D214 Data Analytics Graduate Capstone

Machine Learning [SPAM](https://en.wikipedia.org/wiki/Spamming) Detection Powered by Enron/TREC Public Spam Corpus

Table of Contents

[Problem and Hypothesis Statement 1](#_Toc153538238)

[Summary of Data-Analysis Process 1](#_Toc153538239)

[Outline of Findings 2](#_Toc153538240)

[Limitations of Techniques/Tools Used 2](#_Toc153538241)

[Proposed Actionable Items 2](#_Toc153538242)

[Benefits of Study 2](#_Toc153538243)

# Problem and Hypothesis Statement

**Problem:**  
  
Spam presents a significant challenge for both businesses and households. It not only consumes valuable storage space and impedes the identification of critical emails, but it also poses a substantial security risk. Spam emails can serve as conduits for malicious threats such as viruses, worms, phishing scams, and ransomware. Therefore, implementing models capable of accurately detecting spam is essential. These models are not only crucial for enhancing security measures but also for ensuring that email server storage is reserved exclusively for important organizational communications.   
  
**Hypothesis:**The machine learning model developed using the 2007 TREC Public Spam Corpus and Enron emails dataset can accurately classify new content as SPAM or non-SPAM (HAM) with an accuracy score of 95% or higher. This suggests that the model has learned effective patterns within the dataset, enabling it to differentiate between SPAM and non-SPAM content significantly better than chance.

# Summary of Data-Analysis Process

1. Data Collection
   1. The data collection process was pre-done as it was a prepared and labeled dataset found on [Kaggle.com](https://www.kaggle.com/). The data was already collected and labeled so that it would be ready for a machine learning model to use it. It was collected from Enron emails with email content labeled as SPAM or HAM, where 1 indicates SPAM and 0 indicates HAM, which is not SPAM. The dataset contained about 84000 entries that is almost a 50/50 coming in at 47.38% as HAM and 52.62% SPAM showing the split so very little bias was in the data.
2. Data Extraction and Preparation
   1. Load data:
      1. Due to some size limitations with GitHub where the work is stored the original data had to be split into two files. So, for this process it was simply loading the two files and merging them back together into a single Polars dataset.
   2. Check for missing data:
      1. The dataset creator on Kaggle stated that there was no missing data. This was verified using code, via [Polars](https://pola.rs/) DataFrame function ‘[.null\_count()](https://pola-rs.github.io/polars/py-polars/html/reference/dataframe/api/polars.DataFrame.null_count.html)’ on each feature label and text, which were renamed to IsSpam and EmailContent for ease of understanding.
   3. Check balance of labeled data:
      1. To check the balance of the labeling (IsSpam) Polars Series ‘[.hist()](https://pola-rs.github.io/polars/py-polars/html/reference/series/api/polars.Series.hist.html)’ function was used to get break points which were 1 and 0 and their respective counts. HAM(0) had a count of 39538 and SPAM(1) had a count of 43910. This gave us our previously mentioned 47.38%/52.62% split, which showed the data is relatively balanced and unbiased.
   4. Pre-Process text data into vectors (embeddings):
      1. The machine learning model that was used to detected SPAM was Logistical Regression. Because this is a mathematical model unstructured text like in an email cannot be processed directly. To get around this and allow the Machine Learning Model to learn which kind of text is associated with a label of SPAM or HAM it will need to be vectorized, which is also known as embeddings.   
           
         To do this the text needs to be pre-processed and tokenized. Tokenization is text broken up into smaller parts. These steps include lower casing all text, remove Unicode text, remove emojis from text, removing punctuation from text. After that is completed, the text is converted into tokens and the last step applied is to remove STOP words from the tokenized word collection. STOP words are words that do not carrying specific meaning such words like "the", "is", "in", "for", and "on" in the English language.

# Outline of Findings

# Limitations of Techniques/Tools Used

# Proposed Actionable Items

# Benefits of Study