MSDS 434- Documentation Reflection

**The Role of API Design, Error Handling, and Observability in ML Services**

Fei Yan  
Date: August 2025

Working on this project helped me realize the importance of proper documentation and observability. At first, my code was distributed and only partially commented, and it was very hard when bugs were being found. When the project grew larger, I realized proper documentation was not only useful for other individuals but also for me the programmer.

**API Design and Documntation**

Writing a Flask program with clear routes (/health, /predict, /metrics), I realized simple and uniform API design facilitates testing tremendously. Providing clear instructions on the expected JSON input and example requests allowed me to easily test the service using curl. This preemptively eliminated confusion when introducing extra fields or debugging input parse error.

**Error Handling and Logging**

When In deployment, most failures presented as 500 Internal Server Errors. Without sufficient logging, it would have been close to impossible to identify the root cause. Adding structured error messages ("error": ".", "detail": ".") on API responses was a very important refactoring. This made it possible to give immediate feedback to the client and point me toward configuration issues, ignored parameters, or BigQuery query failures.

**Monitoring and Observability**

Incorporating Prometheus metrics within the Flask web service (request\_count, request\_latency) allowed me insights about how the system acted under incoming requests. Although I didn't implement the full monitoring panel, it did give me knowledge about how these metrics can be scraped and graphed within Prometheus at prod and utilized to identify anomalies. It made me understand that machine learning services are not only about accuracy but about performance and reliableness. In a nutshell, documentation and observability turned this initially brittle prototype of a project into a more maintainable and robust system. Precise input handling, clear error handling, and plain monitoring bred the confidence that the application would continue functioning correctly while it was scaling or being subjected to unexpected input. This was an eye-opener that has led me to believe that decent documentation and observability are at least as valuable as the machine learning model itself.