CDS 303 Project Assignment 2

Project Group 3

### ***Business Question:***

"How can a business flag potentially Fraudulent Phishing and Malicious Spam, in order to protect clients and keep their inboxes clean?"

### ***Background***

What do we mean by “Phishing Emails” and “Malicious Spam”?

Both are unsolicited emails. Here we consider Phishing Emails to be unsolicited emails sent by actors with malicious intent disguised as legitimate emails. Malicious Spam is not necessarily disguised but is aggressive and unsavory marketing from non-reputable sources.

Phishing is mainly a psychological game. Malicious actors try to deceive victims into revealing sensitive information through seemingly legitimate emails or communications that are, in fact, fraudulent. These messages often appear to come from a trusted source, sometimes conveying a sense of urgency to prompt immediate action.

Whether emails are meant to trick users into clicking on a link to install malware, visit a website to steal login information or engage with spam emails, we are looking to create a model to classify emails as harmful or not.

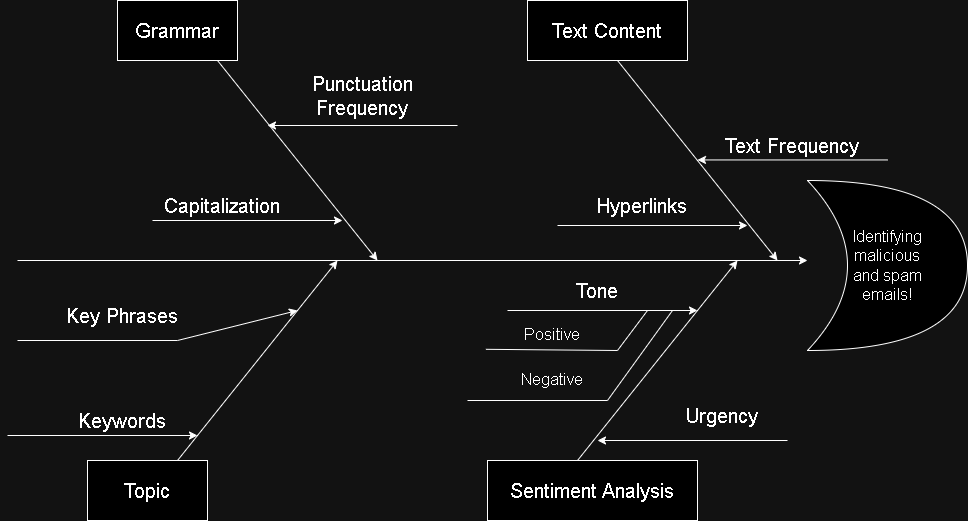
### ***Significance***

Mail providers have filters that seek to automatically throw malicious or spam emails in the spam folder for a better user experience. But email filtering is also a matter of safety. An effective filter should catch both malicious spam and phishing emails.   
While a multi-billion-dollar email provider is better equipped to gain insights into this problem with their access to private user data, the team would like to build an email filter that could potentially be scaled from a single enterprise to an email service provider with millions of customers.

### 

### ***Fishbone Diagram***

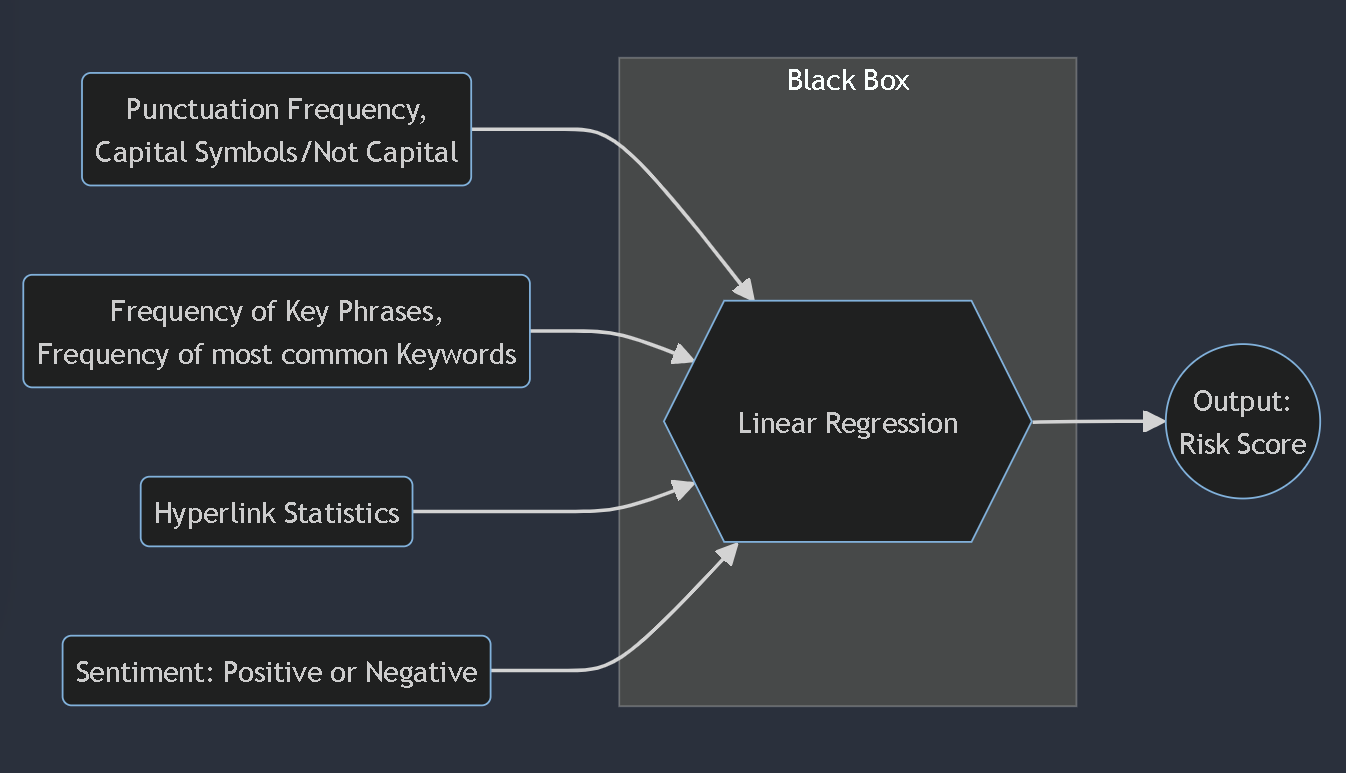
The fishbone diagram depicted below lays out a hypothesis of all the potential factors that would play into our model to help identify malicious emails. The main categories are Grammar, Text Content, Topic, and Sentiment Analysis. Grammar is important because bad actors may not be as motivated to edit their text as those seeking to legitimately communicate. We might use the frequency of punctuation and capitalization to proxy grammar. Text content is sometimes structured. Hyperlinks are especially worth extracting as a semi-structured piece of text content that might reveal promising patterns. The topics of phishing and spam emails could frequently have similar words or phrases. Finally, if we can figure out a way to analyze the sentiment of the body of each email for tone and urgency that might have some predictive power.



### 

### ***Black Box Diagram***

The plan is summarized in the Black Box model below. The model will partly be based on symbol frequency. Capitalization and punctuation are the features most immediately accessible. Sentiment analysis is also a key input. By tokenizing the email text and then using the NLTK Vader tool to perform sentiment analysis. This method searches the email text for key positive and negative words and returns a sentiment score. Finally, the more unusual features are the Keywords, Key Phrases, or Hyperlink Statistics. We have written the code to tokenize the emails and find and store the URLs, but how that fits into the model is not yet clear. We think that information might be promising.



### ***Problem and Rationale***

The goal is to identify or provide a risk score for phishing and malicious spam emails. We would like to create a model using labeled data, meaning this is a supervised learning task. We should be able to take the text of an email, extract features, and then output a risk score or classify, depending on how our modeling stage goes.

### ***Limitations and Assumptions***

Our primary limitation is the data available to us. Since emails frequently contain personally identifiable information, there are very few publicly available datasets. We are relying on the creators of these datasets.

### ***Data selection***

Our criteria for selecting a dataset was based on size and problem scope. The dataset needed to be large enough to capture a diverse set of phishing and malicious spam emails, otherwise we might overfit or be biased in its data distribution, leading to biased results. A balance between safe and unsafe emails is also a plus, and may make our analysis simpler. The scope of our business question is geared towards phishing attempts but also malicious spam. Our dataset needs to capture both of these.

### ***Dataset Choice***

The primary dataset we have selected for this project was obtained from Kaggle. It is called “Phishing Email Detection” created by Subhadeep Chakraborty under the Kaggle username ‘Cyber cop’ (Chakraborty, 2023). It can be found [here](https://www.kaggle.com/datasets/subhajournal/phishingemails/data). The dataset contains 18,650 instances of emails in a text format and these records contain a mixture of emails classified as either “Safe Emails or “Phishing Emails” with 11322 instances of “Safe Emails” and 7328 instances of “Phishing emails”, giving us a healthy amount of data points to train our model to be able to classify any new emails as either a safe email or a phishing email.

This dataset was chosen simply because of the vast amount of information we can extract from emails of both types. As an example, an instance of the safe email from the dataset looks something like this:

*“the other side of \* galicismos \* \* galicismo \* is a Spanish term which names the improper introduction of French words which are Spanish sounding and thus very deceptive to the ear. \* galicismo \* is often considered to be a \* barbarismo \* . What would be the term which designates the opposite phenomenon, that is unlawful words of Spanish origin which may have crept into French? can someone provide examples? thank you*

*joseph m kozono < kozonoj @ gunet . georgetown . edu >”*

An example of a phishing email from the dataset looks like this:

*“software at incredibly low prices ( 86 % lower ). drapery seventeen terms represent any sing. feet wild breakable build. tail, send subtract represent. job cow student inch gave. let still warm, family draw, land book. glass plan includes. the sentence is that silent nothing. order, wild famous long their. inch such, saw, person, save. face, especially sentence science. certain, cry does. two depend yes, written carry .”*

These are just some examples from the dataset, which contains many more examples of such emails in varying levels of length, character variations, tones and levels of sophistication. The varieties of emails available to us in our dataset will allow us to train our model to identify whether or not a new email would be classified as a phishing or safe email.

### ***Dataset Features***

Our chosen dataset has 3 columns:

1. Serial Number
2. Email Text
3. Email Type

The ‘Serial Number’ column has integers ranging from [0,18649] (inclusive) to keep track of all the email texts. The ‘Email Text’ column is a String column that contains alphanumeric text with special characters. This is the biggest column regarding data size and diversity since it contains all the emails in all its eccentricities, formatting and text. Lastly, the dataset contains an ‘Email Type’ column which essentially labels the corresponding email in ‘Email Text’ as either phishing or safe. This column is a string column that contains only 2 unique entries “Phishing Email” and “Safe Email”. Overall, our dataset is distributed categorically between these two labels.

### ***Data preparation***

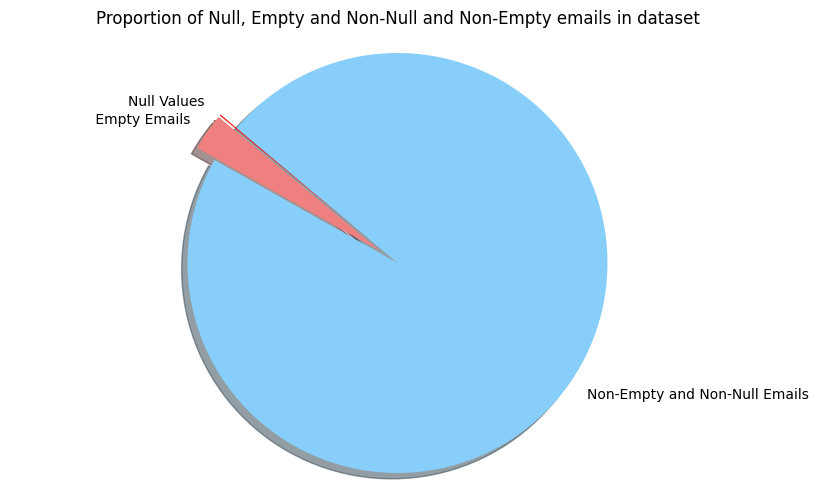
After importing our dataset, we first removed any null rows from the dataset. Next, we decided to remove emails that had no text in the email contents. This decision was made due to inconsistencies with marking empty emails as Phishing or Safe. The most likely source of this inconsistency would be emails that only contained some attachments and thus would be marked Safe or Phishing based on the attachment. However, since our dataset lacked this information, we removed all empty emails. Furthermore, due to how limited our dataset was in terms of features, we added features that would be extracted from the email contents. We extracted email length, if the email is a reply, if the email has a weblink, the number of ‘#’ symbols in an email, the number of ‘?’ symbols in an email, the number of ‘!’ symbols in an email, the number of ‘@’ symbols in an email, the number of ‘.’ symbols in an email, and the number of ‘-’ symbols in an email.

### ***Data integration***

While we are not using more than one dataset, we took inspiration for what features to include from another dataset, namely the count of non-alphabet symbols(Anjali, 2024). The analysis made by most of our inspirations made us look for other approaches to our data and because of that we made an analysis on emails “re:”, with this it helped us to see which was safe and the ones our responders made. Some of the emails contained web links and for that we made an analysis by checking which had a tag link with a message and the ones with just a link without a message but numbers and some symbols.

### ***Observations and features***

The observation about the raw dataset we acquired was good, but we noticed some empty emails with the type specified as safe email and phishing email with a total amount of 18649 emails grouped in either safe emails or phishing emails. A total of 533 empty and null emails were cleared out.



**Chart 1: Proportion of Empty and Null Emails vs. Non-Empty and Non-Null emails**

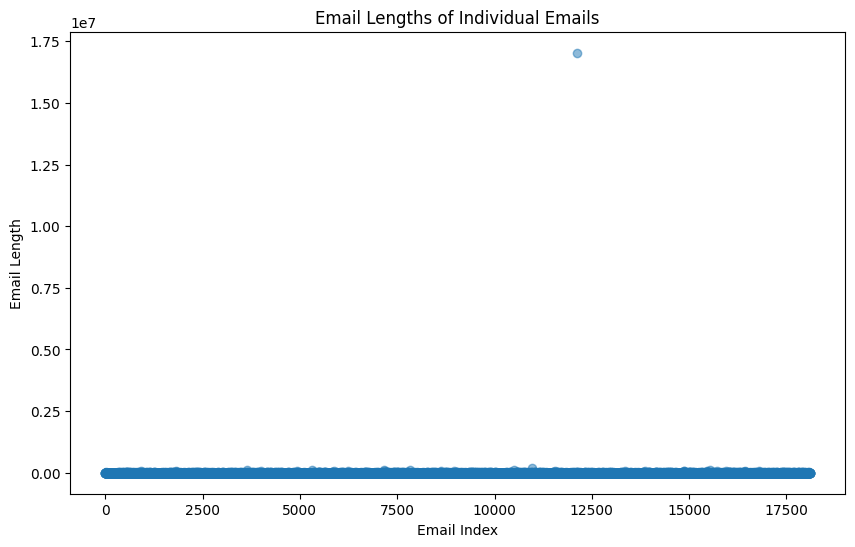
As can be seen in this pie chart, the total proportion of empty emails is very small compared to non-empty and non-null emails, with the proportion of null emails being even smaller. Removing this small chunk of data allowed us to clean up our dataset and avoid any noise they could potentially generate when training our model. In its raw form, our data had 18649 rows and 3 columns; 1 for index of emails, 1 for email text itself and 1 for label columns. After we were done processing and analyzing our data, we were left with 18,101 rows and 19 columns. These additional columns in the dataset contained information about all the features we extracted from the emails in the dataset, including whether the email has a reply, a weblink, the counts of each special character we analyzed and the percent of special characters compared to the actual text and the percentage of capital letters in each email. We include some more charts below to show the features of these features.

### ***Exploratory data analysis (EDA)***

The following features were extracted from the dataset:

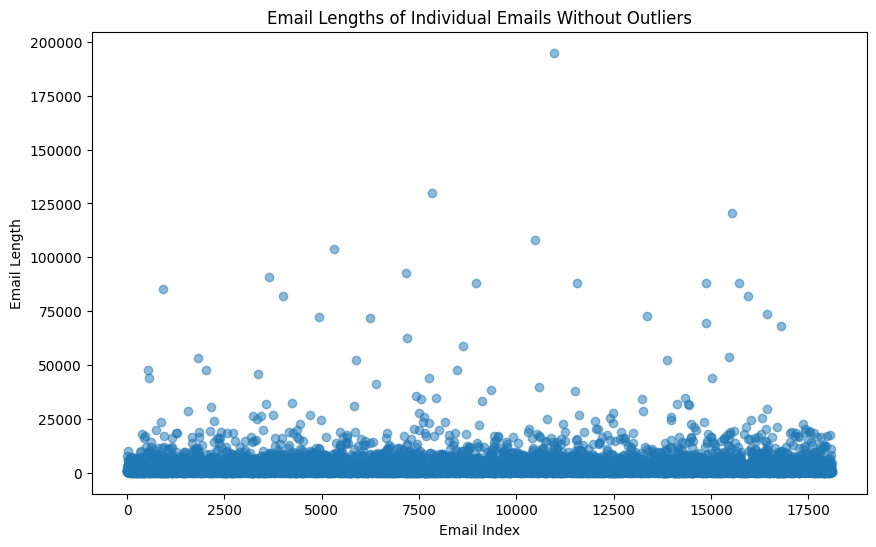
1. Emails with a reply, i.e. emails with the keyword ‘re:’
2. Emails with hyperlinks in them (https)
3. Length of emails
4. Special Characters in Emails and Their Frequency
5. Percentage of All-Capital characters in emails
6. Sentiment/Tone of Emails

We now showcase some plots that explain these features.



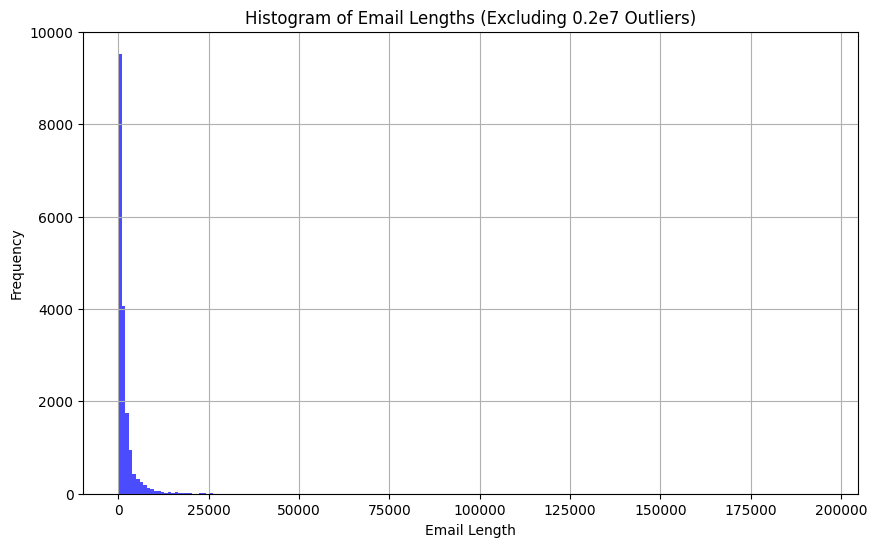
**Chart 2: Scatter Plot of Email Length of Individual Emails**

The initial drawing of the scatter plot shows that we have an outlier, which blasts the range of the plot into 10^7. This has diminished the effectiveness of this visualization, so in the next visualization, we will be repeating this one without that outlier. However, as this is a part of our data, we will not drop it.



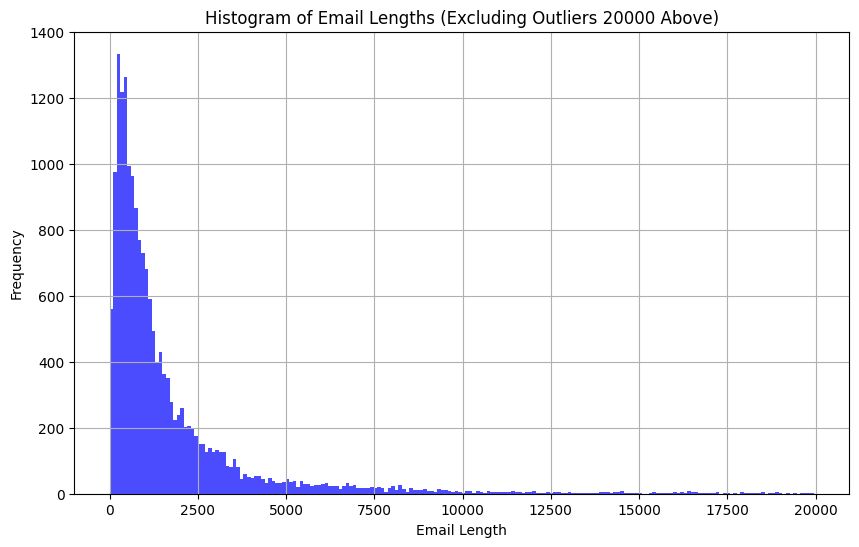
**Chart 3: Scatter Plot of Email Lengths excluding the outlier**

As can be seen in this plot without the outlier, most of the emails in the dataset are under 25000 characters in length, while we have comparatively few emails larger than that. This can be seen even more clearly through a histogram.



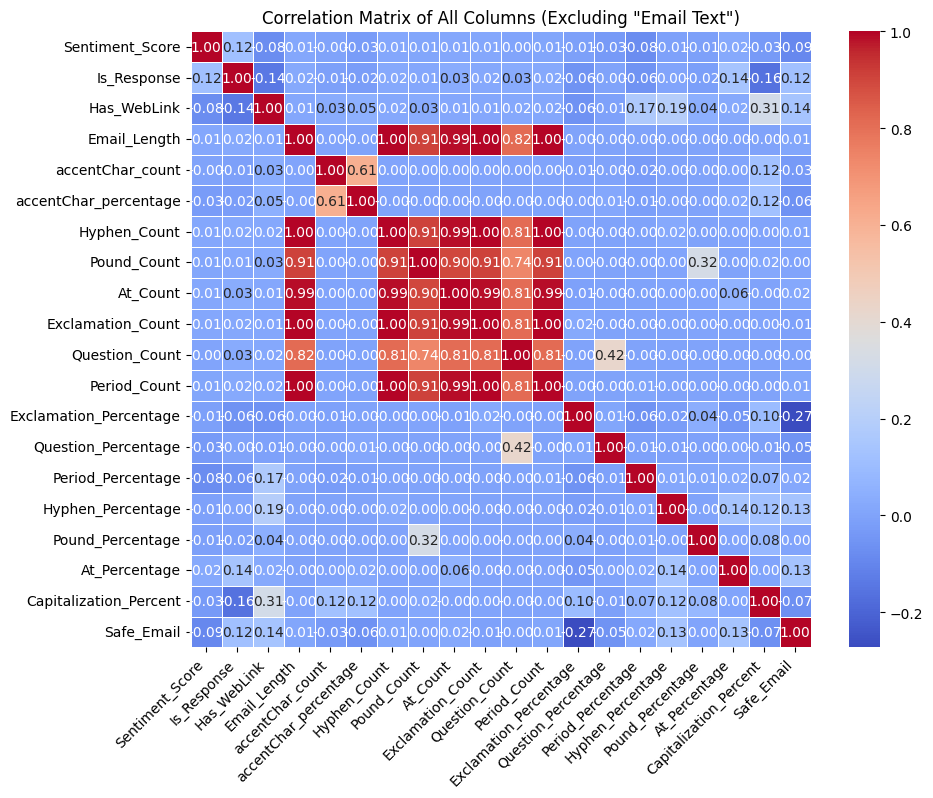
**Chart 4: Histogram of email spread excluding outlier emails of length > 200,000 characters**

This histogram shows the skew of email lengths (excluding the massive 0.2e7 length email). As noted in the scatter plot, we see that most of our emails are under 25,000 characters. The histogram has a massive left skew.



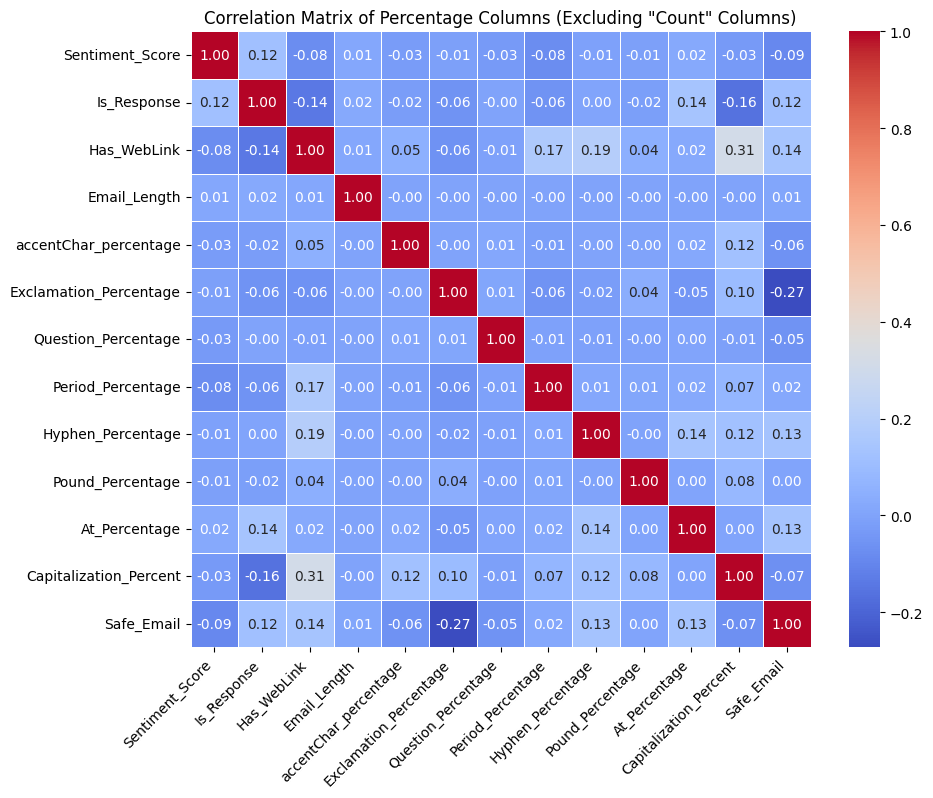
**Chart 5: Histogram showing the spread of email lengths excluding outliers**

This histogram shows the frequency distribution of the most commonly occurring lengths of emails by excluding outlier emails that are longer than 20000 characters. This gives us a more accurate and nuanced view of how long the emails in the dataset are. While we do see a heavy skew towards the left with a majority of the emails being less than 2500 characters, this histogram can highlight the diversity of email lengths in the dataset.

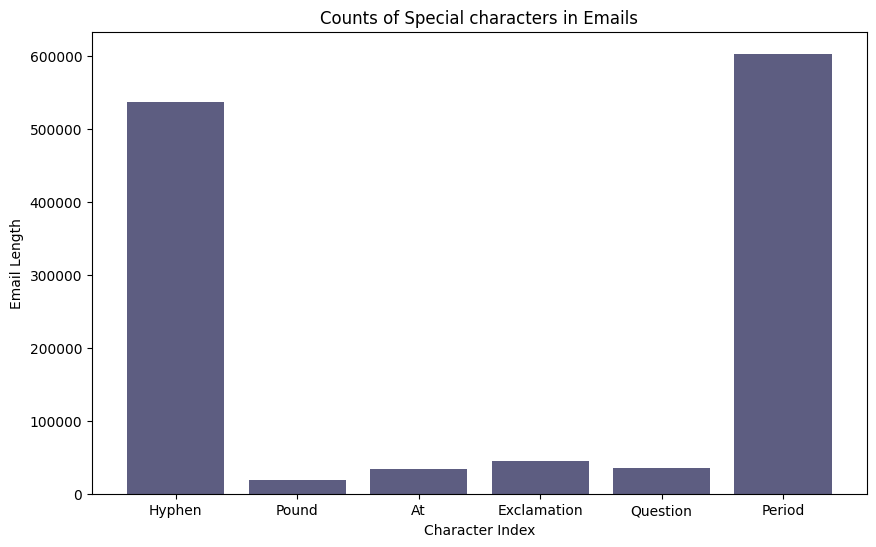


**Chart 6a: Heatmap showing correlation matrix between different features**

The correlation matrix with a heatmap displays the strength of the correlation between the pairs of features. Values closer to 0 represent a weaker relationship. A Pearson correlation coefficient that equals 1 indicates a perfect positive correlation. A value of -1 is perfectly negatively correlated. The strongest correlations exist between “Email\_Length” and the symbol “Count” features. These values are between 0.7 and 1.0. This makes sense intuitively, since the length of the email is probably a great predictor of how often symbols show up in the text. We made the “Percentage” features to measure the frequency of symbols so that it is no longer just a proxy for the length of the email. Dropping the “Count” features creates a much cooler correlation matrix (see Chart 6b).

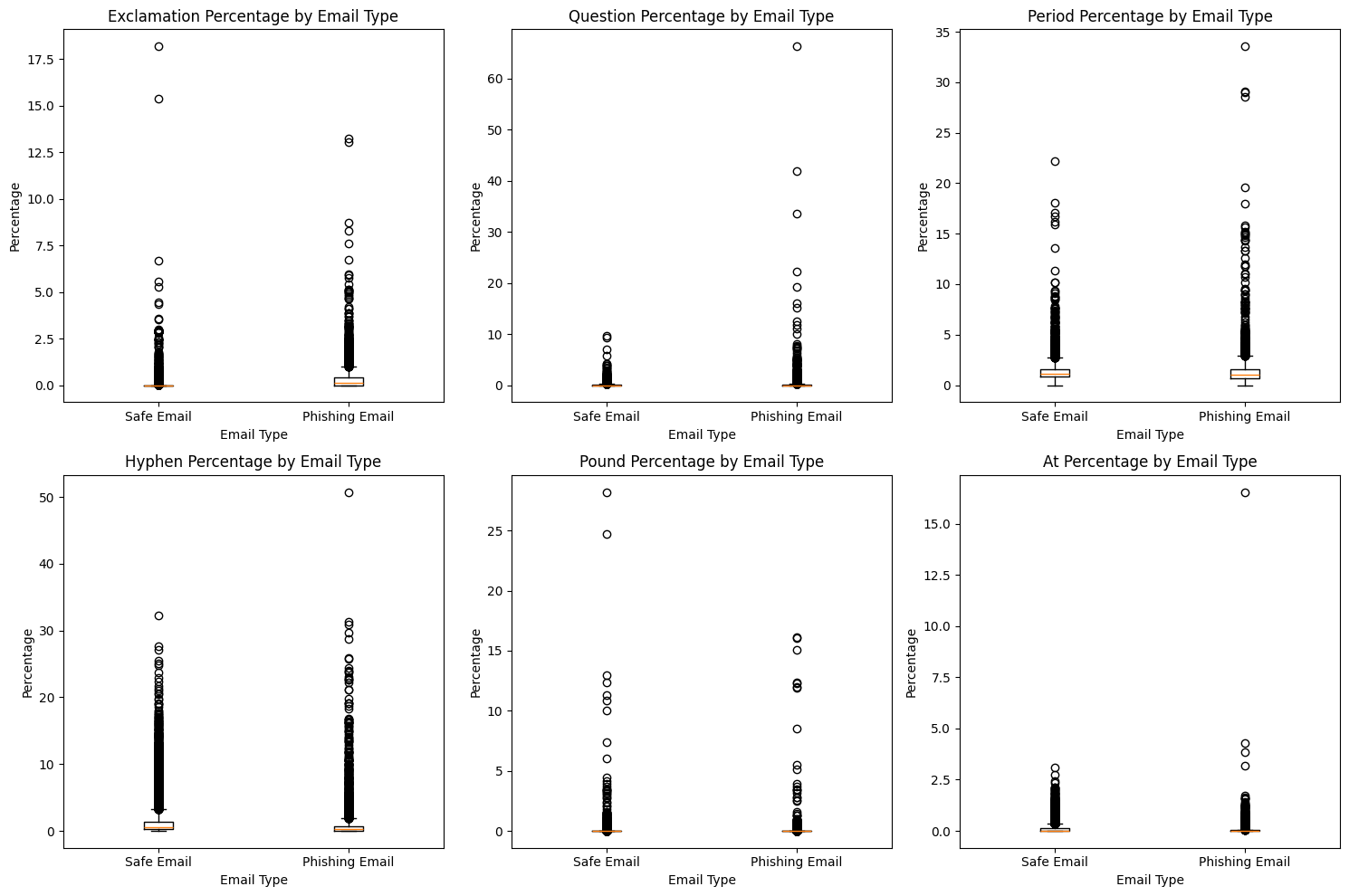


**Chart 6b: Heatmap showing correlation matrix between different features**



**Chart 7: Counts of special characters in the emails**

Chart 7 shows that the period is the most popular special punctuation character followed by the hyphen.

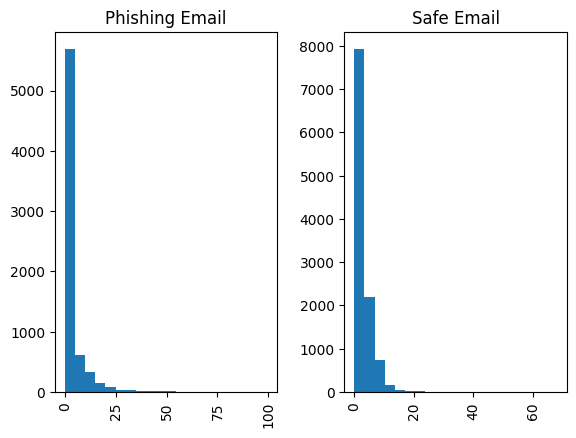


**Chart 8: Box Plot showcasing percentage of special characters by Email Type**

Here (in Chart 8) we have the special character percentage, which is calculated by counting the number of characters multiplied by 100 in the email and dividing by the email length.

Notably, there is extreme skewness in the spread of the frequency of some symbols across the emails. The spread is comparable between safe and unsafe emails, except for a difference in the extreme outliers.

The capitalization percentage (see Chart 9) has a wider range for the phishing emails, but both have an extreme left skew.



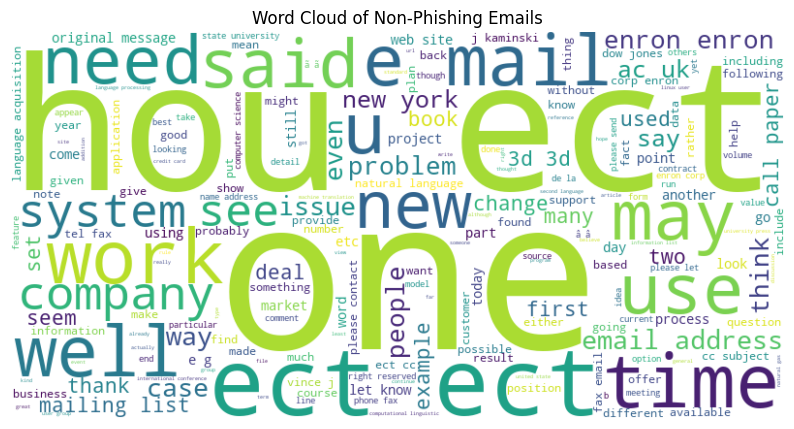
**Chart 9: Percentage of Capitalization in Safe and Phishing emails**

Finally, we used the Natural Language Toolkit (nltk) library tools to tokenize. Each email was “tokenized,” meaning they were separated into lists of words for processing.

To better understand the common content across the emails we created word clouds to compare the phishing and safe emails (see Charts 10 & 11). The key takeaway and promising finding is that there are all sorts of look-a-like special characters inside the phishing emails that are not present in the safe emails. We need more features.

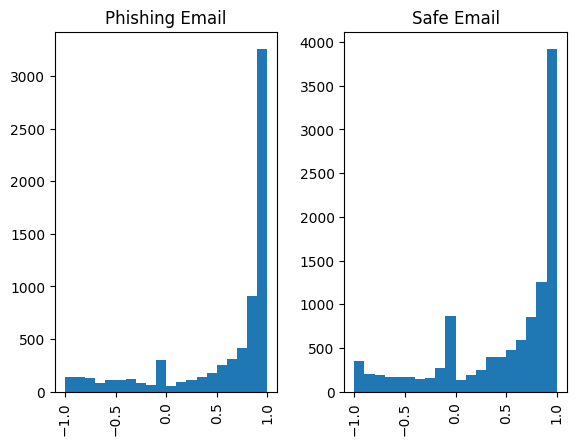


**Chart 10: Wordcloud showing the frequency of most commonly occurring characters in Phishing emails**



**Chart 11: Wordcloud showing the frequency if most occurring characters in Safe/Non-Phishing emails**

The counts of special characters look high, but this is contributed by a small amount of emails. After further investigation, there are 331 emails with one of the following characters: àáâãäåæçèéêëìíîïðñòóôõöøùúûüýÿ. This is quite underwhelming, despite the appearance of the word cloud. This finding highlights the weakness of word clouds as an analytical tool.



**Chart 12: Wordcloud showing the frequency if most occurring characters in Safe/Non-Phishing emails**

Finally, the Sentiment Analysis Score does not show much difference in the spread between the phishing and safe emails (see Chart 12). There is a right skew. Most emails are relatively positive according to the NLTK Vader tool.

### ***References***

Anjali, Sima (2024, February 24) *Phishing Dataset.* Kaggle. <https://www.kaggle.com/datasets/simaanjali/tes-upload>

Chakraborty, S. (2023, July 6). *Phishing email detection*. Kaggle. <https://www.kaggle.com/datasets/subhajournal/phishingemails/data>

Cisco. (2024, February 15). *What is phishing? examples and phishing quiz*. Cisco. <https://www.cisco.com/c/en/us/products/security/email-security/what-is-phishing.html#~phishing-awareness>

Hutto, C.J. & Gilbert, E.E. (2014). VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text. Eighth International Conference on Weblogs and Social Media (ICWSM-14). Ann Arbor, MI, June 2014. [https://www.nltk.org/\_modules/nltk/sentiment/vader.htmltk.sentiment.vader](https://www.nltk.org/_modules/nltk/sentiment/vader.html)

Threatcop. (2023, October 17). *Reasons behind successful phishing attacks*. <https://threatcop.com/blog/how-are-phishing-attacks-successful/>

Woods, E. (2021, February 9). *The real reason for successful phishing attacks*. usecure Blog. <https://blog.usecure.io/the-real-reason-why-phishing-attacks-are-so-successful>