

# 1. Project Description & Dataset

Saturday, July 5, 2025 12:54 PM

## **Car Price Prediction using FastAPI with deployment over Render and Redis**

### **Motivation:**

- To tie together all concepts learned so far
- Implement an end-to-end project with deployment
- Follow industry-level best practices

### **Dataset:**

- 16 features
- 8 categorical features
- 8 numerical features
- Includes missing values

## 2. Project Structure

Saturday, July 5, 2025 1:03 PM

### project-folder/

- **app/**: Main application package containing all FastAPI app components
  - **\_\_init\_\_.py**: Initializes the Python package for **app**
  - **main.py**: sets up routes, middleware, and monitoring
  - **models/**
    - **model.joblib**: Serialized ML model used for prediction
  - **api/**: Contains route definitions for API endpoints
    - **\_\_init\_\_.py**
    - **routes\_predict.py**: Defines the **/predict** route for price predictions
    - **routes\_auth.py**: Defines the **/login** route for user authentication via JWT
  - **core/**: logic for config, security, dependencies, and exception handling
    - **config.py**: Loads environment variables and app-wide settings
    - **security.py**: Handles JWT creation and verification logic
    - **dependencies.py**: Dependency injection logic for API key and JWT token validation
    - **exceptions.py**: Custom exception handlers for consistent error responses
  - **services/**
    - **model\_service.py**: Loads the ML model and performs predictions (with Redis caching)
  - **middleware/**
    - **logging\_middleware.py**: Logs all incoming requests and outgoing responses
  - **cache/**:
    - **redis\_cache.py**
  - **utils/**: Utility modules for common functionality
    - **logger.py**: Custom logger configuration (optional)
- **notebooks/**: Jupyter notebooks for experimentation
- **data/**:
- **training/**:

- **\_\_init\_\_.py:**
- **train\_utils.py:** common functions to support model training
- **train\_model.py:** model training script
- **requirements.txt:** List of Python dependencies required for the app
- **Dockerfile:** Docker image definition to containerize the FastAPI app
- **docker-compose.yml:** Orchestrates FastAPI, Redis, Prometheus, and Grafana services
- **prometheus.yml:** Monitor FastAPI app using the Prometheus FastAPI Instrumentator
- **render.yaml:** Automates deployment settings of web services over Render
- **.env:** Environment variables used by the application
- **README.md:** Project overview, setup instructions, and usage guide

### 3. Project Setup with GitHub

Saturday, July 5, 2025

1:27 PM

- **Create a remote repository on GitHub**
- **Initialize local git repository**
- **Connect local and remote repositories**
- **Make initial commit**

## 4. Configurations and Security

Saturday, July 5, 2025

1:54 PM

### **Files to setup:**

- **.env**
- **app/core/config.py**
- **app/core/security.py**

## 5. Auth & Dependencies

Saturday, July 5, 2025 2:06 PM

### **Files to setup:**

- **app/core/dependencies.py**
- **app/core/exceptions.py**
- **app/api/routes\_auth.py**

## 6. ML Integration with Caching

Saturday, July 5, 2025 2:13 PM

### **Files to setup:**

- **notebooks/sample.ipynb**
- **training/train\_utils.py**
- **training/train\_model.py**
- **app/cache/redis\_cache.py**
- **app/services/model\_service.py**
- **app/api/routes\_predict.py**

## 7. Middlewares and API

Sunday, July 6, 2025 12:04 PM

### **Files to setup:**

- **app/middleware/logging\_middleware.py**
- **app/main.py**



## 8. Monitoring & Containerization

Sunday, July 6, 2025 12:11 PM

### **Files to setup:**

- **prometheus.yml**
- **Dockerfile**
- **docker-compose.yml**

## 9. Running Locally

Sunday, July 6, 2025 12:24 PM

### Files to setup:

- requirements.txt

### Run the application:

- docker-compose up --build

### Interfaces:

- FastAPI endpoint: <http://localhost:8000/>
- FastAPI metrics: <http://localhost:8000/metrics>
- Prometheus UI: <http://localhost:9090/>
  - Run query: http\_requests\_total
- Grafana: <http://localhost:3000>
  - Username: admin
  - Password: admin

### Configure Grafana:

- Navigate to Grafana -> Add data source -> Prometheus
- Set URL: <http://prometheus:9090>
- Save & Test

- Create dashboards using metrics such as:
  - `http_server_requests_total`
  - `http_request_duration_seconds_bucket`
  - `http_request_duration_seconds_sum`

# 10. Deployment

Sunday, July 6, 2025 2:08 PM

## Files to setup:

- **render.yaml**

## Steps:

- **Visit:** <https://redis.com/try-free/>
- **Sign-in with GitHub/Email**
- **Create a new free database**
- **Copy Redis URL**
- **Update `redis_cache.py`**
- **Visit:** <https://render.com/>
- **Sign-in with GitHub**
- **Add new Web Service**
- **Select repository**
- **Deploy Web Service**