****

**Name: Ahsan Ali**

**Roll No: 061**

**Task: 9**

**Section: BSAI-4B**

**Subject: PAI Lab**

**Submitted To: Sir Rasikh**

**Title:**

**Smart Email Classifier using NLP & Flask**

**Objective:**

To design and develop a Flask-based NLP application that automatically classifies the content of an email into predefined categories such as Work, Personal, Promotions, or Spam using machine learning techniques.

**Tools & Technologies Used:**

| **Tool/Library** |  |  | **Purpose** |
| --- | --- | --- | --- |
| Python |  |  | Core programming language |
| Flask |  |  | Lightweight web framework |
| Scikit-learn |  |  | ML pipeline (Tfidf + Naive Bayes) |
| Pandas |  |  | Dataset handling |
| Joblib |  |  | Save/load trained model |
| HTML/CSS |  |  | Frontend design |
| NLTK |  |  | Text preprocessing |

**Dataset Description:**

| **Text** | **Category** |
| --- | --- |
| "Meeting at 11 AM with client" | Work |
| "Hey! Want to hang out tonight?" | Personal |
| "Huge Sale! Get 50% off electronics" | Promotions |
| "You've won a prize! Click here to claim." | Spam |

**ML Model Training:**

* Text Vectorization: TfidfVectorizer
* Classifier: Multinomial Naive Bayes
* Split: 80% Train, 20% Test
* Pipeline: Tfidf → Naive Bayes
* Model File: email\_classifier.pkl

**Web Application Overview:**

**Backend: app.py**

* Loads model
* Accepts POST request with email text
* Predicts category and returns result to template

**Frontend: index.html**

* Simple textarea form
* Button to classify
* Displays result with styling

**User Interface Screenshots:**

**Code:**

**App.py**

from flask import Flask, render\_template, request

import joblib

app = Flask(\_\_name\_\_)

model = joblib.load('email\_classifier.pkl')

@app.route('/', methods=['GET', 'POST'])

def index():

    result = None

    if request.method == 'POST':

        email = request.form['email']

        prediction = model.predict([email])[0]

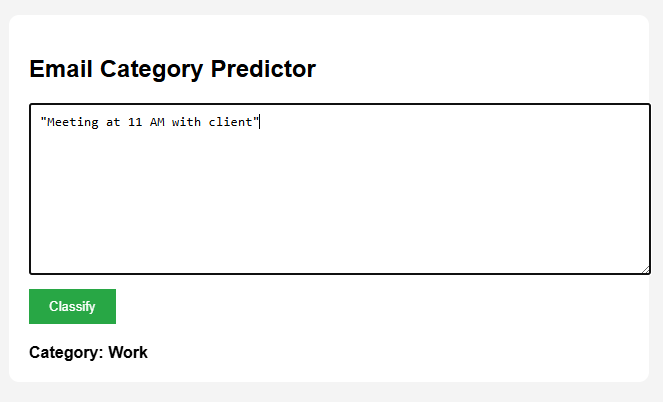
        result = f"Category: {prediction}"

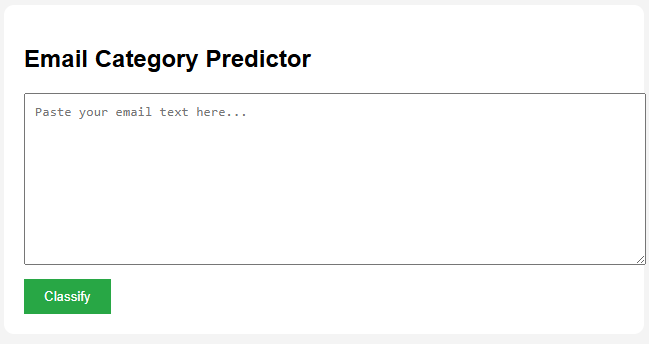
    return render\_template('index.html', result=result)

if \_\_name\_\_ == '\_\_main\_\_':

    app.run(debug=True)

**Output:**

****

****

**Conclusion:**

* Successfully built a smart NLP system to classify emails into relevant categories.
* Implemented Tfidf + Naive Bayes pipeline for fast and accurate results.
* Deployed it as a Flask web application with clean frontend UI.
* Understood the importance of preprocessing, model saving, and user interaction.