# EMAIL CLASSIFICATION USING SPAM DETECTION

# DISCRETE MATHEMATICS – PROJECT

# 

# Center for Computational Engineering and Networking

# AMRITA SCHOOL OF ARTIFICIAL INTELLIGENCE

# AMRITA VISHWA VIDYAPEETHAM

# COIMBATORE-941112 (INDIA)

# SUBMITTED BY:

# Group 6

Jayan S – 23033

K S Jagan – 23034

Keerthivasan S V– 23037

Krish S – 23040

# Table of Contents

## [1. Introduction](#_Introduction)

### [1.1 Literature Survey](#_1.1_Literature_Survey)

### [1.2 Problem Statement](#_1.2_Problem_Statement)

### [1.3 Objectives](#_1.3_Objectives)

### 1.4 Organization of the Report

## 2. Background of the project

## 2.1 Spam and its impacts

## 2.2 Impact of spams

## 2.3 Existing spam classification techniques

## [3. Methodology](#_3._Methodology)

### [3.1 Spam and non-spam email reasons](#_3.1_Spam_and)

### [3.2 Characteristics of Spam Messages](#_3.2_Characteristics_of)

### [3.3 Naïve Bayes Classification Scheme](#_3.3_Naive_Bayes)

### [3.4 Research Analysis](#_3.4_Research_Analysis)

## [4. Dataset Description and Evaluation Metrics](#_4._Dataset_Description)

## [5. Results and Discussions](#_5._Results_and)

## [6. Code](#_6._Code)

### [6.1 Preload File](#_Preload_File)

### [6.2 Run/Detection File](#_Run/Detection_File)

### 7. Literature Review

# Introduction

The ubiquity of email communication in our daily lives has significantly improved information exchange. However, this convenience comes with the persistent challenge of email spam. Beyond cluttering inboxes, spam emails pose security risks and can serve as a conduit for various cyber threats. To address this issue, the application of machine learning algorithms, particularly the Naive Bayes theorem, has emerged as a promising approach for email spam detection.

## 1.1 Literature Survey

A thorough examination of existing literature reveals the evolution of spam detection techniques and the pivotal role played by Naive Bayes in this domain. Past studies have explored diverse machine learning approaches, shedding light on the strengths and limitations of each method. The Naive Bayes algorithm, renowned for its simplicity and efficacy in classification tasks, has shown promise in discerning spam emails.

## 1.2 Problem Statement

The sophistication of spam emails and their ability to mimic legitimate communication present a significant challenge to traditional spam filters. Rule-based filters often struggle to adapt to the dynamic nature of spam, resulting in false positives or negatives. This research addresses the need for a more resilient and adaptive spam detection system, leveraging the Naive Bayes theorem to enhance accuracy and efficiency.

## 1.3 Objectives

This research aims to achieve the following objectives:

* Develop a comprehensive understanding of the Naive Bayes algorithm and its application in email spam detection.
* Investigate and analyze the characteristics of spam emails to enhance feature selection for the Naive Bayes model.
* Design and implement a robust spam detection system based on the Naive Bayes theorem.
* Evaluate the performance of the proposed system using real-world email datasets.
* Compare the Naive Bayes approach's effectiveness with other commonly used machine learning algorithms in spam detection.

## 1.4 Organization of the Report

1. **Introduction**
   * An overview of the project, its context, and its significance.
2. **Literature Survey**
   * A review of previous studies and research related to email spam detection and Naive Bayes classification.
3. **Problem Statement**
   * A clear and concise definition of the problem that this project aims to solve.
4. **Objectives**
   * The specific goals and outcomes that the project aims to achieve.
5. **Organization of the Thesis**
   * An outline of the structure and content of the rest of the thesis.
6. **Background of the project**
   * Detailed context and background information relevant to the project.
7. **Methodology**
   * A description of the methods and approaches used in the project.
8. **Spam and non-spam email reasons**
   * An exploration of the characteristics that differentiate spam emails from non-spam emails.
9. **Characteristics of Spam Messages**
   * A detailed analysis of the common features and patterns found in spam messages.
10. **Naïve Bayes Classification Scheme**
    * An explanation of how the Naive Bayes classification algorithm is used in the project.
11. **Research Analysis**
    * A presentation and interpretation of the data and results obtained from the project.
12. **Dataset Description and Evaluation Metrics**
    * A description of the dataset used in the project and the metrics used to evaluate the classification model's performance.
13. **Results and Discussions**
    * A presentation of the project's results and a discussion of their implications and significance.

# 2. Background of the project

2.1 Spam and its impact:

Spam refers to irrelevant or inappropriate email messages sent in bulk to many recipients for fraudulent purposes.

2.2 Impact of Spams:

Unsolicited commercial emails (UCE): These are typically advertisements for products, services, or offers that the recipient did not sign up for or express any interest in.

Phishing emails: These emails attempt to trick recipients into revealing sensitive information like usernames, passwords, and financial details by posing as legitimate org or individuals.

Malware distribution: Some spam emails contain attachments or links that, when clicked or opened, can download, and install malicious software on the recipient's device.

Nigerian Prince frauds: These are a classic example of spam, where the sender claims to be a wealthy individual in dire need of help and offers a significant reward in return for a financial contribution.

2.3 Existing Spam Classification techniques:

Classifying here on basis 2 categories:

Machine learning based – Naïve bayes algorithm

Rule based – Whitelist and Blacklist

Naïve Bayes algorithm:

  Naïve Bayes is a wonderful machine learning algorithm that has been applied in email spam filtering. A Naive Bayes (NB) classifier simply applies Bayes' theorem on the context classification of each email, with a strong assumption that the words included in the email are independent of each other. NB is desirable for email spam filtering because of its simplicity, ease of implementation, and quick convergence compared to conditional models such as logistic regression. It needs less training data. NB can be used to solve both classification problems involving two or more classes. It can be used to make forecasting that is subject to or involving probability variation. They can effectively manage continuous and discrete data.

NB needs little training time or speedy assessment to detect and filter email spam. NB filters need training that can be offered by the earlier set of non-spam and spam messages. It keeps a record of the changes that take place in each word that occurs in legitimate, illegitimate messages, and in both. NB can be applied to spam messages in diverse datasets having unique features and attribute

Whitelist and Blacklist:

Whitelist is a list of trusted email addresses from which emails are accepted.

Whitelisting very effectively blocks untrusted sources and provides superior protection against malware and attacks. Whitelisting restricts access strictly to already known and trusted sources such as existing and approved apps, users, websites, and IP addresses

A blacklist is a list of email addresses from which emails are marked as spam.

Blacklisting proactively blocks malicious sources and can be accomplished without great technical effort.

# 3. Methodology

## 3.1 Spam and non-spam email reasons

Spam email, also known as junk email, refers to unsolicited messages sent in bulk. These messages are usually marketing in nature and can reach millions of people every day. On the other hand, spam emails are those in which the recipient has given verifiable permission to send the message. Spam emails can have potentially serious consequences because they contain suspicious links or attachments and are often a vector for other serious attacks such as phishing, malware and some common types of spam include carrying advertisements and services for, chain emails that prompt you to spread messages, Email proofing is available with spammers pretending to be someone you know, scams promising miracles, and money scams promising easy money

## 3.2 Characteristics of Spam Messages

Detecting spam can be difficult as spammers often update their methods and messages to fool potential victims. However, there are some common indicators of spam or phishing texts: they often come from an untraceable email address or phone number, contain suspicious links, contain obvious spelling errors, spread an exceptionally good offer, or ask for personal information these indicators Being aware of the topic can help you better distinguish between spam and non-spam email.

## 3.3 Naive Bayes Classification Scheme

Naive Bayes is a probabilistic algorithm based on the Bayes theorem, which calculates the posterior probability of a given class label. In email spam detection, the class label is spam or ham (not spam), and the attributes are words or tokens in the email content. Given the class label, the naive assumption is that the attributes are independent of each other. The algorithm can be trained on a labeled emails dataset, where the default probability of each category and the conditional probabilities of each item given each category are calculated from the data and then the algorithm can distribute an email in another by choosing the highest level next possible.

## 3.4 Research Analysis

In our project, we utilized the TF-IDF (Term Frequency-Inverse Document Frequency) method for text representation and the Naïve Bayes classifier for message classification method.

Feature extraction in the context of spam detection involves identifying the most frequently used words or phrases in messages. This process can help distinguish between spam and non-spam messages. In a typical spam detection algorithm, the features could be the frequency of words often found in spam messages, such as “free,” “win,” “prize,” etc. The algorithm would then classify a message as spam or non-spam based on these features.

This technique significantly improved the performance of our Naïve Bayes classifier, as it was able to understand the frequency of occurrence of words. The metrics were computed using a test set of emails that were not part of the training data, ensuring an unbiased evaluation of our project's performance.

# 4. Dataset Description and Evaluation Metrics

For our project, we obtained a common dataset, that was present in many GitHub repositories as well as Kaggle notebooks, a detail that expresses the quality of the dataset. The dataset was in a Comma Separated Value (CSV) file, containing approximately 5600 rows, and 2 columns. The first row contains the field headers, and the remaining rows contain the data itself. Each row contains a message in the first column, and a label in the second column, either “spam” or “ham (non-spam).”

Our evaluation metrics initially were the frequency of each word in both spam and ham messages, which was used in the employment of the probabilistic machine learning algorithm, the Naïve Bayes Classifier, which was based on Bayes Theorem of Probability. However, we also tried using the weight of each word in each message, corresponding to the significance of the word. This was achieved using the Term Frequency Inverse Document Frequency (TF-IDF) Vectorizer, an algorithm that determines the importance of a word in a document/text.

Due to the similar nature of the Naïve Bayes Classifier and the TF-IDF Vectorizer, and their similar origin stemming from the frequency of words present in text and statistics calculated accordingly, we tested using two combinations. The first was only using the Naïve Bayes Classifier, and the second used both the Naïve Bayes Classifier and the TF-IDF Vectorizer. We found that using only the Naïve Bayer Classifier, which is the primary algorithm used in such applications, had a higher accuracy compared to the second combination of both algorithms.

As such, we reverted to using only the Naïve Bayes Classifier for our evaluation.

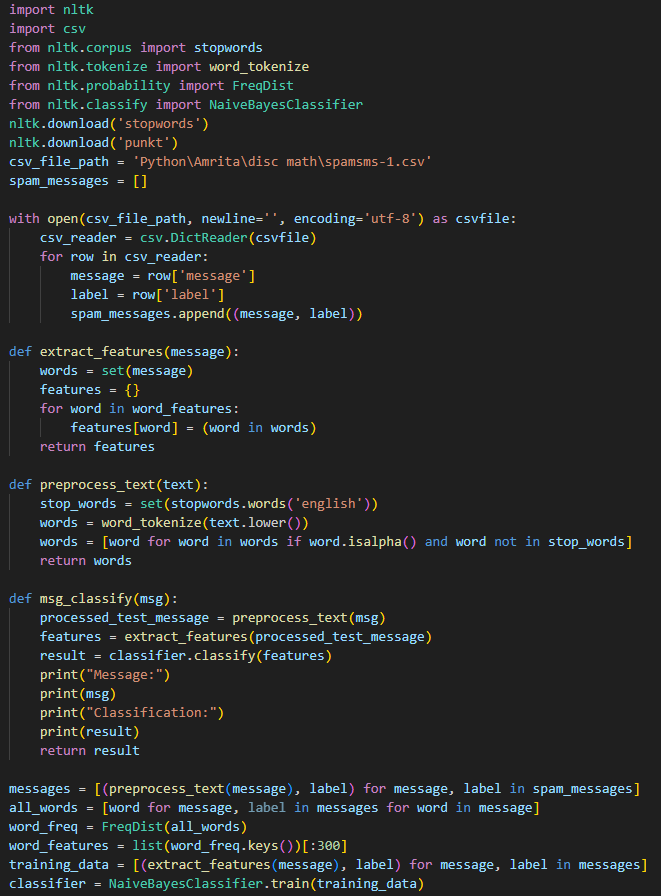
# 5. Results and Discussions

Upon entering diverse types of messages, both spam and ham ones, of distinctive styles and structures, our program was able to accurately classify the message as spam or ham. Immediately after running the program, the required external data is downloaded in case it is not already available on the device, then a window is generated, in which the user can enter messages and the classification result will be returned if the “Submit” button is pressed. The window will stay open, and the program will continue to run, until the user closes the window, after which the program will terminate automatically.

During our initial testing of the combination of both algorithms, our results showed us that using two algorithms that operate on the same features may create inconsistencies and contradictions which negatively impact our results. For example, a spam message such as “You have won a free prize, click the link below to claim now!” was detected correctly when using only the Naïve Bayes Classifier, but was detected incorrectly when using both algorithms, due to the algorithms having an overlap in their input features, which resulted in an incorrect output.

# 6. Code

## Preload File



## Run/Detection File

A computer screen shot of text

Description automatically generated

## 7. Literature Review

* [Kaggle: Spam/Ham detection using Naive Bayes Classifier](https://www.kaggle.com/code/dilip990/spam-ham-detection-using-naive-bayes-classifier): This Kaggle notebook demonstrates the application of the Naive Bayes Classifier for spam/ham detection. [The classifier is trained on a dataset containing various subjects of emails classified as either Spam or Ham](https://www.kaggle.com/code/dilip990/spam-ham-detection-using-naive-bayes-classifier).
* [GitHub: nikhilkr29/Email-Spam-Classifier-using-Naive-Bayes](https://github.com/nikhilkr29/Email-Spam-Classifier-using-Naive-Bayes): This GitHub repository contains a Naive Bayes spam/ham classifier based on Bayes’ Theorem. [The classifier is trained on a set of email subjects and then used to predict whether a previously unseen email subject is Spam or Ham.](https://www.kaggle.com/code/dilip990/spam-ham-detection-using-naive-bayes-classifier)
* [GitHub: stuntgoat/Generic-Naive-Bayesian-Classifier](https://github.com/stuntgoat/Generic-Naive-Bayesian-Classifier/blob/master/naive_bayes_db.py)[: This GitHub repository contains code for a generic Naive Bayes classifier.](https://www.kaggle.com/code/dilip990/spam-ham-detection-using-naive-bayes-classifier)
* [GitHub: vishal-verma27/How-To-Build-Email-Spam-Classification-Model-With-99-Accuracy](https://github.com/vishal-verma27/How-To-Build-Email-Spam-Classification-Model-With-99-Accuracy)[: This GitHub repository provides a guide on how to build an email spam classification model with 99% accuracy using the Naive Bayes algorithm.](https://www.kaggle.com/code/dilip990/spam-ham-detection-using-naive-bayes-classifier)
* [GeeksforGeeks: Naive Bayes Classifiers](https://www.geeksforgeeks.org/naive-bayes-classifiers/): This article explores Naive Bayes classifiers, a family of algorithms based on Bayes’ Theorem. [Despite the “naive” assumption of feature independence, these classifiers are widely utilized for their simplicity and efficiency in machine learning.](https://www.kaggle.com/code/dilip990/spam-ham-detection-using-naive-bayes-classifier)
* [Analytics Vidhya: Naive Bayes Explained](https://www.analyticsvidhya.com/blog/2017/09/naive-bayes-explained/)[: This article provides a comprehensive explanation of the Naive Bayes algorithm, its applications, pros, and cons, and how to implement the Naive Bayes Classifier in R and Python.](https://www.kaggle.com/code/dilip990/spam-ham-detection-using-naive-bayes-classifier)
* [IBM: What are Naive Bayes classifiers?](https://www.ibm.com/topics/naive-bayes): This IBM article explains what Naive Bayes classifiers are, how they work, and why they are considered “naive.” [It also discusses the assumptions made by these classifiers.](https://www.kaggle.com/code/dilip990/spam-ham-detection-using-naive-bayes-classifier)
* [Scikit-learn: Naive Bayes: This documentation from Scikit-learn provides information about the Naive Bayes classifier for multivariate Bernoulli models.](https://scikit-learn.org/stable/modules/generated/sklearn.naive_bayes.BernoulliNB.html)
* [Stack Overflow: A simple explanation of Naive Bayes Classification](https://stackoverflow.com/questions/10059594/a-simple-explanation-of-naive-bayes-classification)[: This Stack Overflow thread provides a simple and easy-to-understand explanation of the Naive Bayes classification algorithm.](https://www.kaggle.com/code/dilip990/spam-ham-detection-using-naive-bayes-classifier)