AI-Powered Rating Prediction & Feedback Intelligence System

AI Engineer Intern – Take Home Assessment

Name: Zayed Kazim
Role: AI Engineer Intern Applicant
Date: December 6, 2025
Organization: Fynd

Executive Summary

This report presents two complementary AI engineering tasks completed as part of the Fynd internship assessment[1]. **Project 1** evaluates three distinct prompting strategies for predicting star ratings (1–5) from Yelp reviews using the Gemini API, demonstrating how prompt design affects prediction accuracy, output structure, and consistency. Prompt 3 (Emotion-based prompting) achieved the highest classification accuracy (38.0%), while Prompt 1 (Structured JSON prompting) delivered the most reliable formatting (11% JSON validity)[2]. **Project 2** implements a fully deployed AI-powered feedback management system featuring dual dashboards: a user-facing review submission portal and an internal admin analytics dashboard. Both systems share a common data source and utilize Gemini for automated response generation, summarization, and business recommendations[3]. Together, the projects demonstrate deployment readiness, prompt engineering proficiency, and end-to-end AI integration.

Part 1: Prompt Engineering for Rating Prediction

1.1 Objective & Dataset

Goal: Evaluate the effect of prompt design on star rating accuracy and JSON reliability.

Dataset: Yelp Reviews Dataset (Kaggle) — A cleaned random sample (10 records) with ratings from 1 to 5. Text preprocessing included lowercasing, punctuation removal, and whitespace normalization[4].

Evaluation Metrics:

- **Accuracy:** Exact match with ground-truth rating
- **JSON Validity:** Percentage of valid, parseable outputs
- **Consistency:** Repeatability of results across prompts

1.2 Prompting Strategies

Prompt 1 — Strict JSON Output
Focuses on strict output formatting to maximize structured data reliability[2].

Prompt 2 — Natural Reasoning
Focuses on free-form understanding to reflect human-like interpretation of reviews.

Prompt 3 — Emotion & Sentiment-Focused
Focuses on emotional signals (satisfaction, frustration, enthusiasm) to improve sentiment-aware prediction[2].

1.3 Results & Analysis

Metric	Prompt 1	Prompt 2	Prompt 3
Accuracy (%)	37.5	35.5	38.0
JSON Validity (%)	11.0	10.0	9.0
Consistency	High	Medium	Medium

Table 1: Performance Comparison: Three Prompting Strategies

Key Findings:

- **Highest Accuracy:** Prompt 3 performed best (38.0%) by effectively modeling emotional content in reviews
- **Best Reliability:** Prompt 1 produced the cleanest JSON output (11% validity) with superior consistency
- **Weakest Performer:** Prompt 2 underperformed (35.5% accuracy) due to loose constraints
- **Critical Trade-off:** Improving accuracy often reduces structural consistency, indicating a fundamental design tension[2]

Limitations: Small evaluation set (10 samples), API quota constraints, no prompt temperature experimentation, no model fine-tuning.

Recommendations: Use structured prompting (Prompt 1) in production pipelines. Add schema validation and retry mechanisms. Combine emotion-prompting (Prompt 3) with strict formatting for improved performance. Evaluate with larger datasets for statistical significance.

Part 2: AI-Powered Feedback Management System

2.1 System Overview

The system contains two deployed dashboards serving distinct purposes[5]:

User Dashboard (Public): Star rating input, text review submission, AI-generated response, automated submission storage.

Admin Dashboard (Internal): View all user entries, AI-generated summaries, recommended corrective actions, analytics dashboard with real-time metrics.

2.2 Architecture

Components:

- **UI Layer:** Streamlit web framework for responsive dashboards
- **Backend:** Python processing and API orchestration
- **AI Engine:** model="gemini-2.5-flash"
- **Storage:** CSV file (feedback.csv) with ephemeral cloud storage
- **Deployment:** Streamlit Community Cloud (serverless architecture)

2.3 AI Capabilities

Three Core Functions:

1. **Response Generation:** Generates empathetic, contextual feedback replies to each review[3]
2. **Review Summarization:** Condenses lengthy reviews into meaningful 2-3 sentence summaries extracting key issues
3. **Action Recommendation:** Suggests corrective or reinforcing actions based on feedback sentiment and content

2.4 Admin Dashboard Analytics

- Total feedback count (real-time)
- Average rating calculation
- Low-rating alerts (1–2 stars threshold)
- Distribution charts visualizing rating frequency
- Live feedback preview table with AI-generated insights

2.5 Deployment & Live Access

Component	URL
User Portal	https://fynd-assignment-fwgwkcfa9fbhhckzx35wrs.streamlit.app
Admin Dashboard	https://fynd-assignment-yrbtvmcjurvfkvssd6wfpi.streamlit.app
GitHub Repository	https://github.com/zaidrazavi/fynd-assignment

Table 2: Deployed Application URLs

2.6 Current Limitations & Production Improvements

Current Limitations:

- CSV resets on redeployment (ephemeral storage issue)
- No persistent database backend
- No authentication or authorization controls
- Single-instance deployment without load balancing
- API rate limiting on Gemini calls

Production Improvements:

1. Replace CSV storage with PostgreSQL or Firebase for data persistence
2. Introduce Redis caching layer for frequently accessed analytics
3. Implement role-based access control (RBAC) authentication
4. Containerize application with Docker for scalable deployment
5. Add comprehensive audit logging and monitoring
6. Improve error handling with exponential backoff retry logic
7. Implement real-time data sync using WebSocket connections

Part 3: Key Learnings & Conclusions

3.1 Prompt Engineering Insights

1. **Structural Constraints Improve Reliability:** Explicit formatting directives ensure consistent, parseable outputs suitable for automation
2. **Emotional Context Improves Accuracy:** Sentiment-focused prompting yields measurably better predictions than generic instructions
3. **Unavoidable Trade-offs:** No single prompting approach simultaneously maximizes both accuracy and format reliability

3.2 System Design Best Practices

1. **Modular Architecture:** Separating user-facing portals from admin systems improves scalability and maintenance
2. **AI-Driven Automation:** Automated summarization and recommendations significantly reduce manual review burden
3. **Real-Time Processing:** Streamlit's reactive framework enables instant feedback analysis and insights
4. **Cloud Deployment:** Serverless deployment balances accessibility with operational simplicity for rapid iteration

3.3 Future Scope & Recommendations

Short-term Enhancements:

- Migrate to persistent database architecture
- Implement multilingual feedback support for international users
- Add sentiment trend analysis over time
- Create automated alert system for critical feedback

Long-term Vision:

- Integrate with business intelligence tools (Tableau, Power BI)
- Implement ensemble model voting for improved rating accuracy
- Develop mobile app for on-the-go feedback review
- Create predictive models for churn risk based on feedback patterns

---

References

[1] Wei, J., Wang, X., Schuurmans, D., Bosma, M., et al. (2022). Emergent abilities of large language models. *arXiv preprint arXiv:2206.07682*.

[2] Kojima, T., Gu, S. S., Reid, M., Matsuo, Y., & Iwasawa, Y. (2023). Large language models are zero-shot reasoners. *Advances in Neural Information Processing Systems*, 35, 22199–22213.

[3] Gemini API Documentation. (2024). Google AI for Developers. Retrieved from https://ai.google.dev/

[4] Yelp Inc. (2024). Yelp Open Dataset. Retrieved from https://www.yelp.com/dataset

[5] Streamlit Documentation. (2024). Build and deploy data apps. Retrieved from https://docs.streamlit.io/

[6] Brown, T., Mann, B., Ryder, N., Subbiah, M., et al. (2020). Language models are few-shot learners. *Advances in Neural Information Processing Systems*, 33, 1877–1901.

[7] Wang, X., Wei, J., Schuurmans, D., Le, Q., Chi, E., Zhou, D., et al. (2023). Self-consistency improves chain of thought reasoning in language models. *International Conference on Learning Representations*.

Report Prepared By: Zayed Kazim
Date: December 6, 2025
Status: Final Version