Overview

Spam emails are a major threat to productivity and security. This project focuses on building a machine learning model to classify emails as Spam or Ham (Not Spam) using Natural Language Processing (NLP) techniques. The system helps automate email filtering and reduce unwanted messages. In addition to model development, an interactive web application was built using Streamlit to allow users to input text and receive real-time spam classification results.

Goal

To build a supervised learning system that detects spam with high accuracy and minimal false positives.

Statement of the Problem

Email spam wastes time, consumes resources, and exposes users to security risks.

Manual filtering is inefficient and error-prone.

Rule-based systems fail against evolving spam patterns

Need for an intelligent, learning-based solution

Objectives

- Design a reliable real world machine learning model for spam detection
- Preprocess and extract meaningful features from email text
- Convert text into numerical features (TF-IDF).
- Train and evaluate multiple ML models.
- Compare performance using metrics.
- Achieve high accuracy with minimal false positives

Dataset

Dataset: SMS Spam Collection Dataset

Total messages: 5,572

Classes:

Ham (Not Spam)

Spam

Source: UCI/Kaggle

Project Scope

- Binary classification (Spam vs Ham).

- Text-based email content analysis
- Model training & evaluation.
- Visualization of results.
- Offline dataset training and testing

Excluded:

- Real-time email server integration
- Image/attachment spam detection

Methodology

We followed a structured workflow: exploring the data, cleaning it, engineering features, training models, and evaluating results.

- **Exploratory Data Analysis (EDA)**
  - EDA was conducted to understand the structure and patterns in the dataset:
  - Checked for missing and duplicate values
  - Analyzed class distribution (Spam vs Ham)
  - Created message length features
  - Visualized distributions using histograms
  - Generated word clouds for Spam and Ham messages

- **Data Preprocessing & Feature Engineering**
  - Text preprocessing steps included:
  - Lowercasing text
  - Removing punctuation, numbers, and special characters
  - Removing stopwords
  - Lemmatization
  - Feature extraction using TF-IDF Vectorization
- **Model Building**

Models Used

The following machine learning models were trained and evaluated:

- Logistic Regression
- Multinomial Naive Bayes

Model Training

Logistic Regression:

- Used class_weight='balanced' to handle imbalance.
- Trained on TF-IDF features.

Naive Bayes:

- Works well with word frequency features.
- Trained on same dataset for fair comparison.

Evaluation Metrics

Model performance was evaluated using:

- Accuracy
- Precision
- Recall
- F1-score
- Confusion Matrix

Results

Metric	Logistic Regression	Naive Bayes
Accuracy	0.9797	0.9797
Precision	0.9104	1.0000
Recall	**0.9313**	0.8397
F1 Score	**0.9208**	0.9129

- Naive Bayes achieved higher precision for spam detection (fewer false spam alerts).
- Logistic Regression showed higher recall (better at catching spam). May wrongly flag some normal messages.
- Both models performed well, with Naive Bayes slightly outperforming Logistic Regression in detecting spam messages

Recommendation

- Use Logistic Regression if the goal is to catch as much spam as possible.
- Use Naive Bayes if the goal is to reduce false alarms.
- For this project Logistic Regression is preferred (better spam detection).

Web Application (Streamlit)

- Built and deployed web application so users can input text and receive real-time predictions.

Tools & Libraries

- Python
- Pandas, NumPy
- NLTK
- Scikit-learn
- Streamlit

Discussion

From the evaluation metrics, although both models achieved the same accuracy, Logistic Regression had a higher recall, meaning it was better at detecting spam messages. Naive Bayes had a higher precision, which means that when it labels messages as spam, it is more likely to be correct.

These results show that each model performs better in different areas. Logistic Regression is more suitable when the goal is to detect as many spam messages as possible, even if a few normal messages are wrongly flagged. Naive Bayes is more suitable when reducing false spam alerts is more important. We'd recommend Logistic Regression, because it detects spam messages more effectively due to its higher recall. Although Naive Bayes has higher precision, Logistic Regression is less likely to classify spam messages as ham.

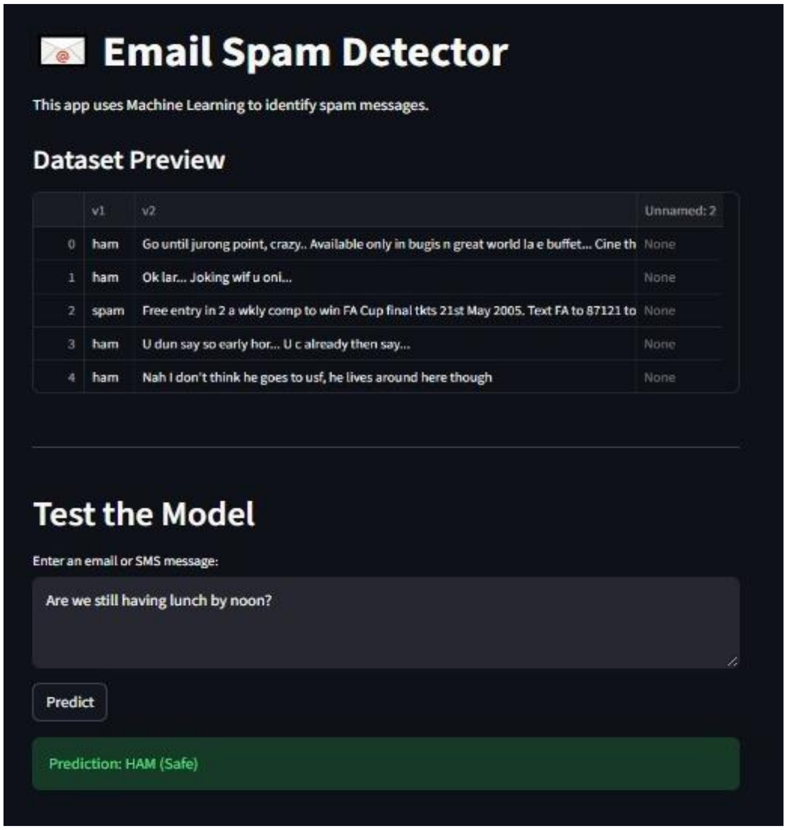
Conclusion

This project successfully demonstrates the application of Natural Language Processing and Machine Learning techniques to email spam detection. Through effective data preprocessing, feature engineering using TF-IDF, and model evaluation, the system achieved high classification accuracy. The results show that machine learning models such as Naive Bayes and Logistic Regression can effectively identify spam emails. This project provides a solid foundation for further improvements, such as experimenting with advanced deep learning techniques.

Future Work

Several extensions can be explored to enhance the performance and applicability of this system:

- Experimentation with advanced feature representations such as word embeddings (Word2Vec, GloVe, or FastText)
- Implementation of deep learning models including Recurrent Neural Networks (RNNs) and Transformer-based architectures
- Incorporation of additional email metadata such as sender information and timestamps
- Evaluation on larger and more diverse email datasets to improve generalization.



Email Spam Detector

This app uses Machine Learning to identify spam messages.

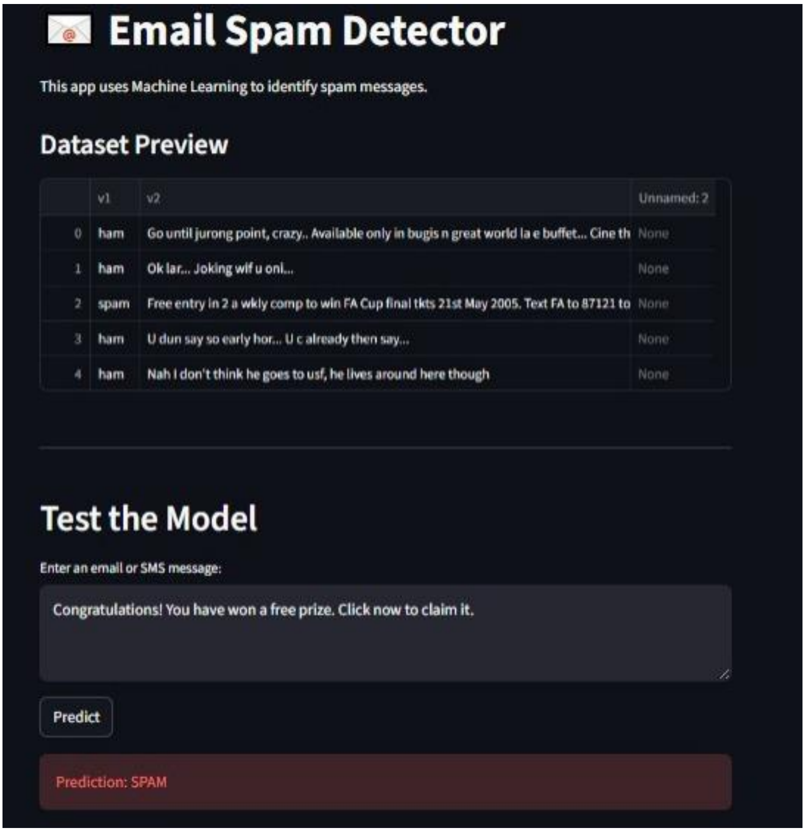
Dataset Preview

	v1	v2	Unnamed: 2
0	ham	Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine th	None
1	ham	Ok lar... Joking wif u oni...	None
2	spam	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to	None
3	ham	U dun say so early hor... U c already then say...	None
4	ham	Nah I don't think he goes to usf, he lives around here though	None

Test the Model

Enter an email or SMS message: Are we still having lunch by noon?

Prediction: HAM (Safe)



Email Spam Detector

This app uses Machine Learning to identify spam messages.

Dataset Preview

	v1	v2	Unnamed: 2
0	ham	Go until jurong point, crazy.. Available only in bugis n great world la e buffet... Cine th	None
1	ham	Ok lar... Joking wif u oni...	None
2	spam	Free entry in 2 a wkly comp to win FA Cup final tkts 21st May 2005. Text FA to 87121 to	None
3	ham	U dun say so early hor... U c already then say...	None
4	ham	Nah I don't think he goes to usf, he lives around here though	None

Test the Model

Enter an email or SMS message: Congratulations! You have won a free prize. Click now to claim it.

Prediction: SPAM

References

1. T. Mitchell, Machine Learning, McGraw-Hill

2. Scikit-learn Documentation

3. UCI Machine Learning Repository

Contributors

Azubuike Uwuma Black – Team Lead

ABE Olaoluwa Ojo

Akande Lawal Bolaji

Omoleye Priscilla Moyinoluwa

Adole Elijah Edache

Amedu Lauretta Unekwu

Bolarinwa Abdulsamad Bolaji

Owolabi Idowu Ebenezer

Ibrahim Jonah Yunana

Odetunde Oluwakemi Esther

Damola Balogun

Ajayi Esther Mosunmola

Yasir Sallau

Lawal Esther Oluwayemisi

Abdulhakeem Abdullah Mujahid

Kuyebi Zion Abiodun

Adejuwon Pelumi

Josiah Bulus