

ASSIGNMENT - 4

KONDAPALLI DEVI PRIYANKA

2303A51108

22 - 01 - 2026

TASK - 1:

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 labeled example inside the prompt.
2. Take a sentence as input.
3. Print the predicted emotion

PROMPT:

```
# sentence = "dull", "sad"
# output: The detected emotion is: sad
# sentence = "excited" , "wahh",
# output: The detected emotion is: excited
```

CODE:

```
import random

emotions = ["happy", "sad", "angry", "excited", "bored"]
sentence = input("Enter a sentence: ")
detected_emotion = random.choice(emotions)
print(f"The detected emotion is: {detected_emotion}")
```

OUTPUT:

```
Enter a sentence: hii i am very dull
The detected emotion is: sad

== Code Execution Successful ==
```

EXPLANATION:

Here the text is taken as input and it is detected according to the words the user has given to it and verifies and says whether the person is happy,sad,excited ,etc

So according to the expected output the prompts are given

TASK - 02:

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

```
# a sentence is taken as input and we have to check whether the detected
words are spam words or not
```

CODE:

```
spam_words = ["buy now", "free", "click here", "subscribe", "limited time
offer"]

sentence = input("Enter a sentence: ").lower()

if any(spam_word in sentence for spam_word in spam_words):
    print("Spam detected")
else:
    print("No spam detected")
```

EXPLANATION:

The program checks the given text for commonly used spam-related keywords. If any such keyword is found, the text is classified as “Spam”; otherwise, it is classified as “Not Spam”. Since no examples are provided in the prompt, the logic is generated purely based on instructions, demonstrating zero-shot prompting.

TASK - 03:**TASK - 04:**

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

PROMPT:

```
# 90 - 100 : A+
# 80 - 89 : b
# 70 - 79 : c
# 60 - 69 : d
# 50 - 59 : f
```

CODE:

```
marks = int(input("Enter your marks: "))

if 90 <= marks <= 100:
    grade = "A+"
elif 80 <= marks < 90:
    grade = "B"
elif 70 <= marks < 80:
    grade = "C"
elif 60 <= marks < 70:
    grade = "D"
elif 50 <= marks < 60:
    grade = "F"
```

EXPLANATION:

Multiple input–output examples help the AI infer grading rules and implement correct conditional logic. Few-shot prompting improves accuracy by clearly defining grade boundaries, resulting in structured and reliable decision-making.

TASK - 05:

Task: Write a Python program that determines whether a student Passes or Fails based on marks using Chain-of-Thought (CoT) prompting.

Result Categories:

['Pass', 'Fail']

PROMPT:

```
# chain of thoughts
#take input of the student marks
# validate the marks a see if it is in the range of 0 to 100
# then assign grades based on the marks
# print the grade
```

CODE:

```
marks = int(input("Enter your marks: "))
if 90 <= marks <= 100:
    grade = "A+"
elif 80 <= marks < 90:
    grade = "B"
elif 70 <= marks < 80:
    grade = "C"
elif 60 <= marks < 70:
    grade = "D"
elif 50 <= marks < 60:
    grade = "F"
else:
    grade = "Fail"
print(f"Your grade is: {grade}")
```

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

```
# Examples:  
# March → Mesha  
# April → Vrishabha  
# May → Mithuna  
# Predict Indian Zodiac sign based on birth month name.
```

```
def indian_zodiac_sign(month):  
    month = month.lower()  
    zodiac_signs = {  
        "january": "Makara",  
        "february": "Kumbha",  
        "march": "Mesha",  
        "april": "Vrishabha",  
        "may": "Mithuna",  
        "june": "Karka",  
        "july": "Simha",  
        "august": "Kanya",  
        "september": "Tula",  
        "october": "Vrishchika",  
        "november": "Dhanu",  
        "december": "Makara"  
    }  
    return zodiac_signs.get(month, "Invalid month name")
```