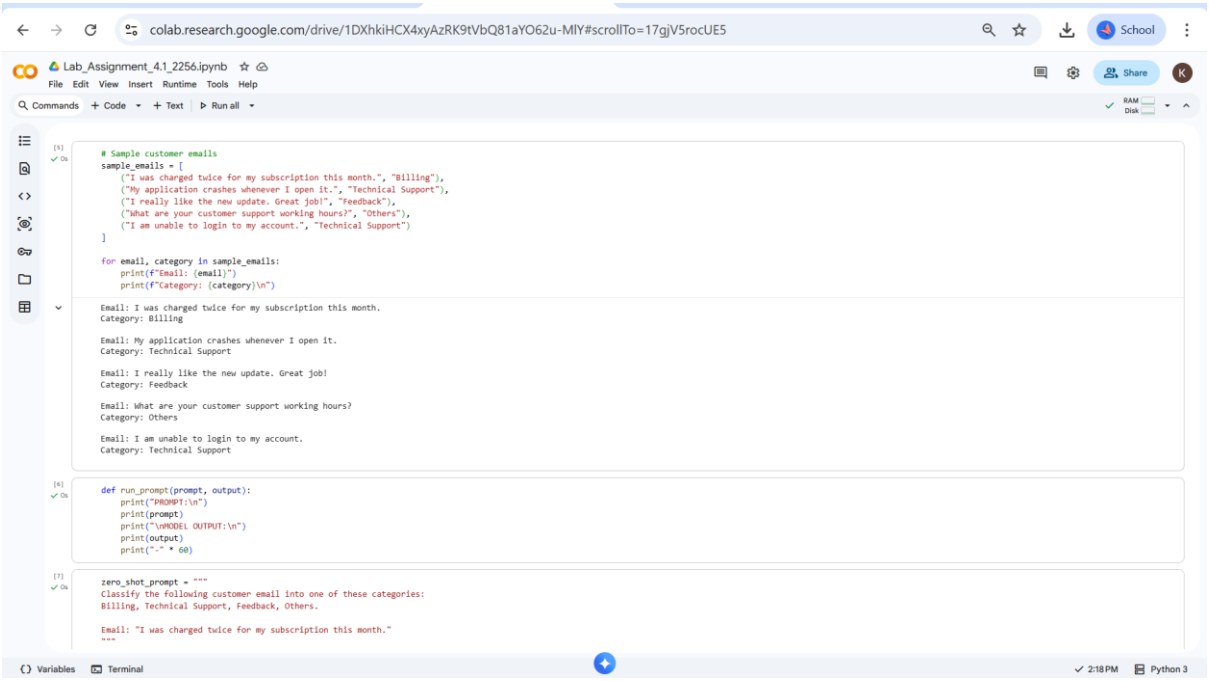
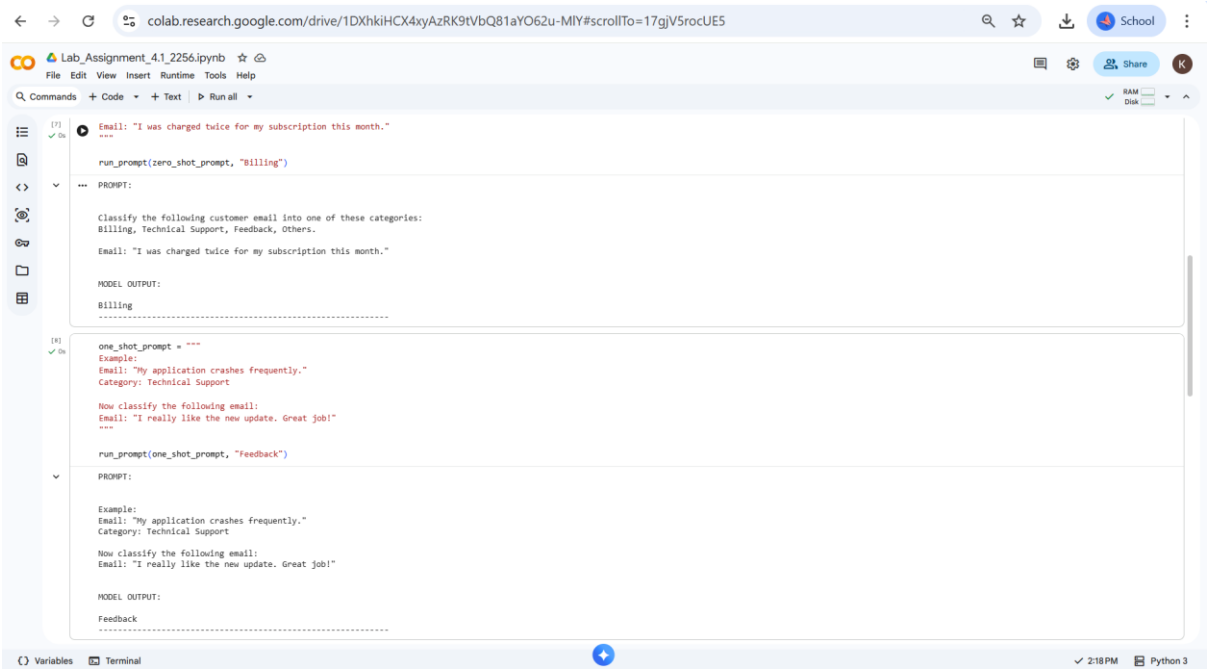
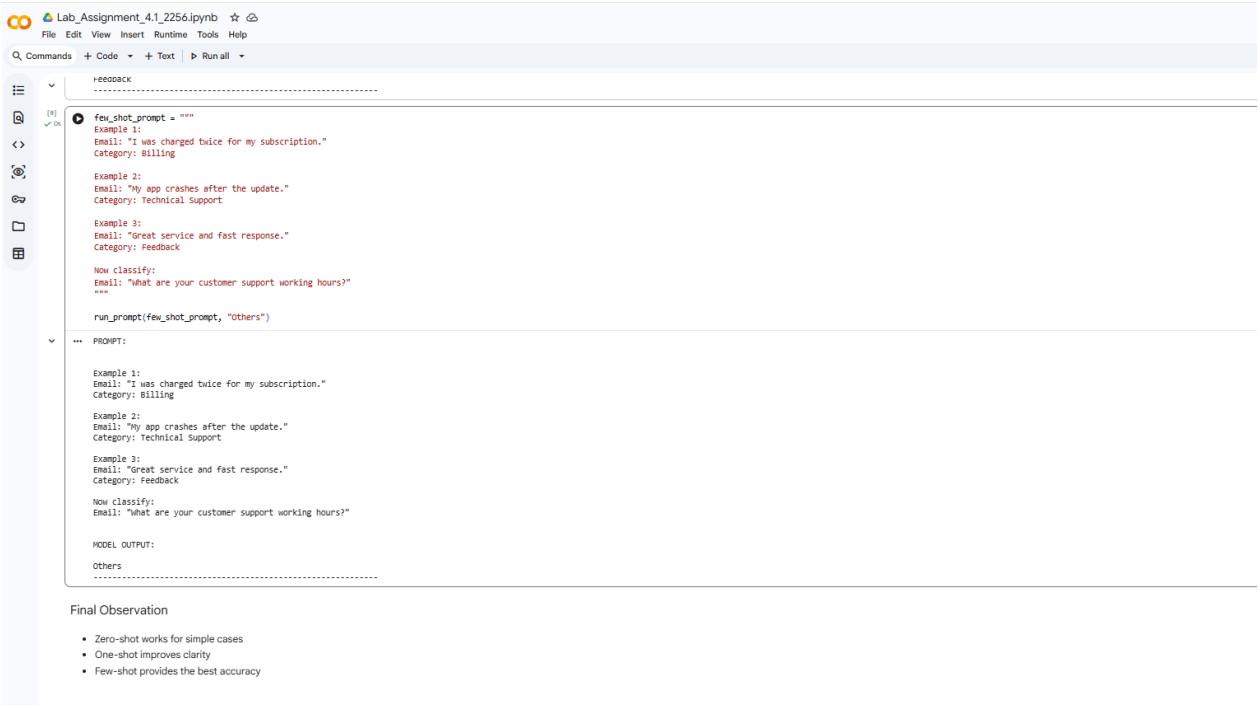
Lab Assignment- 4.1

AI Assisted Coding
Sunny Kolluri
2303A52444

Problem Statement 1: Customer Email Classification



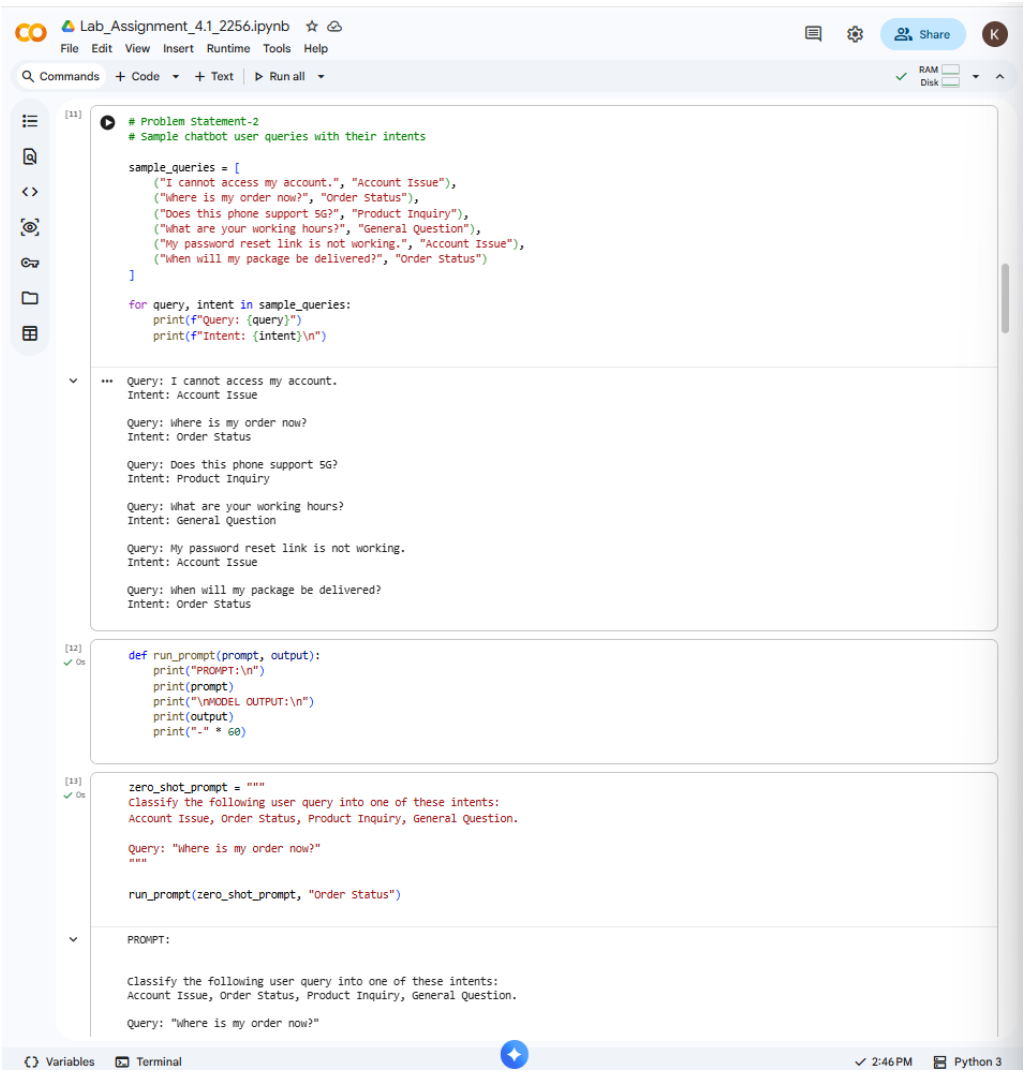


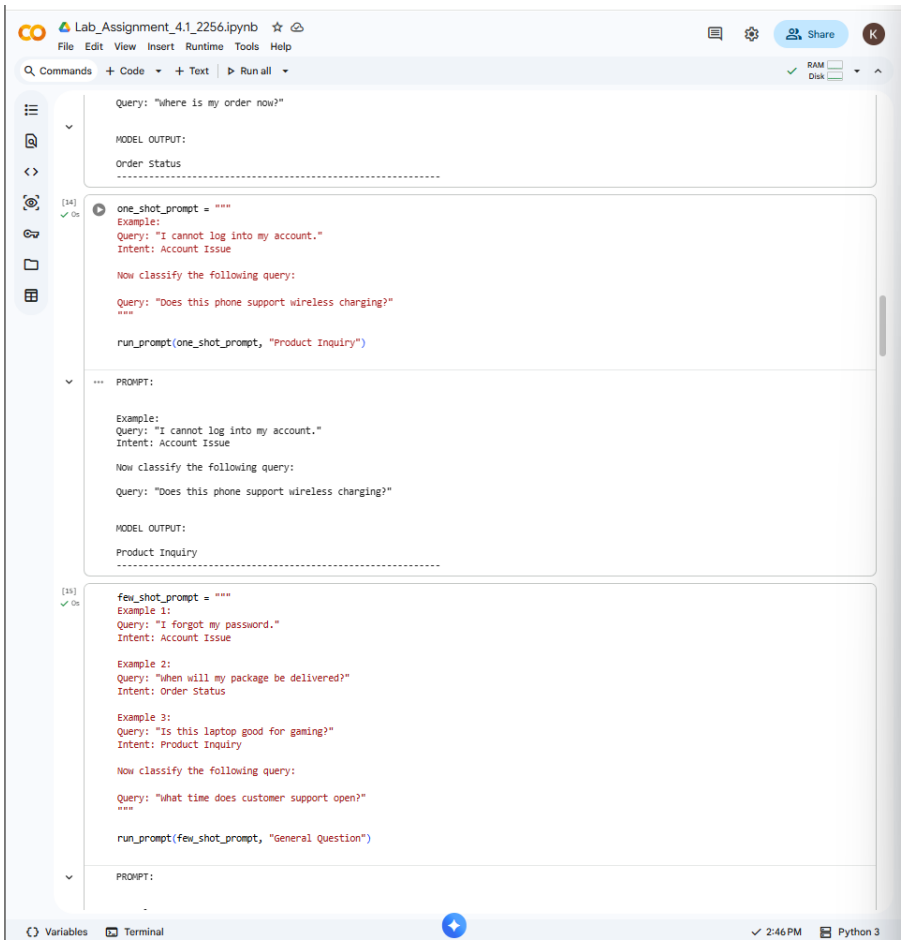


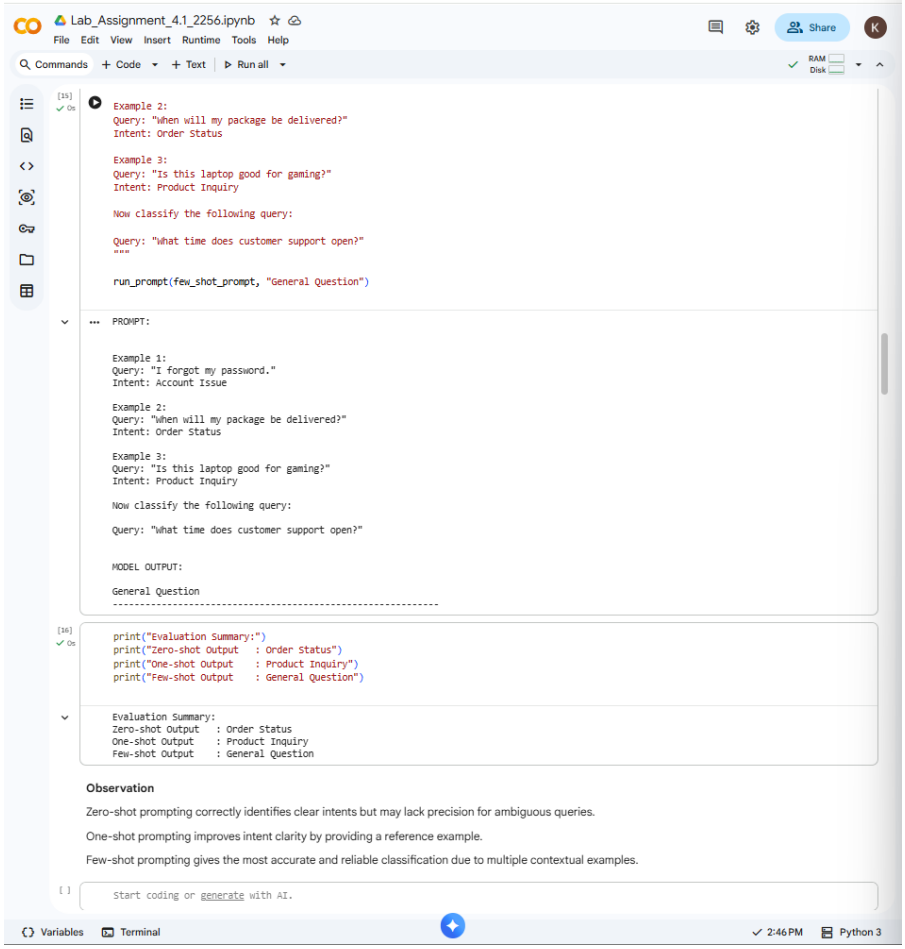
Final Observation

- Zero-shot works for simple cases
- One-shot improves clarity
- Few-shot provides the best accuracy

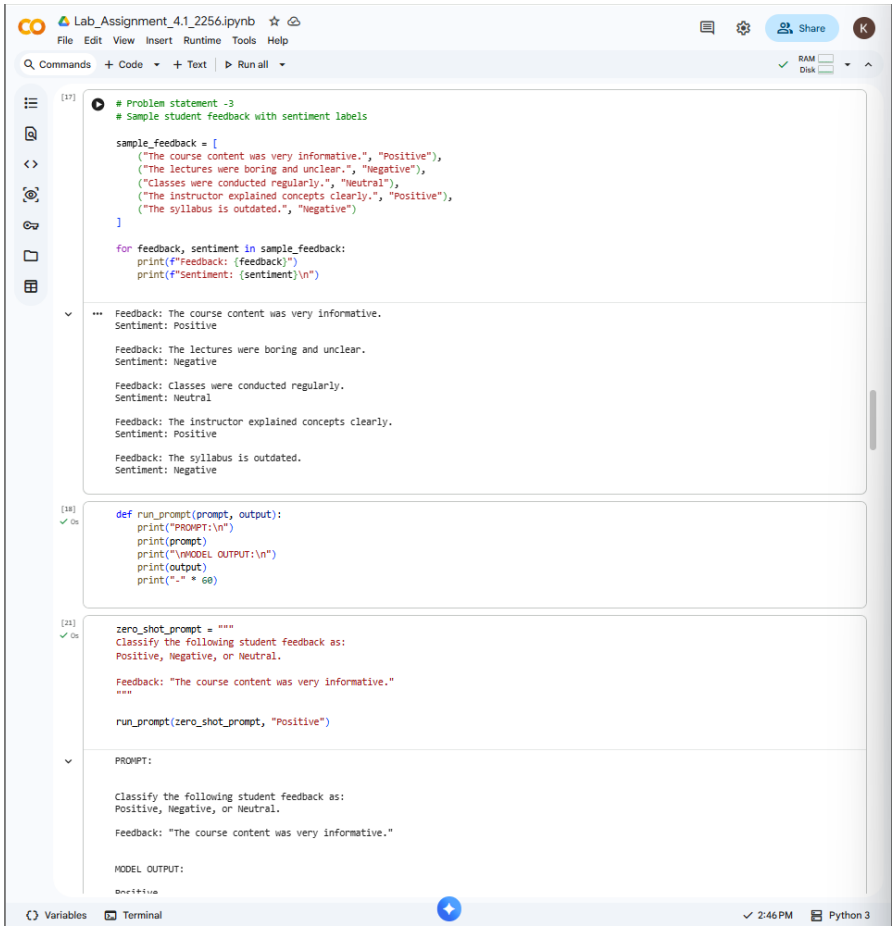
Problem Statement 2: Intent Classification for Chatbot Queries




Lab_Assignment_4.1_2256.ipynb
File Edit View Insert Runtime Tools Help
Share
Commands + Code + Text Run all
RAM Disk
Query: "Where is my order now?"
MODEL OUTPUT:
Order Status
[14]
one_shot_prompt = """
Example:
Query: "I cannot log into my account."
Intent: Account Issue
Now classify the following query:
Query: "Does this phone support wireless charging?"
"""
run_prompt(one_shot_prompt, "Product Inquiry")
PROMPT:
Example:
Query: "I cannot log into my account."
Intent: Account Issue
Now classify the following query:
Query: "Does this phone support wireless charging?"
MODEL OUTPUT:
Product Inquiry
[15]
few_shot_prompt = """
Example 1:
Query: "I forgot my password."
Intent: Account Issue
Example 2:
Query: "When will my package be delivered?"
Intent: Order Status
Example 3:
Query: "Is this laptop good for gaming?"
Intent: Product Inquiry
Now classify the following query:
Query: "What time does customer support open?"
"""
run_prompt(few_shot_prompt, "General Question")
PROMPT:
Variables Terminal
2:46 PM Python 3


Lab_Assignment_4.1_2256.ipynb
File Edit View Insert Runtime Tools Help
Share
Commands + Code + Text Run all
RAM Disk
[15]
Example 2:
Query: "When will my package be delivered?"
Intent: Order Status
Example 3:
Query: "Is this laptop good for gaming?"
Intent: Product Inquiry
Now classify the following query:
Query: "What time does customer support open?"
"""
run_prompt(few_shot_prompt, "General Question")
PROMPT:
Example 1:
Query: "I forgot my password."
Intent: Account Issue
Example 2:
Query: "When will my package be delivered?"
Intent: Order Status
Example 3:
Query: "Is this laptop good for gaming?"
Intent: Product Inquiry
Now classify the following query:
Query: "What time does customer support open?"
MODEL OUTPUT:
General Question
[16]
print("Evaluation Summary:")
print("Zero-shot Output : Order Status")
print("One-shot Output : Product Inquiry")
print("Few-shot Output : General Question")
Evaluation Summary:
Zero-shot Output : Order Status
One-shot Output : Product Inquiry
Few-shot Output : General Question
Observation
Zero-shot prompting correctly identifies clear intents but may lack precision for ambiguous queries.
One-shot prompting improves intent clarity by providing a reference example.
Few-shot prompting gives the most accurate and reliable classification due to multiple contextual examples.
[ ]
Start coding or generate with AI.
Variables Terminal
2:46 PM Python 3

Problem Statement 3: Student Feedback Analysis



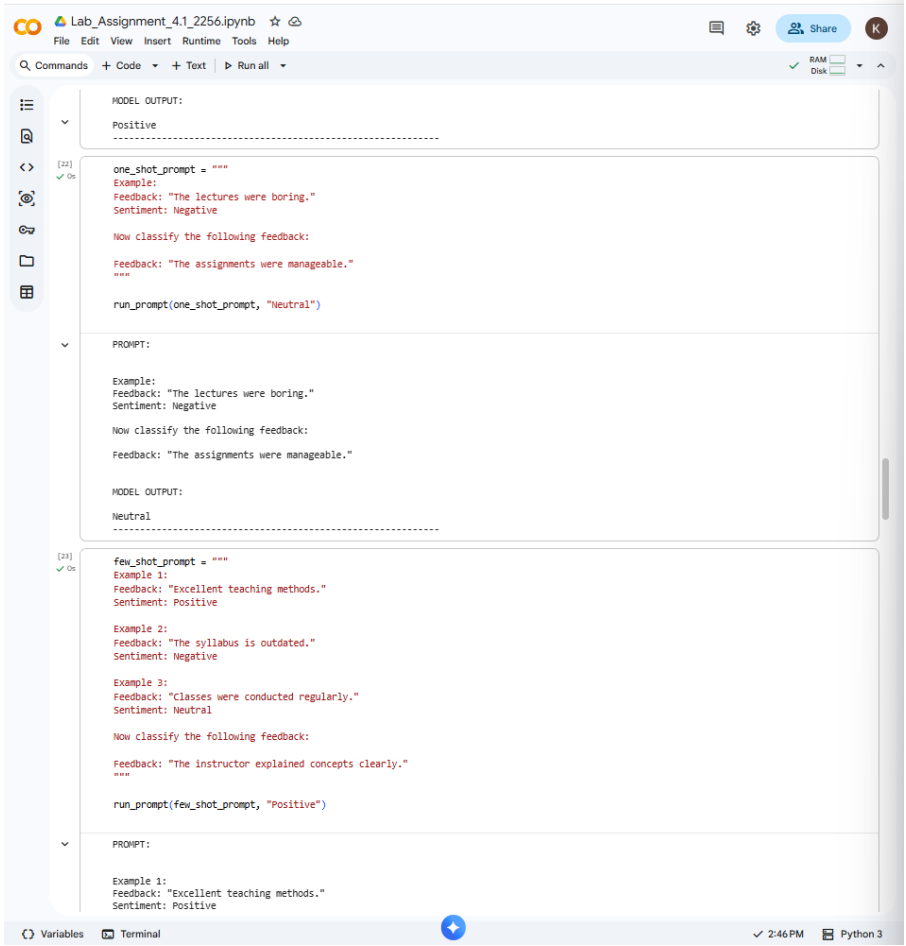
```
# Problem statement -3
# Sample student feedback with sentiment labels

sample_feedback = [
    ("The course content was very informative.", "Positive"),
    ("The lectures were boring and unclear.", "Negative"),
    ("Classes were conducted regularly.", "Neutral"),
    ("The instructor explained concepts clearly.", "Positive"),
    ("The syllabus is outdated.", "Negative")
]

for feedback, sentiment in sample_feedback:
    print(f"Feedback: {feedback}")
    print(f"Sentiment: {sentiment}\n")
```

```
Feedback: The course content was very informative.
Sentiment: Positive

Feedback: The lectures were boring and unclear.
Sentiment: Negative

Feedback: Classes were conducted regularly.
Sentiment: Neutral

Feedback: The instructor explained concepts clearly.
Sentiment: Positive

Feedback: The syllabus is outdated.
Sentiment: Negative
```

```
def run_prompt(prompt, output):
    print("PROMPT:\n")
    print(prompt)
    print("\nMODEL OUTPUT:\n")
    print(output)
    print("-" * 60)
```

```
zero_shot_prompt = """
Classify the following student feedback as:
Positive, Negative, or Neutral.

Feedback: "The course content was very informative."
"""

run_prompt(zero_shot_prompt, "Positive")
```

```
PROMPT:


Classify the following student feedback as:
Positive, Negative, or Neutral.

Feedback: "The course content was very informative."


MODEL OUTPUT:

Positive
```



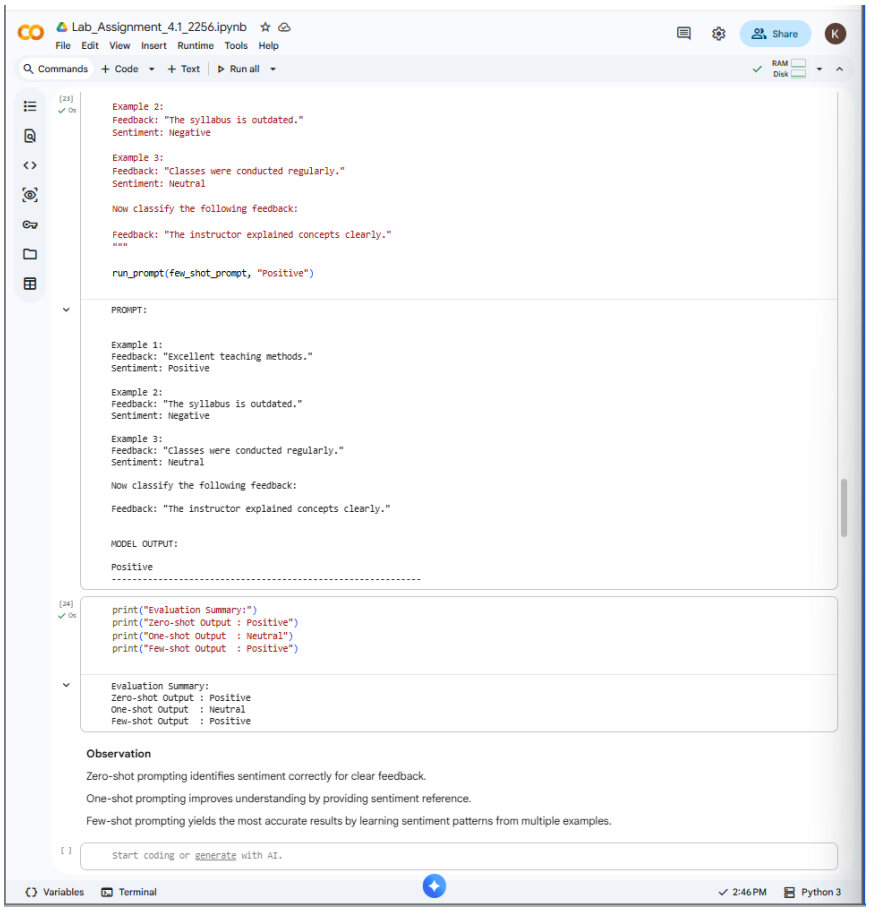
```
MODEL OUTPUT:

Positive
------------------------------------------------------------
```

```
one_shot_prompt = """
Example:
Feedback: "The lectures were boring."
Sentiment: Negative

Now classify the following feedback:

Feedback: "The assignments were manageable."
"""

run_prompt(one_shot_prompt, "Neutral")
```

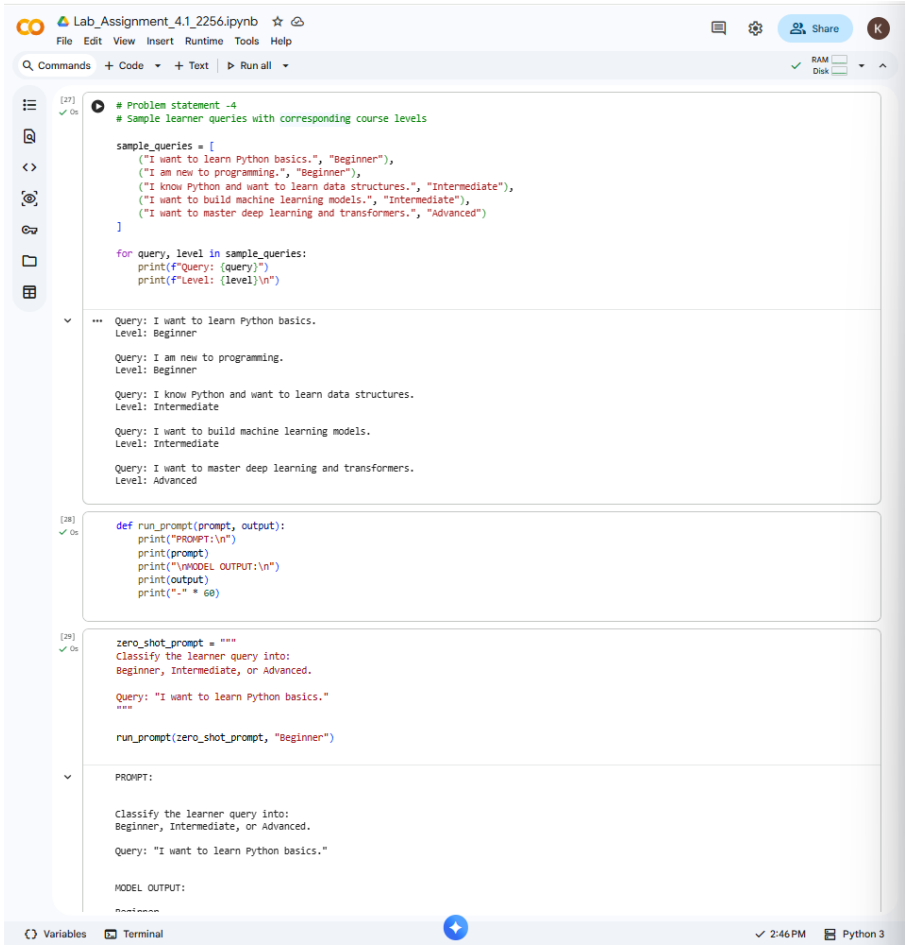
```
PROMPT:


Example:
Feedback: "The lectures were boring."
Sentiment: Negative

Now classify the following feedback:

Feedback: "The assignments were manageable."


MODEL OUTPUT:

Neutral
------------------------------------------------------------
```

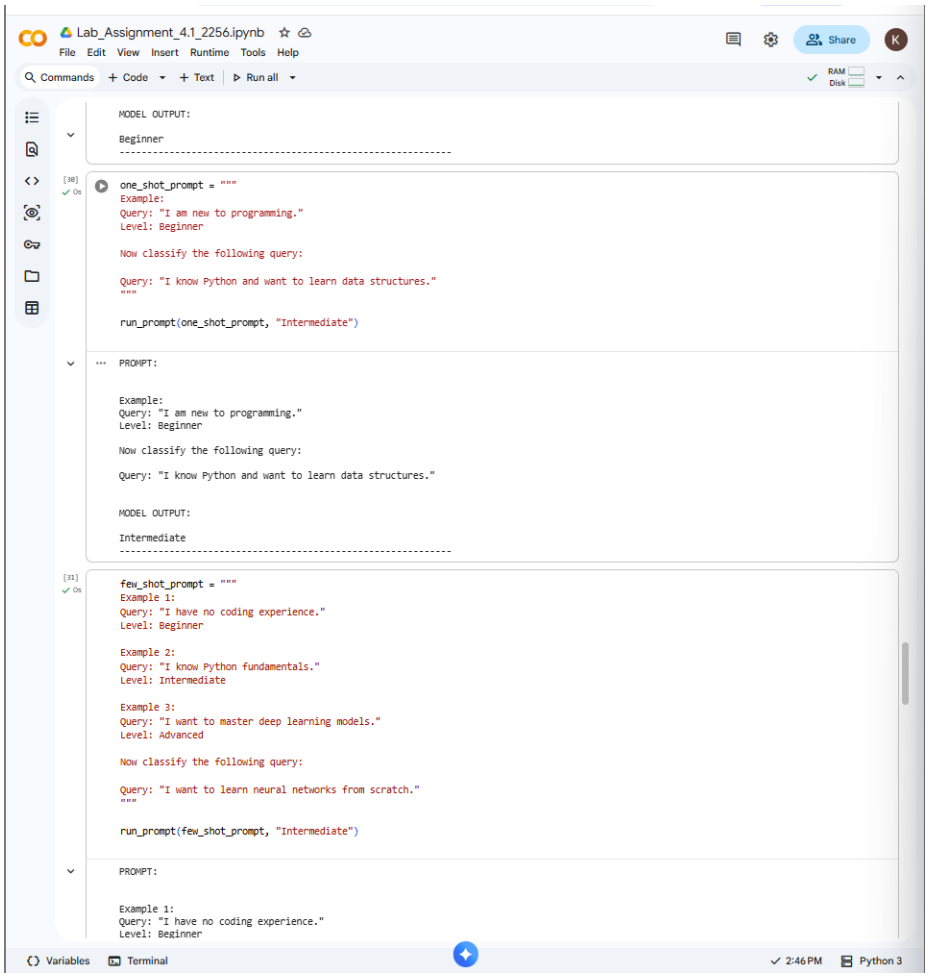
```
few_shot_prompt = """
Example 1:
Feedback: "Excellent teaching methods."
Sentiment: Positive

Example 2:
Feedback: "The syllabus is outdated."
Sentiment: Negative

Example 3:
Feedback: "Classes were conducted regularly."
Sentiment: Neutral

Now classify the following feedback:

Feedback: "The instructor explained concepts clearly."
"""

run_prompt(few_shot_prompt, "Positive")
```

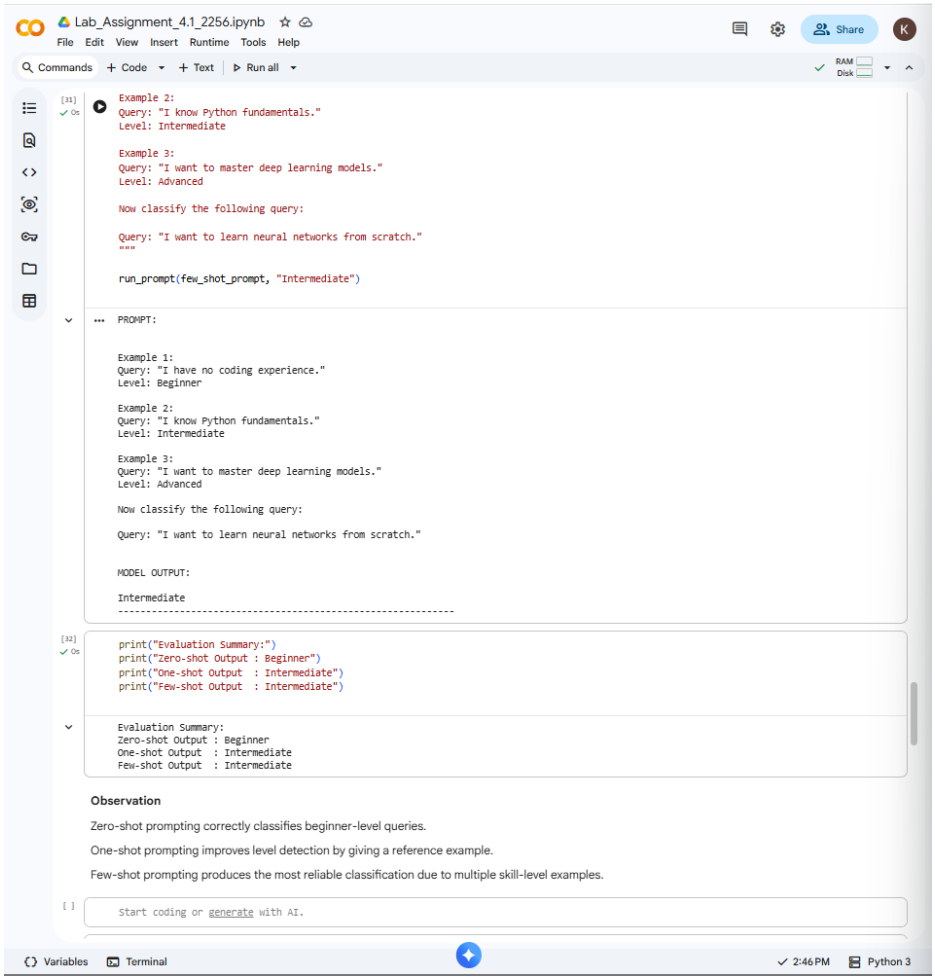
```
PROMPT:


Example 1:
Feedback: "Excellent teaching methods."
Sentiment: Positive
```



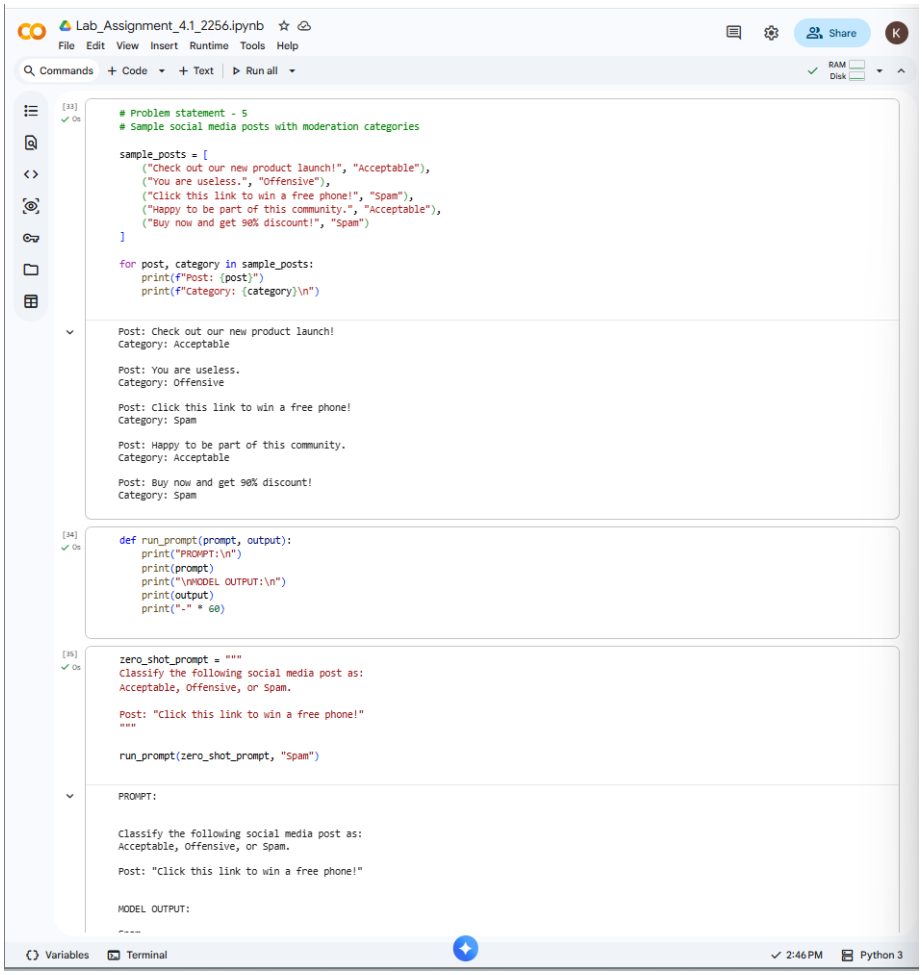
Problem Statement 4: Course Recommendation System




Lab_Assignment_4.1_2256.ipynb
File Edit View Insert Runtime Tools Help
Share
Commands + Code + Text Run all
RAM Disk
MODEL OUTPUT:
Beginner
[30]
one_shot_prompt = """
Example:
Query: "I am new to programming."
Level: Beginner
Now classify the following query:
Query: "I know Python and want to learn data structures."
"""
run_prompt(one_shot_prompt, "Intermediate")
PROMPT:
Example:
Query: "I am new to programming."
Level: Beginner
Now classify the following query:
Query: "I know Python and want to learn data structures."
MODEL OUTPUT:
Intermediate
[31]
few_shot_prompt = """
Example 1:
Query: "I have no coding experience."
Level: Beginner
Example 2:
Query: "I know Python fundamentals."
Level: Intermediate
Example 3:
Query: "I want to master deep learning models."
Level: Advanced
Now classify the following query:
Query: "I want to learn neural networks from scratch."
"""
run_prompt(few_shot_prompt, "Intermediate")
PROMPT:
Example 1:
Query: "I have no coding experience."
Level: Beginner
Variables Terminal
2:46 PM
Python 3


Lab_Assignment_4.1_2256.ipynb
File Edit View Insert Runtime Tools Help
Share
Commands + Code + Text Run all
RAM Disk
[31]
Example 2:
Query: "I know Python fundamentals."
Level: Intermediate
Example 3:
Query: "I want to master deep learning models."
Level: Advanced
Now classify the following query:
Query: "I want to learn neural networks from scratch."
"""
run_prompt(few_shot_prompt, "Intermediate")
PROMPT:
Example 1:
Query: "I have no coding experience."
Level: Beginner
Example 2:
Query: "I know Python fundamentals."
Level: Intermediate
Example 3:
Query: "I want to master deep learning models."
Level: Advanced
Now classify the following query:
Query: "I want to learn neural networks from scratch."
MODEL OUTPUT:
Intermediate
[32]
print("Evaluation Summary:")
print("Zero-shot Output : Beginner")
print("One-shot Output : Intermediate")
print("Few-shot Output : Intermediate")
Evaluation Summary:
Zero-shot Output : Beginner
One-shot Output : Intermediate
Few-shot Output : Intermediate
Observation
Zero-shot prompting correctly classifies beginner-level queries.
One-shot prompting improves level detection by giving a reference example.
Few-shot prompting produces the most reliable classification due to multiple skill-level examples.
[ ]
Start coding or generate with AI.
Variables Terminal
2:46 PM
Python 3

Problem Statement 5: Social Media Post Moderation



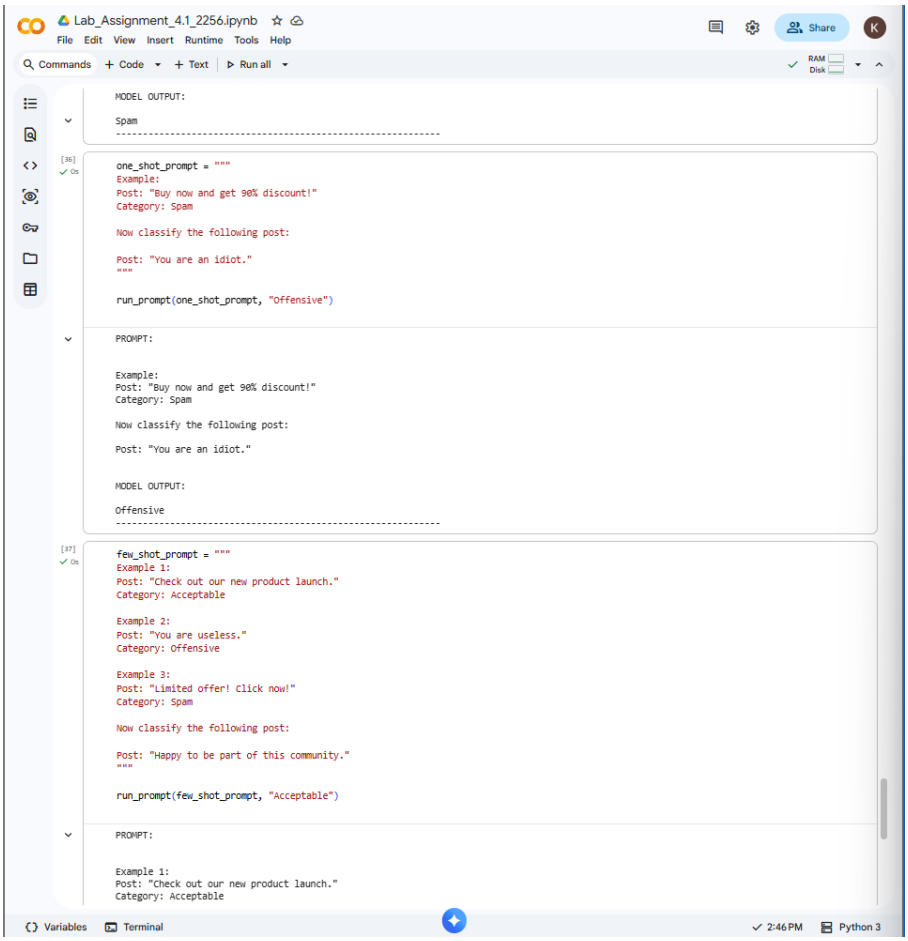
```
# Problem statement - 5
# Sample social media posts with moderation categories

sample_posts = [
    ("Check out our new product launch!", "Acceptable"),
    ("You are useless.", "Offensive"),
    ("Click this link to win a free phone!", "Spam"),
    ("Happy to be part of this community.", "Acceptable"),
    ("Buy now and get 90% discount!", "Spam")
]

for post, category in sample_posts:
    print(f"Post: {post}")
    print(f"Category: {category}\n")
```

```
Post: Check out our new product launch!
Category: Acceptable

Post: You are useless.
Category: Offensive

Post: Click this link to win a free phone!
Category: Spam

Post: Happy to be part of this community.
Category: Acceptable

Post: Buy now and get 90% discount!
Category: Spam
```

```
def run_prompt(prompt, output):
    print("PROMPT:\n")
    print(prompt)
    print("\nMODEL OUTPUT:\n")
    print(output)
    print("-" * 60)
```

```
zero_shot_prompt = """
Classify the following social media post as:
Acceptable, Offensive, or Spam.

Post: "Click this link to win a free phone!"
"""

run_prompt(zero_shot_prompt, "Spam")
```

```
PROMPT:


Classify the following social media post as:
Acceptable, Offensive, or Spam.

Post: "Click this link to win a free phone!"


MODEL OUTPUT:

----
```



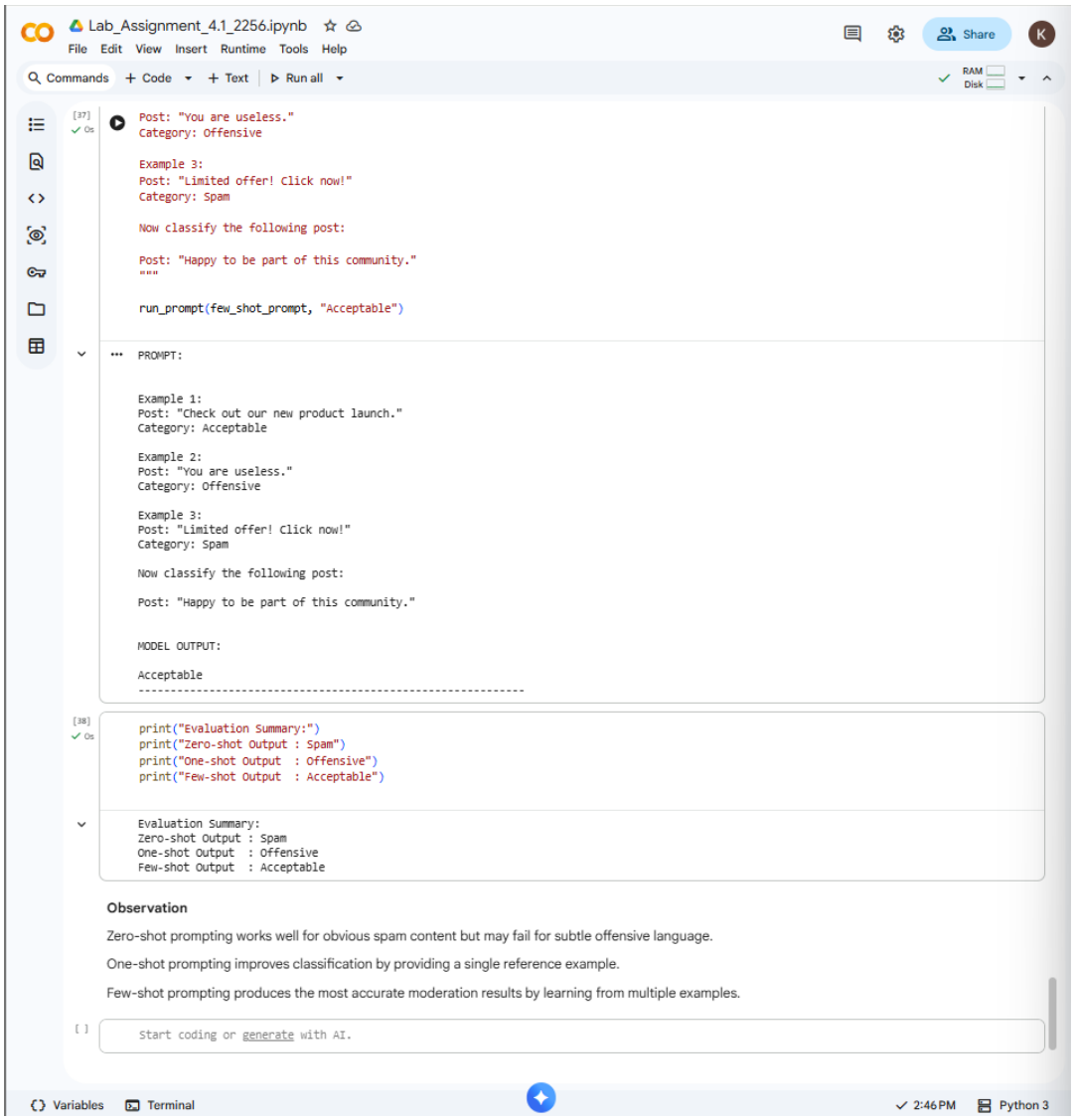
```
MODEL OUTPUT:

Spam
------------------------------------------------------------
```

```
one_shot_prompt = """
Example:
Post: "Buy now and get 90% discount!"
Category: Spam

Now classify the following post:

Post: "You are an idiot."
"""

run_prompt(one_shot_prompt, "Offensive")
```

```
PROMPT:


Example:
Post: "Buy now and get 90% discount!"
Category: Spam

Now classify the following post:

Post: "You are an idiot."


MODEL OUTPUT:

Offensive
------------------------------------------------------------
```

```
few_shot_prompt = """
Example 1:
Post: "Check out our new product launch."
Category: Acceptable

Example 2:
Post: "You are useless."
Category: Offensive

Example 3:
Post: "Limited offer! Click now!"
Category: Spam

Now classify the following post:

Post: "Happy to be part of this community."
"""

run_prompt(few_shot_prompt, "Acceptable")
```

```
PROMPT:


Example 1:
Post: "Check out our new product launch."
Category: Acceptable
```


Lab_Assignment_4.1_2256.ipynb
File Edit View Insert Runtime Tools Help
Share
Commands + Code + Text Run all
RAM Disk
[37]
Post: "You are useless."
Category: Offensive
Example 3:
Post: "Limited offer! Click now!"
Category: Spam
Now classify the following post:
Post: "Happy to be part of this community."
"""
run_prompt(few_shot_prompt, "Acceptable")
PROMPT:
Example 1:
Post: "Check out our new product launch."
Category: Acceptable
Example 2:
Post: "You are useless."
Category: Offensive
Example 3:
Post: "Limited offer! Click now!"
Category: Spam
Now classify the following post:
Post: "Happy to be part of this community."
MODEL OUTPUT:
Acceptable
[38]
print("Evaluation Summary:")
print("Zero-shot Output : Spam")
print("One-shot Output : Offensive")
print("Few-shot Output : Acceptable")
Evaluation Summary:
Zero-shot Output : Spam
One-shot Output : Offensive
Few-shot Output : Acceptable
Observation
Zero-shot prompting works well for obvious spam content but may fail for subtle offensive language.
One-shot prompting improves classification by providing a single reference example.
Few-shot prompting produces the most accurate moderation results by learning from multiple examples.
[ ]
Start coding or generate with AI.
Variables Terminal
2:46 PM Python 3