Spam Detection in Emails P27

Janelle Correia(jlcorrei), Jayesh Gajbhar(jgajbha), Skanda Shastry(sshastr4), and Tushar Kini(tkini)

Department of Computer Science, NC State University

Repo Link: [https://github.ncsu.edu/jlcorrei/engr-csc522-fall2023-P27]

1 Introduction

In an age where email communication has become an integral part of both personal and professional life, the influx of unsolicited and potentially harmful messages, known as spam, has grown to be a significant concern. Spam emails not only clutter inboxes but also pose security risks, making effective spam detection systems essential. Our project, "Spam Detection in Emails," endeavors to perform an analysis on the current methods of email classification and give insight to which method is the most impactful.

This midway report serves as a progress update on our project, highlighting the key milestones achieved and the path forward to the final implementation of our spam detection system. Over the course of this report, we will discuss the project's goals, the methodology employed, the data sources and preprocessing, the machine learning models considered, and our evaluation metrics.

2 Methodology

2.1 Algorithms:

2.1.1 SVM

Support Vector Machine (SVM) is a type of supervised learning method which is typically used for classification as well as regression problems. This method is known to be very efficient for a dataset with high dimensionality. The SVM algorithm attempts to locate a hyperplane in an n-dimensional space where n is the number of features which can distinctly classify any data point. The aim is to find such a plane while maximizing the margin so that a data point can be classified with a high level of confidence going forward.

2.1.2 Naive Bayes

The naive bayes classifier is based on the Bayesian theorem and is usually quite helpful when the number of input dimensions is large. A naive-bayes classifier assumes strong independence among the various attributes of data, but still manages to produce great results. This classifier generates a trained model with efficiency using very few data points given its simple design and oversimplified assumptions. The independence between attributes makes it easy to compute only variances instead of an exhaustive covariance matrix, and this estimation allows us to perform classification.

2.1.3 KMeans

KMeans is an unsupervised machine learning algorithm used for partitioning datasets into K distinct clusters, where each data point belongs to the cluster with the nearest mean. It operates by iteratively assigning data points to centroids and updating centroids based on the mean of points in each cluster. KMeans is computationally efficient, making it suitable for large datasets, but it assumes spherical, equally sized clusters and is sensitive to initial centroid placement. Despite these limitations, it's widely applied in customer segmentation, image compression, and anomaly detection, providing valuable insights into data patterns.

2.1.4 K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) is a simple yet effective supervised machine learning algorithm used for classification and regression tasks. It assigns a data point to the majority class among its K nearest neighbors, determined by a distance metric like Euclidean distance. KNN is intuitive, non-parametric, and doesn't assume any underlying data distribution. Its performance hinges on the choice of K and the distance metric. Despite its simplicity, KNN is widely applied in diverse fields, including recommendation systems and image recognition, for its ability to capture intricate patterns in data based on local neighborhood information.

2.1.5 TF-IDF (Term Frequency-Inverse Document Frequency)

TF-IDF, short for Term Frequency-Inverse Document Frequency, is a widely used text processing technique for information retrieval and text mining. It quantifies the importance of a term in a document relative to a collection of documents. TF-IDF is computed by multiplying the term frequency (TF), representing how often a term appears in a document, with the inverse document frequency (IDF), which measures how unique or rare a term is across the entire document collection. This calculation reduces the significance of common words and highlights rare, distinctive terms, making it valuable for tasks like text classification, clustering, and search engine ranking. Despite its simplicity, TF-IDF is powerful in capturing the semantic meaning of words within a document, aiding in tasks that require understanding the relevance and context of terms within a textual corpus.

2.1.6 Word2Vec

Word2Vec is a popular unsupervised learning algorithm used for natural language processing tasks, particularly word embedding. It represents words as dense vector spaces in a continuous vector space, capturing semantic relationships between words. Word2Vec employs neural networks to learn word embeddings based on the context of words within a large corpus of text. It operates on the principle that words appearing in similar contexts have related meanings. Word2Vec models, such as Continuous Bag of Words (CBOW) and Skip-gram, learn to predict surrounding words from a target word or vice versa. These embeddings preserve semantic information, enabling tasks like text similarity, named entity recognition, and sentiment analysis. Word2Vec's ability to capture nuanced word meanings and relationships makes it indispensable for various natural language processing applications, enhancing the performance of downstream machine learning models.

3 Experiment setup

3.1 Dataset

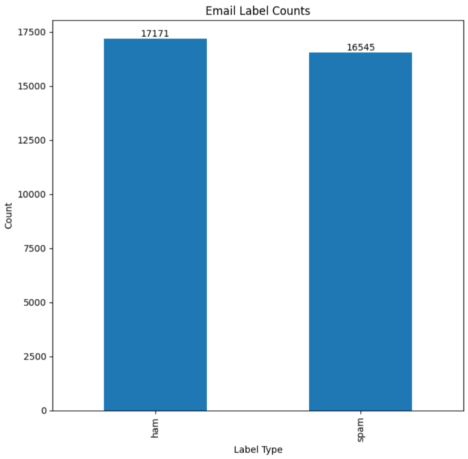
Our project will make use of the Enron-Spam dataset link . It contains over total emails along with the subject and message, each labelled as either spam or ham.

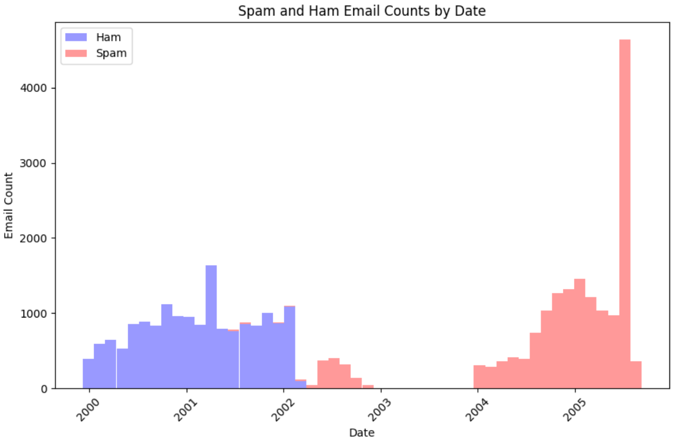
3.2 Pre-processing

Data pre-proccessing played a very important role in the performance optimization of our project so far. We did comprehensive data cleansing to ensure the quality and relevance of our dataset. Specifically, for forwarded emails, we performed a thorough check for null values to ensure that our dataset was complete and free of missing information. We also improved the quality of our subject headers by removing extraneous punctuation and words like "re:". These steps were essential in refining our dataset to allow our algorithms (see above) to operate effectively and to derive meaningful insights from the data. Proper data preprocessing is a critical step that allowed us to identify valuable information and data patterns.

3.3 Exploratory Data Analysis

Based on the resulting data visualization following data preprocessing and tokenization, we see that the Ham vs. Spam histogram contained more instances of ham emails than spam emails, with a difference of 626 between the two subgroups (17,171 ham emails vs 16,545 spam emails).

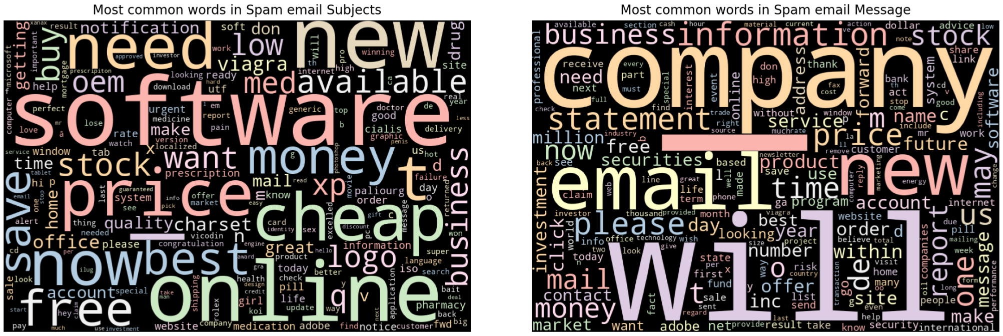




Based on the resulting data visualization of the Ham/Spam Email Counts by Date histogram showed ham data primarily concentrated between 2000 and 2002, with some overlapping spam data during that period, and with spam data concentrated more in mid-2002 to 2003, and heavily from 2004 to 2005. This could suggest a shift in email communication dynamics from 2002 to 2003, given the shift from a higher concentration of ham emails to spam emails during this time. There may also be a possible impact of spam filtering, suggesting that the individuals started receiving a higher volume of unwanted email content and that spam filtering/email security measures became more necessary during this period. Additionally, the Enron dataset may have undergone data cleanup or changes with processing during specific time frames that affected how the emails were categorized.

In our project, we also used Word Cloud analysis to gain insights into the most common terms within the subjects and messages of spam emails. The Word Cloud visualization allowed us to identify patterns and frequently occurring keywords in spam email subjects and messages, which will allow us to understand the common themes and tactic used by spammers, which could benefit email filtering and classification algorithms that we may develop in the future.

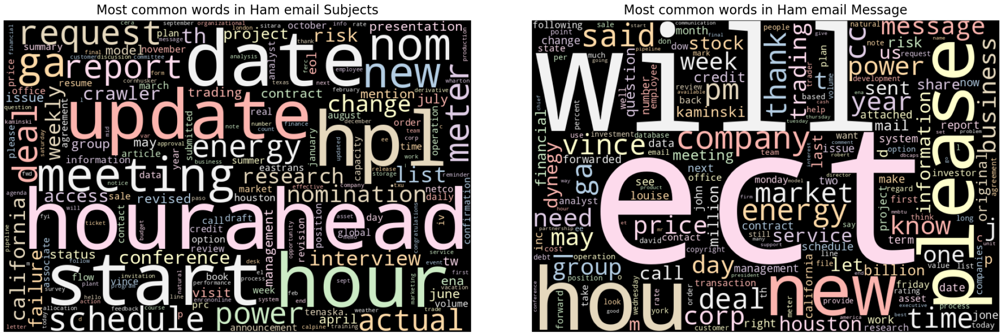
We also analyzed the common words in legitimate (ham) email subjects and messages. By extracting insights from the subjects and messages of legitimate emails, we hope to develop a better understanding of typical subject lines in genuine communication, which will make it easier to distinguish between legitimate and spam emails and improve the accuracy of email classification systems that are developed in the future.



(a) Spam Subject Word Cloud

(b) Spam Message Word Cloud

Figure 1: Spam Dataset Word Clouds



(a) Ham Subject Word Cloud

(b) Ham Message Word Cloud

Figure 2: Ham Dataset Word Clouds

Our Word Cloud analysis offered an intuitive and visual way to discern the most significant terms in various aspects of email data, which enhanced our understanding of email content, provided valuable insights for distinguishing between spam and legitimate emails, and will hopefully contribute to the development of more effective email classification and filtering algorithms that we or others develop in the future.

3.4 PCA

Principal Component Analysis (PCA) is a dimensionality reduction technique commonly used in data analysis. It transforms a dataset into a new coordinate system, where the first principal component explains the most variance in the data, followed by the second, and so on. PCA is useful for simplifying complex datasets, identifying patterns, and reducing noise, making it easier to visualize and analyze data while preserving the most important information.

3.5 Cross Validation

We used the Cross-Validation method in our project to assess the performance and generalization ability of our analysis models. The Cross-Validation technique helps to mitigate issues related to overfitting and gives a more reliable estimate of a model's performance. Specifically, we used a 10-fold Cross-Validation strategy, where the dataset was divided into ten approximately equal-sized subsets. In each iteration, nine subsets were used for training the model, while the remaining subsets were used for testing. We repeated this process 10 times to and assigned each subset as the test set once. We then averaged the results from each iteration to obtain a more accurate evaluation that best represented our models. Going forward, we plan to enhance our models even further by using different values in subsequent phases to refine performance (time permitting). The 10-fold

Cross-Validation approach served as a good foundation for our models, by which we were able to validate our models and increase confidence in their capacity for data generalization.

4 Results

- We see that the spam and ham classes are well segregated when a histogram with the date column is made. This gives us intuition that the K-means algorithm may perform well as there is a feature that separates the 2 distributions.
- We note that that some common words in Spam email subjects are *Software, now, cheap, new, free, online, save, buy and money*
- We note that that some common words in Spam email messages are *company, will, email, new, business and money*
- We see that the 2 classes *Spam* and *Ham* are almost equal in number, but predicting an email as spam is more crucial. So we can use *Accuracy* and *Recall* as we try to reduce the False negative rate.

Metric	SVM	NB
Accuracy	0.97	0.99
F1	0.97	0.99
Recall	0.98	0.99
Precision	0.96	0.98

5 Conclusion

5.1 Next Steps

- We see that the accuracy from SVM and NB classifier is very high that is an indication that there is overfitting of the data or there might be a problem with the preprocessing of the data.
- We need plan to use stemmatization and lemmatization to improve accuracy.
- Since Word2Vec can capture context from a text, we plan to use that as well to compare with TD-IDF and see what method gives better results.
- We plan to use PCA with Word2vec (attribute size=100) and then apply all the 4 algorithms so that the results can be compared. We will use the elbow method to decide the number of PC's to be considered.
- For KNN we will experiment with different values(3,5,10) to get the best metrics.
- We will try to devise a cost matrix that would help to decide the cost of each method and decide on which method can be employed beyond metrics.

References

[1] Mangena Venu Madhavan et al. "Title of the Paper". In: *IOP Conf. Ser.: Mater. Sci. Eng.* 1022 (2021), p. 012113. DOI: 10.1088/1757-899X/1022/1/012113. URL: https://iopscience.iop.org/article/10.1088/1757-899X/1022/1/012113/pdf.

[2] T. Sultana. "Email based Spam Detection". In: *International Journal of Engineering Research* 9 (2020). DOI: 10.17577/IJERTV9IS060087. URL: https://www.researchgate.net/publication/342113653_Email_based_Spam_Detection.

[**Chhabra2010**, 2, 1]