**Spam Detection Using Machine Learning: A Naïve Bayes Approach**

**Title Page**

* **Title:** Spam Detection Using Machine Learning: A Naïve Bayes Approach
* **Authors:** Sujal Mali, Harsh Shubhedar
* **Institution:** Mulund College of Commerce
* **Date:** 06 May 2025

**Abstract**

Spam emails present a persistent challenge to cybersecurity, often leading to data breaches and user inconvenience. This paper explores a machine learning-based spam classification model using the Multinomial Naïve Bayes algorithm alongside TF-IDF vectorization. The dataset is sourced from tab-separated email text, with preprocessing techniques applied for feature extraction. Evaluation of the model demonstrates competitive accuracy rates, highlighting its efficiency in spam filtering. Ethical implications and future advancements in spam classification are discussed.

**Problem Statement & Objective**

Spam emails serve as a vehicle for phishing attacks and fraudulent activities, causing financial and security concerns. Traditional spam filters are rule-based and struggle to adapt to evolving spam patterns. This study aims to develop a robust, data-driven spam detection model utilizing machine learning principles, ensuring accurate email classification while minimizing false positives.

**Literature Review**

Spam filtering techniques range from simple rule-based approaches to advanced AI-driven solutions. Studies have shown that probabilistic models, particularly Naïve Bayes classifiers, outperform traditional filtering methods due to their ability to adapt to textual patterns. TF-IDF (Term Frequency-Inverse Document Frequency) enhances feature representation by assigning importance to specific words in emails. Prior research has demonstrated the efficiency of TF-IDF and Naïve Bayes in spam classification tasks, achieving notable accuracy levels.

**Research Methodology**

The methodology follows a systematic approach:

* **Dataset Preparation:** Extracting structured email data from tab-separated text files.
* **Data Splitting:** Partitioning the dataset into training (80%) and testing (20%) subsets.
* **Feature Extraction:** Implementing TF-IDF vectorization to transform text into numerical feature vectors.
* **Model Training:** Employing the Multinomial Naïve Bayes algorithm for classification.
* **Evaluation Metrics:** Calculating accuracy, precision, recall, and F1-score for performance assessment.

**Tool Implementation**

The implementation is carried out using Python and essential machine learning libraries:

* **Pandas** – Data manipulation and preprocessing
* **Scikit-learn** – Text vectorization, classification, and evaluation
* **Joblib** – Model and vectorizer storage

**Code Implementation**

import pandas as pd

from sklearn.feature\_extraction.text import TfidfVectorizer

from sklearn.model\_selection import train\_test\_split

from sklearn.naive\_bayes import MultinomialNB

from sklearn.metrics import accuracy\_score, precision\_score, recall\_score, f1\_score

import joblib

# Load dataset

data = pd.read\_csv('email.txt', sep='\t', header=None, names=['text', 'label'])

# Split data

X\_train, X\_test, y\_train, y\_test = train\_test\_split(data['text'], data['label'], test\_size=0.2, random\_state=42)

# Text vectorization

vectorizer = TfidfVectorizer(stop\_words='english')

X\_train\_vectors = vectorizer.fit\_transform(X\_train)

X\_test\_vectors = vectorizer.transform(X\_test)

# Train model

model = MultinomialNB()

model.fit(X\_train\_vectors, y\_train)

# Evaluate

predictions = model.predict(X\_test\_vectors)

accuracy = accuracy\_score(y\_test, predictions)

precision = precision\_score(y\_test, predictions, pos\_label=1)

recall = recall\_score(y\_test, predictions, pos\_label=1)

f1 = f1\_score(y\_test, predictions, pos\_label=1)

print(f'Model Accuracy: {accuracy:.2f}')

print(f'Precision: {precision:.2f}')

print(f'Recall: {recall:.2f}')

print(f'F1 Score: {f1:.2f}')

# Save model

joblib.dump(model, 'model.pkl')

joblib.dump(vectorizer, 'vectorizer.pkl')

**Results & Observations**

The spam classifier was evaluated using multiple performance metrics:

| **Metric** | **Score** |
| --- | --- |
| Accuracy | 92% |
| Precision | 88% |
| Recall | 85% |
| F1 Score | 86% |

The high accuracy and precision indicate effective spam detection capabilities, while the recall score suggests minor limitations in detecting certain spam types. The TF-IDF vectorization significantly improves classification by assigning greater importance to spam-related terms.

**Ethical Impact & Market Relevance**

Spam detection models must be optimized to minimize false positives, preventing legitimate emails from being misclassified. Ethical concerns surrounding biased training data need further exploration to avoid discrimination in email filtering. In market applications, integrating AI-driven spam filtering enhances cybersecurity and improves email usability, reducing phishing attack risks.

**Future Scope**

Further enhancements to spam classification include:

* **Deep Learning Integration:** Utilizing transformer-based NLP models such as BERT for improved text analysis.
* **Adaptive Learning:** Implementing reinforcement learning for real-time spam pattern updates.
* **Multilingual Filtering:** Expanding detection capabilities for non-English spam emails.
* **Cloud-Based Deployment:** Developing scalable spam detection systems for enterprise applications.

**References**

1. Goodfellow, I., Bengio, Y., & Courville, A. (2016). *Deep Learning*. MIT Press.
2. Jurafsky, D., & Martin, J. H. (2019). *Speech and Language Processing*. Pearson.
3. Manning, C., Raghavan, P., & Schütze, H. (2008). *Introduction to Information Retrieval*. Cambridge University Press.
4. Mohri, M., Rostamizadeh, A., & Talwalkar, A. (2018). *Foundations of Machine Learning*. MIT Press.
5. Spam detection studies from IEEE Xplore and ACM Digital Library.