Import Libraries and Dataset

In [19]:

```
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 matplotlib.pyplot as plt

dataset = pd.read_csv('house price data.csv')

print(dataset.head())
```

```
                  date      price  bedrooms  bathrooms  sqft_living  sqft_lot  \
0  2014-05-02 00:00:00   313000.0       3.0       1.50         1340      7912   
1  2014-05-02 00:00:00  2384000.0       5.0       2.50         3650      9050   
2  2014-05-02 00:00:00   342000.0       3.0       2.00         1930     11947   
3  2014-05-02 00:00:00   420000.0       3.0       2.25         2000      8030   
4  2014-05-02 00:00:00   550000.0       4.0       2.50         1940     10500   

   floors  waterfront  view  condition  sqft_above  sqft_basement  yr_built  \
0     1.5           0     0          3        1340              0      1955   
1     2.0           0     4          5        3370            280      1921   
2     1.0           0     0          4        1930              0      1966   
3     1.0           0     0          4        1000           1000      1963   
4     1.0           0     0          4        1140            800      1976   

   yr_renovated                    street       city  statezip country  
0          2005      18810 Densmore Ave N  Shoreline  WA 98133     USA  
1             0           709 W Blaine St    Seattle  WA 98119     USA  
2             0  26206-26214 143rd Ave SE       Kent  WA 98042     USA  
3             0           857 170th Pl NE   Bellevue  WA 98008     USA  
4          1992         9105 170th Ave NE    Redmond  WA 98052     USA  
```

Data Preprocessing

In [20]:

```
dataset['date'] = pd.to_datetime(dataset['date'])

dataset['year'] = dataset['date'].dt.year
dataset['month'] = dataset['date'].dt.month
dataset['day'] = dataset['date'].dt.day

dataset = dataset.drop('date', axis=1)

categorical_cols = dataset.select_dtypes(include=['object']).columns
dataset = pd.get_dummies(dataset, columns=categorical_cols, drop_first=True)

dataset = dataset.dropna()

dataset = dataset.drop_duplicates()

X = dataset.drop('price', axis=1)
y = dataset['price']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

print(f'Training data shape: {X_train.shape}, Training target shape: {y_train.shape}')
print(f'Testing data shape: {X_test.shape}, Testing target shape: {y_test.shape}')
```

```
Training data shape: (3680, 4658), Training target shape: (3680,)
Testing data shape: (920, 4658), Testing target shape: (920,)
```

Implement Model

In [21]:

```
model = LinearRegression()

model.fit(X_train, y_train)

y_train_pred = model.predict(X_train)

train_mse = mean_squared_error(y_train, y_train_pred)
train_r2 = r2_score(y_train, y_train_pred)

print(f'Training Mean Squared Error: {train_mse}')
print(f'Training R^2 Score: {train_r2}')
```

```
Training Mean Squared Error: 167122395.32884604
Training R^2 Score: 0.9988237932045246
```

Model Evaluation

In [22]:

```
y_test_pred = model.predict(X_test)

test_mse = mean_squared_error(y_test, y_test_pred)
test_r2 = r2_score(y_test, y_test_pred)

print(f'Testing Mean Squared Error: {test_mse}')
print(f'Testing R^2 Score: {test_r2}')
```

```
Testing Mean Squared Error: 4605084471985.507
Testing R^2 Score: -3.5154689397550483
```

Testing

In [23]:

```
new_test_dataset = pd.read_csv('house price data.csv')

new_test_dataset = new_test_dataset.dropna()
new_test_dataset = new_test_dataset.drop_duplicates()

categorical_cols_new = new_test_dataset.select_dtypes(include=['object']).columns
new_test_dataset = pd.get_dummies(new_test_dataset, columns=categorical_cols_new, drop_first=True)

missing_cols = set(X.columns) - set(new_test_dataset.columns)
for col in missing_cols:
    new_test_dataset[col] = 0
new_test_dataset = new_test_dataset[X.columns]


X_new_test = new_test_dataset.drop('price', axis=1, errors='ignore')

y_new_test_pred = model.predict(X_new_test)

print(y_new_test_pred)
```

```
[ 309088.77655955 2380088.77655477  338088.77651395 ...  404823.68955865
  190254.22534231   90237.59306432]
```