College of computing

Department of Software Enggineering

Fundamentals of ML Assignment documentation

Student Name : Yohannes kidanemariam

ID Number: 1500066

Submitted to: Derbew Felasman(MSc)

Submission Date : 02-06-2017 E

🚗 Car Price Prediction using Machine Learning

📌 Project Overview

The **Car Price Prediction** project aims to estimate the price of a car based on various features like **manufacturing year, mileage, and brand**. This is a **supervised machine learning problem**, where historical car data is used to train a regression model.

The project follows a **structured machine learning pipeline**, from **data collection and preprocessing** to **model training and deployment**. The trained model is deployed using **FastAPI**, allowing real-time price predictions.

This solution is beneficial for:
✅ **Car buyers** – Estimating fair market value for used vehicles.
✅ **Dealerships & sellers** – Setting competitive pricing strategies.
✅ **Automotive analysts** – Studying price trends based on car features.

📂 Folder Structure

📦 Car Price Prediction

```
├── 📁 data/        # Contains dataset files in CSV or JSON format
├── 📁 notebooks/  # Jupyter notebooks for EDA, preprocessing, and model training
├── 📁 src/        # Main Python scripts for training, preprocessing, and prediction
│   ├── preprocess.py  # Data cleaning and feature engineering
│   ├── train.py       # Model training and evaluation
│   ├── predict.py     # Script for making predictions using trained model
├── 📁 api/        # FastAPI-based deployment script
│   ├── app.py         # FastAPI script to serve the model as an API
├── 📄 requirements.txt  # List of required Python dependencies
└── 📄 README.md   # Project description
```

🏗 Project Workflow

This project follows a structured **Machine Learning Pipeline** to ensure data is properly processed, the model is trained effectively, and predictions are accurate.

1️⃣ Data Collection & Preprocessing

- The dataset consists of historical car listings, including **year of manufacture, mileage, and brand**.
- Raw data is processed using preprocess.py, which:
  🔹 **Handles missing values** (e.g., filling missing mileage with the median).
  🔹 **Encodes categorical variables** (e.g., converting car brands into numerical representations).
  🔹 **Scales numerical features** (ensuring features like mileage and year have similar distributions).
  🔹 **Removes outliers** (e.g., unrealistic car prices).

2️⃣ Exploratory Data Analysis (EDA)

Before training the model, **EDA** is performed using **Jupyter notebooks** to:
🔹 Visualize car price distribution.
🔹 Identify correlations between features (e.g., newer cars tend to be more expensive).
🔹 Detect outliers and anomalies.

3️⃣ Model Training

The train.py script trains a **Linear Regression Model**, which is well-suited for numerical price prediction. This script:
🔹 Splits data into **training (80%) and testing (20%) sets**.
🔹 Fits the **Linear Regression** model to learn relationships between features.
🔹 Saves the trained model (model.pkl) and scaler (scaler.pkl) for future use.

Other **alternative models** that can be explored:
✅ **Random Forest Regression** – Handles non-linear relationships better.
✅ **XGBoost** – More powerful for complex data patterns.
✅ **Neural Networks** – Useful for large datasets with high variability.

4️⃣ Model Evaluation

After training, the model is evaluated using key **performance metrics**:
🔹 **Mean Squared Error (MSE)** – Measures average squared prediction error. Lower values indicate better accuracy.
🔹 **R^2 Score** – Determines how well the model explains price variations (closer to 1 is better).

Metric	**Value**
MSE	xxxx.xx

Metric	Value
R^2 Score	0.85 (85% variance explained)

5️⃣ Deployment with FastAPI

Once trained, the model is **deployed using FastAPI**, allowing users to predict car prices dynamically based on input features.

- The app.py script loads the saved model and processes requests.
- A FastAPI server runs locally, receiving inputs and returning predictions.

🔬 Key Insights from Data Analysis

During **Exploratory Data Analysis (EDA)**, several patterns were observed:

✅ **Newer cars** tend to have **higher prices**, but price depreciation is not always linear.
✅ **High mileage** reduces the price significantly, though some brands maintain value better.
✅ **Certain brands** have a consistently **higher price range** due to demand and quality perception.

These insights help refine the model and improve prediction accuracy.

🔥 Features & Enhancements

This project includes **several advanced features** to improve performance and usability:

✅ Data Preprocessing & Feature Engineering

- **Automatic outlier detection** for removing unrealistic prices.
- **Brand encoding** to transform categorical brands into numerical features.

✅ Model Persistence

- Saves trained model and scaler to disk, enabling fast reusability.

✅ Scalable Deployment

- API-based approach allows integration with **web apps or mobile applications**.

✅ **Future Enhancements**
🚀 **Add more predictive features** (e.g., fuel type, transmission, location).
🚀 **Try advanced ML models** (e.g., Random Forest, XGBoost).
🚀 **Deploy the model to the cloud** (e.g., AWS, GCP, or Heroku).

📜 License

This project is licensed under the **MIT License**, allowing open-source contributions and modifications.

✅ **Maintained by:** *Yohannes K/mariyam*
📩 **Contact:** jhonkidan.777@gmail.com