Email Spam Detection

Last date of revision: 26/02/2024

# Document Version Control

|  |  |  |
| --- | --- | --- |
| Date Issued | Description | Author |
| 26/02/2024 | Initial HLD — V1.0 | Sarthak Bathla,  Srishti Gupta |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## **Contents**

[Document Version Control. 2](#_TOC_250021)

[Abstract. 4](#_TOC_250020)

1. [Introduction 5](#_TOC_250019)
   1. [Why this High-Level Design Document?. 5](#_TOC_250018)
   2. [Scope. 5](#_TOC_250017)
   3. [Definitions 5](#_TOC_250016)
2. [General Description. 6](#_TOC_250015)
   1. [Product Perspective 6](#_TOC_250014)
   2. [Problem statement 6](#_TOC_250013)
   3. [PROPOSED SOLUTION 6](#_TOC_250012)
   4. [FURTHER IMPROVEMENTS 6](#_TOC_250011)
   5. [Technical Requirements. 7](#_TOC_250010)
   6. [Data Requirements 7](#_TOC_250009)
   7. [Constraints 9](#_TOC_250007)
   8. [Assumptions. 9](#_TOC_250006)
3. [Design Details 10](#_TOC_250005)
   1. [Process Flow. 10](#_TOC_250004)
      1. [Model Training and Evaluation 10](#_TOC_250003)
      2. [Deployment Process 11](#_TOC_250002)
   2. [Event log 11](#_TOC_250001)
   3. [Error Handling 11](#_TOC_250000)
   4. Performance. 12
   5. Reusability. 12
   6. Application Compatibility 12
   7. Resource Utilization 12
   8. Deployment. 12
4. Dashboards. 13
   1. KPIs (Key Performance Indicators) 13
5. Conclusion 14

**Abstract**

This project focuses on the development of an effective email spam detection system using the Multinomial Naive Bayes (MNB) classifier from the scikit-learn library. With the escalating challenge of spam emails compromising user experience and security, the implementation of a robust spam detection mechanism becomes essential. The project aims to harness the power of MNB, a well-established algorithm for text classification, to accurately distinguish between spam and legitimate emails. By utilizing features derived from email content, the classifier can learn patterns indicative of spam, providing a reliable means to filter out unwanted messages. The proposed solution offers a streamlined and efficient approach to email management, contributing to enhanced user privacy and protection against potential cyber threats such as phishing and malware attacks. The adaptability of the MNB model ensures its relevance in addressing evolving spamming techniques, ultimately fostering a safer and more secure email communication environment.

1. **Introduction**

###### Why this High-Level Design Document?

The purpose of this High-Level Design (HLD) Document is to add the necessary detail to the current project description to represent a suitable model for coding. This document is also intended to help detect contradictions prior to coding, and can be used as a reference manual for how the modules interact at a high level.

The HLD will:

* + - Present all of the design aspects and define them in detail
    - Describe the user interface being implemented
    - Describe software interfaces
    - Describe the performance requirements
    - Include design features and the architecture of the project
    - List and describe the non-functional attributes like: o Security
      * Reliability
      * Maintainability
      * Portability
      * Reusability
      * Application compatibility
      * Resource utilization
      * Serviceability

##### **Scope**

The HLD documentation presents the structure of the system, such as the database architecture, application architecture (layers), application flow (Navigation), and technology architecture. The HLD uses non-technical to mildly-technical terms which should be understandable to the administrators of the system.

* 1. **Definitions**



*Term*

*Database*

*IDE*

*Description*

Collection of all the information monitored by this system

Integrated Development Environment

### General Description

#### Product Perspective

The Email Spam Detection system is a deep learning-based object detection model which will help us to detect the spam emails and filter them from the Valid Emails..

#### 2.2 Problem Statement

To create a Machine Learning model for filtering the spam emails from the valid mails.

* + - To detect spam emails.

### **2.3 Proposed solution**

The proposed solution entails the deployment of an Email Spam Detection System utilizing the Multinomial Naive Bayes (MNB) classifier from the scikit-learn library. This system is designed to address the escalating issue of spam emails, which pose significant threats to user privacy and online security. The MNB classifier, known for its efficacy in text classification, is employed to analyze email content and accurately categorize messages as either spam or legitimate.

The workflow involves extracting relevant features from email content, allowing the MNB model to learn distinctive patterns associated with spam. By leveraging the scikit-learn library's implementation of the MNB algorithm, the system can efficiently process and classify emails. The outcome is a streamlined email experience for users, with the system effectively filtering out spam and mitigating the risks posed by phishing attempts, malware distribution, and unsolicited communications.

* 1. FURTHER IMPROVEMENTS

To enhance the Email Spam Detection System, several avenues for improvement can be explored. Firstly, optimizing feature engineering techniques can contribute to heightened model accuracy by incorporating additional linguistic and contextual elements. Secondly, the implementation of ensemble learning strategies, such as combining multiple classifiers, may bolster the system's resilience. Real-time learning mechanisms can enable the system to dynamically adapt to emerging spam tactics. Integrating user feedback loops, supporting multiple languages, and improving visualization and reporting capabilities are additional measures for refinement. These enhancements aim to fortify the system's effectiveness, adaptability, and user satisfaction in combating evolving email-based cyber threats.

#### Technical Requirements

This document addresses the requirements for detecting the Spam emails and filtering them to keep the user safe from the potential spam emails.

* + - Algorithm: Use Multinomial Naive Bayes from scikit-learn for classification.
    - Feature Extraction: Efficient methods for linguistic and contextual analysis.
    - Real-Time Processing: Swift identification and filtering of spam emails.
    - Multilingual Support: Extension to accommodate diverse languages.

#### Data Requirements

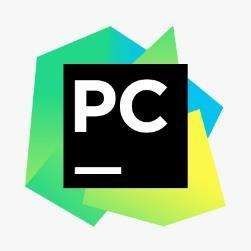
Data requirement completely depend on our problem statement.

* + - We need email message data that is balanced and must have at least 5000 rows.
  1. **Tools used**

Python programming language and frameworks such as NumPy, Pandas, Scikit-learn, and are used to build the whole model.







* + - VSCode is used as IDE.
    - For visualization of the plots, Matplotlib and Seaborn and are used.
    - Streamlit Cloud is used for deployment of the model.
    - CQL is used to retrieve, insert, delete, and update the database.
    - Front end development is done using Streamlit
    - GitHub is used as version control system.

#### Constraints

The Spam detection solution system must be user friendly, as automated as possible and users should not be required to know any of the workings.

#### Assumptions

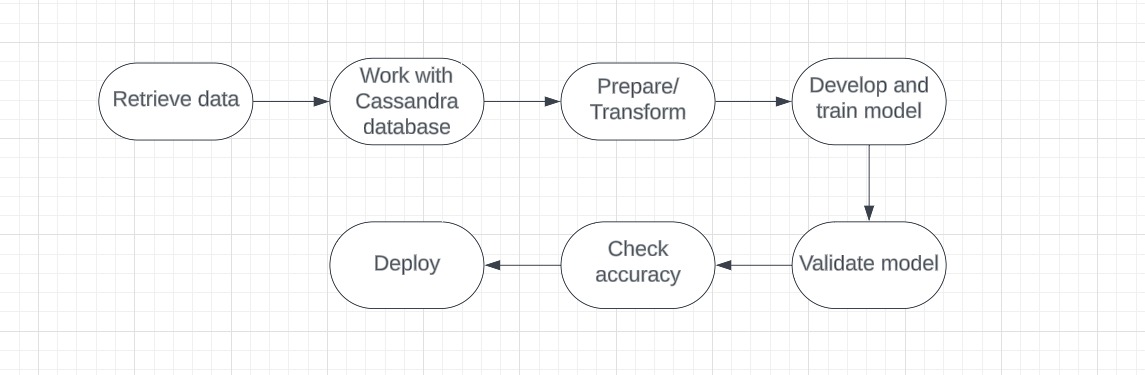
The main objective of the project is to implement the use cases as previously mentioned (2.2 Problem Statement) for new dataset that comes through UGV vehicle which has camera installed for capturing the live videos. Deep Learning based object detection model is used for detecting the above-mentioned use cases based on the input data. It is also assumed that all aspects of this project have the ability to work together in thewythe designer is expecting.

## **Design Details**

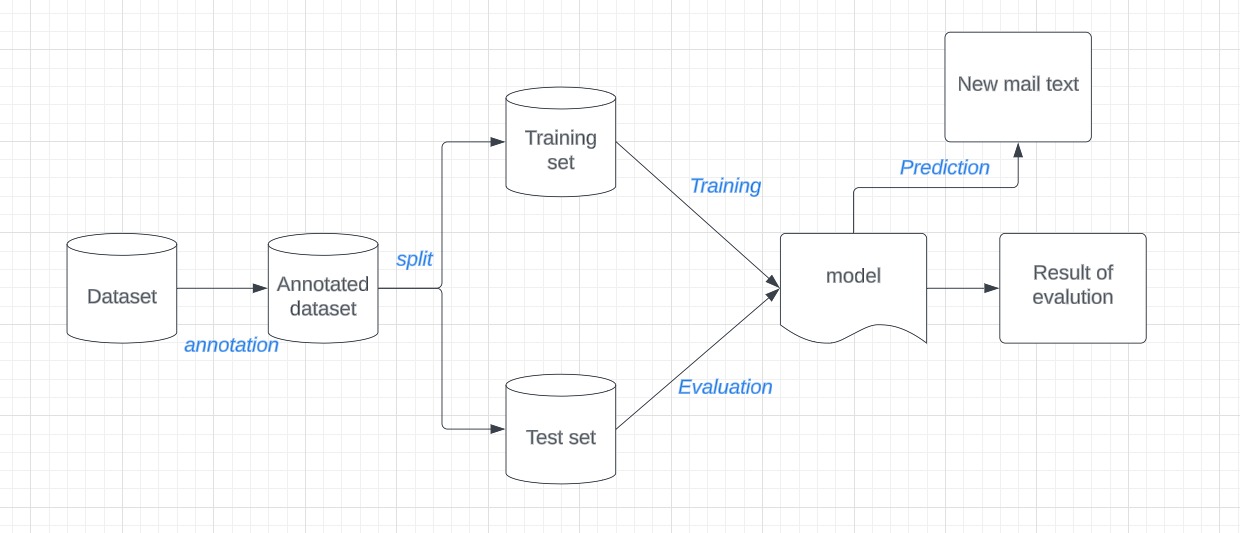
##### **Process Flow**

For identifying the spam emails, we will use a MultinomialNB model. Below is the process flow diagram is as shown below.

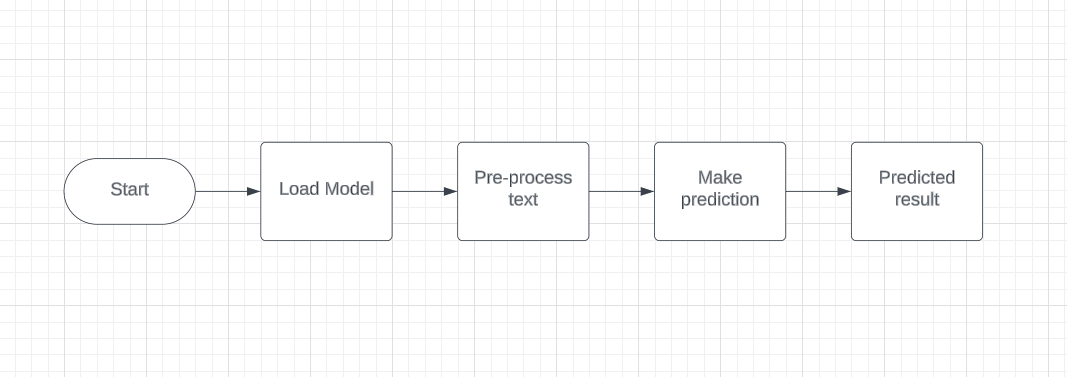
##### **Proposed methodology**



##### Model Training and Evaluation



##### Deployment Process



* 1. **Event log**

The system should log every event so that the user will know what process is running internally.

Initial Step-By-Step Description:

1. The System identifies at what step logging required
2. The System should be able to log each and every system flow.
3. Developer can choose logging method. You can choose database logging/ File logging as well.
4. System should not hang even after using so many loggings. Logging just because we can easily debug issues so logging is mandatory to do.
   1. **Error Handling**

Should errors be encountered, an explanation will be displayed as to what went wrong? An error will be defined as anything that falls outside the normal and intended usage.

1. **Performance**

The Email Spam Detection System boasts a high accuracy rate of 98.3%, ensuring precision in identifying and filtering spam emails. This accuracy is crucial for avoiding false alarms and providing users with a reliable and secure email experience. Ongoing model retraining remains a priority to uphold and enhance performance, adapting to emerging spam tactics and consistently delivering accurate results. This commitment to accuracy and adaptability underscores the system's effectiveness in maintaining the integrity of electronic communication.

#### Reusability

The code written and the components used should have the ability to be reused with no problems.

#### Application Compatibility

The different components for this project will be using Python as an interface between them. Each component will have its own task to perform, and it is the job of the Python to ensure proper transfer of information.

#### Resource Utilization

When any task is performed, it will likely use all the processing power available until that function is finished.

* 1. **Deployment**



#### KPls (Key Performance Indicators)

Accuracy: Accuracy measures the overall correctness of the model's predictions. It is calculated as the ratio of the number of correctly classified emails (both spam and non-spam) to the total number of emails in the dataset. Our measured accuracy is 98.83% .

Precision and Recall: Precision measures the proportion of correctly classified spam emails among all emails classified as spam. Recall measures the proportion of correctly classified spam emails among all actual spam emails in the dataset. Balancing precision and recall is crucial in spam detection, as it helps minimize false positives (non-spam emails classified as spam) and false negatives (spam emails classified as non-spam). Precision and recall in our model is 0.966 and 0.9162 respectively.

F1 Score: The F1 score is the harmonic mean of precision and recall. It provides a single score that balances both precision and recall, making it useful for comparing models and evaluating their overall performance. F1 score of our model is 0.9408.

Confusion Matrix: A confusion matrix provides a detailed breakdown of the model's predictions, showing the number of true positives, true negatives, false positives, and false negatives. It helps in understanding the distribution of prediction errors and identifying areas for improvement.

## 5 Conclusion

In conclusion, the Email Spam Detection System demonstrates exceptional accuracy of 98.83%