Lab Assignment- 4.1

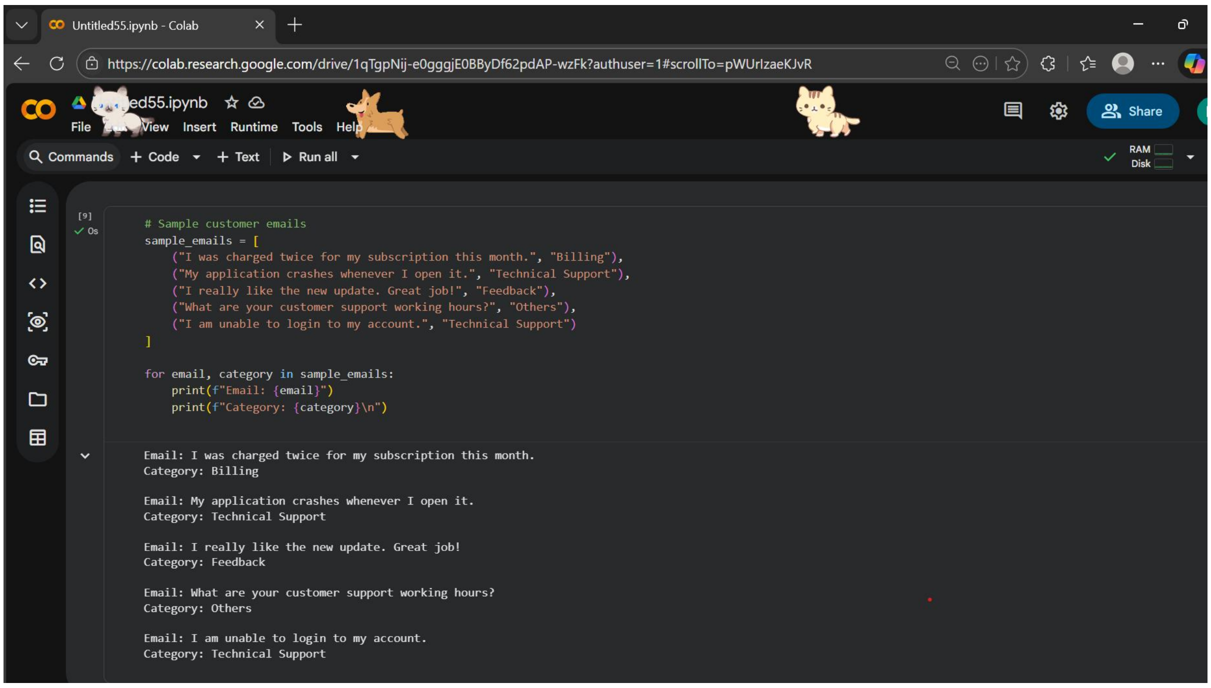
Lab 4: Advanced Prompt Engineering – Zero-shot, One-shot, and Few-shot Techniques

AI Assisted Coding

Name:-Pendli Likitha

HT NO:- 2303A52393

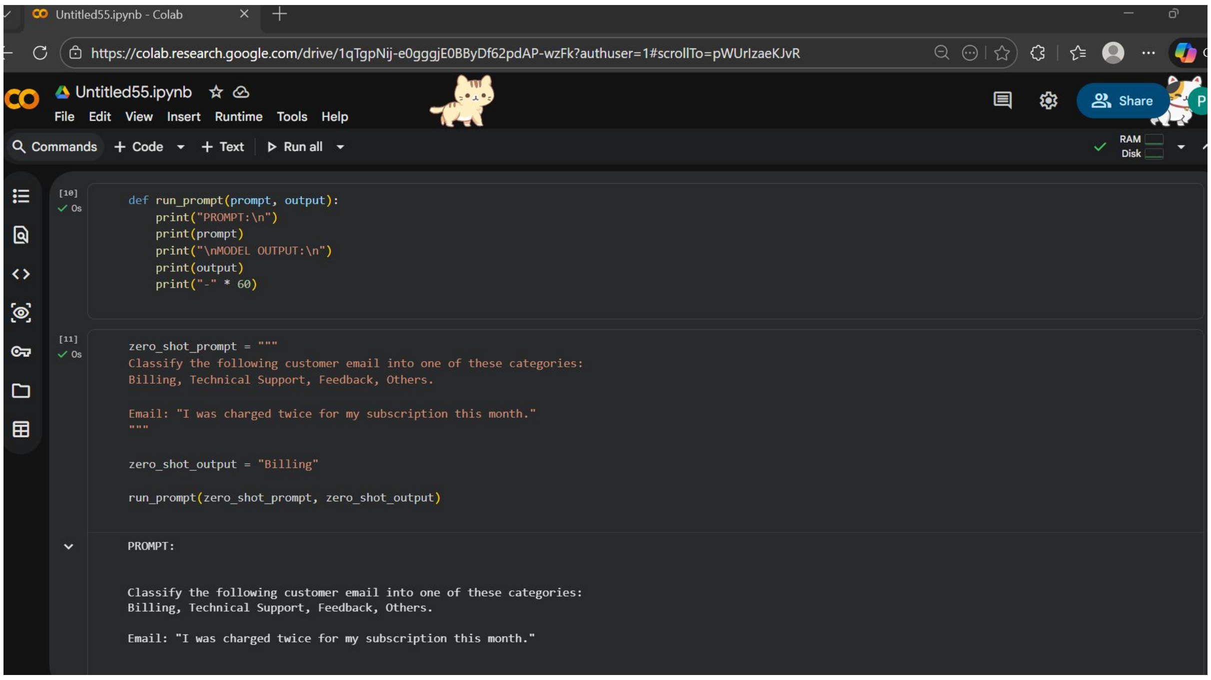
Problem Statement 1: Customer Email Classification



Untitled55.ipynb - Colab
https://colab.research.google.com/drive/1qTgpNij-e0gggjE0BByDf62pdAP-wzFk?authuser=1#scrollTo=pWUrlzaeKJvR
File View Insert Runtime Tools Help
Share
Commands + Code + Text Run all
RAM Disk

```python
# Sample customer emails
sample_emails = [
    ("I was charged twice for my subscription this month.", "Billing"),
    ("My application crashes whenever I open it.", "Technical Support"),
    ("I really like the new update. Great job!", "Feedback"),
    ("What are your customer support working hours?", "Others"),
    ("I am unable to login to my account.", "Technical Support")
]

for email, category in sample_emails:
    print(f"Email: {email}")
    print(f"Category: {category}\n")
```

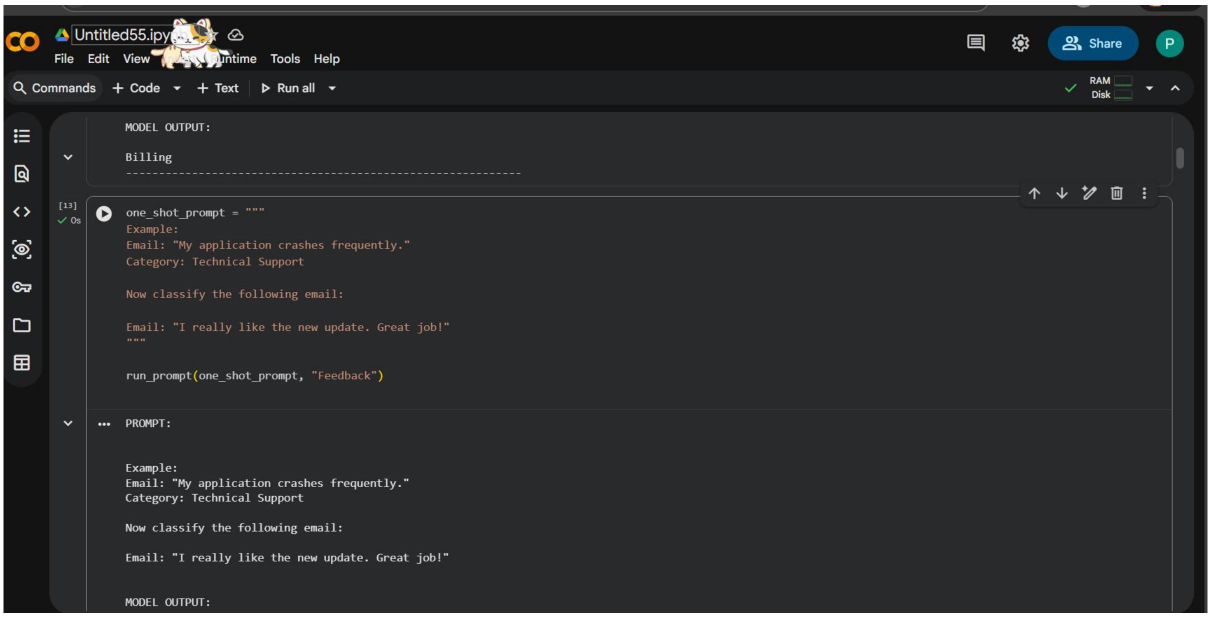
```
Email: I was charged twice for my subscription this month.
Category: Billing

Email: My application crashes whenever I open it.
Category: Technical Support

Email: I really like the new update. Great job!
Category: Feedback

Email: What are your customer support working hours?
Category: Others

Email: I am unable to login to my account.
Category: Technical Support
```


Untitled55.ipynb - Colab
https://colab.research.google.com/drive/1qTgpNij-e0gggjE0BByDf62pdAP-wzFk?authuser=1#scrollTo=pWUrlzaeKJvR
Untitled55.ipynb
File Edit View Insert Runtime Tools Help
Share
Commands + Code + Text Run all
RAM
Disk
[10]
0s
[11]
0s

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

```
zero_shot_prompt = """
Classify the following customer email into one of these categories:
Billing, Technical Support, Feedback, Others.

Email: "I was charged twice for my subscription this month."
"""

zero_shot_output = "Billing"

run_prompt(zero_shot_prompt, zero_shot_output)
```

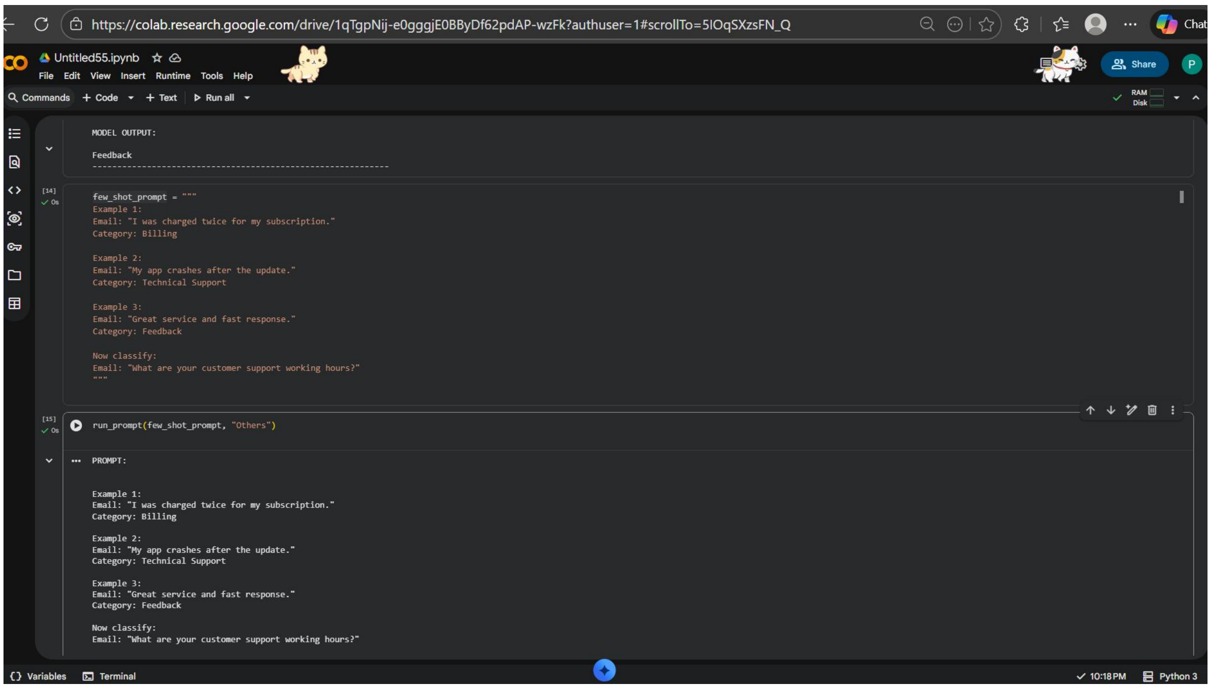
```
PROMPT:


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

Email: "I was charged twice for my subscription this month."
```


Untitled55.ipy
File Edit View Runtime Tools Help
Share
Commands + Code + Text Run all
RAM
Disk
[13]
0s

```
MODEL OUTPUT:

Billing
------------------------------------------------------------
```

```
one_shot_prompt = """
Example:
Email: "My application crashes frequently."
Category: Technical Support

Now classify the following email:

Email: "I really like the new update. Great job!"
"""

run_prompt(one_shot_prompt, "Feedback")
```

```
PROMPT:


Example:
Email: "My application crashes frequently."
Category: Technical Support

Now classify the following email:

Email: "I really like the new update. Great job!"

MODEL OUTPUT:
```

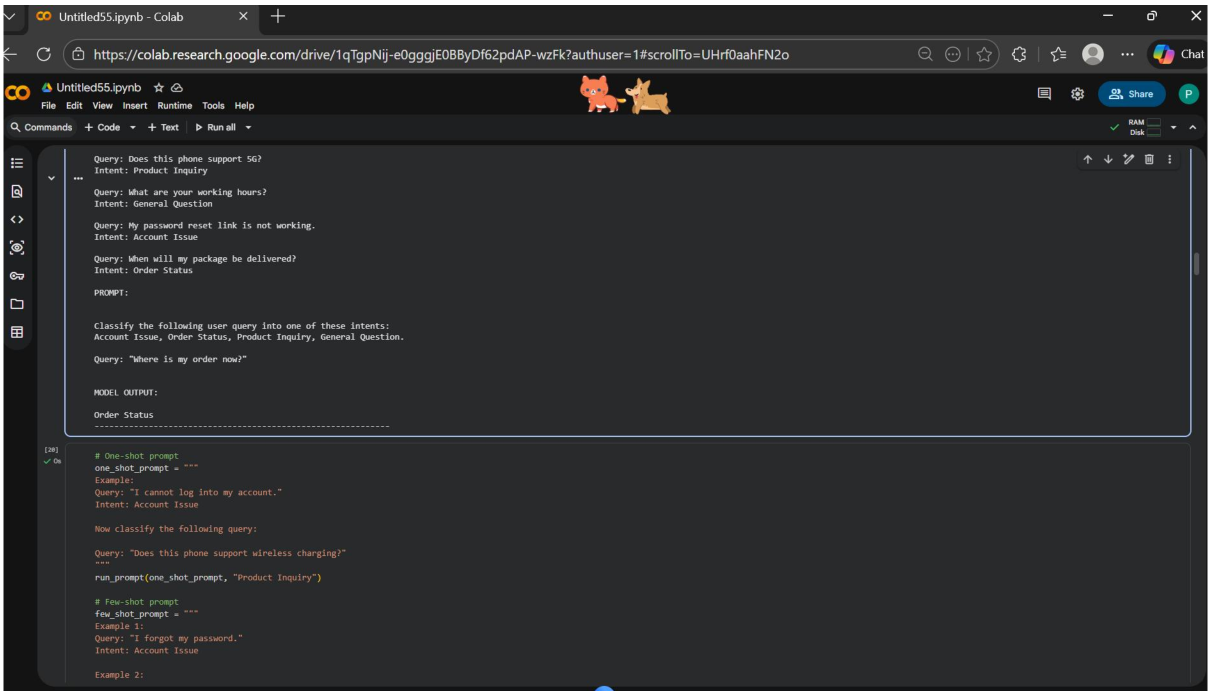

```
MODEL OUTPUT:

Feedback
------------------------------------------------------------
```

```
[14]
few_shot_prompt = """
Example 1:
Email: "I was charged twice for my subscription."
Category: Billing

Example 2:
Email: "My app crashes after the update."
Category: Technical Support

Example 3:
Email: "Great service and fast response."
Category: Feedback

Now classify:
Email: "What are your customer support working hours?"
"""
```

```
[15]
run_prompt(few_shot_prompt, "Others")
```

```
PROMPT:


Example 1:
Email: "I was charged twice for my subscription."
Category: Billing

Example 2:
Email: "My app crashes after the update."
Category: Technical Support

Example 3:
Email: "Great service and fast response."
Category: Feedback

Now classify:
Email: "What are your customer support working hours?"
```

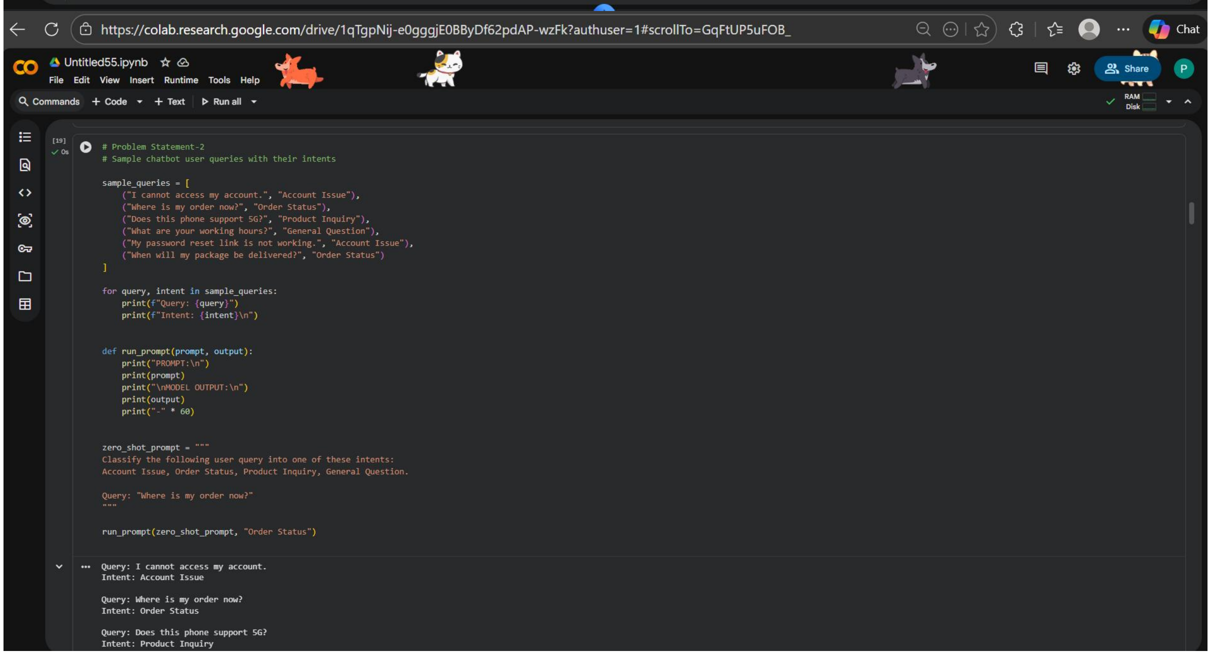
Problem Statement 2: Intent Classification for Chatbot Queries


Untitled55.ipynb - Colab
https://colab.research.google.com/drive/1qTgpNij-e0gggjE0BByDf62pdAP-wzFk?authuser=1#scrollTo=UHrf0aahFN2o
Untitled55.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all
Share
RAM Disk

```
Query: Does this phone support 5G?
Intent: Product Inquiry

Query: What are your working hours?
Intent: General Question

Query: My password reset link is not working.
Intent: Account Issue

Query: When will my package be delivered?
Intent: Order Status

PROMPT:


Classify the following user query into one of these intents:
Account Issue, Order Status, Product Inquiry, General Question.

Query: "Where is my order now?"


MODEL OUTPUT:

Order Status
------------------------------------------------------------
```

[20]

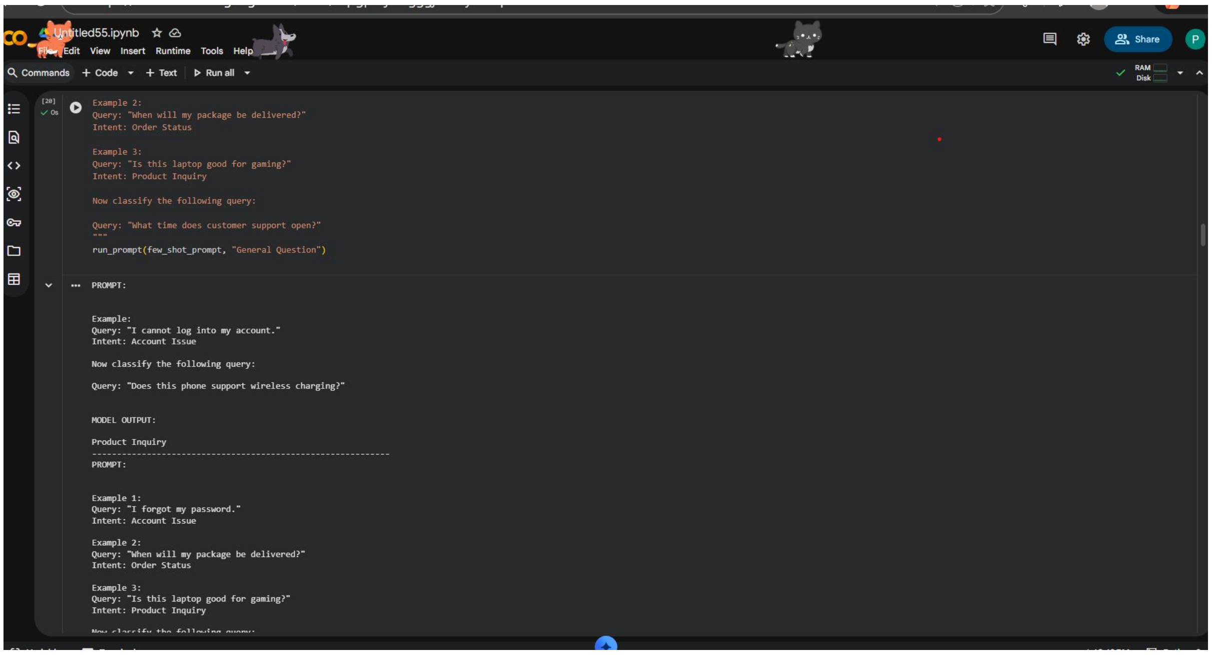
```python
# One-shot prompt
one_shot_prompt = """
Example:
Query: "I cannot log into my account."
Intent: Account Issue

Now classify the following query:

Query: "Does this phone support wireless charging?"
"""
run_prompt(one_shot_prompt, "Product Inquiry")

# Few-shot prompt
few_shot_prompt = """
Example 1:
Query: "I forgot my password."
Intent: Account Issue

Example 2:
```


https://colab.research.google.com/drive/1qTgpNij-e0gggjE0BByDf62pdAP-wzFk?authuser=1#scrollTo=GqFtUP5uFOB_
Untitled55.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all
Share
RAM Disk

[19]

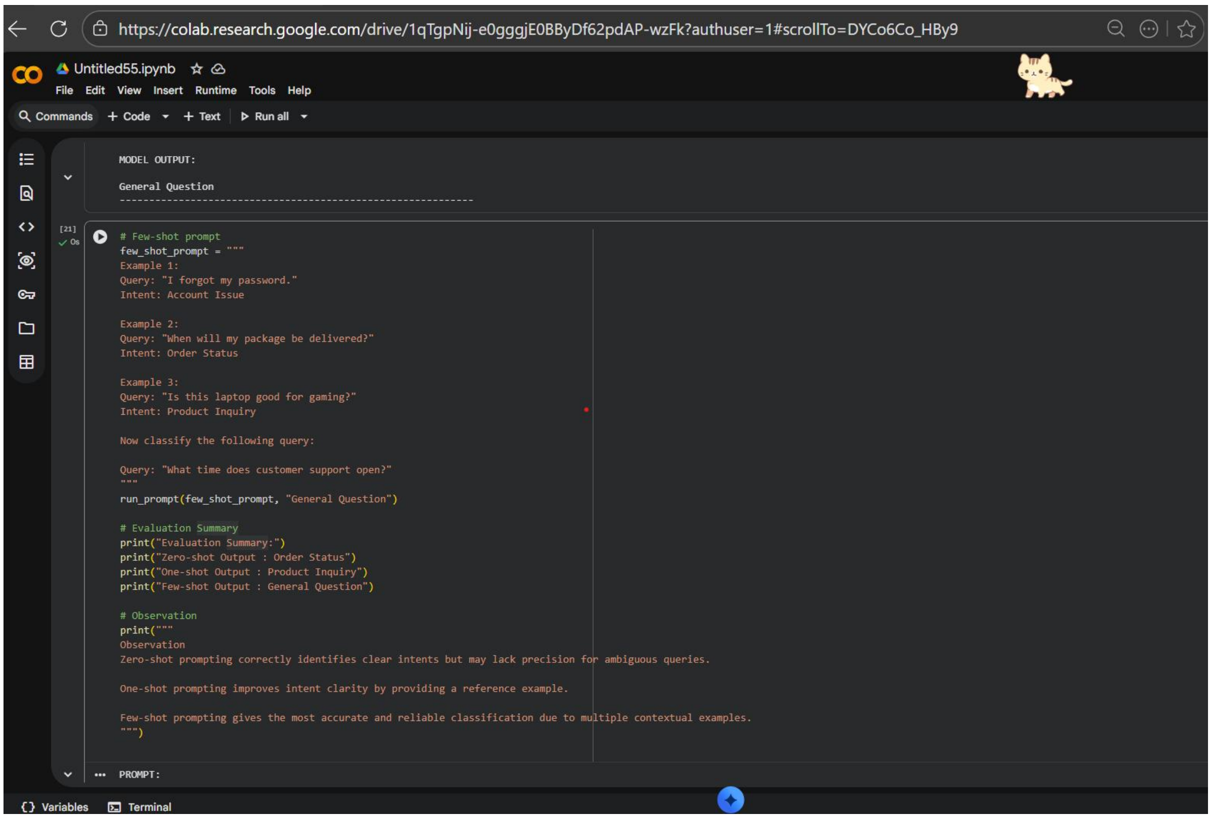
```python
# Problem Statement-2
# Sample chatbot user queries with their intents

sample_queries = [
    ("I cannot access my account.", "Account Issue"),
    ("Where is my order now?", "Order Status"),
    ("Does this phone support 5G?", "Product Inquiry"),
    ("What are your working hours?", "General Question"),
    ("My password reset link is not working.", "Account Issue"),
    ("When will my package be delivered?", "Order Status")
]

for query, intent in sample_queries:
    print(f"Query: {query}")
    print(f"Intent: {intent}\n")


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


zero_shot_prompt = """
Classify the following user query into one of these intents:
Account Issue, Order Status, Product Inquiry, General Question.

Query: "Where is my order now?"
"""

run_prompt(zero_shot_prompt, "Order Status")
```

```
Query: I cannot access my account.
Intent: Account Issue

Query: Where is my order now?
Intent: Order Status

Query: Does this phone support 5G?
Intent: Product Inquiry
```


Untitled55.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all
Share
RAM Disk

```
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:
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
------------------------------------------------------------
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
```

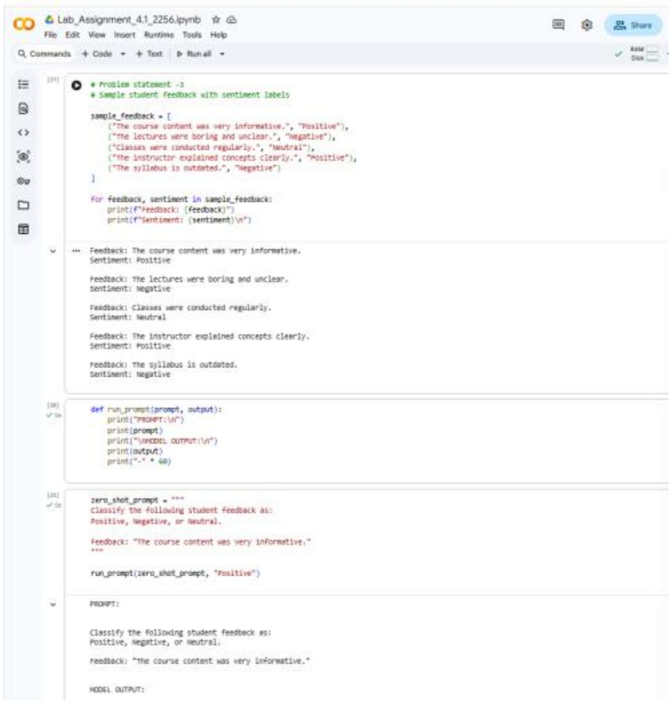

https://colab.research.google.com/drive/1qTgpNij-e0gggjE0BByDf62pdAP-wzFk?authuser=1#scrollTo=DYCo6Co_HBy9
Untitled55.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all
Variables Terminal

```
MODEL OUTPUT:

General Question
------------------------------------------------------------
```

```
# Few-shot prompt
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")

# Evaluation Summary
print("Evaluation Summary:")
print("Zero-shot Output : Order Status")
print("One-shot Output : Product Inquiry")
print("Few-shot Output : General Question")

# Observation
print("""
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.
""")
```

```
PROMPT:
```

Problem Statement 3: Student Feedback Analysis


Lab_Assignment_4.1_2256.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all
Share

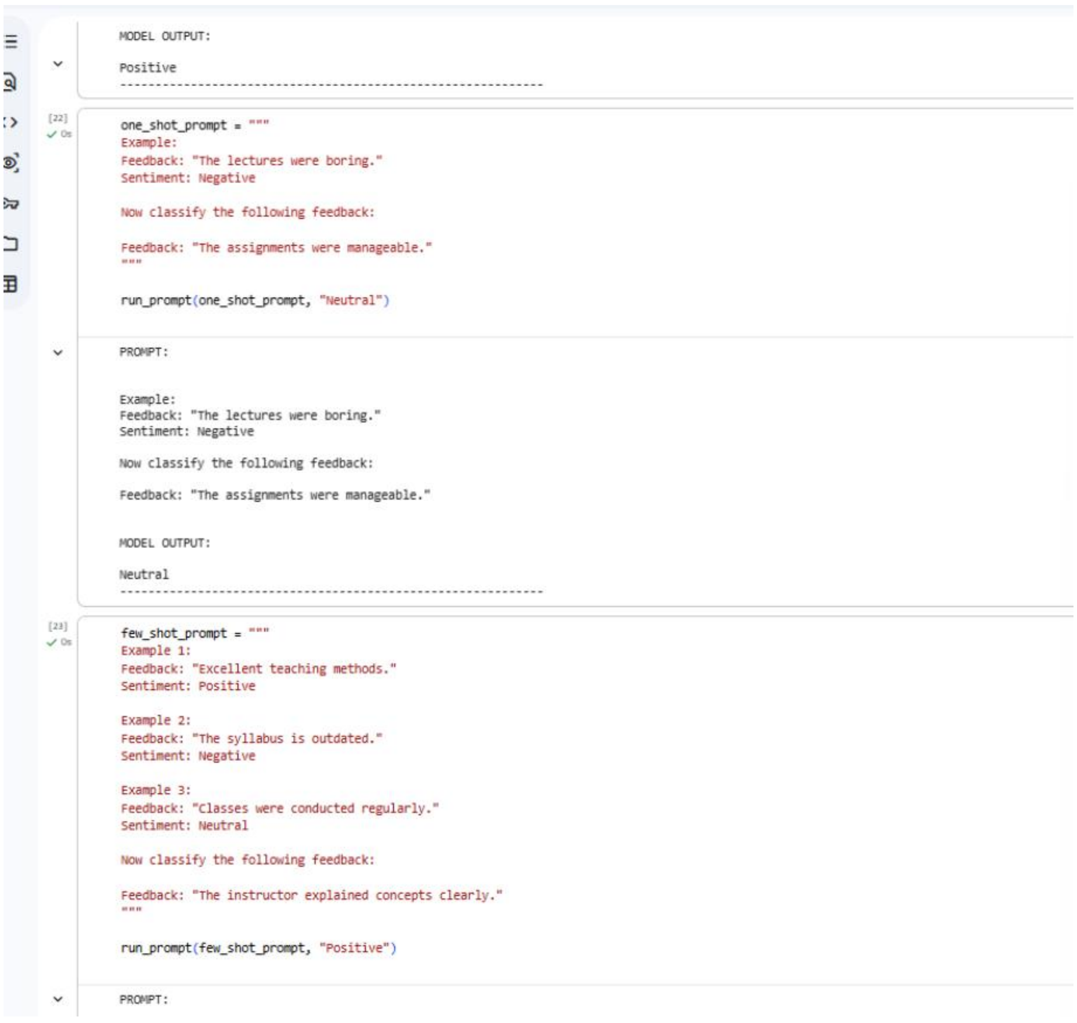
```python
# Problem statement -3
# Sample student feedback with sentiment labels

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
```

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

```python
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:
```



```
MODEL OUTPUT:

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

[22]

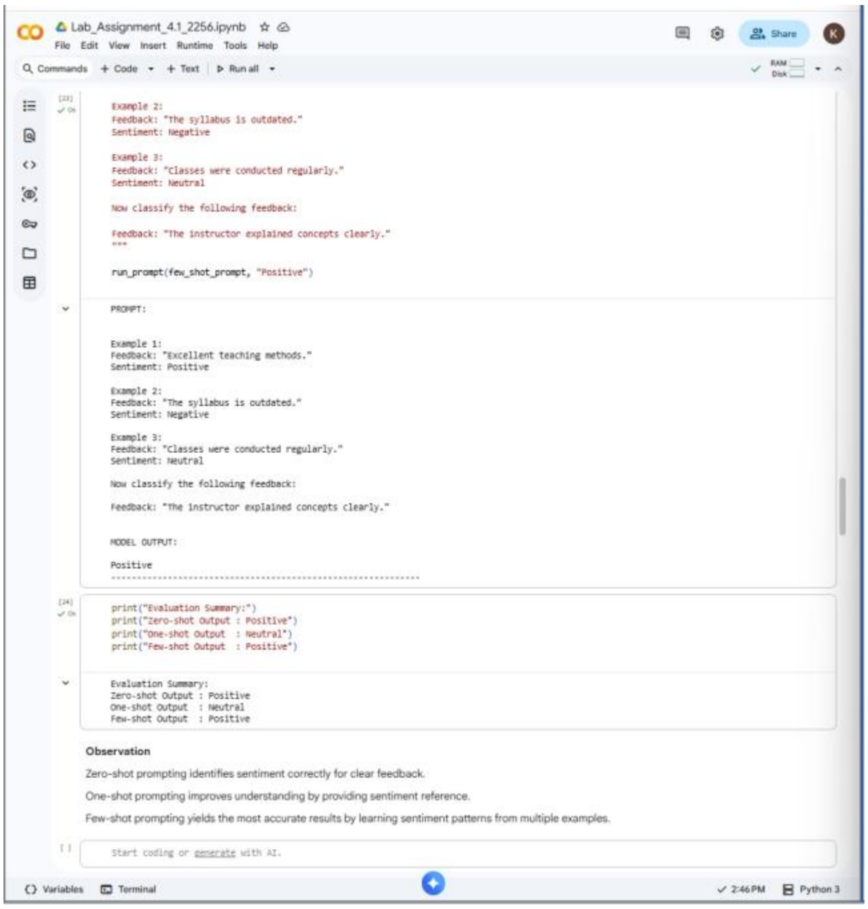
```
one_shot_prompt = """
Example:
Feedback: "The lectures were boring."
Sentiment: Negative

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
------------------------------------------------------------
```

[23]

```
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:
```


Lab_Assignment_4.1_2256.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all
Share
RAM Disk
Variables Terminal
2:46 PM Python 3

```
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

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."


MODEL OUTPUT:

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

```
print("Evaluation Summary:")
print("Zero-shot Output : Positive")
print("One-shot Output  : Neutral")
print("Few-shot Output  : Positive")
```

```
Evaluation Summary:
Zero-shot Output : Positive
One-shot Output  : Neutral
Few-shot Output  : Positive
```

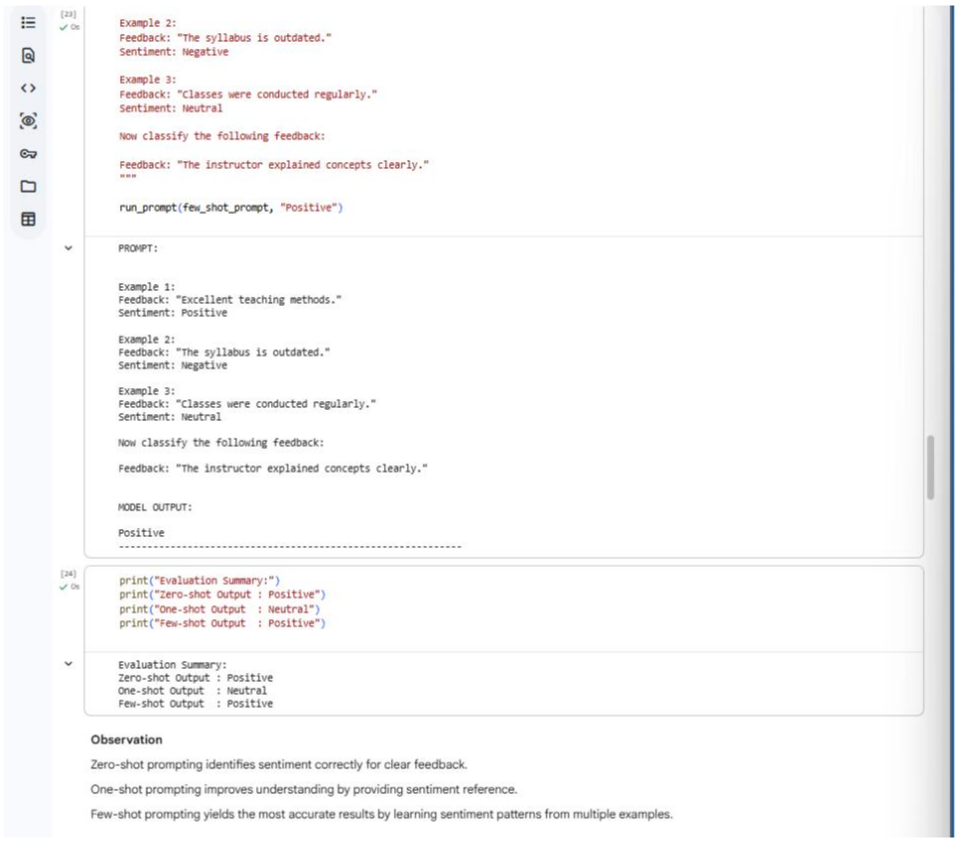
Observation

Zero-shot prompting identifies sentiment correctly for clear feedback.

One-shot prompting improves understanding by providing sentiment reference.

Few-shot prompting yields the most accurate results by learning sentiment patterns from multiple examples.

```
Start coding or generate with AI.
```



```
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

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."


MODEL OUTPUT:

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

```
print("Evaluation Summary:")
print("Zero-shot Output : Positive")
print("One-shot Output  : Neutral")
print("Few-shot Output  : Positive")
```

```
Evaluation Summary:
Zero-shot Output : Positive
One-shot Output  : Neutral
Few-shot Output  : Positive
```

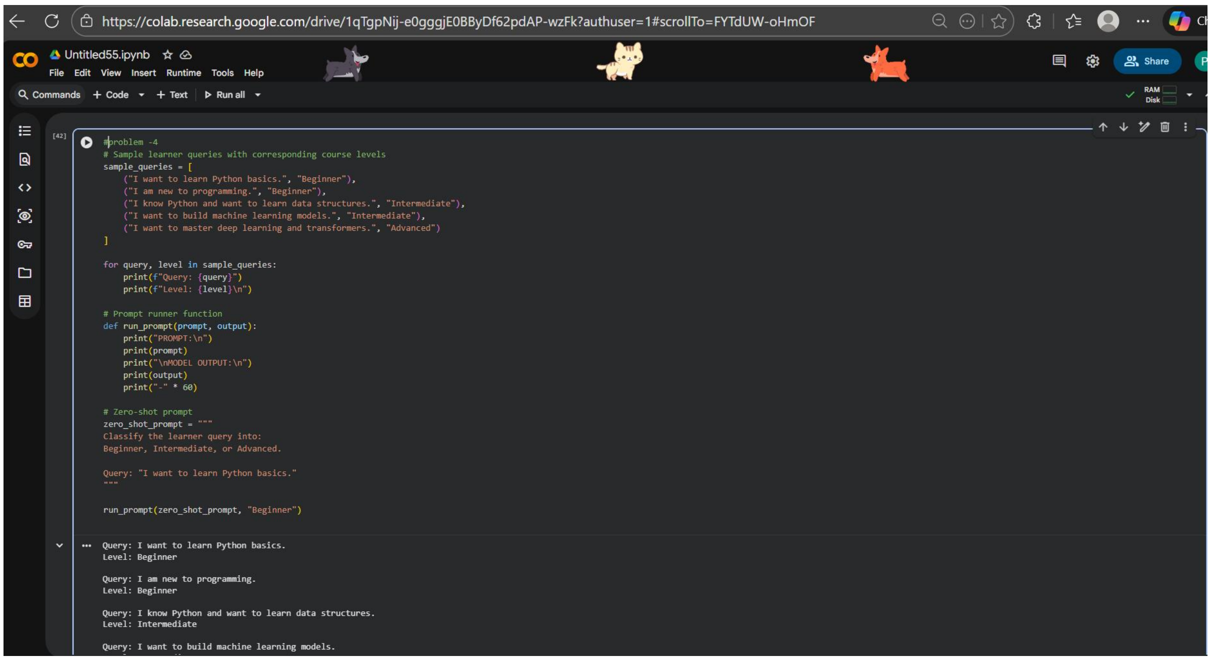
Observation

Zero-shot prompting identifies sentiment correctly for clear feedback.

One-shot prompting improves understanding by providing sentiment reference.

Few-shot prompting yields the most accurate results by learning sentiment patterns from multiple examples.

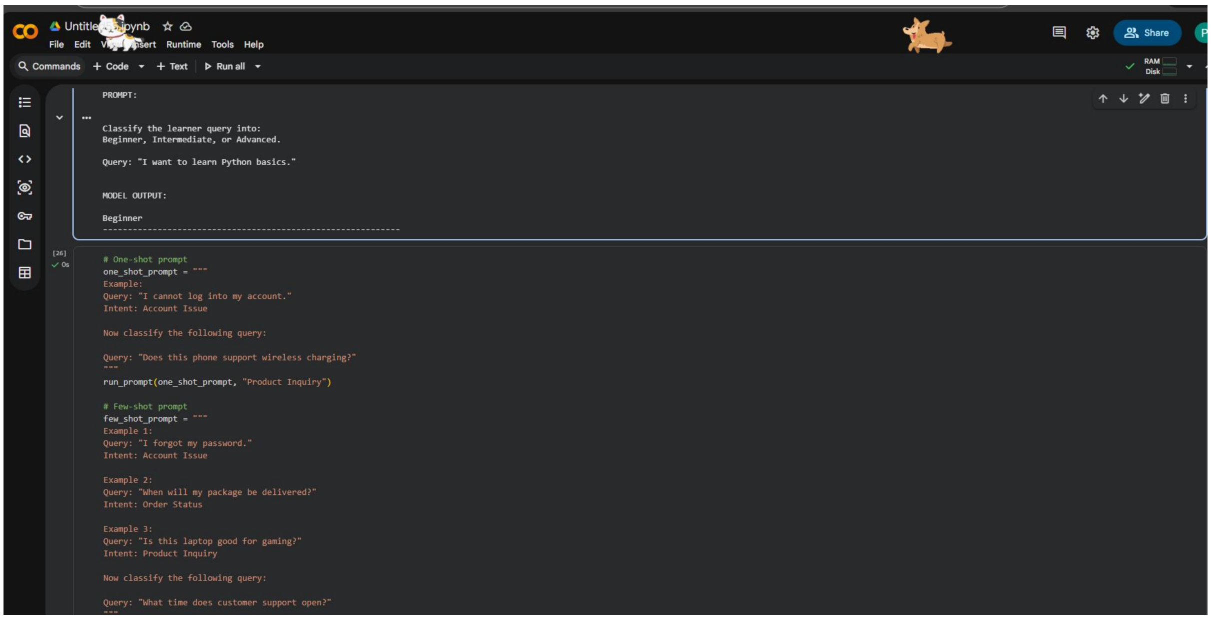
Problem Statement 4: Course Recommendation System


https://colab.research.google.com/drive/1qTgpNij-e0gggjE0BByDf62pdAP-wzFk?authuser=1#scrollTo=FYTdUW-oHmOF
Untitled55.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all
Share
RAM Disk

[42]

```
#problem -4
# Sample learner queries with corresponding course levels
sample_queries = [
    ("I want to learn Python basics.", "Beginner"),
    ("I am new to programming.", "Beginner"),
    ("I know Python and want to learn data structures.", "Intermediate"),
    ("I want to build machine learning models.", "Intermediate"),
    ("I want to master deep learning and transformers.", "Advanced")
]

for query, level in sample_queries:
    print(f"Query: {query}")
    print(f"Level: {level}\n")

# Prompt runner function
def run_prompt(prompt, output):
    print("PROMPT:\n")
    print(prompt)
    print("\nMODEL OUTPUT:\n")
    print(output)
    print("-" * 60)

# Zero-shot prompt
zero_shot_prompt = """
Classify the learner query into:
Beginner, Intermediate, or Advanced.

Query: "I want to learn Python basics."
"""

run_prompt(zero_shot_prompt, "Beginner")
```

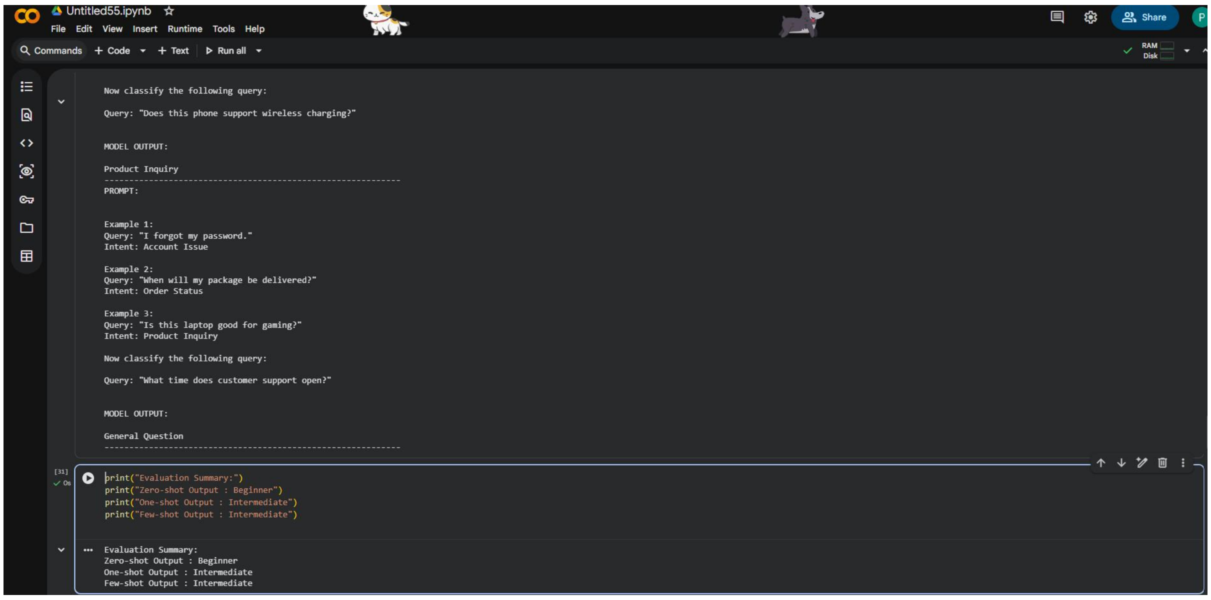
```
Query: I want to learn Python basics.
Level: Beginner

Query: I am new to programming.
Level: Beginner

Query: I know Python and want to learn data structures.
Level: Intermediate

Query: I want to build machine learning models.
```


Untitled55.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all
Share
RAM Disk

```
PROMPT:


Classify the learner query into:
Beginner, Intermediate, or Advanced.

Query: "I want to learn Python basics."


MODEL OUTPUT:

Beginner
------------------------------------------------------------
```

[26] 0s

```
# One-shot prompt
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")

# Few-shot prompt
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?"
"""
```

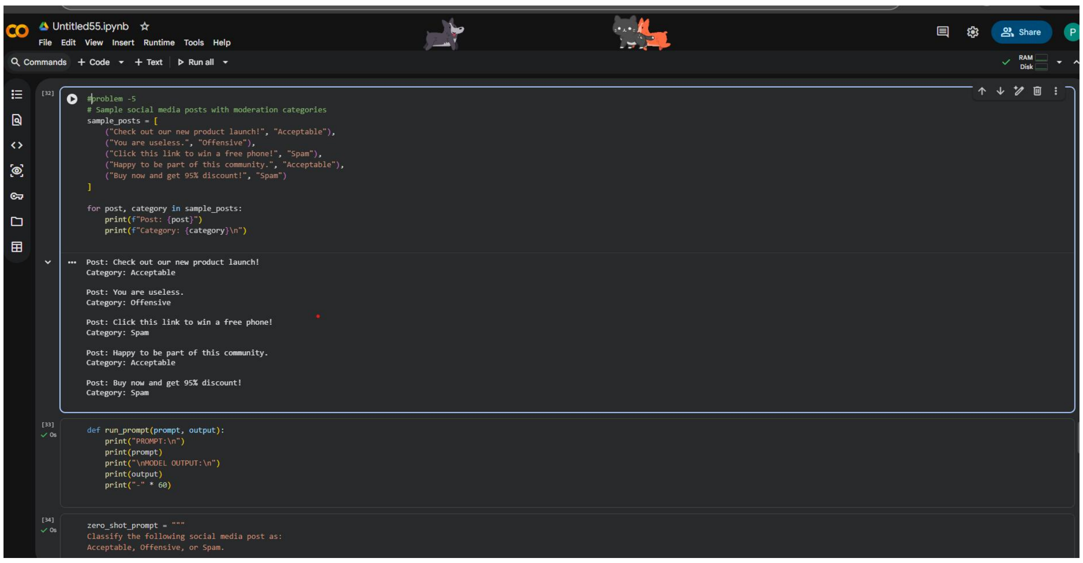


```
Now classify the following query:

Query: "Does this phone support wireless charging?"


MODEL OUTPUT:

Product Inquiry
------------------------------------------------------------
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
------------------------------------------------------------
```

```
[31]
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
```

Problem Statement 5: Social Media Post Moderation

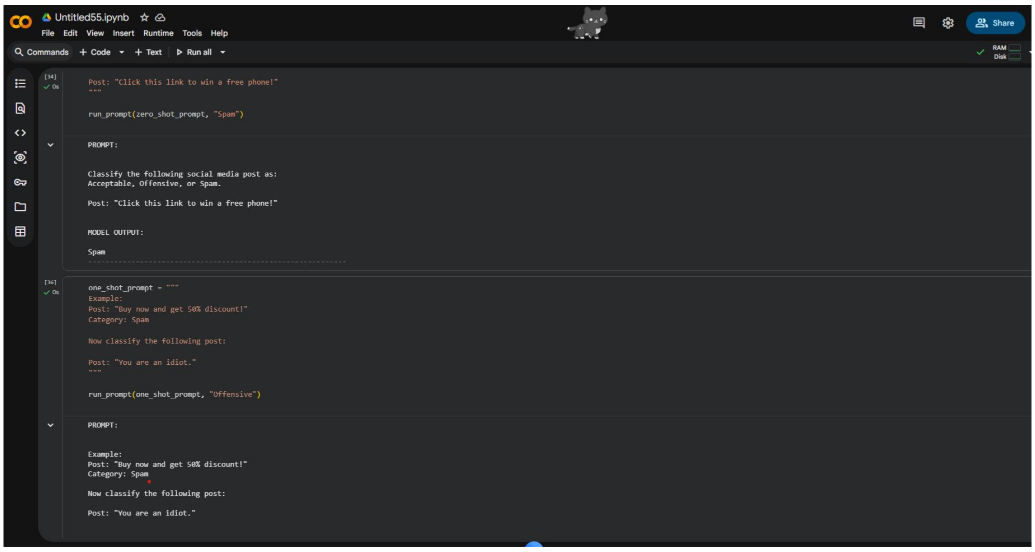


```
[32]
#problem -5
# Sample social media posts with moderation categories
sample_posts = [
    ("Check out our new product launch!", "Acceptable"),
    ("You are useless.", "Offensive"),
    ("Click this link to win a free phone!", "Spam"),
    ("Happy to be part of this community.", "Acceptable"),
    ("Buy now and get 95% 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 95% discount!
Category: Spam
```

```
[33]
def run_prompt(prompt, output):
    print("PROMPT:\n")
    print(prompt)
    print("\nMODEL OUTPUT:\n")
    print(output)
    print("-" * 60)
```

```
[34]
zero_shot_prompt = """
Classify the following social media post as:
Acceptable, Offensive, or Spam.
```

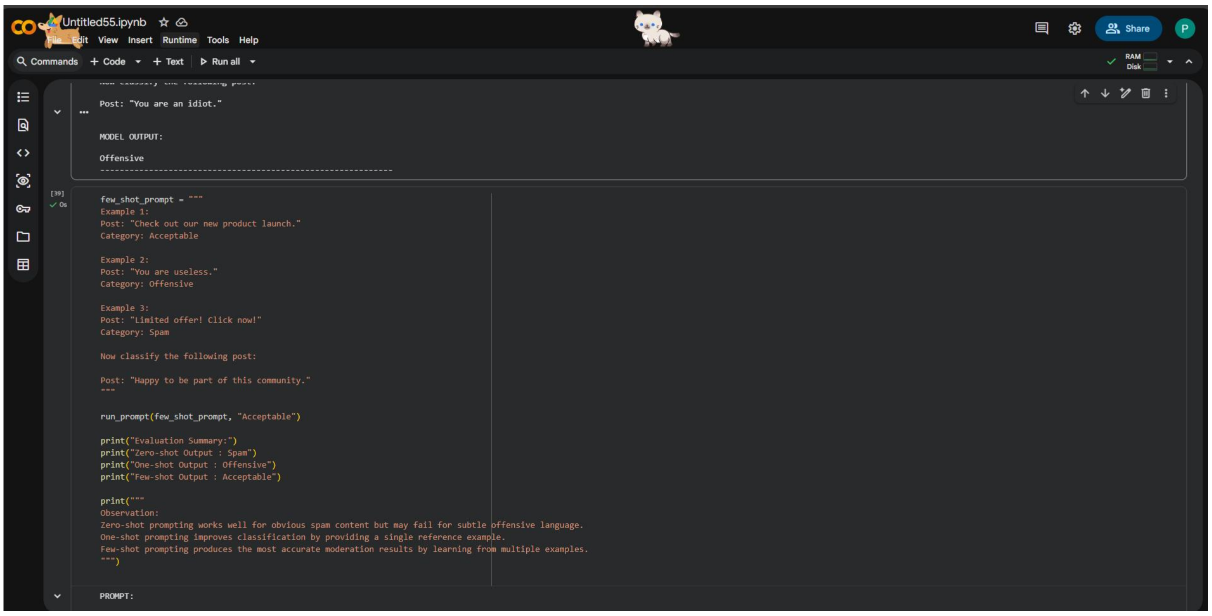

Untitled55.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all
Share
RAM Disk

```
[34]
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:

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

```
[36]
one_shot_prompt = """
Example:
Post: "Buy now and get 50% discount!"
Category: Spam

Now classify the following post:

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

run_prompt(one_shot_prompt, "Offensive")
```

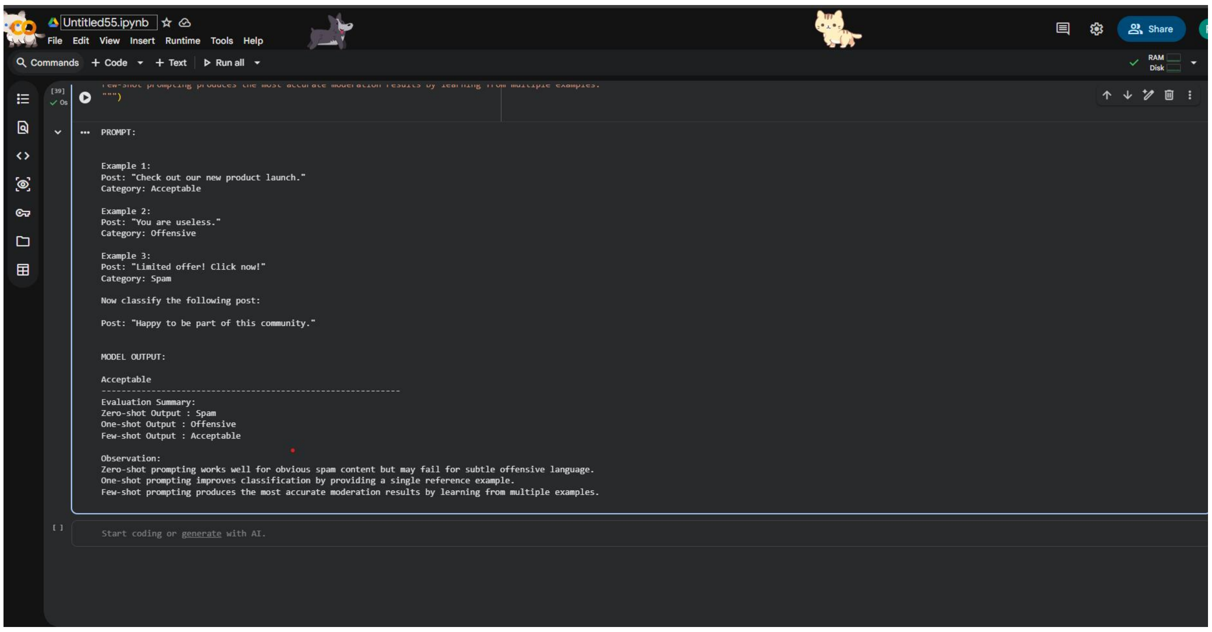
```
PROMPT:


Example:
Post: "Buy now and get 50% discount!"
Category: Spam

Now classify the following post:

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


Untitled55.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all
Share
RAM Disk

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


MODEL OUTPUT:

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

```
[39]
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")

print("Evaluation Summary:")
print("Zero-shot Output : Spam")
print("One-shot Output : Offensive")
print("Few-shot Output : Acceptable")

print("""
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.
""")
```

```
PROMPT:
```


Untitled55.ipynb
File Edit View Insert Runtime Tools Help
Commands + Code + Text Run all
Share
RAM
Disk
[39]
0s

```
Few-shot prompting produces the most accurate moderation results by learning from multiple examples.
""")
```

```
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
------------------------------------------------------------
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.