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](#_Toc153541164)

[Summary of Data-Analysis Process 1](#_Toc153541165)

[Outline of Findings 2](#_Toc153541166)

[Limitations of Techniques/Tools Used 2](#_Toc153541167)

[Proposed Actionable Items 2](#_Toc153541168)

[Benefits of Study 2](#_Toc153541169)

# 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

The data-analysis process for the SPAM detection project can be summarized in several key steps:

1. **Data Collection**:
   * Source: The dataset was obtained from [Kaggle.com](https://www.kaggle.com/), consisting of pre-labeled Enron email contents.
   * Labeling: Emails were categorized as 'SPAM' (1) and 'HAM' (0), where HAM refers to non-SPAM emails.
   * Dataset Size and Balance: The dataset comprised approximately 84,000 entries, with a nearly even split of 47.38% HAM and 52.62% SPAM, indicating minimal bias.
2. **Data Extraction and Preparation**:
   * Loading Data: Due to GitHub size limitations, the dataset was split into two parts, which were subsequently merged into a single Polars dataset.
   * Missing Data Check: The dataset's completeness was affirmed by the creator and verified using the [Polars](https://pola.rs/) DataFrame [**.null\_count()**](https://pola-rs.github.io/polars/py-polars/html/reference/dataframe/api/polars.DataFrame.null_count.html#polars.DataFrame.null_count) function, confirming no missing data.
   * Label Balance Verification: The balance of labeling was checked using the Polars Series [**.hist()**](https://pola-rs.github.io/polars/py-polars/html/reference/series/api/polars.Series.hist.html) function, revealing a balanced distribution of 39,538 HAM and 43,910 SPAM entries.
3. **Text Data Preprocessing**:
   * Preparing for Machine Learning: The project utilized Logistic Regression, necessitating the conversion of unstructured text into a structured format.
   * Vectorization (Embeddings): Email contents were vectorized through a series of preprocessing steps:
     + Text Normalization: Converting all text to lowercase and removing Unicode characters and emojis.
     + Punctuation Removal: Stripping away punctuation from the text.
     + Tokenization: Breaking down the text into smaller parts or tokens.
     + Stop Word Removal: Eliminating common words (e.g., 'the', 'is', 'in') that do not contribute to the specific meaning in the context of SPAM detection.

This structured approach ensured the dataset was optimally prepared for effective machine learning model training and SPAM detection analysis.

# Outline of Findings

# Limitations of Techniques/Tools Used

# Proposed Actionable Items

# Benefits of Study