Subject: Capstone Machine Learning Deployment Assignment

Project : Car Price Prediction API with FastAPI & Google Cloud Run

---

1. Executive Summary

This project aims to build and deploy a machine learning model to predict car prices based on customer demographics and vehicle specifications. The solution uses Python's FastAPI framework for serving predictions and is deployed using Google Cloud Run for scalability and availability. The model is trained on a car sales dataset, preprocessed for quality, and exposed via a REST API.

---

2. Problem Statement & Dataset Description

Problem: Predict the price of a car based on inputs like age, annual income, engine type, company, etc.

Dataset Source: Kaggle - Car Sales Dataset

Features Used:

- Age
- Annual Income
- Company
- Model
- Engine
- Transmission
- Dealer Region

Target Variable:

- Price ($)

	Car_id	Date	Customer Name	Gender	Annual Income	Dealer_Name	Company	Model	Engine	Transmission	Color	Price ($)	Dealer_No	Body Style	Phone
0	C_CND_000001	01-02-2022	Geraldine	Male	13500	Buddy Storbeck's Diesel Service Inc	Ford	Expedition	DoubleÂ Overhead Camshaft	Auto	Black	26000	06457-3834	SUV	8264678
1	C_CND_000002	01-02-2022	Gia	Male	1480000	C & M Motors Inc	Dodge	Durango	DoubleÂ Overhead Camshaft	Auto	Black	19000	60504-7114	SUV	6848189
2	C_CND_000003	01-02-2022	Gianna	Male	1035000	Capitol KIA	Cadillac	Eldorado	Overhead Camshaft	Manual	Red	31500	38701-8047	Passenger	7298798
3	C_CND_000004	01-02-2022	Giselle	Male	13500	Chrysler of Tri-Cities	Toyota	Celica	Overhead Camshaft	Manual	Pale White	14000	99301-3882	SUV	6257557
4	C_CND_000005	01-02-2022	Grace	Male	1465000	Chrysler Plymouth	Acura	TL	DoubleÂ Overhead Camshaft	Auto	Red	24500	53546-9427	Hatchback	7081483

---

3. Model Development Process

Preprocessing:

- Missing value imputation
- Label encoding for categorical variables
- Feature scaling with StandardScaler

Model:

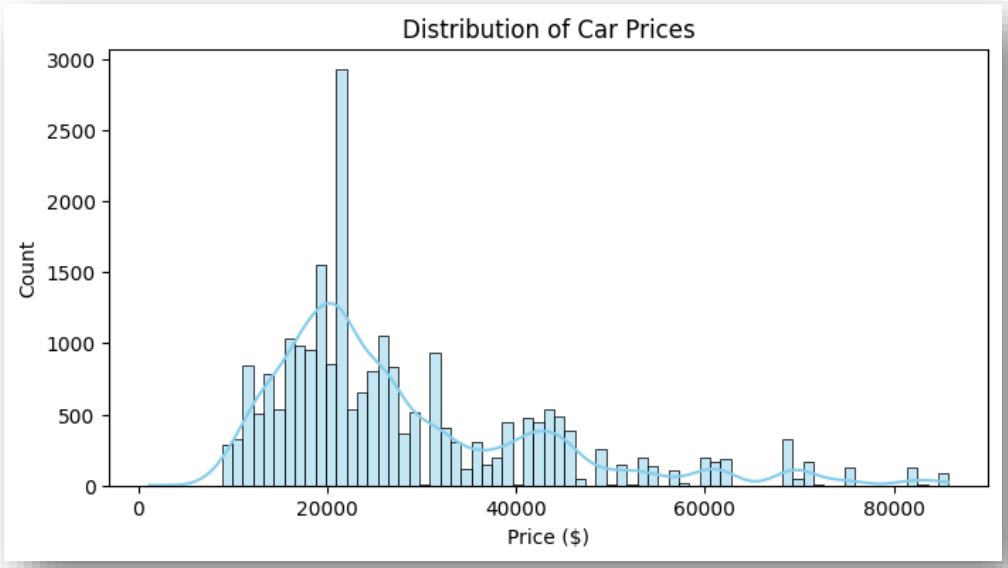
- RandomForestRegressor (with GridSearchCV)

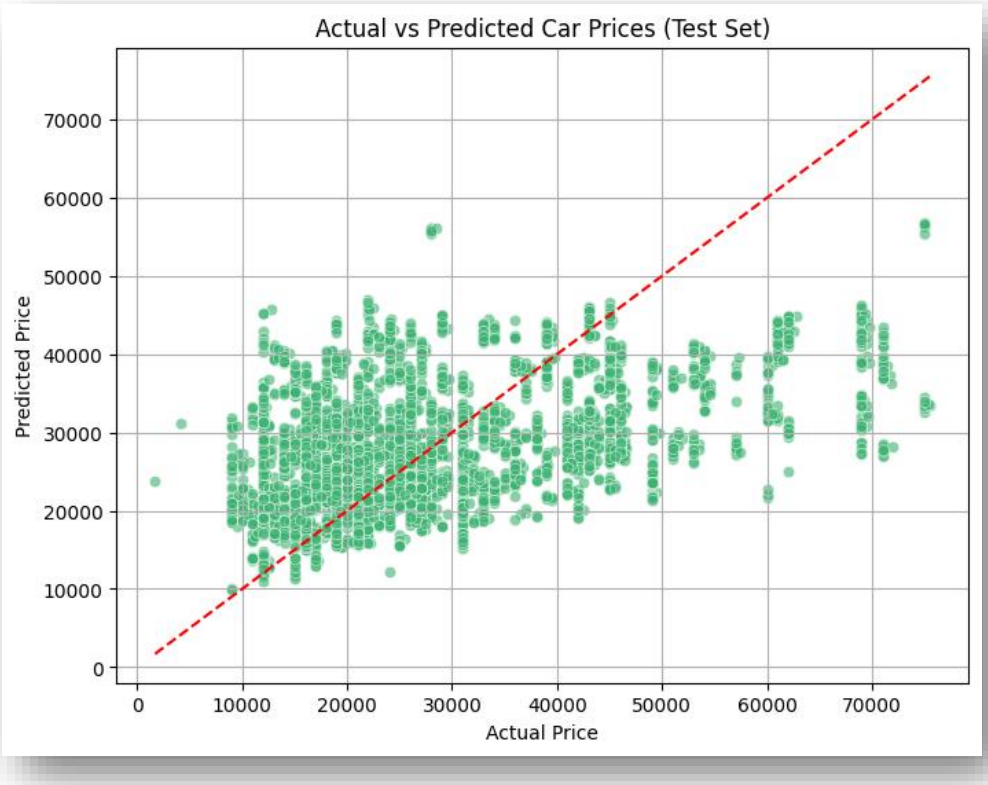
Evaluation Metrics:

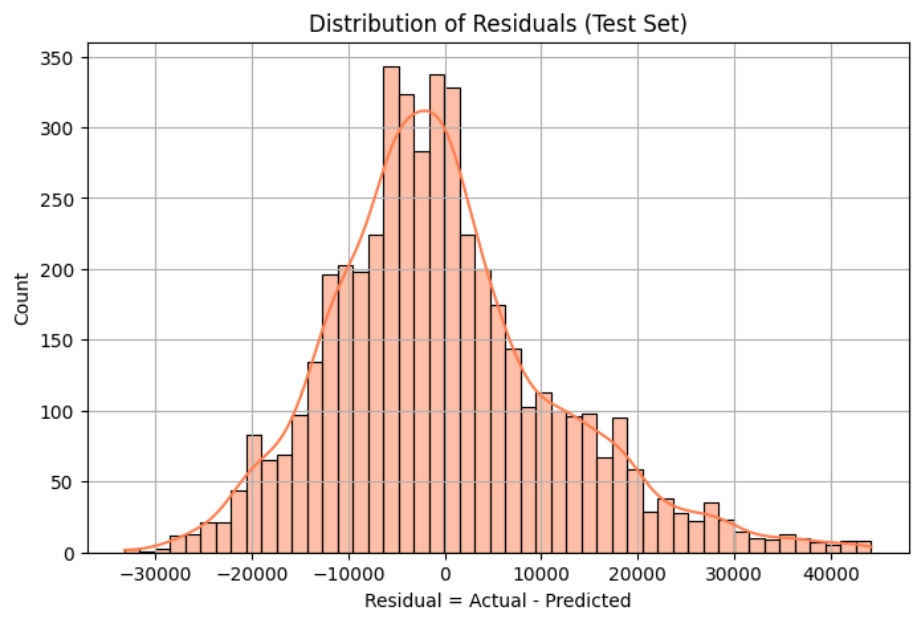
- R^2 Score
- Mean Absolute Error (MAE)

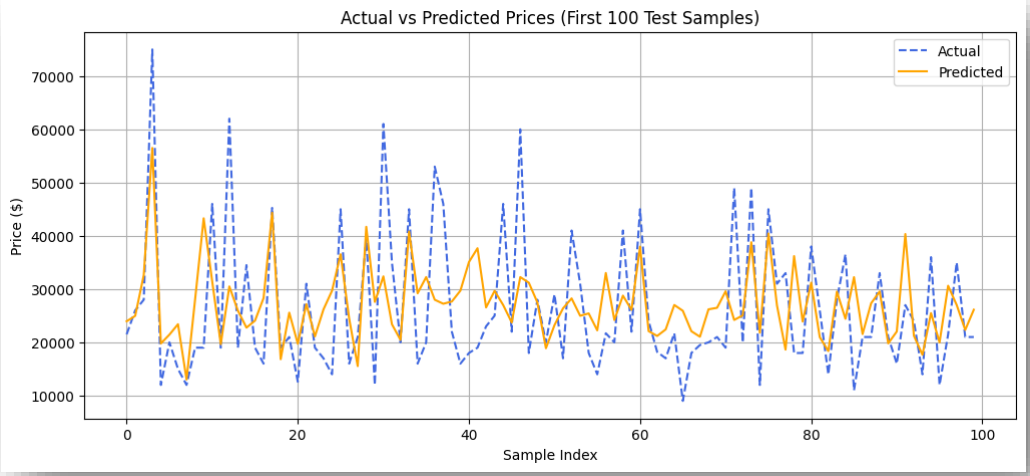
Performance:

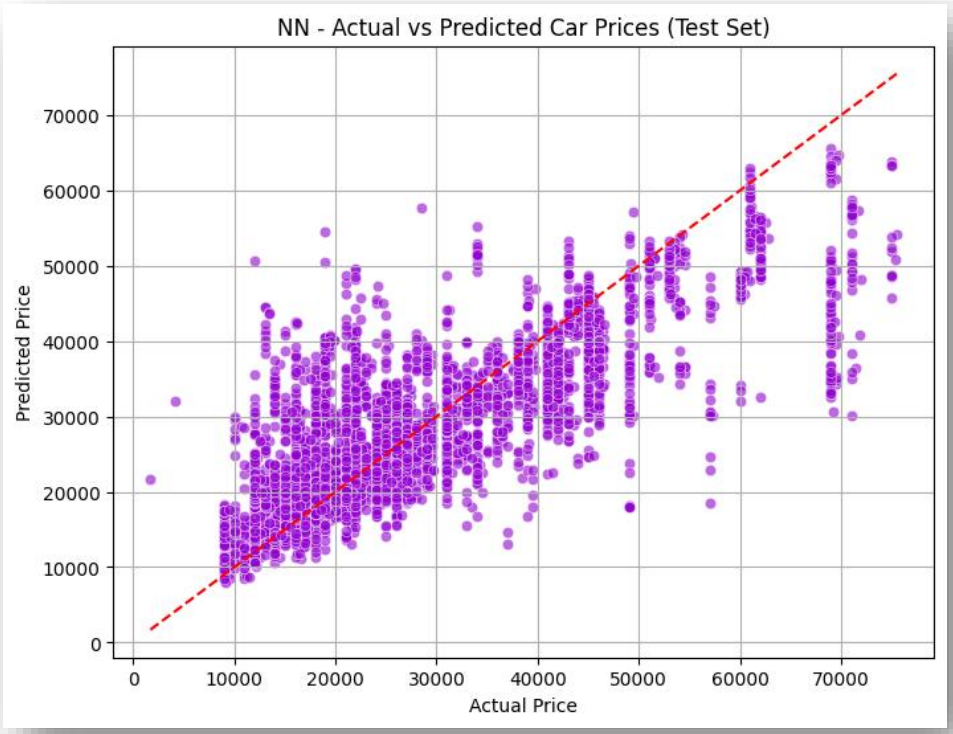
```
 Training Set Evaluation
MAE: 8990.69
RMSE: 11876.50
R²: 0.2705

 Validation Set Evaluation
MAE: 9253.97
RMSE: 12242.53
R²: 0.2264

 Test Set Evaluation
MAE: 9047.68
RMSE: 11872.09
R²: 0.2248
```

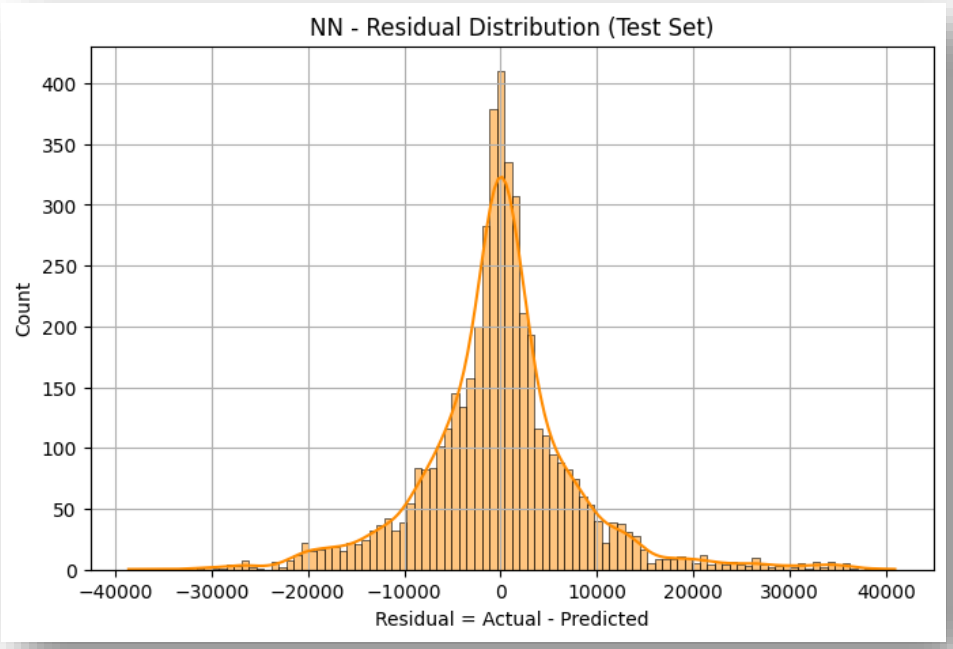

Distribution of Car Prices
Count
Price ($)
0
500
1000
1500
2000
2500
3000
0
20000
40000
60000
80000

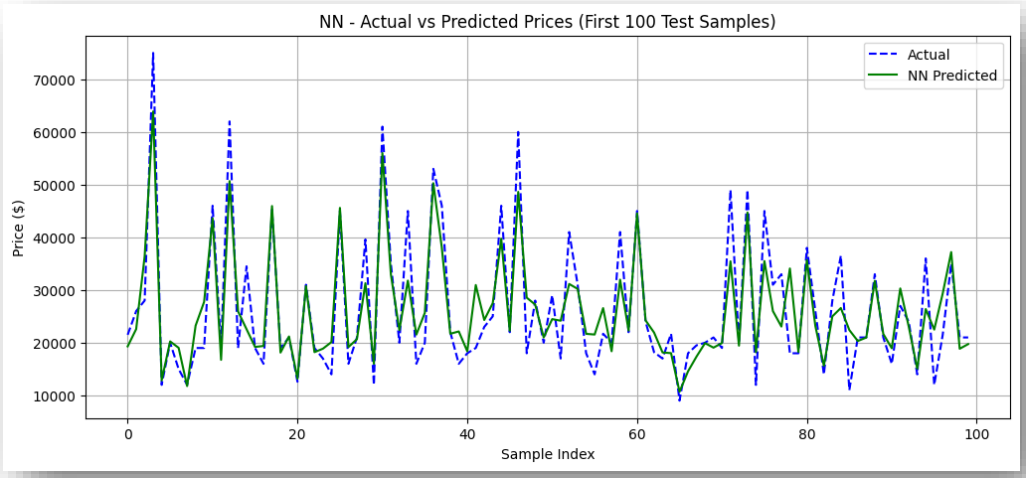

Actual vs Predicted Car Prices (Test Set)
Predicted Price
Actual Price
0
10000
20000
30000
40000
50000
60000
70000

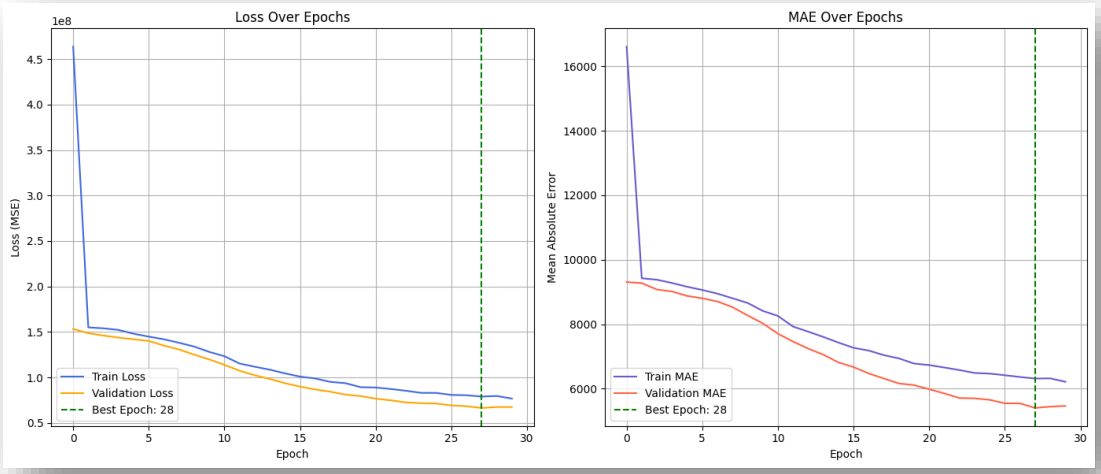

Distribution of Residuals (Test Set)
Count
Residual = Actual - Predicted
350
300
250
200
150
100
50
0
−30000
−20000
−10000
0
10000
20000
30000
40000


Actual vs Predicted Prices (First 100 Test Samples)
Actual
Predicted
Price ($)
Sample Index
70000
60000
50000
40000
30000
20000
10000
0
20
40
60
80
100


NN - Actual vs Predicted Car Prices (Test Set)
Predicted Price
Actual Price
0
10000
20000
30000
40000
50000
60000
70000


NN - Residual Distribution (Test Set)
Count
Residual = Actual - Predicted
0
50
100
150
200
250
300
350
400
−40000
−30000
−20000
−10000
0
10000
20000
30000
40000





---

4. Deployment Architecture

Tools & Services:

- FastAPI
- Docker
- Google Cloud Build
- Google Cloud Run
- Cloud Container Registry

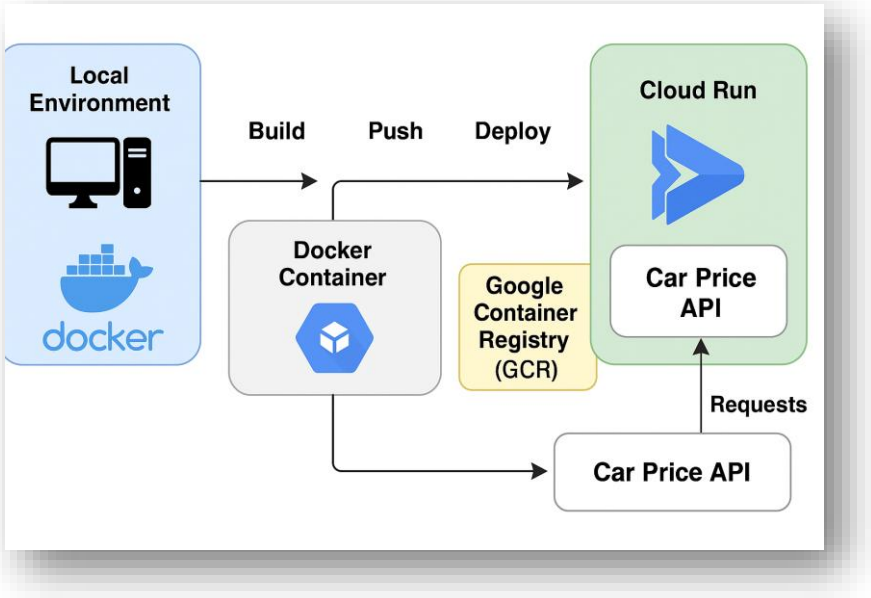
Deployment Steps:

1. Train and pickle model

2. Create FastAPI app in main.py
3. Write Dockerfile
4. Build and push image to GCR
5. Deploy to Cloud Run with memory set to 1Gi

Live API Endpoint:
https://car-price-api-481594874299.us-central1.run.app/docs

Diagram showing architecture (local -> Docker -> GCR -> Cloud Run)



Screenshot of successful deployment confirmation in terminal

```
A:\Seneca\Sem 1\Machine Learning\pROJECT 3\cp_individual>gcloud run deploy car-price-api --image gcr.io/carprice-capston
e-ind/car-price-api --platform managed --region us-central1 --allow-unauthenticated --memory 1Gi
Deploying container to Cloud Run service [car-price-api] in project [carprice-capstone-ind] region [us-central1]
OK Deploying... Done.
  OK Creating Revision...
  OK Routing traffic...
  OK Setting IAM Policy...
Done.
Service [car-price-api] revision [car-price-api-00005-45q] has been deployed and is serving 100 percent of traffic.
Service URL: https://car-price-api-481594874299.us-central1.run.app

A:\Seneca\Sem 1\Machine Learning\pROJECT 3\cp_individual>
```

---

6. Challenges & Solutions

Challenge: Memory limit exceeded 512MiB

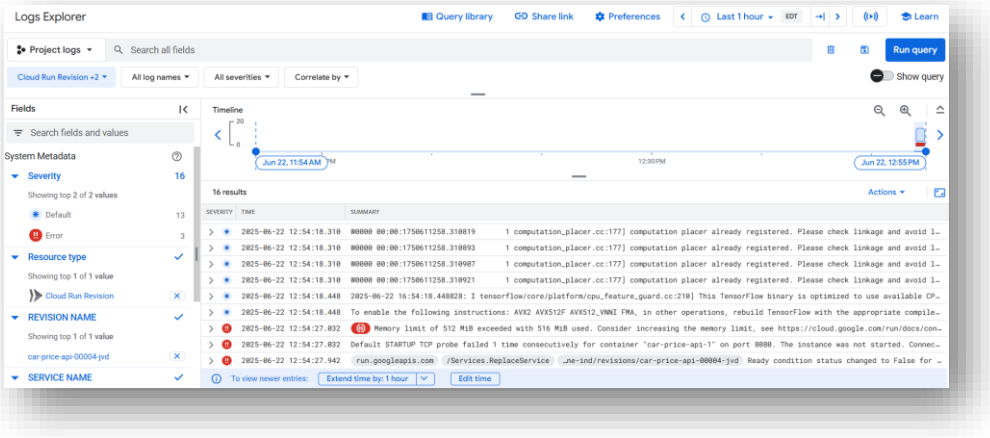
- **Solution:** Increased to –memory 1Gi in deploy command

Challenge: App not listening on port 8080

- **Solution:** Ensured FastAPI uses port=8080 in CMD and EXPOSE in Dockerfile

Challenge: Timeout errors

- **Solution:** Verified logs and updated startupProbe configuration if needed



Error logs or deployment failure messages from Cloud Console

---

7. Conclusion & Future Work

This project demonstrates the ability to train, serve, and deploy an ML model using modern cloud tools. The app is now scalable and available publicly via REST API. Future enhancements include:

- Adding authentication
- UI dashboard for inputs
- Training on additional features like fuel type, mileage, etc.

Working screenshot of API on Swagger UI or Postman.