Lab 4: Advanced Prompt Engineering

Zero-shot, One-shot, and Few-shot Techniques, Chain of Thoughts, Prompt

NAME: CH VISHWANATH RAO

BATCH:02

HT NO:2303A51095

Q1. Zero-Shot Prompting (Basic Lab Task)

Task:

Write a Python function that classifies a given text as Spam or Not Spam

Spam using zero-shot prompting.

Steps:

1. Construct a prompt without any examples.

2. Clearly specify the output labels.

3. Display only the predicted label.

Input:

"Congratulations! You have won a free lottery ticket."

Expected Output:

Spam

PROMPT AND CODE:

```
#Write a Python function that classifies a given text as Spam or Not based on the presence of certain keywords like 'win, free, prize, click here
# user input
```

```
def classify_message(msg):
    keys = ['win', 'free', 'prize', 'click here']
    lowered = msg.lower()
    for token in keys:
        if token in lowered:
            return "Spam"
    return "Not Spam"

user_input = input("Please paste the text to analyze: ")
classification = classify_message(user_input)
print("Classification:", classification)
```

OUTPUT:

```
PS C:\Users\chvis\OneDrive\Desktop\AI ASSIST CODING> & C:/Python314/python.exe "c:/Users/chvis/OneDrive/Desktop/AI ASSIST CODING/lab 3.
Please paste the text to analyze: hello, click the link below to claim the prizes
Classification: Spam
PS C:\Users\chvis\OneDrive\Desktop\AI ASSIST CODING>
```

Q2. One-Shot Prompting (Emotion detection)

Task:

Write a Python program that detects the emotion of a sentence

using one-shot prompting.

Emotions: [‘happy', 'sad', 'angry', 'excited', 'nervous', ’neutral’]

Steps:

1. Provide one labelled example inside the prompt.

2. Take a sentence as input.

3. Print the predicted emotion

PROMPT:

```
'''
s=ohh!
display excietment
user input
'''
```

Code:

```
def classify_emotion(s):
    excitement_words = ['excited', 'excietment', 'thrilled', 'elated', 'overjoyed', 'ecstatic', 'euphoric', 'joyful']
    happiness_words = ['happy', 'joyful', 'content', 'pleased', 'cheerful', 'delighted', 'glad', 'satisfied']
    angry_words = ['angry', 'furious', 'irate', 'livid', 'outraged', 'resentful', 'annoyed', 'frustrated']
    sad_words = ['sad', 'unhappy', 'sorrowful', 'dejected', 'downcast', 'mournful', 'gloomy', 'disheartened']
    s_lower = s.lower()
    if any(word in s_lower for word in excitement_words):
        return "excietment"
    elif any(word in s_lower for word in happiness_words):
        return "happiness"
    elif any(word in s_lower for word in angry_words):
        return "angry"
    elif any(word in s_lower for word in sad_words):
        return "sad"
    else:
        return "unknown emotion"
user_input = input("Enter a sentence expressing your emotion: ")
emotion = classify_emotion(user_input)
print(f"The emotion expressed is: {emotion}")
```

Output:

```
PS C:\Users\chvis\OneDrive\Desktop\AI ASSIST CODING> & C:/Python314/python.exe "c:/Users/chvis/OneDrive/Desktop/AI ASSIST C
Enter a sentence expressing your emotion: iam sad today
The emotion expressed is: sad
PS C:\Users\chvis\OneDrive\Desktop\AI ASSIST CODING> & C:/Python314/python.exe "c:/Users/chvis/OneDrive/Desktop/AI ASSIST C
Enter a sentence expressing your emotion: iam happy today
The emotion expressed is: happiness
PS C:\Users\chvis\OneDrive\Desktop\AI ASSIST CODING>
```

Q3. Few-Shot Prompting (Student Grading Based on Marks)

Task:

Write a Python program that predicts a student's grade based on

marks using few-shot prompting.

Grades:

['A', 'B', 'C', 'D', 'F']

Grading Criteria (to be inferred from examples):

- 90–100 → A
- 80–89 → B
- 70–79 → C
- 60–69 → D
- Below 60 → F

Code:

```python
try:
    student_marks = int(input("Enter the marks (0-100): "))
    if student_marks < 0 or student_marks > 100:
        print("Invalid input. Please enter a positive integer between 0 and 100.")
    else:
        if 90 <= student_marks <= 100:
            student_grade = 'A'
        elif 80 <= student_marks <= 89:
            student_grade = 'B'
        elif 70 <= student_marks <= 79:
            student_grade = 'C'
        elif 60 <= student_marks <= 69:
            student_grade = 'D'
        else:
            student_grade = 'F'
        print(f"The grade for {student_marks} is: {student_grade}")
except ValueError:
    print("Invalid input. Please enter a valid integer.")
```

Output:

Q4. Multi-Shot Prompting (Indian Zodiac Sign Prediction using Month Name)

Task:

Write a Python program that predicts a person's Indian Zodiac sign (Rashi) based on the month of birth (month name) using multi-shot prompting.

Indian Zodiac Order (Simplified Month-Based Model): The Indian Zodiac cycle starts in March with Mesha and follows this order:

March → Mesha

April → Vrishabha

May → Mithuna

June → Karka

July → Simha

August → Kanya

September → Tula

October → Vrischika

November → Dhanu

December → Makara

January → Kumbha

February → Meena

Code:

```
def display(sign):
    print("The zodiac sign is:", sign)
month = input("Enter the month: ").strip().lower()
if month == "march":
    display("Mesha")
elif month == "april":
    display("Vrishabha")
elif month == "may":
    display("Mithuna")
elif month == "june":
    display("Karka")
elif month == "july":
    display("Simha")
elif month == "august":
    display("Kanya")
elif month == "september":
    display("Tula")
elif month == "october":
    display("Vrischika")
elif month == "november":
    display("Dhanu")
elif month == "december":
    display("Makara")
elif month == "january":
    display("Kumbha")
elif month == "february":
    display("Meena")
else:
    print("Invalid month entered.")
```

Output:

```
PS C:\Users\chvis\OneDrive\Desktop\AI ASSIST CODING> & C:/Python314/python.exe "c:/Users/chvis/OneDrive/
Enter the month: december
The zodiac sign is: Makara
PS C:\Users\chvis\OneDrive\Desktop\AI ASSIST CODING>
```