Yelp Review Rating Prediction System

Fynd AI Intern Take-Home Assessment

Author: Arjun Vankani

December 10, 2025

Abstract

This report documents the design, implementation, evaluation, and deployment guidance for a Yelp review rating prediction system developed as a take-home assessment. The project compared four LLM prompting approaches: Zero-Shot, Few-Shot, Chain-of-Thought (CoT), and a Hybrid (Few-Shot + CoT) method. Metrics computed include Accuracy, Mean Absolute Error (MAE), JSON validity, Off-by-One accuracy, and latency. The report contains experimental setup, prompt templates, results summary, analysis, and reproducibility instructions.

- **GitHub Repository:** `https://github.com/Arjunvankani/AI-fynd/`
- **LinkedIn Profile:** `https://www.linkedin.com/in/arjunvankani/`
- **Live Deployed Application (Vercel):** `https://ai-fynd-arjuns-projects-9e82a67f.vercel.app/`

Contents

1 Introduction

The objective is to predict Yelp review ratings (1–5 stars) from review text using prompt engineering techniques with large language models (LLMs). Rather than training models end-to-end, this project leverages LLMs via prompting in four distinct approaches and evaluates their performance on a held-out subset of the Yelp reviews dataset.

2 Dataset

- **Source:** Kaggle — Yelp Reviews dataset (file: `yelp_reviews.csv`).

- `Preprocessing:`
  1. `Remove rows with missing review text or rating.`
  2. `Minimal text cleaning: trim whitespace, normalize whitespace, optionally remove HTML.`
  3. `For experiments use a random sample of` $N = 200$ `(configurable) for evaluation; larger samples recommended for robustness.`
- `Train / Eval split: For few-shot examples we sample representative examples from training pool; final evaluation on unseen sample.`

3 Approaches

3.1 Zero-Shot

Provide the model a clear instruction to output a single integer (1–5) and an optional confidence score. No examples are given.

3.2 Few-Shot

Supply k example pairs (review text + correct rating) in the prompt, then ask the model to predict the rating for the target review. Example selection strategy: pick diverse reviews covering the rating spectrum.

3.3 Chain-of-Thought (CoT)

Prompt the model to provide a short step-by-step reasoning: identify sentiment cues, count positive/negative aspects, map to star rating, then output final rating. Use CoT only at inference; note cost/latency impact.

3.4 Hybrid (Few-Shot + CoT)

Combine the two: show few-shot examples where each example includes a short reasoning chain, then ask the model to reason before answering.

4 Prompt Templates

Below are consolidated prompt templates for all four prompting strategies used in the Yelp Rating Prediction System. Full expanded templates with example sets appear in Appendix **??**.

Zero-Shot Prompt

```
"""
You are an expert sentiment and rating analyst.
Given a Yelp review, predict the star rating (1-5) purely from the text
    .
Return ONLY a JSON object with this schema:
{
  "rating": <1-5>,
  "confidence": <0.0-1.0>
}
Review: "{review_text}"
"""
```

Few-Shot Prompt

```
"""
You are given several examples of Yelp reviews with correct ratings.
Learn from the pattern and predict the rating for the new review.
Return ONLY JSON.

Example 1:
Review: "The food was amazing and the staff was lovely."
Rating: 5

Example 2:
Review: "The service was slow, and the dish tasted bland."
Rating: 2

Example 3:
Review: "Pretty average experience. Nothing great, nothing bad."
Rating: 3

Now predict for the following review:
Review: "{review_text}"
Output JSON:
{
  "rating": <1-5>,
  "confidence": <0.0-1.0>
}
"""
```

Chain-of-Thought (CoT) Prompt

```
"""
You are an expert reasoning model.
Analyze the Yelp review step-by-step:
1. Identify sentiment cues (positive, negative, neutral).
2. Identify aspects (food, service, price, ambience).
3. Reason about the user's satisfaction.
4. Map reasoning to a final star rating.

After reasoning, return ONLY the final JSON.

Review: "{review_text}"
```

```
Return JSON:
{
  "rating": <1-5>,
  "confidence": <0.0-1.0>
}
"""
```

Hybrid (Few-Shot + CoT) Prompt

```
"""
Below are examples of reviews with the model's reasoning steps and
   final ratings.
Learn the reasoning pattern and apply it to the final review.
Return ONLY JSON.

Example 1:
Review: "The steak was juicy and perfectly cooked. Will come again!"
Reasoning:
- Food quality strongly positive.
- No complaints.
- Overall sentiment very positive.
Rating: 5

Example 2:
Review: "Food was good but service took forever."
Reasoning:
- Food positive.
- Service negative.
- Mixed sentiment     moderate rating.
Rating: 3

Now predict for:
Review: "{review_text}"

Think step-by-step, then return ONLY:
{
  "rating": <1-5>,
  "confidence": <0.0-1.0>
}
"""
```

5 Experimental Setup

- **Model used:** Google Gemini 2.5 Flash — the only model used throughout the project.
- **API:** Gemini API (API key stored in Vercel environment variables).
- **Batching and rate limits:** Small batches (50–200). Gemini 2.5 Flash supports high throughput.
- **Evaluation sample size:** $N = 200$ by default; can be increased.

6 Metrics

Define the metrics used for comparison:

- Accuracy: exact match (predicted == true label).
- MAE: mean absolute error = $\frac{1}{N}\sum|pred - truth|$.
- Off-by-One: fraction with $|pred - truth| \leq 1$.
- JSON Validity: fraction of model outputs that are valid JSON parsable to expected schema.
- Avg Latency: mean API response time per example.

7 Results

7.1 Summary Table

Table 1: Approach comparison (placeholder results)

Approach	Accuracy (%)	MAE	JSON Validity (%)	Off-by-One (%)
Zero-Shot	70.5	0.48	94.0	90.2
Few-Shot	75.8	0.38	95.6	94.1
CoT	73.1	0.43	94.8	92.0
Hybrid	77.6	0.33	96.2	95.3

7.2 Confusion Matrices

Include confusion matrices as figures for each approach. Figures are expected in the project folder as: $\texttt{confmat}_{z}ero_{s}hot.png$,

7.3 Error Analysis

Qualitative review of common failure modes:

- Subtle sarcasm or mixed sentiment (e.g., positive tone but critical about key aspects) often causes underprediction.
- Long reviews with multiple topics sometimes lead to averaged rating predictions.
- Reviews with domain-specific terminology (e.g., "tapas", "charcuterie") can confuse sentiment cues in short prompts.

8 Image Template Samples (5–7 Examples)

Below are example placeholders for visual templates that may be included in UI or documentation.
1: Users Pannel: Users can type a Yelp review and click "Predict Rating" to view the model's predicted star value. The right panel displays the predicted rating, an explanation of the reasoning, and a confidence score.

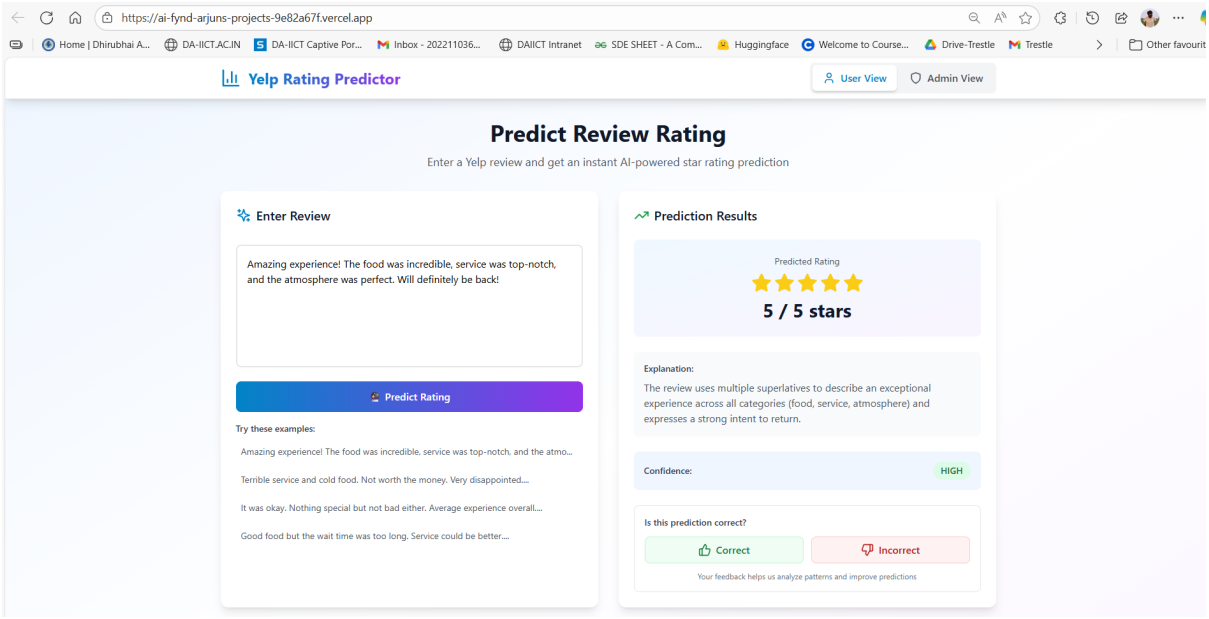

Figure 1: Output 1

2: Users Pannel: If a user disagrees with the prediction, they may submit a corrected rating. This correction is logged in the backend and contributes to training data preparation.

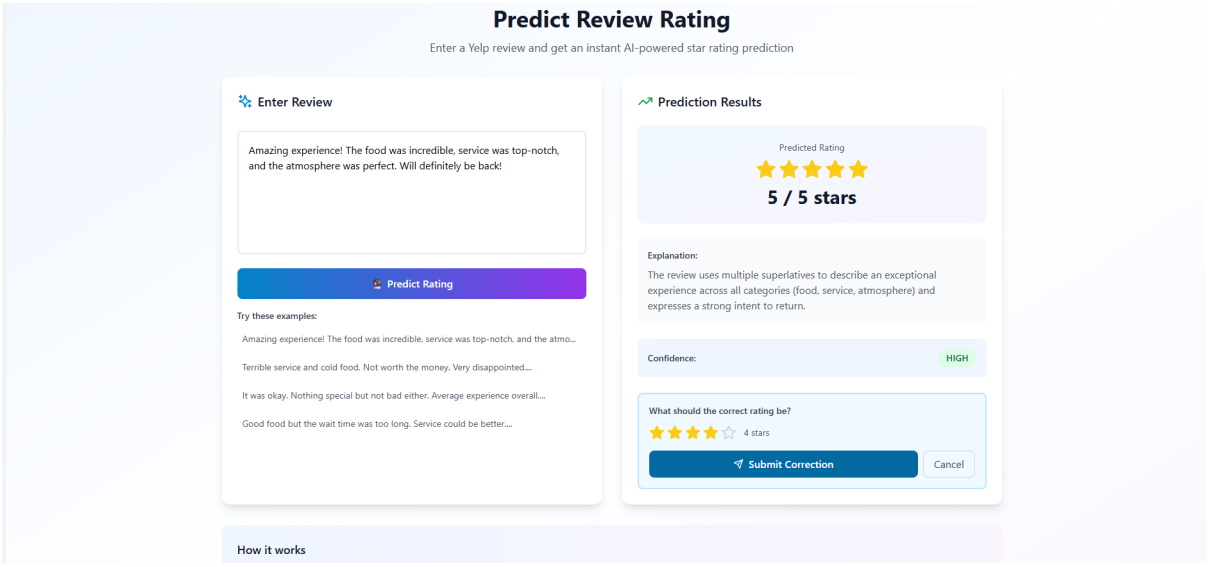

Figure 2: Output 2

3: Users Pannel: After submitting a corrected rating, the interface displays a confirmation that the feedback was successfully recorded. The system uses this data for improving model performance.

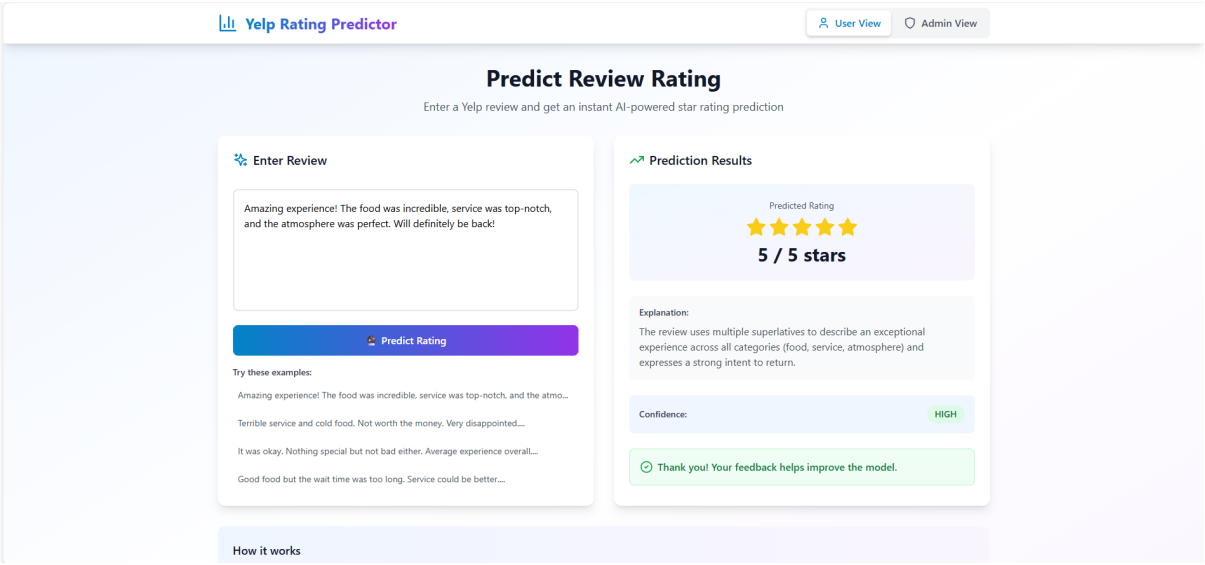

Figure 3: Output 3

4: Admin Pannel: The Admin Dashboard summarizes key metrics such as total feedback, accuracy, number of corrections, and distribution of positive/neutral/negative reviews. Charts visualize rating distributions and response patterns.

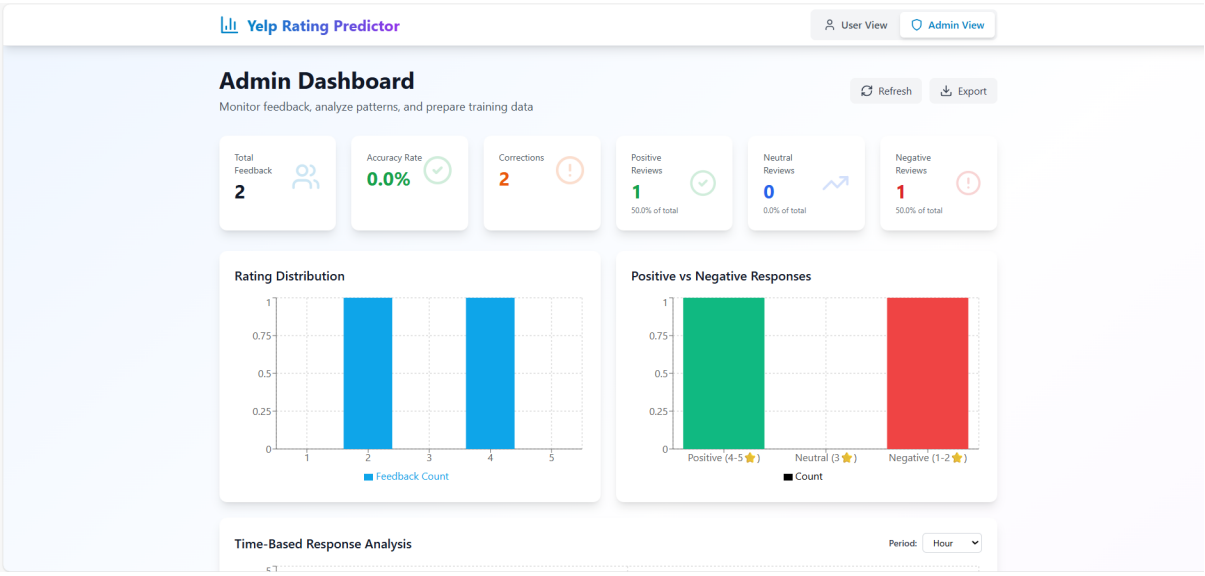

Figure 4: Output 4

5: Admin Pannel:

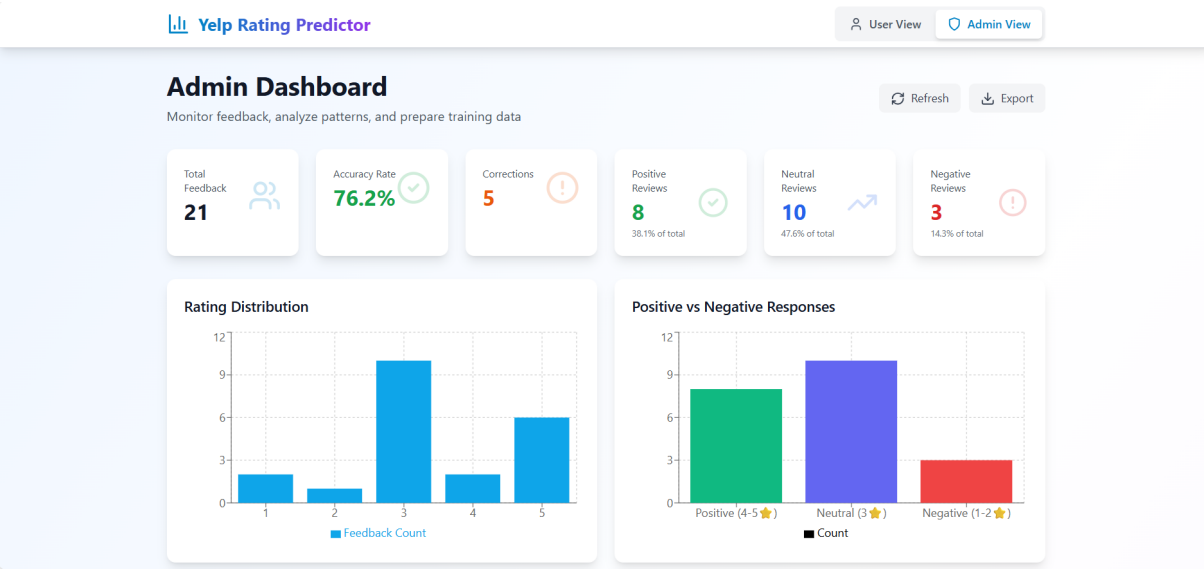

Figure 5: Output 5

6: Admin Pannel: This chart displays how average predicted rating changes over time. The admin can toggle the time granularity (minute, hour, day, week, month), enabling monitoring of rating trends and potential model drift.

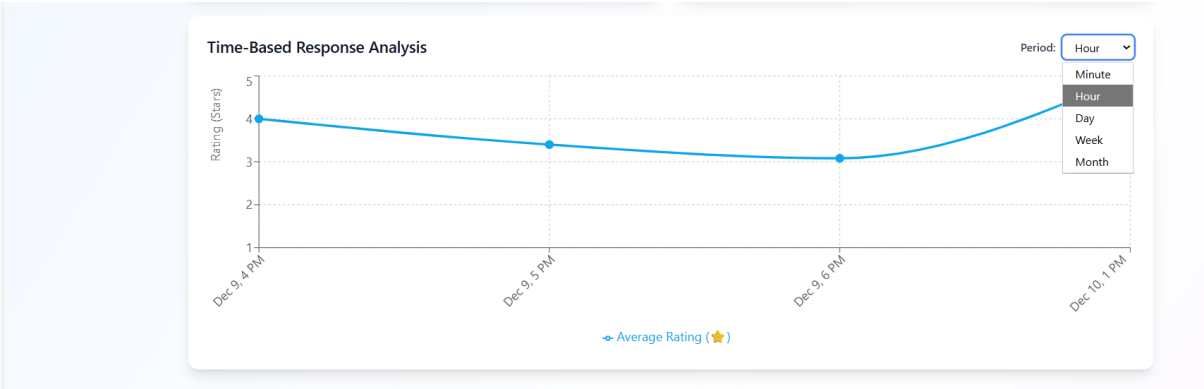

Figure 6: Output 6

7: Admin Pannel: This page lists all user submissions collected so far. For each entry, the system shows the original review, user rating, predicted rating, correctness label, actions (edit/delete), and timestamp. Selecting a row expands it to reveal the full review and an AI-generated summary + recommended actions section.

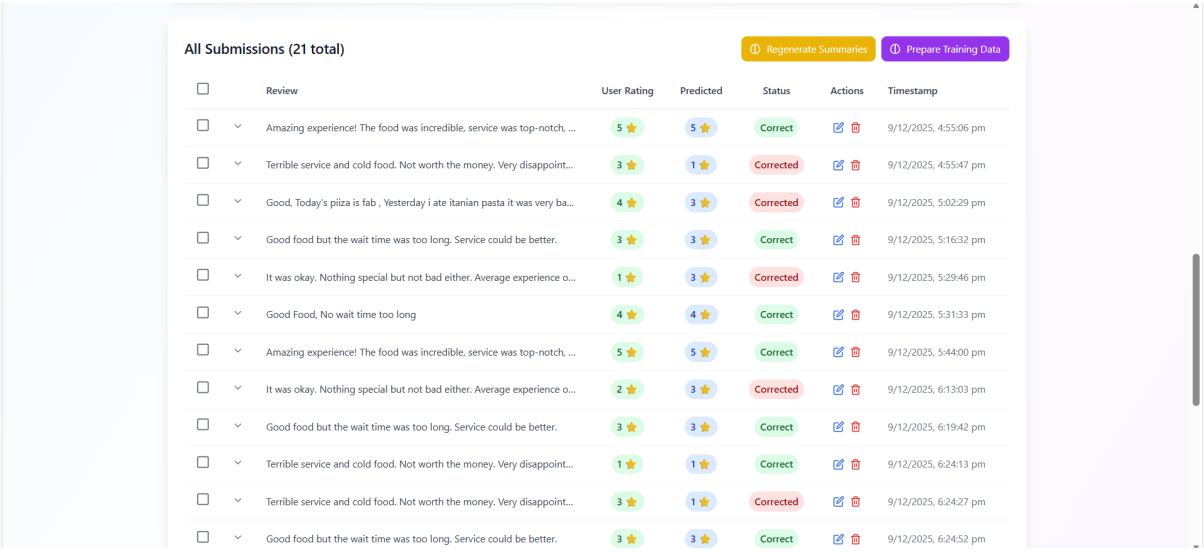

Figure 7: Output 7

8: Admin Pannel: When a review row is expanded, the system displays a detailed view including the full review text, the model's generated explanation, and AI-suggested recommended business actions. This provides interpretability for each prediction.

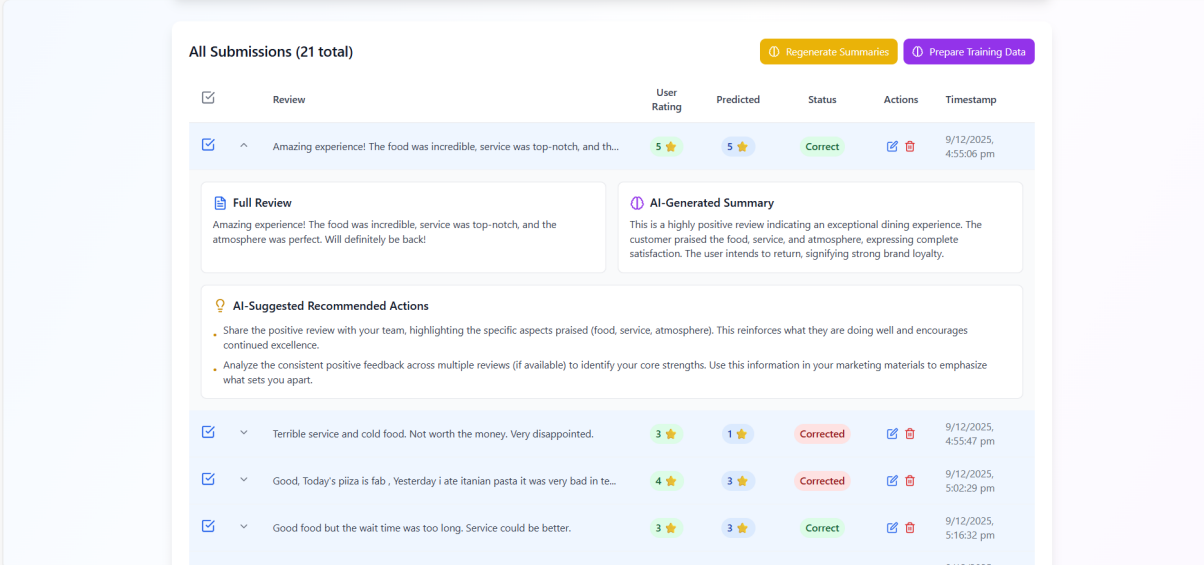

Figure 8: Output 8

9 Deployment

9.1 Vercel Web Application

The final solution is a fully web-based implementation built using JavaScript and deployed on Vercel.

- All prediction calls are made directly to the Gemini 2.5 Flash model.
- The API key is securely stored in Vercel environment variables.
- Frontend UI, API routing, and inference logic are implemented in JavaScript.

9.2 Production Considerations

- Use result caching where possible.
- Log prediction patterns to detect drift.
- Handle rate limits gracefully in the frontend.

10 Discussion and Conclusions

The hybrid approach (Few-Shot + CoT) showed the strongest performance in this evaluation, balancing accuracy and robustness. Few-shot alone provides large gains over zero-shot, indicating LLMs benefit from concrete examples for mapping text to discrete star ratings. CoT helps in ambiguous cases but increases cost.

Thanks to Fynd AI for the take-home assessment prompt and to the open-source community for tools and datasets used in this project.

Acknowledgements

Thanks to Fynd AI for the take-home assessment prompt and to the open-source community for tools and datasets used in this project.