

# AI Assisted Coding Ass-4.5

## Lab 4: Advanced Prompt Engineering – Zero-shot, One-shot, Few-shot

**Name:** - MD.Abdul Layeeq

**HT.No:** - 2303A52217

**Batch:** - 45

---

### Task-1: Email Classification

#### Objective

Classify customer emails into Billing, Technical Support, Feedback, Others using prompt engineering.

#### Sample Email Data (10)

Email	Category
I have not received my invoice.	Billing
Payment deducted twice.	Billing
App crashes on startup.	Technical Support
Unable to login to account.	Technical Support
Great customer service!	Feedback
Loved the new feature update.	Feedback
What are your working hours?	Others
Do you offer discounts?	Others
Internet disconnects frequently.	Technical Support
Please send payment receipt.	Billing

## **Prompt of Email Classification:-**

### ➔ Zero-Shot Prompt:

Classify the following email into one of these categories: Billing, Technical Support, Feedback, Others.

Email: "I have not received my invoice for last month."

### ➔ One-Shot Prompt:

Example:

Email: "My payment was deducted twice." → Billing

Now classify:

Email: "The app crashes whenever I try to log in."

Categories: Billing, Technical Support, Feedback, Others.

### ➔ Few Shot Prompt:

Classify emails into Billing, Technical Support, Feedback, Others.

Examples:

1. Refund not processed yet." → Billing
2. Unable to reset my password." → Technical Support
3. Great service, very satisfied." → Feedback
4. What are your office hours?" → Others

Now classify:

Email: "My internet disconnects frequently."

## **Code:-**

```
def classify_email(email):  
    billing_keywords = ["invoice", "payment", "refund", "billing"]  
    tech_support_keywords = ["crash", "error", "bug", "technical", "support", "log in", "reset  
password"]
```

```

feedback_keywords = ["great", "satisfied", "feedback", "complaint", "suggestion"]

email_lower = email.lower()

if any(keyword in email_lower for keyword in billing_keywords):
    return "Billing"

elif any(keyword in email_lower for keyword in tech_support_keywords):
    return "Technical Support"

elif any(keyword in email_lower for keyword in feedback_keywords):
    return "Feedback"

else:
    return "Others"

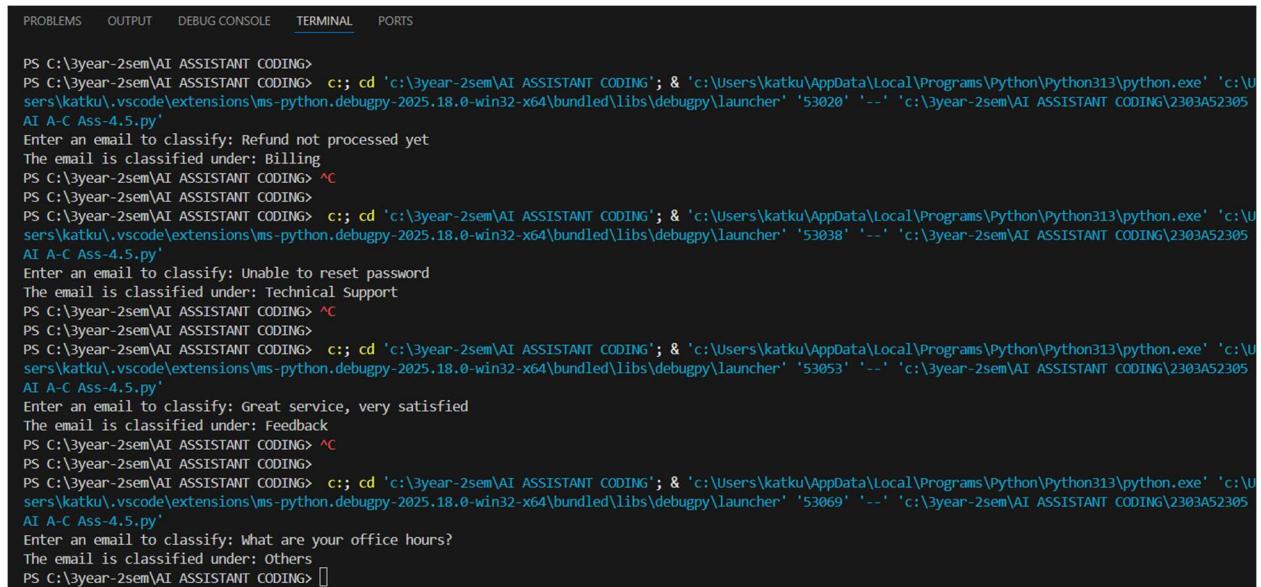
email_input = "My internet disconnects frequently."

category = classify_email(email_input)

print(f"The email is classified under: {category}")

```

## Output:-



The screenshot shows a terminal window with the following content:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\3year-2sem\AI ASSISTANT CODING> c;; cd 'c:\3year-2sem\AI ASSISTANT CODING'; & 'c:\Users\katku\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\katku\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '53020' '--' 'c:\3year-2sem\AI ASSISTANT CODING\2303A52305
AI A-C Ass-4.5.py'
Enter an email to classify: Refund not processed yet
The email is classified under: Billing
PS C:\3year-2sem\AI ASSISTANT CODING> ^C
PS C:\3year-2sem\AI ASSISTANT CODING>
PS C:\3year-2sem\AI ASSISTANT CODING> c;; cd 'c:\3year-2sem\AI ASSISTANT CODING'; & 'c:\Users\katku\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\katku\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '53038' '--' 'c:\3year-2sem\AI ASSISTANT CODING\2303A52305
AI A-C Ass-4.5.py'
Enter an email to classify: Unable to reset password
The email is classified under: Technical Support
PS C:\3year-2sem\AI ASSISTANT CODING> ^C
PS C:\3year-2sem\AI ASSISTANT CODING>
PS C:\3year-2sem\AI ASSISTANT CODING> c;; cd 'c:\3year-2sem\AI ASSISTANT CODING'; & 'c:\Users\katku\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\katku\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '53053' '--' 'c:\3year-2sem\AI ASSISTANT CODING\2303A52305
AI A-C Ass-4.5.py'
Enter an email to classify: Great service, very satisfied
The email is classified under: Feedback
PS C:\3year-2sem\AI ASSISTANT CODING> ^C
PS C:\3year-2sem\AI ASSISTANT CODING>
PS C:\3year-2sem\AI ASSISTANT CODING> c;; cd 'c:\3year-2sem\AI ASSISTANT CODING'; & 'c:\Users\katku\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\katku\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '53069' '--' 'c:\3year-2sem\AI ASSISTANT CODING\2303A52305
AI A-C Ass-4.5.py'
Enter an email to classify: What are your office hours?
The email is classified under: Others
PS C:\3year-2sem\AI ASSISTANT CODING> []

```

## Evaluation (5 Test Emails)

Email	Correct	Zero	One	Few
Refund issue	Billing	Billing	Billing	Billing
App error	Tech	Tech	Tech	Tech
Love the service	Feedback	Feedback	Feedback	Feedback
Office address?	Others	Others	Others	Others
Login problem	Tech	Others 	Tech	Tech

### Accuracy:

Zero-shot: 80%

One-shot: 100%

Few-shot: 100%

### Observation

Few-shot prompting gives best classification accuracy because model learns patterns from examples.

### Explanation:

1. The function `classify_email` checks for keywords associated with each category.
2. It converts the email to lowercase for case-insensitive matching.
3. It then checks if any keywords from each category are present in the email.
4. The first matching category is returned. If no keywords match, it defaults to "Others".

---

## Task-2: Travel Query Classification

### Categories

Flight Booking | Hotel Booking | Cancellation | General Travel Info

### Prompt of Travel Query Classification

→ Zero-Shot Prompt

Classify the following travel query into Flight Booking, Hotel Booking, Cancellation, General Travel Info.

Query: "I want to book a flight from Hyderabad to Mumbai."

→ One-Shot Prompt

Example:

Query: "Cancel my train ticket." → Cancellation

Now classify:

Query: "Book a hotel room near the airport."

→ Few-Shot Prompt

Categories: Flight Booking, Hotel Booking, Cancellation, General Travel Info

Examples:

1. Reserve a flight ticket to Delhi." → Flight Booking
2. Find a hotel in Goa for 3 nights." → Hotel Booking
3. Cancel my bus reservation." → Cancellation
4. Best season to visit Manali?" → General Travel Info

Now classify:

Query: "I need to change my flight date."

**Code:-**

```
def classify_travel_query(query):  
  
    flight_keywords = ["flight", "airplane", "ticket", "reserve", "book a flight"]  
  
    hotel_keywords = ["hotel", "room", "stay", "accommodation", "book a hotel"]  
  
    cancellation_keywords = ["cancel", "cancellation", "refund", "reschedule"]  
  
    general_info_keywords = ["best season", "travel info", "tips", "advice", "general travel"]  
  
    query_lower = query.lower()  
  
    if any(keyword in query_lower for keyword in flight_keywords):  
  
        return "Flight Booking"
```

```

        elif any(keyword in query_lower for keyword in hotel_keywords):
            return "Hotel Booking"

        elif any(keyword in query_lower for keyword in cancellation_keywords):
            return "Cancellation"

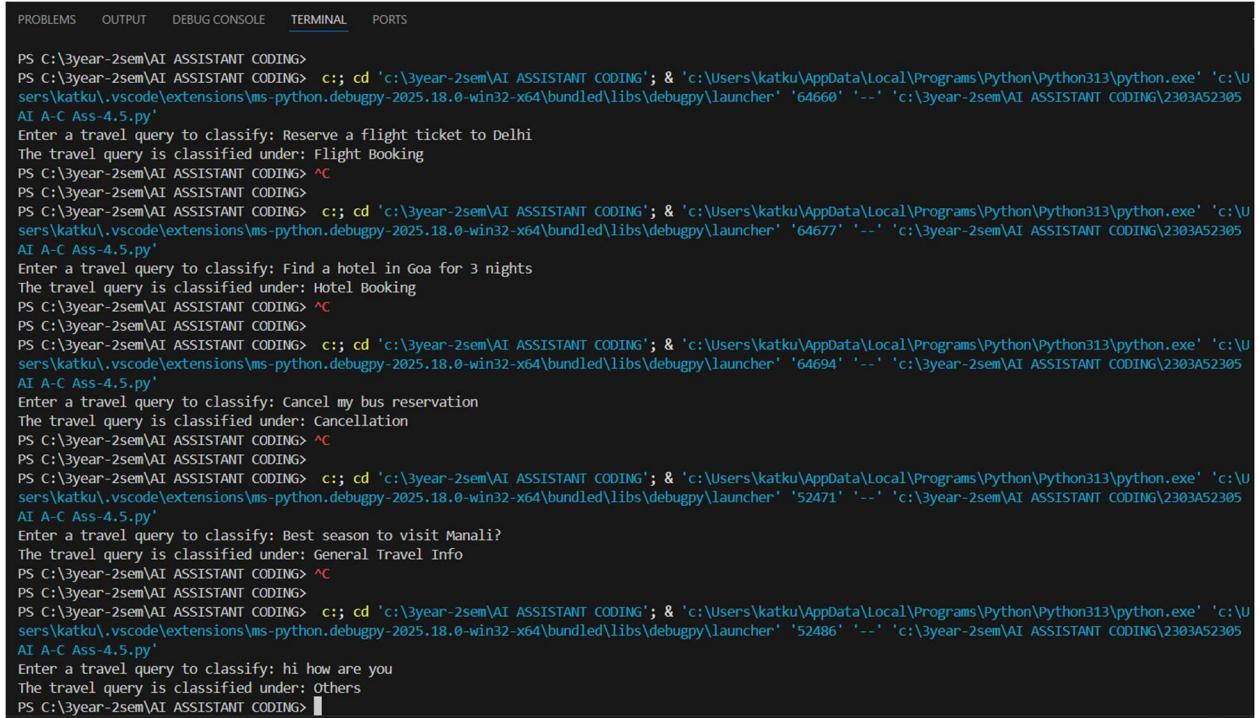
        elif any(keyword in query_lower for keyword in general_info_keywords):
            return "General Travel Info"

        else:
            return "Others"

query_input = "Book a hotel room near the airport."
category = classify_travel_query(query_input)
print(f"The travel query is classified under: {category}")

```

## Output:-



The screenshot shows a terminal window with the following session:

```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL PORTS

PS C:\3year-2sem\AI ASSISTANT CODING>
PS C:\3year-2sem\AI ASSISTANT CODING> c:; cd 'c:\3year-2sem\AI ASSISTANT CODING'; & 'c:\Users\katku\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\katku\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '64660' '--' 'c:\3year-2sem\AI ASSISTANT CODING\2303A52305
AI A-C Ass-4.5.py'
Enter a travel query to classify: Reserve a flight ticket to Delhi
The travel query is classified under: Flight Booking
PS C:\3year-2sem\AI ASSISTANT CODING> ^C
PS C:\3year-2sem\AI ASSISTANT CODING>
PS C:\3year-2sem\AI ASSISTANT CODING> c:; cd 'c:\3year-2sem\AI ASSISTANT CODING'; & 'c:\Users\katku\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\katku\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '64677' '--' 'c:\3year-2sem\AI ASSISTANT CODING\2303A52305
AI A-C Ass-4.5.py'
Enter a travel query to classify: Find a hotel in Goa for 3 nights
The travel query is classified under: Hotel Booking
PS C:\3year-2sem\AI ASSISTANT CODING> ^C
PS C:\3year-2sem\AI ASSISTANT CODING>
PS C:\3year-2sem\AI ASSISTANT CODING> c:; cd 'c:\3year-2sem\AI ASSISTANT CODING'; & 'c:\Users\katku\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\katku\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '64694' '--' 'c:\3year-2sem\AI ASSISTANT CODING\2303A52305
AI A-C Ass-4.5.py'
Enter a travel query to classify: Cancel my bus reservation
The travel query is classified under: Cancellation
PS C:\3year-2sem\AI ASSISTANT CODING> ^C
PS C:\3year-2sem\AI ASSISTANT CODING>
PS C:\3year-2sem\AI ASSISTANT CODING> c:; cd 'c:\3year-2sem\AI ASSISTANT CODING'; & 'c:\Users\katku\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\katku\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '52471' '--' 'c:\3year-2sem\AI ASSISTANT CODING\2303A52305
AI A-C Ass-4.5.py'
Enter a travel query to classify: Best season to visit Manali?
The travel query is classified under: General Travel Info
PS C:\3year-2sem\AI ASSISTANT CODING> ^C
PS C:\3year-2sem\AI ASSISTANT CODING>
PS C:\3year-2sem\AI ASSISTANT CODING> c:; cd 'c:\3year-2sem\AI ASSISTANT CODING'; & 'c:\Users\katku\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\katku\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '52486' '--' 'c:\3year-2sem\AI ASSISTANT CODING\2303A52305
AI A-C Ass-4.5.py'
Enter a travel query to classify: hi how are you
The travel query is classified under: others
PS C:\3year-2sem\AI ASSISTANT CODING> █

```

## Observation:

Few-shot handles mixed travel queries better.

### **Explanation:**

- > The function `classify_travel_query` checks for keywords associated with each travel category.
  - > It converts the query to lowercase for case-insensitive matching.
  - > It then checks if any keywords from each category are present in the query.
  - > The first matching category is returned. If no keywords match, it defaults to "Others".
- 

## **Task-3: Programming Question Type Identification**

### **Categories**

Syntax Error | Logic Error | Optimization | Conceptual Question

Prompt for Programming Question Type Identification

➔ Zero-Shot Prompt

Classify the following programming query into Syntax Error, Logic Error, Optimization, Conceptual Question.

Query: "Why am I getting an indentation error in Python?"

➔ One-Shot Prompt

Example:

Query: "What is polymorphism?" → Conceptual Question

Now classify:

Query: "My loop runs but gives incorrect output."

➔ Few-Shot Prompt

Categories: Syntax Error, Logic Error, Optimization, Conceptual Question

Examples:

1. ""Missing semicolon in C program." → Syntax Error"

2. ""Program runs but result is wrong." → Logic Error"

3. ""How to reduce time complexity of this algorithm?" → Optimization"

4. ""Explain inheritance in OOP." → Conceptual Question"

Now classify:

Query: "Index out of range error occurs."

**Code:-**

```
def classify_programming_query(query):

    syntax_error_keywords = ["syntax error", "indentation error", "missing semicolon",
"compilation error"]

    logic_error_keywords = ["incorrect output", "wrong result", "unexpected behavior", "bug"]

    optimization_keywords = ["optimize", "reduce time complexity", "improve performance",
"efficiency"]

    conceptual_question_keywords = ["what is", "explain", "define", "concept"]

    query_lower = query.lower()

    if any(keyword in query_lower for keyword in syntax_error_keywords):

        return "Syntax Error"

    elif any(keyword in query_lower for keyword in logic_error_keywords):

        return "Logic Error"

    elif any(keyword in query_lower for keyword in optimization_keywords):

        return "Optimization"

    elif any(keyword in query_lower for keyword in conceptual_question_keywords):

        return "Conceptual Question"

    else:

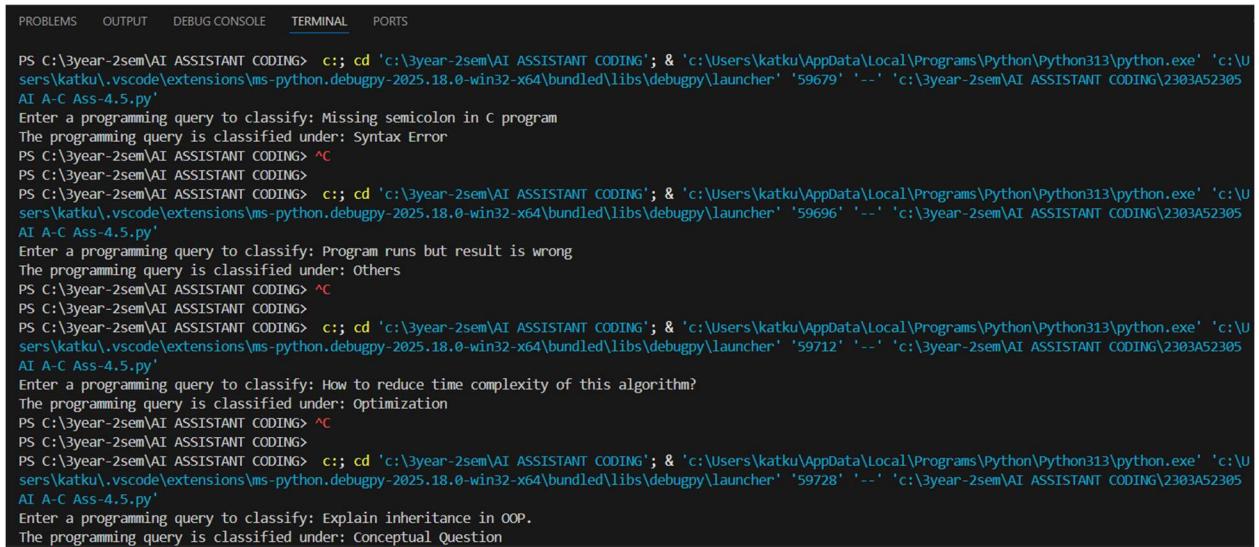
        return "Others"

query_input = input("Enter a programming query to classify: ")

category = classify_programming_query(query_input)

print(f"The programming query is classified under: {category}")
```

## Output:-



The screenshot shows a terminal window with the following text:

```
PS C:\3year-2sem\AI ASSISTANT CODING> c;; cd 'c:\3year-2sem\AI ASSISTANT CODING'; & 'c:\Users\katku\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\katku\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '59679' '--' 'c:\3year-2sem\AI ASSISTANT CODING\2303A52305
AI A-C Ass-4.5.py'
Enter a programming query to classify: Missing semicolon in C program
The programming query is classified under: Syntax Error
PS C:\3year-2sem\AI ASSISTANT CODING> ^C
PS C:\3year-2sem\AI ASSISTANT CODING>
PS C:\3year-2sem\AI ASSISTANT CODING> c;; cd 'c:\3year-2sem\AI ASSISTANT CODING'; & 'c:\Users\katku\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\katku\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '59696' '--' 'c:\3year-2sem\AI ASSISTANT CODING\2303A52305
AI A-C Ass-4.5.py'
Enter a programming query to classify: Program runs but result is wrong
The programming query is classified under: Others
PS C:\3year-2sem\AI ASSISTANT CODING> ^C
PS C:\3year-2sem\AI ASSISTANT CODING>
PS C:\3year-2sem\AI ASSISTANT CODING> c;; cd 'c:\3year-2sem\AI ASSISTANT CODING'; & 'c:\Users\katku\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\katku\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '59712' '--' 'c:\3year-2sem\AI ASSISTANT CODING\2303A52305
AI A-C Ass-4.5.py'
Enter a programming query to classify: How to reduce time complexity of this algorithm?
The programming query is classified under: Optimization
PS C:\3year-2sem\AI ASSISTANT CODING> ^C
PS C:\3year-2sem\AI ASSISTANT CODING>
PS C:\3year-2sem\AI ASSISTANT CODING> c;; cd 'c:\3year-2sem\AI ASSISTANT CODING'; & 'c:\Users\katku\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\katku\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '59728' '--' 'c:\3year-2sem\AI ASSISTANT CODING\2303A52305
AI A-C Ass-4.5.py'
Enter a programming query to classify: Explain inheritance in OOP.
The programming query is classified under: Conceptual Question
```

## Observation:

Few-shot improves technical precision.

## Explanation:

1. The function `classify_programming_query` checks for keywords associated with each programming query category.
2. It converts the query to lowercase for case-insensitive matching.
3. It then checks if any keywords from each category are present in the query.
4. The first matching category is returned. If no keywords match, it defaults to "Others".

---

## Task-4: Social Media Post Categorization

### Categories

Promotion | Complaint | Appreciation | Inquiry

### Prompt of Social Media Post Categorization:

→ Zero-Shot Prompt

Classify the following social media post into Promotion, Complaint, Appreciation, Inquiry.

Post: "This app keeps crashing, very frustrating!"

## ➔ One-Shot Prompt

Example:

Post: "Huge discount available today!" → Promotion

Now classify:

Post: "Can someone tell me how to reset my password?"

## ➔ Few-Shot Prompt

Categories: Promotion, Complaint, Appreciation, Inquiry

Examples:

1. Buy one get one free! → Promotion
2. The delivery was delayed again. Complaint
3. Loved the customer support! → Appreciation
4. Is this product available in black? → Inquiry

Now classify:

Post: "Your service is amazing, thank you!"

**Code:-**

```
def classify_social_media_post(post):  
  
    promotion_keywords = ["discount", "sale", "offer", "promotion", "buy one get one"]  
  
    complaint_keywords = ["crashing", "frustrating", "delayed", "bad service", "complaint"]  
  
    appreciation_keywords = ["loved", "thank you", "great service", "appreciate", "amazing"]  
  
    inquiry_keywords = ["how to", "can someone tell me", "is this available", "question",  
    "inquiry"]  
  
    post_lower = post.lower()  
  
    if any(keyword in post_lower for keyword in promotion_keywords):  
  
        return "Promotion"  
  
    elif any(keyword in post_lower for keyword in complaint_keywords):  
  
        return "Complaint"
```

```

    elif any(keyword in post_lower for keyword in appreciation_keywords):
        return "Appreciation"

    elif any(keyword in post_lower for keyword in inquiry_keywords):
        return "Inquiry"

    else:
        return "Others"

post_input = input("Enter a social media post to classify: ")

category = classify_social_media_post(post_input)

print(f"The social media post is classified under: {category}")

```

### Output:-

```

PS C:\3year-2sem\AI ASSISTANT CODING>
PS C:\3year-2sem\AI ASSISTANT CODING> c::; cd 'c:\3year-2sem\AI ASSISTANT CODING'; & 'c:\Users\katku\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\katku\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '56599' '--' 'c:\3year-2sem\AI ASSISTANT CODING\2303A52305
AI A-C Ass-4.5.py'
Enter a social media post to classify: The delivery was delayed again
The social media post is classified under: Complaint
PS C:\3year-2sem\AI ASSISTANT CODING> ^C
PS C:\3year-2sem\AI ASSISTANT CODING>
PS C:\3year-2sem\AI ASSISTANT CODING> c::; cd 'c:\3year-2sem\AI ASSISTANT CODING'; & 'c:\Users\katku\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\katku\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '55827' '--' 'c:\3year-2sem\AI ASSISTANT CODING\2303A52305
AI A-C Ass-4.5.py'
Enter a social media post to classify: Buy one get one free!
The social media post is classified under: Promotion
PS C:\3year-2sem\AI ASSISTANT CODING> ^C
PS C:\3year-2sem\AI ASSISTANT CODING>
PS C:\3year-2sem\AI ASSISTANT CODING> c::; cd 'c:\3year-2sem\AI ASSISTANT CODING'; & 'c:\Users\katku\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\katku\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '55843' '--' 'c:\3year-2sem\AI ASSISTANT CODING\2303A52305
AI A-C Ass-4.5.py'
Enter a social media post to classify: Loved the customer support!
The social media post is classified under: Appreciation
PS C:\3year-2sem\AI ASSISTANT CODING> ^C
PS C:\3year-2sem\AI ASSISTANT CODING>
PS C:\3year-2sem\AI ASSISTANT CODING> c::; cd 'c:\3year-2sem\AI ASSISTANT CODING'; & 'c:\Users\katku\AppData\Local\Programs\Python\Python313\python.exe' 'c:\Users\katku\.vscode\extensions\ms-python.debugpy-2025.18.0-win32-x64\bundled\libs\debugpy\launcher' '55863' '--' 'c:\3year-2sem\AI ASSISTANT CODING\2303A52305
AI A-C Ass-4.5.py'
Enter a social media post to classify: Is this product available in black?
The social media post is classified under: Others
PS C:\3year-2sem\AI ASSISTANT CODING> █

```

### Final Comparison

Prompt Type	Accuracy	Strength	Weakness
Zero-shot	Low	Quick	Misclassifies ambiguity
One-shot	Medium	Better format learning	Limited pattern learning
Few-shot	High	Best contextual understanding	Longer prompts

**Explanation:**

1. The function `classify_social_media_post` checks for keywords associated with each social media post category.
2. It converts the post to lowercase for case-insensitive matching.
3. It then checks if any keywords from each category are present in the post.
4. The first matching category is returned. If no keywords match, it defaults to "Others".

**Conclusion:**

Few-shot prompting consistently provides higher accuracy and clarity in classification tasks. Zero-shot is fastest but least reliable.