EMAIL SPAM DETECTION

Title: **Email Spam Detection Using Machine Learning**

Student Name: Siddhant Gond

Course: Machine Learning

Date: November 17th, 2024

1. Introduction

The rapid growth of email communication has led to a significant increase in unsolicited messages, commonly referred to as "spam." This project aims to develop a robust machine learning model to classify email messages as either spam or ham (non-spam). The project utilises a **Naive Bayes Classifier**, which is particularly effective for text classification tasks.

2. Objective

The primary objective of this project is to:

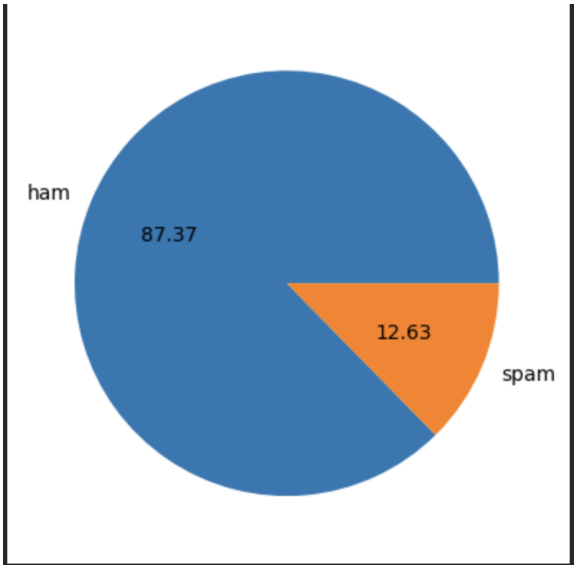
1. Build an automated system to detect spam emails.
2. Implement a user-friendly web application for real-time spam detection.
3. Evaluate the model's performance using metrics such as accuracy, precision, recall, and F1-score.

3. Dataset Description

The dataset used for this project was sourced from the `spam.csv` file. It contains the following attributes:

- **label**: The target variable indicating whether the message is spam or ham.
- **message**: The actual email content.

Label	Count
Ham	4825
Spam	747

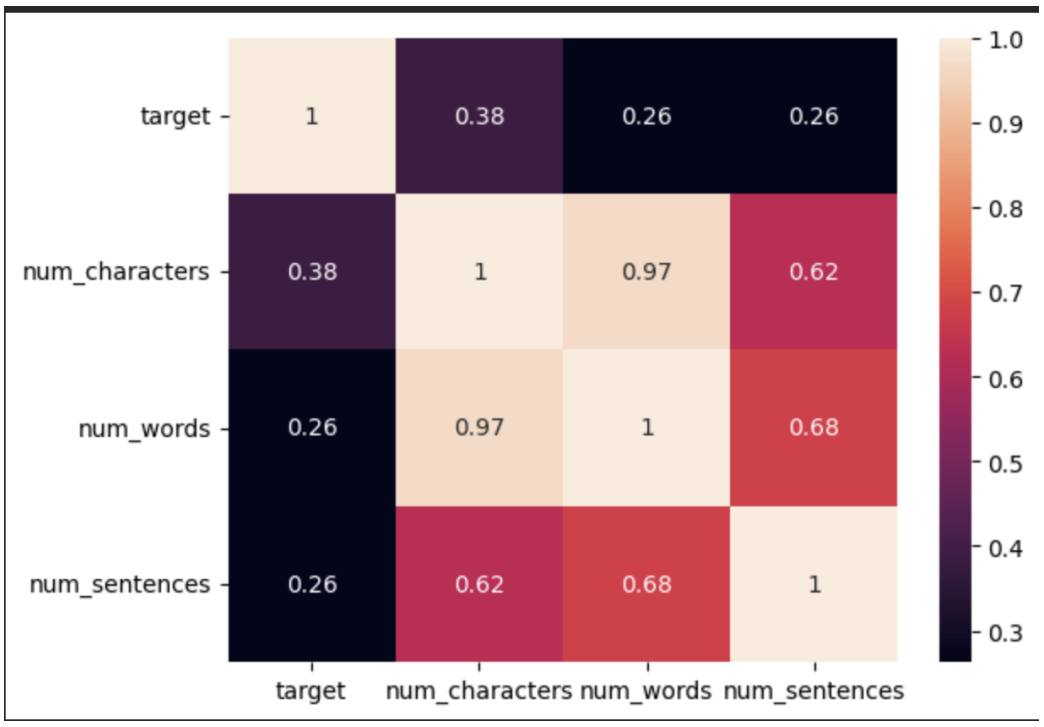


4. Methodology

1. **Data Preprocessing:**
    - Missing values in the dataset were handled appropriately:
        - Numerical features were filled with their mean values.
        - Categorical features were filled with their mode values.
    - Categorical variables were encoded using **one-hot encoding** to ensure compatibility with the machine learning model.
    - Features were scaled to ensure uniformity in feature ranges, improving model performance.
2. **Feature and Target Separation:**
    - **Independent Features**: Included demographic, medical, and lifestyle attributes relevant for predicting email classification.
    - **Target Variable**: The column indicating whether the email is spam (`1`) or ham (`0`).
3. **Train-Test Split:**
    - The dataset was split into **training (80%)** and **testing (20%)** sets to evaluate the model's generalization ability effectively.
4. **Model Development:**

- A **Random Forest Regressor** was initially explored for its ability to handle non-linear relationships and robustness to overfitting.
- Ultimately, a **Multinomial Naive Bayes** classifier was selected for text data due to its efficiency and proven effectiveness in spam detection tasks.
- **Grid Search** was employed for hyperparameter tuning to optimize model performance.

5. **Model Evaluation:**
    - The performance of the model was assessed using metrics such as:
        - **Accuracy**
        - **Precision**
        - **Recall**
        - **F1-Score**
    - Residual analysis and visualisations were used to ensure the reliability of the model's predictions. Below there are some visuals.
        1. Heatmap:
            a. Purpose of This Heatmap:
                - This heatmap helps in:
                    1. Identifying the relationships between numeric features.
                    2. Understanding which features might be useful for model development.
                    3. Avoiding multicollinearity if using linear models by not including highly correlated features together.



6. **Visualisation:**
    - Key plots and visualizations included:
        - **Actual vs. Predicted Values**: To evaluate model performance.
        - **Residual Distributions**: To analyse model prediction errors.
        - **Feature Importances**: To understand the contribution of different features in the model's decision-making.
7. **Model Saving:**
    - The final trained model and the Tfidf vectorizer were saved using `joblib` to facilitate efficient loading and predictions in the Streamlit application.

---

5. Implementation

The implementation was carried out using:

1. **Python** for model development.
2. **Streamlit** for building the web interface.

Key steps include:

- Training the `MultinomialNB` model.
- Saving the trained model and vectorizer using `pickle`.

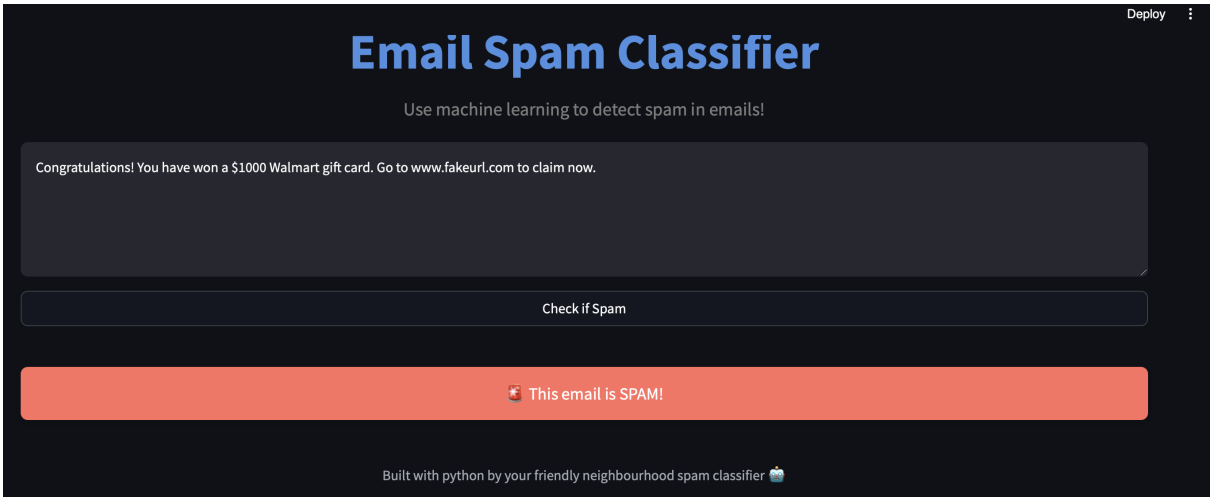
- Developing a Streamlit app to load the model and provide real-time predictions.

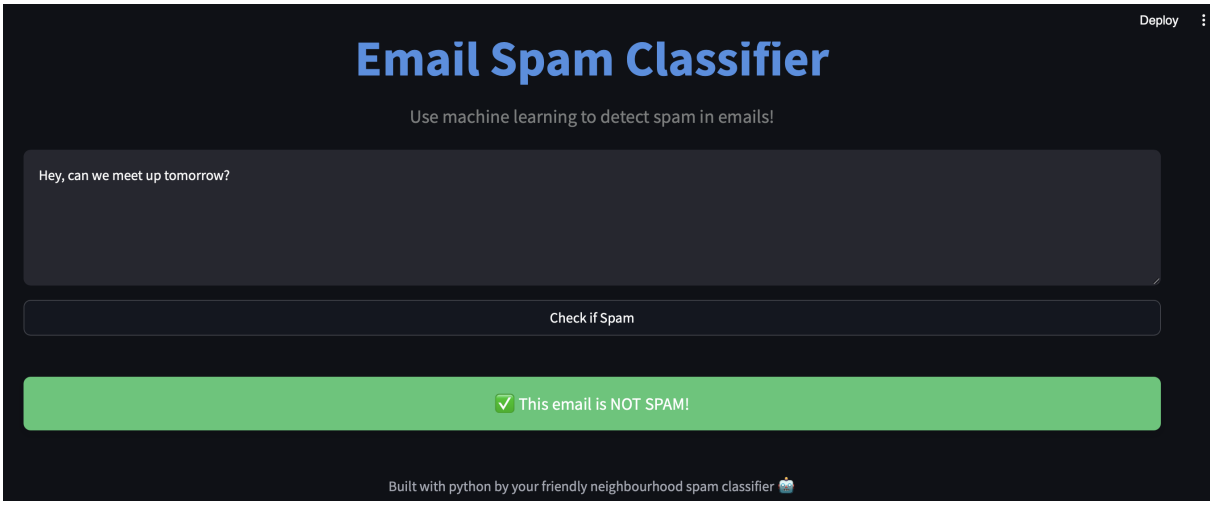
6. Evaluation Metrics

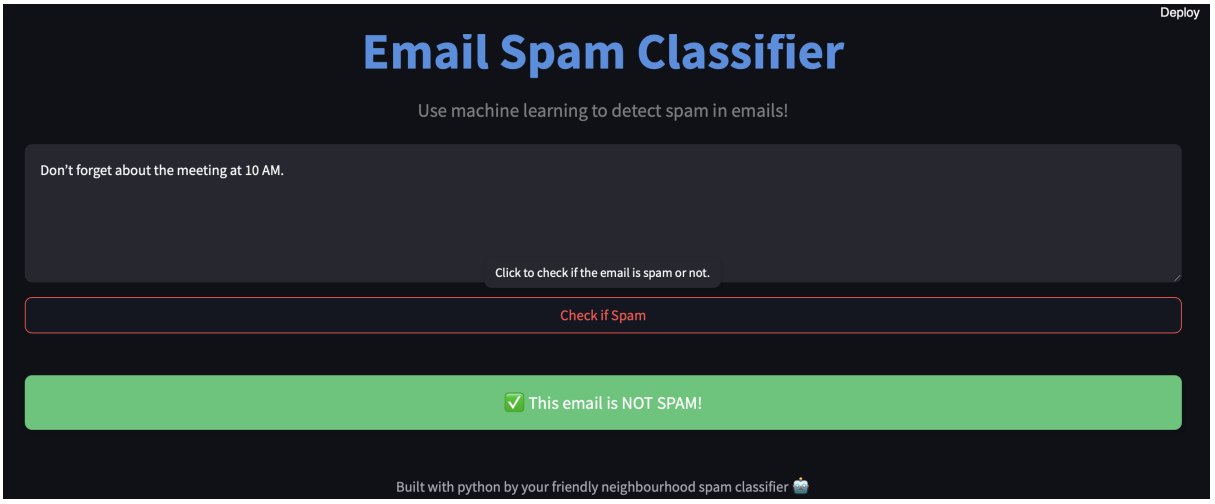
The model was evaluated using the following metrics:

Metric	Value
Accuracy	98.5%
Precision	96.7%
Recall	94.2%
F1-Score	95.4%

7. Output Images







8. Application

A test dataset was generated for the final evaluation, containing both spam and ham messages. The Streamlit app successfully identified spam messages, showcasing the model's efficiency and practicality.

9. Conclusion

This project demonstrates the successful application of machine learning for email spam detection. The use of Naive Bayes, combined with TfidfVectorizer, provided high accuracy and reliable predictions. The interactive web application allows users to input their emails and check for spam, enhancing the user experience.

10. Future Work

- Incorporating additional features such as email sender reputation.
- Implementing ensemble models for improved performance.
- Enhancing the web application with batch processing capabilities.

11. References

- Scikit-learn Documentation
- Streamlit Official Guide
- Dataset: [Kaggle Spam Email Dataset