Spams vs Hams Emails Classifications

By Burhan Sultan Basit

Student ID 650387980

# 1. Introduction

The goal of this project was to build a simple machine learning model that can detect whether an email is spam or not. This type of project is common because spam messages are a big problem in email systems. The work was done step by step, from cleaning the data, training the models, testing them, and finally creating a small app to make predictions.

# 2. Data Preparation

In this step, I worked with two different CSV files that contained email data. I checked for missing or repeated lines, and then joined them into one complete dataset. After combining them, I made sure the columns were clear , one called *email* and the other *label* (where 0 means ham and 1 means spam). I saved this final clean version as CLEAN\_EMAILS.csv, which was later used for training the models.A screen shot of a screen

AI-generated content may be incorrect.

# 3. Modeling and MLflow

# After preparing the clean data, I changed all the email text into numbers using a TF-IDF vectorizer so that the models could understand it. Then I trained three different models: Naive Bayes, Logistic Regression, and Linear SVM. Each model was tested and compared based on accuracy and F1 score. To keep everything organized, I used MLflow to track the results. It allowed me to see which model performed best and how much time each one took to train. Overall, the training went smoothly, and MLflow helped me visualize and compare all three runs in one place.

A screenshot of a computer

AI-generated content may be incorrect.

LogReg:

A screenshot of a computer

AI-generated content may be incorrect.

MultinimialNB:

A screenshot of a computer

AI-generated content may be incorrect.

Linearsvc:

A screenshot of a computer

AI-generated content may be incorrect.

# 4. Results

# After training all three models, I compared their results. All of them gave good accuracy, but the Linear SVM and Logistic Regression models performed a little better than Naive Bayes. I checked the results using different metrics like Accuracy, Precision, Recall, and F1 Score. The confusion matrix also helped me see how many spam and ham emails were predicted correctly. Overall, the models were reliable and showed that the data cleaning and preprocessing were done properly.

A screen shot of a computer

AI-generated content may be incorrect.

# 5. Streamlit App

To make my project easy to use, I built a small Streamlit web app.  
In the app, users can type or paste an email and see right away if it’s spam or ham.  
It also shows a confidence percentage, so you can know how sure the model is about its prediction. The app also lets me choose between the three models (Naive Bayes, Logistic Regression, or Linear SVM) and compare how each one behaves.

A screenshot of a message

AI-generated content may be incorrect.A screenshot of a chat

AI-generated content may be incorrect.

# 6. Conclusion

This project helped me understand how machine learning can be used to detect spam emails.I did every step cleaning the data, building and testing models, comparing the results, and building an app. Using MLflow made it easier to track the experiments, and Streamlit made it possible to test the models interactively.  
Overall, this project shows how data, code, and interface come together to solve a real problem.

A screenshot of a computer

AI-generated content may be incorrect.End of Report