Task 1 – Yelp Review Rating Prediction via Prompting

1. Objective

The goal of Task 1 was to design prompts that classify Yelp reviews into 1–5 stars, returning structured JSON with both the predicted rating and a brief explanation. The prompts were compared for accuracy.

2. Approach

- **Dataset**: Sample of 200 reviews from the Kaggle Yelp Reviews dataset
- **LLM**: HuggingFace Flan-T5-Small (instruction-tuned, free, GPU-compatible)
- **JSON Handling**: The model output was parsed using a robust JSON parser capable of handling partial or free-form outputs (e.g. “3 Stars”)
- **Truncation**: Reviews exceeding 512 tokens were truncated to ensure compatibility with the model
- **Evaluation Metrics**:
    - **Accuracy**: Match between actual and predicted stars
    - **JSON Validity Rate**: Percentage of responses that could be parsed correctly
    - **Consistency**: Stability of predictions across repeated runs

3. Prompt Strategies

Three prompt versions were designed and evaluated:

- **Prompt v1** – Simple Zero-Shot :
    - Directly asks the model to predict star rating and explanation.
- **Prompt v2** – Structured JSON Prompt:
    - Explicitly instructs the model to return output in strict JSON format. Emphasizes schema compliance.
- **Prompt v3** – Reasoned Prompt:
    - Encourages step-by-step reasoning before producing the final answer.

4. Evaluation Results

Prompt Version	Accuracy	JSON Validity Rate	Consistency
v1 – Zero-Shot	0.603	0.315	1.0
v2 – Rubric-Based	0.427	0.995	1.0
v3 – Reasoned	0.000	0.000	1.0

5. Observations

- Prompt v1 achieved the highest accuracy but struggled with JSON compliance (validity)
- Prompt v2 produced highly reliable structured output, making it suitable for downstream systems
- Prompt v3 failed due to excessive verbosity and poor adherence to the required output format
- Clear, constrained prompts significantly improved LLM reliability

6. Key Insights

1. **Tradeoff between structure and prediction quality**: Enforcing strict JSON can reduce accuracy if the model misinterprets instructions.
2. **Model robustness**: Free-form prompting (v1) produces more accurate predictions, but post-processing is needed to parse JSON.
3. **Consistency**: All predictions are consistent for a single run; repeated runs could further evaluate reliability.

7. Conclusion

The evaluation demonstrates that prompt design significantly impacts LLM outputs. Prompt v1 (Zero - Shot) provides better star prediction accuracy, while v2 (Rubric - Guided) ensures highly structured JSON outputs suitable for downstream processing. Prompt v3 (Reasoned) failed as the model wasn't complex enough. Proper post-processing and truncation allow HuggingFace models to handle real-world reviews effectively.

Task 2: Two Dashboard AI Feedback System

1. Objective

To build and deploy a simple web-based system with:

- A **User Dashboard** for submitting reviews
- An **Admin Dashboard** for monitoring and analyzing feedback

2. Technology Stack

- Python
- Streamlit (Web UI)
- Hugging Face Transformers (LLM inference)
- JSON-based storage
- Deployed on Hugging Face Spaces

3. User Dashboard (Public-Facing)

Features:

- Users select a star rating (1–5)
- Write a short review
- Submit the review
- Receive an AI-generated response

LLM Behavior:

- Positive reviews - thank the customer
- Negative reviews - apologize and acknowledge feedback
- Prompts were designed to avoid repetition of the review text

4. Admin Dashboard (Internal-Facing)

Features:

- Displays all submissions in a tabular format
- Periodic re-execution for real time updates
- Columns include:
    - Rating
    - Review
    - AI-generated summary
    - AI-recommended action

AI Logic:

- Summaries are concise and non-repetitive
- Recommended actions follow these strict rules:
    - Positive reviews - thank the customer
    - Negative reviews - apologize and indicate corrective steps

5. Deployment

- Both dashboards were deployed as a Streamlit app on Hugging Face Spaces
- Each dashboard has a public URL
- Both dashboards read/write from the same storage logic

6. Conclusion

This project demonstrates:

- Effective prompt engineering and evaluation
- Practical handling of LLM limitations such as repetition and output instability
- End-to-end system design, deployment, and validation

The solution balances correctness, simplicity, and real-world usability while meeting all assignment requirements.