CSE422 : Artificial Intelligence

Project Report

Project name: Spam Email Detection

Group No :14	
ID	**Name**
21301243	Md.Mezbha-Ul-Haq Fahim
21301309	Abrar Ahmed

Table of Contents

1. Introduction

In the digital age, where communication via email has become ubiquitous, the scourge of spam emails poses a significant challenge. Spam emails are unwanted messages that flood our inboxes, often containing advertisements, scams, or even harmful content like viruses. These emails make our lives harder by wasting our time, cluttering our inboxes, and sometimes even tricking us into giving away personal information or money.

To combat this problem, spam detection has become increasingly popular. Many email providers now offer built-in spam filters that automatically sort suspected spam messages into a separate folder. However, as spammers become more sophisticated, traditional spam detection methods are not always enough.

The primary aim of this project is to design and implement an AI-based system capable of accurately identifying and filtering out spam emails from legitimate ones. AI-powered spam detection systems can analyze large amounts of email data and learn to recognize patterns and characteristics that are common in spam messages. By continuously learning and adapting, these AI systems can keep up with the ever-changing tactics of spammers and provide more accurate and effective spam filtering.

The motivation behind this project lies in mitigating the adverse effects of spam emails, which encompass a wide range of messages, including advertisements, scams, phishing attempts, and malware-laden attachments. Financial motivation is one of the primary reasons for the spammers and it has been estimated that spammers earn around USD 3.5 million from spam every year

With the help of AI, spam detection is becoming smarter and more efficient. AI algorithms can quickly identify and block spam emails, reducing the burden on our inboxes and saving us time and frustration. As AI technology continues to advance, we can expect spam detection to become even more sophisticated, providing a safer and more pleasant email experience for everyone.

Keywords*: Spam Messages, Dataset, Numerical Format, Categorical Encoding, Model Selection, Gaussian Naive Bayes (GNB), Multinomial Naive Bayes (MNB), Logistic Regression (LR), Accuracy, Precision, Real-World Challenges, Comprehensive Analysis Techniques*

2. Dataset Description

The dataset consists of 5572 data points with two main features: 'Category' and 'Messages'. The 'Category' feature classifies the messages into two types: 'ham' and 'spam'. In this context, 'ham' refers to legitimate messages, while 'spam' refers to unsolicited or unwanted messages. The dataset primarily comprises 'ham' data, with 'ham' messages being more prevalent than 'spam' messages.

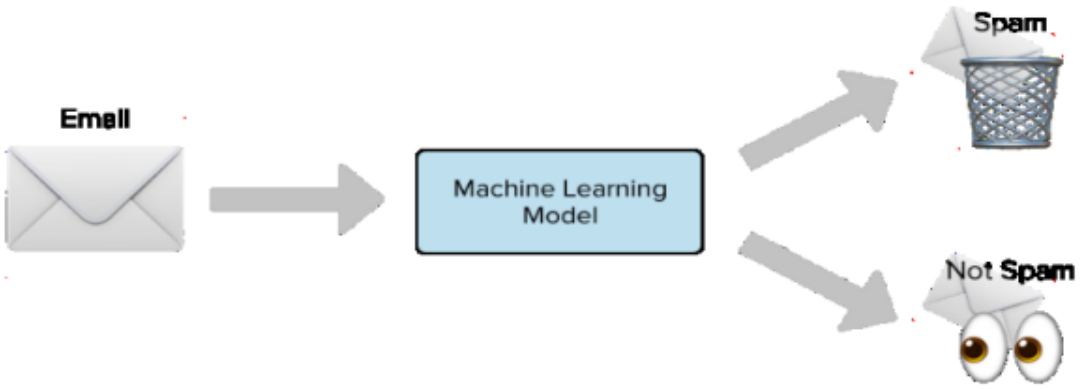
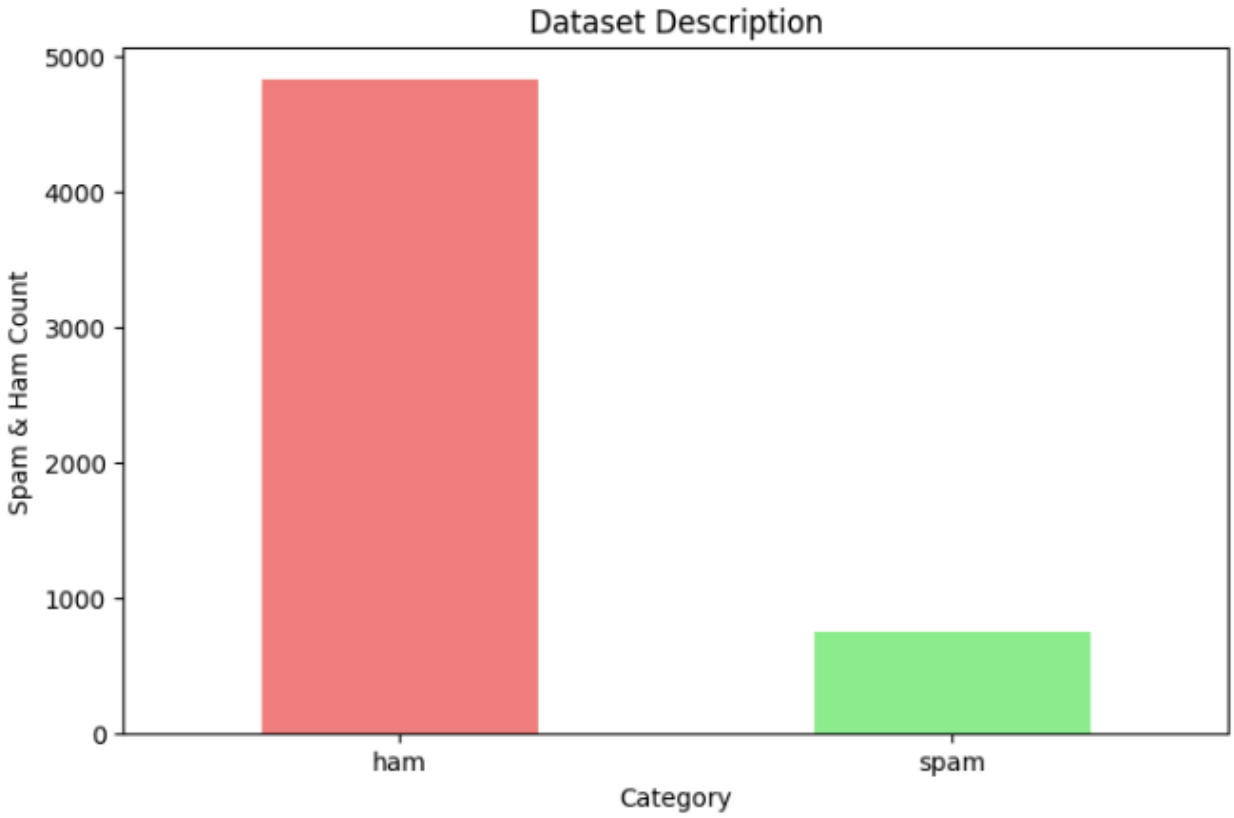


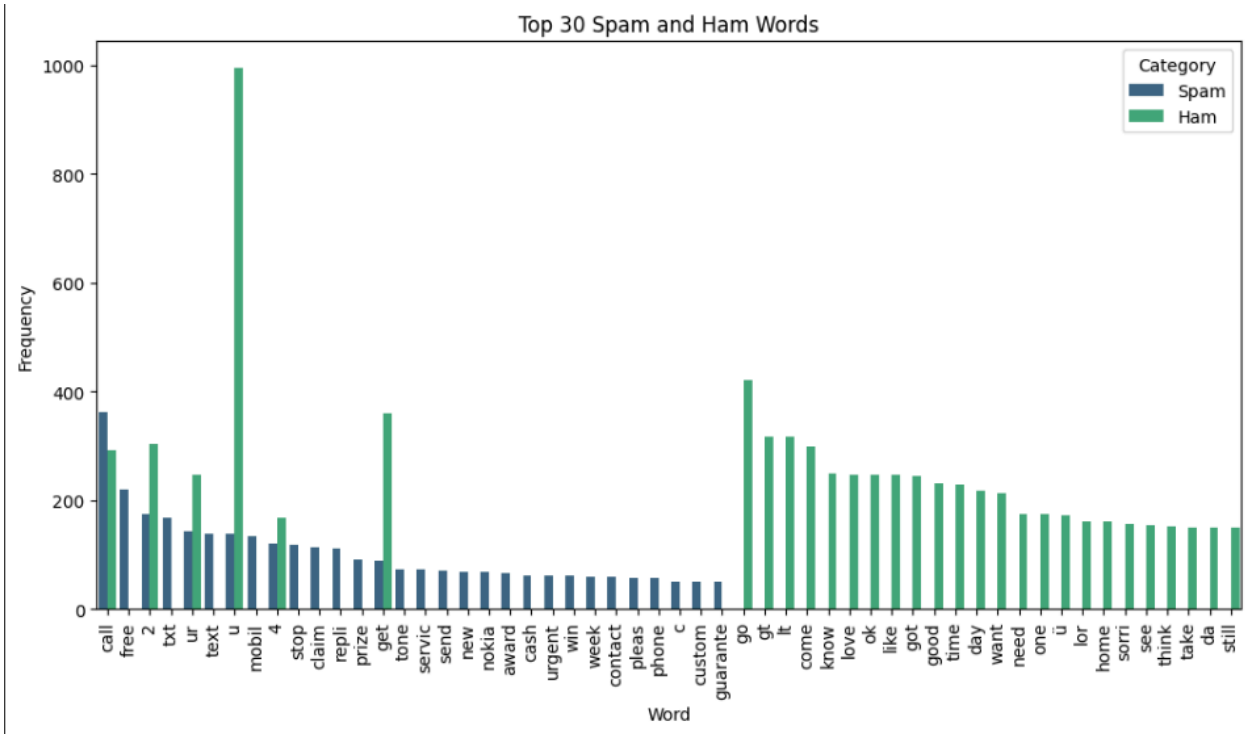
Fig.1. Classification into Spam and non-spam

There are no null values present in the dataset, ensuring data completeness and integrity. It's worth noting that this is a classification problem as the task involves categorizing messages into 'ham' and 'spam' categories, making it inherently a categorical classification task. If the task involved predicting numerical values associated with messages, it would be considered a regression problem.



The features in the dataset are primarily categorical. 'Category' is a categorical feature representing the type of message ('ham' or 'spam'). 'Messages' contain textual data that has been transformed into a numerical format for analysis, making it a special type of categorical feature tailored for text data.

The dataset has been preprocessed to prepare it for machine learning analysis. This preprocessing includes transforming the text messages into binary format suitable for modeling. Also text preprocessing involved converting text to lowercase, tokenization, removing special characters, stopwords, and punctuation, as well as stemming to reduce words to their root forms.



Dataset-Source: https://www.kaggle.com/code/mfaisalqureshi/email-spam-detection-98-accuracy/input

3. Dataset Pre- Processing

The dataset underwent preprocessing to prepare it for machine learning analysis, aiming to enhance accuracy by ensuring the data is in a suitable format for modeling. This preprocessing involved several steps:

A. Handling NULL Values:

The dataset was checked for NULL values, and fortunately, no NULL values were found. Therefore, there was no need for handling NULL values in this dataset.

B. Converting Categorical Values:

The 'Category' feature contained categorical values representing message types: 'ham' and 'spam'. To facilitate machine learning algorithms, these categorical values were

encoded into numerical values. 'ham' was encoded as 1, and 'spam' was encoded as 0, ensuring compatibility with classification algorithms.

C. Text Data Transformation:

The 'Messages' feature contained textual data, which needed to be transformed into a numerical format for modeling. This transformation involved several steps:

- Lowercasing: All text was converted to lowercase to ensure consistency.
- Tokenization: Text was tokenized into individual words to break down sentences into meaningful units.
- Removing Special Characters: Special characters were removed to focus on the textual content's semantic meaning.
- Removing Stopwords & Punctuation: Stopwords (commonly used words like 'the', 'is', 'and', etc.) and punctuation marks were removed as they do not contribute significantly to the message's classification.
- Stemming:Words were stemmed to reduce them to their root forms, which helps in capturing the essence of the words while reducing feature space.

The preprocessing steps ensured that the dataset was appropriately formatted for machine learning analysis, facilitating accurate model training and classification of messages into 'ham' and 'spam' categories.

4. Feature Scaling Description:

In the dataset preprocessing phase, feature scaling was applied to ensure that all features are on a similar scale, which is important for many machine learning algorithms, especially those that rely on distance calculations or gradient descent optimization.

- **Transformed Messages:**

  After preprocessing the textual data in the 'Messages' column into 'Transformed_Messages' through lowercase conversion, tokenization, removal of special characters, stopwords, punctuation, and stemming, the resulting transformed messages are in a numerical format suitable for analysis. However, they may still vary significantly in scale depending on the length of the messages and the frequency of words.

- **Categorical Encoding:**

  The 'Category' feature was encoded into numerical values where 'ham' was encoded as 1 and 'spam' was encoded as 0. This conversion ensures that the categorical feature is represented in a binary format, facilitating compatibility with machine learning algorithms.

5. Dataset Splitting Description:

In the dataset splitting phase, the data was divided into training and testing sets to assess the model's performance on unseen data. This process helps evaluate the model's generalization ability and prevents overfitting.

“train_test_split” Function:

The train_test_split function from the scikit-learn library was used to split the dataset into training and testing sets. This function randomly shuffles and splits the data into two subsets based on the specified proportion, typically a percentage of the data reserved for testing.

Splitting Procedure:

The dataset was split into four sets:

- x_train: Training data features (messages) used to train the model.
- x_test: Testing data features (messages) used to evaluate the trained model.
- y_train: Training data labels (categories) corresponding to x_train.
- y_test: Testing data labels (categories) corresponding to x_test.

6. Model Training and Testing Description:

In the model training and testing phase, several classification algorithms were trained on the training data and evaluated on the testing data to assess their performance in categorizing messages into 'ham' and 'spam' categories.

Models Used:

- **Gaussian Naive Bayes (GNB):**

  Gaussian Naive Bayes (GNB) is a classification algorithm that uses Bayes' theorem and assumes features follow a Gaussian (normal) distribution. The algorithm involves calculating class priors, class-conditional probabilities (means and standard deviations), and then computing the posterior probabilities for each class. Predictions are made based on the highest posterior probability. GNB is simple, efficient, and works well for continuous features, even though it assumes feature independence. It can be evaluated using various metrics, like accuracy and F1-score.

- **Multinomial Naive Bayes (MNB):**

  Multinomial Naive Bayes (MNB) is a classification algorithm for features representing counts or frequencies. The algorithm calculates class priors, class-conditional probabilities, and then computes posterior probabilities. Predictions are based on the highest posterior probability. MNB is particularly effective for text classification tasks, such as spam detection and document categorization. It's simple, efficient, and can be evaluated using various metrics, like accuracy and F1-score.

- **Logistic Regression (LR):**

  Logistic Regression (LR) is a linear classification algorithm for binary outcomes. The dataset should be prepared with numerical features. The algorithm fits a logistic function to the training data, calculates probabilities, and optimizes parameters using techniques like gradient descent. It predicts probabilities for new instances and assigns them to the class with the highest probability. LR can be evaluated using various metrics like accuracy and F1-score. It's widely used for binary classification tasks and works well when the relationship between features and the target variable is linear.

7. Model selection/Comparison analysis:

After rigorously training and testing multiple classification models on the dataset, including Gaussian Naive Bayes (GNB), Multinomial Naive Bayes (MNB), and Logistic Regression (LR), a comprehensive analysis was conducted to select the most suitable

model for categorizing messages into 'ham' and 'spam' categories. The evaluation primarily focused on two key metrics: accuracy and precision.

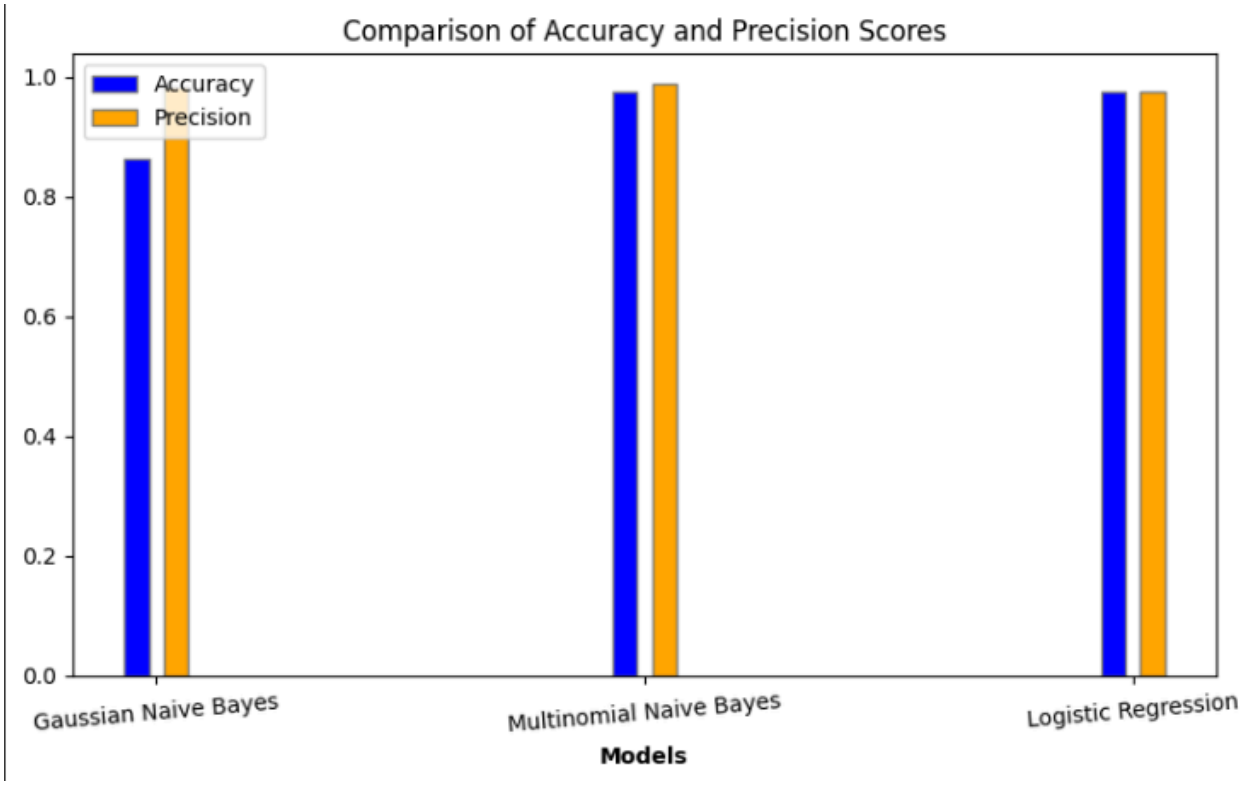
Precision measures the proportion of true positive predictions out of all positive predictions made by the model. It indicates the model's ability to correctly identify positive instances without misclassifying negative instances as positive.

Accuracy measures the proportion of correctly classified instances out of the total instances. It indicates the overall correctness of the model's predictions.

$$\text{Precision} = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$\text{Recall} = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Among the models tested, Multinomial Naive Bayes (MNB) emerged as the top performer, boasting the highest accuracy and precision scores. This indicates that MNB excels in accurately categorizing messages while minimizing false positives and false negatives.



Gaussian Naive Bayes (GNB):

Accuracy Score: 0.8636363636363636
Precision Score: 0.9829589465530596

Multinomial Naive Bayes (MNB):

Accuracy Score: 0.9742822966507177
Precision Score: 0.9890710382513661

Logistic Regression (LR):

Accuracy Score: 0.9766746411483254
Precision Score: 0.9761273209549072

8. Conclusion:

This project set out to solve a common problem we all face: dealing with spam messages.In this project, the primary objective was to develop a robust classification model capable of accurately distinguishing between legitimate ('ham') and unsolicited ('spam') messages. The dataset, comprising 5572 data points, presented a classificational challenge that required careful preprocessing and model selection to achieve optimal performance. Through meticulous data preprocessing, including text transformation to convert messages into a numerical format and categorical encoding of message types, the dataset was meticulously prepared for model training and evaluation. The subsequent exploration of various classification algorithms, including Gaussian Naive Bayes (GNB), Multinomial Naive Bayes (MNB), and Logistic Regression (LR), aimed to identify the most effective model for the task at hand. By prioritizing accuracy and precision as key evaluation metrics, the project strived to deliver a reliable and efficient solution for message categorization. The comprehensive analysis conducted throughout the project underscored the significance of leveraging appropriate techniques and methodologies to address real-world classification challenges effectively. Ultimately, the project's culmination in selecting Multinomial Naive Bayes (MNB) as the optimal model reflects a meticulous and systematic approach to achieving accurate and reliable results in classification tasks.