In [1]:

```python
# Experiment No: 9 - Create a linear regression model using housing dataset
```

In [1]:

```python
# Importing Libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_squared_error, r2_score
```

In [2]:

```python
# Loading the Boston housing dataset from the original source
data_url = "http://lib.stat.cmu.edu/datasets/boston"
raw_df = pd.read_csv(data_url, sep="\s+", skiprows=22, header=None)
```

In [3]:

```python
# Extracting features and target variable
# The dataset contains 13 features and the target variable is the median val
X = np.hstack([raw_df.values[::2, :], raw_df.values[1::2, :2]])
y = raw_df.values[1::2, 2]
```

In [4]:

```python
# Converting to DataFrame for easier manipulation
feature_names = ['CRIM', 'ZN', 'INDUS', 'CHAS', 'NOX', 'RM', 'AGE', 'DIS', '
X = pd.DataFrame(X, columns=feature_names)
```

In [5]:

```python
# Printing data types of the features
print("\nData types of the features:")
print(X.dtypes)
```

```
Data types of the features:
CRIM       float64
ZN         float64
INDUS      float64
CHAS       float64
NOX        float64
RM         float64
AGE        float64
DIS        float64
RAD        float64
TAX        float64
PTRATIO    float64
B          float64
LSTAT      float64
dtype: object
```

In [6]:

```python
# Displaying the head of the dataset
print("First 5 rows of the Boston Housing dataset:")
print(X.head())
```

```
First 5 rows of the Boston Housing dataset:
      CRIM    ZN  INDUS  CHAS    NOX     RM   AGE     DIS  RAD    TAX  \
0  0.00632  18.0   2.31   0.0  0.538  6.575  65.2  4.0900  1.0  296.0
1  0.02731   0.0   7.07   0.0  0.469  6.421  78.9  4.9671  2.0  242.0
2  0.02729   0.0   7.07   0.0  0.469  7.185  61.1  4.9671  2.0  242.0
3  0.03237   0.0   2.18   0.0  0.458  6.998  45.8  6.0622  3.0  222.0
4  0.06905   0.0   2.18   0.0  0.458  7.147  54.2  6.0622  3.0  222.0

   PTRATIO       B  LSTAT
0     15.3  396.90   4.98
1     17.8  396.90   9.14
2     17.8  392.83   4.03
3     18.7  394.63   2.94
4     18.7  396.90   5.33
```

In [7]:

```python
# Splitting the dataset into training and testing sets
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, rar
```

In [8]:

```python
# Creating a linear regression model
model = LinearRegression()
model.fit(X_train, y_train)  # Fitting the model
```

Out[8]:

```
▾ LinearRegression ⓘ ?

LinearRegression()
```

In [9]:

```python
# Making predictions on the test set
y_pred = model.predict(X_test)
```

In [10]:

```python
# Evaluating the model's performance
mse = mean_squared_error(y_test, y_pred)
r2 = r2_score(y_test, y_pred)
```

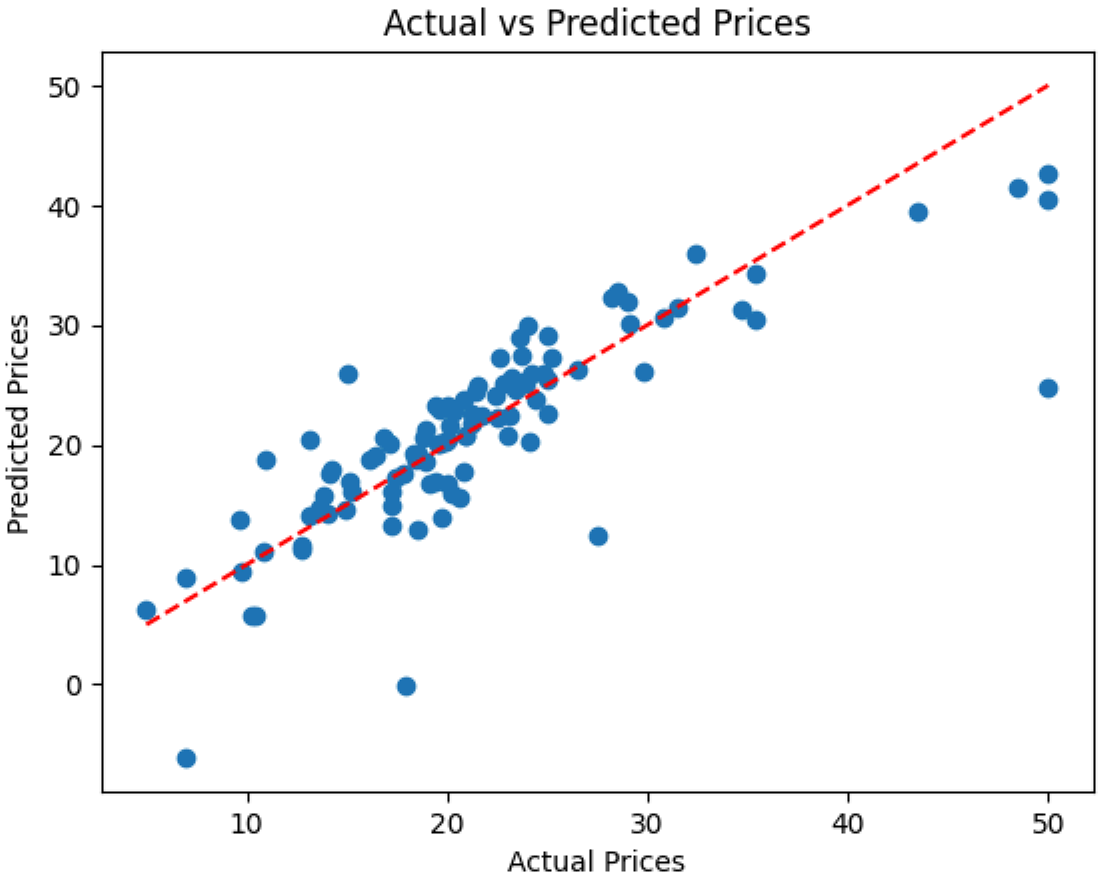
In [11]:

```python
# Displaying the results
print("\nMean Squared Error:", mse)
print("R^2 Score:", r2)
```

```
Mean Squared Error: 24.291119474973478
R^2 Score: 0.6687594935356326
```

In [12]:

```python
# Visualizing the predictions
plt.scatter(y_test, y_pred)
plt.xlabel('Actual Prices')
plt.ylabel('Predicted Prices')
plt.title('Actual vs Predicted Prices')
plt.plot([y.min(), y.max()], [y.min(), y.max()], 'r--')  # Diagonal line
plt.show()
```


Actual vs Predicted Prices
Predicted Prices
Actual Prices
50
40
30
20
10
0
10
20
30
40
50