In [ ]:
```
"C:\Users\DELL\Downloads\kc_house_data.csv.zip"
```

In [1]:
```
pip install pandas scikit-learn
```

```
Requirement already satisfied: pandas in c:\users\dell\anaconda3\lib\site-pac
kages (2.0.3)
Requirement already satisfied: scikit-learn in c:\users\dell\anaconda3\lib\si
te-packages (1.4.2)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\dell\anacon
da3\lib\site-packages (from pandas) (2.8.2)
Requirement already satisfied: pytz>=2020.1 in c:\users\dell\anaconda3\lib\si
te-packages (from pandas) (2023.3.post1)
Requirement already satisfied: tzdata>=2022.1 in c:\users\dell\anaconda3\lib
\site-packages (from pandas) (2023.3)
Requirement already satisfied: numpy>=1.21.0 in c:\users\dell\anaconda3\lib\s
ite-packages (from pandas) (1.24.3)
Requirement already satisfied: scipy>=1.6.0 in c:\users\dell\anaconda3\lib\si
te-packages (from scikit-learn) (1.11.1)
Requirement already satisfied: joblib>=1.2.0 in c:\users\dell\anaconda3\lib\s
ite-packages (from scikit-learn) (1.2.0)
Requirement already satisfied: threadpoolctl>=2.0.0 in c:\users\dell\anaconda
3\lib\site-packages (from scikit-learn) (2.2.0)
Requirement already satisfied: six>=1.5 in c:\users\dell\anaconda3\lib\site-p
ackages (from python-dateutil>=2.8.2->pandas) (1.16.0)
Note: you may need to restart the kernel to use updated packages.
```

In [5]:
```
import zipfile
import pandas as pd
import os

# Path to the zip file
zip_path = 'C:/Users/DELL/Downloads/kc_house_data.csv.zip'

# Path to extract the file (ensure this directory exists or create it)
unzip_path = 'C:/Users/DELL/Downloads/kc_house_data/'

# Extract the zip file
with zipfile.ZipFile(zip_path, 'r') as zip_ref:
    zip_ref.extractall(unzip_path)

# Path to the CSV file inside the extracted folder
csv_path = os.path.join(unzip_path, 'kc_house_data.csv')

# Load the dataset
housing_data = pd.read_csv(csv_path)

# Display the first few rows of the dataset to ensure it loaded correctly
print(housing_data.head())
```

```
           id             date     price  bedrooms  bathrooms  sqft_living  \
0  7129300520  20141013T000000  221900.0         3       1.00         1180
1  6414100192  20141209T000000  538000.0         3       2.25         2570
2  5631500400  20150225T000000  180000.0         2       1.00          770
3  2487200875  20141209T000000  604000.0         4       3.00         1960
4  1954400510  20150218T000000  510000.0         3       2.00         1680

   sqft_lot  floors  waterfront  view  ...  grade  sqft_above  sqft_basement
\
0      5650     1.0           0     0  ...      7        1180              0
1      7242     2.0           0     0  ...      7        2170            400
2     10000     1.0           0     0  ...      6         770              0
3      5000     1.0           0     0  ...      7        1050            910
4      8080     1.0           0     0  ...      8        1680              0

   yr_built  yr_renovated  zipcode      lat     long  sqft_living15  \
0      1955             0    98178  47.5112 -122.257           1340
1      1951          1991    98125  47.7210 -122.319           1690
2      1933             0    98028  47.7379 -122.233           2720
3      1965             0    98136  47.5208 -122.393           1360
4      1987             0    98074  47.6168 -122.045           1800

   sqft_lot15
0        5650
1        7639
2        8062
3        5000
4        7503

[5 rows x 21 columns]
```

In [6]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

# Select relevant features
features = ['bedrooms', 'bathrooms', 'sqft_living', 'sqft_lot', 'floors', 'wate
            'view', 'condition', 'grade', 'sqft_above', 'sqft_basement', 'yr_bu
            'yr_renovated', 'zipcode', 'lat', 'long', 'sqft_living15', 'sqft_lo
target = 'price'

# Split the data into training and testing sets
X = housing_data[features]
y = housing_data[target]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random
```

In [7]:
```
# Train the Linear Regression model
model = LinearRegression()
model.fit(X_train, y_train)
```

Out[7]:
```
LinearRegression
LinearRegression()
```
(https://scikit-learn.org/1.4/modules/generated/sklearn.linear_model.LinearRegress

In [16]:

```
import zipfile
import pandas as pd
import os
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score

# Path to the zip file
zip_path = 'C:/Users/DELL/Downloads/kc_house_data.csv.zip'

# Path to extract the file (ensure this directory exists or create it)
unzip_path = 'C:/Users/DELL/Downloads/kc_house_data/'

# Create the directory if it doesn't exist
if not os.path.exists(unzip_path):
    os.makedirs(unzip_path)

# Extract the zip file
with zipfile.ZipFile(zip_path, 'r') as zip_ref:
    zip_ref.extractall(unzip_path)

# Path to the CSV file inside the extracted folder
csv_path = os.path.join(unzip_path, 'kc_house_data.csv')

# Load the dataset
housing_data = pd.read_csv(csv_path)

# Display the first few rows of the dataset to ensure it loaded correctly
print(housing_data.head())

# Check for missing values
missing_values = housing_data.isnull().sum()
print(missing_values)

# Select relevant features
features = ['bedrooms', 'bathrooms', 'sqft_living', 'sqft_lot', 'floors', 'wate
            'view', 'condition', 'grade', 'sqft_above', 'sqft_basement', 'yr_bu
            'yr_renovated', 'zipcode', 'lat', 'long', 'sqft_living15', 'sqft_lo
target = 'price'

# Split the data into training and testing sets
X = housing_data[features]
y = housing_data[target]
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random

# Train the Linear Regression model
model = LinearRegression()
model.fit(X_train, y_train)

# Predict on the test set
y_pred = model.predict(X_test)

# Evaluate the model
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = mse ** 0.5  # Calculate RMSE from MSE
r2 = r2_score(y_test, y_pred)
```

```
print(f'MAE: {mae}')
print(f'MSE: {mse}')
print(f'RMSE: {rmse}')
print(f'R²: {r2}')
```

```
           id             date     price  bedrooms  bathrooms  sqft_living  \
0  7129300520  20141013T000000  221900.0         3       1.00         1180
1  6414100192  20141209T000000  538000.0         3       2.25         2570
2  5631500400  20150225T000000  180000.0         2       1.00          770
3  2487200875  20141209T000000  604000.0         4       3.00         1960
4  1954400510  20150218T000000  510000.0         3       2.00         1680

   sqft_lot  floors  waterfront  view  ...  grade  sqft_above  sqft_basement
\
0      5650     1.0           0     0  ...      7        1180              0
1      7242     2.0           0     0  ...      7        2170            400
2     10000     1.0           0     0  ...      6         770              0
3      5000     1.0           0     0  ...      7        1050            910
4      8080     1.0           0     0  ...      8        1680              0

   yr_built  yr_renovated  zipcode      lat     long  sqft_living15  \
0      1955             0    98178  47.5112 -122.257           1340
1      1951          1991    98125  47.7210 -122.319           1690
2      1933             0    98028  47.7379 -122.233           2720
3      1965             0    98136  47.5208 -122.393           1360
4      1987             0    98074  47.6168 -122.045           1800

   sqft_lot15
0        5650
1        7639
2        8062
3        5000
4        7503

[5 rows x 21 columns]
id               0
date             0
price            0
bedrooms         0
bathrooms        0
sqft_living      0
sqft_lot         0
floors           0
waterfront       0
view             0
condition        0
grade            0
sqft_above       0
sqft_basement    0
yr_built         0
yr_renovated     0
zipcode          0
lat              0
long             0
sqft_living15    0
sqft_lot15       0
dtype: int64
MAE: 127493.34208658228
MSE: 45173046132.79252
RMSE: 212539.51663818312
R²: 0.7011904448878257
```

In [ ]: