ICS 474 PTOJECT

PREDICT CAR PRICES MODEL

NAMES:
AHMED ALZAKARI
ABDULLAH ALAMOUDI

INTRODUCTION

This project aims to develop a machine learning model that accurately predicts used car prices based on key vehicle features. By analyzing a comprehensive dataset and applying advanced regression techniques, we can provide valuable insights for buyers and sellers in the automotive marke

PART 1: DATA UNDERSTANDING AND EXPLORATION

Dataset Overview

Our analysis utilizes the "Car Details Dataset" from Kaggle, which contains comprehensive information about used cars in the automotive resale market. The dataset captures essential valuation factors including age, mileage, fuel type, and ownership history.

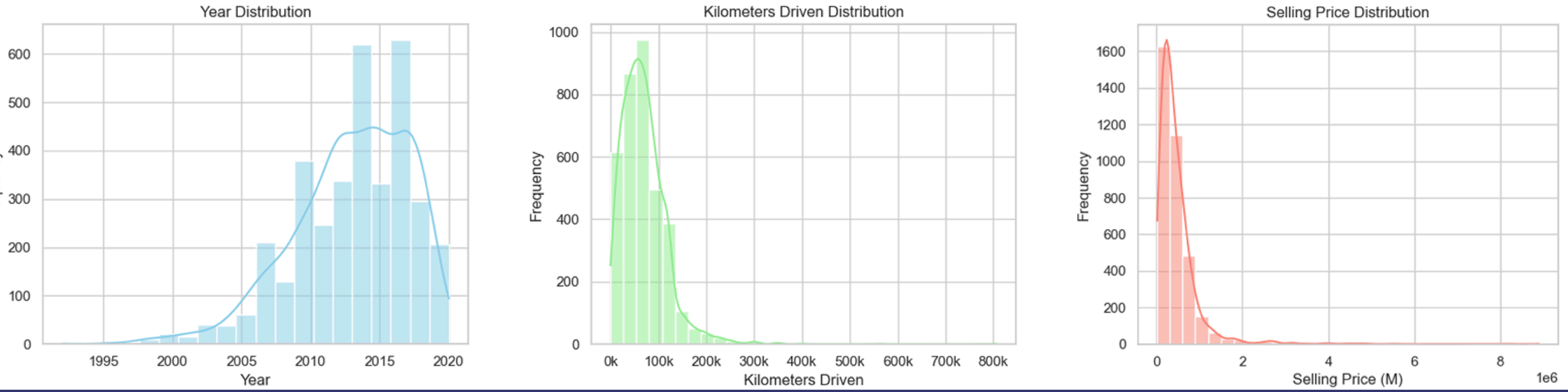
Dataset Structure, Missing values and Statistical Summary

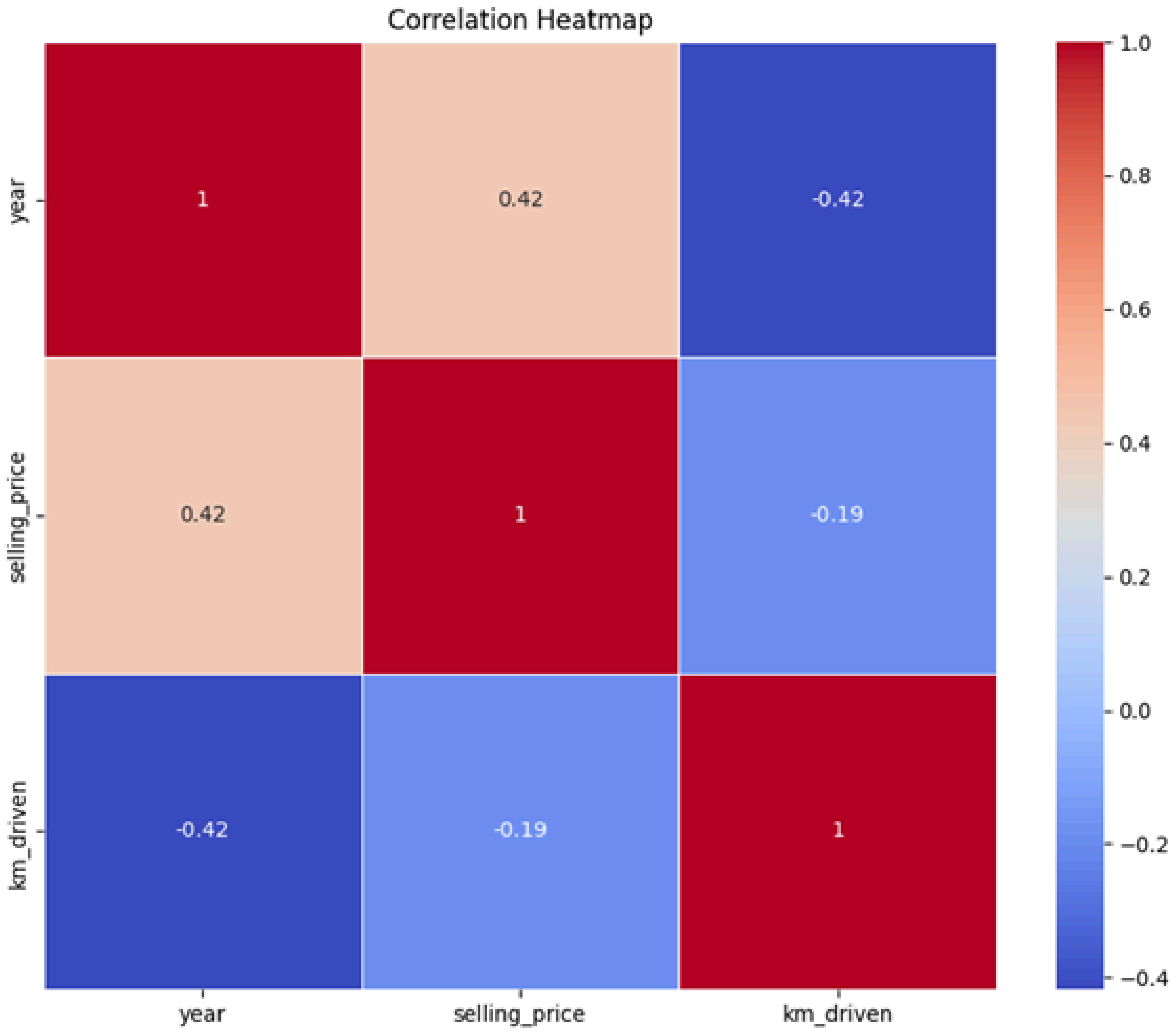
```
Dataset Information:
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4340 entries, 0 to 4339
Data columns (total 8 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   name           4340 non-null   object
 1   year           4340 non-null   int64
 2   selling_price  4340 non-null   int64
 3   km_driven      4340 non-null   int64
 4   fuel           4340 non-null   object
 5   seller_type    4340 non-null   object
 6   transmission   4340 non-null   object
 7   owner          4340 non-null   object
dtypes: int64(3), object(5)
memory usage: 271.4+ KB
None

First few rows of the dataset:
                       name  year  selling_price  km_driven    fuel seller_type trans
First few rows of the dataset:
                       name  year  selling_price  km_driven    fuel seller_type trans
0             Maruti 800 AC  2007          60000      70000  Petrol  Individual
1  Maruti Wagon R LXI Minor  2007         135000      50000  Petrol  Individual
2      Hyundai Verna 1.6 SX  2012         600000     100000  Diesel  Individual
3    Datsun RediGO T Option  2017         250000      46000  Petrol  Individual
4     Honda Amaze VX i-DTEC  2014         450000     141000  Diesel  Individual

Dataset dimensions:
```

```
Missing Values in Each Column:
name             0
year             0
selling_price    0
km_driven        0
fuel             0
seller_type      0
transmission     0
owner            0
dtype: int64

Number of Duplicate Rows: 763
Duplicates have been removed.
```

```
Statistical Summary of Numerical Features
          year  selling_price  km_driven
count  3577.00        3577.00    3577.00
mean   2012.96      473912.54   69250.55
std       4.25      509301.81   47579.94
min    1992.00       20000.00       1.00
25%    2010.00      200000.00   36000.00
50%    2013.00      350000.00   60000.00
75%    2016.00      600000.00   90000.00
max    2020.00     8900000.00  806599.00
```

Dataset Distribution





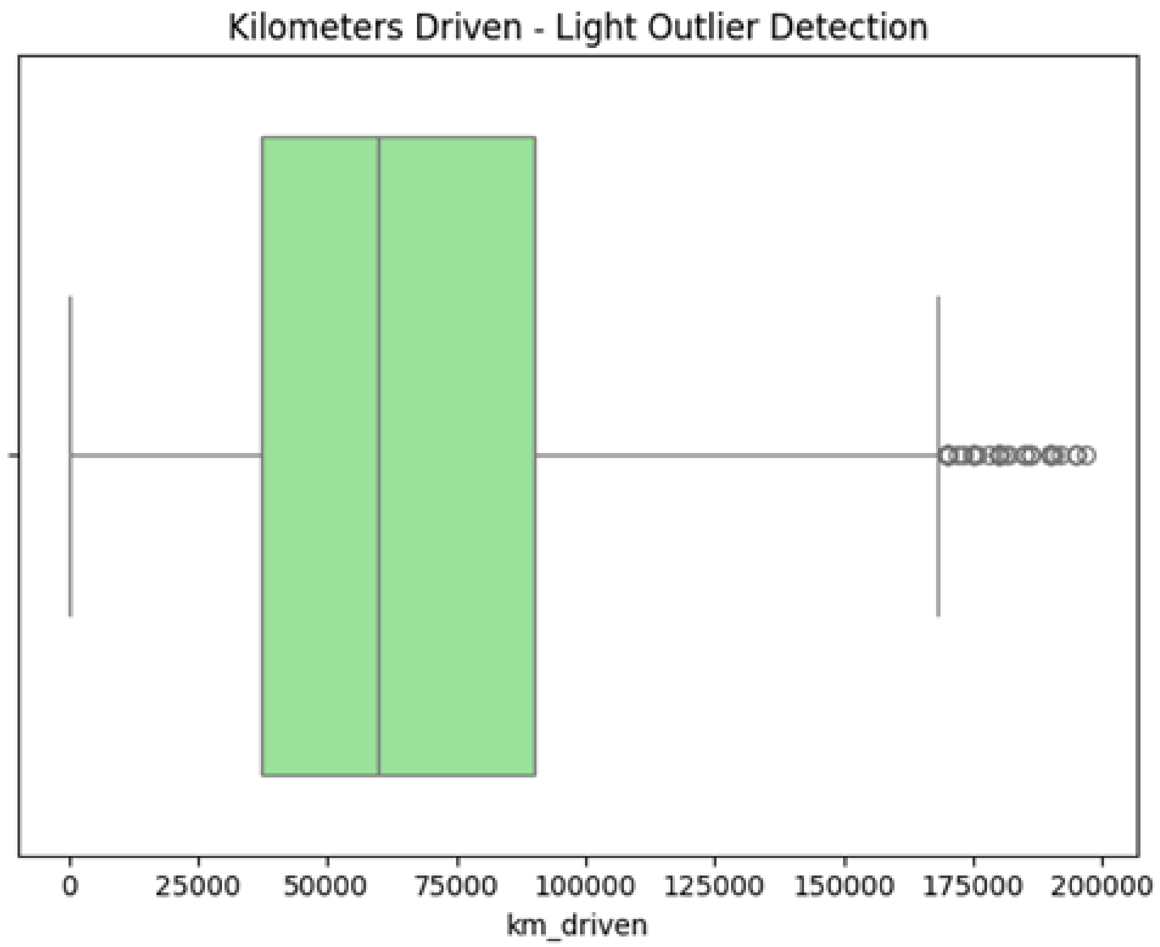
Correlation Coefficients

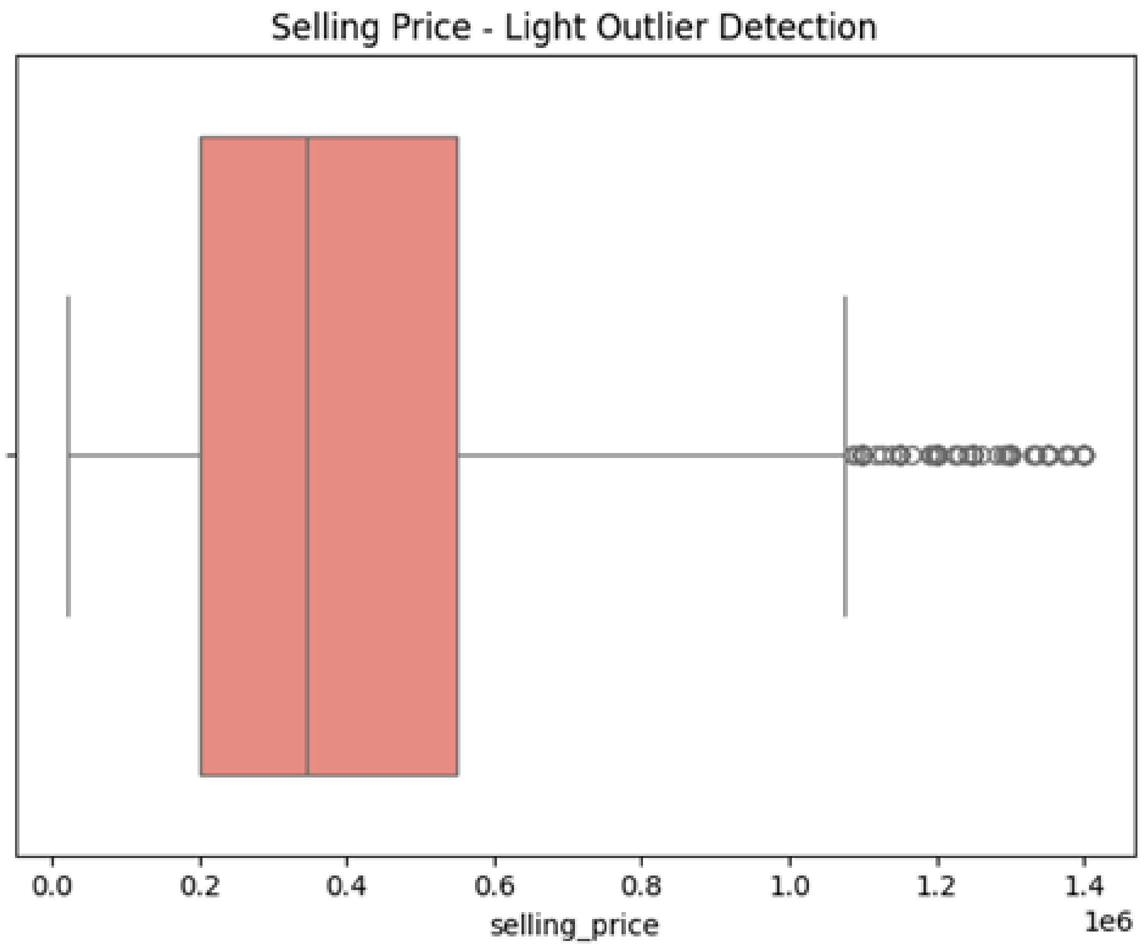
Outlier Detection

Overall Outliers

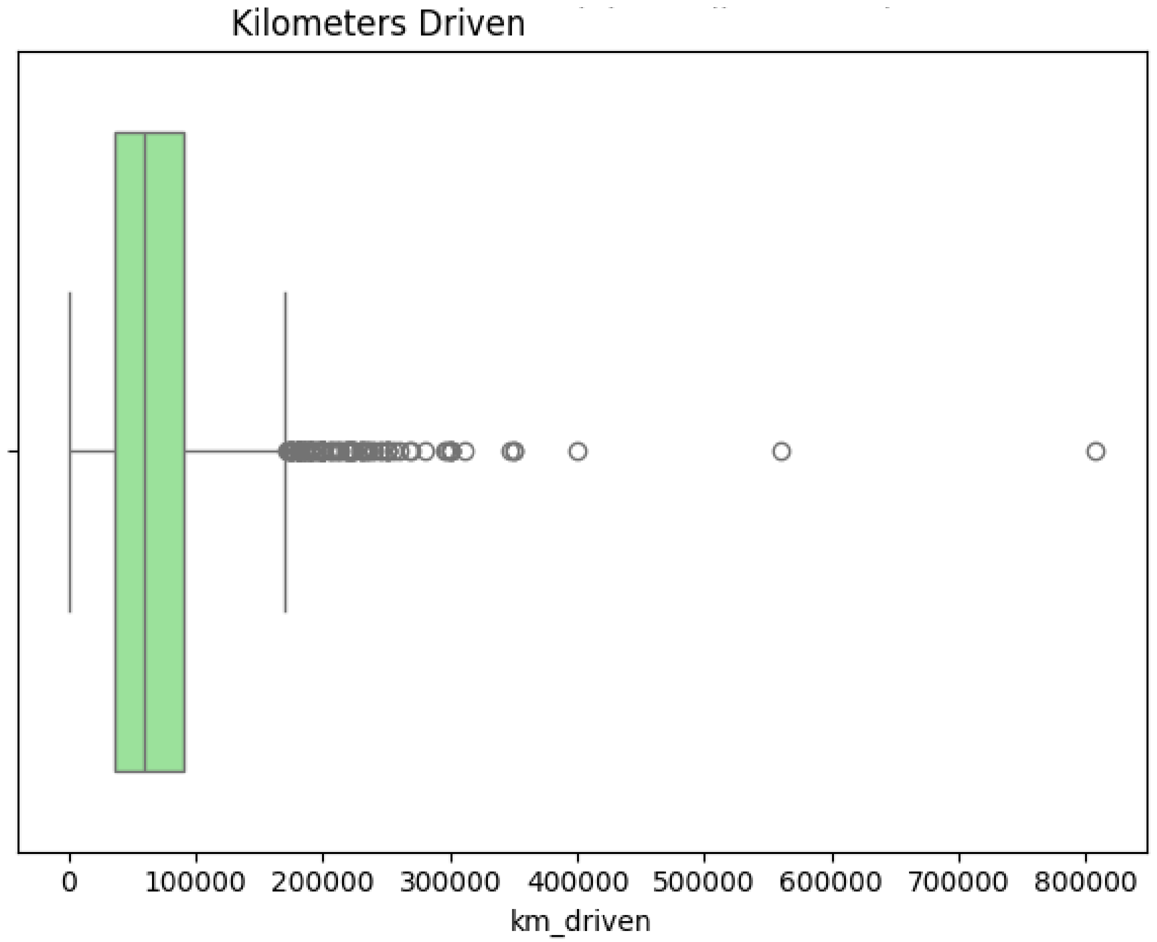
```
Kilometers Driven Outliers: 106
Selling Price Outliers: 170
```

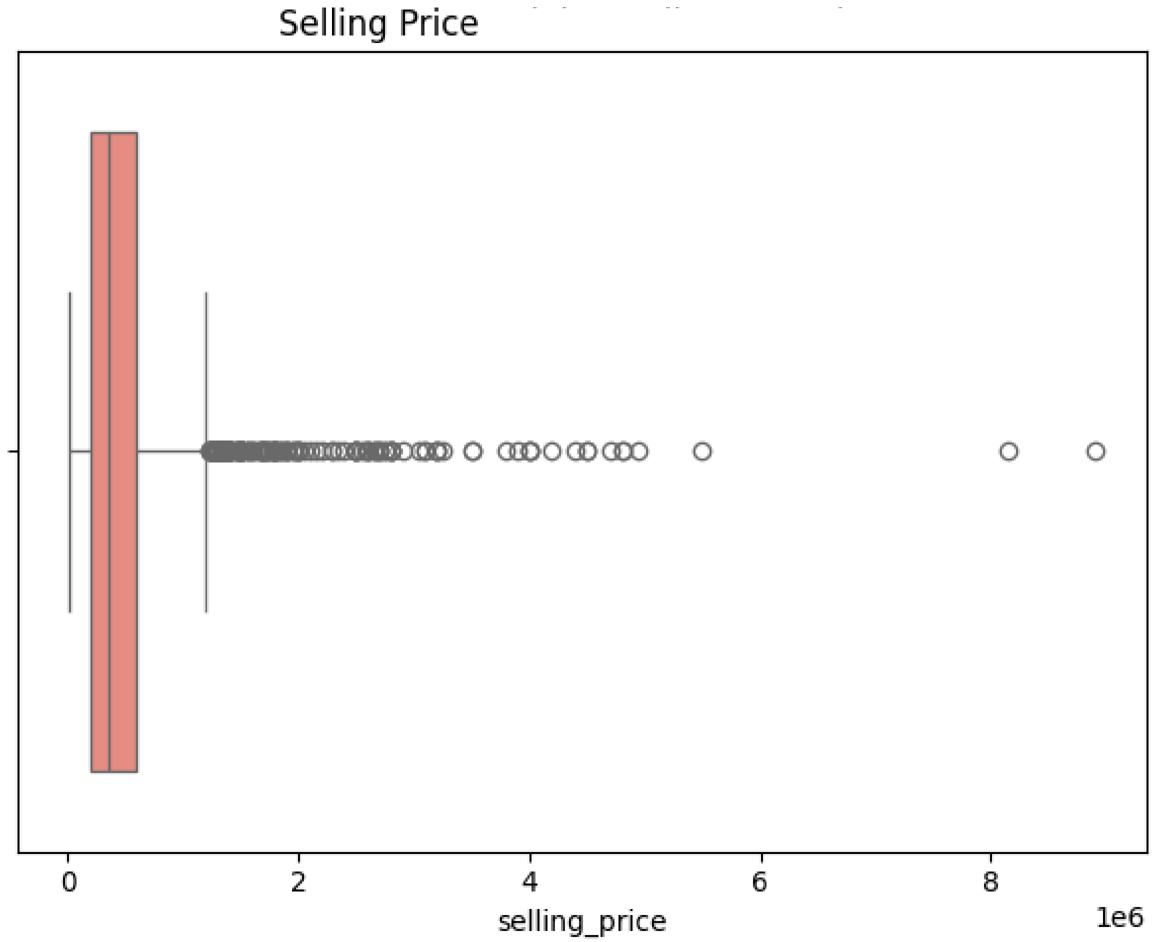
Without extreme Outliers





With extreme Outliers

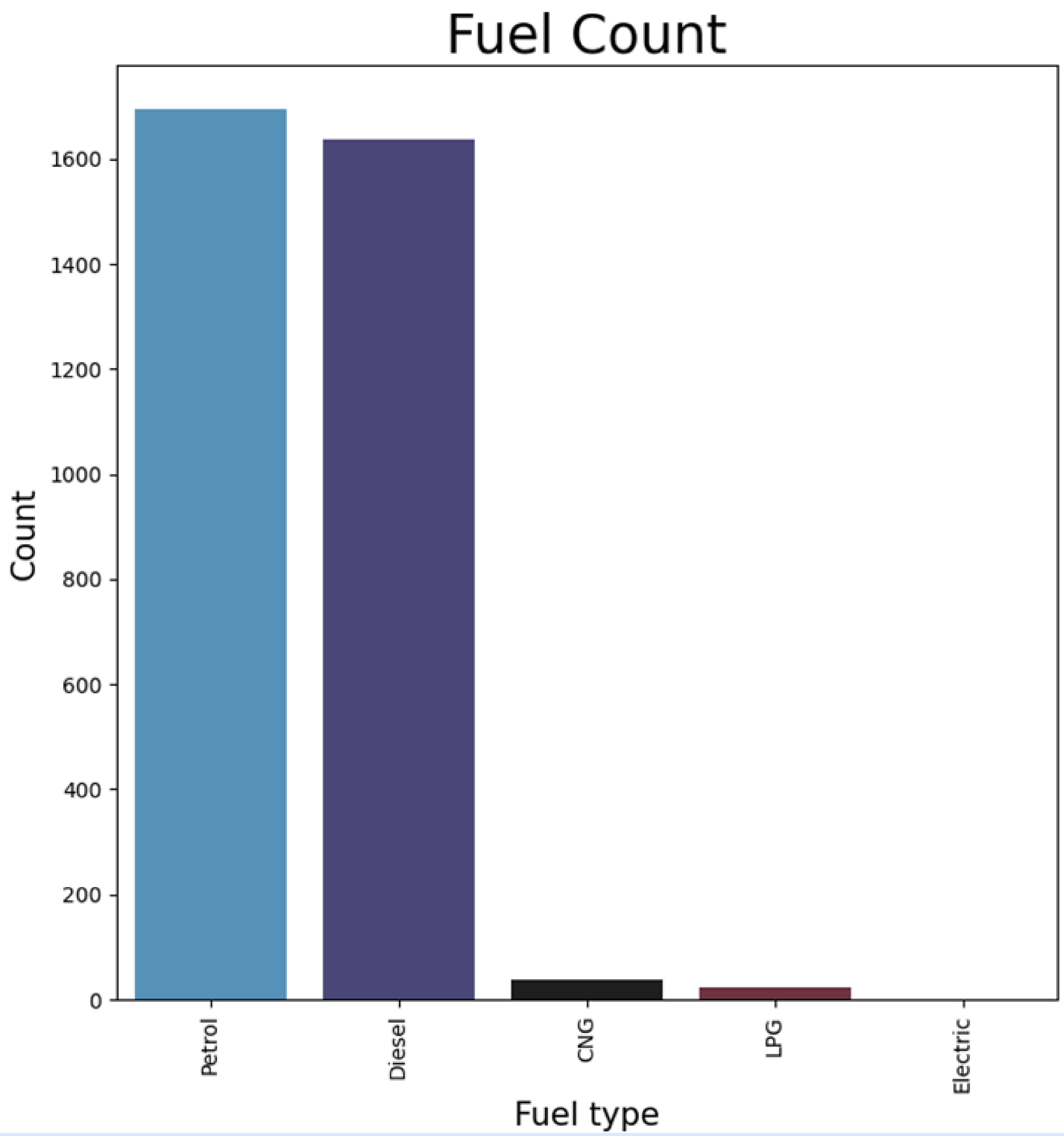




PART 2: DATA PREPROCESSING

ENCODING

```
4. Data types of encoded columns:
name                              object
year                               int64
selling_price                      int64
km_driven                          int64
owner                              int64
fuel_Diesel                        int32
fuel_Petrol                        int32
seller_type_Dealer                 int32
seller_type_Individual             int32
seller_type_Trustmark Dealer       int32
transmission_Automatic             int32
transmission_Manual                int32
dtype: object
```



SCALING

The scaling process:

1. Defined the features to be scaled: year, selling_price, and km_driven.
2. Created a new column for each scaled feature by applying MinMaxScaler.
3. Saved the updated DataFrame to a new CSV file to preserve the transformations for future use (for VS Code).

- Encoded Numerical Features: These include owner, fuel type (as one-hot encoded columns for fuel_Diesel and fuel_Petrol), seller_type (as one-hot encoded columns for Dealer, Individual, and Trustmark Dealer), and transmission (as one-hot encoded columns for Automatic and Manual). In addition to scaled ones.
- Numerical Features: To compare model accuracy, both the scaled and unscaled versions of year, selling_price, and km_driven will be used as separate sets of numerical inputs.

PART 3: MODELING

Algorithm

To predict car prices based on features like year, mileage, and fuel type, we employed three regression algorithms:

1. **Linear Regression** as a baseline model for simple linear relationships
2. **Random Forest Regressor** for handling complex interactions and preventing overfitting
3. **Gradient Boosting Regressor** for enhanced accuracy through sequential error correction

These models provide a range of approaches from basic to advanced, allowing us to identify the most effective prediction method.

Data Splitting

To optimize model evaluation and tuning, the data will be split into training, validation, and testing sets:

·**Training Set** (70%): For training the model by learning patterns in the data.

·**Validation Set** (15%): Used to tune hyperparameters and assess model improvements without bias from training data.

·**Testing Set** (15%): Used for final, unbiased performance evaluation on unseen data.

In addition, 5-fold **cross-validation** will be applied to the training set for a more robust validation method, helping avoid overfitting by testing each fold while training on the remaining data.

Model Evaluation

In this analysis, Mean Absolute Error (MAE), Mean Squared Error (MSE), and R-squared (R^2) metrics were chosen to evaluate model performance. These metrics are suitable for regression tasks as they measure the error between predicted and actual values:

- **MAE** provides a straightforward interpretation by averaging the absolute differences, helping to identify model precision.
- **MSE** measures the average squared difference between the predicted and actual values, lower value indicate better model performance.
- **R^2** Score measures the proportion of variance explained by the model, offering an intuitive sense of overall fit.

PERFORMANCE ANALYSIS

```
For scaling and encoding columns:
Linear Regression - MAE: 0.1071, MSE: 0.0217, R2 Score: 0.5321
Random Forest Regressor - MAE: 0.1047, MSE: 0.0213, R2 Score: 0.5412
Gradient Boosting Regressor - MAE: 0.0939, MSE: 0.0181, R2 Score: 0.6086
Best Random Forest Parameters: {'max_depth': 10, 'n_estimators': 200}
Best Random Forest R2 Score: 0.5334382294064354
Test MAE: 0.09443934539861136
Test MSE: 0.017528601995773028
Test R2 Score: 0.5417815962867407

For not scaled columns ('year', 'km_driven' and 'selling_price' only):
Linear Regression - MAE: 165800.8846, MSE: 54662892876.7665, R2 Score: 0.3803
Random Forest Regressor - MAE: 163975.1370, MSE: 57459184426.7822, R2 Score: 0.3486
Gradient Boosting Regressor - MAE: 154779.6914, MSE: 52036045976.2502, R2 Score: 0.4101
Best Random Forest Parameters: {'max_depth': 10, 'n_estimators': 200}
Best Random Forest R2 Score: 0.33768544655478716
Test MAE: 161142.50974440476
Test MSE: 51824080914.49932
Test R2 Score: 0.28862486025600187
```

Gradient Boosting Regressor

- Achieved the lowest Mean Absolute Error (MAE) of 0.0939, smallest average prediction error
- Obtained the highest R^2 score of 0.6086, indicating it explained about 61% of the variance in car prices

Prediction Error Calculation

The Input Values (taken from row 32):

- Car Model: 2014
- Km Driven: 64,000
- Gas: Petrol
- Seller Type: Individual
- Transmission Type: Manual
- Owner Level: Second

Results:

- Actual Price: 290,000 Indian Rupees (INR)
- Predicted Price: 312,510.96 Indian Rupees (INR)

Calculation:

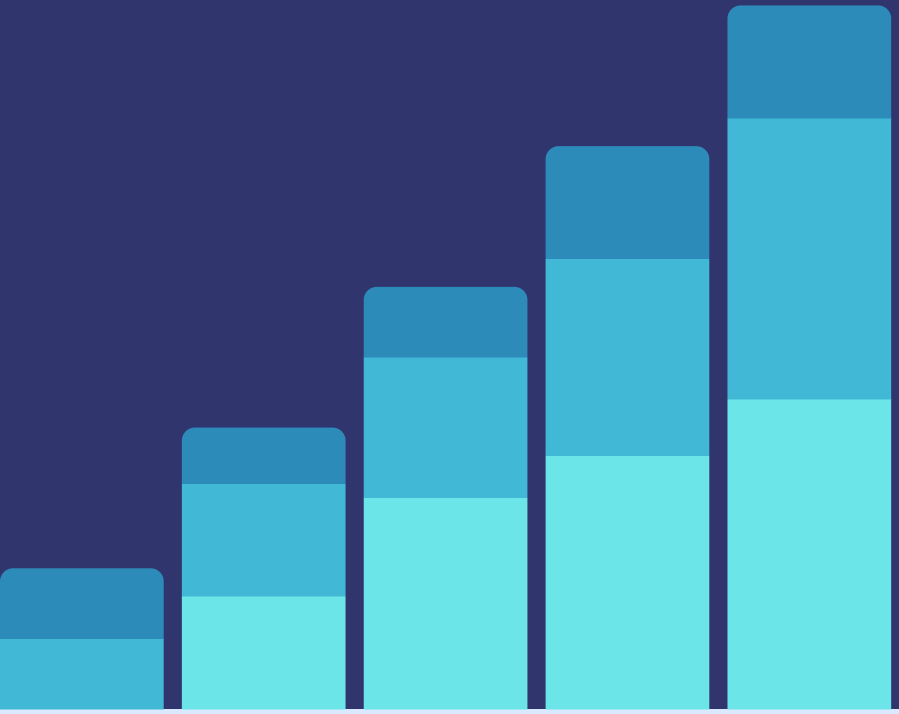
Percentage Error= ((312,510.96 - 290,000) / 290,000) * 100 = 7.8%

Conclusion:

The prediction error is 7.8%.

FINAL MODEL

Having identified Gradient Boosting Regressor as our superior model, we developed a final predictive tool that generates accurate car valuations based on user-input features. This practical application offers <u>invaluable market guidance for both buyers and sellers in the used car marketplace</u>, enabling data-driven pricing decisions.



CONCLUSION

Our car price prediction model, utilizing Gradient Boosting Regressor with scaled features, achieved strong performance metrics with an R-squared score of 0.6086 and MAE of 0.0939. Through optimized data preprocessing and feature scaling, the model effectively captures the complex relationships between car attributes and prices, delivering reliable valuations for the used car market.

THANK YOU