Email Spam Detection and Filtering System

Samruddhi Khairnar - kbtug20170@kbtcoe.org

Introduction

As a part of my internship at InternsElite, I chose ***Email Spam Filtering System*** as the topic of my minor project. This report briefly outlines the system design of my project, starting with the creation of a relevant dataset to classify spam and non-spam emails, till the possible deployment of the system for real world applications.

Problem Definition

Email Spam is a common phenomenon nowadays. Spam emails are driven by commercial / financial motives where spammers make false claims and deceive recipients into believing something that isn't true. Hence, in order to address this issue, I intend to design a spam filtering system using machine learning techniques.

Objectives

- To learn web scraping to scrape common spam as well as non-spam email phrases.
- To learn and implement NLP techniques on the scraped email dataset.
- To train a classification model to classify emails as spam or non-spam.
- To increase the prediction accuracy by tuning the models' hyperparameter.
- To deploy the trained spam classification model to automatically delete spam emails in GMail Inbox.

Project Category

Data Science – *Machine Learning **(ML)** and Natural Language Processing **(NLP)***

Software Tools Required

- **Python 3** - It is a high-level, general-purpose programming language.
- **JupyterLab** - It is a web-based interactive computing platform.
- **Libraries** :
    1. **Pandas** - Open source data analysis and manipulation tool, using Python.
    2. **Scikit-Learn** - Library for implementing machine learning in Python.
    3. **NLTK, Spacy** - Suite of libraries for natural language processing for English.
    4. **Beautiful-Soup** - Library to scrape information from web pages.

Hardware and Software Requirements

- **Microprocessor** - Intel Core i5 / i7 (>= 6th gen)
- **RAM** - 8 GB or more
- **Operating Systems** - Windows / Linux / Macintosh

Requirement Specifications

Functionality

- Users should be able to classify the emails they've received, as spam or non-spam, using the trained model.
- They should be able to customize the deployment script and integrate the spam filtering model with their GMail accounts.

Platform

The trained model shall be deployed on any cloud platform, to be accessible to all users.

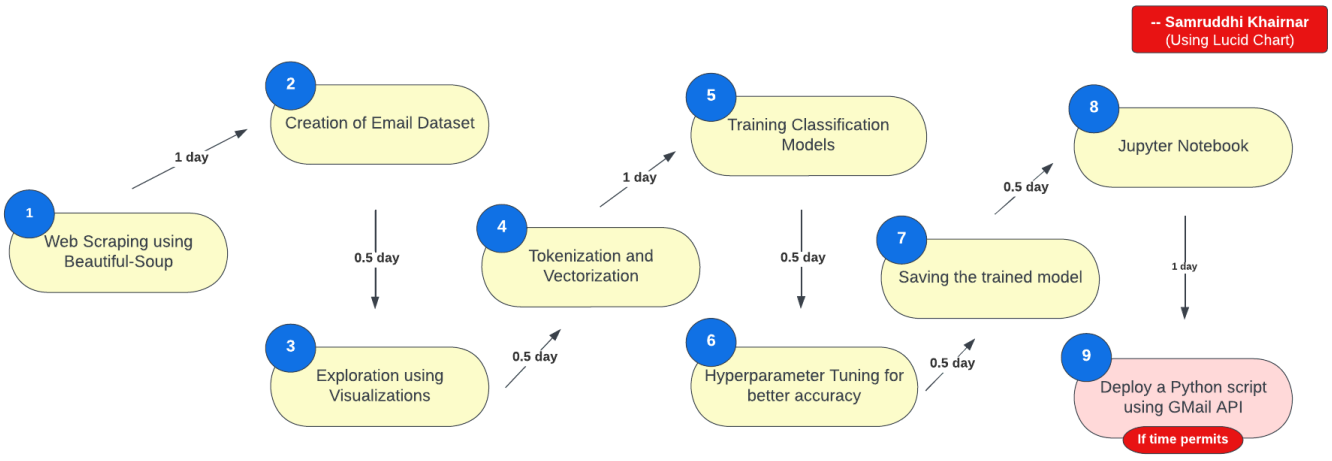
Deliverables

1. This report.
2. Dataset Scraping Source code as a Jupyter Notebook.
3. Model Training Source code as a Jupyter Notebook.
4. The trained model - a pickle file.
5. The deployment script - using GMail Python API *(will be implemented if time permits)*.

Project Scope

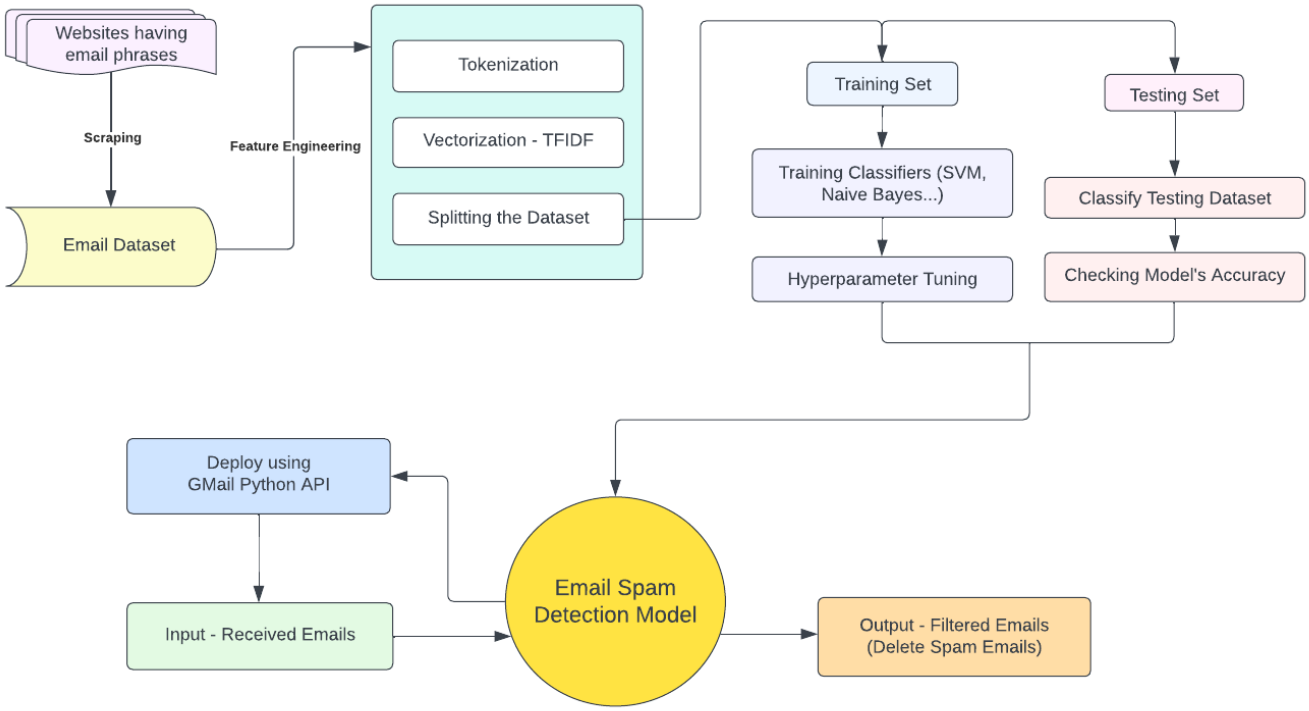
The scope of this project is restricted to extracting simple feature vectors from email data scraped from 2-3 websites and training classification models to classify the emails as spam or non-spam. If time permits, the scope will include building a short python script to filter email spam in GMail inboxes.

Project Scheduling - PERT



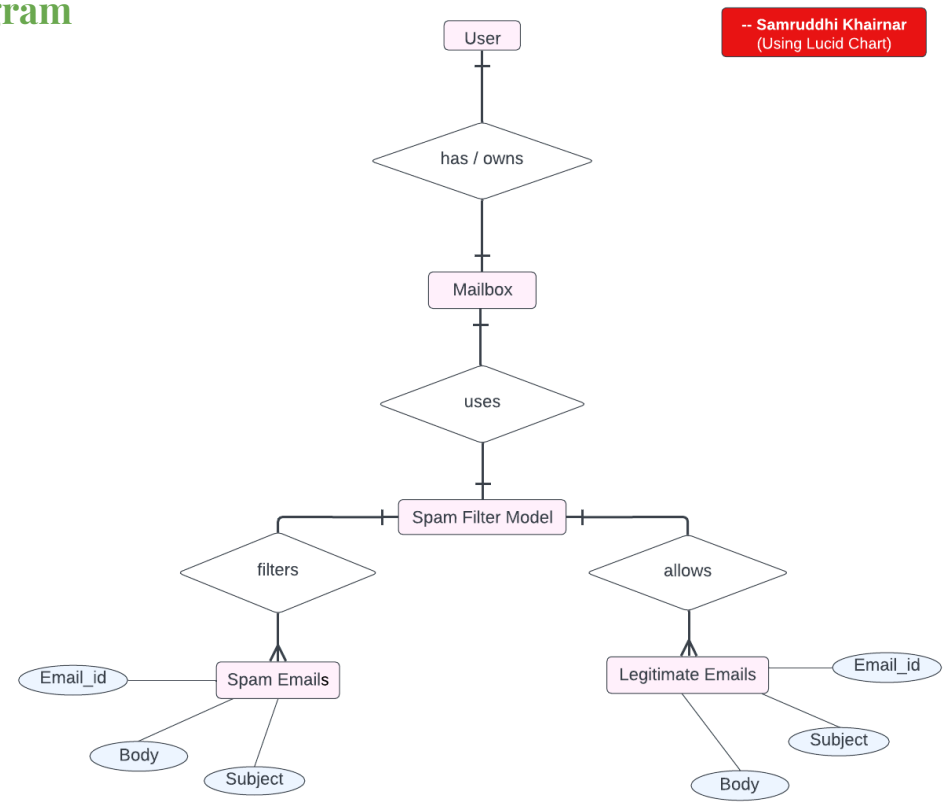
Analysis - DFD level 0

-- Samruddhi Khairnar
(Using Lucid Chart)

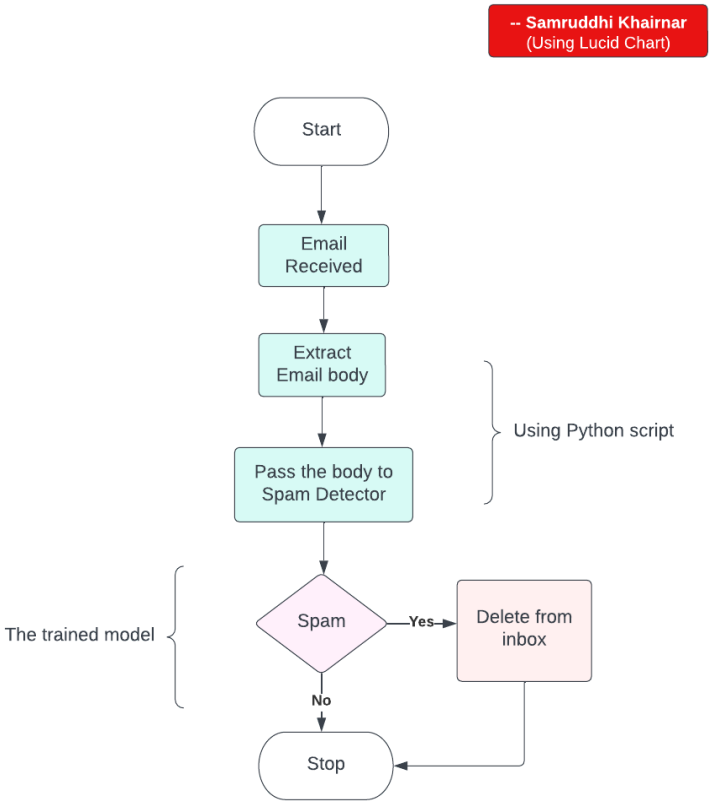


ER Diagram

-- Samruddhi Khairnar
(Using Lucid Chart)



System Design Flowchart



Dataset

References *(to scrape email data):*

1. Spam Email Phrases : https://www.softwarepundit.com/email-marketing/email-spam-words#nogo
2. Non-Spam Email Phrases: https://www.getmailbird.com/business-email-example/

Features

Of the scraped dataset, (for example) :

Email Phrase - (object type)	Spam - (int type, 0=No, 1=Yes)
Hi there good to see you	No
Please invest in our scheme	Yes

Implementation procedure

1. I plan to create my own scraped dataset by scraping common email spam + legitimate email phrases from the above mentioned websites.
2. After scraping, I will use tokenization and vectorization to create feature vectors for training the ML models.
3. I will train 3-4 classification models (SVM, Decision Trees, Naive Bayes) and evaluate their accuracy, preceeded by tuning of their hyperparameters.
4. After selecting the most accurate model, I will save it and test it on unseen data.
5. Lastly, I will try to deploy the model to filter spam emails in my GMail inbox, **if time permits**.

Security Concerns

In future, if I plan to deploy the spam detection script on any cloud platform, I will have to employ certain encryption techniques to secure the logic ids and passwords of the users (required to access the emails in their mailbox to filter and remove spam).