

Fynd AI Intern – Take Home Assessment Report

1. Overview

This assignment consists of two parts:

- **Task 1:** Using prompt engineering with Large Language Models (LLMs) to predict Yelp review star ratings (1–5) while returning structured JSON output.
- **Task 2:** Building and deploying a web-based AI feedback system with two dashboards (User and Admin) that share a common data source and use LLMs for responses, summaries, and recommendations.

The solution focuses on practical LLM usage, prompt iteration, evaluation rigor, system design, and deployment readiness.

2. Task 1 – Rating Prediction via Prompting

2.1 Dataset & Setup

- Dataset: Yelp Reviews dataset from Kaggle
- A random sample of approximately **200 reviews** was used for evaluation efficiency.
- Each review was classified into a **1–5 star rating** using LLM prompting.
- Required output format:

```
{  
  "predicted_stars": 4,  
  "explanation": "Brief reasoning for the assigned rating."  
}
```

2.2 Prompt Iterations & Design Decisions

Three prompting strategies were designed and iteratively improved based on observed shortcomings.

Prompt 1: Zero-Shot Prompt

The initial prompt provided a simple instruction to classify the review sentiment and return a star rating in JSON format.

Why this was changed:

- The model frequently returned verbose natural language instead of strict JSON.
- Output structure was inconsistent.
- JSON validity was very low.

Learning:

Generic zero-shot instructions are insufficient when strict formatting and reliability are required.

Prompt 2: Rule-Based Prompt

The second prompt introduced **explicit sentiment-to-rating rules**, clearly mapping sentiment intensity to star values, along with strict instructions to output only valid JSON.

Why this improvement was made:

- To reduce ambiguity in how ratings are interpreted.
- To constrain model behavior for consistency.
- To significantly improve structured output reliability.

Observed impact:

- JSON validity increased dramatically.
- Output became far more consistent.
- Accuracy improved modestly due to clearer rating boundaries.

Trade-off:

The rigid rules occasionally limited nuanced sentiment interpretation.

Prompt 3: Few-Shot Prompt

The final prompt added **example input-output pairs** demonstrating correct JSON structure and rating logic.

Why this improvement was made:

- To demonstrate expected reasoning patterns.
- To improve semantic understanding of nuanced reviews.
- To balance reasoning quality with structured output.

Observed impact:

- Achieved the **highest accuracy** among all prompts.
- Improved alignment between review sentiment and predicted rating.
- Maintained high, though slightly reduced, JSON validity compared to the rule-based prompt.

Trade-off:

Longer context occasionally led to minor formatting deviations.

2.3 Evaluation Metrics

Each prompt was evaluated using:

- **Accuracy:** Predicted stars vs actual stars
- **JSON Validity Rate:** Percentage of outputs that were valid JSON
- **Reliability & Consistency:** Stability of structured output across samples

Prompt Type	Accuracy	JSON Validity
Zero-shot	0.533	0.075
Rule-based	0.579	0.975
Few-shot	0.625	0.800

2.5 Key Insights

- Zero-shot prompting is unreliable for structured outputs.
- Rule-based prompts maximize output consistency.
- Few-shot prompting improves semantic accuracy while maintaining acceptable structure.
- There is a clear trade-off between **format reliability** and **reasoning quality**.

Prompt engineering is an iterative process driven by empirical evaluation rather than assumptions.

3. Task 2 – Two-Dashboard AI Feedback System

3.1 System Architecture

Task 2 was implemented as a **single Streamlit web application** containing two dashboards:

- **User Dashboard (Public-Facing)**
- **Admin Dashboard (Internal-Facing)**

Both dashboards read and write to a **single persistent data source (data_store.json)**, ensuring live updates and data consistency.

This design avoids synchronization issues and reflects real-world SaaS architecture.

3.2 User Dashboard

The User Dashboard allows users to:

- Select a star rating
- Write a short review
- Submit feedback

On submission:

- An **AI-generated user-facing response** is returned
 - The review, rating, and AI-generated insights are stored persistently
-

3.3 Admin Dashboard

The Admin Dashboard displays a **live-updating list of all submissions**, including:

- User rating
- User review
- AI-generated summary

- AI-suggested recommended action

Additional analytics include:

- Total number of submissions
 - Average rating
 - Rating distribution
-

3.4 LLM Usage in Task 2

LLMs are used for:

- Generating polite, user-facing responses
- Summarising customer feedback
- Suggesting recommended next actions for administrators

Defensive parsing and fallback logic are implemented to handle malformed LLM outputs gracefully.

4. Deployment

- The application is fully deployed as a **web-based Streamlit app**.
 - Both dashboards are accessible via **public URLs**.
 - The Admin Dashboard is accessed via sidebar navigation within the same deployed app.
 - This ensures both dashboards operate on the **same stored data source**, satisfying the requirement for live updates.
-

5. Security Considerations

- No API keys are hard-coded in the repository.
- Task 1 API usage is environment-variable based and disabled by default for safe evaluation.
- Task 2 does not expose any secrets.
- This follows standard security and deployment best practices.

6. Conclusion

This project demonstrates:

- Practical prompt engineering and evaluation for LLM-based classification
- Structured output enforcement and reliability analysis
- Clean system design with shared persistence
- Robust handling of LLM variability
- Production-aware deployment decisions

All assignment requirements are fully satisfied.

7. Submission Links

- **GitHub Repository:** <https://github.com/diyabodiwala/fynd-ai-intern-assignment>
- **User Dashboard URL:** <https://huggingface.co/spaces/BlaBla234/fynd-admin-dashboard>
- **Admin Dashboard URL:** <https://huggingface.co/spaces/BlaBla234/fynd-admin-dashboard> *(same app, accessible via sidebar)*