[35]:
```
import pandas as pd
import numpy as np
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
import seaborn as sns
import matplotlib.pyplot as plt
```

[39]:
```
# Specify the file path
file = "D:\\intership\\New folder\\car prediction\\CarPrice_Assignment.csv"


data = pd.read_csv(file)

data.head()
```

[39]:

| | car_ID | symboling | CarName | fueltype | aspiration | doornumber | carbody | drivewheel | enginelocation | wheelbase | ... | enginesize | fuelsystem | boreratio | stroke | co |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 3 | alfa-romero giulia | gas | std | two | convertible | rwd | front | 88.6 | ... | 130 | mpfi | 3.47 | 2.68 | |
| **1** | 2 | 3 | alfa-romero stelvio | gas | std | two | convertible | rwd | front | 88.6 | ... | 130 | mpfi | 3.47 | 2.68 | |
| **2** | 3 | 1 | alfa-romero Quadrifoglio | gas | std | two | hatchback | rwd | front | 94.5 | ... | 152 | mpfi | 2.68 | 3.47 | |
| **3** | 4 | 2 | audi 100 ls | gas | std | four | sedan | fwd | front | 99.8 | ... | 109 | mpfi | 3.19 | 3.40 | |
| **4** | 5 | 2 | audi 100ls | gas | std | four | sedan | 4wd | front | 99.4 | ... | 136 | mpfi | 3.19 | 3.40 | |

5 rows × 26 columns

[40]:
```
# Check missing values
data.isnull().sum()
```

[40]:
```
car_ID            0
symboling         0
CarName           0
fueltype          0
aspiration        0
doornumber        0
carbody           0
drivewheel        0
enginelocation    0
wheelbase         0
carlength         0
carwidth          0
carheight         0
curbweight        0
enginetype        0
cylindernumber    0
enginesize        0
fuelsystem        0
boreratio         0
stroke            0
compressionratio  0
horsepower        0
peakrpm           0
citympg           0
highwaympg        0
price             0
dtype: int64
```

[42]:
```
# Handle missing values
data = data.dropna()
data
```

[42]:

| | car_ID | symboling | CarName | fueltype | aspiration | doornumber | carbody | drivewheel | enginelocation | wheelbase | ... | enginesize | fuelsystem | boreratio | stroke |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 3 | alfa-romero giulia | gas | std | two | convertible | rwd | front | 88.6 | ... | 130 | mpfi | 3.47 | 2.68 |
| **1** | 2 | 3 | alfa-romero stelvio | gas | std | two | convertible | rwd | front | 88.6 | ... | 130 | mpfi | 3.47 | 2.68 |
| **2** | 3 | 1 | alfa-romero Quadrifoglio | gas | std | two | hatchback | rwd | front | 94.5 | ... | 152 | mpfi | 2.68 | 3.47 |
| **3** | 4 | 2 | audi 100 ls | gas | std | four | sedan | fwd | front | 99.8 | ... | 109 | mpfi | 3.19 | 3.40 |
| **4** | 5 | 2 | audi 100ls | gas | std | four | sedan | 4wd | front | 99.4 | ... | 136 | mpfi | 3.19 | 3.40 |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... |
| **200** | 201 | -1 | volvo 145e (sw) | gas | std | four | sedan | rwd | front | 109.1 | ... | 141 | mpfi | 3.78 | 3.15 |
| **201** | 202 | -1 | volvo 144ea | gas | turbo | four | sedan | rwd | front | 109.1 | ... | 141 | mpfi | 3.78 | 3.15 |
| **202** | 203 | -1 | volvo 244dl | gas | std | four | sedan | rwd | front | 109.1 | ... | 173 | mpfi | 3.58 | 2.87 |
| **203** | 204 | -1 | volvo 246 | diesel | turbo | four | sedan | rwd | front | 109.1 | ... | 145 | idi | 3.01 | 3.40 |
| **204** | 205 | -1 | volvo 264gl | gas | turbo | four | sedan | rwd | front | 109.1 | ... | 141 | mpfi | 3.78 | 3.15 |

205 rows × 26 columns

[49]:
```
# Convert categorical variables to numerical values using one-hot encoding
data = pd.get_dummies(data, drop_first=True)

data.columns
```

[49]:
```
Index(['car_ID', 'symboling', 'wheelbase', 'carlength', 'carwidth',
```

```
       'carheight', 'curbweight', 'enginesize', 'boreratio', 'stroke',
       ...
       'cylindernumber_three', 'cylindernumber_twelve', 'cylindernumber_two',
       'fuelsystem_2bbl', 'fuelsystem_4bbl', 'fuelsystem_idi',
       'fuelsystem_mfi', 'fuelsystem_mpfi', 'fuelsystem_spdi',
       'fuelsystem_spfi'],
      dtype='object', length=191)
```

[50]:
```
data
```

[50]:

| | car_ID | symboling | wheelbase | carlength | carwidth | carheight | curbweight | enginesize | boreratio | stroke | ... | cylindernumber_three | cylindernumber_twelve | cylindern |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| **0** | 1 | 3 | 88.6 | 168.8 | 64.1 | 48.8 | 2548 | 130 | 3.47 | 2.68 | ... | False | False | |
| **1** | 2 | 3 | 88.6 | 168.8 | 64.1 | 48.8 | 2548 | 130 | 3.47 | 2.68 | ... | False | False | |
| **2** | 3 | 1 | 94.5 | 171.2 | 65.5 | 52.4 | 2823 | 152 | 2.68 | 3.47 | ... | False | False | |
| **3** | 4 | 2 | 99.8 | 176.6 | 66.2 | 54.3 | 2337 | 109 | 3.19 | 3.40 | ... | False | False | |
| **4** | 5 | 2 | 99.4 | 176.6 | 66.4 | 54.3 | 2824 | 136 | 3.19 | 3.40 | ... | False | False | |
| **...** | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | ... | |
| **200** | 201 | -1 | 109.1 | 188.8 | 68.9 | 55.5 | 2952 | 141 | 3.78 | 3.15 | ... | False | False | |
| **201** | 202 | -1 | 109.1 | 188.8 | 68.8 | 55.5 | 3049 | 141 | 3.78 | 3.15 | ... | False | False | |
| **202** | 203 | -1 | 109.1 | 188.8 | 68.9 | 55.5 | 3012 | 173 | 3.58 | 2.87 | ... | False | False | |
| **203** | 204 | -1 | 109.1 | 188.8 | 68.9 | 55.5 | 3217 | 145 | 3.01 | 3.40 | ... | False | False | |
| **204** | 205 | -1 | 109.1 | 188.8 | 68.9 | 55.5 | 3062 | 141 | 3.78 | 3.15 | ... | False | False | |

205 rows × 191 columns

*[52]:
```
# Define features and target variable
X = data.drop(columns=['price'])
y = data['price']

# Display the features and target variable
X
```

[52]:
```
0      13495.0
1      16500.0
2      16500.0
3      13950.0
4      17450.0
        ...
200    16845.0
201    19045.0
202    21485.0
203    22470.0
204    22625.0
Name: price, Length: 205, dtype: float64
```

[53]:
```
y
```

[53]:
```
0      13495.0
1      16500.0
2      16500.0
3      13950.0
4      17450.0
        ...
200    16845.0
201    19045.0
202    21485.0
203    22470.0
204    22625.0
Name: price, Length: 205, dtype: float64
```

[54]:
```
# Split the data into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=42)
```

[56]:
```
# Initialize the Linear Regression model
model = LinearRegression()

# Train the model
model.fit(X_train, y_train)
```

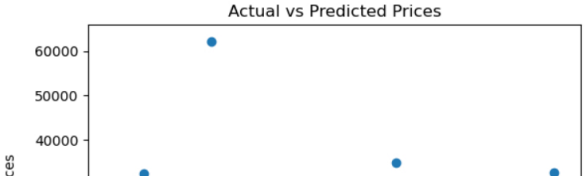
[56]:
```
▾ LinearRegression
LinearRegression()
```

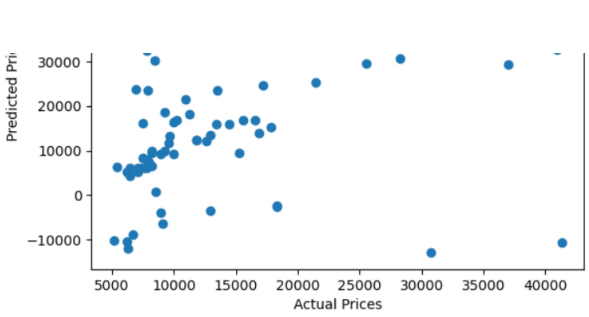
[58]:
```
# Make predictions on the test set
y_pred = lr_model.predict(X_test)

# Calculate the Mean Squared Error (MSE) and R^2 score
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
print(f'Mean Squared Error: {mse:.2f}')
print(f'R^2 Score: {r2}')

# Plot the actual vs predicted prices
plt.scatter(y_test, y_pred)
plt.xlabel('Actual Prices')
plt.ylabel('Predicted Prices')
plt.title('Actual vs Predicted Prices')
plt.show()
```

```
Mean Squared Error: 196438429.48
R^2 Score: -1.8352484848727344
```




Predicted Pri
30000
20000
10000
0
−10000
5000
10000
15000
20000
25000
30000
35000
40000
Actual Prices

[ ]:

[ ]: