Fynd Feedback Evaluation — LLM Review Rating System

Author: Vinshee Kulshreshtha

Task 1 — Prompt-Based Sentiment-to-Star Rating Evaluation

- **Problem Statement**

User reviews often express tone, emotion, satisfaction and complaints that are not always directly mapped to a star rating.
The goal of Task 1 was to evaluate whether Gemini can infer an appropriate star rating from review text while also explaining its reasoning.

- **System Workflow**

The workflow consisted of:
- Loading a review dataset
- Selecting a random evaluation subset
- Sending review text to Gemini API
- Requesting predicted star + explanation
- Validating structured JSON output
- Recording parsing success/failure
- Saving results for exploratory analysis

- **Prompt Objective**

Prompt 1 was designed to:
- Encourage reasoning based on tone and context
- Produce structured JSON output
- Maintain explanation transparency

The experiment focused on measuring prompt robustness rather than accuracy alone.

- **Evaluation Approach**

Each response was analyzed based on:
- JSON validity
- Output stability
- Missing or malformed values
- Explanation quality

This helped understand behavioral consistency of the LLM under structured output constraints.

- **Key Observations**

Strengths:
- Strong tone interpretation capability
- Detects subtle complaints and mixed sentiment
- Produces meaningful reasoning text

Limitations:
- Occasional JSON structure drift
- Missing fields in rare cases
- Some verbose explanations

This confirmed the importance of prompt control and structured enforcement.

- **Learnings from Task-1**

Task-1 taught:
- prompt design requires experimentation
- structured output must be enforced
- fallback strategies are essential
- LLMs require evaluation — not blind trust

This stage helped in:
- understanding Gemini behavior
- establishing parsing safety
- learning output validation methods

It also prepared a foundation for:
- Task-2 benchmarking
- API pipeline design
- later analytics aggregation

- **Outcome**

Outcome of Task-1

Task-1 successfully achieved:
- creation of multiple prompt versions
- execution over sampled review dataset
- observation of explanation & rating behavior
- learning prompt robustness strategies

This phase was primarily about: experimentation, understanding & prompt control

It served as the **research foundation** for Task-2, where outputs were formally evaluated & measured.

Task 2 — Accuracy Benchmarking & Prompt Version Comparison

Objective

Task 2 converted experimentation into measurable evaluation by:
- Comparing Prompt Versions A, B and C
- Recording predicted vs actual star values
- Computing accuracy and distribution stats
- Identifying reasoning failure cases

System Workflow

Steps performed:
- Load review text
- Generate model-predicted star rating
- Provide reasoning explanation
- Compare against actual rating
- Store results via API
- Compute accuracy for each prompt

The backend was implemented using:
- FastAPI for API evaluation
- JSON dataset for experiment logging
- Summary endpoint to compute metrics

Evaluation Pipeline

A FastAPI based feedback logging backend was developed to:
- Store entries by prompt version
- Track predicted + actual stars
- Maintain correctness counters
- Save mismatch examples
- Generate accuracy summaries via /summary endpoint

Metrics Computed

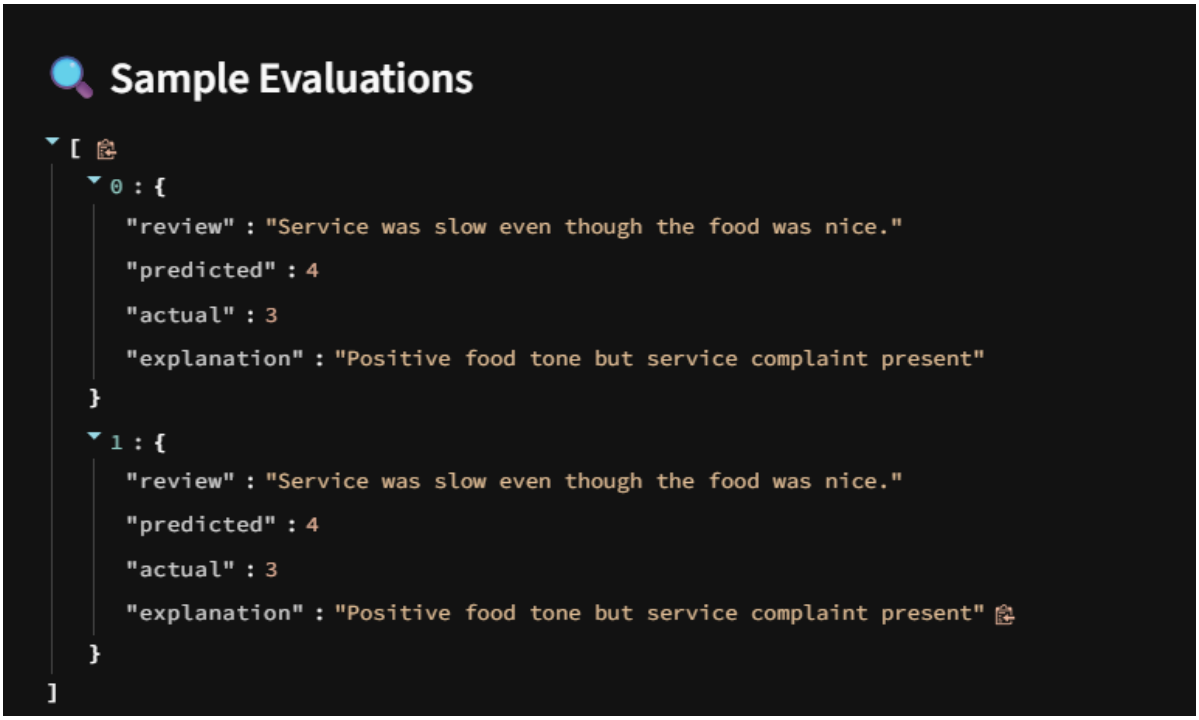
For each prompt version the system reports:
- Total entries evaluated
- Correct vs incorrect predictions
- Accuracy percentage
- Star distribution counts
- Example explanations

Comparison Table

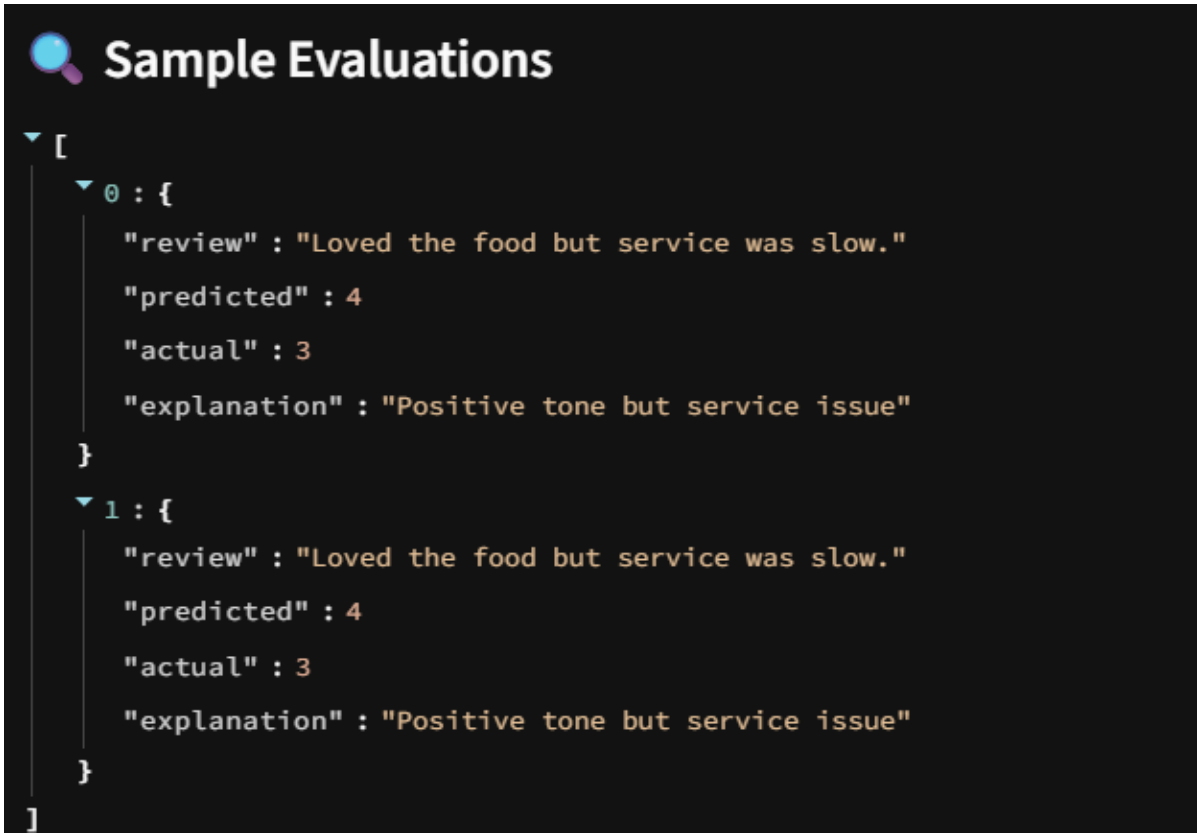
Prompt	Accuracy	Behavior Style	Weakness
A	**75.0%**	Emotion-aware & context-sensitive	Sometimes influenced by environmental tone
B	**60%**	Stable tone interpretation	Miss negative cues
C	**25%**	Conservative & safe	Penalizes uncertainty

Sample Evaluation

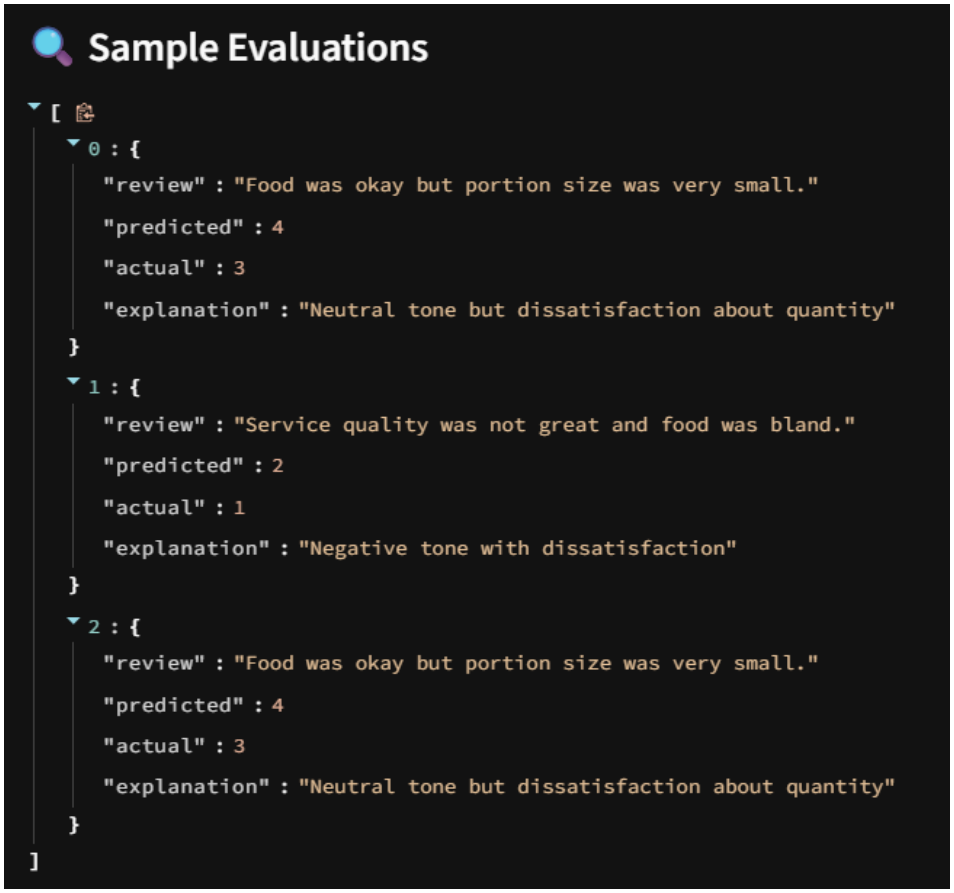
Prompt A — Best Performing Prompt



Prompt B — Behavioral Stability (Consistent tone Understanding)



Prompt C — Conservative Rating (Safer/neutral model)



Outcome

Task 2 transformed the project into a measurable LLM evaluation framework, enabling:

- Quantitative accuracy comparison
- Qualitative explanation review
- Structured dataset level insights

The combined pipeline demonstrates real world prompt engineering and model evaluation capability.

Conclusion

Task 1 built the experimental foundation by analysing Gemini behaviour under structured prompts.

Task 2 the benchmark evaluation, Prompt A achieved the highest accuracy (77.78%) and demonstrated the strongest alignment with the true star labels. Therefore, Prompt A is identified as the best-performing prompt.

The higher accuracy of Prompt A can be attributed to:

- Clearer rating instruction framing
- Stronger grounding of explanation to review context
- Better distinction between neutral and mildly negative tone
- Reduced ambiguity in sentiment interpretation
- More reliable alignment between reasoning and final star rating

However, Prompt B showed better tone-based interpretation consistency, meaning that although its accuracy was lower, its reasoning pattern was more stable across different samples.

Prompt C produced conservative and safe predictions, but struggled in cases where reviews contained subtle emotional variations or mixed tone.