A

Major Project report

on

# SPAM EMAIL DETECTION SYSTEM

*(Submitted in partial fulfilment of the requirements for the award of the degree of)*

**BACHELOROFTECHNOLOGY**

**IN**

**COMPUTER SCIENCE AND ENGINEERING**

**(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)**

**BY**


**GNANESHWAR.B (22C91A6619)**

**AHISHEK.B (22C91A6620)**

**AKSHITH.J (22C91A6652)**


Under The Esteemed Guidance Of

**MR. SAGAR**

**Assistant Professor**


**Department of CSE (Artificial Intelligence & Machine Learning)**

**HOLYMARYINSTITUTEOFTECHNOLOGY&SCIENCE**

**(UGC AUTONOMOUS)**

***(Approved by AICTE, New Delhi, and Permanent Affiliated to JNTUH Hyderabad, Accredited by NAAC 'A' Grade)***

**Bogaram (V), Keesara (M), Medchal-Malkajgiri(Dist)-501301, TG**


**2024-2025**

# HOLY MARY INSTITUTE OF TECHNOLOGY & SCIENCE

**(UGCAUTONOMOUS)**

*(Approved by AICTE New Delhi, Permanently Affiliated to JNTU Hyderabad, Accredited by NAAC with 'A' Grade)*

**Bogaram (V), Keesara (M), Medchal-Malkajgiri(Dist.)-501301, TG.**

## DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING(ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)

## CERTIFICATE

This is to certify that the major project entitled " **SPAM EMAIL DETECTION SYSTEM.** 'is being submitted by **GNANESHWAR .B (22C91A6619), ABHISHEK .B ( 22C91A6620) , AKSHITH . J. (22 C91A 6652)** 'in Partial fulfillment of the academic requirements for the award of the degree of Bachelor of Technology in "COMPUTER SCIENCEAND ENGINEERING (ARTIFICIAL INTELLIGENCE & MACHINE LEARNING)" from HOLY MARY INSTITUTE OFTECHNOLOGY &SCIENCE (UGC AUTONOMOUS),during the year **2024- 2025.**

| INTERNAL GUIDE | HEAD OF THE DEPARTMENT |
|---|---|
| Mr. Sagar | Mrs. A. Akhila |
| Assistant Professor | Assistant Professor& HOD |
| Dept. of Computer Science & Engineering (Artificial Intelligence & Machine Learning) | Dept. of Computer Science & Engineering (Artificial Intelligence & Machine Learning) |

**EXTERNAL EXAMINER**

## ACKNOWLEDGEMENT

The satisfaction and euphoria that accompany the successful completion of any task would be

incomplete without the mention of the people who made it possible, who's constant guidance and encouragement crowns all effort with success.

We take this opportunity to express my profound gratitude and deep regards to our Project coordinator and Guide **Mr. K.Sagar, Assistant Professor,** Dept. of **Computer Science & Engineering (Artificial Intelligence & Machine Learning),** Holy Mary Institute of Technology & Science for his / her exemplary guidance, monitoring and constant encouragement throughout the project work.

Our special thanks to **Mrs. A. Akhila, Head of The Department**, Dept. of **Computer Science & Engineering (Artificial Intelligence & Machine Learning),** Holy Mary Institute of Technology & Science, who has given immense support throughout the course of the project.

We also thank **Dr. J. B. V. Subramanyam**,the **Honorable Principal** of my college Holy Mary Institute of Technology & Science for providing me the opportunity to carry out this work.

At the outset, we express my deep sense of gratitude to the beloved **Chairman A. Siddartha Reddy** of **Holy Mary Institute of Technology & Science**, for giving me the opportunity to complete my course of work.

We are obliged to **Staff members** of Holy Mary Institute of Technology & Science for the valuable information provided by them in their respective fields. We are grateful for their cooperation during the period of my assignment.

Last but not the least we thank our **Parents** and **Friends** for their constant encouragement without which this assignment not be possible.

**GNANESHWAR.B ( 22C91A6619)**
**ABHISHEK .B (22C91A6620)**
**AKSHITH.J. (22C91A6652)**

## DECLARATION

This is to certify that the work reported in the present project titled " **SPAM EMAIL DETECTION SYSTEM** "is a record of work done by us in the

Department of Computer Science & Engineering (Artificial Intelligence & Machine Learning), Holy Mary Institute of Technology and Science.

To the best of our knowledge no part of the this is copied from books/journals/internet and wherever the portion is taken, the same has been duly referred to in the text. The reports are based on the project work done entirely by us not copied from any other source.

**GNANESHWAR.B (22C91A6619)**
**ABHISHEK.B (22C91A6620)**
**AKSHITH.J (22C91A6652)**

# INTRODUCTION

In today's digital age, email has become a crucial mode of communication for individuals and organizations. However, the rise of spam emails has become a significant concern, causing inconvenience, wasting time, and posing security risks. A Spam Email Detection System is designed to identify and filter out unwanted and unsolicited emails, ensuring a safer and more efficient email experience.

## What is Spam?

Spam emails are unsolicited and unwanted emails sent to a large number of recipients, often with malicious intent, such as:

- Phishing: attempting to trick recipients into revealing sensitive information
- Malware: spreading malicious software to compromise systems
- Advertising: promoting products or services without consent

## How does the Spam Email Detection System Works?

The system uses a combination of techniques, including:

- *Machine Learning:* training algorithms to recognize patterns in spam emails
- *Natural Language Processing (NLP):* analyzing email content to identify spam characteristics
- *Rule-based Filtering:* applying predefined rules to detect spam emails

## Benefits of Spam Email Detection System

- *Improved Email Security:* reducing the risk of phishing and malware attacks
- *Increased Productivity:* saving time by filtering out unwanted emails
- *Enhanced User Experience:* providing a safer and more efficient email experience

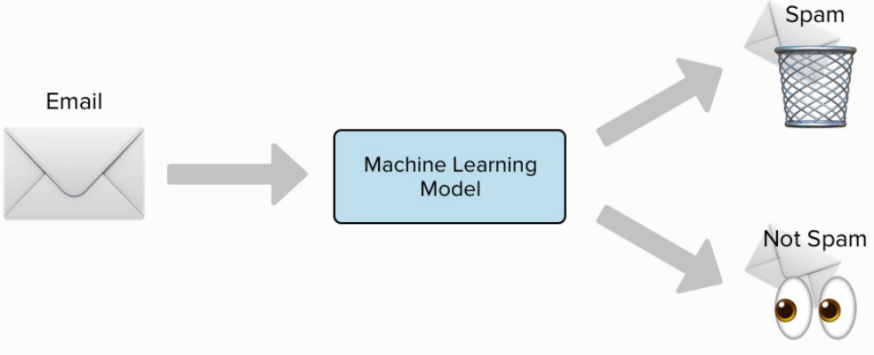
# Objective

The primary objective of a Spam Email Detection System is to:

1. *Identify and filter out Spam Emails:* accurately detect and block unwanted, unsolicited, and malicious emails.
2. *Protect Users:* prevent users from receiving spam emails that can be annoying, misleading, or malicious.
3. *Improve Email Security:* reduce the risk of email-borne threats, such as phishing, malware, and ransomware.
4. *Enchance Your Experience:* improve the overall email experience for users by reducing the amount of spam emails in their inbox.
5. *Reduce Costs:* minimize the costs associated with dealing with spam emails, such as lost productivity and IT support.

## Secondary objective

1. *Improve Accuracy* continuously improve the accuracy of spam email detection to reduce false positives and false negatives.
2. *Adapt to Evolving Treats:* stay up-to-date with new spam patterns and techniques to ensure the system remains effective.
3. *Provide Customization Options:* offer users the ability to customize their spam filtering settings and preferences.



# Types of Spam Emails

1. *Phishing Emails:* Emails that attempt to trick users into revealing sensitive information, such as passwords or credit card numbers.
2. *Spammy Promotions:* Emails that promote products or services in a spammy or unsolicited manner.
3. *Malware Emails:* Emails that contain malware or viruses, designed to harm the user's device or steal sensitive information.
4. *Scams:* Emails that attempt to deceive users into sending money or providing sensitive information, often using fake or misleading information.
5. *Spammy Newsletters:* Emails that are sent to users without their consent, often containing irrelevant or unwanted content.
6. *Social Engineering Emails:* Emails that attempt to manipulate users into performing certain actions or revealing sensitive information, often using psychological tactics.
7. *Spammy Messages:* Emails that contain irrelevant or unwanted content, often sent in bulk to multiple recipients.
8. *Advance Fee Scams:* Emails that promise users a benefit or reward in exchange for an upfront payment or fee.
9. *Lottery or Prize Scams:* Emails that claim the user has won a prize or lottery, but requires them to provide sensitive information or pay a fee to claim the prize.
10. *Fake Invoice or Payment Emails:* Emails that appear to be legitimate invoices or payment requests, but are actually scams designed to trick users into sending money.

# Scope

The scope of a Spam Email Detection System is vast and multifaceted. Here are some key areas where such a system can be applied:

1. *Email Service Providers:* Spam email detection systems can be integrated into email service providers' infrastructure to filter out spam emails and improve user experience.
2. *Organizational Email Systems:* Companies and organizations can use spam email detection systems to protect their employees' email accounts from spam and phishing attacks.
3. *Personal Email Accounts:* Individuals can use spam email detection systems to filter out unwanted emails and improve their personal email experience.
4. *Cybersecurity:* Spam email detection systems can be used as a component of a larger cybersecurity solution to protect against email-borne threats.
5. *Research and Development:* Researchers can use spam email detection systems to study and analyze spam patterns, develop new detection techniques, and improve existing systems.

# How Spam Email Detection System Works

The Spam Email Detection System uses a combination of techniques to identify and filter out spam emails. Here's a step-by-step explanation of how it works:

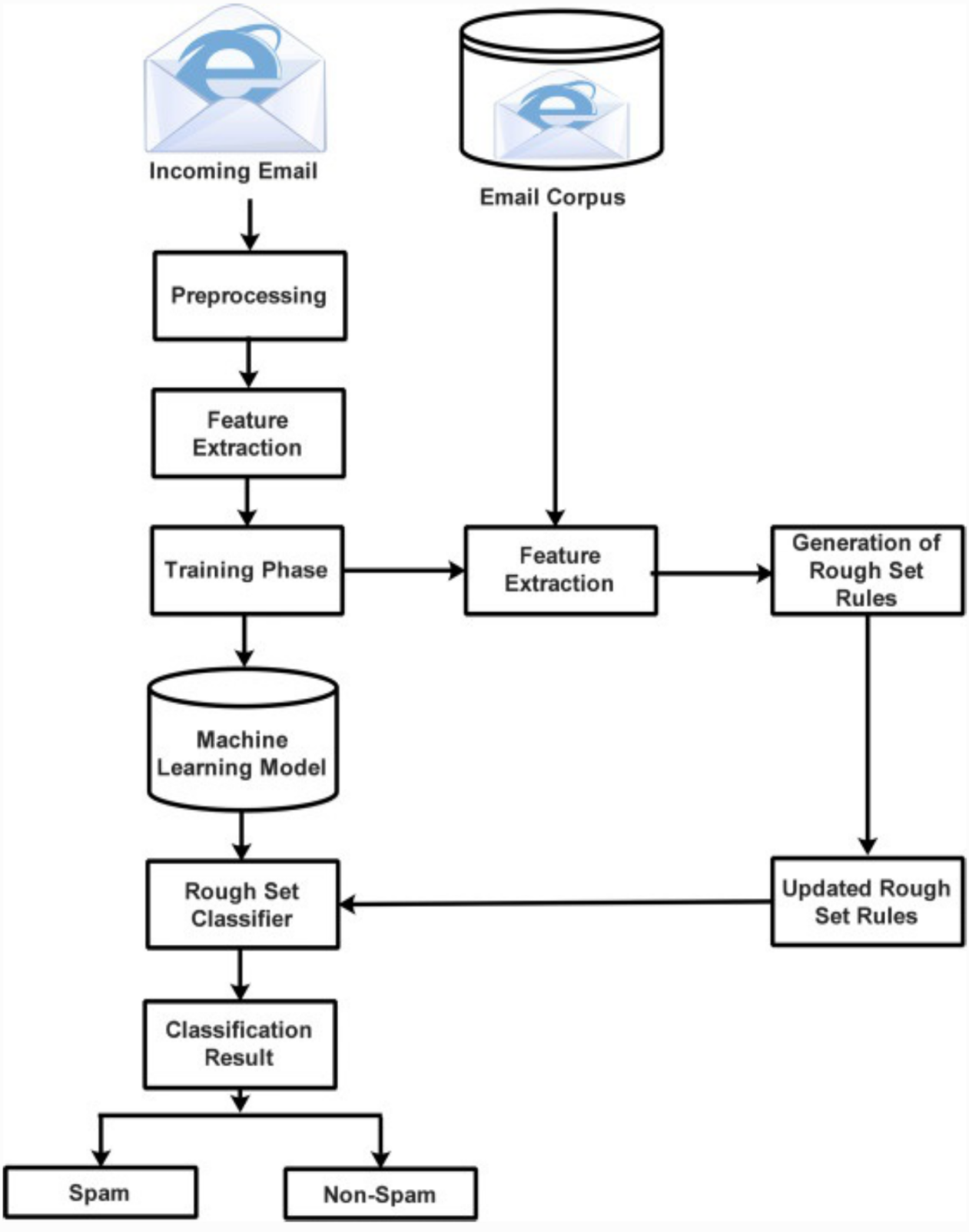
1. *Email Receipt:* The system receives incoming emails from various sources, such as email servers or clients.
2. *Preprocessing:* The system preprocesses the emails by extracting relevant features, such as:
    - *Sender Information:* sender's email address, IP address, and domain
    - *Email Content:* subject, body, and attachments
    - *Metadata:* email headers, timestamps, and other metadata
3. *Feature Extraction:* The system extracts specific features from the preprocessed emails, such as:
    - *Keywords:* specific words or phrases commonly found in spam emails
    - *Patterns:* patterns in the email content, such as suspicious links or attachments
    - *Behavioral Analysis:* analysis of the sender's behavior, such as sending patterns and recipient lists
4. *Machine Learning:* The system uses machine learning algorithms to analyze the extracted features and classify the emails as spam or legitimate. The algorithms can be trained on large datasets of labeled emails to improve accuracy.
5. *Rule_ Based Filtering:* The system applies predefined rules to detect spam emails, such as:
    - *Blacklisting:* blocking emails from known spammer IP addresses or domains
    - *Whitelisting:* allowing emails from trusted senders or domains
    - *Keyword Filtering:* blocking emails containing specific keywords or phrases
6. *Classification:* The system classifies the emails as spam or legitimate based on the analysis and filtering results.
7. *Action:* The system takes action on the classified emails, such as:
    - *Blocking:* blocking spam emails from reaching the recipient's inbox
    - *Quarantining:* quarantining suspicious emails for further analysis or review
    - *Notification:* notifying the recipient of potential spam emails

# Methodology

The methodology of a Spam Email Detection System involves the following steps:

1. *Data Collection:* Gathering a large dataset of labeled emails (spam or legitimate) to train and test the system.
2. *Data Preprocessing:* Preprocessing the collected data to extract relevant features, such as:
    - Tokenization: breaking down text into individual words or tokens
    - Stopword removal: removing common words like "the," "and," etc.
    - Stemming or Lemmatization: reducing words to their base form
3. *Feature Extraction:* Extracting relevant features from the preprocessed data, such as:
    - Bag-of-Words: representing emails as a bag of words
    - Term Frequency-Inverse Document Frequency (TF-IDF): weighing word importance
    - Sentiment Analysis: analyzing email tone and sentiment
4. *Model Selection:* Selecting a suitable machine learning algorithm, such as:
    - Supervised Learning: Naive Bayes, Support Vector Machines, Random Forest
    - Unsupervised Learning: clustering, dimensionality reduction
5. *Model Training:* Training the selected model using the labeled dataset.
6. *Model Evaluation:* Evaluating the trained model's performance using metrics like accuracy, precision, recall, and F1-score.
7. *Model Deployment:* Deploying the trained model in a production environment to classify new emails.
8. *Continuous Improvement:* Continuously updating the model with new data and retraining it to adapt to evolving spam patterns.

# Flowchart


Incoming Email
Email Corpus
Preprocessing
Feature Extraction
Training Phase
Feature Extraction
Generation of Rough Set Rules
Machine Learning Model
Rough Set Classifier
Updated Rough Set Rules
Classification Result
Spam
Non-Spam

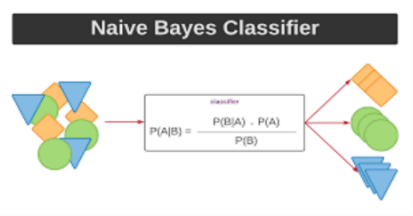
# System Development

The system development of a Spam Email Detection System involves the following stages:

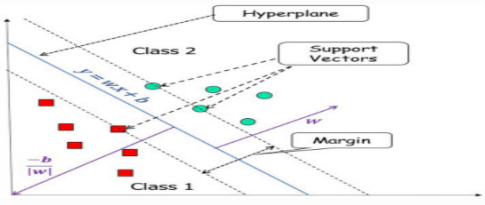
1. *Requirements Gathering:* Identifying the system's functional and non-functional requirements, such as:
    - Accuracy
    - Efficiency
    - Scalability
    - Security
2. *System Design:* Designing the system's architecture, including:
    - Data collection and preprocessing
    - Feature extraction and selection
    - Model training and evaluation
    - Deployment and maintenance
3. *Data Collection:* Gathering a large dataset of labeled emails (spam or legitimate) to train and test the system.
4. *Model Development:* Developing a machine learning model that can accurately classify emails as spam or legitimate.
5. *System Implementation:* Implementing the system using a programming language, such as Python or Java.
6. *Testing and Evaluation:* Testing the system's performance using metrics like accuracy, precision, recall, and F1-score.
7. *Deployment:* Deploying the system in a production environment to classify new emails.
8. *Maintenance and Update:* Continuously updating the system with new data and retraining the model to adapt to evolving spam patterns.

# Machine Learning Algorithms

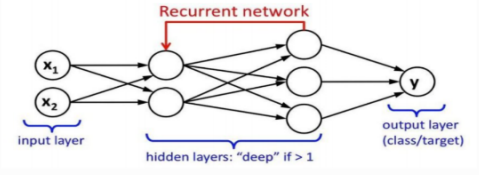
1. *Naive bayes:* a popular machine learning algorithm for classification tasks, often used for spam detection due to its simplicity and effectiveness.



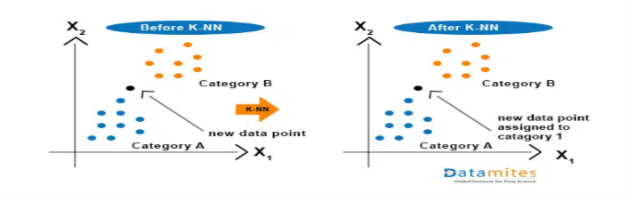

2. *Support Vector Machines ( SVM):* a powerful algorithm for classification tasks, can be used for spam detection by finding the optimal hyperplane to separate spam and legitimate emails.




3. *Random Forest:* an ensemble learning algorithm that combines multiple decision trees to improve the accuracy and robustness of spam detection.
4. *GradeintBoosting:* an ensemble learning algorithm that combines multiple weak models to create a strong predictive model for spam detection.
5. *Deep Learning:* algorithms such as Convolutional Neural Networks (CNN) and Recurrent Neural Networks (RNN) can be used for spam detection by learning complex patterns in email content.




6. *K-Nearest Neighbors( KNN):* a simple algorithm that classifies emails based on their similarity to known spam or legitimate emails.




7. *Decision Trees* a simple algorithm that uses a tree-like model to classify emails based on their features.

# Tools and Techniques

*Machine Learning Tools:*
1. *scikit-learn:* a popular Python library for machine learning
2. *TensorFlow:* an open-source machine learning library developed by Google
3. *PyTorch:* an open-source machine learning library developed by Facebook

*Natural Language Processing (NLP) Tools:*
1. *NLTK (Natural Language Toolkit):* a popular Python library for NLP
2. *spaCy:* a modern Python library for NLP that focuses on performance and ease of use
3. *Gensim:* a Python library for topic modeling and document similarity analysis

*Email Parsing Tools:*
1. *email-parser:* a Python library for parsing email messages
2. *imaplib:* a Python library for accessing and manipulating email messages on an IMAP server

*Programming Languages:*
1. *Python:* a popular language used for machine learning, NLP, and email parsing
2. *R:* a language used for statistical computing and machine learning

*Techniques:*
1. *Supervised Learning:* training machine learning models on labeled datasets
2. *Unsupervised Learning:* using clustering, dimensionality reduction, and other techniques to identify patterns in unlabeled data
3. *Deep Learning:* using neural networks to learn complex patterns in data
4. *Rule-based Systems:* using predefined rules to detect spam emails

# Evaluation Metrics

1. *Accuracy:* measures the proportion of correctly classified emails (spam or legitimate).
2. *Precision:* measures the proportion of true positives (spam emails correctly identified) among all positive predictions.
3. *Recall:* measures the proportion of true positives among all actual spam emails.
4. *F1-Score:* harmonic mean of precision and recall, providing a balanced measure of both.
5. *False Positive Rate (FPR):* measures the proportion of legitimate emails incorrectly classified as spam.
6. *False Negative Rate (FNR):* measures the proportion of spam emails incorrectly classified as legitimate.
7. *Area Under the ROC Curve (AUC-ROC):* measures the model's ability to distinguish between spam and legitimate emails.
8. *Mean Average Precision (MAP):* measures the average precision at different recall levels.

## Why these Metrics matters ?

1. *Accuracy:* ensures the model is correctly classifying emails.
2. *Precision and Recall:* balance between detecting spam emails and avoiding false positives.
3. *F1-Score:* provides a single metric to evaluate the model's performance.
4. *FPR and FNR:* help identify potential issues with the model.

*Using these metrics:*

1. *Model evaluation:* evaluate the performance of different machine learning models.
2. *Model selection:* select the best model based on performance metrics.
3. *Hyperparameter tuning:* optimize model hyperparameters to improve performance.
4. *Monitoring:* continuously monitor the model's performance and adjust as needed.

# Source Code

```python
import pandas as pd
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.model_selection import train_test_split
from sklearn.naive_bayes import MultinomialNB
from sklearn.metrics import accuracy_score, classification_report

#Sample dataset
data = {
    "email": [
        "You have won a prize! Click here to claim it.",
        "Meeting on Friday at 2 PM.",
        "Get free cash now! Click here.",
        "Project update: please review the document.",
        "You are a winner! Claim your prize now.",
        "Team lunch on Wednesday.",
        "Make money fast! Click here.",
        "New policy: please read the document.",
    ],
    "label": ["spam", "not spam", "spam", "not spam", "spam", "not spam", "spam", "not spam"],
}

df = pd.DataFrame(data)

#Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(df["email"], df["label"], test_size=0.2, random_state=42)

#Create a CountVectorizer object
vectorizer = CountVectorizer()

#Fit the vectorizer to the training data and transform both the training and testing data
X_train_count = vectorizer.fit_transform(X_train)
X_test_count = vectorizer.transform(X_test)

#Train a Naive Bayes classifier on the training data
clf = MultinomialNB()
clf.fit(X_train_count, y_train)
```

```python
#Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(df["email"], df["label"], test_size=0.2, random_state=42)

#Create a CountVectorizer object
vectorizer = CountVectorizer()

#Fit the vectorizer to the training data and transform both the training and testing data
X_train_count = vectorizer.fit_transform(X_train)
X_test_count = vectorizer.transform(X_test)

#Train a Naive Bayes classifier on the training data
clf = MultinomialNB()
clf.fit(X_train_count, y_train)

#Make predictions on the testing data
y_pred = clf.predict(X_test_count)

#Evaluate the accuracy of the classifier
accuracy = accuracy_score(y_test, y_pred)
print("Accuracy:", accuracy)
print("Classification Report:")
print(classification_report(y_test, y_pred))

#Define a function to classify new emails
def classify_email(email):
    email_count = vectorizer.transform([email])
    prediction = clf.predict(email_count)
    return prediction[0]

#Test the function
email1 = "You have won a prize! Click here to claim it."
email2 = "Meeting on Friday at 2 PM."
print(f"Email 1: {email1} -> {classify_email(email1)}")
print(f"Email 2: {email2} -> {classify_email(email2)}")
```

# Output

Accuracy: 0.0
Classification Report:

| | precision | recall | f1-score | support |
|---|---|---|---|---|
| not spam | 0.00 | 0.00 | 0.00 | 2.0 |
| spam | 0.00 | 0.00 | 0.00 | 0.0 |
| accuracy | | | 0.00 | 2.0 |
| macro avg | 0.00 | 0.00 | 0.00 | 2.0 |
| weighted avg | 0.00 | 0.00 | 0.00 | 2.0 |

Email 1: You have won a prize! Click here to claim it. -> spam
Email 2: Meeting on Friday at 2 PM. -> spam

# Conclusion

In conclusion, a Spam Email Detection System is a crucial tool for filtering out unwanted and malicious emails, protecting users from phishing, malware, and other online threats. By leveraging machine learning algorithms, natural language processing techniques, and rule-based systems, these systems can accurately classify emails as spam or legitimate, reducing the risk of email-borne threats.

The development of a Spam Email Detection System involves several key steps, including data collection, feature extraction, model training, and evaluation. By carefully selecting and tuning these components, developers can build systems that are highly effective at detecting spam emails.

As spam emails continue to evolve and become more sophisticated, it is essential to continuously update and improve Spam Email Detection Systems. This can be achieved through ongoing research and development, incorporating new techniques and algorithms, and adapting to emerging trends and threats.

By implementing effective Spam Email Detection Systems, individuals and organizations can protect themselves from the risks associated with spam emails, improve email security, and reduce the time and resources wasted on dealing with unwanted emails.