Project Overview Report

Introduction

The project, a Multi-Platform Email Spam Identifying System, aims to offer users a versatile solution for detecting spam emails across different platforms. The system includes a web-based app, a browser extension, and a Windows application.

Project Scope

The scope encompasses a web-based app where users input an email to determine its spam status.

Additionally, a browser extension offers the same functionality for seamless integration into web browsers.

The Windows app connects to users' Gmail accounts using the `simplegmail` module in Python, reads emails for a specified duration, displays their spam or ham status, and generates a bar graph illustrating the count of spam and ham emails.

Technologies Used

Web-Based App:

- Frontend: HTML, Bootstrap (v5.0.2), CSS
- Backend: Python (Flask)
- Machine Learning: Pickle (for model persistence)

Browser Extension:

- Manifest Version 3: Utilizing the latest Chrome extension manifest version.
- Frontend: HTML, CSS, JavaScript
- Backend: Flask (Python)
- Machine Learning: Pickle (for model persistence)

Windows App:

- Frontend: PyQt6.

- Backend: Python.
- Gmail Integration: `simplegmail` module using client_secret JSON file.

Architecture Overview

The architecture involves a web-based app with client-server communication for email analysis. The browser extension integrates seamlessly with web browsers.

The Windows app connects to Gmail using the `simplegmail` module, reads and analyzes emails, and visualizes the results.

Features and Functionality

Web-Based App:

- The Web application is designed to predict whether a given email is spam or not. The application provides users with a simple interface to input an email message and receive an instant classification result.
- It is clean and intuitive user interface built using Bootstrap, making it easy for users to interact with the system.
- It is powered by a machine learning model (loaded using the Pickle library), the application accurately classifies emails as either spam or ham (not spam).
- The application is designed to be responsive, ensuring a seamless user experience across various devices.

Browser Extension:

- It is a tool designed to quickly classify emails as spam or not. With a user-friendly
interface, users can easily paste email content and receive instant predictions on whether the communication is spam or not.
- The extension provides a dedicated popup that allows users to conveniently paste
email content and receive instant predictions without leaving their browsing session.

- Utilizing a machine learning model, the extension performs real-time classification.

Windows App:

- Connects to Gmail using the `simplegmail` module and client_secret JSON file.
- Reads emails for a specified duration.
- Displays spam or ham status.
- Generates a bar graph showing the count of spam and ham emails

---

Codebase Structure

Web Application:

Libraries and Technologies Used:

- **Flask:**
  - A micro web framework for Python utilized for building the web application and handling HTTP requests.
- **render_template:**
  - A Flask function used for rendering HTML templates.
- **request:**
  - A Flask module employed for handling incoming HTTP requests.
- **pickle:**
  - A Python module used for serializing and deserializing objects, providing functionality for model persistence.

Web Extension:

Libraries and Technologies Used:

- **Flask:**
  - Utilized for the backend of the Chrome extension, handling HTTP requests.

- **Flask-CORS:**
    - An extension facilitating Cross-Origin Resource Sharing (CORS) in Flask applications.
- **jsonify:**
    - A Flask function employed for creating a JSON response.
- **pickle:**
    - Used for model persistence.

Windows Application:

Libraries and Technologies Used:

- **sys:**
    - A Python module providing access to variables used or maintained by the interpreter.
- **PyQt6:**
    - A set of Python bindings for Qt libraries, used for building the graphical user interface (GUI).
- **QApplication, QMainWindow, QWidget, QVBoxLayout, QTextEdit, QPushButton, QComboBox, QHBoxLayout, QLabel:**
    - PyQt6 classes utilized for building various GUI elements.
- **simplegmail:**
    - A Python library for interacting with Gmail services.
- **construct_query:**
    - A function from the simplegmail library for constructing Gmail query strings.
- **dateutil.parser:**
    - A module used for parsing date strings.
- **numpy:**
    - A library for numerical operations in Python.

- **matplotlib.pyplot:**
  - A library employed for creating visualizations in Python.

Future Improvements:

1. **User Preferences:**
   - Allow users to save preferences, including default settings for email classification.
   - Provide a personalized experience by enabling users to customize their spam detection preferences.
2. **Push Notifications and Live Updates:**
   - Integrate push notifications or live updates for the Chrome extension to promptly notify users of email classification results.
   - Enhance user engagement by ensuring real-time updates on the status of incoming emails.
3. **Multilingual Support:**
   - Enable users to interact with the application and extension in their preferred language.
   - Implement multilingual support to enhance accessibility and user experience for a diverse user base.
4. **Security Measures:**
   - Enhance security measures to protect user data and ensure a secure connection between the client and server.
   - Implement encryption protocols and secure authentication methods to safeguard sensitive user information.
5. **Automatic Blocking of Spam Emails:**
   - Implement a feature that allows automatic blocking of spam emails upon user approval.

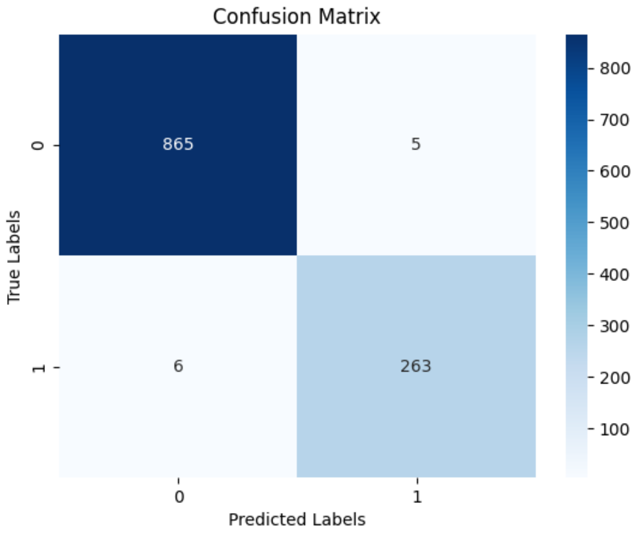
- Provide users with the option to streamline the spam-blocking process and enhance the efficiency of the system.

Classification Report and Confusion Matrix

Classification Report

	Precision	**Recall**	**F1-Score**	**Support**
0	0.99	0.99	0.99	870
1	0.98	0.98	0.98	269
Accuracy			0.99	1139
Macro Average	0.99	0.99	0.99	1139
Weighted Average	0.99	0.99	0.99	1139

Confusion Matrix



Conclusion:

In summary, the Multi-Platform Email Spam Identifying System enhances user experience through effective spam detection across web, browser, and Windows platforms. With features like user preferences, real-time notifications, and multilingual support, the system prioritizes customization and accessibility. Future improvements aim to bolster security and provide users with insightful performance metrics.