LAB ASSIGNMENT – 4.4

Name : Ch.Jyothika
Hall Ticket No. : 2303A51280
Batch No.: 05

1. Sentiment Classification for Customer Reviews
Scenario:
An e-commerce platform wants to analyze customer reviews and classify them into Positive, Negative, or Neutral sentiments using prompt engineering.
Tasks:
a) Prepare 6 short customer reviews mapped to sentiment labels.
b) Design a Zero-shot prompt to classify sentiment.
c) Design a One-shot prompt with one labeled example.
d) Design a Few-shot prompt with 3–5 labeled examples.
e) Compare the outputs and discuss accuracy differences.
Scenario
An e-commerce platform wants to classify customer reviews as **Positive, Negative, or Neutral** using prompt engineering.
(a) Sample Customer Reviews

Review	Sentiment
“The product quality is excellent and delivery was fast.”	Positive
“Very satisfied with the purchase.”	Positive
“The item is okay but nothing special.”	Neutral
“Delivery was delayed but product is fine.”	Neutral
“Worst experience ever, totally disappointed.”	Negative
“The product stopped working in two days.”	Negative

AI Generated Code:
(b) Zero-Shot Prompt
Classify the sentiment of the following review as Positive, Negative, or Neutral:
"The product quality is excellent and delivery was fast."

```
Classify the sentiment of the following review as Positive, Negative, or Neutral:

"The product quality is excellent and delivery was fast."
```

Output: Positive
(c) One-Shot Prompt
Example:
Review: "The product is terrible and useless."
Sentiment: Negative

Now classify:
Review: "The product quality is excellent and delivery was fast."

```
Example:
Review: "The product is terrible and useless."
Sentiment: Negative

Now classify:
Review: "The product quality is excellent and delivery was fast."
```

Output: Positive

(d) Few-Shot Prompt
Review: "The product is amazing and works perfectly."
Sentiment: Positive
Review: "Delivery was late and the box was damaged."
Sentiment: Negative
Review: "It's okay, not too good or bad."
Sentiment: Neutral
Now classify:
Review: "Very satisfied with the purchase."

```
Review: "The product is amazing and works perfectly."
Sentiment: Positive

Review: "Delivery was late and the box was damaged."
Sentiment: Negative

Review: "It's okay, not too good or bad."
Sentiment: Neutral

Now classify:
Review: "Very satisfied with the purchase."
```

Output: Positive

(e) Comparison & Analysis

Method	Accuracy	Reason
Zero-shot	Medium	No examples provided
One-shot	Better	Model understands pattern
Few-shot	Best	Context improves classification

Few-shot prompting gives the most accurate and consistent results.

2. Email Priority Classification

Scenario:

A company wants to automatically prioritize incoming emails into High Priority, Medium Priority, or Low Priority.

Tasks:

1. Create 6 sample email messages with priority labels.
2. Perform intent classification using Zero-shot prompting.
3. Perform classification using One-shot prompting.
4. Perform classification using Few-shot prompting.
5. Evaluate which technique produces the most reliable results and why.

Sample Emails

Email	Priority
“Server is down, fix immediately!”	High
“Client meeting postponed to tomorrow.”	Medium
“Thank you for your support.”	Low
“Urgent payment issue, respond ASAP.”	High
“Weekly report attached.”	Medium
“Just checking in.”	Low

Zero-Shot Prompt

Classify this email as High, Medium, or Low priority:
"Server is down, fix immediately!"

```
Classify this email as High, Medium, or Low priority:
"Server is down, fix immediately!"
```

Output: High

One-Shot Prompt

Email: "Payment failed, urgent fix needed."
Priority: High
Now classify:
"Client meeting postponed to tomorrow."

```
Email: "Payment failed, urgent fix needed."
Priority: High

Now classify:
"Client meeting postponed to tomorrow."
```

Output: Medium

Few-Shot Prompt

Email: "System crash reported."
Priority: High

Email: "Weekly update attached."
Priority: Medium
Email: "Thanks for your help."
Priority: Low
Now classify:
"Urgent payment issue, respond ASAP."

```
Email: "System crash reported."
Priority: High

Email: "Weekly update attached."
Priority: Medium

Email: "Thanks for your help."
Priority: Low

Now classify:
"Urgent payment issue, respond ASAP."
```

Output: High

Evaluation

Method	Reliability
Zero-shot	Moderate
One-shot	Good
Few-shot	Best

Few-shot gives context → better accuracy

3. Student Query Routing System

Scenario:

A university chatbot must route student queries to Admissions, Exams, Academics, or Placements.

Tasks:

1. Create 6 sample student queries mapped to departments.
2. Implement Zero-shot intent classification using an LLM.
3. Improve results using One-shot prompting.
4. Further refine results using Few-shot prompting.
5. Analyze how contextual examples affect classification accuracy.

Departments

- Admissions
- Exams
- Academics
- Placements

Sample Queries

Query	Department
“What is the admission process?”	Admissions
“When are semester exams?”	Exams
“Explain Python syllabus.”	Academics
“Placement companies list?”	Placements
“Fee structure details”	Admissions
“Result declaration date?”	Exams

Zero-Shot Prompt

Classify the query:
"When are semester exams?"

```
Classify the query:

"When are semester exams?"
```

Output: Exams

One-Shot Prompt

Query: "How to apply for admission?"
Department: Admissions
Now classify:
"When are semester exams?"

```
Query: "How to apply for admission?"
Department: Admissions

Now classify:
"When are semester exams?"
```

Output: Exams

Few-Shot Prompt

Query: "Syllabus for AI subject?"
Department: Academics
Query: "Placement drive details?"
Department: Placements
Query: "Fee structure?"
Department: Admissions
Now classify:
"When will results be announced?"

```
Query: "Syllabus for AI subject?"
Department: Academics

Query: "Placement drive details?"
Department: Placements

Query: "Fee structure?"
Department: Admissions

Now classify:
"When will results be announced?"
```

Output: Exams

Analysis

Few-shot prompting gives better domain understanding and routing accuracy.

4. Chatbot Question Type Detection

Scenario:

A chatbot must identify whether a user query is Informational, Transactional, Complaint, or Feedback.

Tasks:

1. Prepare 6 chatbot queries mapped to question types.
2. Design prompts for Zero-shot, One-shot, and Few-shot learning.
3. Test all prompts on the same unseen queries.
4. Compare response correctness and ambiguity handling.
5. Document observations.

Question Types

- Informational
- Transactional
- Complaint
- Feedback

Sample Queries

Query	Type
“What is AI?”	Informational
“Book a ticket for tomorrow.”	Transactional
“The app crashes often.”	Complaint
“Great service, thank you!”	Feedback
“How to reset password?”	Informational
“Refund my payment.”	Transactional

Prompting Comparison

Zero-Shot:
Classify the query: "The app crashes often."

```
Classify the query: "The app crashes often."
```

→ Complaint
One-Shot:
Query: "Great service!"
Type: Feedback
Now classify:
"The app crashes often."

```
Query: "Great service!"
Type: Feedback

Now classify:
"The app crashes often."
```

→ Complaint
Few-Shot:
Query: "What is AI?"
Type: Informational

Query: "Book my movie tickets."
Type: Transactional

Query: "The app is not working properly."
Type: Complaint

Query: "Excellent customer support!"
Type: Feedback

Now classify:
Query: "I want to cancel my order."

```
Query: "What is AI?"
Type: Informational

Query: "Book my movie tickets."
Type: Transactional

Query: "The app is not working properly."
Type: Complaint

Query: "Excellent customer support!"
Type: Feedback

Now classify:
Query: "I want to cancel my order."
```

→ Transactional

Comparison Table

Method	Accuracy	Reason
Zero-shot	Medium	No learning examples
One-shot	Good	Learns from one pattern
Few-shot	Best	Learns multiple patterns

Observations

✓ Few-shot prompting gives **highest accuracy**

✓ Context improves intent understanding

✓ Reduces ambiguity

✓ Best suited for real-world chatbot systems

Conclusion

Few-shot prompting is the **most reliable method** for chatbot question classification because:

- It understands patterns better
- Handles ambiguous queries
- Produces consistent results
- Suitable for production chatbots

5. Emotion Detection in Text

Scenario:

A mental-health chatbot needs to detect emotions: Happy, Sad, Angry, Anxious, Neutral.

Tasks:

1. Create labeled emotion samples.
2. Use Zero-shot prompting to identify emotions.

3. Use One-shot prompting with an example.
4. Use Few-shot prompting with multiple emotions.
5. Discuss ambiguity handling across techniques.

Scenario

A mental-health chatbot must detect user emotions in order to respond appropriately.
The emotions to be identified are:

- **Happy**
- **Sad**
- **Angry**
- **Anxious**
- **Neutral**

Sample Emotion Data

Text	Emotion
“I am very happy today!”	Happy
“I feel lonely and depressed.”	Sad
“This is unacceptable and annoying!”	Angry
“I am worried about my exams.”	Anxious
“Okay, noted.”	Neutral
“Everything is going well.”	Happy

Zero-Shot Prompt

Prompt

Detect the emotion in the following sentence:
"I am worried about my exams."

```
Detect the emotion in the following sentence:

"I am worried about my exams."
```

Output

Anxious

One-Shot Prompt

Prompt

Example:
Text: "I feel very sad today."
Emotion: Sad

Now classify:
Text: "I am worried about my exams."

```
Example:
Text: "I feel very sad today."
Emotion: Sad

Now classify:
Text: "I am worried about my exams."
```

Output

Anxious

Few-Shot Prompt

Prompt

Text: "I am very happy today."
Emotion: Happy

Text: "This is extremely frustrating."
Emotion: Angry

Text: "I feel nervous about my interview."
Emotion: Anxious

Text: "Nothing special, just normal."
Emotion: Neutral

Now classify:
Text: "I am worried about my exams."

```
Text: "I am very happy today."
Emotion: Happy

Text: "This is extremely frustrating."
Emotion: Angry

Text: "I feel nervous about my interview."
Emotion: Anxious

Text: "Nothing special, just normal."
Emotion: Neutral

Now classify:
Text: "I am worried about my exams."
```

Output

Anxious

Comparison & Analysis

Method	Accuracy	Remarks
Zero-shot	Medium	No context provided
One-shot	Good	Learns from one example
Few-shot	Best	Handles emotion clearly

Observation

- Few-shot prompting gives **highest accuracy**
- Emotional ambiguity is handled better
- Useful for mental-health chatbots
- Improves response quality and reliability

Conclusion

Few-shot prompting is the **most effective method** for emotion detection because it:

- Provides context
- Improves classification accuracy
- Reduces ambiguity
- Works well in real-world chatbot applications