MSDS 434- Final Project Reflection

**Building and Deploying a Cloud-Native Used Car Price Prediction System**  
Fei Yan  
Date: August 2025

During the course of ten weeks, what started as a basic data exploration project developed into a full cloud-native machine learning system. In hind-sight it is clear how each of the ten weeks was a build on and refinement of what was done the proceeding week and how the iterative process led up to the eventual system success.

Week 1-3: Exploring Data and Data Preprocessing

In the early phase, my priority was learning about the raw dataset of pre-owned vehicles. I pre-processed the data by cleaning it up and accommodating the missing values. Also, simple exploratory data analysis was done and significant features like year of manufacture, manufacturer, model of vehicle, odometer reading, cylinders, and transmission were identified. They became the basis of the predictive model at a later point of time. I gained insights at this time about the paramount role of proper data pre-processing. Even finer details like column names normalization or removing extreme outlier observations affected the stability of the model.

Week 4-5: Construction and Training of the Model

Once I cleaned the data, I utilized BigQuery ML and created an AutoML regression model to predict prices. At first, I was encountering schema mismatches and errors caused by field names being inconsistent. This made me go back to the preprocessing step and take extra care matching the training dataset with the output anticipated by the model. Through Week 5, I had a working model that was generating price outputs, albeit at points where values did not quite sound realistic. This was a very valuable lesson about checking outputs and never assuming that the results of the model were accurate because the program had run properly.

Week 6-7: Containerization and API Development After training the model, the attention was shifted toward hosting it using a cloud service. I created a Flask app (app.py) that interacted with BigQuery ML, and it served /health and /predict endpoints. Prometheus metrics were added for monitoring purposes. Containerizing this app using Docker was yet another learning achievement, as I had to properly configure the environment variables (e.g., MODEL\_TABLE) and ensure the application was able to talk to BigQuery. At first, I was encountering repeated 500 Internal Server Error situations, which were annoying but compelled me to explore logs, perms, and API response handling. These debugging sessions refined my skills on systematically eliminating probable causes.

Week 8-9: Running on Cloud Run and Debugging

Deploying the service on Cloud Run was both the most thrilling and the most troublesome step. I faced several deployment failures, including inappropriate service account permissions, environment variable problems, and failed prediction query issues. Frequently the logs contained very little information and were therefore hard to use when debugging. After granting the service account of the Cloud Run service the appropriate BigQuery roles (user, jobUser, dataViewer) and redeploying the service using new configurations, these problems were overcome. It was at the point when the /predict endpoint at last returned a valid predicted\_price that a breakthrough was achieved. It proved all of the steps undertaken previously, from cleaning data to training models to deployment.

Week 10: Finalize and Reflect

In With the end-to-end service now operational, I can generalize about the end-to-end experience. Designing and operating a complex service of this kind was technically and psychologically challenging. At times it was discouraging when mistakes repeated and were not always obvious. Each failure, though, did make me solidify my debugging procedure, review logs carefully, and double-check system configurations. This project has been a great illustration of the iterative development of data science and engineering work that is needed in the real world and the value of patience and systematic troubleshooting. In short, I'm pleased that I built a complete machine learning application: raw data all the way through training a model, containerizing an API, and deploying it up to a cloud service of a production-grade. This project allowed me to get a realistic understanding of what it takes to deploy machine learning and the many details involved beyond the actual model.