Vehicle Price Prediction Report

1. Introduction

The objective of this project is to predict vehicle prices using various machine learning models based on historical data. As the automotive market evolves, understanding vehicle pricing becomes increasingly important for both buyers and sellers. Accurate predictions can help consumers make informed purchasing decisions and assist sellers in setting competitive prices. This project employs a dataset that encompasses various features related to vehicle specifications and market factors.

2. Data Description

The dataset utilized for this project consists of two primary files: train-data.csv and test-data.csv, both of which provide valuable information regarding the vehicles.

Train Dataset

The training dataset contains various features that describe each vehicle, including:

- **Name**: The model name of the vehicle, which may include brand and model variations.
- **Location**: The city where the vehicle is located, which can affect its market price.
- **Fuel_Type**: Type of fuel used (e.g., Petrol, Diesel, CNG, LPG, Electric).
- **Transmission**: Transmission type (Automatic or Manual).
- **Owner_Type**: Type of ownership (First, Second, etc.).
- **Mileage**: Fuel efficiency of the vehicle, generally measured in kilometers per liter (kmpl).
- **Engine**: Engine capacity, typically expressed in liters or cubic centimeters (cc).
- **Power**: Engine power, usually measured in horsepower (hp) or kilowatts (kW).
- **Seats**: Number of seats in the vehicle, indicating its size and usability.
- **Price**: Target variable representing the vehicle's price, which is crucial for prediction.

Test Dataset

The test dataset has a similar structure but does not include the target variable (Price), allowing us to make predictions based on the learned model.

3. Data Preprocessing

Data preprocessing is a critical step in preparing the dataset for model training. The following subsections detail the steps taken to clean, normalize, and prepare the data for analysis.

3.1 Data Cleaning

- **Loading the Data**: The data was loaded using the pandas library, and unnecessary index columns were dropped for clarity.

- **Handling Missing Values**: The New_Price column had a significant amount of missing data. Therefore, it was decided to remove the entire column to prevent any bias in the model due to incomplete information.

3.2 Categorical Encoding

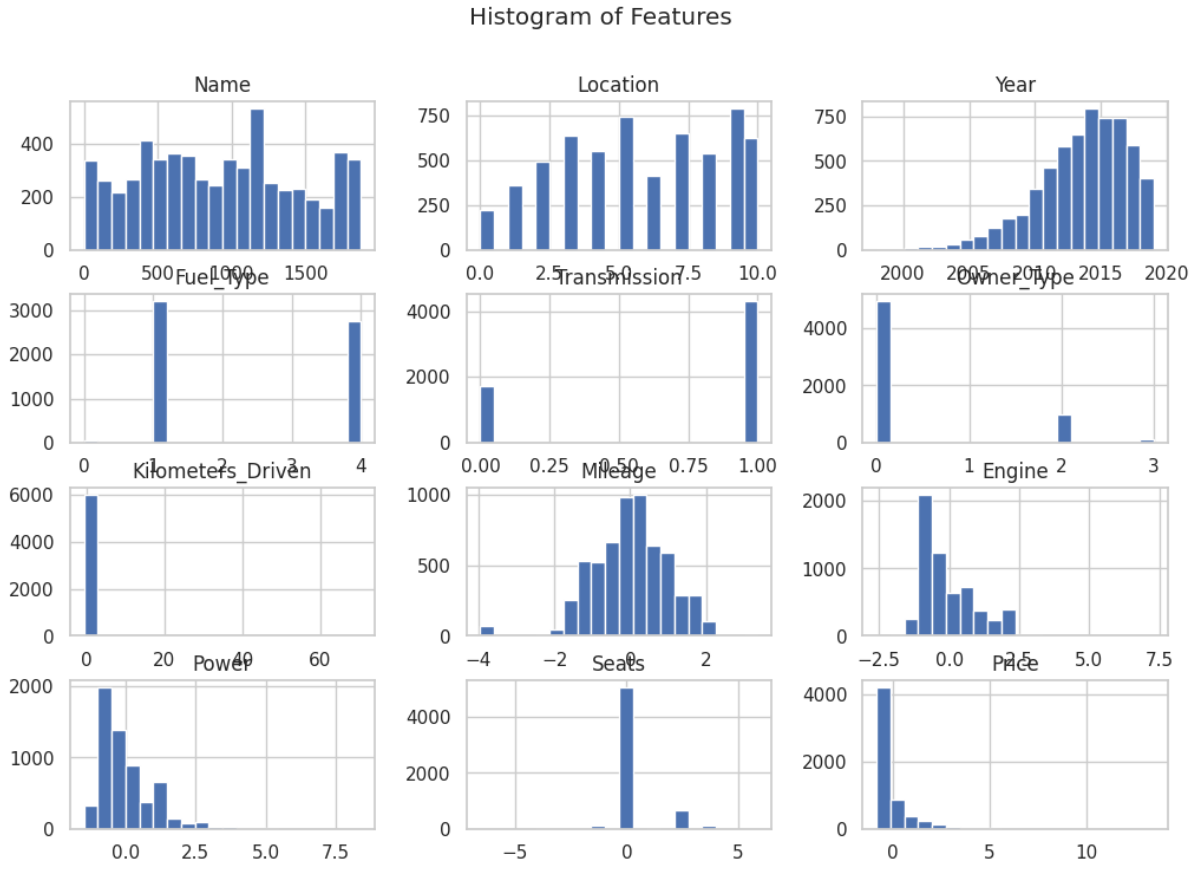
- **Numerical Replacement**: String features, such as Name, Location, Fuel_Type, Transmission, and Owner_Type, were replaced with their numerical equivalents. This was done to facilitate model training, as machine learning algorithms require numerical input.
- **Removal of Temporary Columns**: After replacing the original columns with numerical equivalents, the intermediate string columns were removed to maintain a clean dataset.

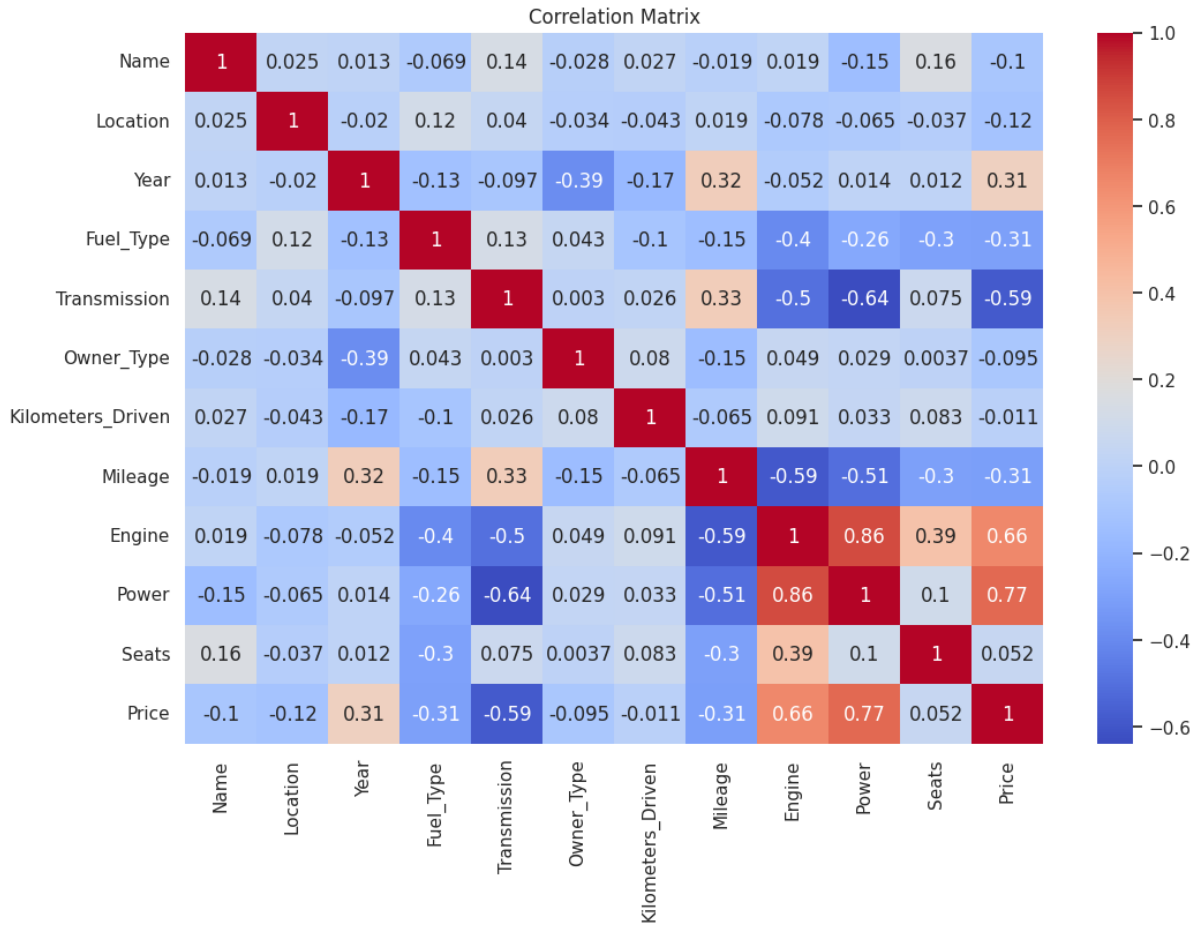
3.3 Unit Conversion

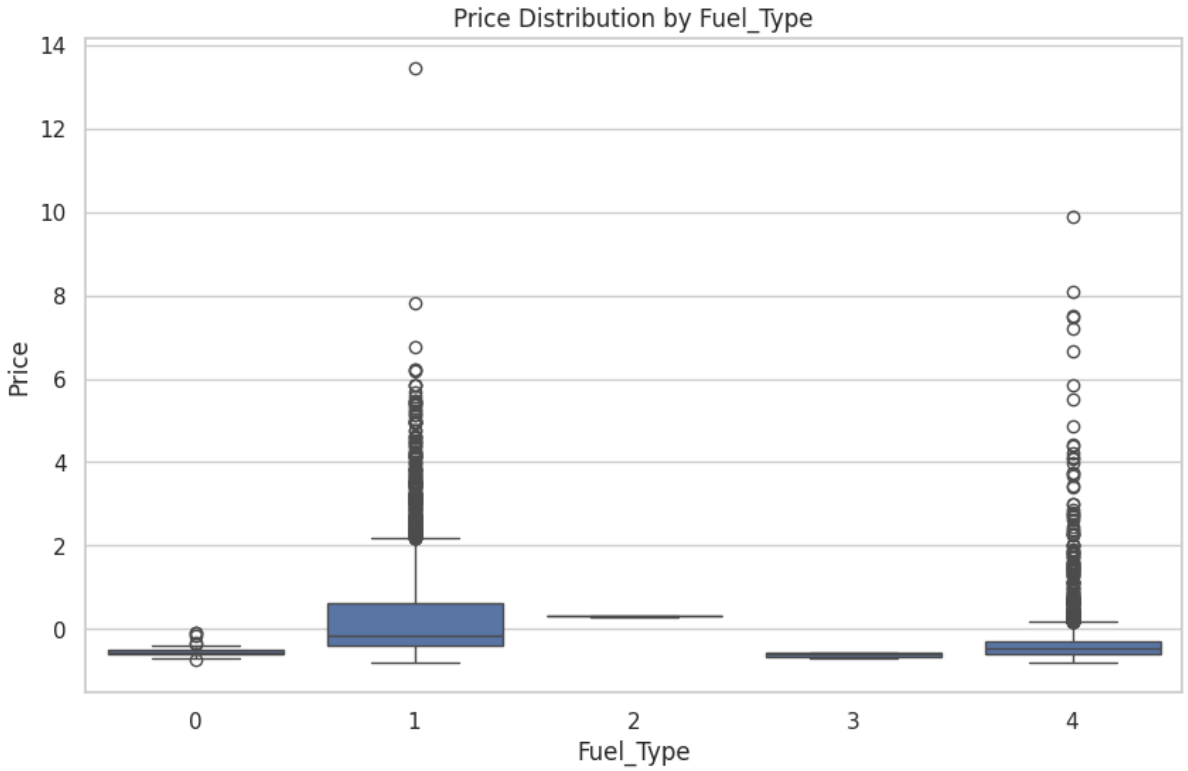
- **Converting Units**: To ensure all data was in numerical format, string values in the Engine, Power, and Mileage columns were cleaned by replacing units with an empty string. The resultant values were then cast to a numeric data type.
- **Fuel Type Consideration**: A major issue encountered was the need to convert fuel efficiency measurements from kilometers per liter (kmpl) into kilometers per kilogram (km/kg). The dataset includes five fuel types: CNG, Petrol, Diesel, LPG, and Electric, necessitating careful consideration of how each fuel type impacts mileage and pricing.

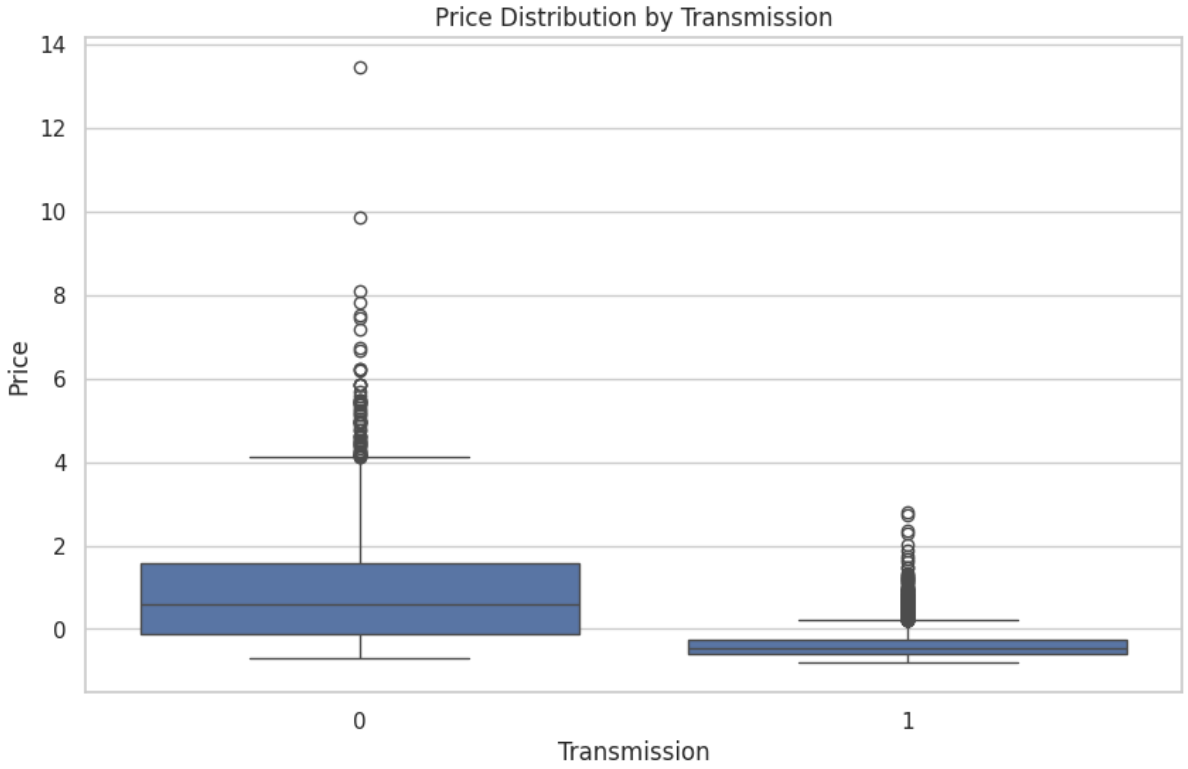
3.4 Data Normalization

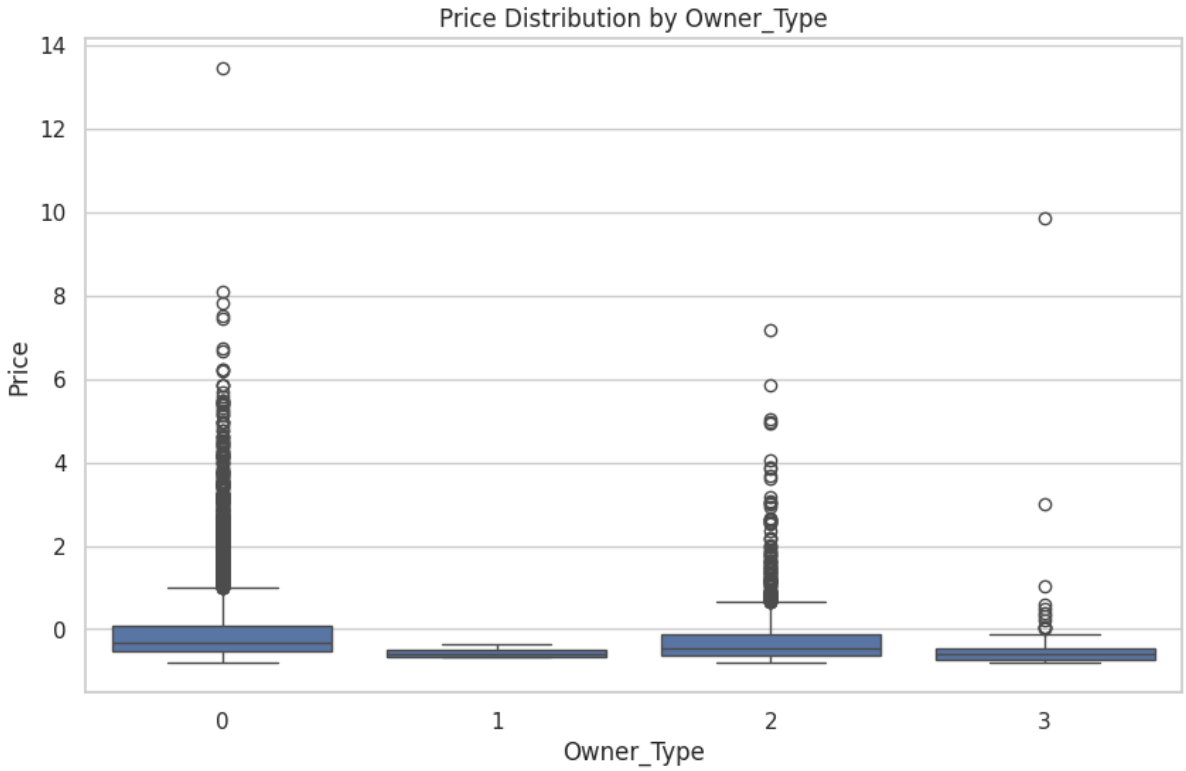
- **Normalization Process**: To ensure that features converted from categorical to numerical format were not normalized, a list of original numerical columns was created. This list guided the normalization process to ensure only relevant columns were scaled.
- **Standard Scaling**: StandardScaler was applied to the identified numerical columns to normalize the data, improving model performance by ensuring equal contribution from each feature in distance calculations.
- **Reorganizing Data**: After standardization, the resulting dataset had the Price column not at the end, as is conventional. The dataframe was reorganized to place the Price column at the end for consistency and easier interpretation.

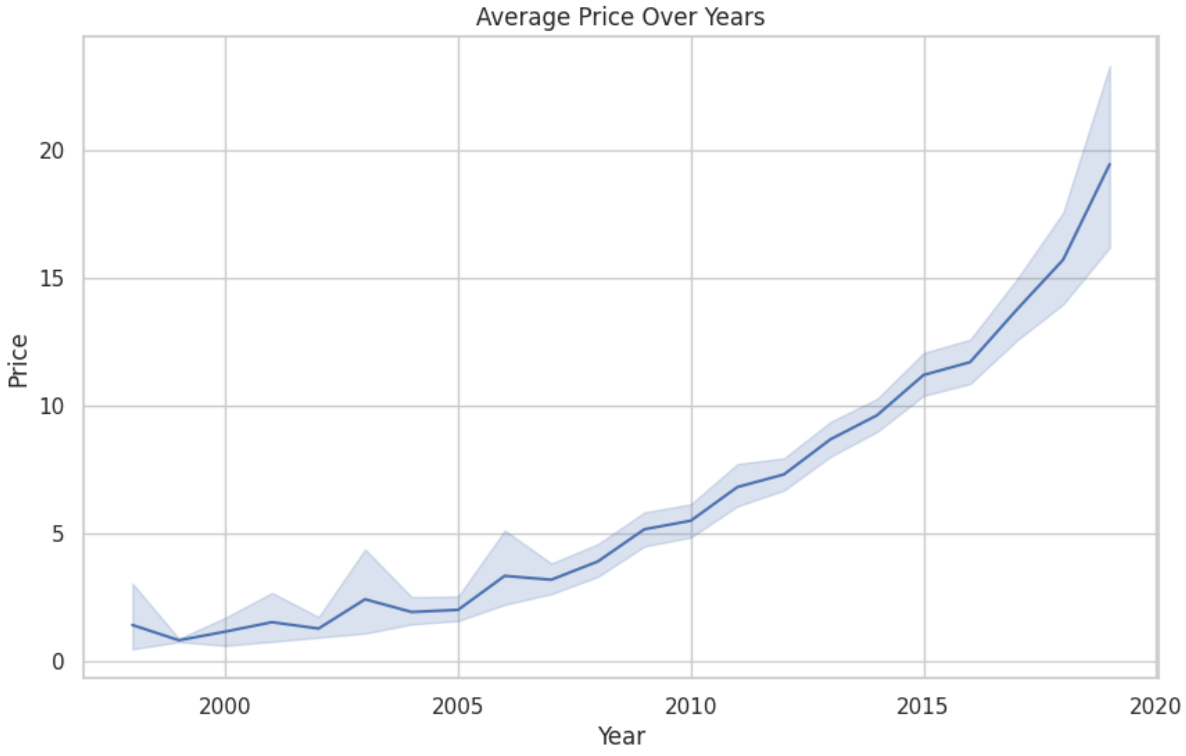

Histogram of Features
Name
Location
Year
Fuel_Type
Transmission
Owner_Type
Kilometers_Driven
Mileage
Engine
Power
Seats
Price


Correlation Matrix
Name 1 0.025 0.013 -0.069 0.14 -0.028 0.027 -0.019 0.019 -0.15 0.16 -0.1
Location 0.025 1 -0.02 0.12 0.04 -0.034 -0.043 0.019 -0.078 -0.065 -0.037 -0.12
Year 0.013 -0.02 1 -0.13 -0.097 -0.39 -0.17 0.32 -0.052 0.014 0.012 0.31
Fuel_Type -0.069 0.12 -0.13 1 0.13 0.043 -0.1 -0.15 -0.4 -0.26 -0.3 -0.31
Transmission 0.14 0.04 -0.097 0.13 1 0.003 0.026 0.33 -0.5 -0.64 0.075 -0.59
Owner_Type -0.028 -0.034 -0.39 0.043 0.003 1 0.08 -0.15 0.049 0.029 0.0037 -0.095
Kilometers_Driven 0.027 -0.043 -0.17 -0.1 0.026 0.08 1 -0.065 0.091 0.033 0.083 -0.011
Mileage -0.019 0.019 0.32 -0.15 0.33 -0.15 -0.065 1 -0.59 -0.51 -0.3 -0.31
Engine 0.019 -0.078 -0.052 -0.4 -0.5 0.049 0.091 -0.59 1 0.86 0.39 0.66
Power -0.15 -0.065 0.014 -0.26 -0.64 0.029 0.033 -0.51 0.86 1 0.1 0.77
Seats 0.16 -0.037 0.012 -0.3 0.075 0.0037 0.083 -0.3 0.39 0.1 1 0.052
Price -0.1 -0.12 0.31 -0.31 -0.59 -0.095 -0.011 -0.31 0.66 0.77 0.052 1
Name Location Year Fuel_Type Transmission Owner_Type Kilometers_Driven Mileage Engine Power Seats Price


Price Distribution by Fuel_Type
Price
Fuel_Type
0
1
2
3
4
0
2
4
6
8
10
12
14


Price Distribution by Transmission
Price
Transmission
0
1
0
2
4
6
8
10
12
14





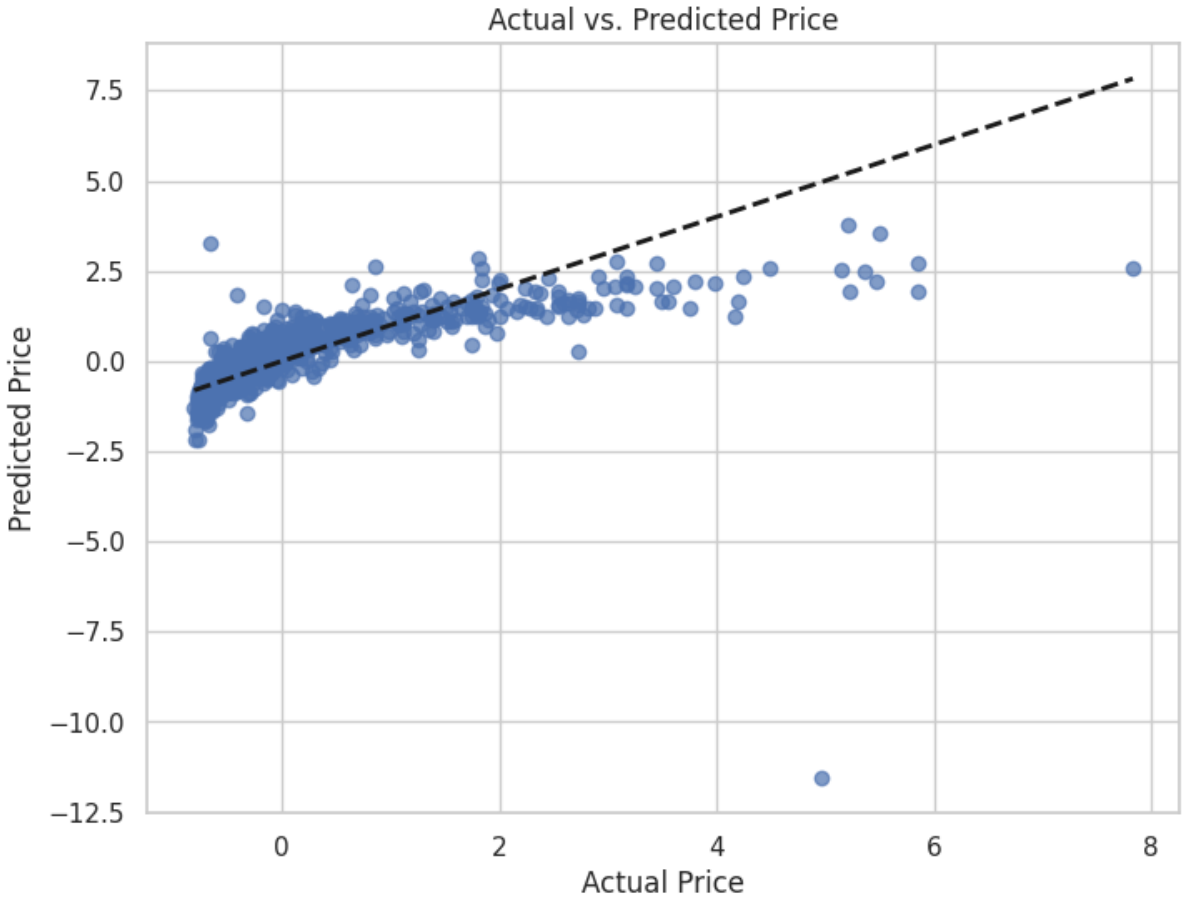
4. Model Development

Multiple regression models were evaluated to determine the most effective approach for predicting vehicle prices.

4.1 Model Selection

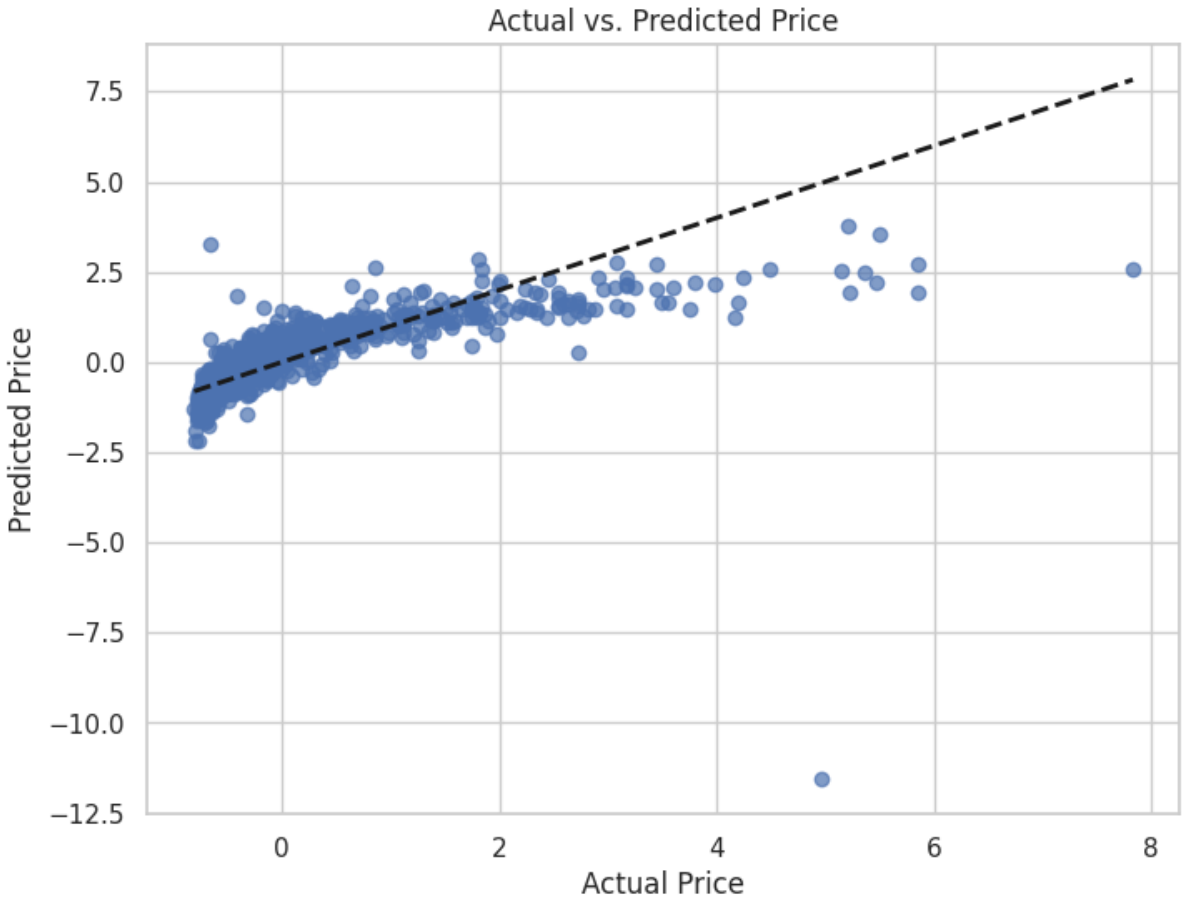
The following models were explored:

- **Linear Regression**: This served as an initial baseline model for comparison with more complex algorithms.
    - **Testing Mean Squared Error**: 0.5309
    - **Testing R^2 Score**: 0.4601



- **K-Nearest Neighbors (KNN)**: A weighted KNN regressor was implemented, with various values of K tested through cross-validation. The process involved iterating through possible K values and printing their Mean Squared Error (MSE) values. The K value that yielded the lowest MSE was deemed the most suitable. The MSE values for different K values are as follows:
    - k=1: MSE = 0.1377
    - k=2: MSE = 0.1343
    - k=3: MSE = 0.1260 (best performing)
    - k=4: MSE = 0.1325
    - k=5: MSE = 0.1351
    - k=6: MSE = 0.1303
    - k=7: MSE = 0.1289

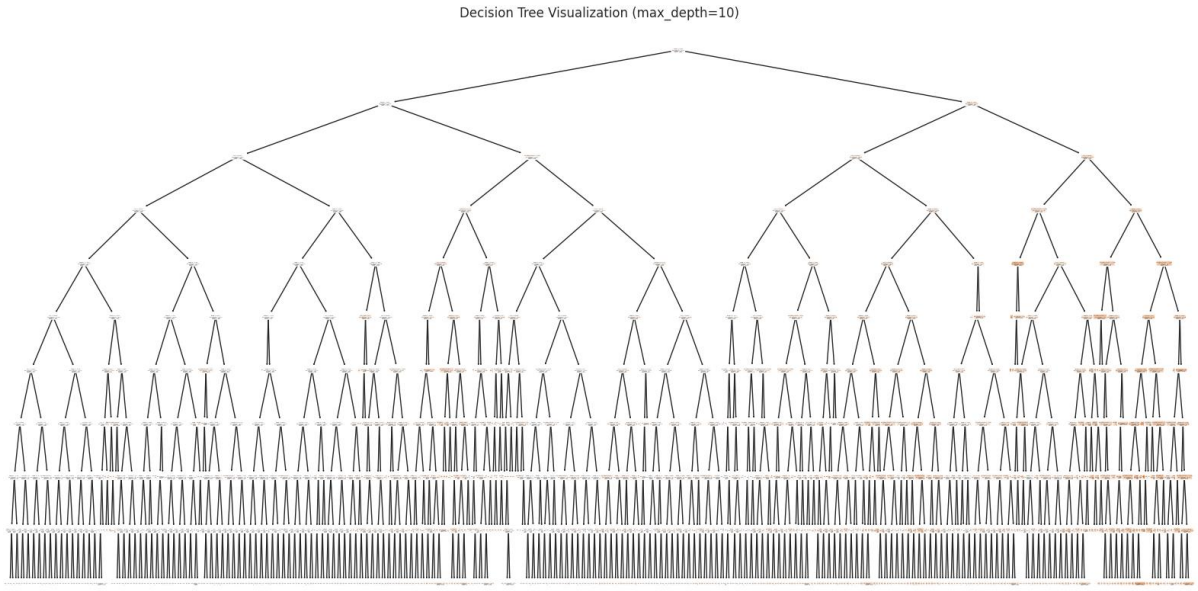
- k=8: MSE = 0.1330
- k=9: MSE = 0.1362
- k=10: MSE = 0.1355
- k=11: MSE = 0.1380
- k=12: MSE = 0.1430
- k=13: MSE = 0.1430
- k=14: MSE = 0.1483
- k=15: MSE = 0.1518
- k=16: MSE = 0.1564
- k=17: MSE = 0.1606
- k=18: MSE = 0.1669
- k=19: MSE = 0.1707
- k=20: MSE = 0.1753



- **Support Vector Regression (SVR)**: This was tested using both polynomial and radial basis function (RBF) kernels.

- Using SVR with polynomial kernel (degree=3):
  - **Testing Mean Squared Error**: 0.9641
  - **Testing R^2 Score**: 0.0195
- Using SVR with RBF kernel:
  - **Testing Mean Squared Error**: 0.9620
  - **Testing R^2 Score**: 0.0217

- **Decision Tree Regression**: Different maximum depths for the decision tree were explored:
  - Max Depth: None
    - **Testing Mean Squared Error**: 0.1909
    - **Testing R^2 Score**: 0.8058
  - Max Depth: 3
    - **Testing Mean Squared Error**: 0.2767
    - **Testing R^2 Score**: 0.7186
  - Max Depth: 5
    - **Testing Mean Squared Error**: 0.2040
    - **Testing R^2 Score**: 0.7925
  - Max Depth: 10
    - **Testing Mean Squared Error**: 0.1610
    - **Testing R^2 Score**: 0.8363 **(BEST PERFORMING)**
  - Max Depth: 15
    - **Testing Mean Squared Error**: 0.1792
    - **Testing R^2 Score**: 0.8178
  - Max Depth: 20
    - **Testing Mean Squared Error**: 0.1832
    - **Testing R^2 Score**: 0.8137
  - Max Depth: 25
    - **Testing Mean Squared Error**: 0.1914
    - **Testing R^2 Score**: 0.8054



- **Random Forest Regression**: This ensemble method utilized multiple decision trees to enhance prediction accuracy.
    - **Testing Mean Squared Error**: 0.1144
    - **Testing R^2 Score**: 0.8836

Despite increasing the number of estimators in Random Forest by 10 times, the improvement in performance was not significant. However, the computational time increased substantially, indicating diminishing returns on performance improvements with excessive model complexity.

5. Model Evaluation

5.1 Performance Metrics

For each model, performance was assessed using the following metrics:

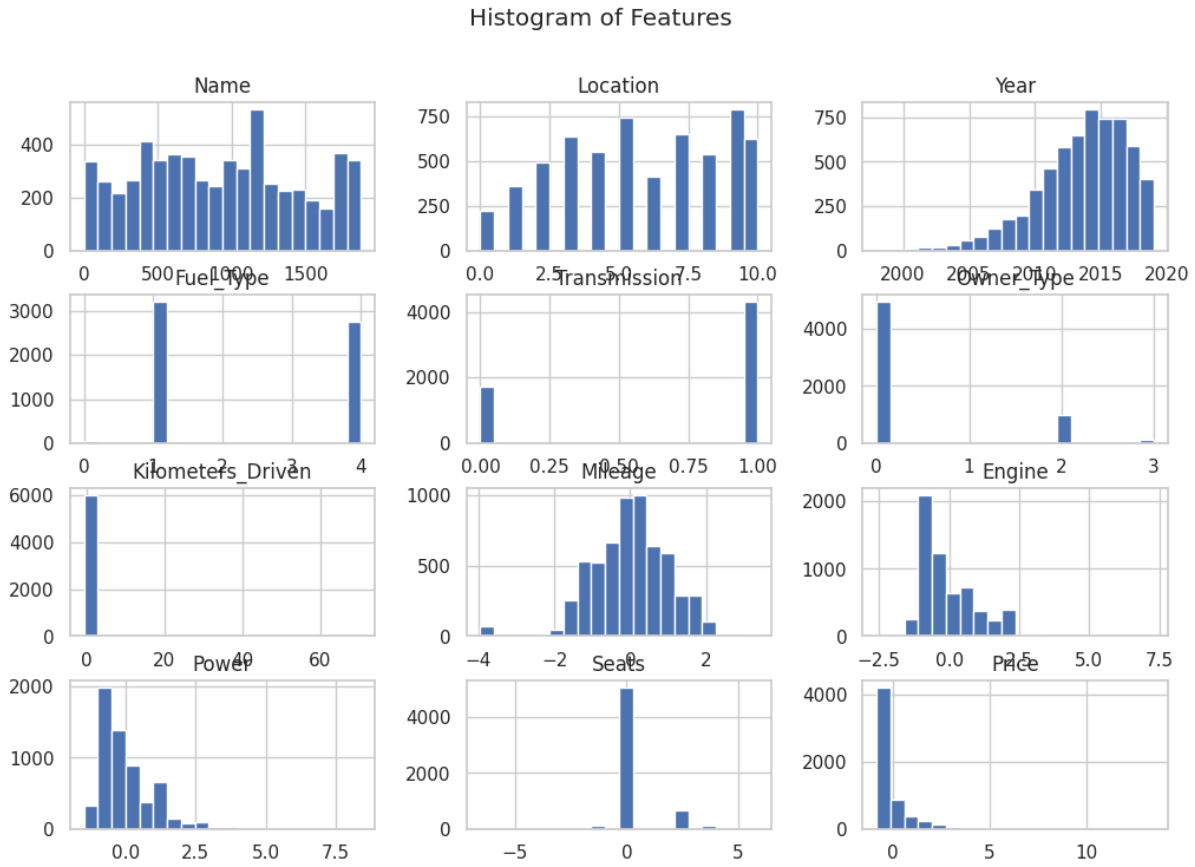
- **Mean Squared Error (MSE)**: A measure of the average squared difference between predicted and actual values.
- **R^2 Score**: This indicates the proportion of variance in the dependent variable (Price) that can be explained by the independent variables (features).

5.2 Results Summary

Model	**Mean Squared Error**	**R^2 Score**
Linear Regression	0.5309	0.4601
KNN (K=3)	0.1260	Highest Value
Support Vector Regression	0.9641 (poly)	0.0195
Support Vector Regression	0.9620 (RBF)	0.0217

Model	Mean Squared Error	R^2 Score
Decision Tree Regression (Max Depth: 10)	0.1610	0.8363
Random Forest Regression	0.1144	0.8836

The KNN model with K=3 demonstrated the best performance among all evaluated models, achieving the lowest MSE and highest R^2 score.



6. Final Predictions

Using the selected KNN model, predictions were made for the test dataset. The predicted prices were converted back to their original scale using the mean and standard deviation derived from the training data.

6.1 Results in Test Data

The final test data was enriched with predicted prices and analyzed to provide insights into the expected market prices of vehicles based on their features. This information can aid potential buyers in assessing the value of vehicles based on their specifications.

7. Conclusion

The project successfully developed a robust model for predicting vehicle prices using KNN, which outperformed other models. The data preprocessing steps, particularly handling of categorical features and normalization, were crucial in improving model performance.