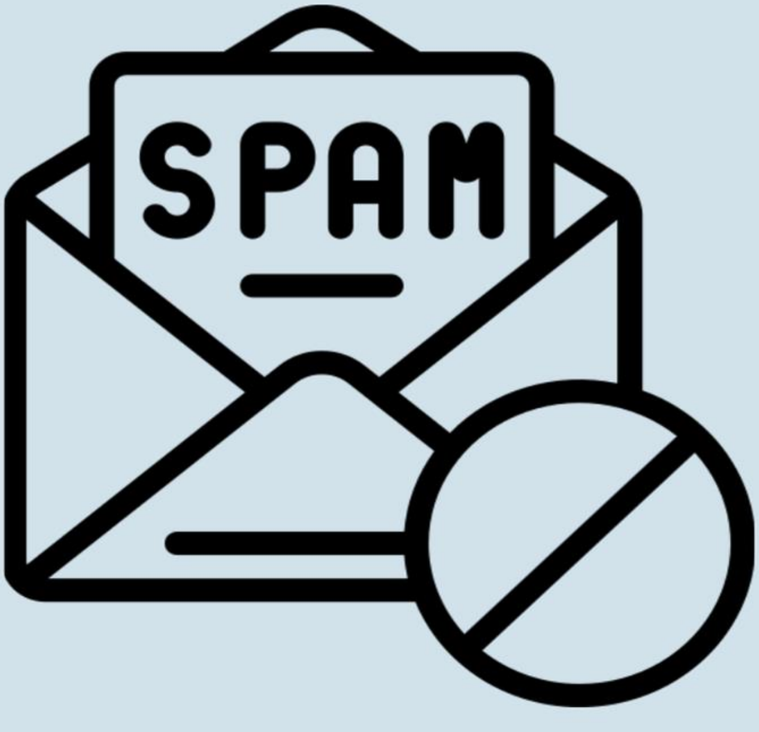
Egyptian E-Learning University

collage of computers and information technology

spam detection

social network using machine learning



Egyptian E-Learning University

College of Computing and Information Technology

spam detection social network using machine learning

Presented by:

Ahmed Ibrahem Abdullatif

Nourhan Ahmed Mohamed

Mostafa Mohamed Mounes
Selvia Emad Sobhy

Ahmed Tarek Abdelfatah

Ahmed Mahmoud Abdelsamea

Ahmed Abdelnaser Maria
Mohamed Ayman Abass

Supervised by:

Dr. Shimaa Musaad

ACKNOWLEDGMENT

We have taken efforts in this project. However, it would not have been possible without the kind support and help of many individuals and organizations. I would like to extend my sincere thanks to all of them.

We highly indebted to (Doctor Shimaa Musaad) for the guidance and constant supervision as well as for providing necessary information regarding the project & also for the support in completing the project.

I would like to express my gratitude to wards my parents & member of (Egyptian E-Learning University) for the kind co- operation and encouragement which help us in completion of this project.

I would like to express my special gratitude and thanks to supervisor`s persons for giving me such attention and time.

My thanks and appreciations also go to my colleagues in developing the project and people who have willingly help me out with their abilities.

Table of Contents

Abstract

Spam, consisting of unsolicited and often malicious messages, poses significant threats in the digital world. These messages can include phishing attempts and malware, leading to identity theft and financial loss, thus compromising personal security and privacy. Managing spam also drains system resources and degrades the performance of communication platforms.

Spam can contain phishing links and malware that steal personal and financial information. It inundates inboxes with unwanted messages, violating privacy and consuming system resources, ultimately eroding trust in communication platforms.

Detecting spam is crucial for protecting users from cyber threats, preserving their privacy, and optimizing system performance. Techniques like machine learning, natural language processing, and behavioral analysis help identify and filter spam, ensuring safer and more reliable digital communication.

Spam detection is vital for defending against digital threats and requires continuous development to counter increasingly sophisticated tactics. Advanced detection techniques help protect users and maintain the integrity of digital communications, fostering trust in digital platforms..

1

Introduction

1.1 Introduction

The topic focuses on developing an effective method to protect users from spam using machine learning. Spam, consisting of unsolicited and often malicious messages, has seen a significant increase globally, affecting email systems, social media, and messaging platforms. These messages often include phishing attempts, malware, and fraudulent schemes, posing severe risks to personal security and privacy.

In today's digital world, spam and cyber attacks are growing rapidly. Spam emails are not just annoying; they can also lead to serious cyber threats. As the amount of spam increases, so does its complexity, making it harder for old spam filters to work effectively.

Spam has changed from being a simple nuisance to a serious cyber threat. Cybercriminals use spam emails to spread malware, run phishing scams, and trick people. The number of spam emails is huge. A report by Symantec found that nearly 54% of all email traffic in 2021 was spam. This flood of unwanted emails gives cybercriminals many chances to trick people.

Cybersecurity attacks are also becoming more common, with spam playing a big role. The Verizon 2021 Data Breach Investigations Report showed that 36% of data breaches involved phishing, often starting with spam emails. These attacks can lead to big financial losses, stolen data, and damaged reputations.

Clicking on links in spam emails can be very dangerous. These links often lead to harmful websites designed to steal personal information, install malware, or trick people into giving away sensitive data. A study by IBM found that spam and phishing emails were the main ways ransomware attacks started. Ransomware can lock your data and demand money to unlock it. People who fall for these attacks can suffer from identity theft and financial loss.

Given the growing threat, strong spam detection is more important than ever to protect people from cyber threats. Traditional spam filters, which rely on simple rules, are not enough to handle advanced attacks that can trick these defenses.

Machine learning offers a powerful Techniques to tackle spam detection. By studying large amounts of data and learning from patterns, machine learning algorithms can spot spam accurately. Here are some key techniques used in this field:

- **Naive Bayes Classifier:** This simple model calculates the chance that an email is spam based on how often certain words appear.

- **Support Vector Machines (SVM):** SVMs classify emails by finding the best boundary between spam and non-spam messages. This method works well with large datasets.

- **Artificial Neural Networks (ANN):** ANNs work like the human brain to recognize complex patterns in email data. They can adapt to new types of spam by learning from new data.

- **Random Forests:** This technique uses multiple decision trees to improve accuracy. It is effective for spam detection and can handle complex data.

Using these machine learning techniques in spam detection systems greatly improves their ability to find and filter out harmful emails, protecting users and organizations from threats.

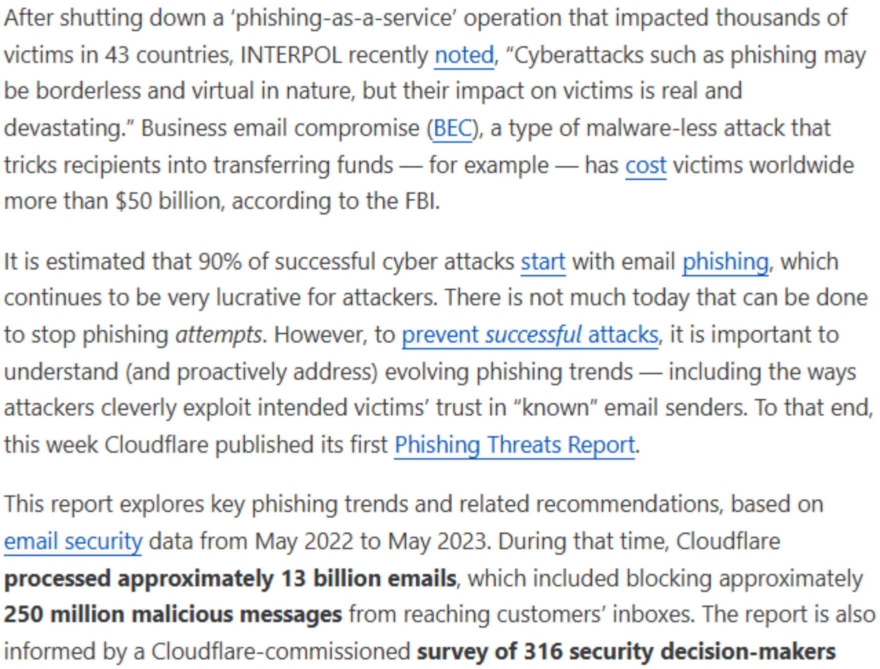
So in this project we are aiming to develop spam detection social network using machine learning

Figure 1.1

1.2 Motivation

Users are increasingly facing the problem of spam messages, especially on email and social media platforms, where it is challenging to filter them effectively. Our project provides a solution by utilizing machine learning to detect and prevent spam. Numerous global reports highlight the growing threat of spam. According to the "Global Spam and Phishing Report," spam poses significant security risks, including phishing and malware, making advanced detection methods essential.:

Figure 1.2



After shutting down a 'phishing-as-a-service' operation that impacted thousands of victims in 43 countries, INTERPOL recently noted, "Cyberattacks such as phishing may be borderless and virtual in nature, but their impact on victims is real and devastating." Business email compromise (BEC), a type of malware-less attack that tricks recipients into transferring funds — for example — has cost victims worldwide more than $50 billion, according to the FBI.

It is estimated that 90% of successful cyber attacks start with email phishing, which continues to be very lucrative for attackers. There is not much today that can be done to stop phishing *attempts*. However, to prevent *successful* attacks, it is important to understand (and proactively address) evolving phishing trends — including the ways attackers cleverly exploit intended victims' trust in "known" email senders. To that end, this week Cloudflare published its first Phishing Threats Report.

This report explores key phishing trends and related recommendations, based on email security data from May 2022 to May 2023. During that time, Cloudflare **processed approximately 13 billion emails**, which included blocking approximately **250 million malicious messages** from reaching customers' inboxes. The report is also informed by a Cloudflare-commissioned **survey of 316 security decision-makers**

Machine learning models for spam detection have some important limitations. Conclusions drawn from these models must be used with caution. While the patterns and associations discovered are real, they apply only to the datasets used and are not necessarily representative of all spam. Additionally, these models rely on the quality and diversity of the training data, and it is not clear how consistently comprehensive this data is across different platforms and regions.

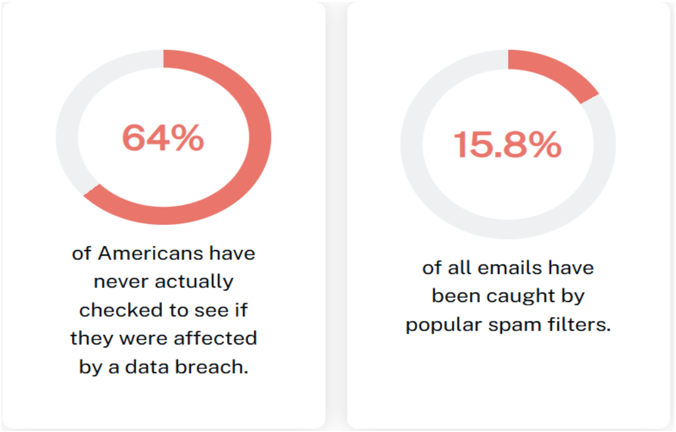


Figure 1.3

In 2023, nearly 45.6 percent of all e-mails worldwide were identified as spam, down from almost 49 percent in 2022. While remaining a big part of the e-mail traffic, since 2011, the share of spam e-mails has decreased significantly. In 2023, the highest volume of spam e-mails was registered in May, approximately 50 percent of e-mail traffic worldwide.

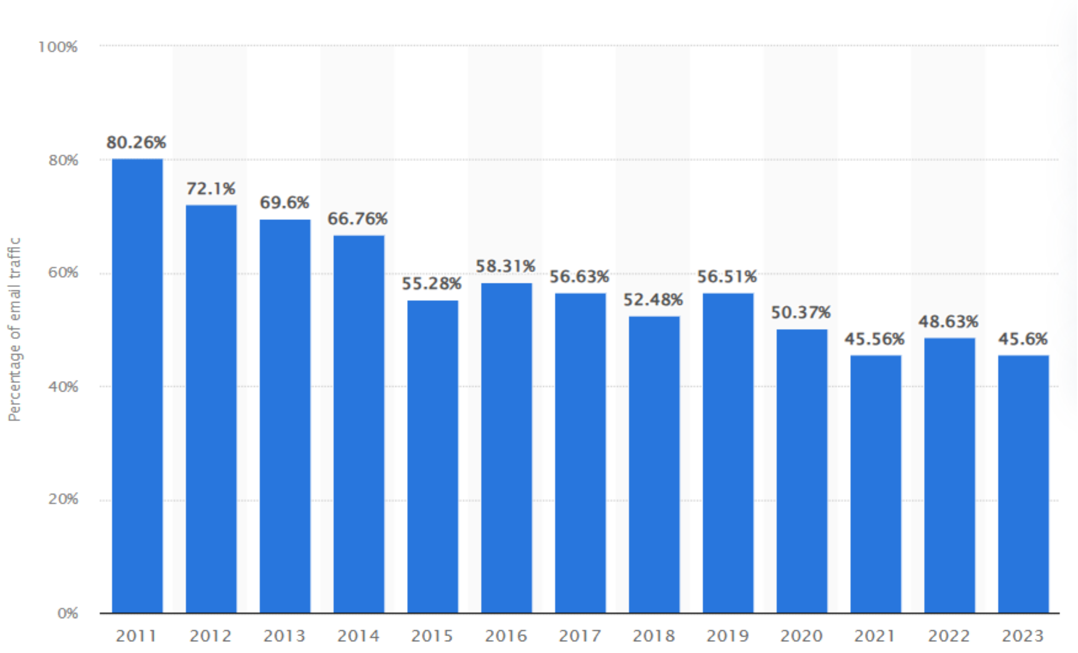


Figure 1.4

1.3 Problem Definition

The pervasive issue of email spam poses significant challenges in today's digital landscape. Users are inundated with unsolicited and often malicious messages, which not only clutter their inboxes but also expose them to various security risks. Phishing scams, malware distribution, and fraudulent schemes are prevalent within spam emails, jeopardizing users' personal security and privacy. Moreover, the sheer volume of spam messages overwhelms users, making it difficult to identify legitimate correspondence and increasing the likelihood of inadvertently engaging with harmful content. Traditional spam filtering methods have proven inadequate in effectively mitigating this problem, as spammers continually evolve their tactics to bypass detection mechanisms. As a result, users experience frustration, loss of productivity, and decreased trust in email communication platforms. Addressing the issue of email spam requires innovative solutions that leverage advanced technologies such as machine learning and artificial intelligence to accurately detect and filter out spam messages in real-time. By implementing robust spam detection systems, users can enjoy a safer and more secure email experience, free from the pervasive threat of spam.

1.4 Issues

- The sheer volume of spam emails overwhelms users, making it challenging to sift through legitimate correspondence.
- Users waste time and effort dealing with unwanted messages.
- Spam emails often contain phishing attempts, malware, and fraudulent schemes.
- Clicking on malicious links can compromise personal security and privacy.
- Sorting through spam detracts from the overall email experience.
- Users struggle to find important emails amidst the deluge of unwanted messages
- Striking the right balance between catching spam (without false positives) is essential.
- Missing legitimate emails due to aggressive filtering frustrates users

1.5 Objectives

- Ensure users can confidently use email services without fear of malicious or unwanted content.

- Protect against phishing attempts, malware, and fraudulent schemes.

- Minimize the risk of users inadvertently interacting with spam emails.

- Prevent security breaches resulting from clicking on malicious links.

- Streamline email communication by filtering out spam.

- Allow users to focus on legitimate correspondence without distraction.

- Achieve a balance between catching spam (without false positives) and avoiding false negatives (missing legitimate emails).

- Continuously update spam detection algorithms to handle evolving tactics.

- Stay ahead of new spam techniques.

.

2

Background

2.1 What is machine learning?

Machine learning is a common type of artificial intelligence. Learn more about this exciting technology, how it works, and the major types powering the services and applications we rely on every day.

Machine learning is a subfield of artificial intelligence that uses algorithms trained on data sets to create models that enable machines to perform tasks that would otherwise only be possible for humans, such as categorizing images, analyzing data, or predicting price fluctuations.

Today, machine learning is one of the most common forms of artificial intelligence and often powers many of the digital goods and services we use every day.

you'll learn more about what machine learning is, including how it works, different types of it, and how it's actually used in the real world. We'll take a look at the benefits and dangers that machine learning poses, and in the end, you'll find some cost-effective, flexible courses that can help you learn even more about machine learning

Machine learning is a subfield of artificial intelligence (AI) that uses algorithms trained on data sets to create self-learning models that are capable of predicting outcomes and classifying information without human intervention. Machine learning is used today for a wide range of commercial purposes, including suggesting products to consumers based on their past purchases, predicting stock market fluctuations, and translating text from one language to another.

Types of machine learning

Several different types of machine learning power the many different digital goods and services we use every day. While each of these different types attempts to accomplish similar goals – to create machines and applications that can act without human oversight – the precise methods they use differ somewhat to help us get a better idea of how these types differ from one another, here's an overview of the four different types of machine learning primarily in use today.

does not exist. In this undergraduate student design project, we employ both RFID and wireless sensor network technologies to build a wireless localization system in a children’s theme park. Our system design supports real-time detection of RFID tags and remote data collection through the underlying wireless sensor network. These capabilities are implemented with very low power consumption.

1. **Supervised machine learning**

In supervised machine learning, algorithms are trained on labeled data sets that include tags describing each piece of data. In other words, the algorithms are fed data that includes an “answer key” describing how the data should be interpreted. For example, an algorithm may be fed images of flowers that include tags for each flower type so that it will be able to identify the flower better again when fed a new photograph.

Supervised machine learning is often used to create machine learning models used for prediction and classification purposes.

2. **Unsupervised machine learning**

Unsupervised machine learning uses unlabeled data sets to train algorithms. In this process, the algorithm is fed data that doesn't include tags, which requires it to uncover patterns on its own without any outside guidance. For instance, an algorithm may be fed a large amount of unlabeled user data culled from a social media site in order to identify behavioral trends on the platform.

Unsupervised machine learning is often used by researchers and data scientists to identify patterns within large, unlabeled data sets quickly and efficiently.

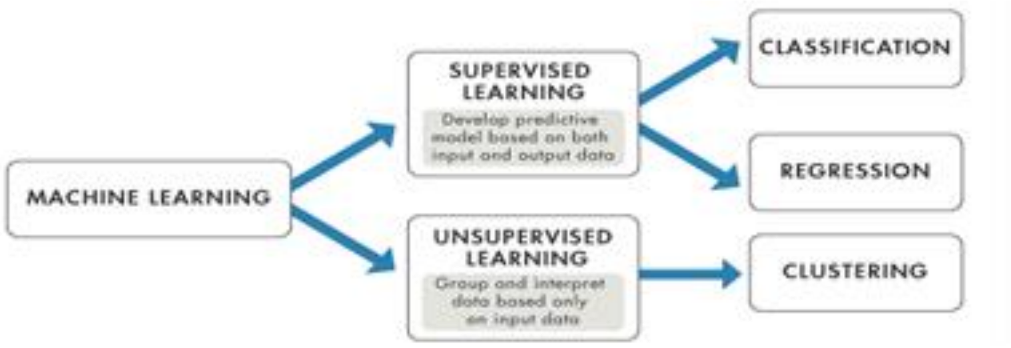

MACHINE LEARNING
SUPERVISED LEARNING
Develop predictive model based on both input and output data
UNSUPERVISED LEARNING
Group and interpret data based only on input data
CLASSIFICATION
REGRESSION
CLUSTERING

Figure 2.1

3. Semi-supervised machine learning

Semi-supervised machine learning uses both unlabeled and labeled data sets to train algorithms. Generally, during semi-supervised machine learning, algorithms are first fed a small amount of labeled data to help direct their development and then fed much larger quantities of unlabeled data to complete the model. For example, an algorithm may be fed a smaller quantity of labeled speech data and then trained on a much larger set of unlabeled speech data in order to create a machine learning model capable of speech recognition.

Semi-supervised machine learning is often employed to train algorithms for classification and prediction purposes in the event that large volumes of labeled data is unavailable.

4. Reinforcement learning

Reinforcement learning uses trial and error to train algorithms and create models. During the training process, algorithms operate in specific environments and then are provided with feedback following each outcome. Much like how a child learns, the algorithm slowly begins to acquire an understanding of its environment and begins to optimize actions to achieve particular outcomes. For instance, an algorithm may be optimized by playing successive games of chess, which allows it to learn from its past successes and failures playing each game.

Reinforcement learning is often used to create algorithms that must effectively make sequences of decisions or actions to achieve their aims, such as playing a game or summarizing an entire text.

Table2.1 Difference between Machine Learning and Traditional Programming

Machine Learning	**Traditional Programming**	**Artificial Intelligence**
Machine Learning is a subset of artificial intelligence(AI) that focus on learning from data to develop an algorithm that can be used to make a prediction.	In traditional programming, rule-based code is written by the developers depending on the problem statements.	Artificial Intelligence involves making the machine as much capable, So that it can perform the tasks that typically require human intelligence.
Machine Learning uses a data-driven approach, It is typically trained on historical data and then used to make predictions on new data.	Traditional programming is typically rule-based and deterministic. It hasn't self-learning features like Machine Learning and AI.	AI can involve many different techniques, including Machine Learning and Deep Learning, as well as traditional rule-based programming.
ML can find patterns and insights in large datasets that might be difficult for humans to discover.	Traditional programming is totally dependent on the intelligence of developers. So, it has very limited capability.	Sometimes AI uses a combination of both Data and Pre-defined rules, which gives it a great edge in solving complex tasks with good accuracy which seem impossible to humans.
Machine Learning is the subset of AI. And Now it is used in various AI-based tasks like Chatbot Question answering, self-driven car., etc.	Traditional programming is often used to build applications and software systems that have specific functionality.	AI is a broad field that includes many different applications, including natural language processing, computer vision, and robotics.

Different between Machine learning and AI and deep learning

Machine learning is often confused with artificial intelligence or deep learning. Let's take a look at how these terms differ from one another. For a more in-depth look, check out our comparison guides on AI vs machine learning and machine learning vs deep learning.

AI refers to the development of programs that behave intelligently and mimic human intelligence through a set of algorithms. The field focuses on three skills: learning, reasoning, and self-correction to obtain maximum efficiency. AI can refer to either machine learning-based programs or even explicitly programmed computer programs.

Machine learning is a subset of AI, which uses algorithms that learn from data to make predictions. These predictions can be generated through supervised learning, where algorithms learn patterns from existing data, or unsupervised learning, where they discover general patterns in data. ML models can predict numerical values based on historical data, categorize events as true or false, and cluster data points based on commonalities.

Deep learning, on the other hand, is a subfield of machine learning dealing with algorithms based essentially on multi-layered artificial neural networks (ANN) that are inspired by the structure of the human brain.

Unlike conventional machine learning algorithms, deep learning algorithms are less linear, more complex, and hierarchical, capable of learning from enormous amounts of data, and able to produce highly accurate results. Language translation, image recognition, and personalized medicines are some examples of deep learning applications.

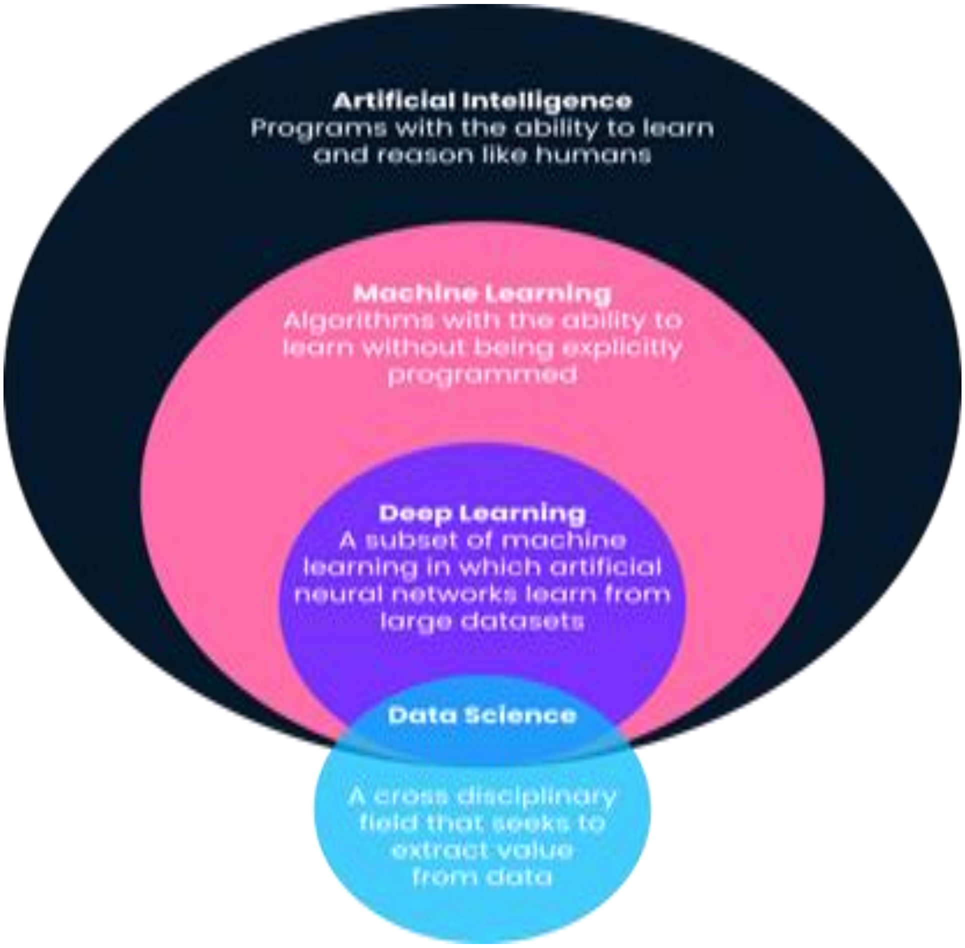

Artificial Intelligence
Programs with the ability to learn and reason like humans
Machine Learning
Algorithms with the ability to learn without being explicitly programmed
Deep Learning
A subset of machine learning in which artificial neural networks learn from large datasets
Data Science
A cross disciplinary field that seeks to extract value from data

Figure 2.2

2.2 How Does machine learning algorithms Work?

Machine Learning works in the following manner.

A machine learning algorithm works by learning patterns and relationships from data to make predictions or decisions without being explicitly programmed for each task. Here's a simplified overview of how a typical machine learning algorithm works:

1. Data Collection:

First, relevant data is collected or curated. This data could include examples, features, or attributes that are important for the task at hand, such as images, text, numerical data, etc.

2. Data Preprocessing:

Before feeding the data into the algorithm, it often needs to be preprocessed. This step may involve cleaning the data (handling missing values, outliers), transforming the data (normalization, scaling), and splitting it into training and test sets.

3. Choosing a Model:

Depending on the task (e.g., classification, regression, clustering), a suitable machine learning model is chosen. Examples include decision trees, neural networks, support vector machines, and more advanced models like deep learning architectures.

4. Training the Model:

The selected model is trained using the training data. During training, the algorithm learns patterns and relationships in the data. This involves adjusting model parameters iteratively to minimize the difference between predicted outputs and actual outputs (labels or targets) in the training data.

5. Evaluating the Model:

Once trained, the model is evaluated using the test data to assess its performance. Metrics such as accuracy, precision, recall, or mean squared error are used to evaluate how well the model generalizes to new, unseen data.

6. Fine-tuning:

Models may be fine-tuned by adjusting hyperparameters (parameters that are not directly learned during training, like learning rate or number of hidden layers in a neural network) to improve performance.

7. Prediction or Inference:

Finally, the trained model is used to make predictions or decisions on new data. This process involves applying the learned patterns to new inputs to generate outputs, such as class labels in classification tasks or numerical values in regression tasks

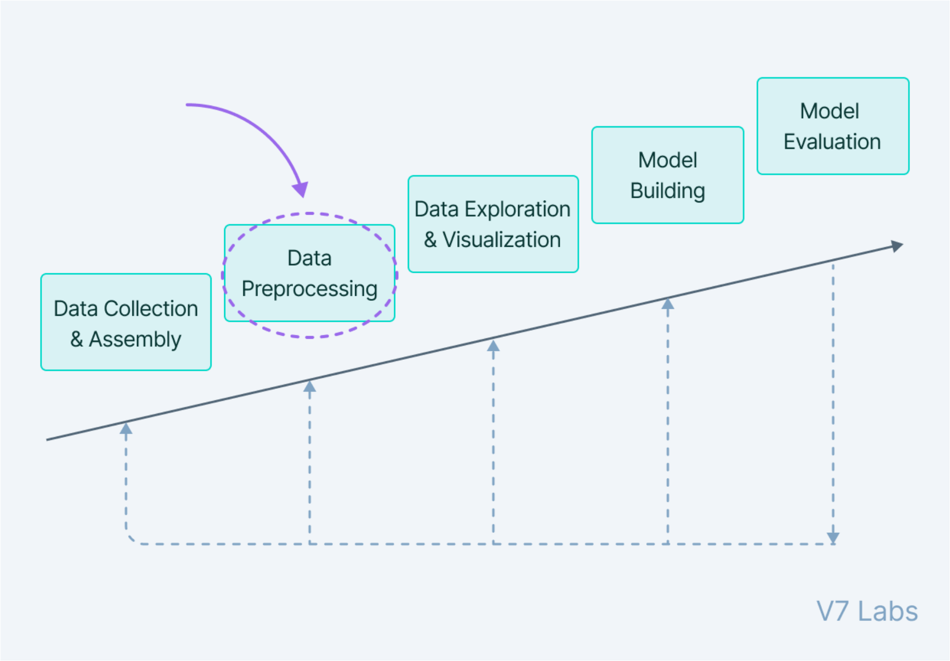


Figure 2.3

2.3 Machine learning life cycle

The lifecycle of a machine learning project involves a series of steps that include:

1. Study the Problems:

The first step is to study the problem. This step involves understanding the business problem and defining the objectives of the model.

2. Data Collection:

When the problem is well-defined, we can collect the relevant data required for the model. The data could come from various sources such as databases, APIs, or web scraping.

3. Data Preparation:

When our problem-related data is collected. then it is a good idea to check the data properly and make it in the desired format so that it can be used by the model to find the hidden patterns. This can be done in the following steps:

Data cleaning

Data Transformation

Explanatory Data Analysis and Feature Engineering

Split the dataset for training and testing.

4. Model Selection:

The next step is to select the appropriate machine learning algorithm that is suitable for our problem. This step requires knowledge of the strengths and weaknesses of different algorithms. Sometimes we use multiple models and compare their results and select the best model as per our requirements.

5. Model building and Training:

After selecting the algorithm, we have to build the model.

In the case of traditional machine learning building mode is easy it is just a few hyperparameter tunings.

In the case of deep learning, we have to define layer-wise architecture along with input and output size, number of nodes in each layer, loss function, gradient descent optimizer, etc.

After that model is trained using the pre-processed dataset.

6. Model Evaluation:

Once the model is trained, it can be evaluated on the test dataset to determine its accuracy and performance using different techniques. like classification report, F1 score, precision, recall, ROC Curve, Mean Square error, absolute error, etc.

7. Model Tuning:

Based on the evaluation results, the model may need to be tuned or optimized to improve its performance. This involves tweaking the hyperparameters of the model.

8. Deployment:

Once the model is trained and tuned, it can be deployed in a production environment to make predictions on new data. This step requires integrating the model into an existing software system or creating a new system for the model.

9. Monitoring and Maintenance:

Finally, it is essential to monitor the model’s performance in the production environment and perform maintenance tasks as required. This involves monitoring for data drift, retraining the model as needed, and updating the model as new data becomes available.

3

Algorithm

3.1 which algorithms we chose?

At first we chose support vector machine algorithm (SVM):

Using the Support Vector Machine (SVM) algorithm for spam detection offers several advantages**:**

1. **Effective Separation of Classes:**
    1. SVM aims to find the hyperplane that best separates different classes (spam vs. non-spam).
    2. It maximizes the margin between data points of different classes, leading to robust classification.

2. **Handling High-Dimensional Data:**

1. SVM performs well even in high-dimensional feature spaces (such as text data).
2. In spam detection, features can represent word frequencies, patterns, or other relevant information.

3. **Tolerance to Noise and Outliers:**

1. SVM is less sensitive to noisy data points or outliers.
2. This robustness is crucial when dealing with real-world email data.

And we chose Random forest algorithm also :

Using the Random Forest algorithm for spam detection offers several advantages:

1. **Ensemble Learning:**
    1. Random Forest combines multiple decision trees to create a robust model.
    2. Each tree learns from a random subset of features and data points, reducing overfitting.
2. **Feature Importance:**

1. Random Forest provides feature importance scores.
2. It helps identify which features (words, patterns) contribute most to spam detection.

3. **Non-Linear Decision Boundaries:**

1. Random Forest can handle complex relationships in data.
2. It’s effective even when spam features are non-linearly related.

So we should know some information about SVM and Random forest

3.2 What is SVM algorithm?

Support Vector Machines (SVMs) are a class of supervised learning algorithms that have demonstrated remarkable success in a wide range of classification and regression tasks. At the core of SVMs is the concept of finding the optimal hyperplane that maximizes the margin between two or more classes in a high-dimensional feature space.

The key principle of SVMs is to identify the most informative training examples, known as "support vectors," and use them to construct the optimal decision boundary. This is achieved by formulating an optimization problem that seeks to maximize the margin between the support vectors and the decision boundary. The use of kernel functions allows SVMs to efficiently handle non-linear problems by implicitly mapping the data into a higher-dimensional space.

SVMs possess several attractive properties that have contributed to their widespread success. They have a strong theoretical foundation, provide high generalization performance, and are relatively robust to overfitting. Additionally, SVMs can be adapted to handle a variety of task-specific requirements, such as multi-class classification, regression, and anomaly detection.

Despite their strengths, SVMs also face some challenges, such as the computational complexity involved in training large-scale models and the sensitivity to the selection of hyperparameters. Ongoing research in the field of SVMs focuses on addressing these limitations, as well as exploring new applications and extensions, such as in the domains of deep learning, online learning, and transfer learning.

I. Introduction to Support Vector Machines (SVMs)

A. Definition and Purpose:

1. Supervised learning algorithm.

- SVMs are a class of supervised learning algorithms used for both classification and regression tasks.

2. Used for classification and regression tasks.

- SVMs are particularly effective at solving complex, high-dimensional classification problems.
- They can also be extended to handle regression problems.

B. Historical Background:

1. Developed by Vladimir Vapnik and Alexey Chervonenkis in the 1960s.
- The theoretical foundations of SVMs were laid by Vladimir Vapnik and Alexey Chervonenkis in the 1960s.
- They introduced the concept of structural risk minimization, which forms the basis of SVM optimization.
2. Popularized in the 1990s.
- SVMs gained widespread popularity in the 1990s, largely due to the work of Corinna Cortes and Vladimir Vapnik.
- The development of efficient optimization algorithms, such as the Sequential Minimal Optimization (SMO) algorithm, contributed to the widespread adoption of SVMs**.**

II. Basic Concepts of SVMs

A. Hyperplane and Decision Boundary:

1. Definition of a hyperplane in feature space:
- In SVMs, the decision boundary is represented by a hyperplane in the feature space.
- A hyperplane is a dimensional flat subspace in a dimensional space.

2. Role of the decision boundary in classification:
- The hyperplane serves as the decision boundary, separating the feature space into different classes.
- The goal of SVMs is to find the optimal hyperplane that best separates the classes with the maximum margin.

B. Support Vectors:

1. Definition and significance:
- Support vectors are the training examples that lie closest to the decision boundary.
- These critical data points define the optimal hyperplane and determine the maximum-margin solution.

2. Determining the optimal hyperplane:
- The optimal hyperplane is the one that maximizes the margin between the support vectors and the decision boundary.
- The support vectors are the only training examples that influence the position and orientation of the optimal hyperplane.

III. Mathematical Formulation

A. Linear SVM

1. Objective function and constraints:
- The objective function of a linear SVM is to find the hyperplane that maximizes the margin between the two classes.
- This is formulated as a constrained optimization problem, where the goal is to minimize the norm of the weight vector subject to the correct classification of all training examples.

2. Lagrange multipliers and dual formulation:
- The SVM optimization problem can be solved using Lagrange multipliers, which leads to the dual formulation of the problem.
- The dual formulation allows the optimization to be performed in the dual space, which can be more computationally efficient.

B. Non-Linear SVM

1. Kernel trick introduction:
- To handle non-linear decision boundaries, SVMs use the "kernel trick" to implicitly map the data into a higher-dimensional feature space.
- This allows SVMs to find complex, non-linear decision boundaries without explicitly computing the mapping to the higher-dimensional space.
2. Common kernel functions (linear, polynomial, RBF, sigmoid):
- Popular kernel functions used in non-linear SVMs include the linear kernel, polynomial kernel, Radial Basis Function (RBF) kernel, and sigmoid kernel.
- The choice of kernel function depends on the characteristics of the data and the underlying problem.

IV. Training an SVM

A. Data Preparation

1. Feature scaling and normalization:
- Before training an SVM, it is important to preprocess the data by scaling and normalizing the features.
- This helps ensure that all features contribute equally to the optimization process and prevents numerical issues during training.

2. Handling imbalanced datasets:
- SVMs can be sensitive to class imbalance in the training data, where one class is

significantly underrepresented.
- Techniques like oversampling, under sampling, or adjusting the class weights can be used to mitigate the effects of class imbalance.

B. Parameter Selection

1. Regularization parameter (C):
- The regularization parameter, C, controls the trade-off between maximizing the margin and minimizing the training error.
- Tuning the value of C is crucial for achieving good generalization performance.

2. Kernel parameters (e.g., gamma in RBF kernel):
- For non-linear SVMs, the choice of kernel function and its associated parameters (e.g., gamma in the RBF kernel) can significantly impact the model's performance.

- Hyperparameter optimization techniques, such as grid search or cross-validation, are often used to find the optimal kernel parameters.

3.3 Advantages and Disadvantages of SVM algorithm

Advantages*:*

1. Effective in high-dimensional spaces

- SVMs are well-suited for problems with high-dimensional feature spaces, as they can effectively handle the increased complexity.

- The kernel trick allows SVMs to work in high-dimensional spaces without explicitly computing the feature mapping.

2. Robust to overfitting, especially in high-dimensional space

- SVMs are less prone to overfitting than many other machine learning models, particularly in high-dimensional spaces.

- The principle of maximum margin helps SVMs generalize well, even when the feature space is large.

3. Works well with clear margin of separation

- SVMs excel when there is a clear margin of separation between the classes in the feature space.

- The optimization process focuses on finding the optimal hyperplane that maximizes the margin between the classes.

B. Disadvantages

1. Inefficient with large datasets

- Training SVMs can be computationally expensive, especially for large datasets, due to the need to solve a quadratic programming problem.

- The time complexity of training an SVM scales quadratically with the number of training examples.

2. Choosing the right kernel can be complex

- Selecting the appropriate kernel function and tuning its parameters can be a challenging task, as it requires domain knowledge and extensive experimentation.

- The performance of non-linear SVMs is highly dependent on the choice of kernel.

VII. Comparison with Other Algorithms

A. Decision Trees and Random Forests

- Decision trees and random forests are more intuitive and easier to interpret compared to SVMs.

- Decision trees and random forests can handle both numerical and categorical features, while.

SVMs are generally more effective with numerical features.

- Random forests can often outperform SVMs on large datasets, as they are less computationally intensive.

- However, SVMs can be more accurate than decision trees and random forests when there is a clear margin of separation between classes.

B. Neural Networks

- Neural networks can be more flexible and expressive than SVMs, as they can learn complex non-linear decision boundaries.

- Neural networks can handle high-dimensional and noisy data better than SVMs, but they require more training data and can be more sensitive to hyperparameter tuning.

- SVMs tend to have better generalization performance, especially when the training data is limited or the problem is well-defined.

- Neural networks are generally more computationally intensive to train compared to SVMs.

C. Logistic Regression

- Logistic regression is a simpler and more interpretable model compared to SVMs, making it easier to understand the underlying relationships in the data.

- Logistic regression is better suited for linear decision boundaries, while SVMs can handle nonlinear problems more effectively.

- SVMs are generally more robust to outliers and can achieve higher accuracy in complex, high dimensional problems.

- Logistic regression is more efficient to train than SVMs, especially for large datasets.

So :

- SVMs are a powerful and versatile machine learning algorithm that can effectively tackle a wide range of classification and regression problems.

. - They work by finding the optimal hyperplane that maximizes the margin between the classes, which helps to improve the generalization performance of the model.

- SVMs can handle both linear and non-linear decision boundaries through the use of kernel functions, making them suitable for complex problems.

- SVMs are robust to high-dimensional feature spaces and can perform well even with a limited amount of training data.

3.4 What is Random Forest algorithm?

Random forest, a popular machine learning algorithm developed by Leo Breiman and Adele Cutler, merges the outputs of numerous decision trees to produce a single outcome. Its popularity stems from its user-friendliness and versatility, making it suitable for both classification and regression tasks.

Its widespread popularity stems from its user-friendly nature and adaptability, enabling it to tackle both classification and regression problems effectively. The algorithm's strength lies in its ability to handle complex datasets and mitigate overfitting, making it a valuable tool for various predictive tasks in machine learning.

One of the most important features of the Random Forest Algorithm is that it can handle the data set containing continuous variables, as in the case of regression, and categorical variables, as in the case of classification. It performs better for classification and regression tasks. In this tutorial, we will understand the working of random forest and implement random forest on a classification task.

2- Random Forest Applications

Customer churn prediction: Businesses can use random forests to predict which customers are likely to churn (cancel their service) so that they can take steps to retain them. For example, a telecom company might use a random forest model to identify customers who are using their phone less frequently or who have a history of late payments.

Fraud detection: Random forests can identify fraudulent transactions in real-time. For instance, a bank might employ a random forest model to spot transactions made from unusual locations or involving unusually large amounts of money.

Stock price prediction: Random forest can predict future stock prices. However, it is important to note that stock price prediction is a very difficult task, and no model is ever going to be perfectly accurate.

Medical diagnosis: These can help doctors diagnose diseases. For example, a doctor might use a random forest model to help them diagnose a patient with cancer.

Image recognition: It can recognize objects in images. For example, a self-driving car might use a random forest model to identify pedestrians and other vehicles on the road.

3- Real-Life Analogy of Random Forest

Let’s dive into a real-life analogy to understand this concept further. A student named X wants to choose a course, and he cant decide which course fit for his skill set. So he decides to consult various people like his cousins, teachers, parents, degree students, and working people. He asks them varied questions like why he should choose, job opportunities with that course, course fee, etc. Finally, after consulting various people about the course he decides to take the course suggested by most people.

Working of Random Forest Algorithm

Before understanding the working of the random forest algorithm in machine learning, we must look into the ensemble learning technique. Ensemble simplymeans combining multiple models. Thus a collection of models is used to make predictions rather than an individual model.

Ensemble uses two types of methods:

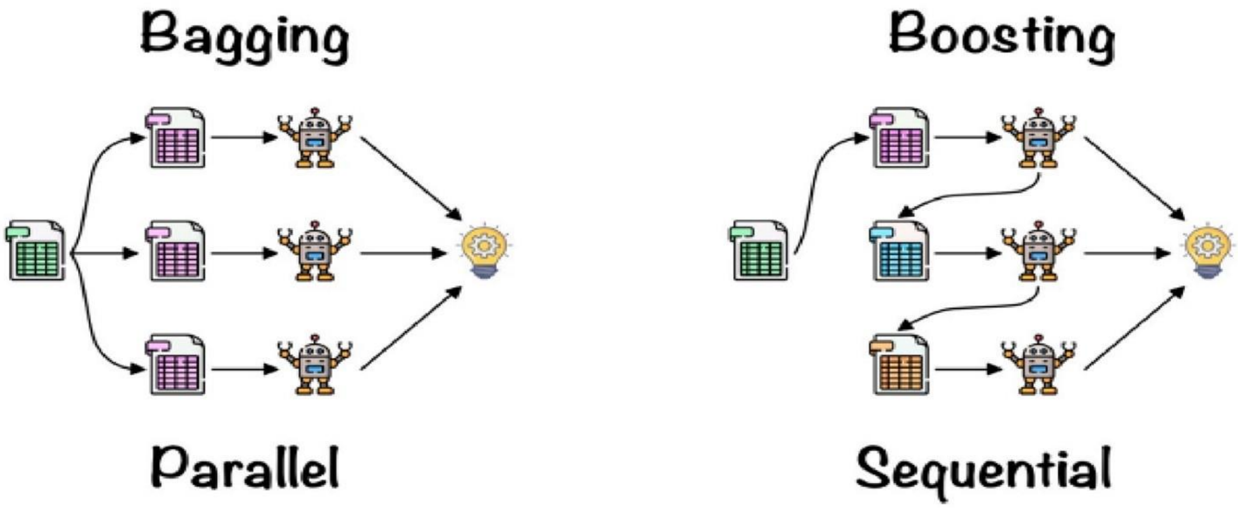


Figure 3.1

3.5 Advantages and Disadvantages of Random Forest algorithm ?

Advantages

- It can be used in classification and regression problems.
- It solves the problem of overfitting as output is based on majority voting or averaging.
- It performs well even if the data contains null/missing values.
- Each decision tree created is independent of the other; thus, it shows the property of parallelization.
- It is highly stable as the average answers given by a large number of trees are taken.
- It maintains diversity as all the attributes are not considered while making each decision tree though it is not true in all cases.
- It is immune to the curse of dimensionality. Since each tree does not consider all the attributes, feature space is reduced.
- We don’t have to segregate data into train and test as there will always be 30% of the data, which is not seen by the decision tree made out of bootstrap.

Disadvantages

- Random forest is highly complex compared to decision trees, where decisions can be made by following the path of the tree.
- Training time is more than other models due to its complexity. Whenever it has to make a prediction, each decision tree has to generate output for the given input data.

So:

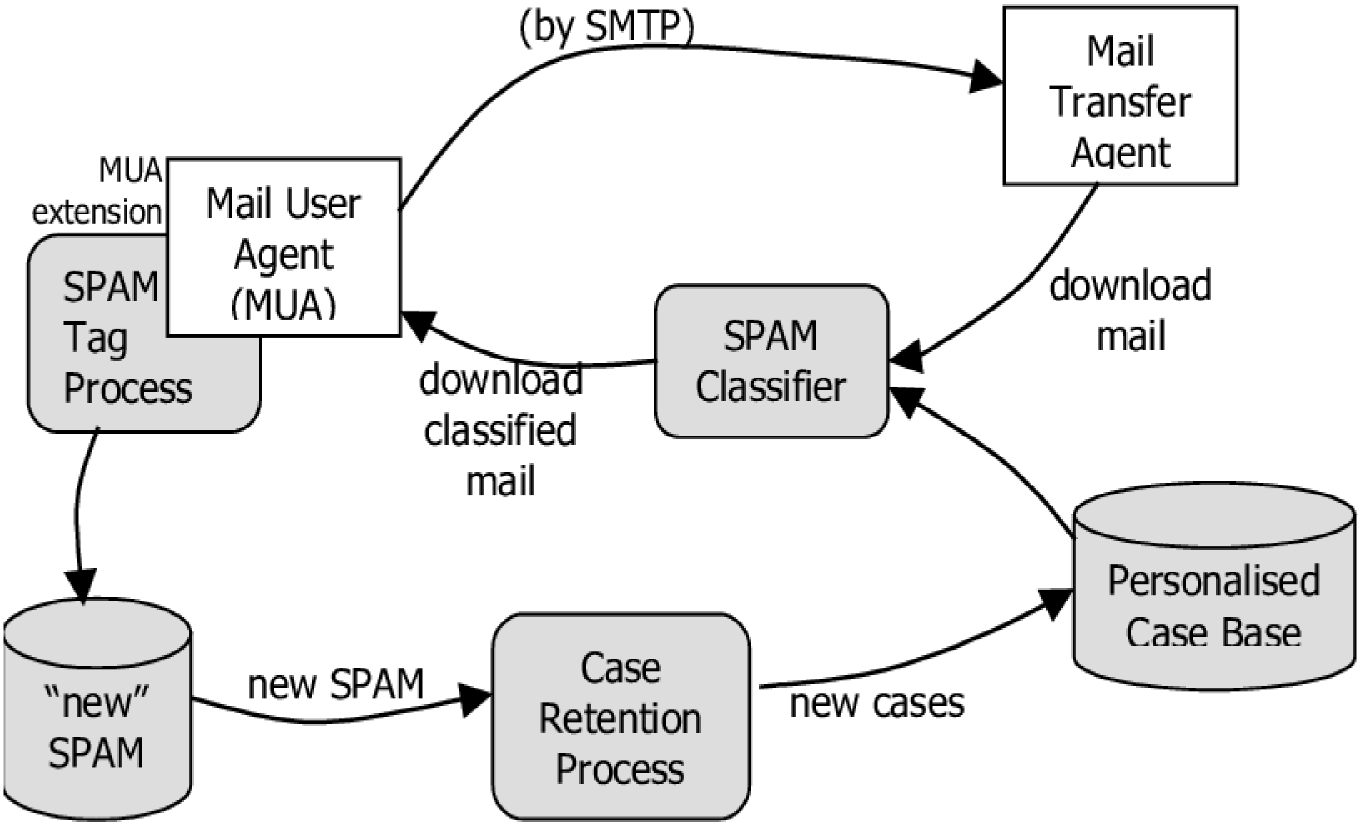
Random forest is a great choice if anyone wants to build the model fast and efficiently, as one of the best things about the random forest Classifier is it can handle missing values. It is one of the best techniques with high performance, widely used in various industries for its efficiency. It can handle binary, continuous, and categorical data. Overall, random forest is a fast, simple, flexible, and robust model with some limitations.

4

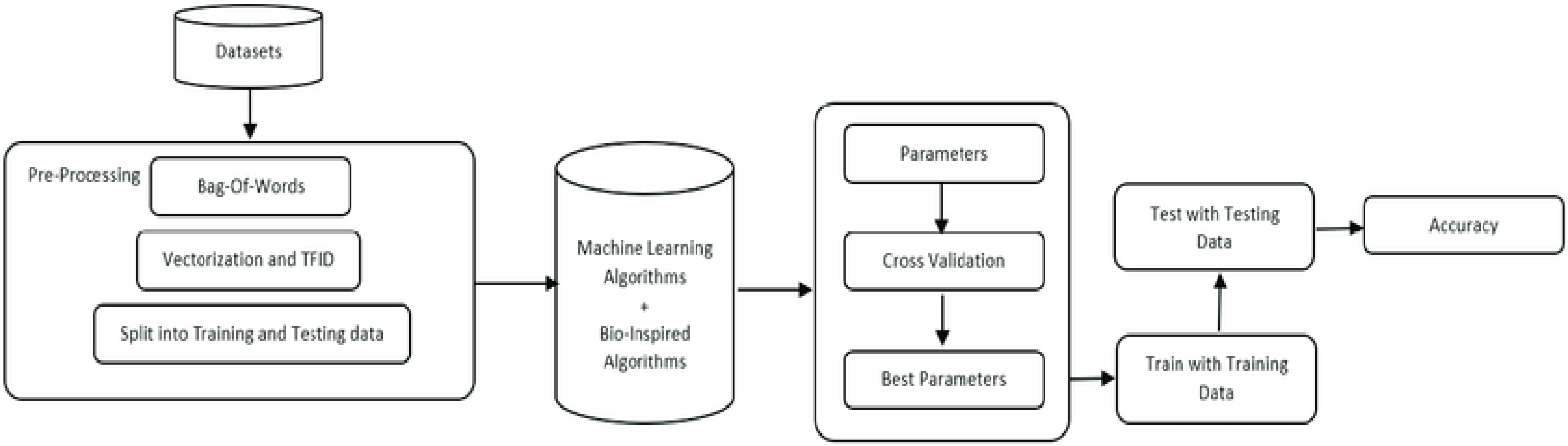
Dataset & Design

4.1 UML Diagram:

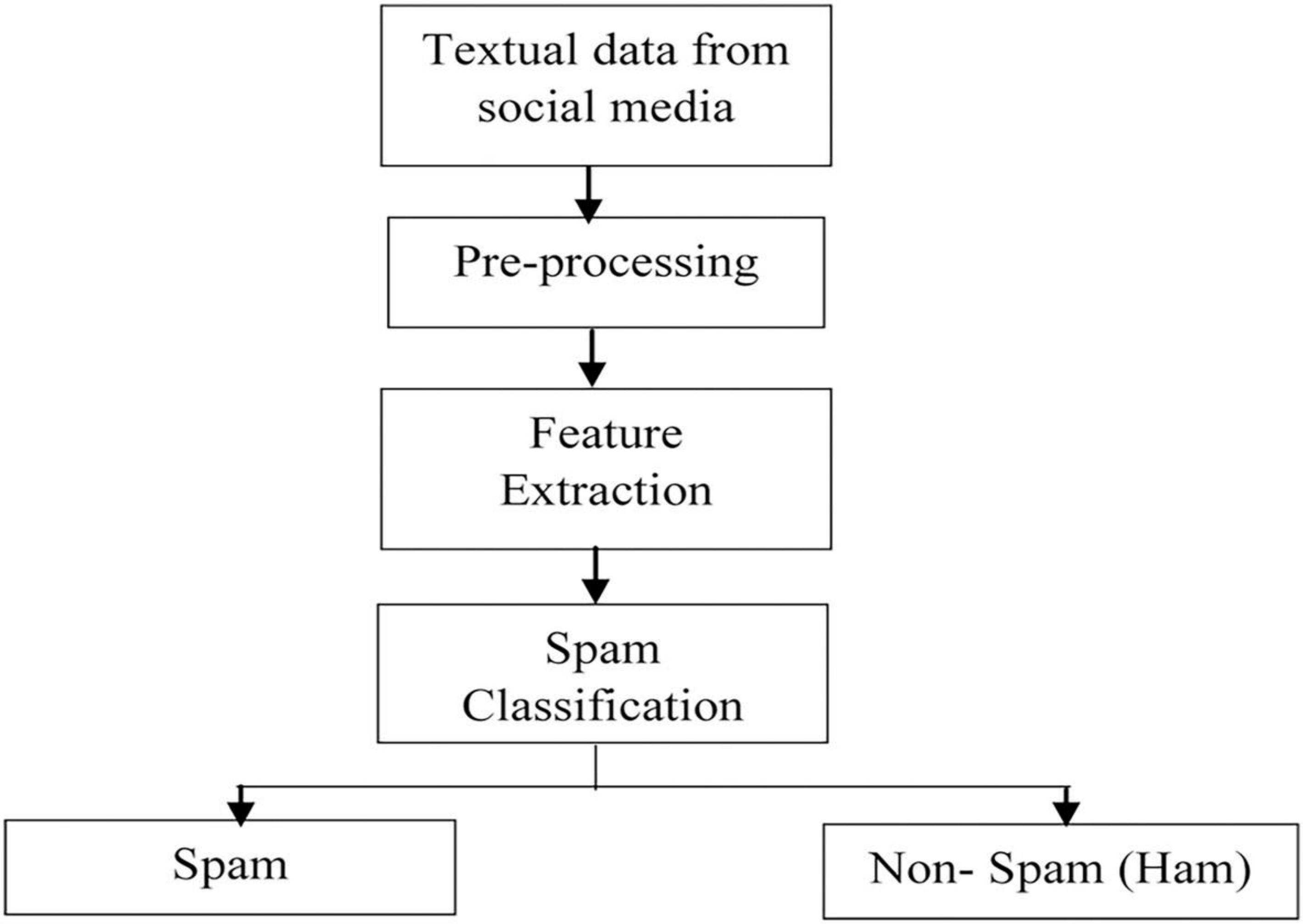
We used the Use Case methodology in our system analysis for spam detection to identify, clarify, and organize our system requirements. This approach helps us specify the roles played by the actors within the system and the relationships between and among them.



UML diagram Figure 4.1



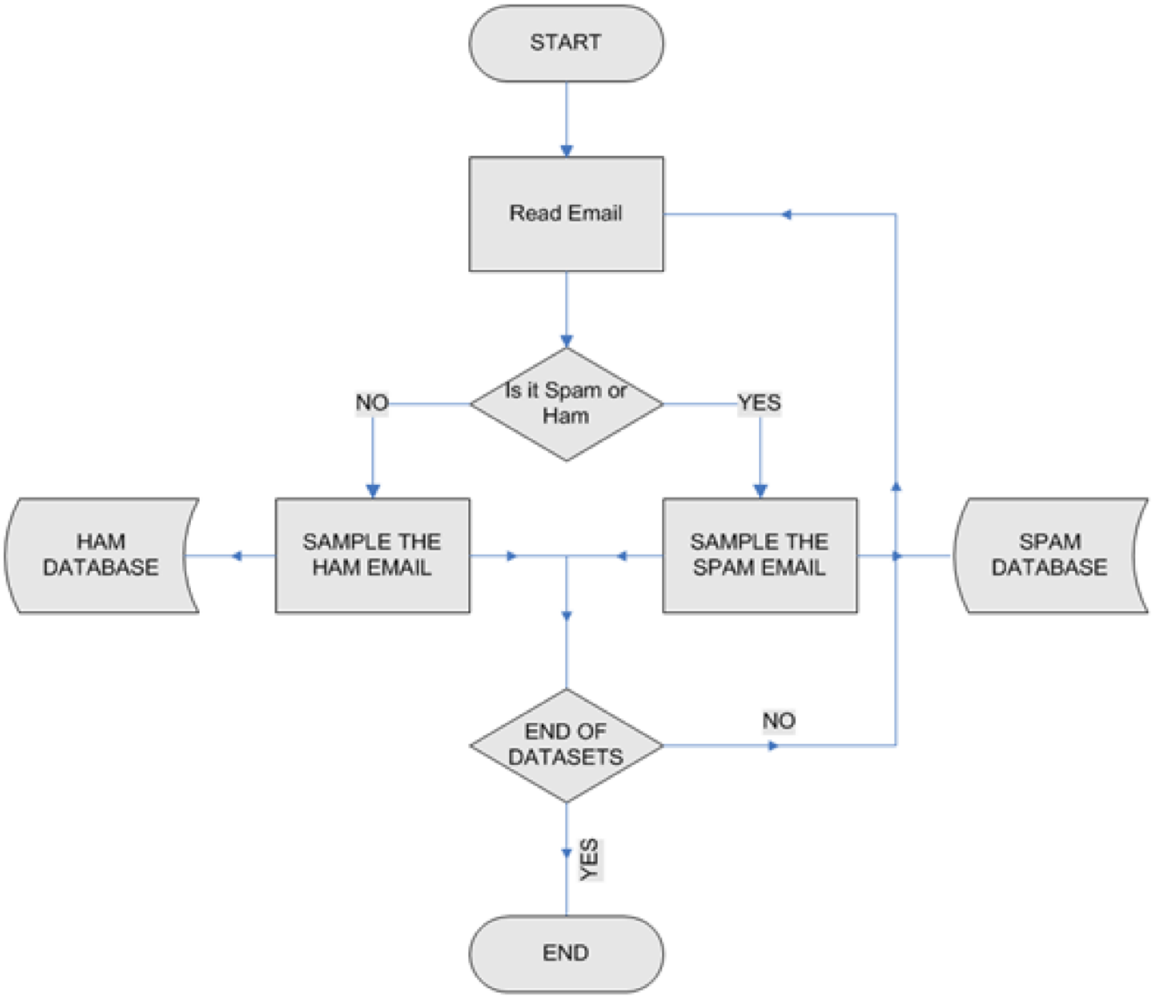
Model work Figure 4.2



Process Figure 4.3

4.2 Flowchart

A flowchart is a type of diagram that represents a workflow or process. A flowchart can also be defined as a diagrammatic representation of an algorithm, a step-by-step approach to solving a task.
The flowchart shows the steps as boxes of various kinds, and their order by connecting the boxes with arrows. This diagrammatic representation illustrates a solution model to a given problem. Flowcharts are used in analysing, designing, documenting or managing a process or program in various fields



Flowchart Figure 4.4

Flowchart Explanation:

The flowchart outlines a systematic process for detecting and classifying spam emails. Here are the key steps:

1. **Start**: The process begins.
2. **Read Email**: The system reads an incoming email.

3. **Spam or Ham Decision**: The system determines whether the email is spam or legitimate (ham)

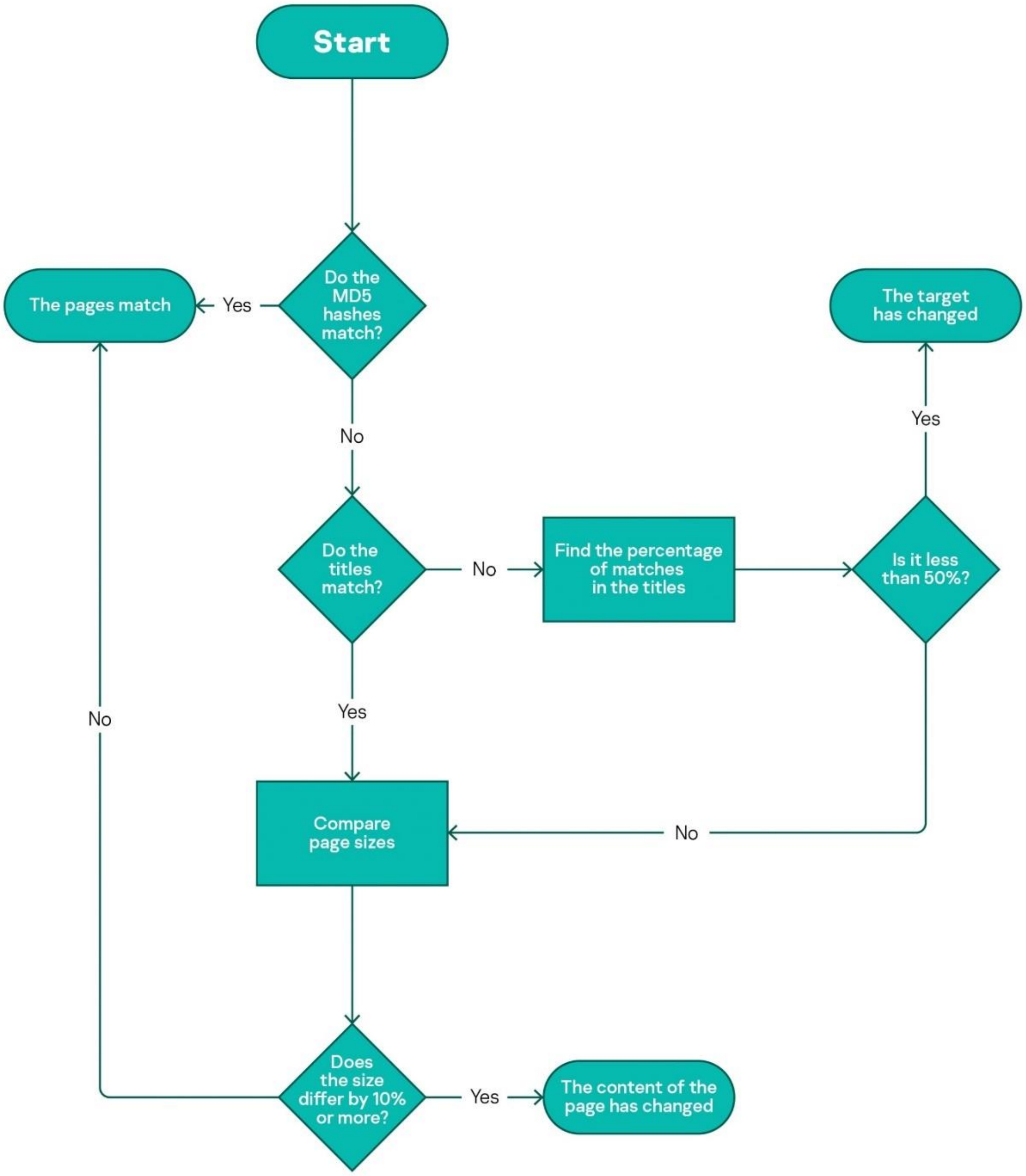
4. **Spam Handling**: If spam, the email is sampled and stored in the Spam Database.

5. **Ham Handling**: If ham, the email is sampled and stored in the Ham Database.

6. **End of Datasets Check**: The system checks if all emails have been processed:
    1. If yes, the process ends.
    2. If no, the process loops back to read the next email.

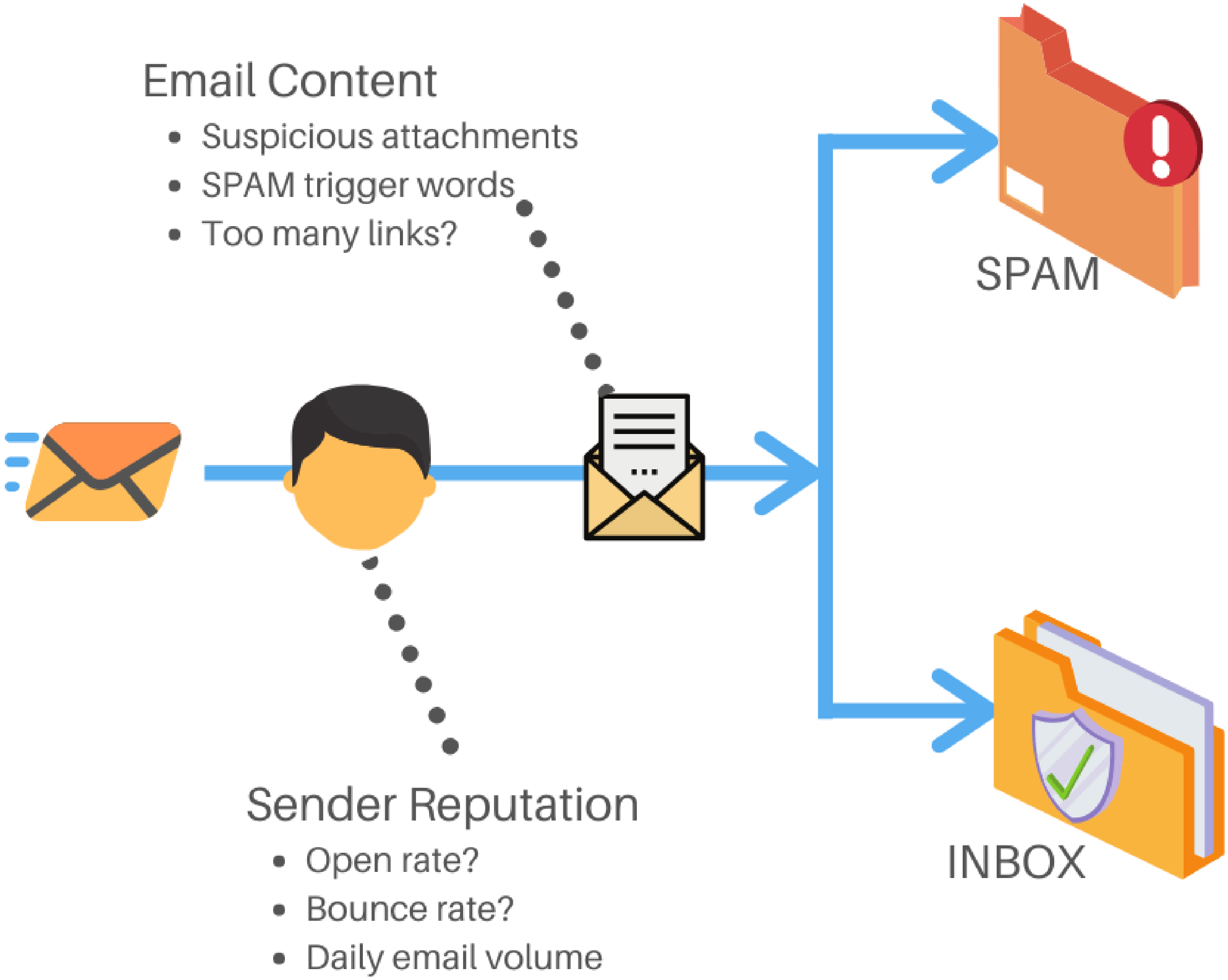
This methodical approach ensures efficient classification and storage of emails, enhancing spam detection and user email experience.

And this is kind of spam that we have to detect (phishing pages life cycle)



phishing pages life cycle Figure 4.5

4.3 Spam detecting process diagram :



Spam detecting process diagram Figure 4.6

4.5 Dataset we used

The choice of a dataset for machine learning is influenced by several factors, which are crucial for the success of the model. Here are some key considerations:

Relevance: The data should be relevant to the problem you're trying to solve. It must contain the features necessary to model the problem accurately**.**

Quality: High-quality data is essential. It should be clean, well-documented, and free from errors or biases that could affect the model's performance.

Size: The amount of data should be sufficient to train the model effectively. More data can improve the model's accuracy, but it also requires more computational resources.

Variety: A diverse dataset can improve the model's ability to generalize and perform well on unseen data. It should represent the different scenarios the model will encounter.

Balance: The dataset should ideally be balanced, especially for classification problems, to prevent the model from being biased towards certain classes.

Timeliness: The data should be up-to-date and reflect current trends or patterns relevant to the problem domain**.**

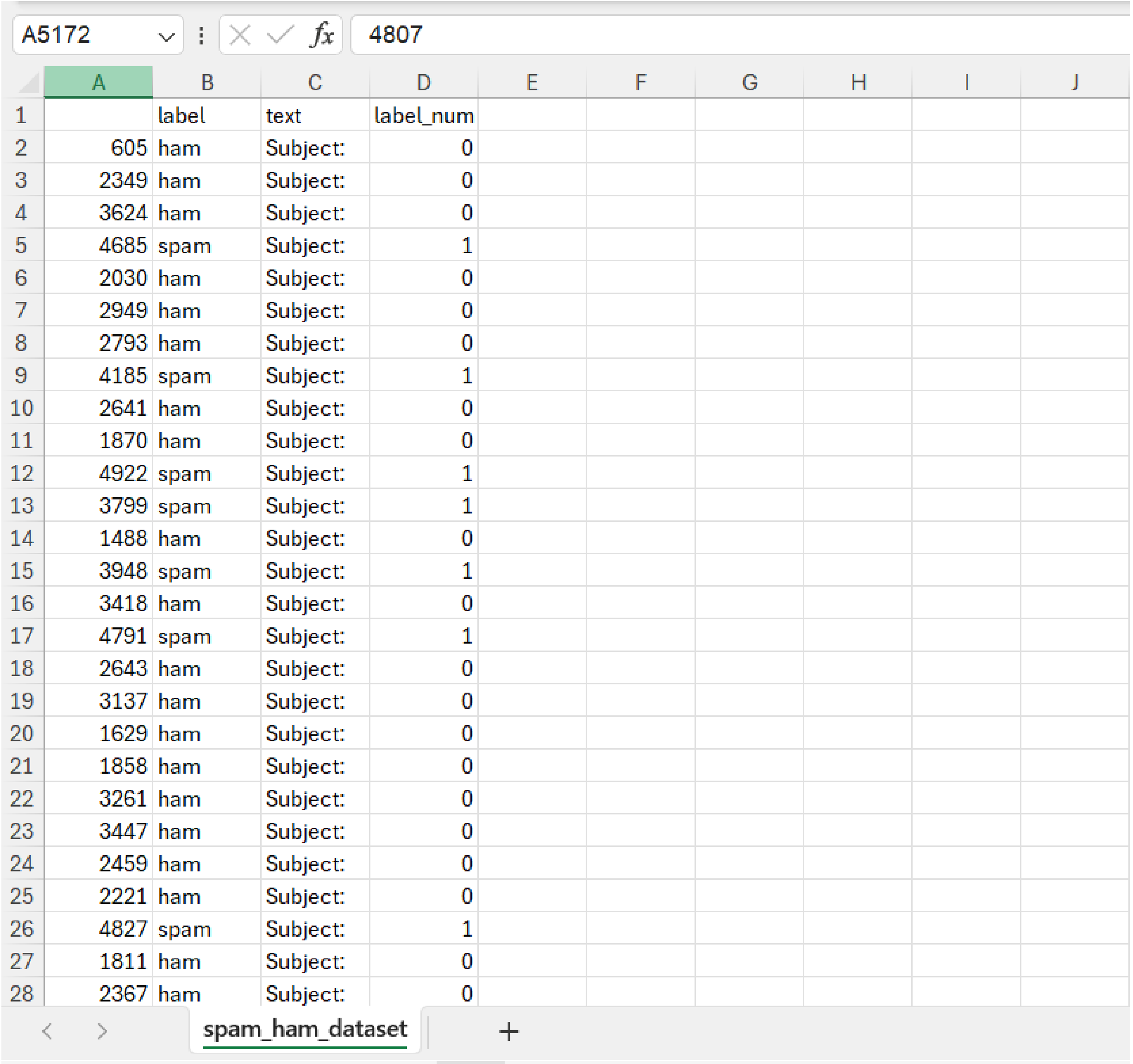
Accessibility: The data must be accessible and available for use, considering any legal or privacy constraints.

Format: The data format should be compatible with the tools and software you plan to use for analysis and modeling**.**

Computational Resources: The size and complexity of the data should match the computational resources available to process and analyze it efficiently**.**

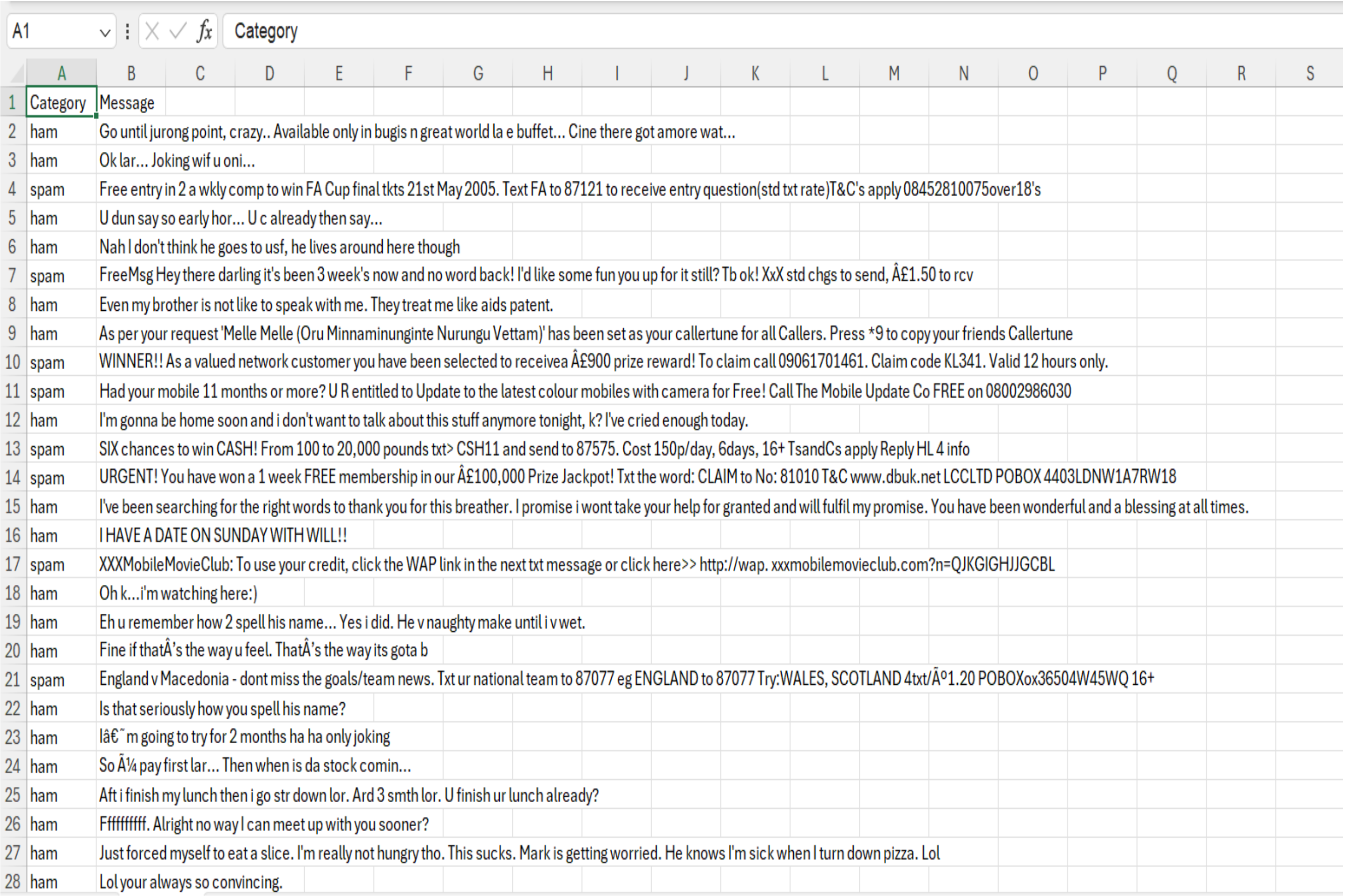
So it was not easy to choose dataset because we chose dataset several times, but we didn't get the results we wanted because the prediction doesn't give accurate results

Here are some examples of what we've chosen:



	A	B	C	D
1		label	text	label_num
2	605	ham	Subject:	0
3	2349	ham	Subject:	0
4	3624	ham	Subject:	0
5	4685	spam	Subject:	1
6	2030	ham	Subject:	0
7	2949	ham	Subject:	0
8	2793	ham	Subject:	0
9	4185	spam	Subject:	1
10	2641	ham	Subject:	0
11	1870	ham	Subject:	0
12	4922	spam	Subject:	1
13	3799	spam	Subject:	1
14	1488	ham	Subject:	0
15	3948	spam	Subject:	1
16	3418	ham	Subject:	0
17	4791	spam	Subject:	1
18	2643	ham	Subject:	0
19	3137	ham	Subject:	0
20	1629	ham	Subject:	0
21	1858	ham	Subject:	0
22	3261	ham	Subject:	0
23	3447	ham	Subject:	0
24	2459	ham	Subject:	0
25	2221	ham	Subject:	0
26	4827	spam	Subject:	1
27	1811	ham	Subject:	0
28	2367	ham	Subject:	0

It was 5172 rows and 3 column Figure 4.6

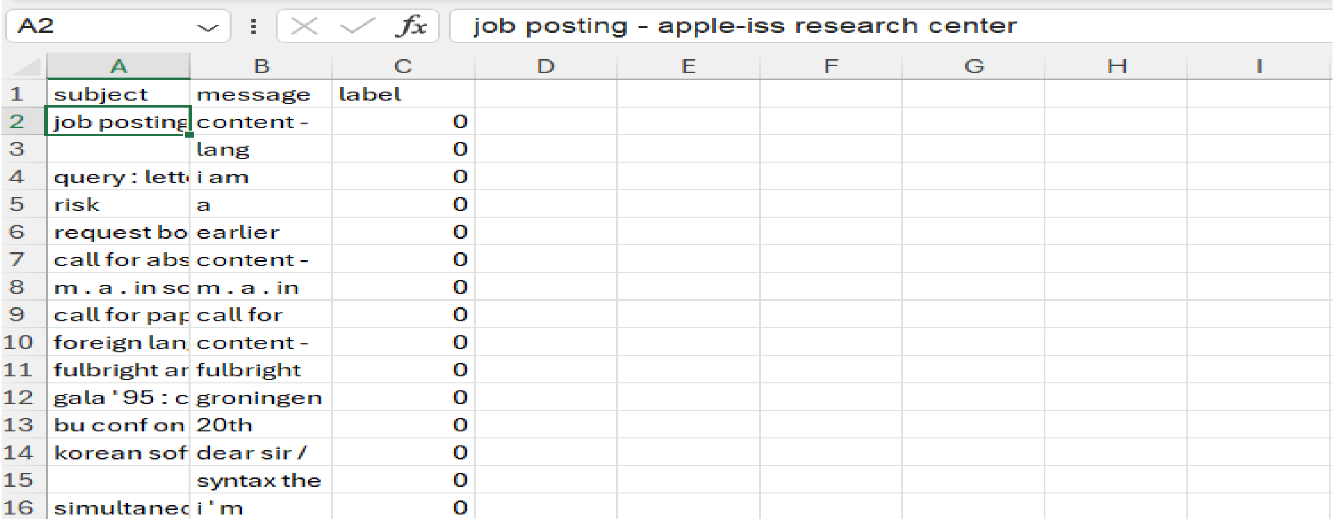


A1 Category

	A	B
1	Category	Message
2	ham	Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine there got amore wat...
3	ham	Ok lar... Joking wif u oni...
4	spam	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to receive entry question(std txt rate)T&C's apply 08452810075over18's
5	ham	U dun say so early hor... U c already then say...
6	ham	Nah I don't think he goes to usf, he lives around here though
7	spam	FreeMsg Hey there darling it's been 3 week's now and no word back! I'd like some fun you up for it still? Tb ok! XxX std chgs to send, Â£1.50 to rcv
8	ham	Even my brother is not like to speak with me. They treat me like aids patent.
9	ham	As per your request 'Melle Melle (Oru Minnaminunginte Nurungu Vettam)' has been set as your callertune for all Callers. Press *9 to copy your friends Callertune
10	spam	WINNER!! As a valued network customer you have been selected to receivea Â£900 prize reward! To claim call 09061701461. Claim code KL341. Valid 12 hours only.
11	spam	Had your mobile 11 months or more? U R entitled to Update to the latest colour mobiles with camera for Free! Call The Mobile Update Co FREE on 08002986030
12	ham	I'm gonna be home soon and i don't want to talk about this stuff anymore tonight, k? I've cried enough today.
13	spam	SIX chances to win CASH! From 100 to 20,000 pounds txt> CSH11 and send to 87575. Cost 150p/day, 6days, 16+ TsandCs apply Reply HL 4 info
14	spam	URGENT! You have won a 1 week FREE membership in our Â£100,000 Prize Jackpot! Txt the word: CLAIM to No: 81010 T&C www.dbuk.net LCCLTD POBOX 4403LDNW1A7RW18
15	ham	I've been searching for the right words to thank you for this breather. I promise i wont take your help for granted and will fulfil my promise. You have been wonderful and a blessing at all times.
16	ham	I HAVE A DATE ON SUNDAY WITH WILL!!
17	spam	XXXMobileMovieClub: To use your credit, click the WAP link in the next txt message or click here>> http://wap. xxxmobilemovieclub.com?n=QJKGIGHJJGCBL
18	ham	Oh k...i'm watching here:)
19	ham	Eh u remember how 2 spell his name... Yes i did. He v naughty make until i v wet.
20	ham	Fine if thatÂ's the way u feel. ThatÂ's the way its gota b
21	spam	England v Macedonia - dont miss the goals/team news. Txt ur national team to 87077 eg ENGLAND to 87077 Try:WALES, SCOTLAND 4txt/Ã°1.20 POBOXox36504W45WQ 16+
22	ham	Is that seriously how you spell his name?
23	ham	Iâ€˜m going to try for 2 months ha ha only joking
24	ham	So Ã¼ pay first lar... Then when is da stock comin...
25	ham	Aft i finish my lunch then i go str down lor. Ard 3 smth lor. U finish ur lunch already?
26	ham	Fffffffff. Alright no way I can meet up with you sooner?
27	ham	Just forced myself to eat a slice. I'm really not hungry tho. This sucks. Mark is getting worried. He knows I'm sick when I turn down pizza. Lol
28	ham	Lol your always so convincing.

it was 5573 rows and 2 column

Figure 4.7



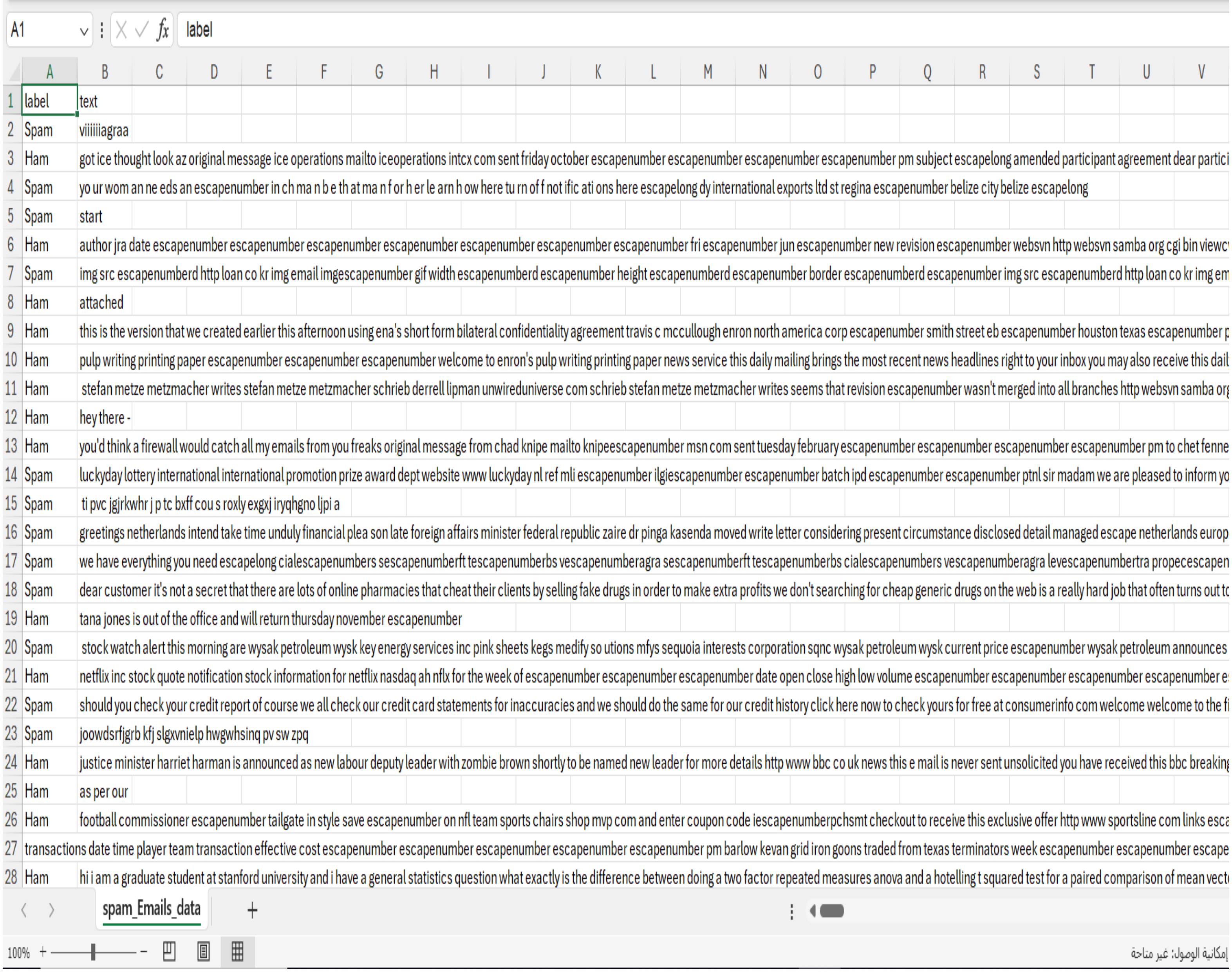
A2 job posting - apple-iss research center

	A	B	C
1	subject	message	label
2	job posting	content -	0
3		lang	0
4	query : lett	i am	0
5	risk	a	0
6	request bo	earlier	0
7	call for abs	content -	0
8	m . a . in sc	m . a . in	0
9	call for pap	call for	0
10	foreign lan	content -	0
11	fulbright ar	fulbright	0
12	gala ' 95 : c	groningen	0
13	bu conf on	20th	0
14	korean sof	dear sir /	0
15		syntax the	0
16	simultanec	i ' m	0

2894 rows

Figure 4.8

And In the end, we chose this one because it contains as much data as we need, which gives more accurate results



	A	B
1	label	text
2	Spam	viiiiiiagraa
3	Ham	got ice thought look az original message ice operations mailto iceoperations intcx com sent friday october escapenumber escapenumber escapenumber escapenumber pm subject escapelong amended participant agreement dear partici
4	Spam	yo ur wom an ne eds an escapenumber in ch ma n b e th at ma n f or h er le arn h ow here tu rn of f not ific ati ons here escapelong dy international exports ltd st regina escapenumber belize city belize escapelong
5	Spam	start
6	Ham	author jra date escapenumber escapenumber escapenumber escapenumber escapenumber escapenumber escapenumber fri escapenumber jun escapenumber new revision escapenumber websvn http websvn samba org cgi bin viewc
7	Spam	img src escapenumberd http loan co kr img email imgescapenumber gif width escapenumberd escapenumber height escapenumberd escapenumber border escapenumberd escapenumber img src escapenumberd http loan co kr img em
8	Ham	attached
9	Ham	this is the version that we created earlier this afternoon using ena's short form bilateral confidentiality agreement travis c mccullough enron north america corp escapenumber smith street eb escapenumber houston texas escapenumber p
10	Ham	pulp writing printing paper escapenumber escapenumber escapenumber welcome to enron's pulp writing printing paper news service this daily mailing brings the most recent news headlines right to your inbox you may also receive this dail
11	Ham	stefan metze metzmacher writes stefan metze metzmacher schrieb derrell lipman unwireduniverse com schrieb stefan metze metzmacher writes seems that revision escapenumber wasn't merged into all branches http websvn samba org
12	Ham	hey there -
13	Ham	you'd think a firewall would catch all my emails from you freaks original message from chad knipe mailto knipeescapenumber msn com sent tuesday february escapenumber escapenumber escapenumber escapenumber pm to chet fenne
14	Spam	luckyday lottery international international promotion prize award dept website www luckyday nl ref mli escapenumber ilgiescapenumber escapenumber batch ipd escapenumber escapenumber ptnl sir madam we are pleased to inform yo
15	Spam	ti pvc jgjrkwhr j p tc bxff cou s roxly exgxj iryqhgno ljpi a
16	Spam	greetings netherlands intend take time unduly financial plea son late foreign affairs minister federal republic zaire dr pinga kasenda moved write letter considering present circumstance disclosed detail managed escape netherlands europ
17	Spam	we have everything you need escapelong cialescapenumbers sescapenumberft tescapenumberbs vescapenumberagra sescapenumberft tescapenumberbs cialescapenumbers vescapenumberagra levescapenumbertra propecescapen
18	Spam	dear customer it's not a secret that there are lots of online pharmacies that cheat their clients by selling fake drugs in order to make extra profits we don't searching for cheap generic drugs on the web is a really hard job that often turns out to
19	Ham	tana jones is out of the office and will return thursday november escapenumber
20	Spam	stock watch alert this morning are wysak petroleum wysk key energy services inc pink sheets kegs medify so utions mfys sequoia interests corporation sqnc wysak petroleum wysk current price escapenumber wysak petroleum announces
21	Ham	netflix inc stock quote notification stock information for netflix nasdaq ah nflx for the week of escapenumber escapenumber escapenumber date open close high low volume escapenumber escapenumber escapenumber escapenumber e
22	Spam	should you check your credit report of course we all check our credit card statements for inaccuracies and we should do the same for our credit history click here now to check yours for free at consumerinfo com welcome welcome to the fi
23	Spam	joowdsrfjgrb kfj slgxvnielp hwgwhsinq pv sw zpq
24	Ham	justice minister harriet harman is announced as new labour deputy leader with zombie brown shortly to be named new leader for more details http www bbc co uk news this e mail is never sent unsolicited you have received this bbc breaking
25	Ham	as per our
26	Ham	football commissioner escapenumber tailgate in style save escapenumber on nfl team sports chairs shop mvp com and enter coupon code iescapenumberpchsmt checkout to receive this exclusive offer http www sportsline com links esca
27	transactions date time player team transaction effective cost escapenumber escapenumber escapenumber escapenumber escapenumber pm barlow kevan grid iron goons traded from texas terminators week escapenumber escapenumber escape	
28	Ham	hi i am a graduate student at stanford university and i have a general statistics question what exactly is the difference between doing a two factor repeated measures anova and a hotelling t squared test for a paired comparison of mean vect

+190000 message 47% ham 53% spam

Figure 4.9

5

Implementation

5.1 Model result

5.1.1. SVM

5.1.2. Random Forest

5.2 Web Application

5.1 Model result

Overview of dataset first :
This dataset contains over **190,000+** emails labeled as either spam or ham (non-spam). Each email is represented by its text content along with its corresponding label.

Description
The dataset provides a comprehensive collection of emails, categorized as either spam or ham, intended to facilitate research and development in email classification algorithms. With a vast corpus of emails, this dataset offers ample opportunities for training and evaluating machine learning models for effective spam detection.

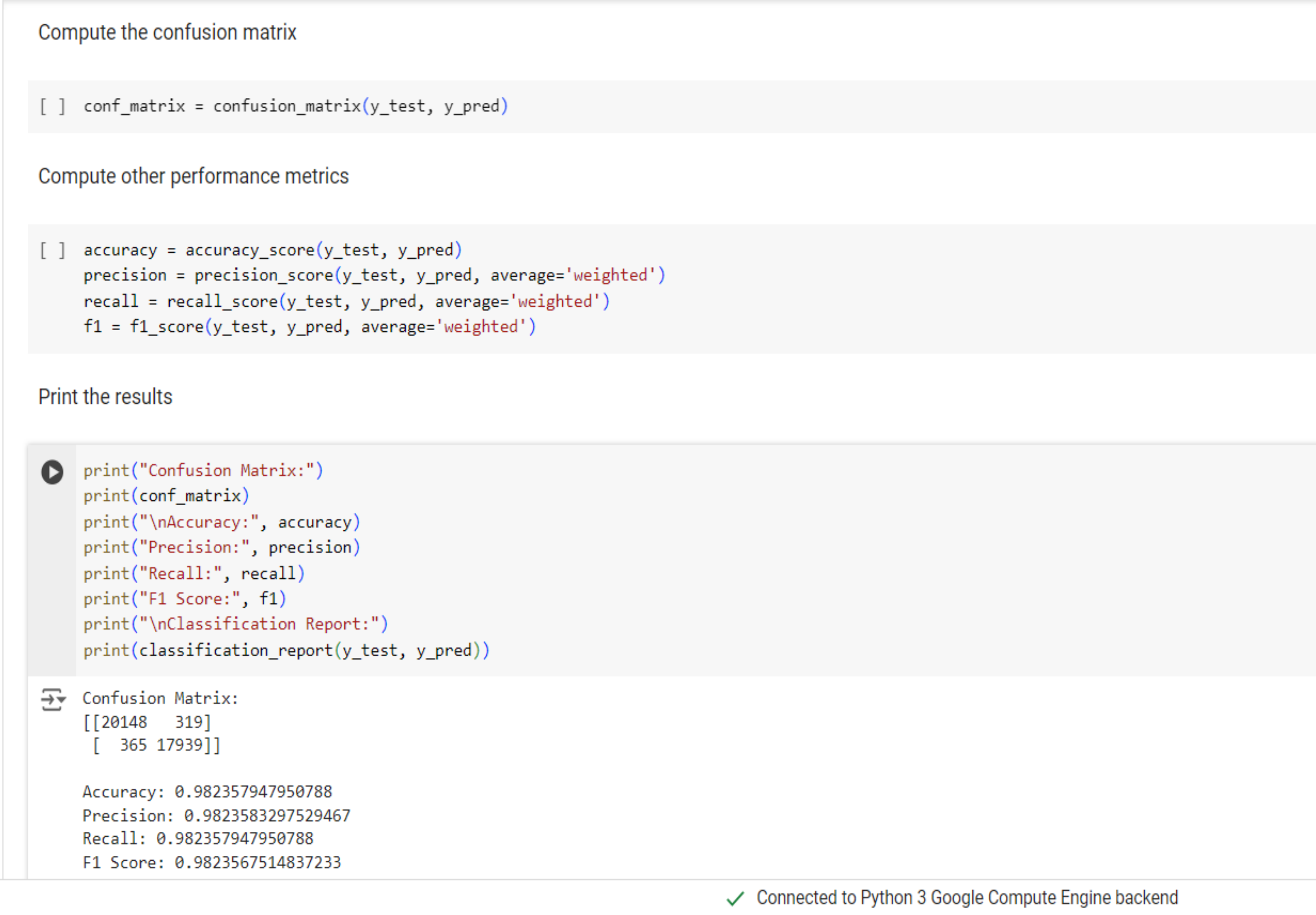
Features
Text: The content of the email.
Label: The classification label indicating whether the email is spam (1) or ham (0).

Usage
Researchers and practitioners can leverage this dataset to:

Develop and evaluate machine learning models for email classification.
Explore natural language processing techniques for spam detection.
Conduct comparative studies on the effectiveness of different classification algorithms.
Investigate emerging trends and patterns in email spamming behavior.

Linke of dataset we used in our model:
https://www.kaggle.com/datasets/meruvulikith/190k-spam-ham-email-dataset-for-classification/data

The accuracy of our model :



Compute the confusion matrix

```
[ ]  conf_matrix = confusion_matrix(y_test, y_pred)
```

Compute other performance metrics

```
[ ]  accuracy = accuracy_score(y_test, y_pred)
     precision = precision_score(y_test, y_pred, average='weighted')
     recall = recall_score(y_test, y_pred, average='weighted')
     f1 = f1_score(y_test, y_pred, average='weighted')
```

Print the results

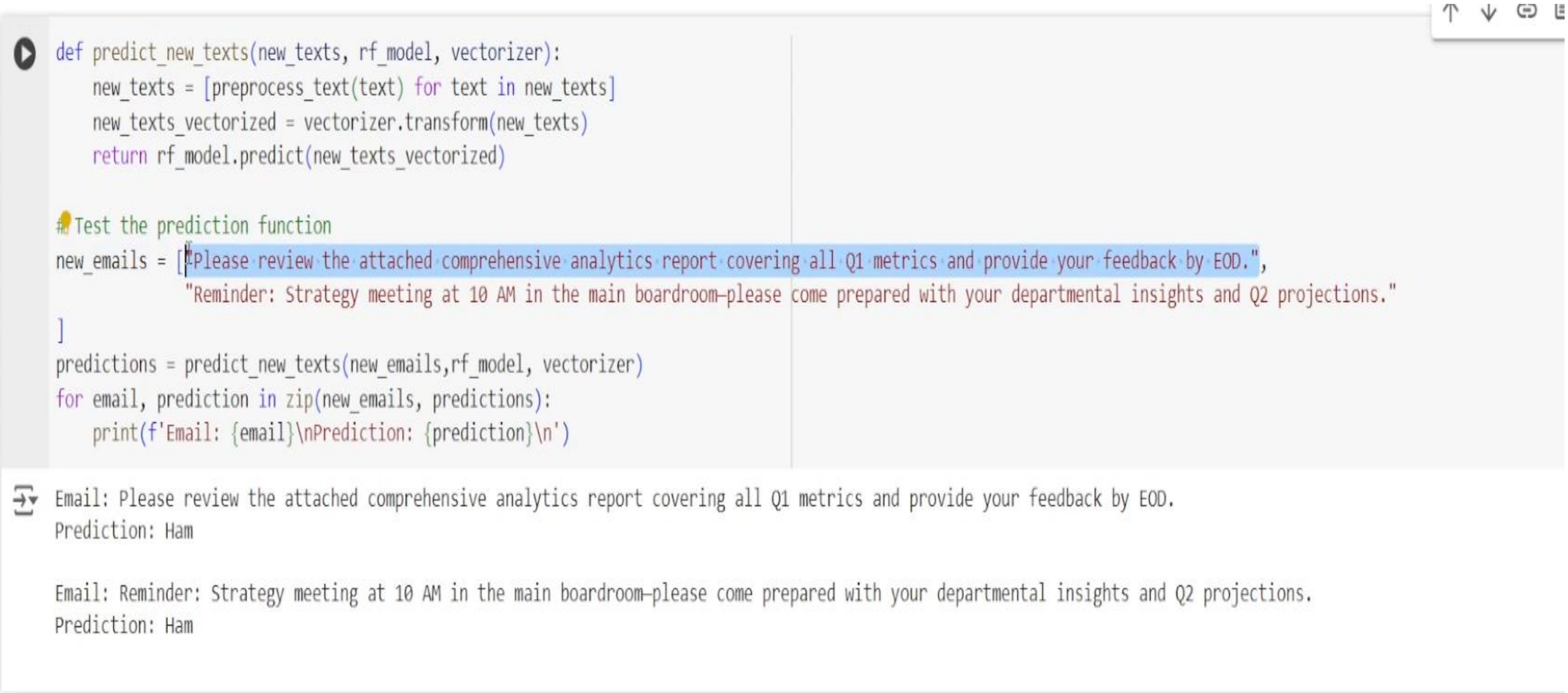
```
print("Confusion Matrix:")
print(conf_matrix)
print("\nAccuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("F1 Score:", f1)
print("\nClassification Report:")
print(classification_report(y_test, y_pred))
```

```
Confusion Matrix:
[[20148   319]
 [  365 17939]]

Accuracy: 0.982357947950788
Precision: 0.9823583297529467
Recall: 0.982357947950788
F1 Score: 0.9823567514837233
```

✓ Connected to Python 3 Google Compute Engine backend

5.1.1 SVM :

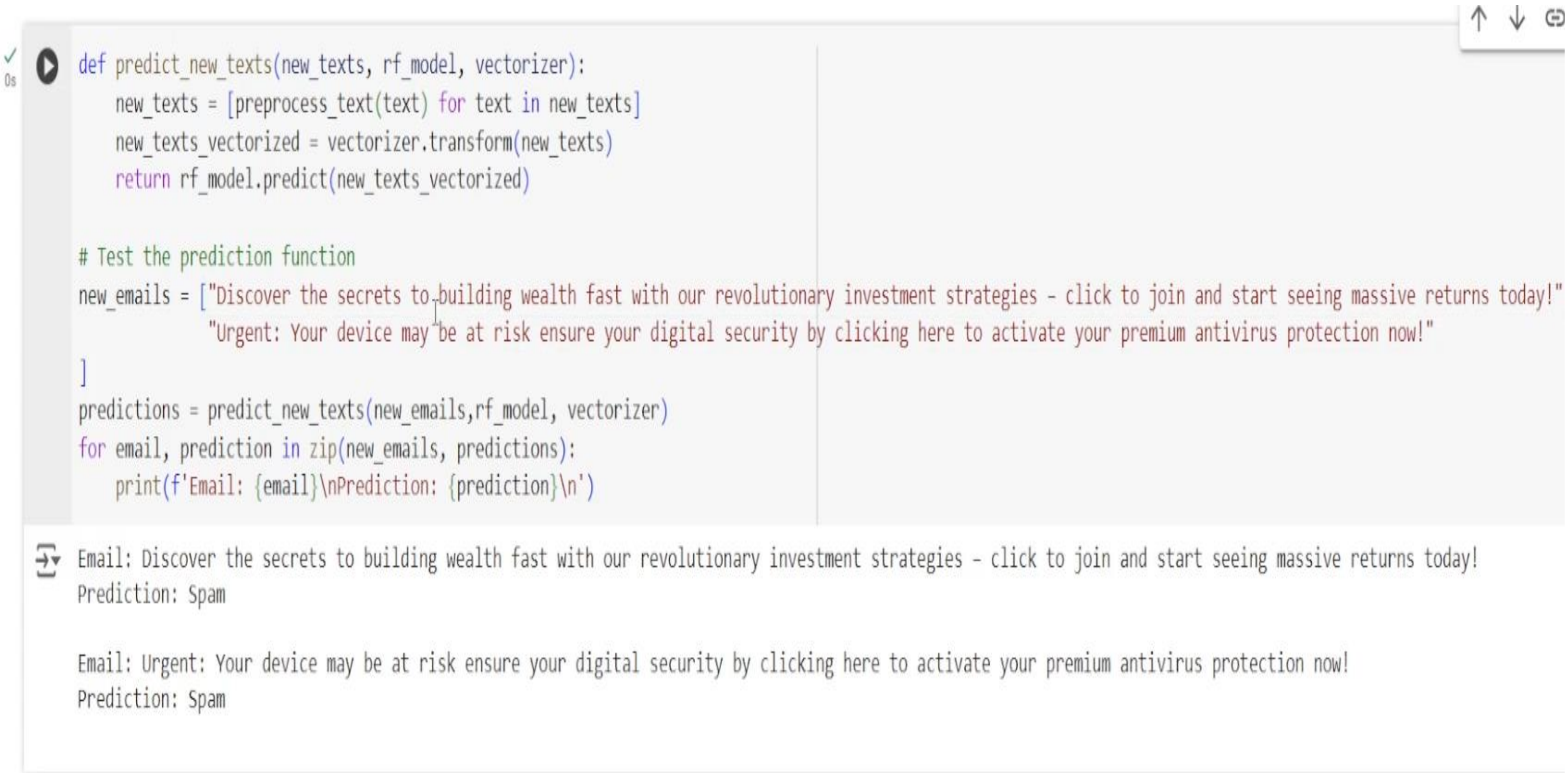


```
def predict_new_texts(new_texts, rf_model, vectorizer):
    new_texts = [preprocess_text(text) for text in new_texts]
    new_texts_vectorized = vectorizer.transform(new_texts)
    return rf_model.predict(new_texts_vectorized)

# Test the prediction function
new_emails = ["Please review the attached comprehensive analytics report covering all Q1 metrics and provide your feedback by EOD.",
              "Reminder: Strategy meeting at 10 AM in the main boardroom—please come prepared with your departmental insights and Q2 projections."
]
predictions = predict_new_texts(new_emails,rf_model, vectorizer)
for email, prediction in zip(new_emails, predictions):
    print(f'Email: {email}\nPrediction: {prediction}\n')
```

```
Email: Please review the attached comprehensive analytics report covering all Q1 metrics and provide your feedback by EOD.
Prediction: Ham

Email: Reminder: Strategy meeting at 10 AM in the main boardroom—please come prepared with your departmental insights and Q2 projections.
Prediction: Ham
```



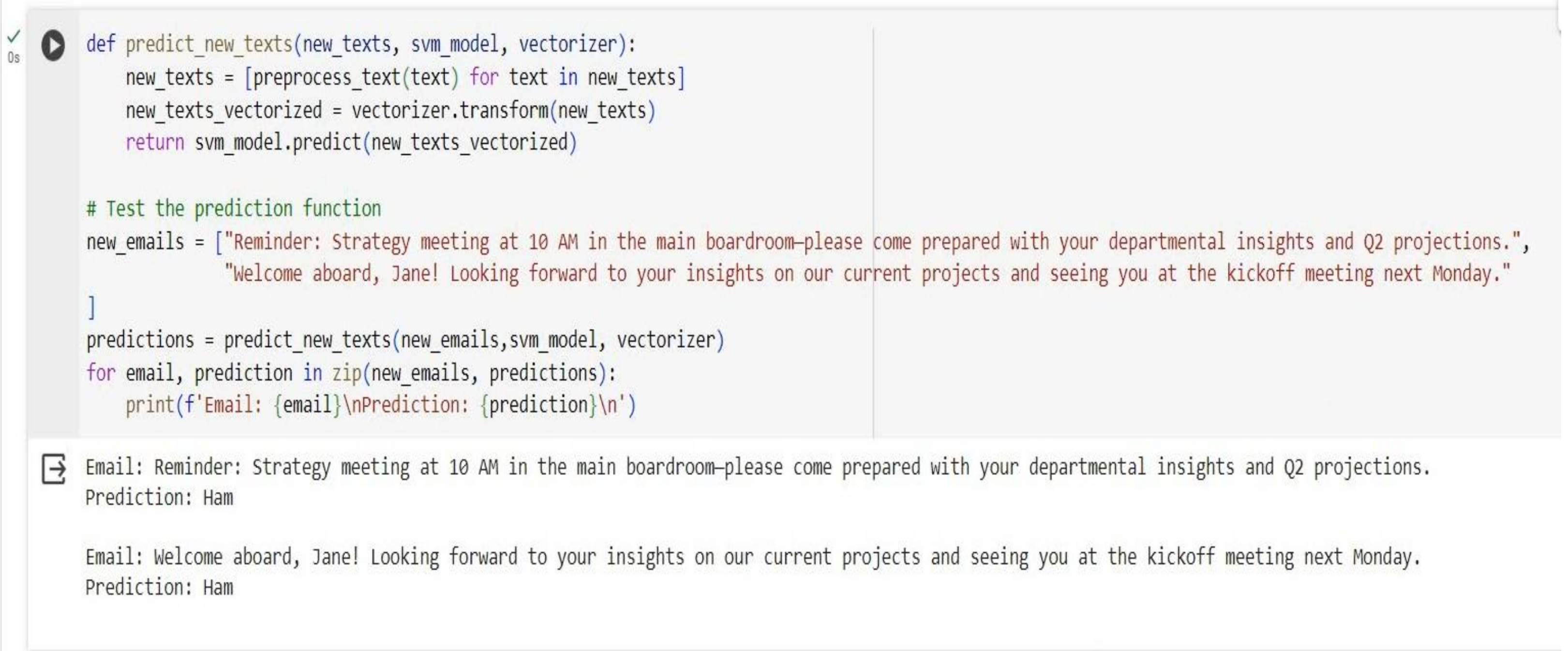
```
def predict_new_texts(new_texts, rf_model, vectorizer):
    new_texts = [preprocess_text(text) for text in new_texts]
    new_texts_vectorized = vectorizer.transform(new_texts)
    return rf_model.predict(new_texts_vectorized)

# Test the prediction function
new_emails = ["Discover the secrets to building wealth fast with our revolutionary investment strategies - click to join and start seeing massive returns today!"
              "Urgent: Your device may be at risk ensure your digital security by clicking here to activate your premium antivirus protection now!"
]
predictions = predict_new_texts(new_emails,rf_model, vectorizer)
for email, prediction in zip(new_emails, predictions):
    print(f'Email: {email}\nPrediction: {prediction}\n')
```

```
Email: Discover the secrets to building wealth fast with our revolutionary investment strategies - click to join and start seeing massive returns today!
Prediction: Spam

Email: Urgent: Your device may be at risk ensure your digital security by clicking here to activate your premium antivirus protection now!
Prediction: Spam
```

5.1.2 *Random Forest :*

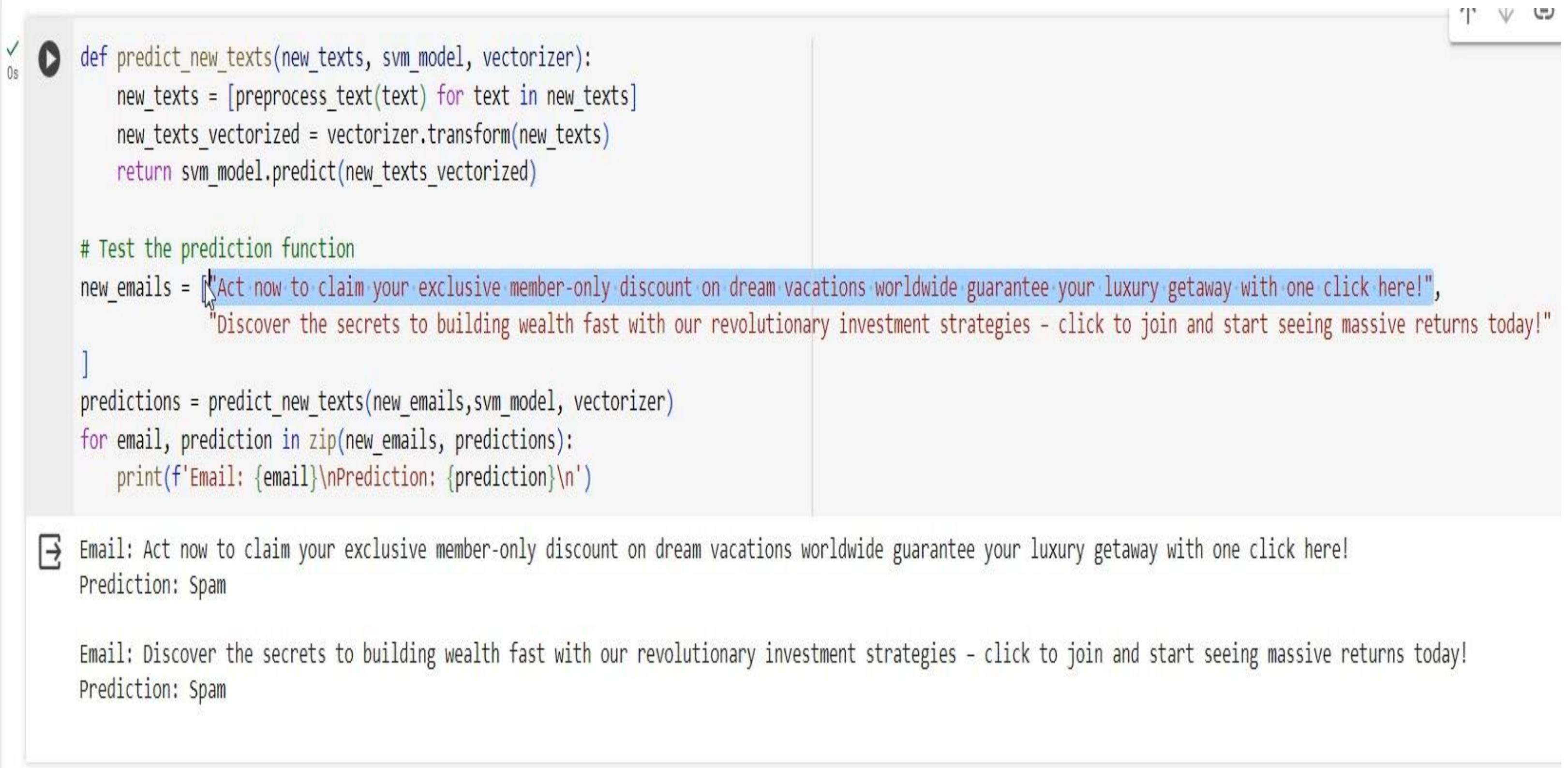


```
def predict_new_texts(new_texts, svm_model, vectorizer):
    new_texts = [preprocess_text(text) for text in new_texts]
    new_texts_vectorized = vectorizer.transform(new_texts)
    return svm_model.predict(new_texts_vectorized)

# Test the prediction function
new_emails = ["Reminder: Strategy meeting at 10 AM in the main boardroom—please come prepared with your departmental insights and Q2 projections.",
              "Welcome aboard, Jane! Looking forward to your insights on our current projects and seeing you at the kickoff meeting next Monday."
]
predictions = predict_new_texts(new_emails,svm_model, vectorizer)
for email, prediction in zip(new_emails, predictions):
    print(f'Email: {email}\nPrediction: {prediction}\n')
```

```
Email: Reminder: Strategy meeting at 10 AM in the main boardroom—please come prepared with your departmental insights and Q2 projections.
Prediction: Ham

Email: Welcome aboard, Jane! Looking forward to your insights on our current projects and seeing you at the kickoff meeting next Monday.
Prediction: Ham
```



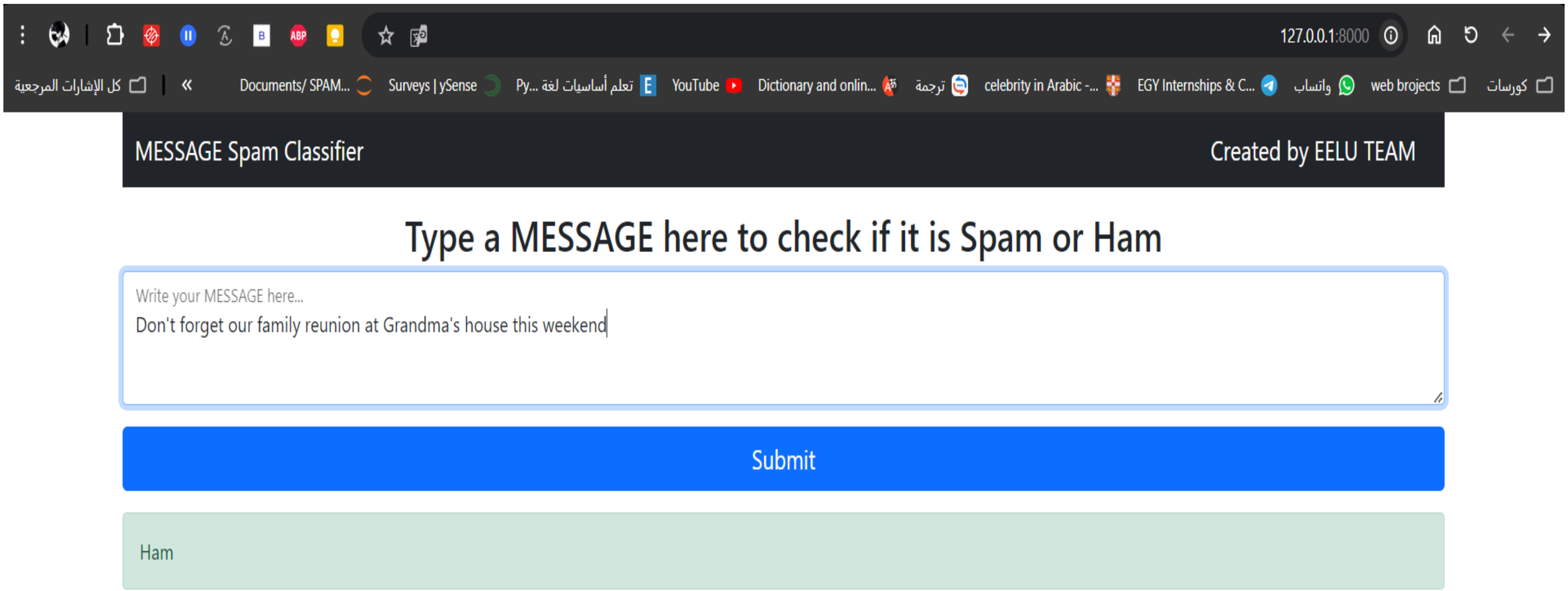
```
def predict_new_texts(new_texts, svm_model, vectorizer):
    new_texts = [preprocess_text(text) for text in new_texts]
    new_texts_vectorized = vectorizer.transform(new_texts)
    return svm_model.predict(new_texts_vectorized)

# Test the prediction function
new_emails = ["Act now to claim your exclusive member-only discount on dream vacations worldwide guarantee your luxury getaway with one click here!",
              "Discover the secrets to building wealth fast with our revolutionary investment strategies - click to join and start seeing massive returns today!"
]
predictions = predict_new_texts(new_emails,svm_model, vectorizer)
for email, prediction in zip(new_emails, predictions):
    print(f'Email: {email}\nPrediction: {prediction}\n')
```

```
Email: Act now to claim your exclusive member-only discount on dream vacations worldwide guarantee your luxury getaway with one click here!
Prediction: Spam

Email: Discover the secrets to building wealth fast with our revolutionary investment strategies - click to join and start seeing massive returns today!
Prediction: Spam
```

5.2 Web Application results :



MESSAGE Spam Classifier

Created by EELU TEAM

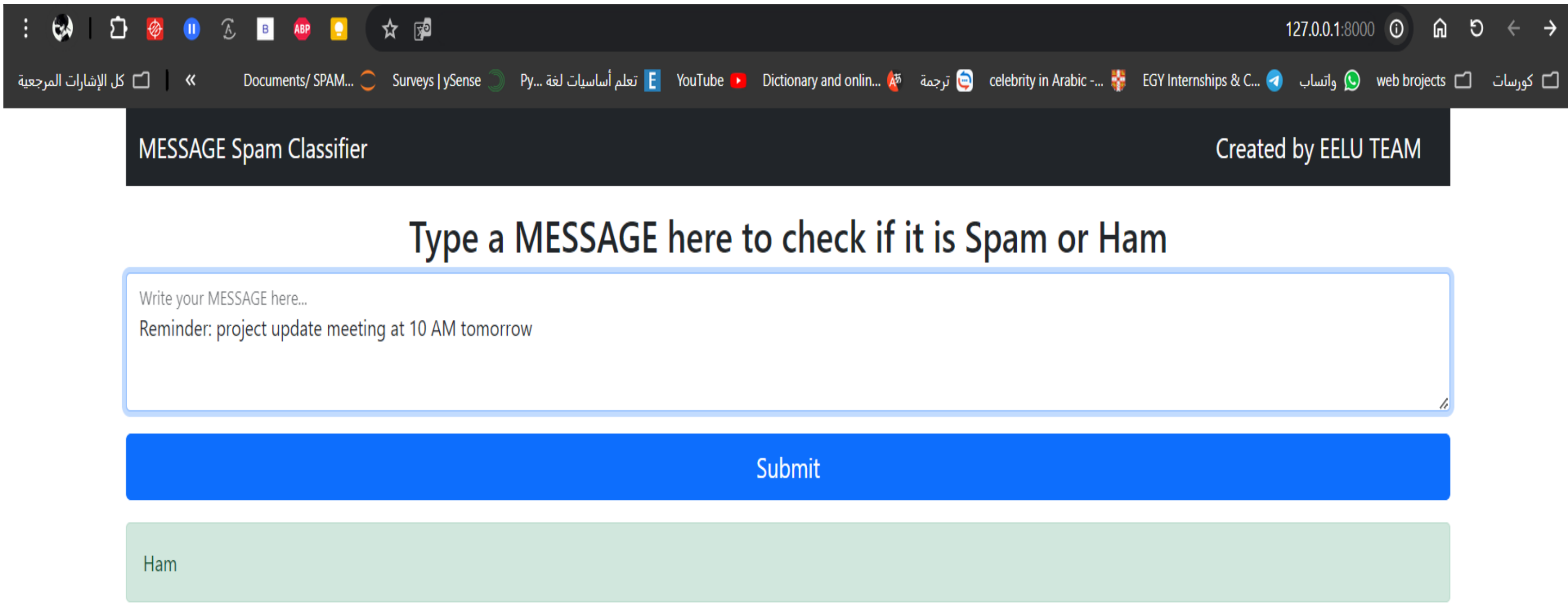
Type a MESSAGE here to check if it is Spam or Ham

Write your MESSAGE here...

Don't forget our family reunion at Grandma's house this weekend

Ham

Result Figure 5.1



MESSAGE Spam Classifier

Created by EELU TEAM

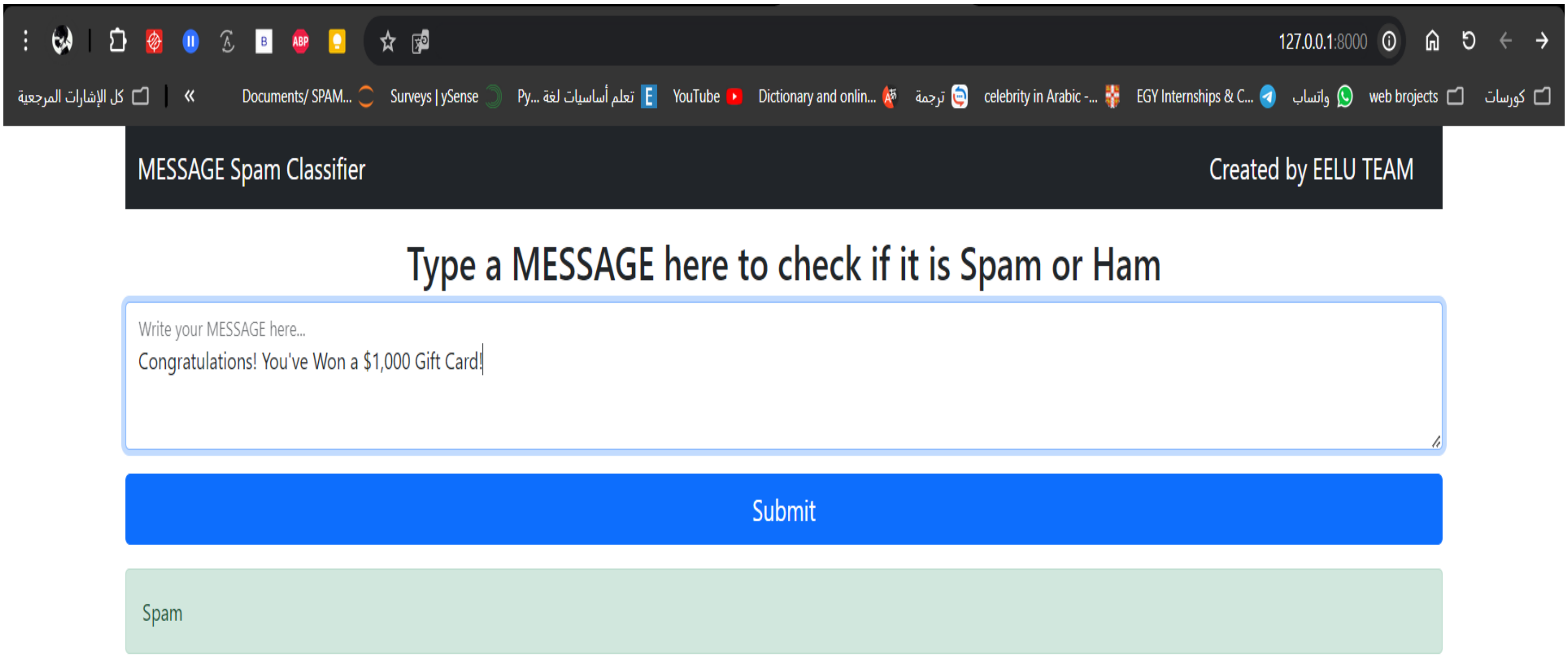
Type a MESSAGE here to check if it is Spam or Ham

Write your MESSAGE here...

Reminder: project update meeting at 10 AM tomorrow

Ham

Result Figure 5.2



MESSAGE Spam Classifier

Created by EELU TEAM

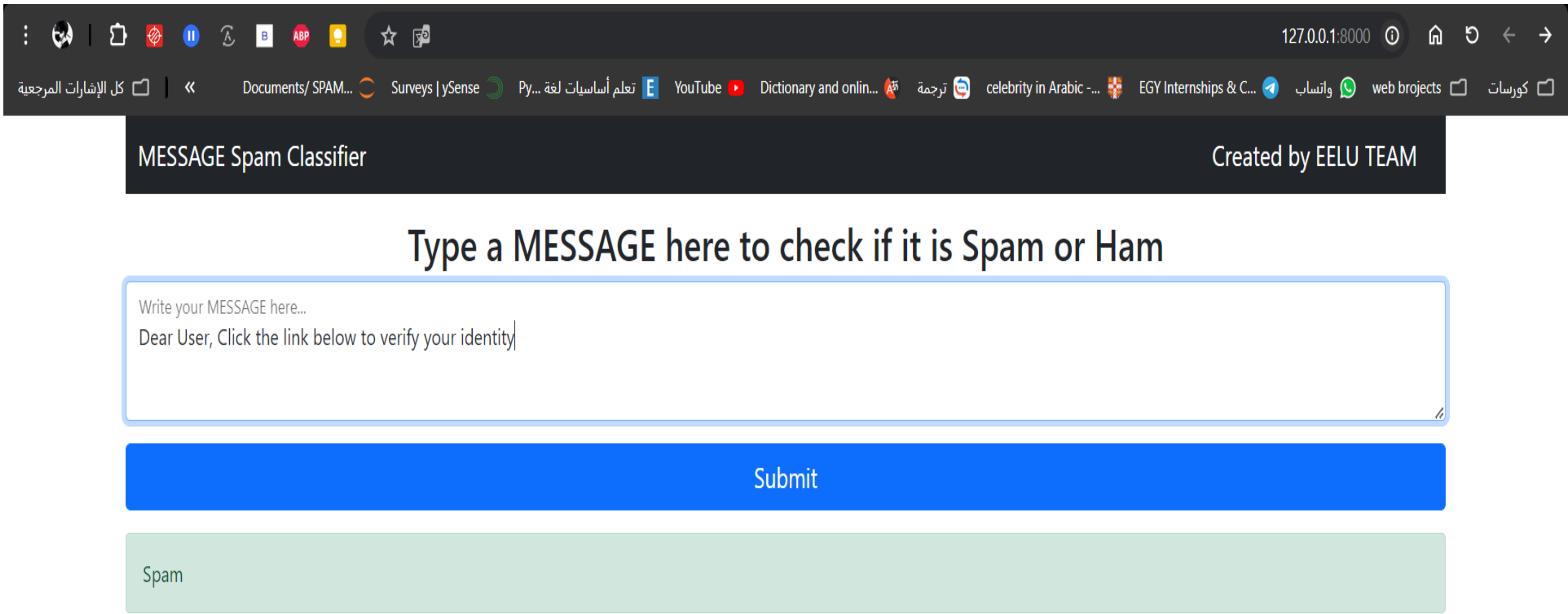
Type a MESSAGE here to check if it is Spam or Ham

Write your MESSAGE here...

Congratulations! You've Won a $1,000 Gift Card!

Spam

Result Figure 5.3



MESSAGE Spam Classifier

Created by EELU TEAM

Type a MESSAGE here to check if it is Spam or Ham

Write your MESSAGE here...

Dear User, Click the link below to verify your identity

Spam

Result Figure 5.4

6

Conclusion & Future Work

6.1 Future work

Future Directions in Spam Detection Using Machine Learning

1. **Enhancing Data Quality and Diversity**
    1. Exploring methods to improve the quality and diversity of datasets used for training spam detection models.
    2. Developing techniques to automatically update and enrich training datasets with new types of spam messages.

2. **Advanced Machine Learning Models**
    1. Investigating the use of more complex models like Deep Neural Networks (DNNs) and Generative Adversarial Networks (GANs) for better spam detection[1].
    2. Evaluating the effectiveness of **transfer learning** and **few-shot learning** approaches in spam detection.

3. **Interpretable AI for Trust and Transparency**
    1. Implementing Explainable AI (XAI) methods to make the decision-making process of spam detection models more transparent[2].
    2. Researching ways to enhance user trust by providing clear explanations for why certain messages are flagged as spam
    .
4. **Real-Time Detection and Response**
    1. Developing real-time spam detection systems that can operate at scale and respond to threats instantaneously.
    2. Creating adaptive systems that can learn from user feedback to improve accuracy over time.

5. **Cross-Platform Spam Detection**
    1. Designing unified models capable of detecting spam across various platforms, from emails to social media.
    2. Addressing the challenges of integrating spam detection systems into existing communication infrastructures.
6. **Combating Evolving Spam Techniques**
    1. Staying ahead of spammers by researching emerging spamming techniques and developing preemptive detection strategies.
    2. Collaborating with cybersecurity experts to anticipate and counteract new types of cyber threats.

7. **Legal and Ethical Considerations**
    1. Navigating the legal and ethical implications of spam detection, particularly regarding privacy and data protection.
    2. Establishing guidelines for the responsible use of machine learning in spam detection.

8. **Global Collaboration and Standardization**
    1. Fostering global collaboration among researchers, practitioners, and policymakers to standardize spam detection methodologies.
    2. Encouraging the sharing of resources and knowledge to accelerate advancements in the field.

9. **Educational Initiatives**
    1. Launching educational programs to raise awareness about spam and teach best practices for prevention.
    2. Training machine learning practitioners in specialized techniques for spam detection.

10. **Future Technologies**
    1. Exploring the potential of quantum computing and other cutting-edge technologies to revolutionize spam detection.
    2. Assessing the impact of advancements in hardware on the deployment and efficiency of spam detection systems.

This outline provides a roadmap for future research and development in the field of spam detection using machine learning, highlighting the importance of continuous innovation and adaptation to new challenges.

6.2 Conclusion

As The compelling arguments presented after the facts clearly demonstrate that spam detection is crucial. It offers users much-needed comfort by managing the overwhelming amount of information and ensuring safe data usage.

Furthermore, it provides easier access to files and messages without worrying about potential threats. The inclusion of automatic deletion features further enhances safety by removing hazardous messages. Machine learning and artificial intelligence distinguish themselves by creating a secure and user-friendly environment for managing files.

This versatile model can be seamlessly integrated into any website or software, significantly improving functionality.

Appendix A:

SVM model:

```
from google.colab import drive

drive.mount('/content/drive')



import numpy as np

import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.feature_extraction.text import CountVectorizer

from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.svm import SVC

from sklearn.metrics import accuracy_score, classification_report
import re

import string

from nltk.corpus import stopwords

from nltk.tokenize import word_tokenize

from nltk.stem import PorterStemmer

import nltk

nltk.download('stopwords')

nltk.download('punkt')
```

```
def preprocess_text(text):

   # Check if the text is NaN or not a string

   if pd.isna(text) or not isinstance(text, str):
 return ""  # Return an empty string if the text is NaN or not a string

   # Convert text to lowercase
   text = text.lower()

   # Remove numbers
   text = re.sub(r'\d+', '', text)

   # Remove punctuation
   text = text.translate(str.maketrans('', '', string.punctuation))

   # Tokenize the text
   tokens = word_tokenize(text)

   # Remove stopwords
   stop_words = set(stopwords.words('english'))

   tokens = [word for word in tokens if word not in stop_words]

   # Stemming
   stemmer = PorterStemmer()

   tokens = [stemmer.stem(word) for word in tokens]

   # Join tokens back into text
   preprocessed_text = ' '.join(tokens)

   return preprocessed_text
       };
```

```
data = pd.read_csv('/content/drive/MyDrive/Colab
Notebooks/spam_Emails_data.csv', encoding='latin1')
data.head()
```

```
data['text'] = data['text'].apply(preprocess_text)
```

```
X = data['text']

y = data['label']
```

```
vectorizer = CountVectorizer()

X_vectorized = vectorizer.fit_transform(X)
```

```
X_train, X_test, y_train, y_test =
train_test_split(X_vectorized, y, test_size=0.1,
random_state=42)
```

```
svm_model = SVC(kernel='linear')
```

```
svm_model.fit(X_train, y_train)
```

```
y_pred = svm_model.predict(X_test)
```

```
# Evaluate model performance
accuracy = accuracy_score(y_test, y_pred)

print("Accuracy:", accuracy)
```

```
def predict_new_text(new_texts, model):

    # Preprocess new texts

    new_texts_preprocessed = [preprocess_text(text) for text in
new_texts]

    # Convert new texts to the same vectorized format as the
training data

    new_texts_vectorized =
vectorizer.transform(new_texts_preprocessed)

    # Use the trained model to predict

    predictions = svm_model.predict(new_texts_vectorized)

    # Convert numeric predictions to labels

    predicted_labels = ['spam' if prediction == 1 else 'ham' for
prediction in predictions]

    # Debug: Print final labeled predictions
    print("Predicted labels:", predicted_labels)

    return predicted_labels
```

```python
def predict_new_texts(new_texts, svm_model, vectorizer):
    new_texts = [preprocess_text(text) for text in new_texts]
    new_texts_vectorized = vectorizer.transform(new_texts)
    return svm_model.predict(new_texts_vectorized)
# Test the prediction function
new_emails = [ ]
predictions = predict_new_texts(new_emails,svm_model, vectorizer)
for email, prediction in zip(new_emails, predictions):
    print(f'Email: {email}\nPrediction: {prediction}\n')
```

Appendix B:

Random Forest model:

```
from google.colab import drive

drive.mount('/content/drive')



import numpy as np

import pandas as pd

from sklearn.model_selection import train_test_split

from sklearn.feature_extraction.text import CountVectorizer

from sklearn.feature_extraction.text import TfidfVectorizer

from sklearn.svm import SVC

from sklearn.metrics import accuracy_score, classification_report
import re

import string

from nltk.corpus import stopwords

from nltk.tokenize import word_tokenize

from nltk.stem import PorterStemmer

import nltk

nltk.download('stopwords')

nltk.download('punkt')
```

```
def preprocess_text(text):

    # Check if the text is NaN or not a string

    if pd.isna(text) or not isinstance(text, str):
  return ""  # Return an empty string if the text is NaN or not a string

    # Convert text to lowercase
    text = text.lower()

    # Remove numbers
    text = re.sub(r'\d+', '', text)

    # Remove punctuation
    text = text.translate(str.maketrans('', '', string.punctuation))

    # Tokenize the text
    tokens = word_tokenize(text)

    # Remove stopwords
    stop_words = set(stopwords.words('english'))

    tokens = [word for word in tokens if word not in stop_words]

    # Stemming
    stemmer = PorterStemmer()

    tokens = [stemmer.stem(word) for word in tokens]

    # Join tokens back into text
    preprocessed_text = ' '.join(tokens)

    return preprocessed_text
```

```
data = pd.read_csv('/content/drive/MyDrive/Colab
Notebooks/spam_Emails_data.csv', encoding='latin1')
data.head()
```

```
data['text'] = data['text'].apply(preprocess_text)
```

```
X = data['text']

y = data['label']
```

```
vectorizer = CountVectorizer()

X_vectorized = vectorizer.fit_transform(X)
```

```
X_train, X_test, y_train, y_test =
train_test_split(X_vectorized, y, test_size=0.1,
random_state=42)
```

```
rf_model = RandomForestClassifier(n_estimators=100,
random_state=42)
```

```
rf_model.fit(X_train, y_train)
```

```
y_pred = rf_model.predict(X_test)
```

```
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
```

```
def predict_new_text(new_texts, model):

    # Preprocess new texts

    new_texts_preprocessed = [preprocess_text(text) for text in
new_texts]

    # Convert new texts to the same vectorized format as the
training data

    new_texts_vectorized =
vectorizer.transform(new_texts_preprocessed)

    # Use the trained model to predict
    predictions = rf_model.predict(new_texts_vectorized)

    # Convert numeric predictions to labels

    predicted_labels = ['spam' if prediction == 1 else 'ham' for
prediction in predictions]

    # Debug: Print final labeled predictions

    print("Predicted labels:", predicted_labels)

    return predicted_labels
```

```
def predict_new_texts(new_texts, rf_model, vectorizer):
    new_texts = [preprocess_text(text) for text in new_texts]
    new_texts_vectorized = vectorizer.transform(new_texts)
    return rf_model.predict(new_texts_vectorized)

# Test the prediction function
new_emails = [   ]
predictions = predict_new_texts(new_emails,rf_model, vectorizer)
for email, prediction in zip(new_emails, predictions):
    print(f'Email: {email}\nPrediction: {prediction}\n')
```

Appendix C:

```
port joblib
import os

model_path = os.path.join(os.path.dirname(__file__), 'rf_model.pkl')

victroizer_path = os.path.join(os.path.dirname(__file__), 'vectorizer.pkl')

model = joblib.load(model_path)

vectorizer = joblib.load(victroizer_path)


def predict(text):

    transformed_text = vectorizer.transform([text])

    return model.predict(transformed_text)[0]
```

```
from django.shortcuts import render
from .utils import predict
def check_spam(request):
    if request.method == 'POST'
:
        message = request.POST.get('MESSAGE')
        if not message:
            return render(request, 'index.html', {'error': 'No message provided'})
        result = predict(message)  # Use your prediction function here
        return render(request, 'index.html', {'result': result})
    return render(request, 'index.html')
```

```
<!doctype html>

<html lang="en">

<head>

  <!-- Required meta tags -->

  <meta charset="utf-8">

  <meta name="viewport" content="width=device-width, initial-scale=1">

  <!-- Bootstrap CSS -->

  <link href="https://cdn.jsdelivr.net/npm/bootstrap@5.0.0-beta2/dist/css/bootstrap.min.css"
rel="stylesheet"
    integrity="sha384-
BmbxuPwQa2lc/FVzBcNJ7UAyJxM6wuqIj61tLrc4wSX0szH/Ev+nYRRuWlolflfl"
crossorigin="anonymous">


  <title>MESSAGE Spam Classifier</title>

</head>

<body style="background :url(vv.jpg);background-repeat: no-repeat;background-size: 100% ;">

  <!-- Navbar -->

  <div class="container">

    <nav class="navbar navbar-dark bg-dark">

      <div class="container-fluid">

        <a class="navbar-brand" href="#">MESSAGE Spam Classifier</a>
        <a class="navbar-brand" href="#">Created by EELU TEAM </a>

      </div>

    </nav>
  </div>

  <!-- Input form -->

  <div class="container">

    <h2 class="text-center mt-3">Type a MESSAGE here to check if it is Spam or Ham</h2>

    <form action="/" method="POST">

      {% csrf_token %}

      <div class="form-floating">

        <textarea class="form-control" placeholder="Write your MESSAGE here.." id="MESSAGE" name="MESSAGE" style="height: 100px"></textarea>

        <label for="MESSAGE">Write your MESSAGE here...</label>

        <button class="w-100 btn btn-lg btn-primary mt-3">Submit</button>

      </div>

      {% if error %}

        <div class="alert alert-danger mt-3">{{ error }}</div>

      {% endif %}

      {% if result %}
```

```
        <div class="alert alert-success mt-3">{{ result }}</div>

      {% endif %}

    </form>

  </div>

</body>
</html>
```

References:

2. Dai, Yuehao, and Ruixun Zhang. “Estimating Market Liquidity from Daily Data: Marrying Microstructure Models and Machine Learning.” SSRN Electronic Journal, 2023. https://doi.org/10.2139/ssrn.4371650.

3. Fisichella, Marco, and Filippo Garolla. “Can Deep Learning Improve Technical Analysis of Forex Data to Predict Future Price Movements?” IEEE Access 9 (2021): 153083–101. https://doi.org/10.1109/access.2021.3127570.

4. Hossain, Emam, Mohammad Shahadat Hossain, Pär-Ola Zander, and Karl Andersson. “Machine learning with Belief Rule-Based Expert Systems to predict stock price movements.” Expert Systems With Applications 206 (November 2022): 117706. https://doi.org/10.1016/j.eswa.2022.117706.

5. Price, Joseph, Kasey Buckles, Jacob Van Leeuwen, and Isaac Riley. “Combining family history and machine learning to link historical records: The Census Tree data set.” Explorations in Economic History 80 (April 2021): 101391. https://doi.org/10.1016/j.eeh.2021.101391.

6. Sheng, Yankai, Yuanyu Qu, and Ding Ma. “Stock Price Crash Prediction Based on Multimodal Data Machine Learning Models.” SSRN Electronic Journal, 2023. https://doi.org/10.2139/ssrn.4575784.

7. “Stock price crash prediction based on multimodal data machine learning models.” Finance Research Letters 62 (April 2024): 105195.
https://doi.org/10.1016/j.frl.2024.105195

.

8. “Estimating Market Liquidity from Daily Data: Marrying Microstructure Models and Machine Learning.” SSRN Electronic Journal, 2023.
https://doi.org/10.2139/ssrn.4371650

Symantec’s Internet Security Threat Report (2021) provides analysis on emerging trends in attacks, malicious code activity, phishing, and spam
https://www.insight.com/en_US/content-and-resources/brands/symantec/internet-security-threat-report.html

Verizon’s Data Breach Investigations Report (2021) analyzes 29,207 quality incidents, of which 5,258 were confirmed breaches, highlighting the increase in phishing and ransomware attacks2.
https://www.verizon.com/about/news/verizon-2021-data-breach-investigations-report

Microsoft AI Lab explored the use of Support Vector Machines (SVMs) for spam detection, demonstrating their performance in identifying spam messages
https://mashuai-ms.github.io/pubs/tdsc2021.pdf

.