NB-Mail Classifier

Semester project

Session 2023-2027

BS(AI) in Software Engineering



Department of Software Engineering

Faculty of Computer Science & Information Technology

The Superior University, Lahore

FALL 2024

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Type (Nature of project) | | | [ ✓ ] **D**evelopment [ ] **R**esearch [ ] **R**&**D** | | |
| Area of specialization | | | Artificial Intelligence | | |
| **Project Group Members** | | | | | |
| Sr.# | Reg. # | Student Name | | Email ID | \*Signature |
| (i) | Su92-bsaim-f23-030 | Hammad Arshad | | HammadArshad026@gmail.com | Hammad |
| (ii) |  |  | |  |  |
| (iii) |  |  | |  |  |

**Table of Contents**

[List of Figures iv](#_Toc184014295)

[Chapter 1 2](#_Toc184014296)

[Introduction 2](#_Toc184014297)

[Overview of the Project 3](#_Toc184014298)

[Project Components 3](#_Toc184014299)

[1. Dataset 3](#_Toc184014300)

[2. Machine Learning Model 3](#_Toc184014301)

[3. Steps in the Project 4](#_Toc184014302)

[Importance of Email Classification 4](#_Toc184014303)

[Scope of the Project 4](#_Toc184014304)

[Chapter 2 6](#_Toc184014305)

[Tool & Technology 6](#_Toc184014306)

[1. Operating System 7](#_Toc184014307)

[2. Programming Environment 7](#_Toc184014308)

[3. Dataset 8](#_Toc184014309)

[4. Libraries and Frameworks 8](#_Toc184014310)

[5. Online Resources 8](#_Toc184014311)

[Why These Tools Were Chosen 9](#_Toc184014312)

[Chapter 3 10](#_Toc184014313)

[Implementation Code 10](#_Toc184014314)

[Chapter 4 13](#_Toc184014315)

[Conclusion 13](#_Toc184014316)

# 

# List of Figures

[Figure 1: import libraries 11](#_Toc184012338)

[Figure 2:Import Data 11](#_Toc184012339)

[Figure 3:Data Exploration 11](#_Toc184012340)

[Figure 4:Data splitting 12](#_Toc184012341)

[Figure 5:Trained the model 12](#_Toc184012342)

# 

# Chapter 1

# Introduction

Machine learning is revolutionizing the way we interact with data. One of its prominent applications is in text classification, specifically in identifying spam and legitimate emails. This project focuses on **classifying emails into two categories: spam and ham (not spam)** using the Naive Bayes algorithm, a powerful and efficient probabilistic classifier.

## ****Overview of the Project****

The primary goal of this project is to build a machine learning model capable of analyzing email text and categorizing it based on its content. The process involves training a classifier with labeled email data and then using it to predict the category of unseen messages.

## ****Project Components****

## 1. ****Dataset****

* **Source and Structure**:
  + The dataset is a CSV file containing two columns:
    - **Category**: Specifies if the email is "spam" or "ham".
    - **Message**: The text content of the email.
  + This labeled dataset is the foundation for training the model.
* **Significance**:
  + The diversity and accuracy of the dataset ensure the model learns patterns effectively, enabling precise predictions.

## 2. ****Machine Learning Model****

* **Algorithm Used**:
  + This project employs the **Naive Bayes algorithm**, a popular choice for text classification tasks. Its simplicity and high efficiency with small datasets make it ideal for this project.
* **Why Naive Bayes?**
  + It works well with text data by utilizing word frequencies.
  + It is computationally lightweight, making it suitable for real-time applications.

## 3. ****Steps in the Project****

* **Data Preprocessing**:
  + Cleaning and tokenizing the email content to make it suitable for training the model.
* **Feature Extraction**:
  + Transforming the raw email text into numerical features using techniques like bag-of-words or term frequency-inverse document frequency (TF-IDF).
* **Model Training**:
  + Feeding the preprocessed data into the Naive Bayes classifier.
* **Evaluation**:
  + Assessing the performance of the model using metrics such as accuracy, precision, recall, and F1-score.
* **Prediction**:
  + Using the trained model to classify new, unseen emails into spam or ham.

## ****Importance of Email Classification****

In the digital age, email remains a critical communication tool. With the rise of unwanted and malicious messages, effective spam detection is vital for:

* **Enhancing productivity** by reducing clutter in inboxes.
* **Protecting users** from phishing and fraudulent activities.
* **Improving user experience** by prioritizing relevant emails.

## ****Scope of the Project****

This project provides a simple yet effective solution for filtering spam emails. By leveraging machine learning techniques, it can:

* Be integrated into larger email management systems.
* Adapt to new datasets for enhanced performance.
* Serve as a foundation for exploring more advanced natural language processing (NLP) techniques.

# Chapter 2

# Tool & Technology

The successful implementation of this email classification project relies on a combination of software tools, technologies, and resources. Each component played a critical role in streamlining development, testing, and deployment. Below is an overview of the tools and technologies used:

## ****1. Operating System****

**Windows 10**:

* + The development environment was set up on Windows 10, a user-friendly and versatile operating system.
  + It offers excellent compatibility with programming tools and libraries, ensuring a smooth workflow.
  + Features like file management and robust system support made it ideal for handling project files and datasets.

## ****2. Programming Environment****

* **Visual Studio Code (VSCode)**:
  + VSCode was the primary code editor used in this project.
  + Its lightweight nature and extensive features, such as syntax highlighting, debugging tools, and integrated terminal, enhanced productivity.
  + Extensions like Python and Jupyter Notebook support were utilized for seamless integration.
* **Jupyter Notebook (.ipynb files)**:
  + Jupyter Notebook provided an interactive development environment for coding and testing.
  + Its cell-based structure allowed incremental code execution, making debugging and visualization easier.
  + Graphs and outputs were displayed inline, simplifying data analysis and model evaluation.

## ****3. Dataset****

* **Source**:
  + The dataset used for this project was sourced from **Kaggle**, a well-known platform for data science and machine learning resources.
  + Kaggle provided a clean and labeled dataset in CSV format, ensuring ease of use.
* **Structure**:
  + The dataset consisted of two columns:
    - **Category**: Labeled as "spam" or "ham" (not spam).
    - **Message**: Contains the text content of the email.
  + The dataset served as the backbone for training and evaluating the machine learning model.

## ****4. Libraries and Frameworks****

* The project leveraged powerful Python libraries for various tasks:
  + **pandas**: For reading and preprocessing the dataset.
  + **scikit-learn**: To implement the Naive Bayes algorithm and perform model evaluation.
  + **NumPy**: For numerical computations.
  + **Matplotlib/Seaborn**: To visualize data trends and results.

## ****5. Online Resources****

* **Online Research**:
  + Research conducted on platforms like **Google Scholar**, **Medium**, and **Stack Overflow** helped in understanding spam classification techniques and resolving coding challenges.
  + Tutorials and guides from YouTube and Kaggle were referenced for dataset exploration and algorithm optimization.
* **Documentation**:
  + Official documentation of Python libraries (e.g., pandas, scikit-learn) was utilized to ensure best practices in implementation.

## ****Why These Tools Were Chosen****

* **Accessibility**:
  + The tools used are widely available and free, making the project accessible to anyone interested in replicating or extending it.
* **Ease of Use**:
  + VSCode and Jupyter Notebook offer intuitive interfaces, ideal for beginners and experienced developers alike.
* **Community Support**:
  + The Python ecosystem, Kaggle, and Windows 10 have large user communities, providing abundant resources for troubleshooting and enhancement.

# Chapter 3

# Implementation Code

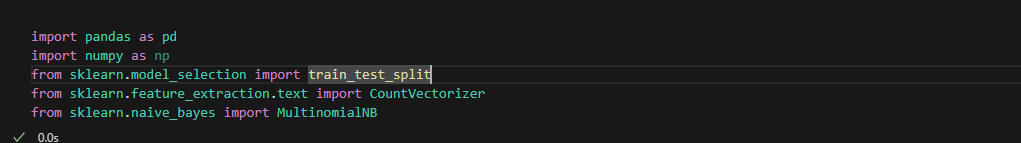
****

Figure 1: import libraries

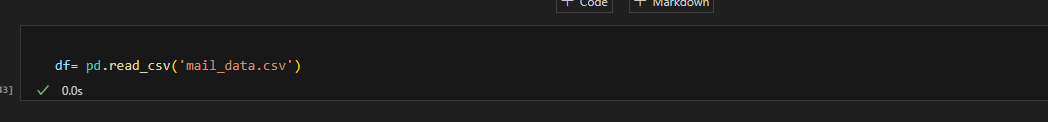


Figure 2:Import Data

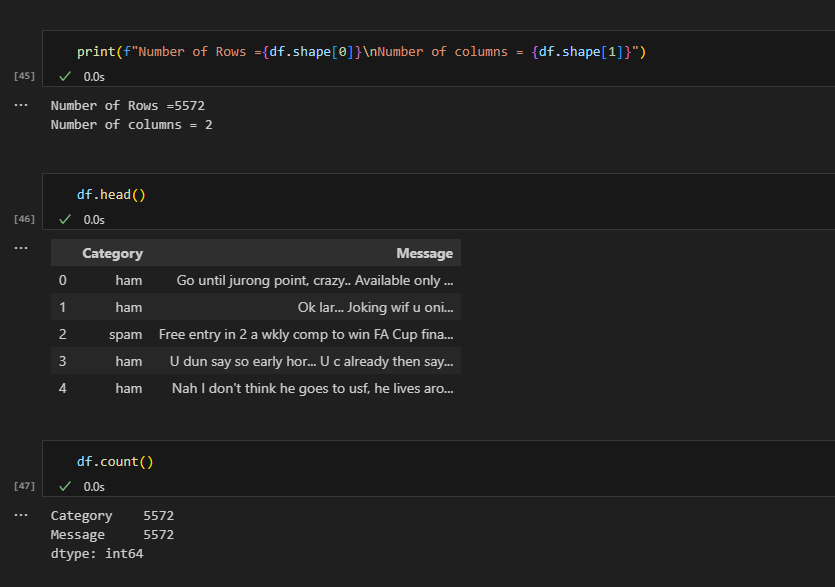


Figure 3:Data Exploration

****

Figure 4:Data splitting

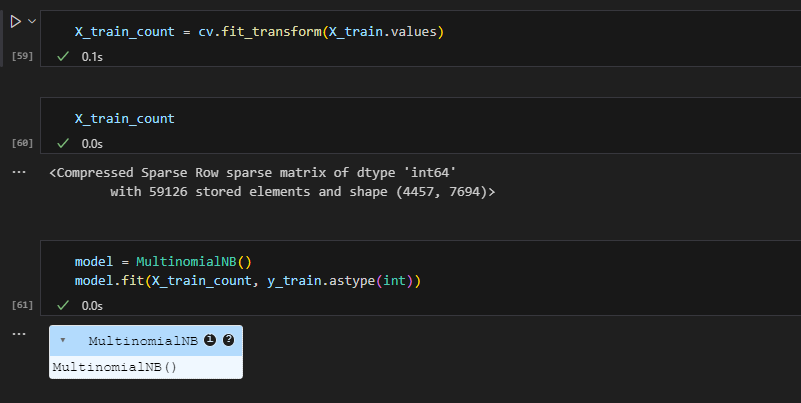


Figure 5:Trained the model

# 

# Chapter 4

# Conclusion

In conclusion, this project successfully implemented a Naive Bayes classifier to categorize emails as spam or ham. Using a dataset from Kaggle, the model achieved accurate results and demonstrated the effectiveness of machine learning in email filtering. While the project faced challenges in data preprocessing and feature extraction, future improvements could enhance performance by exploring advanced algorithms and preprocessing techniques. The system has practical applications in real-time email classification, providing a foundation for smarter email management. Overall, the project highlights the potential of machine learning in automating everyday tasks.