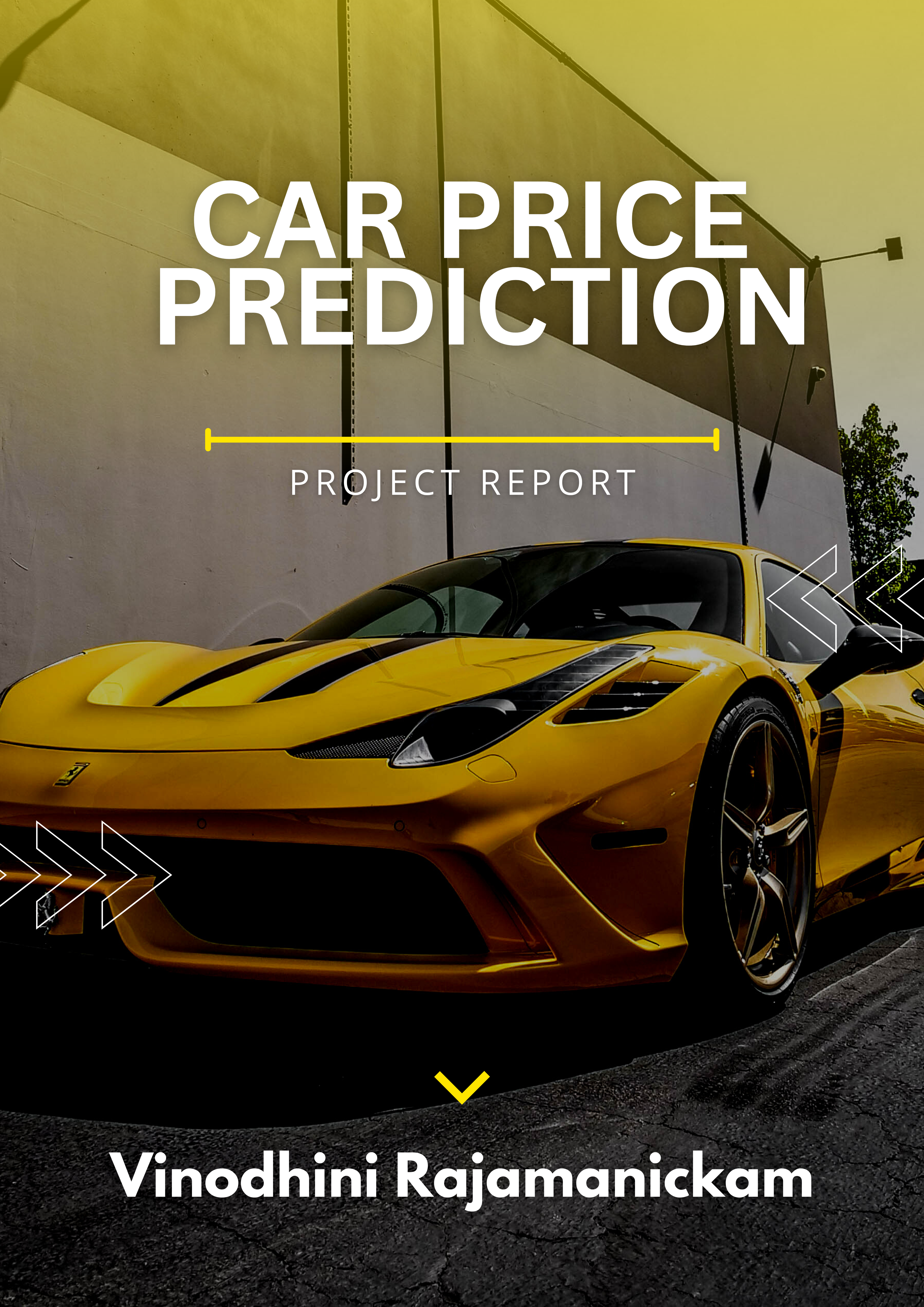
CAR PRICE PREDICTION

PROJECT REPORT



Vinodhini Rajamanickam

INDEX

Abstract

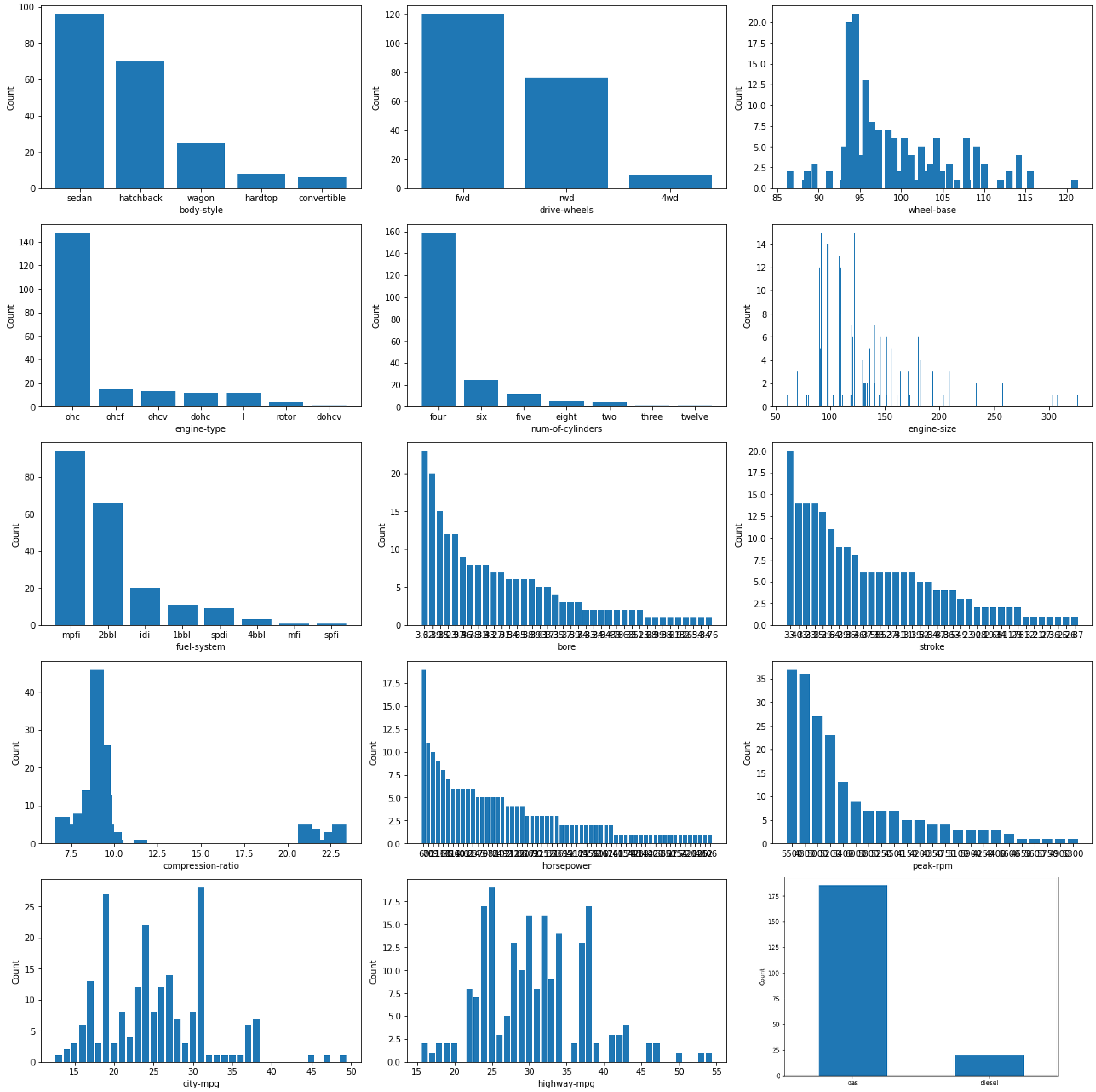
his report presents a multi-class classification approach to predict the class of cars based on various features. The goal is to develop a model that accurately identifies the class of a car given its attributes. The report covers the preprocessing steps, train-test split, model comparison, evaluation using five different algorithms, and the selection of the best-performing mode based on accuracy, F1 score, and confusion matrix.

Introduction

The automotive industry is constantly evolving, and with the wide range of car models available on the market, determining the accurate price of a car can be a challenging task. The ability to predict car prices accurately is invaluable for various stakeholders, including car dealerships, buyers, and insurance companies. In this project, we aim to develop a machine learning model that can predict the price of a car based on its relevant features.

The main objective of this project is to build a robust machine learning model that can predict the price of a car with a high degree of accuracy. By using historical data and employing advanced regression techniques, we seek to develop a model that can effectively capture the complex relationships between car features and their corresponding prices.

For this project, we will utilize a comprehensive dataset containing information about various cars, including features such as make, model, fuel type, engine size, horsepower, mileage, and more. The dataset encompasses a diverse range of car types and specifications, providing a rich source of information to train and evaluate our prediction model.



Step 1: Load the Data and Understand the Values

To start the data analysis, the first step is to load the data into a Pandas DataFrame and gain an understanding of the dataset and its values.

Step 2: Exploratory Data Analysis (EDA)

After loading the data into a Pandas data frame, the next step is to perform Exploratory Data Analysis (EDA) to gain insights and a better understanding of the data. EDA involves examining the shape of the data, obtaining basic information, statistical details, value counts of each feature, correlation matrix, and assessing the target variable ("Price"). Additionally, it is essential to , identify duplicate values, and determine if there are any null values present in the dataset.

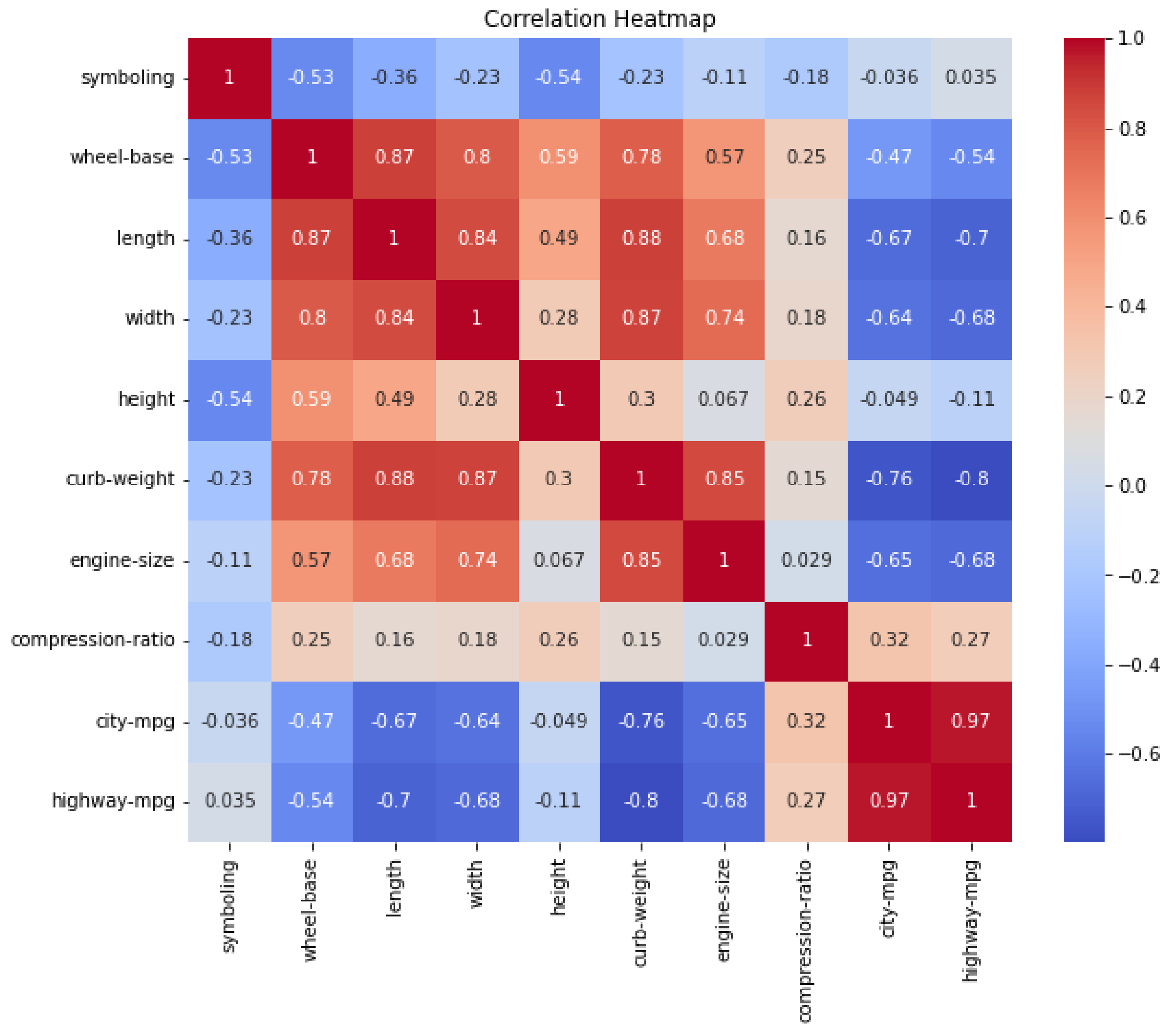
1. Get the shape of the data frame using df.shape. This will provide the number of rows and columns in the dataset. the shape of the data was 205, 26

2.Basic Information: Use df.info() to obtain basic information about the data, including the data types of each feature, memory usage, and the presence of any null values.

3.Statistical Details: Employ df.describe() to generate statistical details such as count, mean, standard deviation, minimum, quartiles, and maximum values for each numerical feature.

4.Value Counts of Each Feature: Utilize df[column].value_counts() to obtain the count of unique values in each feature. This will help identify the frequency of different categories within categorical variables.

5.Correlation Matrix: Calculate the correlation matrix using df.corr() to determine the relationships between numerical features. A correlation matrix provides insights into the strength and direction of linear relationships, assisting in feature selection and identifying potential multicollinearity.

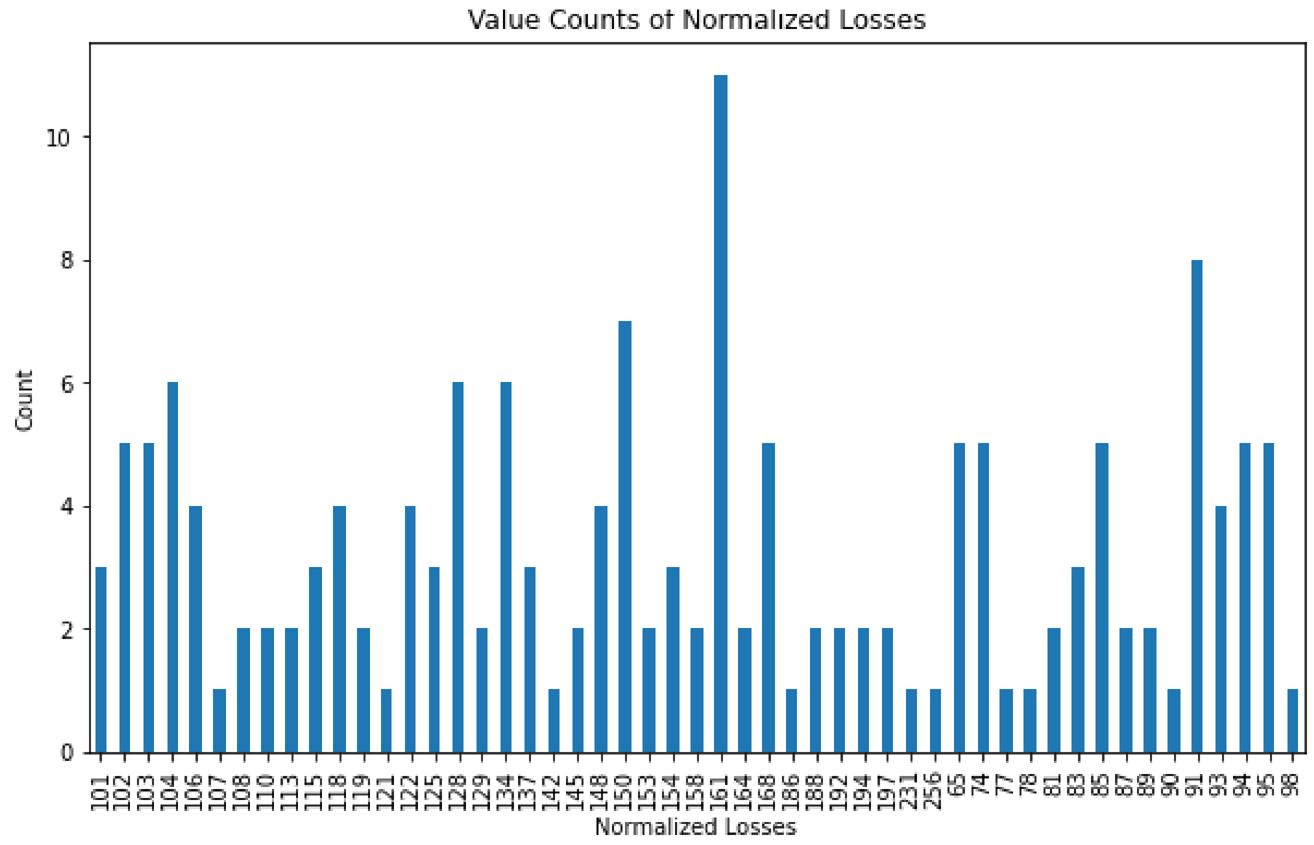


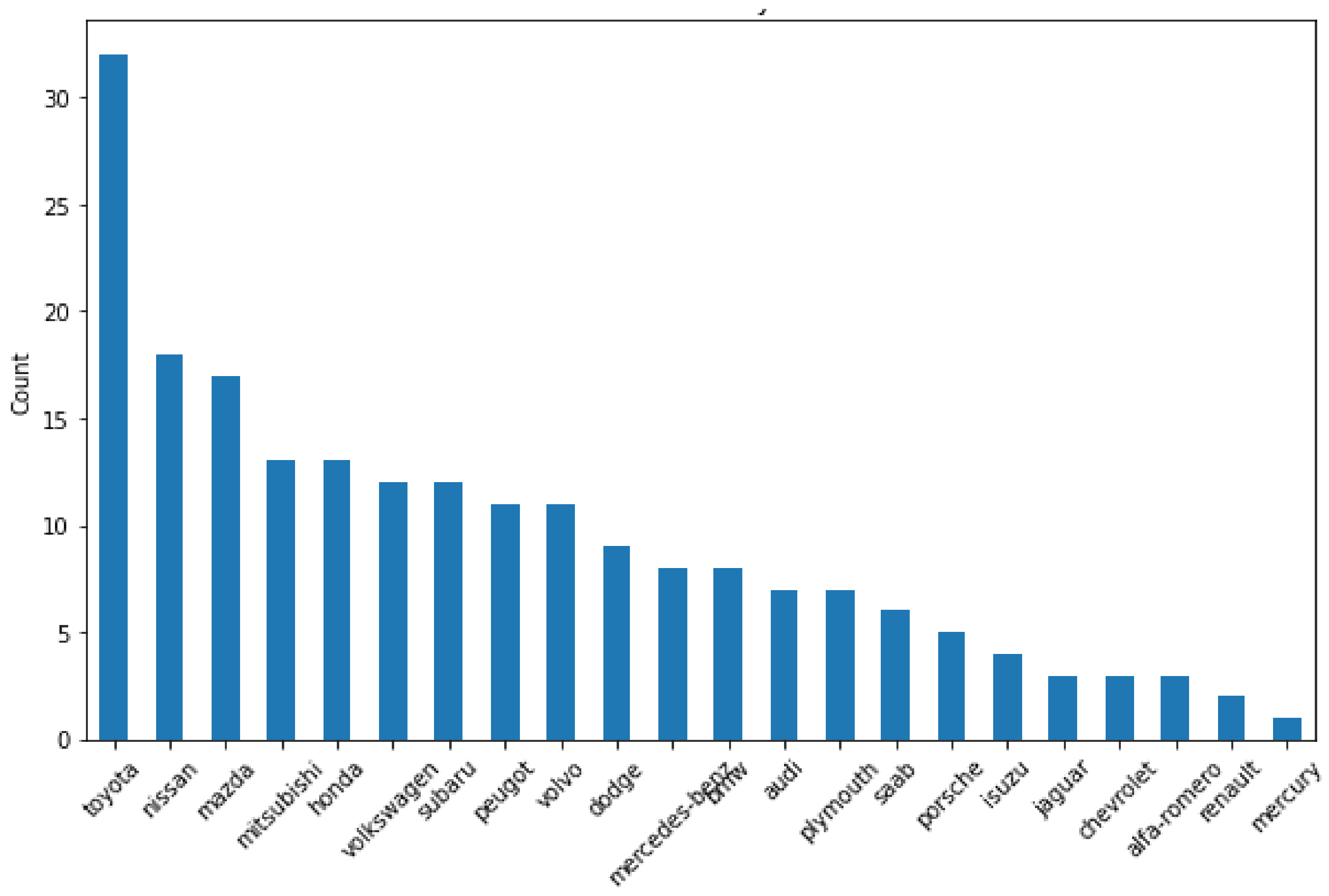
6.Target Variable Analysis:
Analyze the target variable, which in this case is "Price". There were 4 null values which I removed during preprocessing the data.

7. Duplicate Values:
Check for duplicate values using df.duplicated().sum() .data does not have any duplicate values

8. Null Values:
Evaluate the presence of null values in the dataset using df.isnull().sum(). the data did not contain any null values. Instead of having explicit null values, there are "?" marks present. it is essential to convert these "?" marks to appropriate null values. I used NumPy library to replace "?" with np.nan. there were 57 total null values.which I will handle later.

By performing these EDA steps, I have gained valuable insights into the structure, quality, and characteristics of the dataset. These insights guided me for further preprocessing steps and model building, leading to more accurate predictions in my car class prediction project.





Step: 3. Train-Test Split

The dataset was split into two subsets: a training set and a testing set. The training set, which comprises 80% of the data, was used to train the models, while the remaining 20% was used for model evaluation.

Step: 4. Preprocessing Steps

Before training the models, several preprocessing steps were performed to ensure the data is in a suitable format for classification:

1. Data Cleaning:

a. since there were no duplicate values, no need to remove any data.

b. null values :

- Target variable "Price" had 4 null values which I droppped.
- Due to the distinctiveness of all the features and the varying values within each feature, we will assign unique null values for each car individually.
- The datatype of these columns were object type . Hence, I converted them into float type to perform numerical operations.
- "horsepower" column where the make is "renault" with the value 110.
- "peak-rpm" column where the make is "renault" with the value 5250.
- "bore" column where make is "mazda" with the mean of the "bore" column where make is also "mazda".
- "stroke" column where make is "mazda" with the mean of the "stroke" column where make is also "mazda".
- "normalized-losses" column where the make is "audi" with the mean value of the "normalized-losses" column for the "audi" make
- "normalized-losses" column where the make is "bmw" with the value 188.
- "normalized-losses" column where the make is "jaguar" with the value 145.
- "normalized-losses" column where the make is "mercedes-benz" with the value 142.
- "normalized-losses" column where the make is "mitsubishi" with the value 157(mean of that make).
- "normalized-losses" column where the make is "peugot" with the value 161.
- "normalized-losses" column where the make is "mercedes-benz" with the value 119.
- "normalized-losses" column where the make is "renault" with the value 150.
- "normalized-losses" column where the make is "mazda" with the mean value of the "normalized-losses" column for the "mazda" make.
- "normalized-losses" column where the make is "volkswagen" with the mean value of the"normalized-losses" column for the "volkswagen" make
- "normalized-losses" column where make is "toyota" with the mean of the "normalized-losses" column where body style is "wagon" and make is "toyota".
- "no of doors" column where the make is "mazda" with the value 4.
- "no of doors" column where the make is "dodge" with the value 4.

c. Drop columns : I have dropped the following columns "symboling", "length", "width", "height", "highway-mpg","num-of-doors".

2. Encoding :

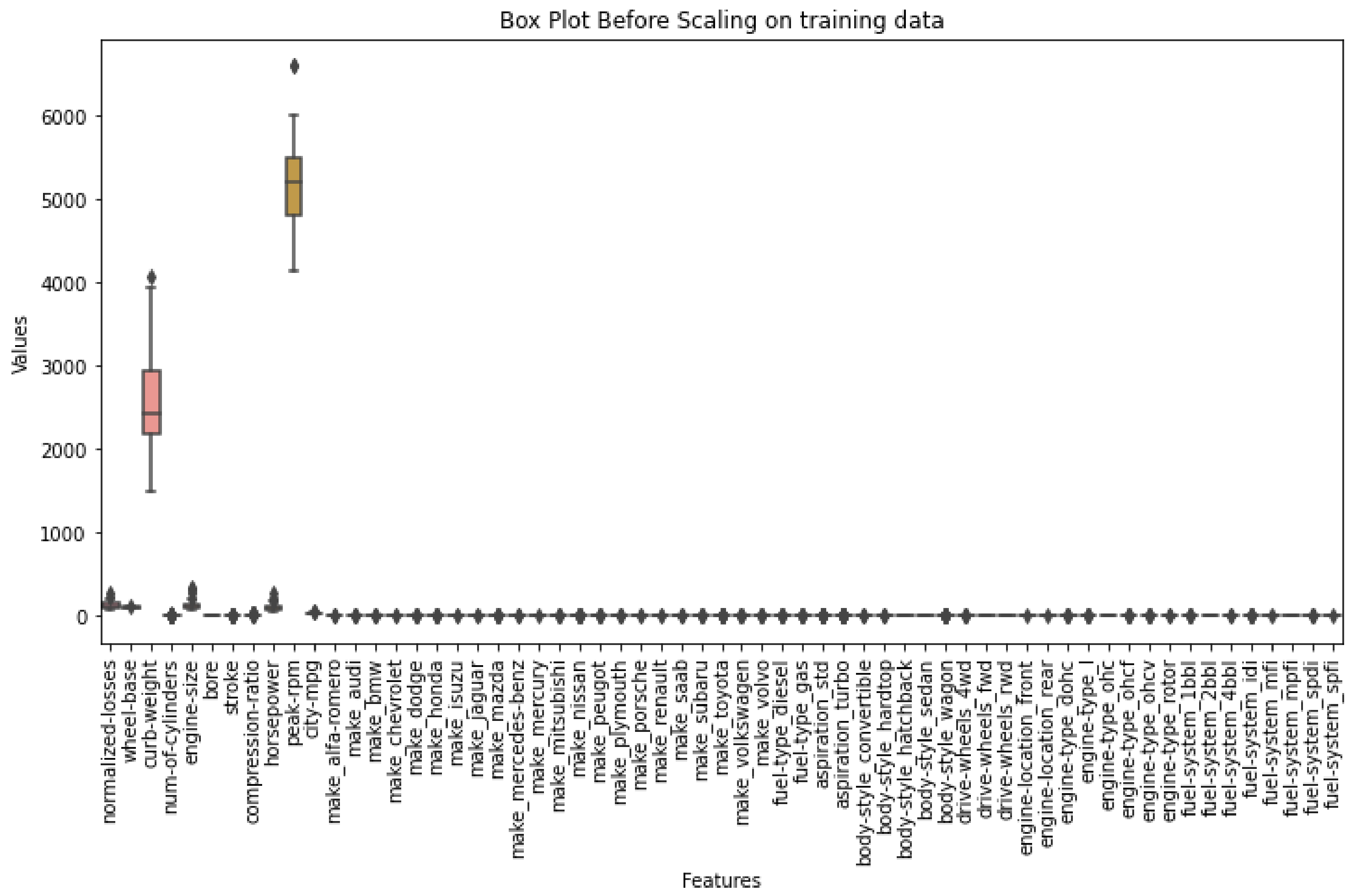
- use One Hot Encoder for the columns ['make', 'fuel-type',"aspiration","body-style","drive-wheels","engine-location","engine-type","fuel-system"].
- number of cylinders column, I simply converted categorical value of numbers into numerical value.

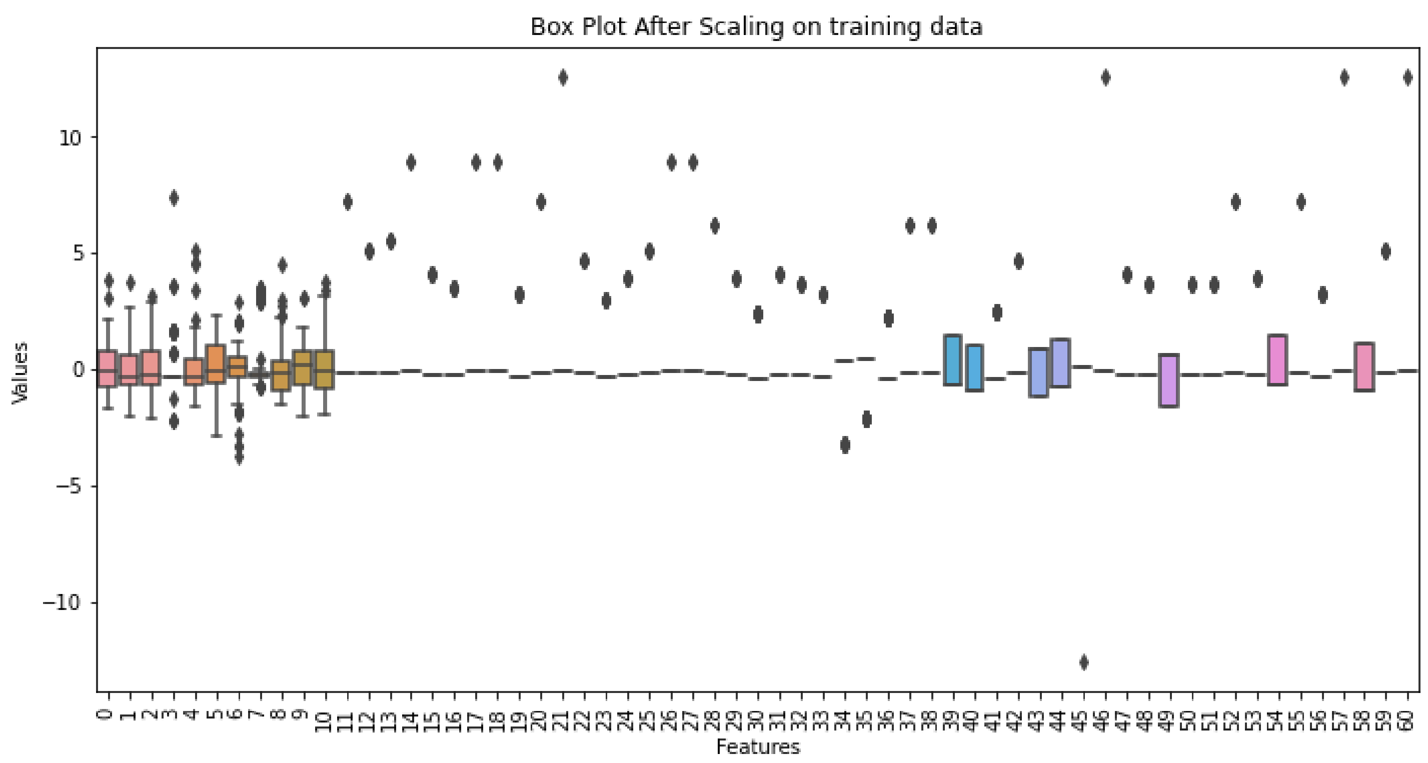
3. Feature Scaling :

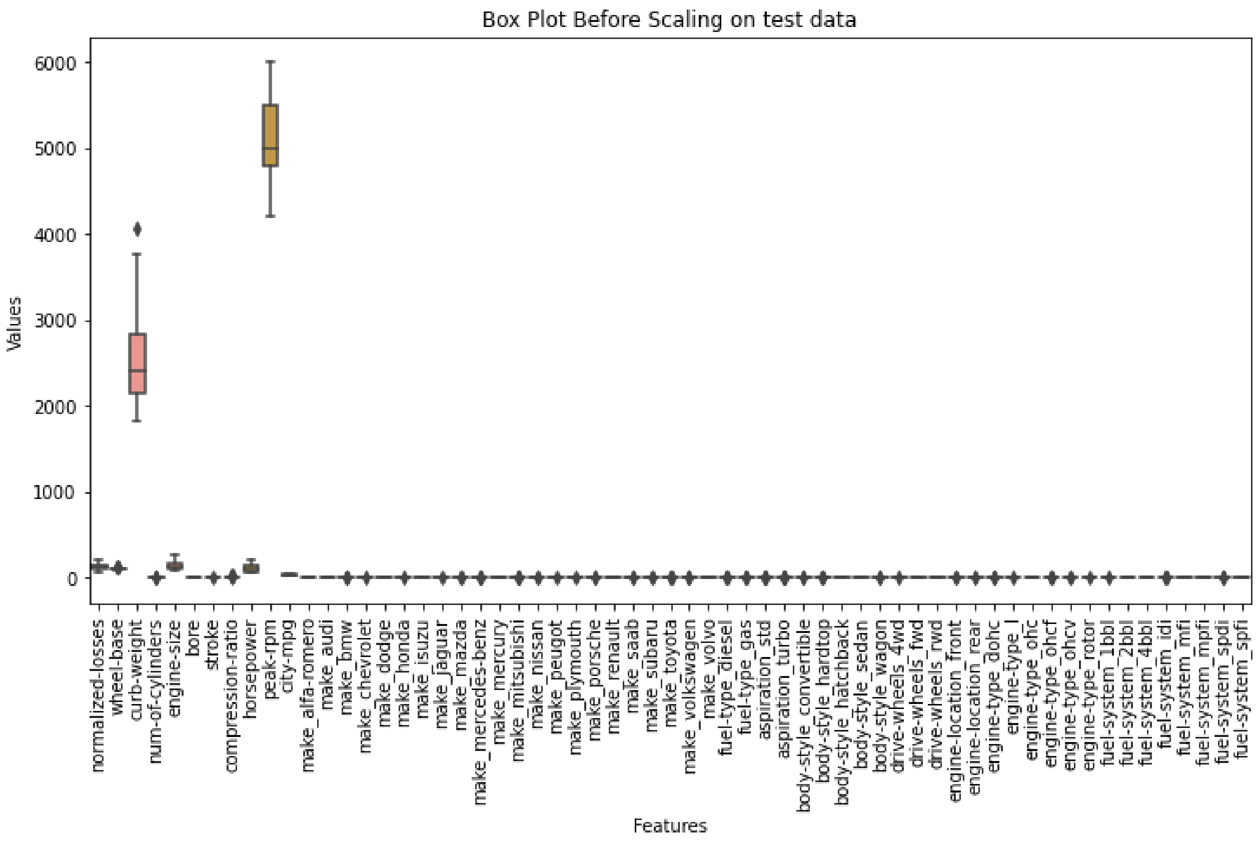
- The box plots showed huge variations in the ranges and scales of different features, it indicates the need for scaling. hence, I applied Standard Scaler

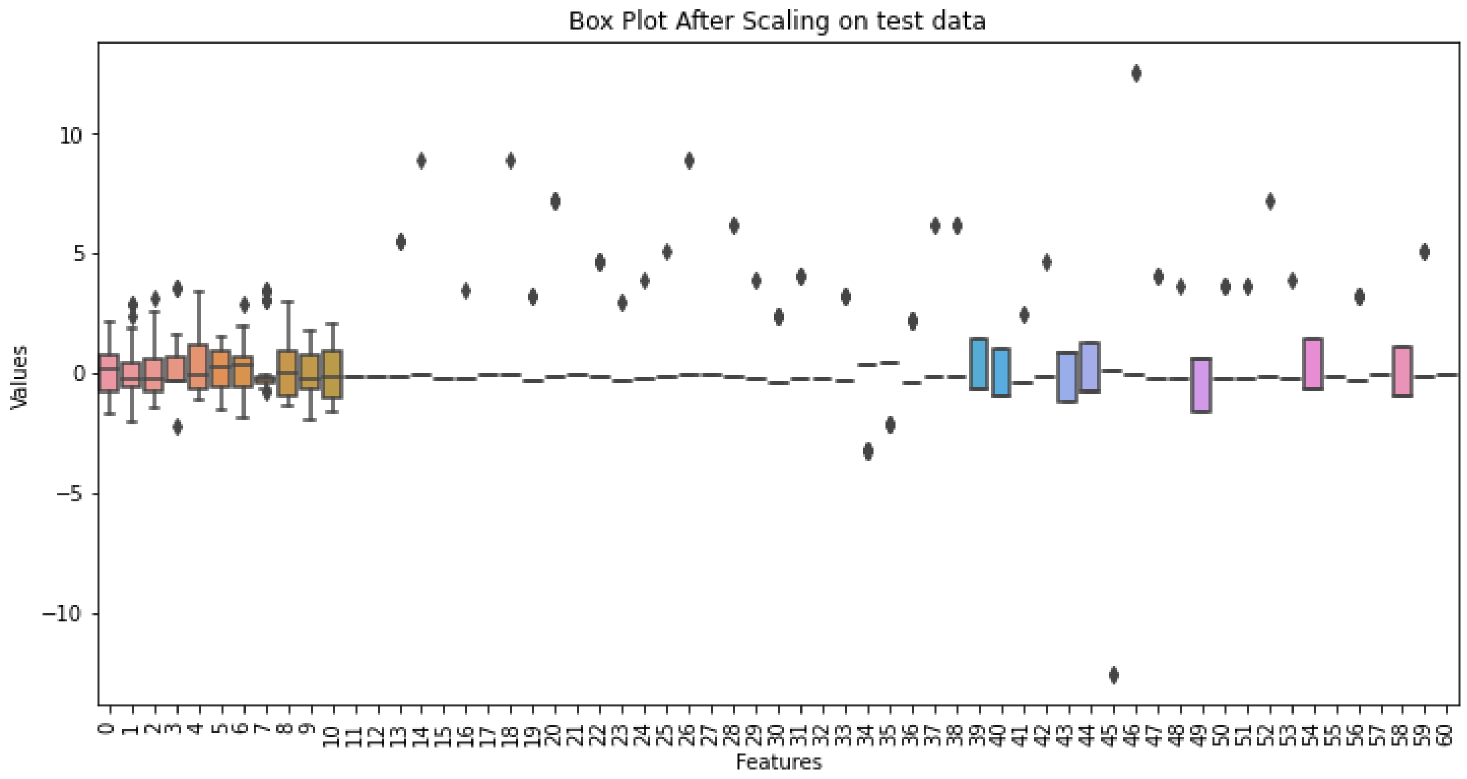
4. Feature Extraction:

- for feature extaction I applied Principal component analysis (PCA). PCA is an unsupervised linear transformation technique which is primarily used for feature extraction and dimensionality reduction.









Step: 5. Model Comparison

To find the best model for car class prediction, five different classification algorithms were evaluated:

a. Linear Regression: A linear model that predicts a continuous target variable by fitting a linear equation to the input features. It assumes a linear relationship between the predictors and the target variable.

b. Support Vector Regression (SVR): A regression model that utilizes support vector machines to perform regression tasks. SVR aims to find a hyperplane that fits the training data as closely as possible while still maintaining a specified tolerance margin.

c. Decision Tree Regressor: A tree-based model that recursively splits the dataset based on different features to create decision rules for predicting continuous target variables. Each leaf node of the tree represents a predicted value.

d. Random Forest Regressor: An ensemble model that combines multiple decision trees to make predictions on continuous target variables. Each tree in the random forest independently predicts the target variable, and the final prediction is obtained by aggregating the predictions of all the trees.

Step: 6. Model Evaluation

The models were evaluated using three metrics: MAE, R2 score, and MSE.

1. Mean Squared Error (MSE): It measures the average squared difference between the predicted and actual values. Lower values indicate better model performance.

2. Mean Absolute Error (MAE): It measures the average absolute difference between the predicted and actual values. It provides a measure of the average magnitude of the errors.

3. R-squared (R2) Score: It represents the proportion of the variance in the target variable that can be explained by the model. It ranges from 0 to 1, where 1 indicates a perfect fit.

Based on the evaluation metrics , here is the interpretation for each model:

Linear Regression:
Mean Squared Error (MSE): 3.887578529603833e+28
Mean Absolute Error (MAE): 30792692077468.805
R2-score: -3.1775080221988264e+20

Support Vector Machine (SVM):
Mean Squared Error (MSE): 149241026.71963352
Mean Absolute Error (MAE): 7947.505188402194
R2-score: -0.21981988539058572

Decision Tree:
Mean Squared Error (MSE): 10601716.878048781
Mean Absolute Error (MAE): 2369.609756097561
R2-score: 0.9133469840607603

Random Forest:
Mean Squared Error (MSE): 20179907.302163295
Mean Absolute Error (MAE): 2753.364613821138
R2-score: 0.835059750300691

The Linear Regression model performs poorly, with extremely high MSE and MAE values. The negative R2-score indicates that the model performs worse than simply using the mean of the target variable for predictions. It fails to capture the relationship between the features and the car prices effectively.

The Decision Tree Regression model shows significantly improved performance compared to Linear Regression. It exhibits relatively low MSE and MAE values, indicating better accuracy in predicting car prices. The high R2-score suggests that the model explains a substantial portion of the variance in the data and fits the car price prediction task quite well.

The Random Forest Regression model performs well, with relatively low MSE and MAE values. It outperforms the Linear Regression model, demonstrating better accuracy in predicting car prices. The R2-score of 0.835 indicates a good fit to the data, explaining a significant portion of the variance in the car prices.

The SVR model performs poorly compared to the other models. It exhibits high MSE and MAE values, indicating lower accuracy in predicting car prices. The negative R2-score suggests that the model does not fit the data well and fails to capture the underlying relationship between the features and car prices.

Overall, based on the evaluation metrics, the Decision Tree Regression model performs the best among the four models. It demonstrates relatively low MSE and MAE values, indicating better accuracy and precision in predicting car prices. Additionally, it has a high R2-score, suggesting a good fit to the data and a significant explanation of the variance in the car prices.

The Random Forest Regression model also performs well, providing competitive accuracy and model fit. However, it slightly lags behind the Decision Tree Regression model in terms of the evaluation metrics.

On the other hand, the Linear Regression model performs poorly, while the SVR model shows the weakest performance among the four models.
Therefore, based on this model evaluation, the Decision Tree Regression model appears to be the most suitable choice for the car price prediction task, providing accurate and reliable predictions.

Step: 7. HyperParameter Tuning

Decision Tree Regressor :

Best Parameters: {'max_depth': 7, 'min_samples_leaf': 2, 'min_samples_split': 2}
Mean Squared Error (MSE): 21537189.576416876
Mean Absolute Error (MAE): 2911.36268531803
R2-score: 0.8239660186063034

Random Forest Regressor :

Best Parameters: {'max_depth': 10, 'min_samples_leaf': 1, 'min_samples_split': 2'n_estimators': 300}
Mean Squared Error (MSE): 22542077.768463176
Mean Absolute Error (MAE): 2830.5147372262877
R2-score: 0.8157525760550467

Step: 8. Model Selection

After comaprison and evaluation I selected Decision Tree as the Final model of my project with a

Mean Squared Error (MSE): 10364759.219512194
Mean Absolute Error (MAE): 2206.048780487805

R2-score: 0.9152837548685726

Step: 9. Feature Importances

Feature importance refers to a technique used to determine the relevance or importance of each feature (or variable) in a dataset in relation to the target variable. It helps in understanding which features have the most significant impact on the target variable and can be used to identify the key drivers of a particular outcome.

Decision Tree is a machine learning algorithm that is often used to calculate feature importances.

decision_tree.feature_importances_

Conclusion

In conclusion, this project focused on Predicting the price of cars based on various features. The objective was to build a machine learning model that could accurately predict the Price of a car given its features.

Started exploring the dataset and understanding the distribution and the characteristics of the features. Data preprocessing techniques were applied, including handling missing values, encoding categorical variables, and scaling numerical features.

Several regression algorithms, including Linear Regression, Random Forest, SVR,and decision tree , were trained and evaluated using performance metrics such as MSE, MAE and R2-score. The models were tuned and optimized to achieve the best possible performance.

After comparing the results, it was determined that the Decision Tree algorithm outperformed the other models in terms of R2 score. It exhibited the highest accuracy in Predicting Price of a car based on the given features.

The project also involved feature importance analysis using techniques such as Decision Tree's feature importances. This analysis provided insights into the significance of each feature in predicting car Prices.