MINI PROJECT REPORT

TITLE:

Spam Mail Classifier Using Machine Learning

DEPARTMENT:

Computer Science and Engineering (AI & ML)

STUDENT DETAILS:

Name: THIRUVASAGAN R

Register Number: 2117240030160

---

1. ABSTRACT

Email is one of the most widely used communication tools today, but it is also a major target for spam messages that waste time and resources. A spam mail classifier is a machine learning system that automatically identifies and filters out unwanted or harmful emails.

In this project, a supervised learning model is developed to classify emails as either Spam or Ham (Not Spam) based on their content. The system uses Natural Language Processing (NLP) techniques to extract meaningful features from email text, and a Naive Bayes Classifier is trained on a labeled dataset. The model helps in reducing the amount of junk mail and improves email productivity.

---

2. INTRODUCTION

Spam emails are unsolicited messages sent in bulk to multiple recipients, often for advertising or phishing purposes. Detecting spam manually is impractical, so automated classification using machine learning has become essential.

Machine Learning models can analyze text patterns and predict whether a message is spam or not based on previous data. By applying text preprocessing, feature extraction (TF–IDF), and a classification algorithm, spam detection becomes efficient and accurate.

---

3. OBJECTIVES

To build a machine learning model that classifies emails as spam or not spam.

To apply natural language processing for text cleaning and feature extraction.

To evaluate the performance of the classifier using accuracy, precision, and recall.

To demonstrate automation in email filtering through AI-based techniques.

---

4. EXISTING SYSTEM

Traditional spam filters rely on rule-based or keyword-based detection. While effective to some extent, they fail when spammers modify their language patterns or use obfuscation. These systems also require constant manual updates and are not adaptive.

---

5. PROPOSED SYSTEM

The proposed system uses a Naive Bayes Classifier trained on a labeled dataset of spam and non-spam emails. Text is cleaned and converted into numerical vectors using TF–IDF (Term Frequency – Inverse Document Frequency). The trained model then predicts whether a new email is spam or not based on its content.

This approach adapts automatically as new data becomes available and offers high accuracy in real-world scenarios.

---

6. SYSTEM REQUIREMENTS

Software Requirements:

Python 3.8+

Jupyter Notebook / Google Colab

Libraries: scikit-learn, pandas, numpy, matplotlib, nltk

Hardware Requirements:

Processor: Intel i3 or above

RAM: 4 GB minimum

Storage: 2 GB free space

---

7. SYSTEM DESIGN

Modules:

1. Data Collection:

Dataset of spam and ham emails (e.g., from Kaggle).

2. Data Preprocessing:

Remove punctuation, stopwords, and special symbols.

Convert text to lowercase.

Tokenization and stemming.

3. Feature Extraction:

Use TF–IDF Vectorizer to convert text into numeric features.

4. Model Training:

Apply Naive Bayes algorithm to classify spam and ham.

5. Evaluation:

Measure accuracy, precision, recall, and confusion matrix.

---

8. IMPLEMENTATION

Python Code:

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

# Load dataset
data = pd.read_csv("spam.csv", encoding='latin-1')[['v1','v2']]
data.columns = ['label', 'message']

# Convert labels to binary
data['label'] = data['label'].map({'ham': 0, 'spam': 1})

# Split data
X_train, X_test, y_train, y_test = train_test_split(data['message'], data['label'],
test_size=0.2, random_state=42)

# Vectorize text
tfidf = TfidfVectorizer(stop_words='english', max_df=0.7)
X_train_tfidf = tfidf.fit_transform(X_train)
X_test_tfidf = tfidf.transform(X_test)

# Train model
model = MultinomialNB()
model.fit(X_train_tfidf, y_train)
```

```
# Predict
y_pred = model.predict(X_test_tfidf)

# Evaluation
print("Accuracy:", accuracy_score(y_test, y_pred))
print("Confusion Matrix:\n", confusion_matrix(y_test, y_pred))
print("Classification Report:\n", classification_report(y_test, y_pred))
```

---

9. OUTPUT

Accuracy: Around 97–98%

Confusion Matrix: Shows correct vs incorrect predictions

Graphical Results:

Bar chart comparing predicted vs actual labels (optional for visualization)

---

10. CONCLUSION

The spam mail classifier successfully distinguishes spam emails from legitimate ones using machine learning. The Naive Bayes algorithm combined with TF–IDF vectorization provides a lightweight yet effective solution.

This project demonstrates how AI can enhance cybersecurity and improve user productivity by filtering unwanted emails automatically.

---

11. FUTURE ENHANCEMENTS

Integrate real-time spam detection in email clients.

Use deep learning models such as LSTM for improved accuracy.

Include multilingual spam detection support.

Develop a web-based or mobile app for live classification.

---

12. REFERENCES

1. Kaggle Spam Collection Dataset – https://www.kaggle.com/uciml/sms-spam-collection-dataset

2. Scikit-learn Documentation – https://scikit-learn.org/

3. NLTK Natural Language Toolkit – https://www.nltk.org/

4. https://github.com/thiruvasaga-n/Spam-Mail-Classifier-Using-Machine-Learning