Lab Assignment – 4.5

Hall Ticket No.: 2303A510E6

Name – Anushka Boora

Batch – 29

Suppose that you work for a company that receives hundreds of customer emails daily. Management wants to automatically classify emails into categories like "Billing", "Technical Support", "Feedback", and "Others" before assigning them to appropriate departments. Instead of training a new model, your task is to use prompt engineering techniques with an existing LLM to handle the classification.

Tasks to be completed are as below

a. Prepare Sample Data:

• Create or collect 10 short email samples, each belonging to one of the 4 categories.

b. Zero-shot Prompting:

• Design a prompt that asks the LLM to classify a single email without providing any examples.

• Example prompt:

“Classify the following email into one of the following categories: Billing, Technical Support, Feedback, Others. Email: ‘I have not received my invoice for last month.’”

c. One-shot Prompting:

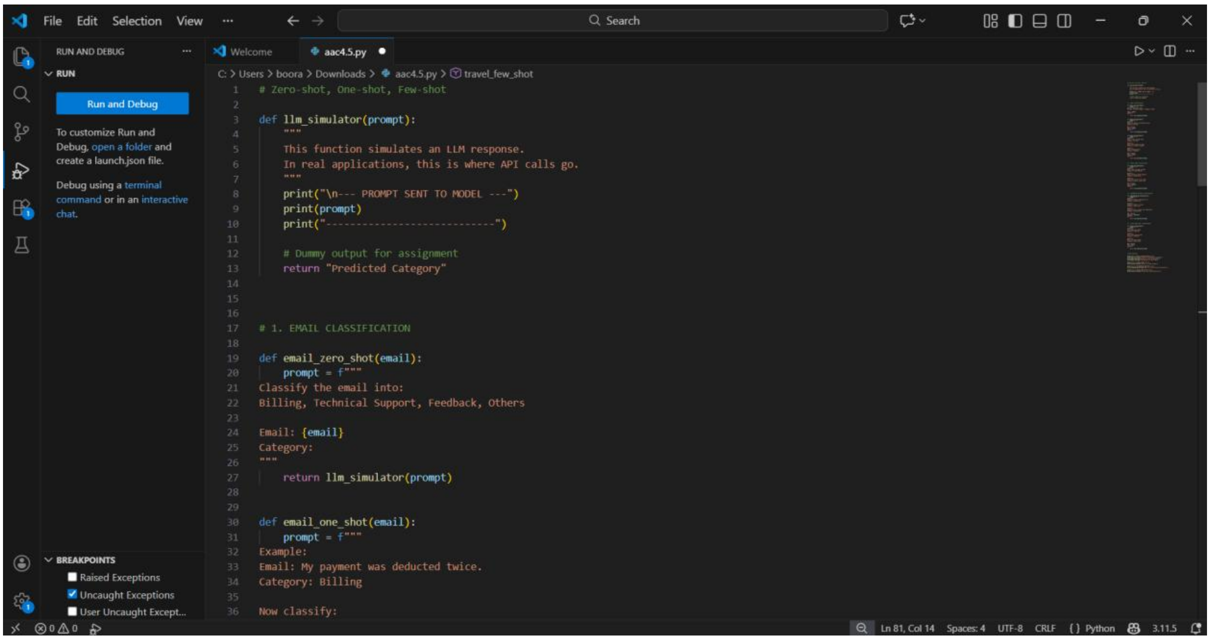
• Add one labeled example before asking the model to classify a new email.

d. Few-shot Prompting:

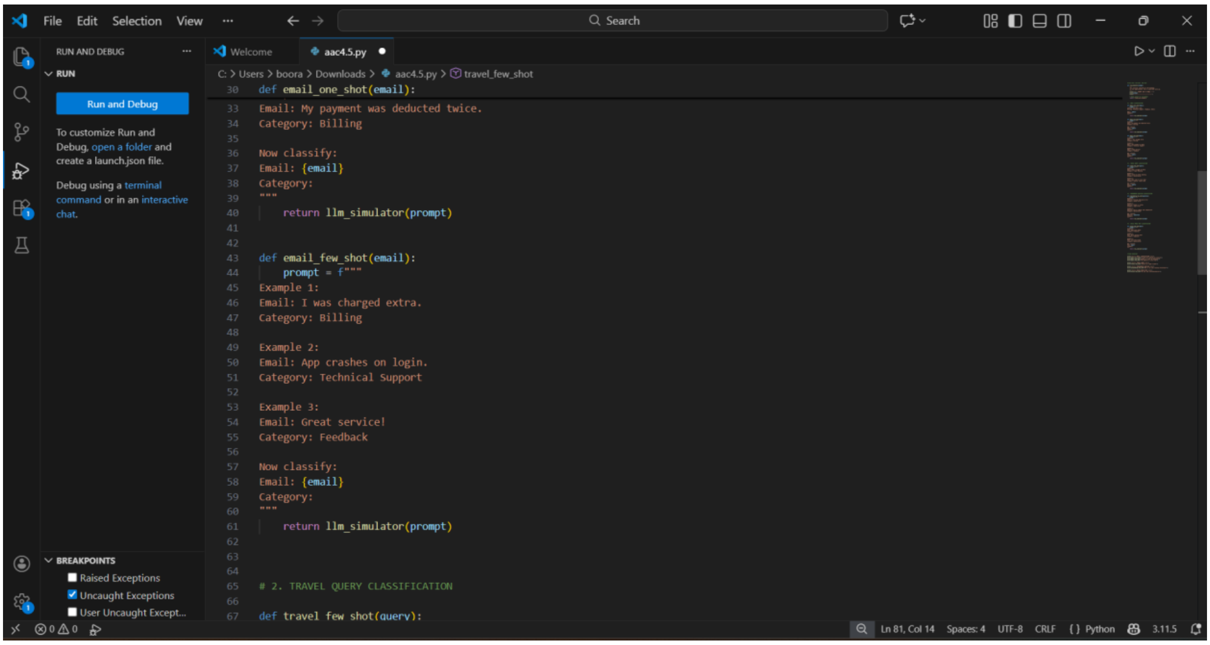
• Use 3–5 labeled examples in your prompt before asking the model to classify a new email.

e. Evaluation:

- Run all three techniques on the same set of 5 test emails.
- Compare and document the accuracy and clarity of responses.


File Edit Selection View ...
Search
RUN AND DEBUG
RUN
Run and Debug
To customize Run and Debug, open a folder and create a launch.json file.
Debug using a terminal command or in an interactive chat.
BREAKPOINTS
□ Raised Exceptions
☑ Uncaught Exceptions
□ User Uncaught Except...
Welcome
aac4.5.py
C: > Users > boora > Downloads > aac4.5.py > travel_few_shot
Ln 81, Col 14 Spaces: 4 UTF-8 CRLF {} Python 3.11.5

```python
# Zero-shot, One-shot, Few-shot

def llm_simulator(prompt):
    """
    This function simulates an LLM response.
    In real applications, this is where API calls go.
    """
    print("\n--- PROMPT SENT TO MODEL ---")
    print(prompt)
    print("---------------------------")

    # Dummy output for assignment
    return "Predicted Category"



# 1. EMAIL CLASSIFICATION

def email_zero_shot(email):
    prompt = f"""
Classify the email into:
Billing, Technical Support, Feedback, Others

Email: {email}
Category:
"""
    return llm_simulator(prompt)


def email_one_shot(email):
    prompt = f"""
Example:
Email: My payment was deducted twice.
Category: Billing

Now classify:
```


File Edit Selection View ...
Search
RUN AND DEBUG
RUN
Run and Debug
To customize Run and Debug, open a folder and create a launch.json file.
Debug using a terminal command or in an interactive chat.
BREAKPOINTS
□ Raised Exceptions
☑ Uncaught Exceptions
□ User Uncaught Except...
Welcome
aac4.5.py
C: > Users > boora > Downloads > aac4.5.py > travel_few_shot
Ln 81, Col 14 Spaces: 4 UTF-8 CRLF {} Python 3.11.5

```python
def email_one_shot(email):
Email: My payment was deducted twice.
Category: Billing

Now classify:
Email: {email}
Category:
"""
    return llm_simulator(prompt)


def email_few_shot(email):
    prompt = f"""
Example 1:
Email: I was charged extra.
Category: Billing

Example 2:
Email: App crashes on login.
Category: Technical Support

Example 3:
Email: Great service!
Category: Feedback

Now classify:
Email: {email}
Category:
"""
    return llm_simulator(prompt)



# 2. TRAVEL QUERY CLASSIFICATION

def travel_few_shot(query):
```

2. Travel Query Classification

Scenario:

A travel assistant must classify queries into Flight Booking, Hotel Booking, Cancellation, or General Travel Info.

Tasks:

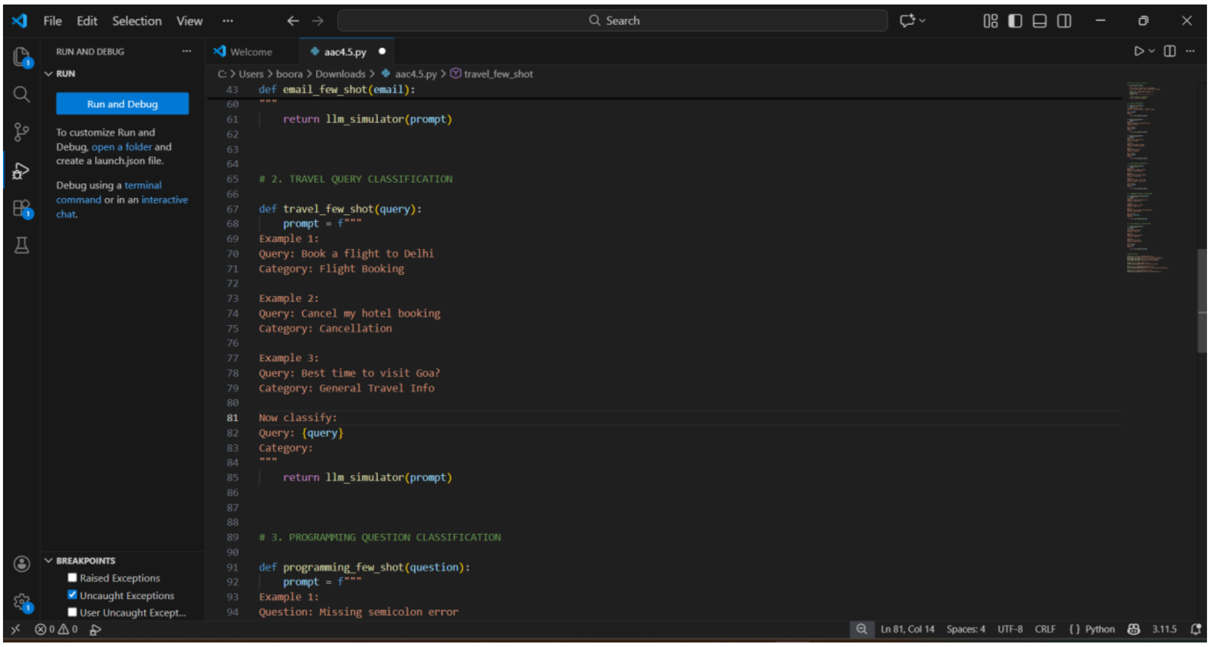
a. Prepare labeled travel queries.

b. Apply Zero-shot prompting.

c. Apply One-shot prompting.

d. Apply Few-shot prompting.

e. Compare response consistency.



To customize Run and Debug, open a folder and create a launch.json file.

Debug using a terminal command or in an interactive chat.

```python
def email_few_shot(email):
"""
    return llm_simulator(prompt)



# 2. TRAVEL QUERY CLASSIFICATION

def travel_few_shot(query):
    prompt = f"""
Example 1:
Query: Book a flight to Delhi
Category: Flight Booking

Example 2:
Query: Cancel my hotel booking
Category: Cancellation

Example 3:
Query: Best time to visit Goa?
Category: General Travel Info

Now classify:
Query: {query}
Category:
"""
    return llm_simulator(prompt)



# 3. PROGRAMMING QUESTION CLASSIFICATION

def programming_few_shot(question):
    prompt = f"""
Example 1:
Question: Missing semicolon error
```

3. Programming Question Type Identification

Scenario:

A coding help chatbot must classify queries into Syntax Error, Logic Error, Optimization, or Conceptual Question.

Tasks:

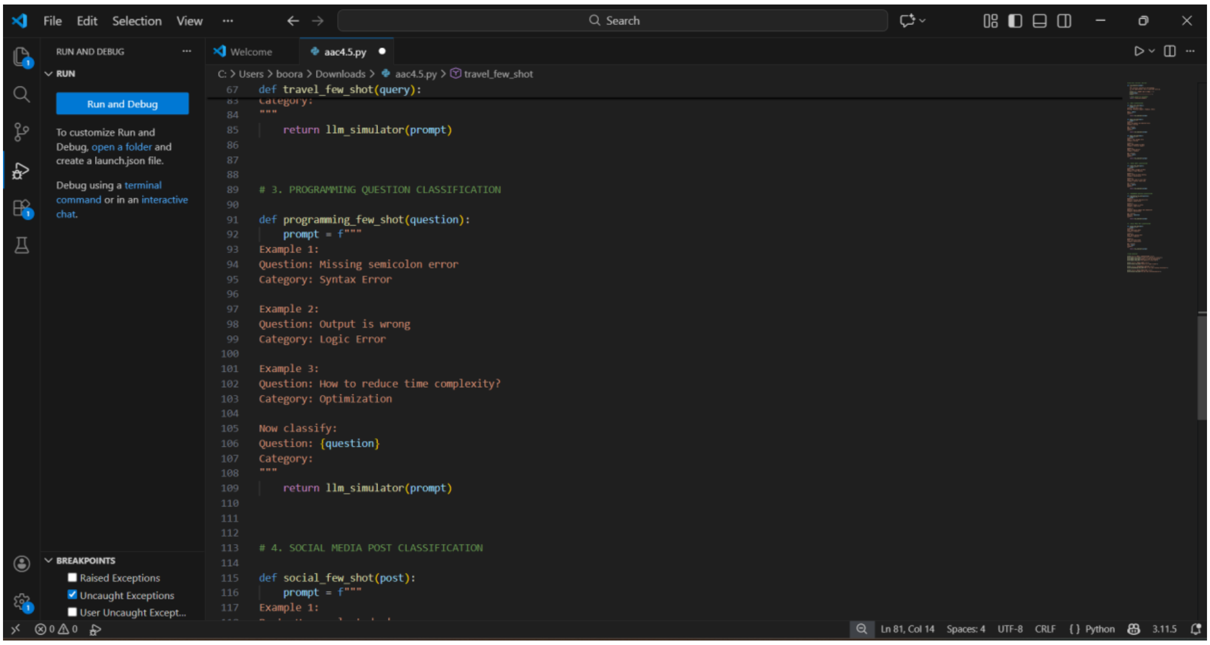
a. Prepare coding-related user queries.

b. Perform Zero-shot classification.

c. Perform One-shot classification.

d. Perform Few-shot classification.

e. Analyze improvements in technical accuracy.



To customize Run and Debug, open a folder and create a launch.json file.

Debug using a terminal command or in an interactive chat.

☐ Raised Exceptions
☑ Uncaught Exceptions
☐ User Uncaught Except...

```python
def travel_few_shot(query):
Category:
"""
    return llm_simulator(prompt)



# 3. PROGRAMMING QUESTION CLASSIFICATION

def programming_few_shot(question):
    prompt = f"""
Example 1:
Question: Missing semicolon error
Category: Syntax Error

Example 2:
Question: Output is wrong
Category: Logic Error

Example 3:
Question: How to reduce time complexity?
Category: Optimization

Now classify:
Question: {question}
Category:
"""
    return llm_simulator(prompt)



# 4. SOCIAL MEDIA POST CLASSIFICATION

def social_few_shot(post):
    prompt = f"""
Example 1:
```

4. Social Media Post Categorization

Scenario:

A social media analytics tool must classify posts into Promotion,

Complaint, Appreciation, or Inquiry.

Tasks:

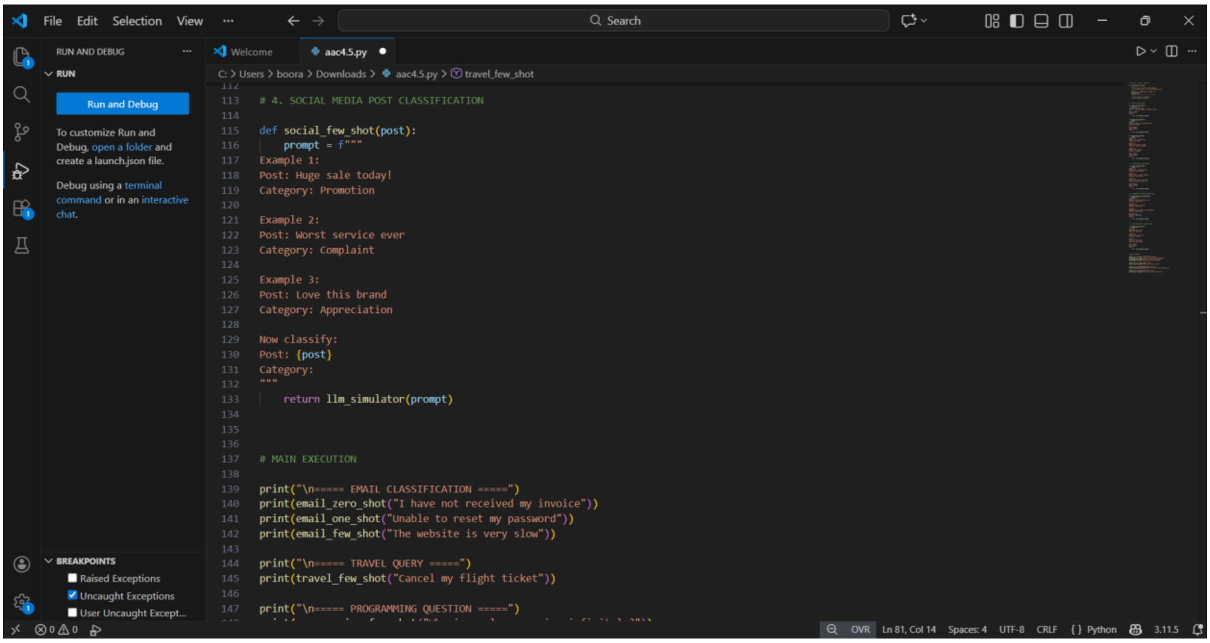
1. Prepare sample social media posts.

2. Use Zero-shot prompting.

3. Use One-shot prompting.

4. Use Few-shot prompting.

5. Analyze informal language handling.

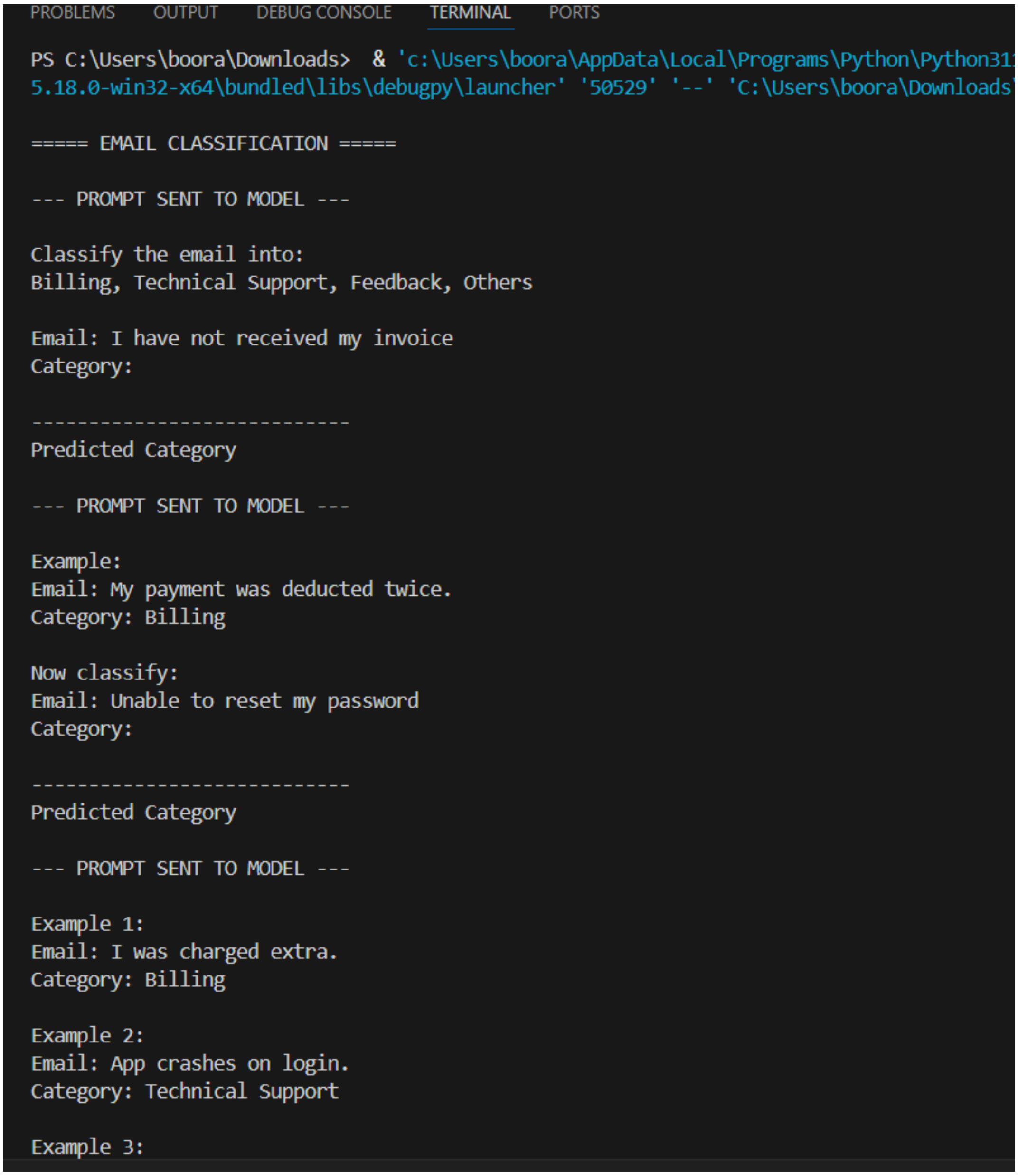


To customize Run and Debug, open a folder and create a launch.json file.

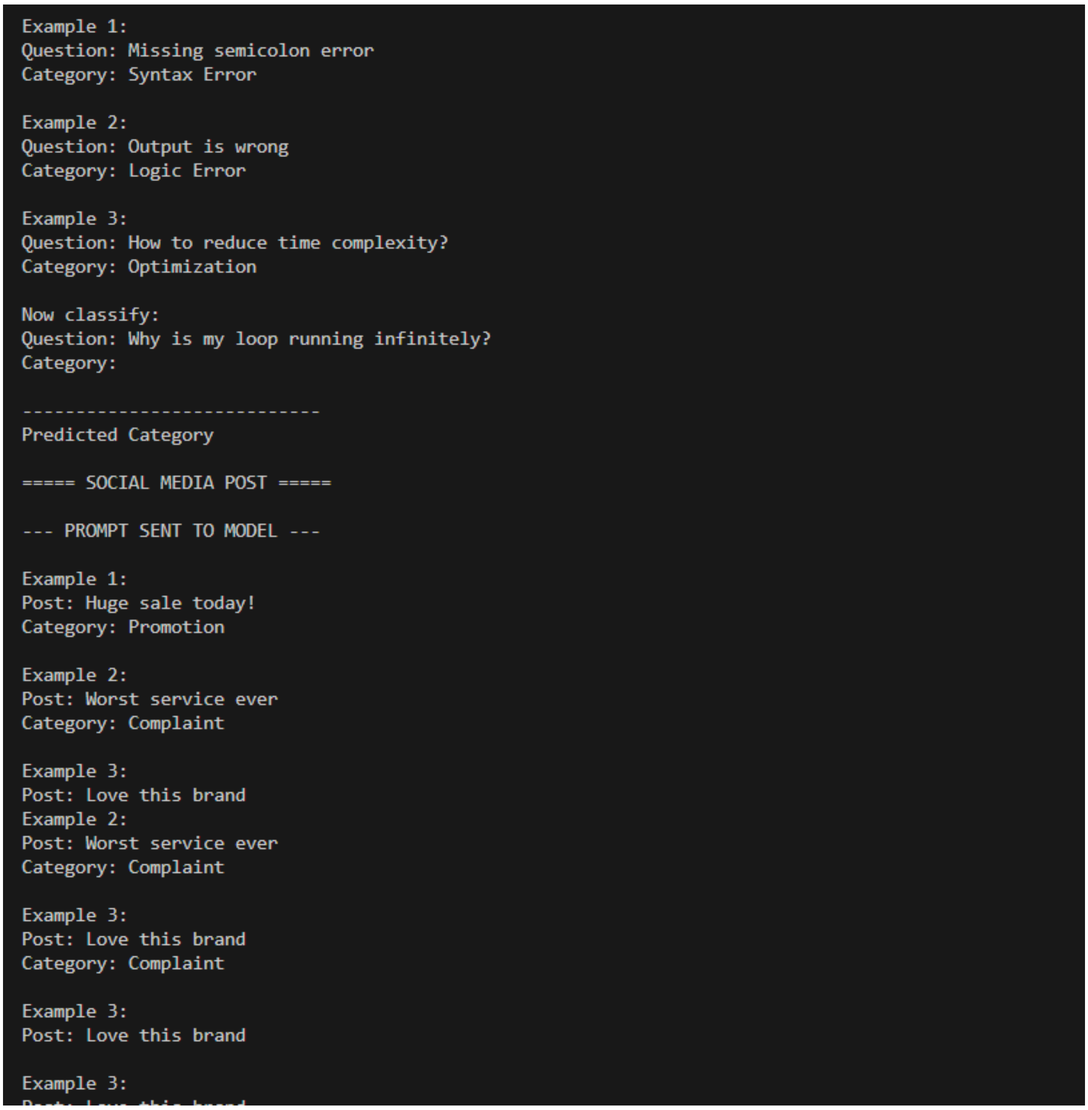
Debug using a terminal command or in an interactive chat.

```python
# 4. SOCIAL MEDIA POST CLASSIFICATION

def social_few_shot(post):
    prompt = f"""
Example 1:
Post: Huge sale today!
Category: Promotion

Example 2:
Post: Worst service ever
Category: Complaint

Example 3:
Post: Love this brand
Category: Appreciation

Now classify:
Post: {post}
Category:
"""
    return llm_simulator(prompt)



# MAIN EXECUTION

print("\n===== EMAIL CLASSIFICATION =====")
print(email_zero_shot("I have not received my invoice"))
print(email_one_shot("Unable to reset my password"))
print(email_few_shot("The website is very slow"))

print("\n===== TRAVEL QUERY =====")
print(travel_few_shot("Cancel my flight ticket"))

print("\n===== PROGRAMMING QUESTION =====")
```

OUTPUT:



```
PS C:\Users\boora\Downloads>  & 'c:\Users\boora\AppData\Local\Programs\Python\Python31
5.18.0-win32-x64\bundled\libs\debugpy\launcher' '50529' '--' 'C:\Users\boora\Downloads'

===== EMAIL CLASSIFICATION =====

--- PROMPT SENT TO MODEL ---

Classify the email into:
Billing, Technical Support, Feedback, Others

Email: I have not received my invoice
Category:

----------------------------
Predicted Category

--- PROMPT SENT TO MODEL ---

Example:
Email: My payment was deducted twice.
Category: Billing

Now classify:
Email: Unable to reset my password
Category:

----------------------------
Predicted Category

--- PROMPT SENT TO MODEL ---

Example 1:
Email: I was charged extra.
Category: Billing

Example 2:
Email: App crashes on login.
Category: Technical Support

Example 3:
```



```
Example 1:
Question: Missing semicolon error
Category: Syntax Error

Example 2:
Question: Output is wrong
Category: Logic Error

Example 3:
Question: How to reduce time complexity?
Category: Optimization

Now classify:
Question: Why is my loop running infinitely?
Category:

----------------------------
Predicted Category

===== SOCIAL MEDIA POST =====

--- PROMPT SENT TO MODEL ---

Example 1:
Post: Huge sale today!
Category: Promotion

Example 2:
Post: Worst service ever
Category: Complaint

Example 3:
Post: Love this brand
Example 2:
Post: Worst service ever
Category: Complaint

Example 3:
Post: Love this brand
Category: Complaint

Example 3:
Post: Love this brand

Example 3:
Post: Love this brand
```