learn" library. Although this feature will not be part of our final project presentation, it has served as a useful tool for testing datasets and interacting with the GUI. This allows us to continue development on the GUI while not having a working prototype, and it has given us a solid foundation upon which we can begin to build our nearest neighbor model.

GUI Prototype
We have a GUI prototype developed. This was developed in Python using the built-in tkinter library. As we mentioned in the project proposal, this GUI is one key feature we believe distinguishes this project from other spam detection solutions. We plan to use this GUI as a vehicle for displaying the results of our model in the form of useful information to the user.

Nearest Neighbor Python Solution
Considering the foundation of our project centers around the k-nearest neighbor machine learning technique, our team spent time developing a nearest neighbor implementation. This is meant to ensure we were familiar with the concept before we began applying it to our spam detection module. Moreover, we implemented multiple normalization techniques which will hopefully lead to a more accurate model.

Datasets
One of the most important parts of our project (and any machine learning project) is the data. Fortunately, we have downloaded thousands of spam and ham emails which we plan to use during the training phase of our project development plan (see timeline above). From here, we need to write our feature extraction method to prepare the model for training based on the extracted data in each email.

Most of our research thus far has gone towards finding data, building the preliminary model for testing, developing the GUI using Tkinter, and learning more about the nearest neighbor machine learning algorithm. One important lesson **we have learned so far** is that converting an email to data that can be used for training is actually a large percentage of the work for this project. This process includes preparing the text by removing stop words (and, or, the, of), in order to ensure we are passing useful data to the model. We have also explored the idea of lemmatization which involves grouping words like "change", "changes", and "changed" so that they are represented as the same. Tuning the data is an unexpected obstacle that we are exploring currently, mainly with regards to optimization (we are using thousands of emails). The primary feature for our model will be the frequency of each word in an email, so we need to make sure this portion of our project is correct and efficient.

Plan for the Rest of the Term:

Our team's plan for the rest of the term is to continue on pace with the timeline and stay on track with task management within the group. We plan to enhance our GUI to include more functionality for the user, as we feel that front-end is an important part of what we are trying to accomplish with this project. Similar to the aforementioned, if we hit all of our intended goals by Week 9, we would love to expand from email to other social media spam such as Facebook or Instagram assuming we can find accurate datasets. We have left room in our timeline to expand this project, assuming all goes well with the preliminary model!