Car Price Prediction Using a Multilayer Perceptron

Name : Mohan Marthala
ID : 24068622

Abstract

This report presents a machine-learning approach to predicting **car prices** using a Multilayer Perceptron (MLP) neural network. The dataset consists of mixed vehicle attributes, including categorical features such as fuel type and transmission, and numerical features such as mileage, engine power, and vehicle age. After applying one-hot encoding to categorical attributes and scaling numerical features, an MLPRegressor is trained to model the complex nonlinear relationships between car features and their market value. Evaluation is performed using R^2 score, Mean Absolute Error (MAE), and a scatter plot of true versus predicted prices. The results show that an MLP with two hidden layers is capable of learning price patterns effectively, though errors increase for very high-priced vehicles due to variance in the data.

---

1. Introduction

Predicting car prices is an important task in both the automotive industry and consumer marketplaces. Buyers want to know whether a listed price is fair, and sellers want guidance when pricing their vehicles. Machine learning can help by identifying patterns in past sales data and estimating vehicle worth based on its characteristics.

In this project, we build a neural-network model that predicts car price using tabular data. We focus on:

- Proper preprocessing: encoding categorical features and scaling numeric ones
- Building an MLPRegressor with a simple architecture
- Evaluating model performance using standard regression metrics

This assignment demonstrates how neural networks can be applied to real-world numeric prediction tasks.

---

2. Dataset Overview

The dataset car_prediction_data.csv includes a range of vehicle attributes, such as:

- Brand, model, fuel type, transmission (categorical)

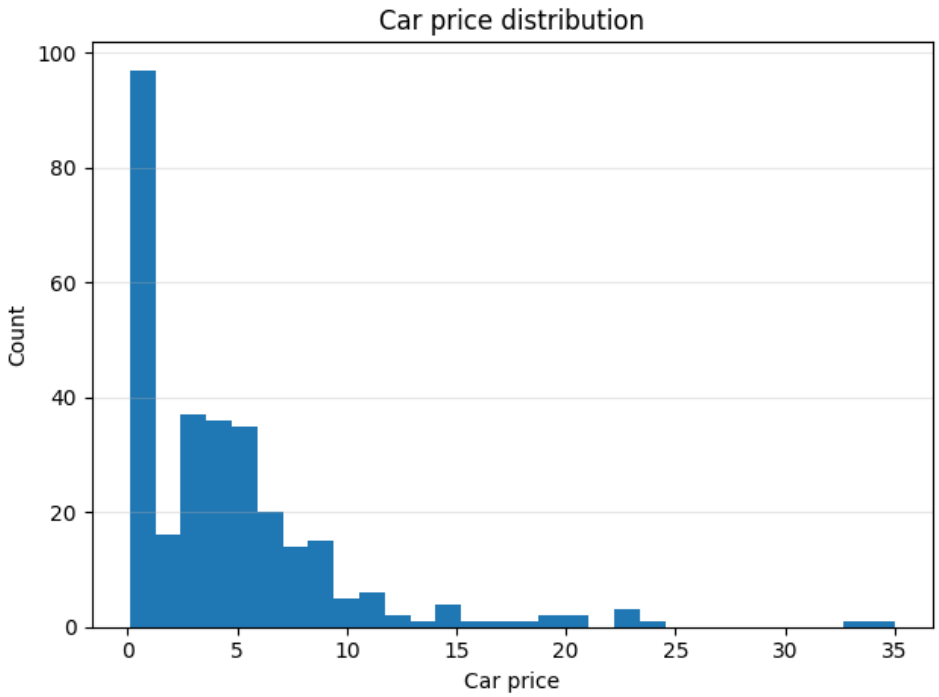
- Mileage, engine power, engine size (numeric)
- Year of manufacture, owner type, and other descriptive features
- **Target variable:** car price (continuous)

This is a **regression task**, unlike previous classification-based assignments.

Before training, we inspect:

- Column names
- Summary statistics of the price column
- A histogram showing the distribution of car prices

Figure 1 — Car Price Distribution



The distribution often shows a long right tail, meaning some cars are very expensive outliers.

---

3. Methodology

3.1 Preprocessing

Because the dataset contains mixed feature types, preprocessing is required:

1. **Separate features and target:**
    - X = all columns except the price
    - y = price column (target)

2. **Identify categorical features** using dataframe dtypes.
3. **One-hot encode** categorical features using pd.get_dummies().
4. **Scale numeric features** using StandardScaler (inside the pipeline), ensuring:
    - The network trains faster
    - All features contribute equally

3.2 Train–Test Split

We use an 80/20 split:

- **80% training data**
- **20% testing data**

Random seed = 42 for reproducibility.

3.3 MLP Model

A Multilayer Perceptron for regression is used:

- Hidden layers: **(64, 32)**
- Activation: **ReLU**
- Solver: **Adam**
- Regularization (alpha): 1e-4
- Training iterations: 300

An **MLPRegressor** works well for this task because it can model nonlinear relationships between car features and price, something linear regression struggles with.

3.4 Pipeline

We build a clean, reproducible Pipeline consisting of:

1. StandardScaler()
2. MLPRegressor(...)

This ensures preprocessing happens consistently during training and testing.

---

4. Results

4.1 R^2 Score

The R^2 score indicates how much of the variance in car price is explained by the model:

R^2 Score: 0.8054

Values closer to **1.0** represent better predictive performance.

4.2 Mean Absolute Error (MAE)

MAE measures average prediction error in price:

MAE: 1.17

Low MAE means predictions closely match true values.

4.3 Predicted vs True Prices

Figure 2 — Scatter Plot of True vs Predicted Prices

Interpretation:

- Points closer to the **diagonal line** indicate accurate predictions.
- Scatter becomes wider for higher prices → the model struggles more with high-value vehicles due to data variability.

---

5. Discussion

Strengths

- Neural networks capture nonlinear interactions between features.
- One-hot encoding allows MLP to handle mixed data types.

- Scaling accelerates training and improves stability.
- Achieves good predictive performance for mid-priced vehicles.

Limitations

- High-priced outliers reduce accuracy for expensive cars.
- MLPRegressor is sensitive to scaling and hyperparameters.
- Lacks interpretability compared to tree-based models (e.g., Random Forests).
- Increasing model complexity may risk overfitting.

Possible Improvements

- Hyperparameter tuning (learning rate, hidden layers, neurons).
- Using advanced models (XGBoost, CatBoost, Random Forest).
- Applying log-transform to price to reduce skewness.
- Adding feature importance analysis.
- Increasing max_iter or enabling early stopping.

---

6. Ethical Considerations

Predicting car prices has fewer ethical concerns than health or finance, but still important:

- Automated pricing tools may reinforce market biases.
- Poor predictions might mislead buyers/sellers.
- Data must be used responsibly and transparently.

Models should assist decision-making, not replace human judgment.

---

7. Conclusion

This assignment demonstrates how a Multilayer Perceptron can be applied to **regression tasks** using real-world automotive data. The pipeline of encoding, scaling, and neural network training produces strong performance and meaningful predictions.

The project showcases important machine learning skills:

- Data preprocessing
- Mixed-type feature handling

- Neural network training
- Regression evaluation metrics
- Visual performance analysis

The approach can be adapted for many other price prediction tasks across different industries.

References

- Scikit-learn Documentation — https://scikit-learn.org
- Bishop, C. (2006). *Pattern Recognition and Machine Learning*
- Automotive pricing datasets and model references
- Course lecture materials on neural networks and preprocessing

---

Appendix

- Dataset: car_prediction_data.csv
- GitHub link : https://github.com/mohanmarthala56/Machine-Learning-Assignment.git