Car Price Prediction Using Linear Regression

1. Introduction

This project aims to build a machine learning model to predict car prices using the Linear Regression algorithm. The dataset used is the "Car Price Prediction" dataset from Kaggle, which includes various technical specifications of cars. The primary goal is to explore the data, process it effectively, and train a predictive model for the price column.

---

2. Dataset Overview

The dataset contains various features including car make, model, engine type, fuel system, number of doors, horsepower, etc. The **price** column is the target variable.

- **Number of rows:** 205
- **Number of columns:** 26
- **Target Variable:** price
- **Input Variables:** Combination of categorical and numerical features

---

3. Exploratory Data Analysis (EDA)

EDA was performed to understand the structure, distribution, and relationships in the data.

3.1 Null Values Check

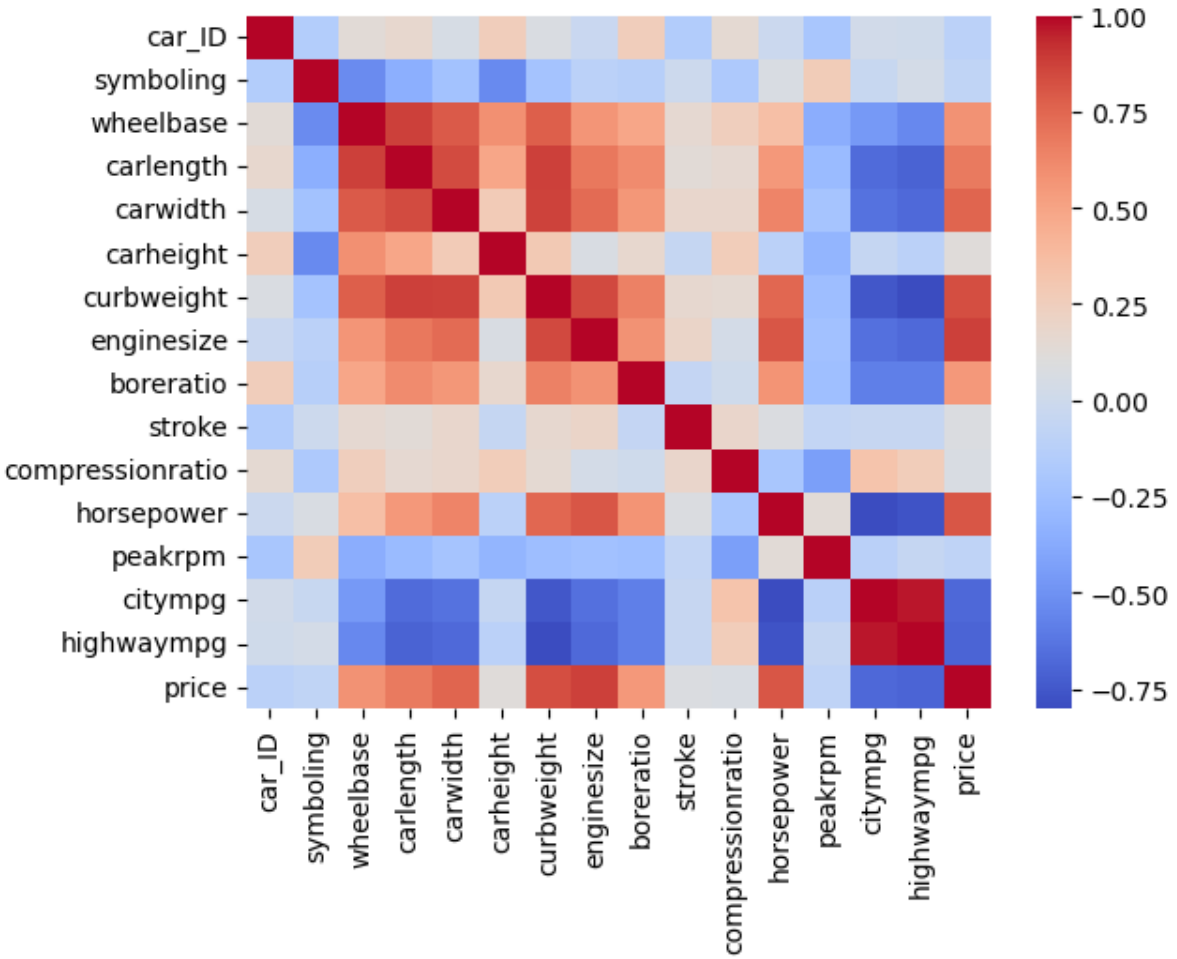
- Checked for missing or null values using df.isnull().sum()
- Dataset had no missing values

3.2 Descriptive Statistics

- Summary statistics were reviewed using df.describe()
- This helped in understanding the spread and range of numerical features

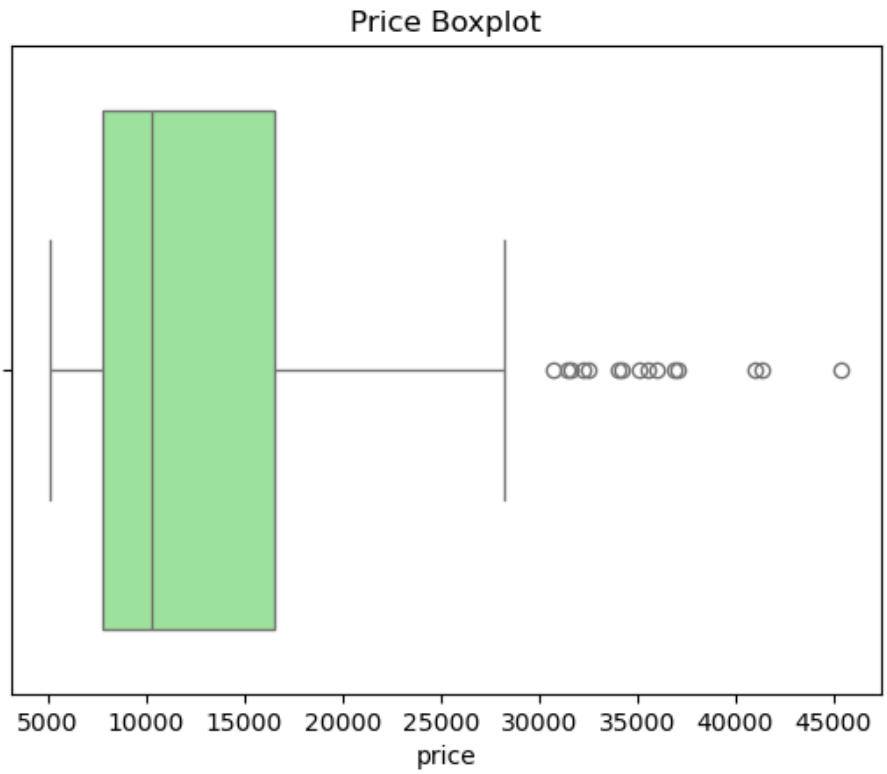
3.3 Correlation Heatmap

- A correlation heatmap was plotted using Seaborn to examine relationships between numerical variables.



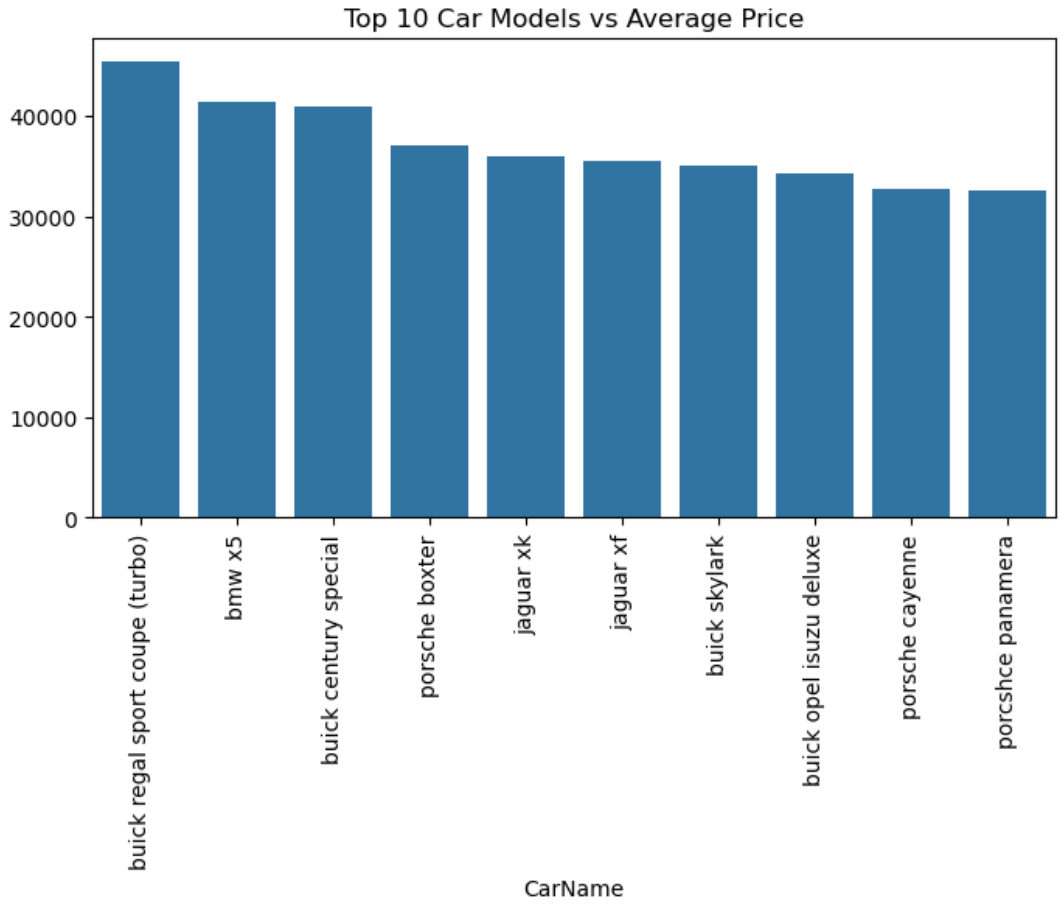
3.4 Price Distribution

- Histogram and boxplot were used to visualize the distribution and outliers in car prices.



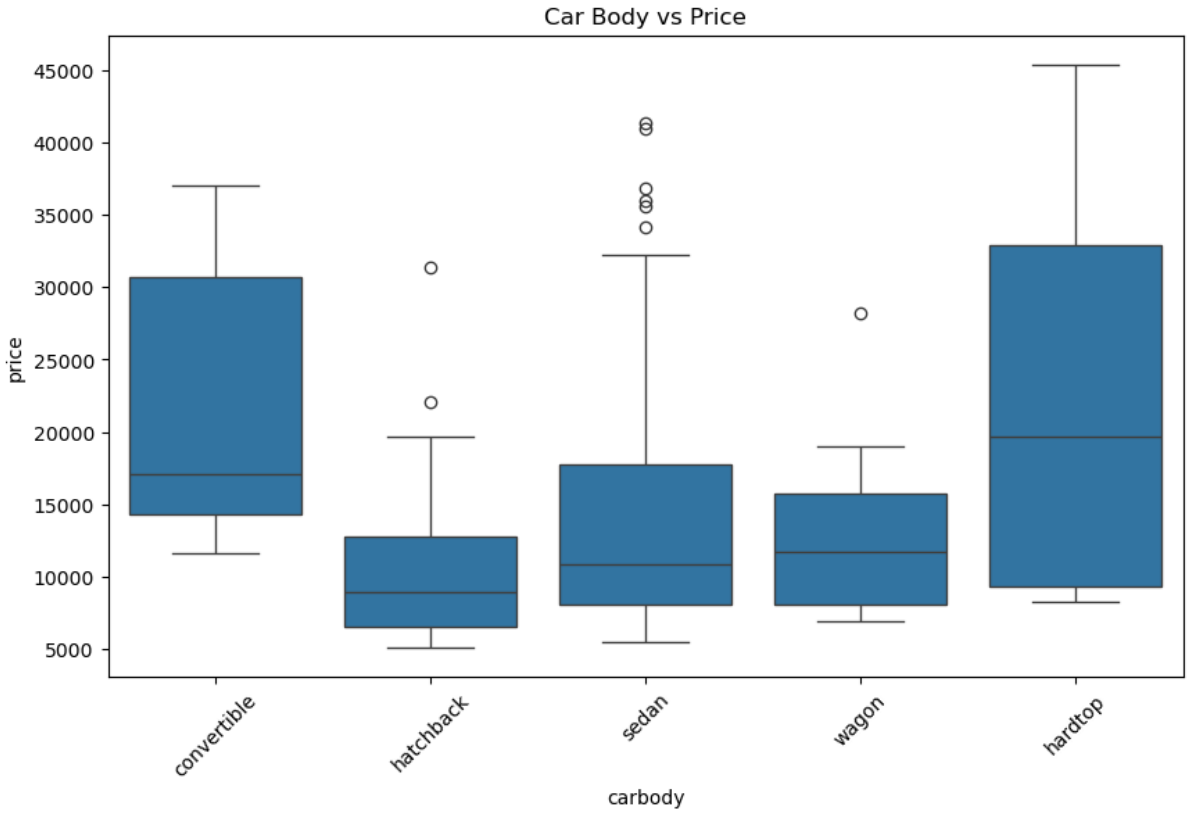
3.5 Car Models vs Price

- The average price for top 10 car models was visualized using a bar plot.



3.6 Car Body Type vs Price

- Explored how car body types affect pricing using bar plots or box plots.



---

4. Data Preprocessing

Several preprocessing steps were applied to clean and prepare the data:

- **Categorical Encoding:**
  - Categorical columns such as fueltype, aspiration, carbody, drivewheel, etc., were encoded using one-hot encoding.
  - CarName was also encoded after extracting the brand name (e.g., from “mazda rx-7 gs” to “mazda”).
- **Feature Scaling:**
  - Numerical features were scaled using StandardScaler to normalize the values.
- **Final Features Used for Training:**
  - All numeric columns
  - One-hot encoded categorical columns

---

5. Linear Regression Algorithm

Linear Regression is a supervised learning algorithm used for regression tasks. It models the relationship between the dependent variable (target) and one or more independent variables (features) by fitting a linear equation:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \beta_2 x_2 + \ldots + \beta_n x_n$$

The model learns the best values for the coefficients by minimizing the difference between the predicted values and actual values using a cost function, typically **Mean Squared Error (MSE)**.

---

6. Model Training

A **Linear Regression** model from sklearn.linear_model was used:

- The dataset was split into **train** and **test** sets (80-20 ratio).
- Model was trained using .fit() on the training data.
- Predictions were made on the test set using .predict().

---

7. Model Evaluation

The model was evaluated using the following metrics:

- **MAE** (Mean Absolute Error)
- **MSE** (Mean Squared Error)
- **RMSE** (Root Mean Squared Error)
- **R^2 Score** (Coefficient of Determination)

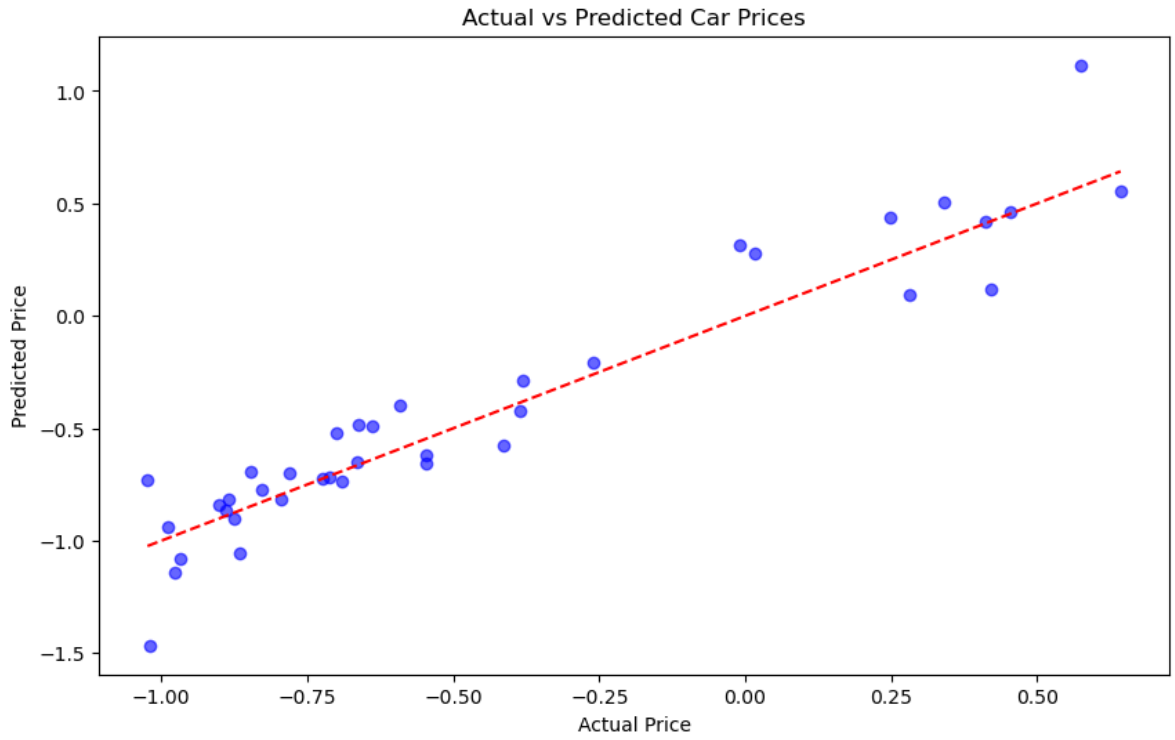
```
...  Mean Absolute Error: 0.1346890902761289
     Mean Squared Error: 0.0329787060391422
     Root Mean Squared Error: 0.18160040208970407
     R-squared: 0.8753985802018274
```

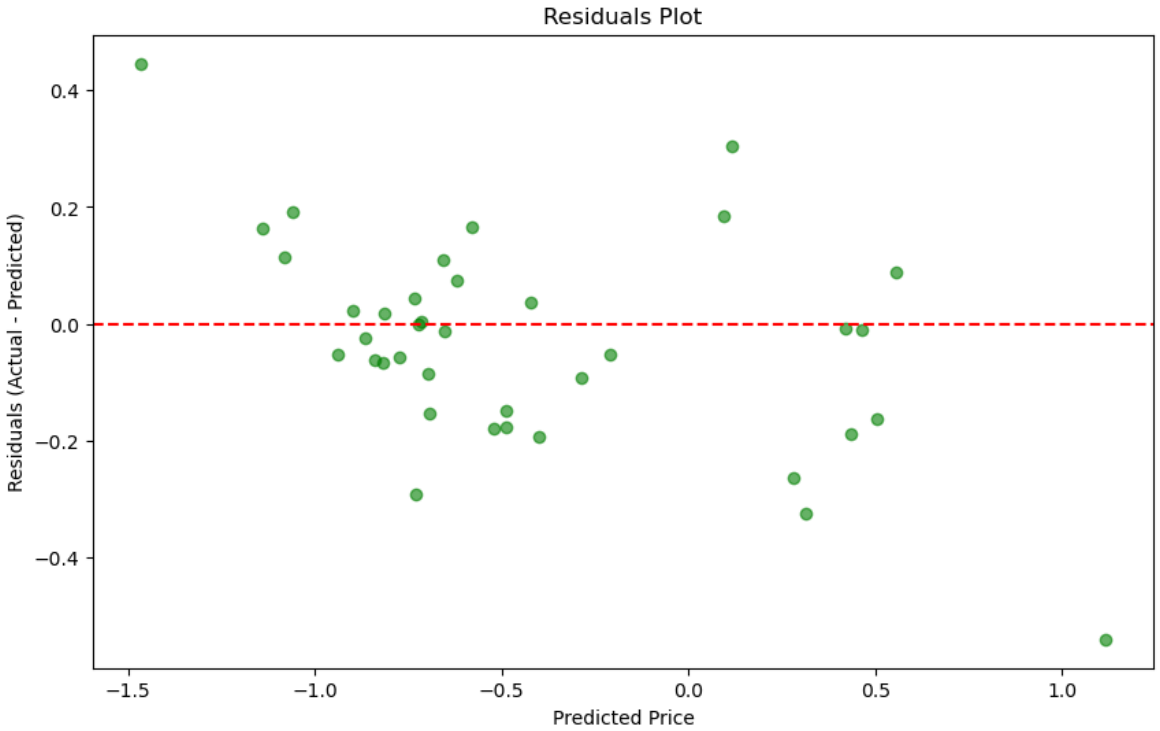
---

8. Visualizations

Additional visualizations include:

- Actual vs Predicted Price Plot
- Residual Plot





---

9. Conclusion

- A Linear Regression model was successfully trained to predict car prices.
- The model learned from a combination of numerical and encoded categorical features.
- The results show that while linear regression can capture basic patterns, improvements like more complex models or deeper feature engineering may yield better performance.