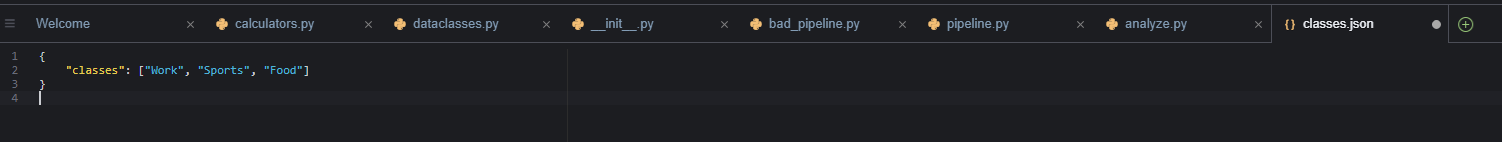
**MLOps Homework 2 - Email Classification**Modifying an existing email classification system to load classes from a file, allowing users to add new classes via an API, and demonstrating how the system works with additional classes.  
  
 **1. Clone the mlops\_sentiment\_lab repo**

<https://github.com/jhoward/mlops_sentiment_lab>  
  
 ANS: Cloned the repo in bash  
  
**2. Remove the hardcoded classes and change them to a ones that are loaded from a file**

ANS: A screenshot of a computer

AI-generated content may be incorrect.  
  
 1. **Created a classes.json file** to store class labels dynamically.

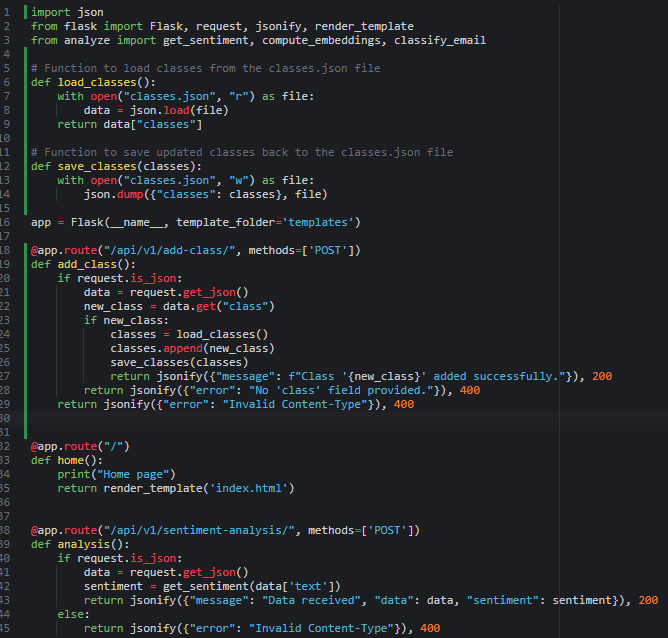


2. **Modified the code of Analyse.py:** to read the classes from classes.json instead of using hardcoded values.

3. **Implemented a function (load\_classes())** that loads class labels from the file dynamically.  
  
A screen shot of a computer

AI-generated content may be incorrect.  
  
4. **Classifying Example**:  
A screenshot of a computer

AI-generated content may be incorrect.

**3. Allow users to add new classes via an API. This should update the class file:**  
  
ANS: To allow users to add new classes, I implemented an API using **Flask  
  
Flask API Code**:  
  
  
A computer screen shot of a program code

AI-generated content may be incorrect.

**Method**: POST  
  
When a POST request is made to the /add class endpoint, the system reads the class name from the JSON payload.  
  
A screenshot of a computer

AI-generated content may be incorrect.

appends the new class if it’s not already in the list, and saves the updated classes back to the file.

A screenshot of a computer

AI-generated content may be incorrect.

**4. Demonstrate this working with many additional classes. Invoke the  
classification endpoints with some different email text to show the system  
end 2 end.**  
  
ANS: We’ll be working with many additional classes:

Work,Sports,Food,Travel,Finance,Shopping,Health,Entertainment,Family,Education,Technology,Politics

A screenshot of a computer

AI-generated content may be incorrect.

After adding the new classes, I invoked the classification endpoint using curl to classify a sample email.  
  
**Email Text: "I am going to a football match tomorrow!"**  
  
$ curl -X POST http://127.0.0.1:3000/api/v1/classify/ -H "Content-Type: application/json" -d '{"text": "I am going to a football match tomorrow!"}'

A screenshot of a computer

AI-generated content may be incorrect.

({"classifications":[{"class":"Sports","similarity":0.33779534697532654},{"class":"Food","similarity":0.15516254305839539},{"class":"Work","similarity":0.1520056426525116},{"class":"Family","similarity":0.1493988037109375},{"class":"Entertainment","similarity":0.10950297117233276},{"class":"Finance","similarity":0.10245766490697861},{"class":"Politics","similarity":0.10082963109016418},{"class":"Health","similarity":0.09934098273515701},{"class":"Travel","similarity":0.09581462293863297},{"class":"Shopping","similarity":0.07858021557331085},{"class":"Education","similarity":0.07420410960912704},{"class":"Technology","similarity":0.026121918112039566}],"message":"Email classified"})  
  
The system automatically identifies the classes ("Sports" has more value) based on the text  
  
  
**Email Text: "I just got a promotion at work!"**  
  
A computer screen with white text

AI-generated content may be incorrect.  
  
The classification automatically detects "Work" has more value based on the job promotion text.  
**Email Text: "I just booked a flight for my vacation!"**A computer screen with white text

AI-generated content may be incorrect.The system detects "Travel" as the highest similarity due to the reference to vacation and flight booking.  
  
**Github Link: https://github.com/vsirigiri0828/Assignment-2**