ASSIGNMENT-4.5

M.HARIKRISHNA

2203A52100

B-50

Task 1

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

Prompt:

Classify the email into Billing, Technical Support, Feedback, or Others. Email: "I have not received my invoice."

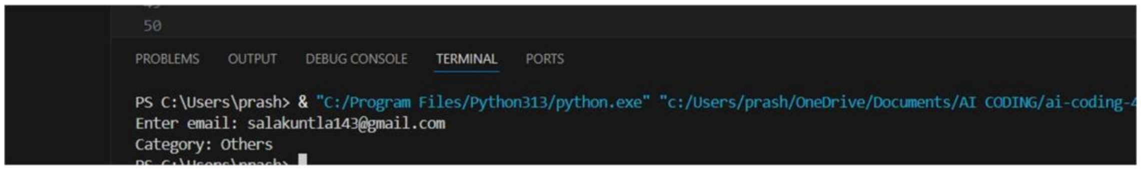
Code:

```
def classify_email(email):
    email = email.lower()

    if "invoice" in email or "payment" in email or "refund" in email:
        return "Billing"     elif "error" in email or "login" in
email or "crash" in email:
        return "Technical Support"     elif "love" in email or
"good" in email or "excellent" in email:
        return "Feedback"
else:
        return "Others"

email = input("Enter email: ")
print("Category:", classify_email(email))
```

Output:



```
PROBLEMS    OUTPUT    DEBUG CONSOLE    TERMINAL    PORTS

PS C:\Users\prash> & "C:/Program Files/Python313/python.exe" "c:/Users/prash/OneDrive/Documents/AI CODING/ai-coding-4
Enter email: salakuntla143@gmail.com
Category: Others
```

Code Explanation:

A leap year is divisible by 4 but not by 100 unless divisible by 400.

This task demonstrates how prompt engineering can classify emails without training a new model. Zero-shot prompting works by directly asking the model to classify without examples. One-shot improves accuracy by giving one example to guide the model. Few-shot provides multiple examples which improves classification consistency. The Python code simulates classification using keyword matching, but real LLMs use semantic understanding.

Task 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.

Prompt:

Classify query into Flight Booking, Hotel Booking, Cancellation, or General Travel Info.
Query: "Cancel my ticket."

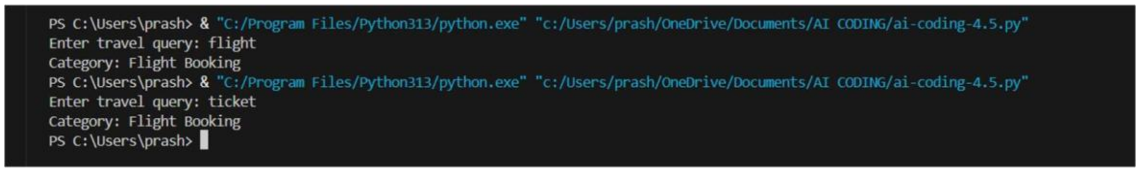
Code:

```
def classify_travel(query):
query = query.lower()


  if "flight" in query or "ticket" in query:
      return "Flight Booking"    elif
"hotel" in query or "room" in query:
      return "Hotel Booking"    elif "cancel"
in query or "refund" in query:
      return "Cancellation"
else:
      return "General Travel Info"


query = input("Enter travel query: ")
print("Category:", classify_travel(query))
```

Output:



```
PS C:\Users\prash> & "C:/Program Files/Python313/python.exe" "c:/Users/prash/OneDrive/Documents/AI CODING/ai-coding-4.5.py"
Enter travel query: flight
Category: Flight Booking
PS C:\Users\prash> & "C:/Program Files/Python313/python.exe" "c:/Users/prash/OneDrive/Documents/AI CODING/ai-coding-4.5.py"
Enter travel query: ticket
Category: Flight Booking
PS C:\Users\prash> 
```

Code Explanation:

This system classifies travel queries using prompt engineering concepts. Few-shot prompting would help when queries are ambiguous. For example, “change booking” may need examples to classify correctly.

Task 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.

Prompt:

Classify query into Syntax Error, Logic Error, Optimization, Conceptual Question. Query: "My loop is not working correctly."

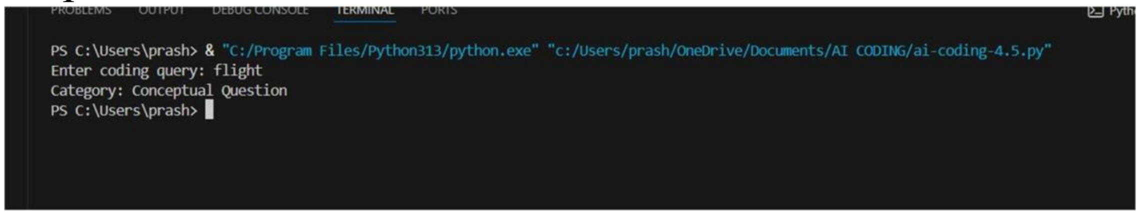
Code:

```
def classify_programming(query):
    query = query.lower()


    if "syntax" in query or "indent" in query:
        return "Syntax Error"     elif "wrong output"
in query or "logic" in query:
        return "Logic Error"     elif "optimize"
in query or "fast" in query:
        return "Optimization"
else:
    return "Conceptual Question"

query = input("Enter coding query: ")
print("Category:", classify_programming(query))
```

Output:



```
PS C:\Users\prash> & "C:/Program Files/Python313/python.exe" "c:/Users/prash/OneDrive/Documents/AI CODING/ai-coding-4.5.py"
Enter coding query: flight
Category: Conceptual Question
PS C:\Users\prash>
```

Code Explanation:

Few-shot prompting improves technical classification because coding queries can be complex. Examples help AI distinguish between logic errors and syntax errors.

Task 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

Prompt:

Classify post into Promotion, Complaint, Appreciation, Inquiry.

Post: "Your product is amazing!"

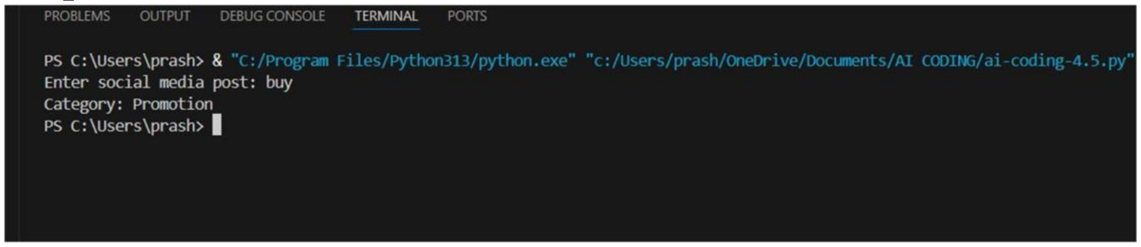
Code:

```
def classify_post(post):
post = post.lower()

    if "buy" in post or "offer" in post or "sale" in post:
        return "Promotion"    elif "bad" in post or "worst"
in post or "issue" in post:
        return "Complaint"    elif "love"
in post or "great" in post:
        return "Appreciation"
else:
        return "Inquiry"


post = input("Enter social media post: ")
print("Category:", classify_post(post))
```

Output:



```
PS C:\Users\prash> & "C:/Program Files/Python313/python.exe" "c:/Users/prash/OneDrive/Documents/AI CODING/ai-coding-4.5.py"
Enter social media post: buy
Category: Promotion
PS C:\Users\prash> 
```

Code explanation:

Social media language is informal, so few-shot prompting helps LLM understand slang and short sentences. Example posts improve classification reliability.