Read the data

In [1]:

```
import numpy as np
import pandas as pd
```

In [4]:

```
df = pd.read_csv('house_data.csv')
df.head(6)
```

Out[4]:

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	...	grade	sqft_above	sqft_basem
0	7129300520	20141013T000000	221900.0	3	1.00	1180	5650	1.0	0	0	...	7	1180	
1	6414100192	20141209T000000	538000.0	3	2.25	2570	7242	2.0	0	0	...	7	2170	
2	5631500400	20150225T000000	180000.0	2	1.00	770	10000	1.0	0	0	...	6	770	
3	2487200875	20141209T000000	604000.0	4	3.00	1960	5000	1.0	0	0	...	7	1050	
4	1954400510	20150218T000000	510000.0	3	2.00	1680	8080	1.0	0	0	...	8	1680	
5	7237550310	20140512T000000	1225000.0	4	4.50	5420	101930	1.0	0	0	...	11	3890	1

6 rows × 21 columns

Clean the data

In [5]:

```
df.isnull().sum()
```

Out[5]:

```
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
```

Feature engineering

In [8]:

```
df.columns
```

Out[8]:

```
Index(['id', 'date', 'price', 'bedrooms', 'bathrooms', 'sqft_living',
       'sqft_lot', 'floors', 'waterfront', 'view', 'condition', 'grade',
       'sqft_above', 'sqft_basement', 'yr_built', 'yr_renovated', 'zipcode',
       'lat', 'long', 'sqft_living15', 'sqft_lot15'],
      dtype='object')
```

In [9]:

```
df = df.drop(['id','date'], axis=1)
df.head(6)
```

Out[9]:

	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	grade	sqft_above	sqft_basement	yr_built	yr_renovat
0	221900.0	3	1.00	1180	5650	1.0	0	0	3	7	1180	0	1955	
1	538000.0	3	2.25	2570	7242	2.0	0	0	3	7	2170	400	1951	19
2	180000.0	2	1.00	770	10000	1.0	0	0	3	6	770	0	1933	
3	604000.0	4	3.00	1960	5000	1.0	0	0	5	7	1050	910	1965	
4	510000.0	3	2.00	1680	8080	1.0	0	0	3	8	1680	0	1987	
5	1225000.0	4	4.50	5420	101930	1.0	0	0	3	11	3890	1530	2001	

Preparing the data

In [10]:

```
x = df.drop('price',axis=1)
y = df['price']
x.head(6)
```

Out[10]:

	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	grade	sqft_above	sqft_basement	yr_built	yr_renovated	zipcode
0	3	1.00	1180	5650	1.0	0	0	3	7	1180	0	1955	0	98178
1	3	2.25	2570	7242	2.0	0	0	3	7	2170	400	1951	1991	98125
2	2	1.00	770	10000	1.0	0	0	3	6	770	0	1933	0	98028
3	4	3.00	1960	5000	1.0	0	0	5	7	1050	910	1965	0	98136
4	3	2.00	1680	8080	1.0	0	0	3	8	1680	0	1987	0	98074
5	4	4.50	5420	101930	1.0	0	0	3	11	3890	1530	2001	0	98053

In [11]:
```
y.head(6)
```

Out[11]:
```
0     221900.0
1     538000.0
2     180000.0
3     604000.0
4     510000.0
5    1225000.0
Name: price, dtype: float64
```

In [12]:
```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split(x,y,test_size=0.2)
x_train.shape, x_test.shape, y_train.shape, y_test.shape
```

Out[12]:
```
((17290, 18), (4323, 18), (17290,), (4323,))
```

Building model

In [16]:
```
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
```

In [17]:
```
lr.fit(x_train,y_train)
```

Out[17]:
```
▾ LinearRegression

LinearRegression()
```

In [20]:
```
lr.intercept_
```

Out[20]:
```
9139949.72571687
```

In [21]:
```
lr.coef_
```

Out[21]:
```
array([-4.00374206e+04,  4.09444415e+04,  1.13783788e+02,  1.46724256e-01,
        8.25390097e+03,  5.86575816e+05,  5.20534078e+04,  2.58942501e+04,
        9.66058334e+04,  6.78884178e+01,  4.58953698e+01, -2.66742658e+03,
        1.97306722e+01, -5.95041998e+02,  6.00345921e+05, -2.06526687e+05,
        2.40971618e+01, -4.36851210e-01])
```

Testing

In [22]:
```
y_pred = lr.predict(x_test)
y_pred
```

Out[22]:
```
array([ 174055.44188106, 1149487.71114055,  453979.34174441, ...,
        212350.57831155,  925387.66507674,  325570.19610983])
```

In [23]:
```
y_test
```

Out[23]:
```
11196    209000.0
6805     890000.0
19785    290000.0
4603     305000.0
19234    200000.0
           ...
6485     710000.0
20083    388000.0
15555    175000.0
6189     738515.0
9782     265000.0
Name: price, Length: 4323, dtype: float64
```

In [24]:
```
from sklearn.metrics import mean_absolute_error
print("MAE", mean_absolute_error(y_pred,y_test))
```

```
MAE 126421.84812385455
```

In [ ]:
```

```