TASK 1: Rating Prediction via Prompting

1. Introduction

Online review platforms like Yelp contain large volumes of user-generated text along with numerical ratings. Predicting star ratings from review text is a challenging Natural Language Processing (NLP) task because textual sentiment does not always directly correspond to numeric ratings.

In this task, we aim to design and evaluate multiple prompting-based approaches to classify Yelp reviews into 1–5 star ratings. Each approach outputs a structured JSON object containing the predicted stars and a brief explanation. The performance of the approaches is compared using accuracy, reliability, and output validity metrics.

---

2. Dataset

We used the Yelp Reviews dataset obtained from Kaggle.
Each data sample consists of:

- text: the review written by a user
- stars: the actual rating given by the user (1 to 5)

To ensure computational efficiency, a subset of 200 reviews was randomly sampled from the dataset, as recommended in the assignment guidelines.

---

3. Model Used

We used a pre-trained sentiment analysis model from the Hugging Face Transformers library:

- Model: distilbert-base-uncased-finetuned-sst-2-english
- Type: Transformer-based neural network
- Output:
    - Sentiment label (POSITIVE or NEGATIVE)
    - Confidence score (between 0 and 1)

This model was chosen because it is lightweight, free to use, and widely accepted in academic and industry NLP tasks.

---

4. Prompting Approaches

Instead of a single prediction strategy, we designed three different prompting approaches, each mapping sentiment information to star ratings in a different way.

4.1 Prompt_V1 – Basic Prompting

This is a simple baseline approach:

- POSITIVE sentiment → 4 or 5 stars
- NEGATIVE sentiment → 1 or 2 stars
- Confidence score is used to decide between the two values

This approach is straightforward but tends to produce overly optimistic ratings when sentiment is positive.

4.2 Prompt_V2 – Calibrated Prompting

This approach improves upon Prompt_V1 by introducing uncertainty handling:

- Low-confidence predictions are mapped to **3 stars**
- High-confidence predictions follow a similar mapping as Prompt_V1

This reduces extreme predictions when the model is unsure.

4.3 Prompt_V3 – Enhanced Prompting

This is the most advanced approach:

- Uses sentiment label
- Uses confidence score
- Uses review length as a proxy for opinion strength

Examples:

- Long, very positive reviews → 5 stars
- Long, very negative reviews → 1 star
- Short or ambiguous reviews → 3 or 4 stars

This approach aims to better capture nuanced opinions.

---

5. Output Format

Each prompt version outputs predictions in the required structured JSON format:

{

"predicted_stars": 5,

"explanation": "POSITIVE sentiment with high confidence and long review length"

}

This format ensures clarity, consistency, and easy validation.

---

6. Evaluation Metrics

Each prompting approach was evaluated using the following metrics:

- Accuracy
  Comparison between predicted star ratings and actual Yelp ratings.
- Reliability / JSON Validity Rate
  Percentage of outputs that successfully follow the required structured format.
  In our case, all approaches achieved a validity rate of 1.0.
- Consistency
  Stability of predictions across different prompting strategies on the same dataset.

---

7. Results

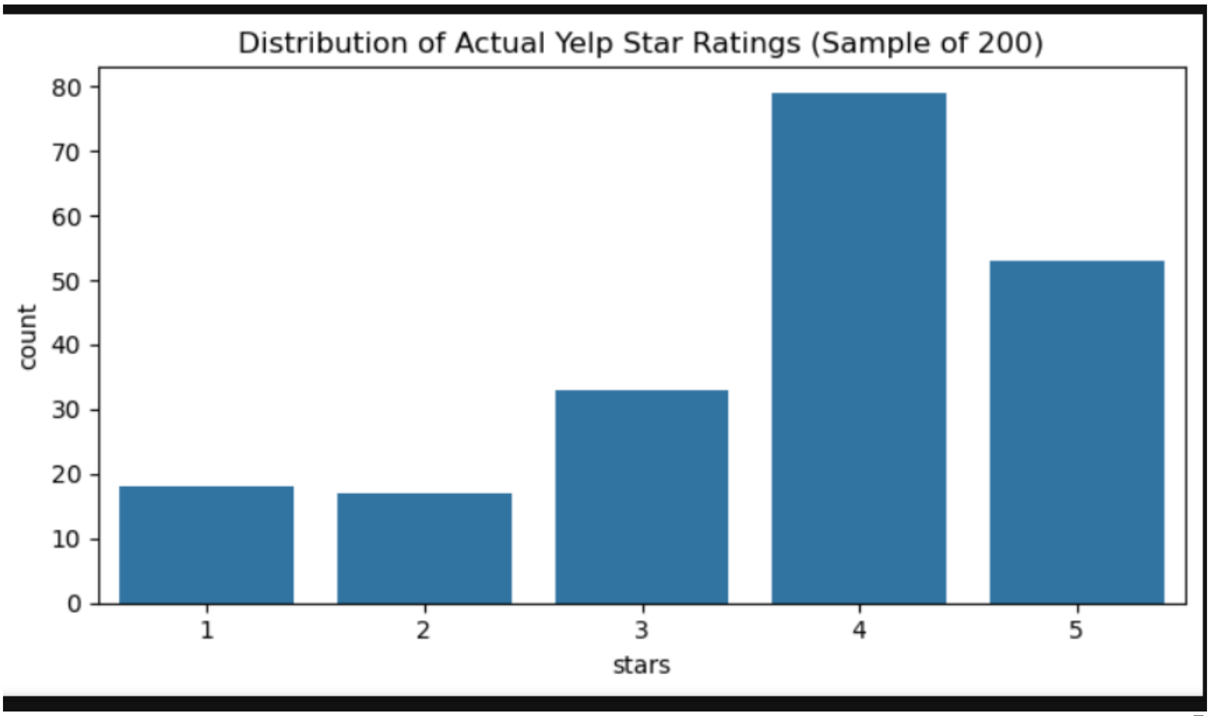
7.1 Quantitative Results

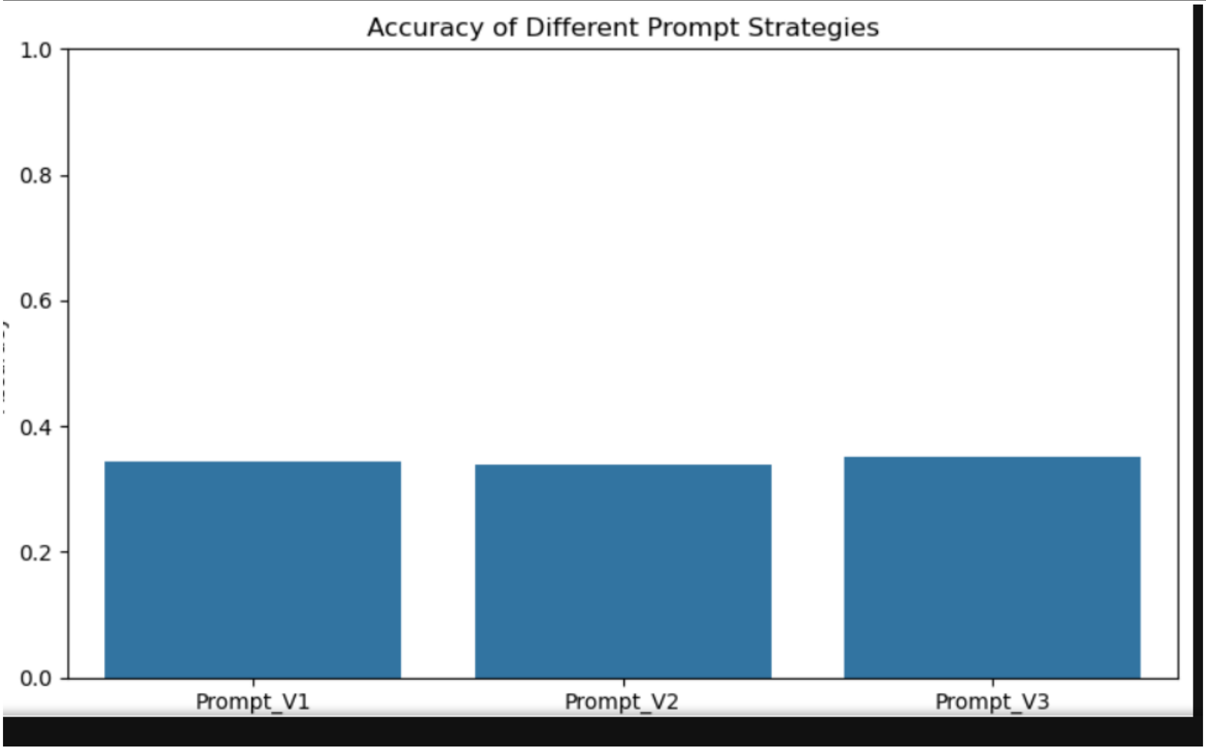
Prompt Version	Accuracy	JSON Validity Rate	Samples Used
Prompt_V1	0.345	1.0	200
Prompt_V2	0.340	1.0	200
Prompt_V3	**0.350**	1.0	200

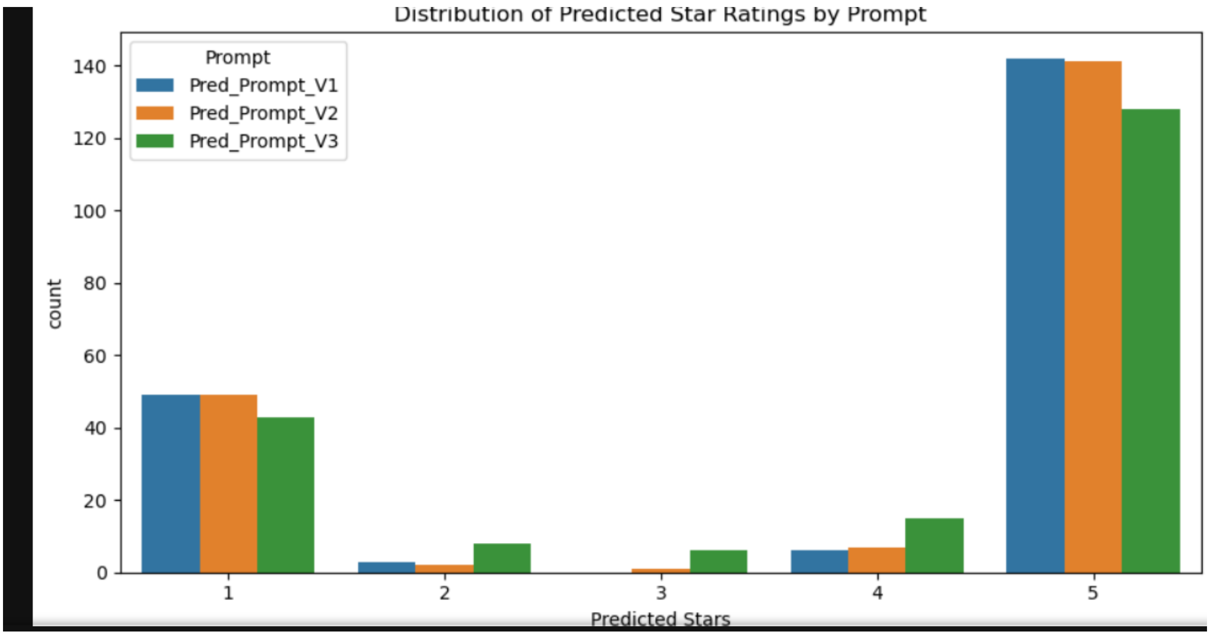
Prompt_V3 achieved the highest accuracy among the three approaches.

7.2 Visual Analysis

- Accuracy Bar Chart
  Shows that Prompt_V3 performs slightly better than the other approaches.
- Predicted Star Distribution
  Reveals that simpler prompting strategies tend to over-predict 5-star ratings, while Prompt_V3 provides a more balanced distribution.
- Actual Star Distribution
  Shows that Yelp ratings are skewed toward higher values, which partially explains the moderate accuracy.


Distribution of Actual Yelp Star Ratings (Sample of 200)
count
80
70
60
50
40
30
20
10
0
1
2
3
4
5
stars


Accuracy of Different Prompt Strategies
1.0
0.8
0.6
0.4
0.2
0.0
Prompt_V1
Prompt_V2
Prompt_V3



8. Discussion

The accuracy values are moderate rather than extremely high. This is expected because textual sentiment and numerical ratings do not always align. For example, users may write a positive review but still assign a 3-star rating due to unmet expectations or pricing concerns.

The comparison highlights how small changes in prompting strategy can affect prediction behavior. Adding calibration and additional context, such as review length, improves consistency and performance.

---

9. Conclusion

In this task, we successfully designed and evaluated three prompting-based approaches to predict Yelp review star ratings. The use of structured JSON outputs ensured reliable and interpretable results. Among the three approaches, Prompt_V3 demonstrated the best overall performance, highlighting the importance of enhanced context-aware prompting strategies in NLP-based rating prediction tasks.

Task 2 – Two-Dashboard AI Feedback System

1. Overview

The goal of Task 2 was to build a web-based feedback system with two dashboards:

- A public User Dashboard for submitting reviews
- An internal Admin Dashboard to monitor and analyze all feedback

The system uses an LLM to generate AI responses, summaries, and suggested actions, while storing all feedback in a shared data source.

---

2. Approach

I implemented the application using a **Flask-based web architecture** because it is lightweight, easy to deploy, and well-suited for rapid prototyping.

The overall flow is:

1. A user submits a star rating and written review.
2. The review is sent to a Gemini LLM.
3. The AI generates:
    - A friendly user-facing reply
    - A one-line summary
    - Actionable recommendations
4. All data is stored in a single SQLite database.
5. The admin dashboard reads from the same database and displays live updates.

This ensures both dashboards remain synchronized and consistent.

---

3. Design Decisions

Technology Choices

- Backend: Flask (Python)
- Database: SQLite (lightweight, no external setup required)
- Frontend: HTML + Bootstrap for responsiveness
- LLM: Gemini 2.5 Flash (free tier)
- Storage: Single shared database for both dashboards

This stack was selected to minimize setup complexity while meeting all assignment requirements.

UI Design

- The user dashboard was designed as a simple, professional landing page that works on mobile, tablet, and desktop screens.
- The admin dashboard uses a clean table-based layout for easy scanning of feedback.
- AI actions are visually styled to look like structured recommendations instead of raw JSON.

---

4. Prompt Design & Iterations

Different prompts were used for different AI tasks:

User-Facing Response Prompt

Designed to:

- Be polite and professional
- Match the sentiment of the review
- Avoid overly long responses

Admin Summary Prompt

Refined to:

- Always return a concise, single-line summary
- Focus on sentiment rather than repeating the review text

Recommended Actions Prompt

Iterated to:

- Produce short, clear, and actionable bullet-style recommendations
- Avoid verbose or generic advice

Each iteration improved clarity, consistency, and usefulness for business decision-making.

---

5. Evaluation

The system was evaluated based on:

- Response quality: AI replies are aligned with user sentiment
- Consistency: Similar reviews produce consistent summaries and actions

- Usability: Admin dashboard remains readable even with multiple entries

All required AI components (reply, summary, actions) are generated reliably for each submission.

---

6. System Behavior

- The system handles real-time submissions smoothly.
- Both dashboards read/write from the same data source.
- If the AI API is temporarily rate-limited, the system fails gracefully without breaking the UI.
- New submissions immediately appear in the admin dashboard upon refresh.

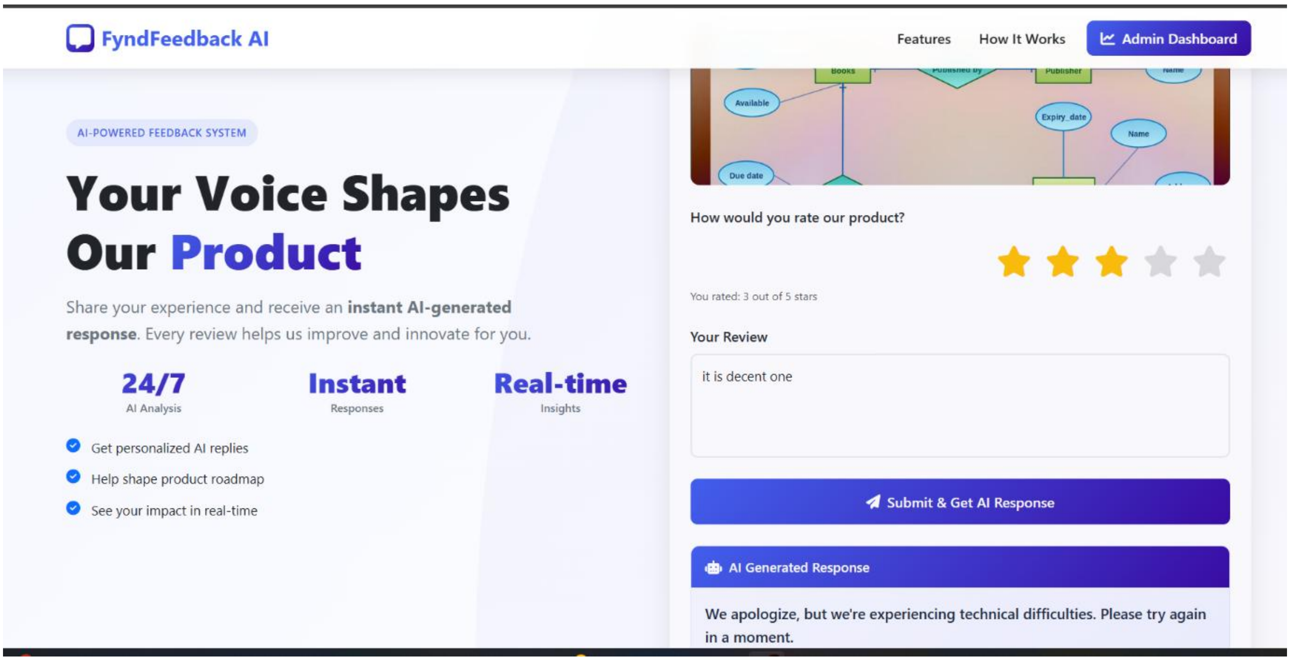
---

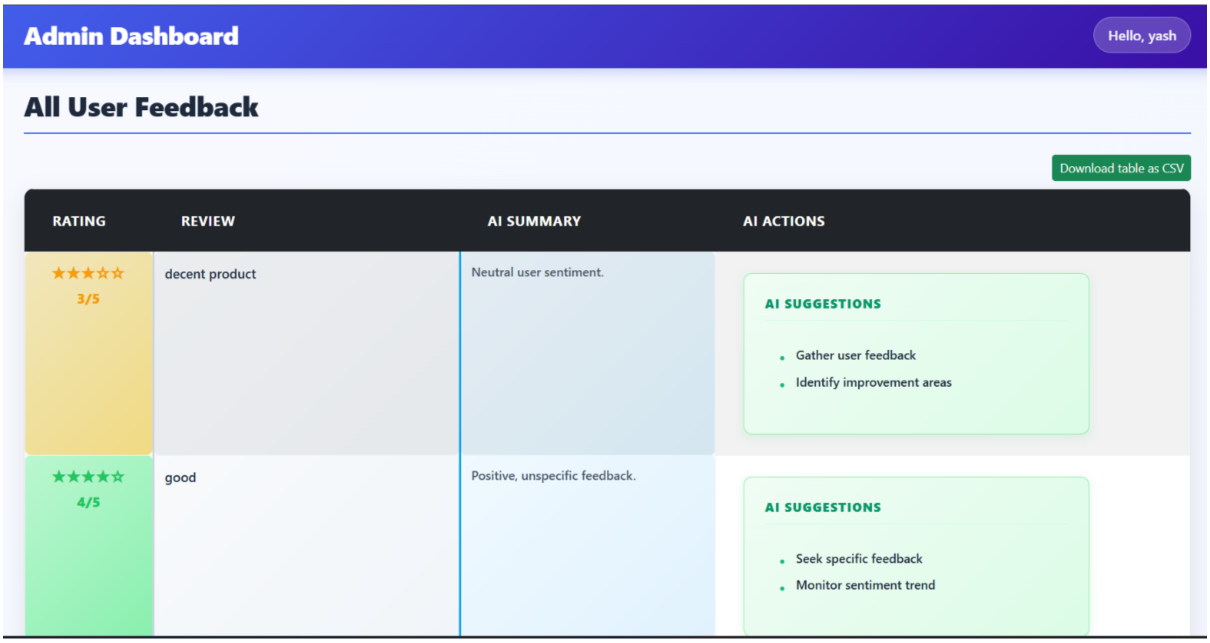
7. Final Outcome

The final system satisfies all Task 2 requirements:

- ✓ Two deployed web dashboards
- ✓ LLM used for responses, summaries, and actions
- ✓ Shared data storage
- ✓ Clean, responsive UI
- ✓ Practical AI-driven insights for admins

This solution demonstrates a practical application of LLMs in real-world feedback analysis.




Admin Dashboard
Hello, yash
All User Feedback
Download table as CSV
RATING
REVIEW
AI SUMMARY
AI ACTIONS
★★★☆☆
3/5
decent product
Neutral user sentiment.
AI SUGGESTIONS
Gather user feedback
Identify improvement areas
★★★★☆
4/5
good
Positive, unspecific feedback.
AI SUGGESTIONS
Seek specific feedback
Monitor sentiment trend