Project Report – Email Spam Classifier

## **1. Project Overview**

The **Email Spam Classifier** is a machine learning model that automatically classifies incoming emails into two categories:

* **Spam** (Unwanted or promotional messages)
* **Not Spam** (Legitimate messages)

The project uses **Naive Bayes** classification and Support Vector Classification algorithms with **TF-IDF** vectorization for feature extraction and is integrated into a **Streamlit web application** for ease of use.

## **2. Objectives**

* Build a **robust spam detection system** with high accuracy.
* Implement **text preprocessing** techniques for email content.
* Deploy the model in an **interactive and user-friendly interface**.

## **3. Dataset**

* **Source:** Public spam email dataset from Kaggle.
* **Format:** CSV with Message and Category columns.
* **Label Encoding:** spam → 1, ham → 0

## **4. Methodology**

### **Step 1: Data Preprocessing**

* Removal of punctuation, stopwords, and special characters.
* Conversion of text to lowercase.
* Tokenization and lemmatization.

### **Step 2: Feature Extraction**

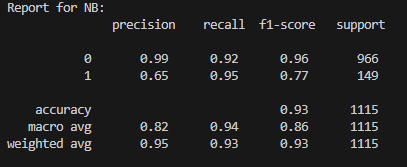
* Applied **TF-IDF Vectorizer** to convert text into numerical vectors.

### **Step 3: Model Training**

* Algorithm: **Multinomial Naive Bayes and Support Vector Classifier**
* Training/Testing split: **80% train / 20% test**
* Performed Random Under Sampling due to imbalanced dataset.

### **Step 4: Model Evaluation**

* **Accuracy Score for Naïve Bayes:** 92.55%
* **Accuracy Score for SVC:** 97.04%
* **Classification Report for Naïve Bayes:**



* **Classification Report for Naïve Bayes:**

A screenshot of a computer screen

AI-generated content may be incorrect.

## **5. Deployment**

* **Platform:** Streamlit (local web application)
* **Usage:** User inputs email text → Model predicts **Spam** or **Not Spam** instantly.

## **6. Key Features**

* Fast and lightweight model.
* High accuracy for spam detection.
* Easy-to-use web interface.
* Extendable for real-world email filtering.

## **7. Future Improvements**

* Integration with **email APIs** for real-time filtering.
* Add **deep learning models** (e.g., LSTM) for better accuracy.
* Multi-language spam detection.

## **8. Conclusion**

The Email Spam Classifier successfully achieves its goal of distinguishing spam from non-spam emails. With further optimization, it can be deployed in production environments for real-world email security systems.