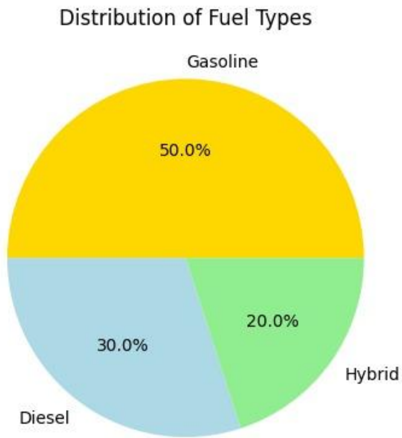
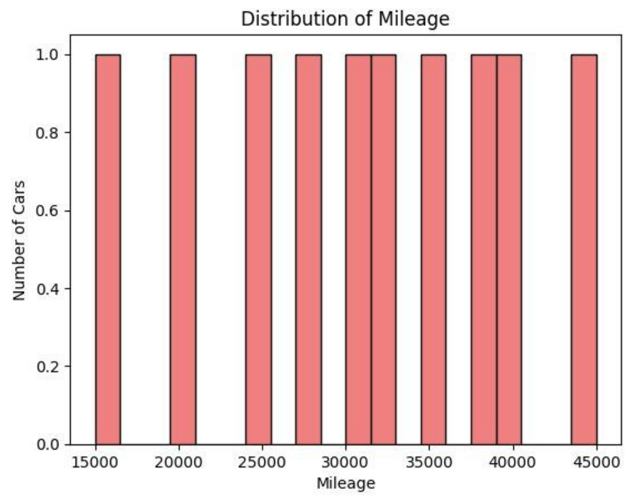
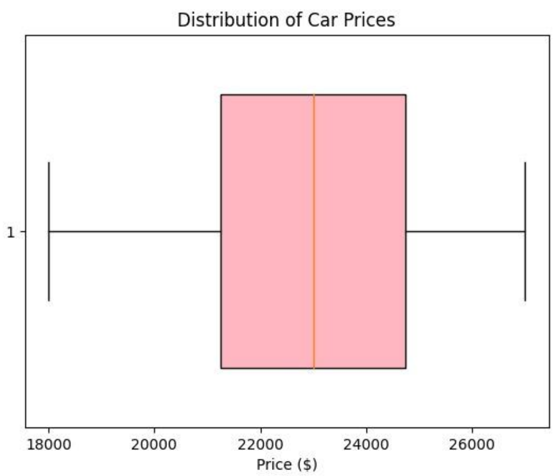
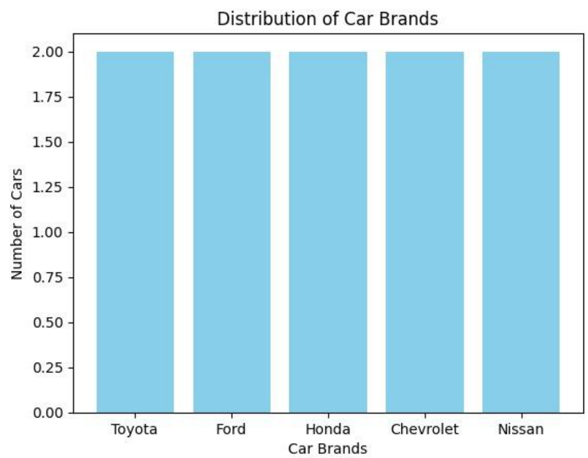
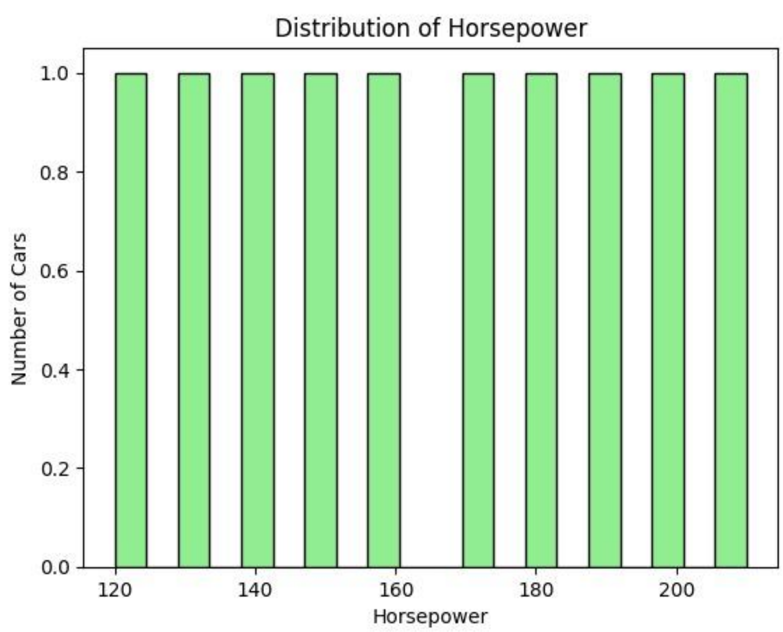

Distribution of Fuel Types
Gasoline
50.0%
30.0%
20.0%
Hybrid
Diesel


Distribution of Mileage
Number of Cars
0.0
0.2
0.4
0.6
0.8
1.0
15000
20000
25000
30000
35000
40000
45000
Mileage


Distribution of Car Prices
1
18000
20000
22000
24000
26000
Price ($)


Distribution of Car Brands
Number of Cars
0.00
0.25
0.50
0.75
1.00
1.25
1.50
1.75
2.00
Toyota
Ford
Honda
Chevrolet
Nissan
Car Brands


Distribution of Horsepower
Number of Cars
0.0
0.2
0.4
0.6
0.8
1.0
120
140
160
180
200
Horsepower

Label Encoding:

- Pros: Saves memory, and can be effective when there is ordinality in the categories (i.e., some inherent order).
- Cons: Introduces ordinality that may not exist, which could mislead the model. Not suitable for categories without ordinal relationship.

One-Hot Encoding:

- Pros: Preserves the independence of categories, suitable for non-ordinal data. Does not impose false ordinality.
- Cons: Expands the feature space, potentially leading to a sparse matrix if there are many categories.

Therefore, the choice between Label Encoding and One-Hot Encoding depends on the context and the type of data being used. One-Hot Encoding is a good universal method that works for all commonly used machine learning models, while Label Encoding tends to work best on tree-based models. However, One-Hot Encoding can lead to a large feature set, which could cause memory or learning problems, depending on the model used. On the other hand, Label Encoding saves memory but can introduce ordinality that may not exist, which could mislead the model.

Upon analyzing the dataset, it is evident that the brand "Toyota" dominates in terms of the highest number of cars, indicating its prevalence among the given vehicles.'Horsepower' values corresponding to that category, potentially requiring an adjustment of the bin edges for a more representative distribution. Moving on to the scaling operations, the sum of scaled values for both the 'Horsepower' and 'Mileage' features is remarkably close to zero after standardization. This aligns with the expected outcome of standardization, aiming to center the data around zero with a standard deviation of 1.