In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
```

In [4]:
```
cars=pd.read_csv("final_cars.csv")
```

In [5]:
```
cars.columns
```

Out[5]:
```
Index(['make', 'fuel-type', 'num-of-doors', 'body-style', 'drive-wheels',
       'curb-weight', 'engine-size', 'highway-mpg', 'price'],
      dtype='object')
```

In [6]:
```
y=cars['price']
X=cars.drop(columns=['price','make','fuel-type','num-of-doors','body-style','drive-wheels'])
```

linear regression

In [7]:
```
from sklearn.linear_model import LinearRegression
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
```

In [8]:
```
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.3,random_state=0)
```

In [11]:
```
lr_model=LinearRegression()
lr_model.fit(X_train,y_train)
```

Out[11]:
```
LinearRegression()
```

In [14]:
```
for t in zip(X.columns,lr_model.coef_):
    print(f"{t[0]:25s}{t[1]:10.2f}")
```

```
curb-weight                   6.97
engine-size                  72.95
highway-mpg                 -43.71
```

In [16]:
```
y_pred=lr_model.predict(X_test)
```

In [17]:
```
mse=mean_squared_error(y_test,y_pred)
print("MSE:",mse)
print("RMSE:",np.sqrt(mse))
```

```
MSE: 21997867.032136828
RMSE: 4690.188379173786
```

ridge regression

In [18]:
```
from sklearn.linear_model import Ridge
from sklearn.model_selection import train_test_split
```

In [19]:
```
ridge=Ridge(normalize=True,alpha=3.0)
ridge.fit(X_train,y_train)
```

Out[19]:
```
Ridge(alpha=3.0, normalize=True)
```

In [20]:
```
# Display coefficient for each column
for t in zip(X.columns, ridge.coef_):
  print(f"{t[0]:25s} {t[1]:10.2f}")
```

```
curb-weight                   2.26
engine-size                  28.75
highway-mpg                -123.89
```

In [21]:
```
y_pred=ridge.predict(X_test)
```

In [22]:
```
mse = mean_squared_error(y_test,y_pred)
```

```
print("MSE  : ", mse)
print("RMSE : ", np.sqrt(mse))
```

```
MSE  :  47608942.17753899
RMSE :  6899.92334577269
```

Lasso CV

In [25]:
```
from sklearn.linear_model import LassoCV
from sklearn.metrics import r2_score
```

In [26]:
```
lm = LassoCV(normalize = True, cv=5,alphas=(3,4,5,6,7))
lm.fit(X,y)
```

Out[26]:
```
LassoCV(alphas=(3, 4, 5, 6, 7), cv=5, normalize=True)
```

In [29]:
```
lm.alpha_
```

Out[29]:
```
7
```

In [30]:
```
for t in zip(X.columns, lm.coef_):
  print(f"{t[0]:25s} {t[1]:10.2f}")
```

```
curb-weight                    3.88
engine-size                  110.69
highway-mpg                 -114.46
```

In [31]:
```
y_test  = y[:50]
X_test = X[:50]
y_pred = lm.predict(X_test)
```

In [32]:
```
from sklearn.metrics import mean_squared_error
mse = mean_squared_error(y_test,y_pred)
print("MSE  : ",mse)
print("RMSE : ", np.sqrt(mse))
```

```
MSE  :  14012276.660115926
RMSE :  3743.2975649974614
```

In [33]:

```
r2score = r2_score(y_test,y_pred)
print(f"R2 Score: {r2score:0.2f}")
```

```
R2 Score: 0.85
```

In [ ]: