Project Description:

This project focuses on building an intelligent email classification system capable of sorting organizational emails into meaningful categories such as Important, Promotional, and Spam. The solution integrates modern Natural Language Processing (NLP) techniques with a production-ready web interface, creating a robust tool that enhances workplace productivity by reducing inbox clutter and highlighting priority communication.

The system uses a fine-tuned DeBERTa transformer model for contextual understanding of email content and a Flask-based backend for real-time inference. The classifier evaluates both subject and body text, applies preprocessing, and predicts the most relevant label along with confidence scores. It supports individual email classification and batch processing through CSV uploads, making it applicable across organizational teams and administrative workflows.

Project Scenarios

Scenario 1: Workplace Email Prioritization

Employees often receive hundreds of emails daily, making it challenging to identify urgent messages. The system analyzes incoming emails and categorizes them instantly. Important updates, meeting invites, alerts, and requests are surfaced, while promotional and spam messages are de-prioritized, reducing manual effort.

Scenario 2: Enterprise-Level Email Management

Large organizations use this system to automate the sorting of internal and external communication. HR newsletters, event announcements, marketing emails, and alert notifications can be classified automatically and routed to respective departments, ensuring no critical communication is missed.

Scenario 3: Customer Support Ticket Pre-Processing

Support centers deal with high volumes of incoming emails. By passing messages through this classifier, the system distinguishes urgent service issues from general inquiries or promotional content. Agents receive categorized messages aligned with their priority, improving response times.

Prerequisites

To build and run this system, the following tools and concepts are required:

Software

- Python 3.x
- Anaconda or any Python environment manager
- Flask web framework
- Transformer libraries (HuggingFace Transformers)
- PyTorch
- Pandas, NumPy, scikit-learn
- NLTK for text preprocessing

Required Python Packages

(from requirements.txt)

requirements

- torch
- transformers
- flask
- pandas
- numpy
- scikit-learn
- nltk
- matplotlib
- seaborn
- joblib
- werkzeug

Prior Knowledge

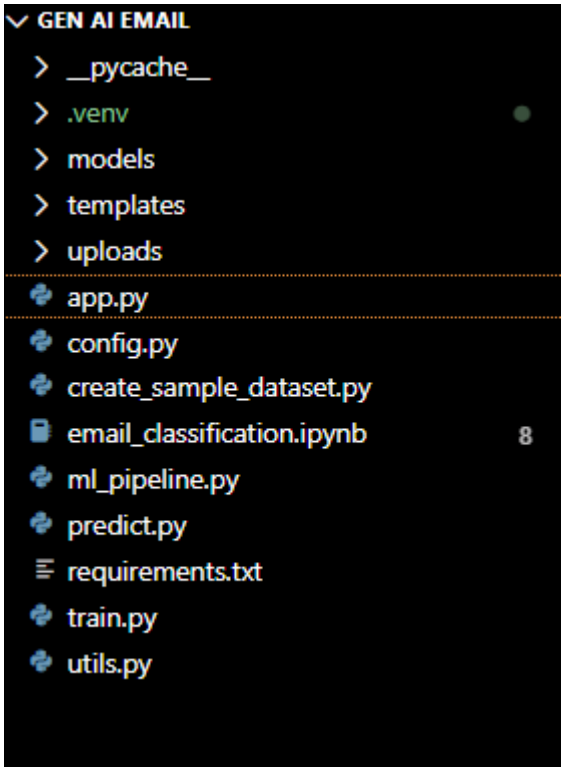
Students should understand:

- Basics of Natural Language Processing
- Tokenization and embeddings
- Deep learning concepts
- Transformer architecture (BERT/DeBERTa)

- Flask routing and template rendering
- JSON formatting
- CSV handling

Project Structure:

Create the Project folder which contains files as shown below



Milestone 1: Data Collection & Preparation

Activity 1.1: Collecting the Dataset

The system uses a curated dataset of organizational emails containing text and label fields. The training pipeline loads the dataset from:

https://www.kaggle.com/datasets/wcukierski/enron-email-dataset

Activity 1.2: Importing Required Libraries

For preprocessing, tokenization, and model training, the pipeline imports components such as:

- AutoTokenizer
- DebertaV2 or DeBERTa-v3 transformer
- PyTorch DataLoader
- scikit-learn utilities

```
from flask import Flask, render_template, request, jsonify
from transformers import DebertaV2Tokenizer, DebertaV2ForSequenceClassification
import torch
import os
from config import Config
```

Activity 1.3: Reading the Dataset

load_dataset() is used to read the CSV file, validate columns, and extract email content and categories. The function also removes empty rows and prepares label mappings.

Activity 1.4: Data Preparation

The pipeline cleans emails by:

- Removing metadata headers
- Removing signatures, URLs, emails, long numerics
- Lowercasing content
- Removing special characters

This is handled by preprocess_text() using regex patterns.

Activity 1.5: Handling Missing Values

During dataset loading, empty texts are removed and only valid entries are retained.

Activity 1.6: Handling Duplicates

Duplicates are automatically eliminated when mapping clean text.

Activity 1.7: Checking for Outliers

Email content is textual, so outlier detection is performed implicitly through preprocessing rather than statistical outlier removal.

Milestone 2: Exploratory Data Analysis

Because this is an NLP dataset, analysis focuses on:

Activity 2.1: Token Length Distribution

Examining maximum and average token lengths ensures compatibility with DeBERTa's 512-token limit.

Activity 2.2: Category Distribution

Understanding class imbalance (e.g., more promotional emails than important ones).

Activity 2.3: Text Frequency Patterns

Frequent words indicating spam or promotions are identified.

Activity 2.4: Bivariate Text Relationship

For example, "offer", "discount", "free" appear strongly in promotional/spam emails.

Activity 2.5: Correlation of Categories

Mapping the original categories to broader groups:

- Important
- Promotion
- Spam

Handled by map_to_broad_category() in the pipeline.

Milestone 3: Model Building

Activity 3.1: Loading Tokenizer and Model

The system uses DeBERTa-v3:

AutoModelForSequenceClassification.from_pretrained(HF_MODEL_NAME)

Found in train_model() inside ml_pipeline.py.

Activity 3.2: Dataset Preparation

The custom EmailDataset class tokenizes all text samples and attaches labels. This appears in both ml_pipeline.py and utils.py.

Activity 3.3: Training the Model

The training loop:

- Runs for 3 epochs
- Uses AdamW optimizer
- Applies gradient clipping
- Computes loss for backpropagation

Shown in train_model() logic.

Activity 3.4: Evaluation

Predictions and true labels are compared to generate:

- Accuracy
- Confusion matrix
- Classification report

Metrics are later displayed on metrics.html.

Activity 3.5: Saving the Model

Model and tokenizer are stored in:

models/deberta_email_classifier/

Milestone 4: Performance Testing & Model Comparison

Activity 4.1: Model Metrics

The classifier reports:

- Overall accuracy
- Precision, recall, F1-score
- Confusion matrix

Displayed in metrics.html with clear tables and labels.

Activity 4.2: Confidence Analysis

Each prediction outputs a probability from the softmax function, handled in:

predict_single_email()andFlaskpredict()routes.

Milestone 5: Model Deployment

Activity 5.1: Flask Backend

app.py handles:

- Loading model & tokenizer at startup
- Tokenizing incoming emails
- Running predictions
- Returning category label + confidence
- Managing uploads in batch classification

Activity 5.2: Frontend Implementation

HTML pages provide a clean interface:

- **welcome.html** — Landing page introducing the system.
- **login.html / register.html** — Authentication UI.
- **classify.html** — Form to classify a single email.
- **batch.html** — Upload CSV for bulk classification.
- **dashboard.html** — Live monitoring dashboard.
- **result.html** — Shows formatted prediction results.

These templates incorporate Bootstrap styling, badges, icons, and responsive design.

Key UI Features

- Auto-expanding text areas
- Confidence progress bars
- Color-coded category badges
- Session-based dashboards
- CSV download for batch results

Application Screenshots and Workflow

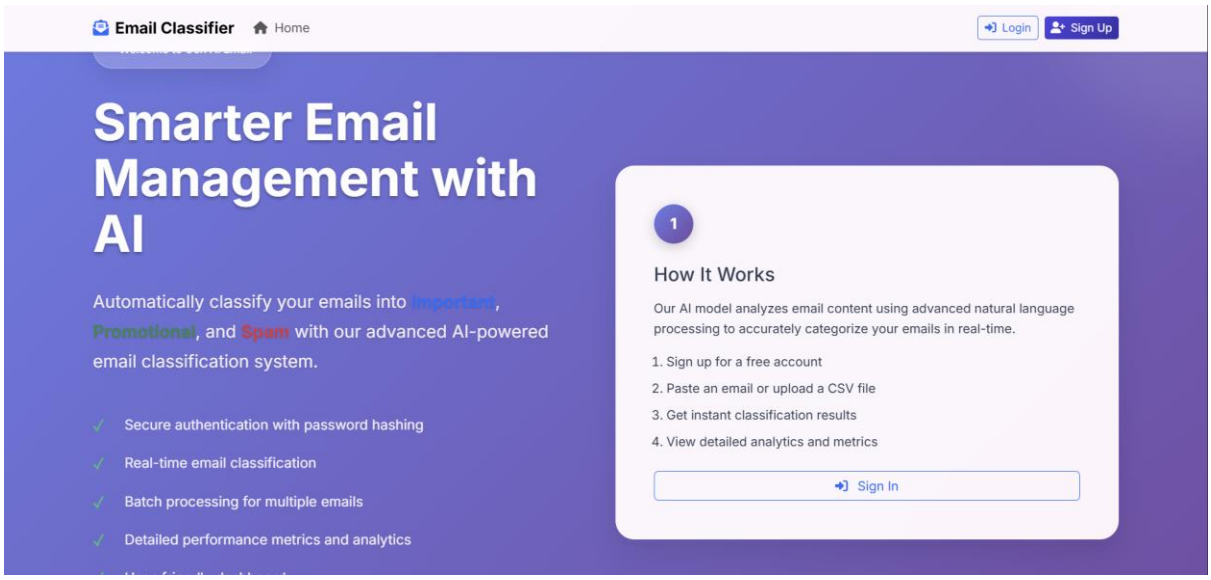
The web application has been designed with clarity and usability in mind, ensuring that users can comfortably navigate through each feature. The workflow progresses in a natural and intuitive order, beginning from the welcome interface and moving toward real-time classification, batch processing, and model evaluation. The following sections describe each major screen and its purpose within the system.

1. Welcome Interface

The first screen a user encounters is the welcome interface, which serves as an introduction to theentiresystem.
It presents a clean layout with a short description of what the platform does—namely, offering an AI-powered solution for organizing and categorizing emails.

Prominent call-to-action buttons encourage users to get started immediately, either by registering a new account or signing in.
This page also highlights system advantages such as automated sorting, time savings, and intelligent prioritization, helping users understand why the tool is useful in a modern workplace.

The visual design includes subtle animations, soft background gradients, and icons, giving the platform a polished and professional appearance.

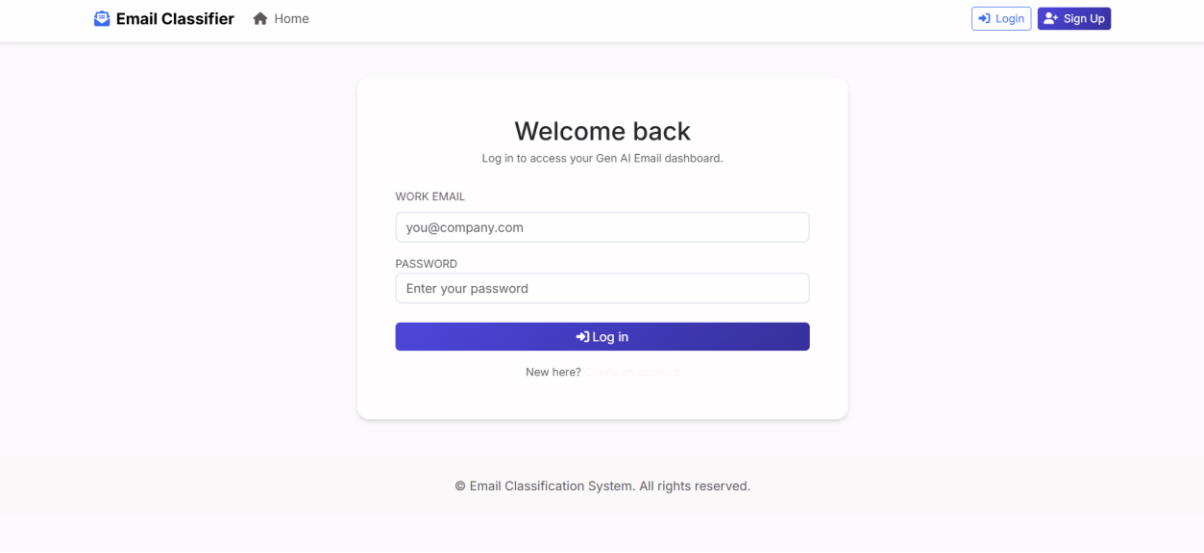


2. User Authentication Screens

Before entering the main application, users must authenticate themselves. This enhances security and ensures that email classification activities occur in a protected environment.

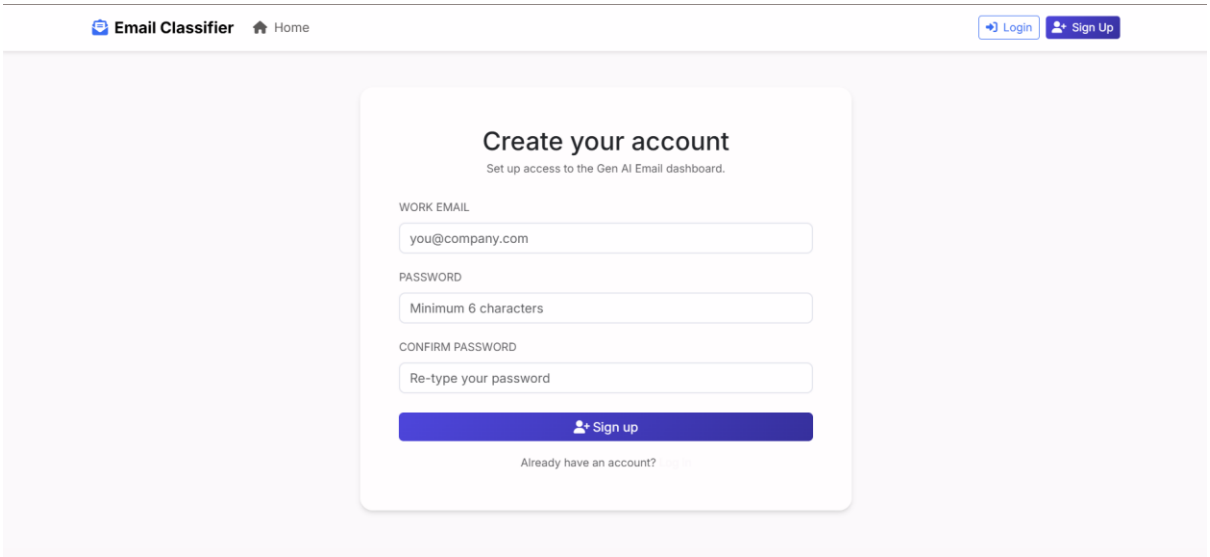
Login Page

The login page features a simple two-field form for entering credentials. Clear labeling and unobtrusive design help users focus on the login process without distraction. If incorrect details are entered, gentle error prompts guide the user to fix the mistake.



Registration Page

For new users, the registration page allows account creation through a straightforward form. Password confirmation fields help prevent typing errors, and concise instructions make onboarding smooth, especially for first-time users. Both pages maintain consistent styling, reinforcing the system's identity.



3. Single Email Classification Page

This is one of the most crucial parts of the application. The layout is designed to make the email classification process as simple and efficient as possible.

A large text box occupies the main section of the screen, inviting users to paste the complete content of any email—ideally both subject and body. This ensures that the model gathers full context before making a prediction.

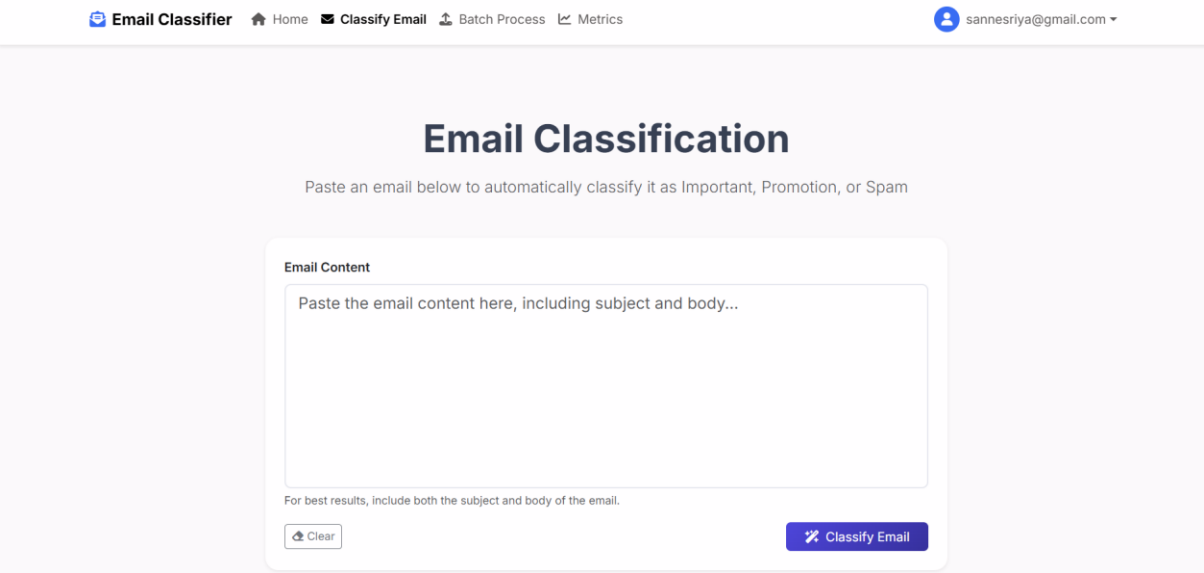
Once the user submits the email, the system instantly processes the content and displays:

- The predicted category (Important, Promotion, or Spam)
- A visually distinct badge representing the classification
- A percentage confidence score
- A short explanation describing how the decision was made

A progress bar reinforces the confidence value visually, helping users quickly understand how strongly the model feels about its prediction.

Additionally, a helpful link encourages users to explore more detailed model performance statistics if they want deeper insights.

This page focuses heavily on simplicity, speed, and clarity—allowing users to classify several emails in quick succession.



4. Interactive Dashboard

The dashboard serves as a central working area for users who want to analyze multiple emails manually.

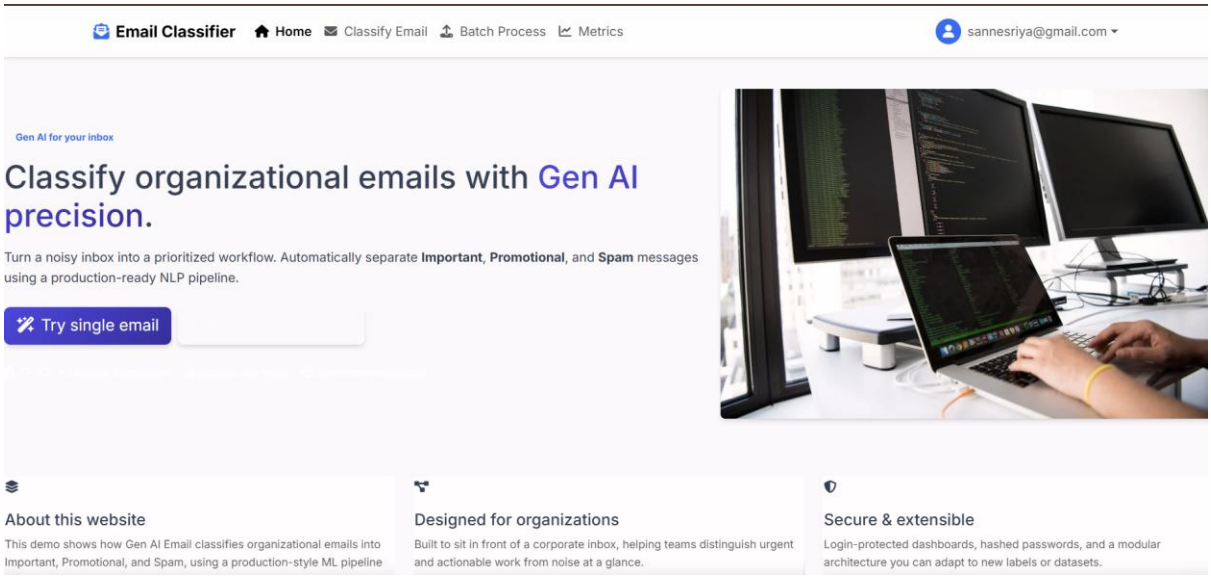
A neatly arranged input area allows users to repeatedly paste new email content without leaving the page.
Each time the user submits text, the system evaluates the content and updates a “Most Recent Prediction” section.

This section includes:

- A colored badge showing the predicted class
- A confidence bar displaying how strongly the model feels
- Numerical confidence percentages
- A short interpretation of the result

The dashboard also includes tips on how to get the best predictions, encouraging users to include complete email bodies or remove extra formatting.

This page provides a smooth, continuous workflow, making it ideal for real-time sorting of messages.



5. Batch Processing Interface

For users managing corporate mailboxes or bulk email datasets, the system includes a batch classification page designed for large-scale processing.

Here, users can upload a spreadsheet containing many email entries. Once uploaded, the system processes every row and generates a structured table of results.

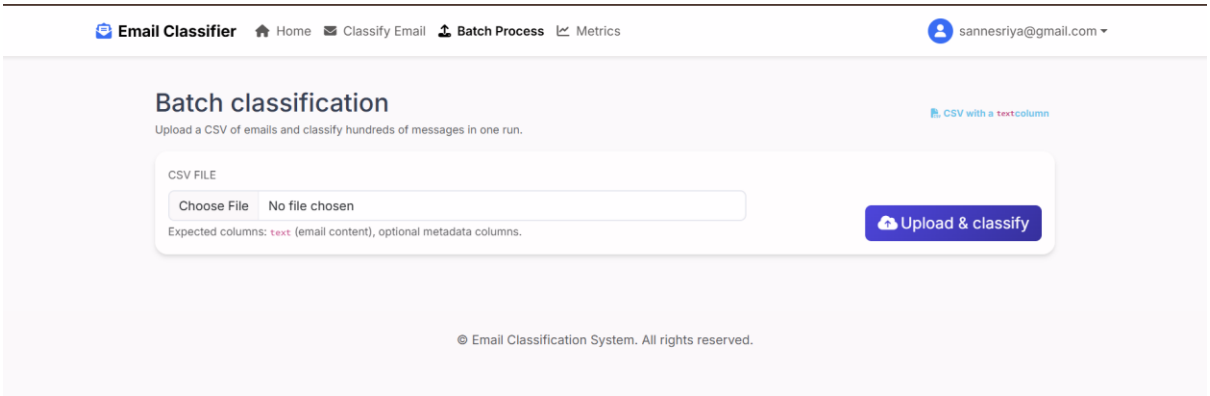
The output table includes:

- A shortened preview of each email
- The predicted label
- The associated confidence level

To enhance usability, the results table is scrollable and arranged in clean rows, ensuring that users can browse large datasets comfortably.

A download option allows users to export the processed results into a new file. This is particularly useful for organizations that need to import these predictions into other systems or reports.

This feature showcases the scalability of the email classification tool and its suitability for high-volume environments.



6. Metrics and Performance Analysis Page

A dedicated model evaluation page gives users insights into how well the system performs overall.

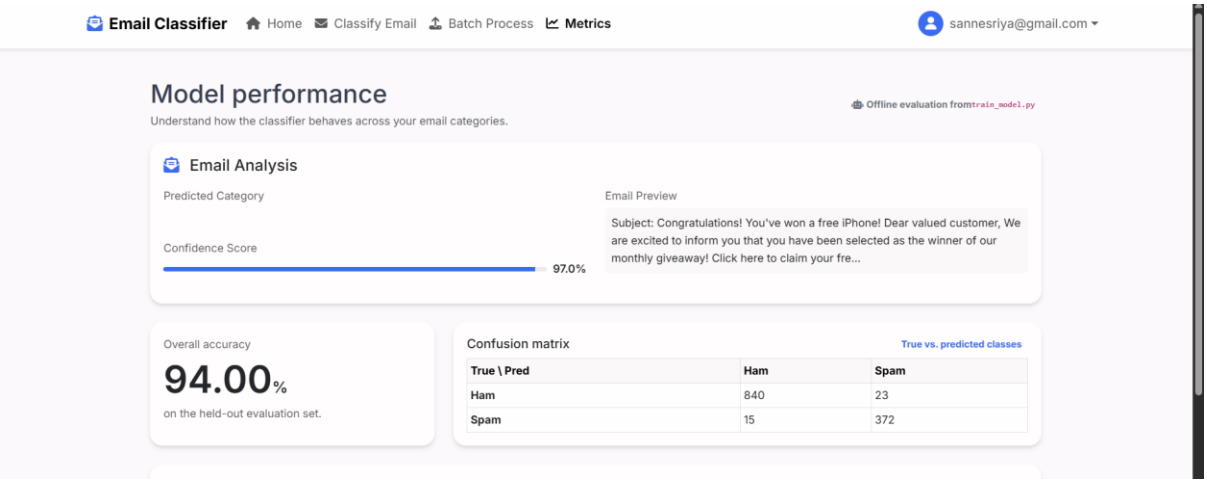
This page displays:

- The classifier's overall accuracy
- A confusion matrix, showing how often each category was correctly or incorrectly predicted
- Precision, recall, and F1-scores for each class
- A formatted classification report presented in a code-style block for easy reading

Users can scroll through the report to understand the strengths and weaknesses of the AI model. Visual cues, such as highlighted headers and neatly spaced rows, help present technical information in a digestible format.

For added clarity, the page may also show how a recently analyzed email performed under the model, including its predicted category and confidence.

This section demonstrates transparency in the system's performance, giving users confidence in its reliability and accuracy.



7. Overall Workflow Summary

The complete workflow of the system can be summarized as follows:

1. The user is greeted by an introductory page highlighting system features.
2. They sign in or create an account to access the secure workspace.
3. They can either classify a single email, use the dashboard for repeated testing, or upload a batch file.
4. The system analyzes each email using advanced language modeling techniques.
5. Predictions are displayed instantly with clear, intuitive visual cues.
6. Users interested in technical validity can explore the metrics page to review model performance.

This seamless and organized flow ensures that users at all technical levels—from beginners to experts—can interact with the system effortlessly and efficiently.

Future Implementations

The Email Classification System, in its current form, provides a reliable and efficient solution for sorting organizational emails into three major categories—Important, Promotional, and Spam. Although the system already performs well in real-world scenarios, several enhancements can significantly broaden its capabilities and improve its long-term applicability. The following ideas outline meaningful improvements that can be explored in subsequent development phases.

1. Integration with Corporate Email Platforms

A natural next step is to integrate the system directly with enterprise email services such as Outlook, Gmail, or custom mail servers.
Instead of manually copying and pasting email content, the system could automatically analyze incoming messages in real time.
This integration would allow:

- Automatic sorting of emails into folders
- Priority notifications for critical communication
- Seamless workflow across organizations

Such integration would make the system highly valuable for large teams handling high email traffic.

2. Expansion of Classification Categories

Currently, the classifier groups emails into three broad categories.
However, real-world emails often fall into more nuanced groups such as:

- Human Resources
- Finance
- Technical Support
- Announcements
- Meeting Requests
- System Alerts

Introducing additional classes would give organizations deeper insights and finer control over email routing.
With more data, the model can learn to distinguish these categories effectively.

3. Multilingual Email Classification

Many organizations operate across countries and receive messages in different languages.
Adding multilingual support would greatly increase the system's global usability.
This could be achieved by fine-tuning multilingual versions of transformer models, enabling classification in languages such as:

- Spanish

- French
- German
- Hindi
- Arabic

This enhancement would support international teams and multinational corporations.

4. Adaptive Learning & Continuous Model Updates

Over time, email patterns evolve.
Promotional messages change formats, spam strategies shift, and organizations adopt new communication styles.
A future version of the system could include:

- Automated retraining with newly labeled emails
- Periodic model updates
- Live feedback loops where users can mark misclassified emails

This creates a self-improving system that stays accurate even as communication trends change.

5. Threat Detection & Security Layer

In addition to spam identification, the system can be extended to detect:

- Phishing attempts
- Malware-linked messages
- Fraudulent content
- Suspicious links or attachments

Integrating a threat-detection module would transform this project into a more comprehensive cybersecurity tool, providing safer email handling for individuals and organizations.

6. Mobile Application for On-the-Go Classification

A companion mobile app would allow users to classify emails from their smartphones. This is especially beneficial for:

- Remote workers
- Field staff
- Managers needing quick prioritization

With a mobile-friendly layout and push notifications, users could instantly know which emails require immediate attention.

7. Analytics Dashboard for Organizational Insights

A future enhancement could include a dedicated analytics system that presents aggregated statistics such as:

- Monthly email volume
- Category distribution trends
- Peak communication periods
- Most frequent senders
- Priority alerts

These insights can help organizations understand communication patterns and improve workflow efficiency.

8. Integration with Task Management Systems

Emails often translate into actionable tasks.
A smart extension would allow the classifier to connect with:

- Jira
- Trello
- Asana
- Microsoft Teams
- Slack

This would convert important emails into tasks or alerts automatically, ensuring nothing slips through the cracks.

Conclusion

The Email Classification System successfully demonstrates how advanced natural language processing techniques can be applied to streamline communication in modern organizations. By leveraging a transformer-based model capable of understanding context and linguistic patterns, the system offers reliable categorization of emails into Important, Promotional, and Spam groups. This allows users to focus their attention on high-priority messages while reducing the distraction caused by unwanted or low-value content.

Throughout the development process, the project followed a structured workflow that included data preparation, text preprocessing, model training, evaluation, and deployment. The resulting system is both accurate and user-friendly, offering real-time predictions through an intuitive interface. Features such as confidence scoring, batch processing, and a detailed performance dashboard further enhance its practicality, making the tool suitable for both individual and organizational use.

Beyond its current capabilities, the project lays a strong foundation for continued growth. Several promising enhancements—such as multilingual support, deeper email categorization, integration with corporate mail platforms, and intelligent threat detection—can further extend its usefulness. These future directions highlight the adaptability of the system and its potential to evolve into a comprehensive email-management and communication-support platform.

Overall, the project achieves its primary goal of transforming raw email text into actionable, categorized information with speed and accuracy. It showcases the power of AI-driven automation in improving productivity, reducing information overload, and supporting efficient digital communication. The successful deployment of the email classifier illustrates how machine learning solutions can be translated from theoretical concepts into practical tools that deliver measurable value in real-world environments.