

Import Libraries

```
In [136... import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
import warnings
warnings.filterwarnings('ignore')
```

Load the Dataset

```
In [139... data = pd.read_csv(r"D:\FSDS Material\Dataset\House_data.csv")
```

```
In [141... data
```

Out[141...

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	...	grade
0	7129300520	20141013T000000	221900.0	3	1.00	1180	5650	1.0	0	0	...	7
1	6414100192	20141209T000000	538000.0	3	2.25	2570	7242	2.0	0	0	...	7
2	5631500400	20150225T000000	180000.0	2	1.00	770	10000	1.0	0	0	...	6
3	2487200875	20141209T000000	604000.0	4	3.00	1960	5000	1.0	0	0	...	7
4	1954400510	20150218T000000	510000.0	3	2.00	1680	8080	1.0	0	0	...	8
...
21608	263000018	20140521T000000	360000.0	3	2.50	1530	1131	3.0	0	0	...	8
21609	6600060120	20150223T000000	400000.0	4	2.50	2310	5813	2.0	0	0	...	8
21610	1523300141	20140623T000000	402101.0	2	0.75	1020	1350	2.0	0	0	...	7
21611	291310100	20150116T000000	400000.0	3	2.50	1600	2388	2.0	0	0	...	8
21612	1523300157	20141015T000000	325000.0	2	0.75	1020	1076	2.0	0	0	...	7

21613 rows × 21 columns



Check the columns

In [144...

data.columns

Out[144...

```
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')
```

Check the information

In [147... `data.info()`

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 21613 entries, 0 to 21612
Data columns (total 21 columns):
#   Column                Non-Null Count  Dtype
---  -
0   id                     21613 non-null  int64
1   date                   21613 non-null  object
2   price                  21613 non-null  float64
3   bedrooms               21613 non-null  int64
4   bathrooms              21613 non-null  float64
5   sqft_living            21613 non-null  int64
6   sqft_lot               21613 non-null  int64
7   floors                 21613 non-null  float64
8   waterfront             21613 non-null  int64
9   view                   21613 non-null  int64
10  condition              21613 non-null  int64
11  grade                  21613 non-null  int64
12  sqft_above             21613 non-null  int64
13  sqft_basement          21613 non-null  int64
14  yr_built                21613 non-null  int64
15  yr_renovated            21613 non-null  int64
16  zipcode                21613 non-null  int64
17  lat                    21613 non-null  float64
18  long                   21613 non-null  float64
19  sqft_living15           21613 non-null  int64
20  sqft_lot15             21613 non-null  int64
dtypes: float64(5), int64(15), object(1)
memory usage: 3.5+ MB
```

Check the rows & columns

In [150... `data.shape`

Out[150... (21613, 21)

In [152... `data.head()`

Out[152...

	id	date	price	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	...	grade	sq
0	7129300520	20141013T000000	221900.0	3	1.00	1180	5650	1.0	0	0	...	7	
1	6414100192	20141209T000000	538000.0	3	2.25	2570	7242	2.0	0	0	...	7	
2	5631500400	20150225T000000	180000.0	2	1.00	770	10000	1.0	0	0	...	6	
3	2487200875	20141209T000000	604000.0	4	3.00	1960	5000	1.0	0	0	...	7	
4	1954400510	20150218T000000	510000.0	3	2.00	1680	8080	1.0	0	0	...	8	

5 rows × 21 columns



In [154...

```
# find the number of missing values
data.isnull().sum()
```

Out[154...

```
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
```

```
In [156... data.duplicated().sum() # Here is no null values in the dadaset
```

```
Out[156... 0
```

```
In [158... space = data['sqft_living']  
price = data['price']
```

```
In [160... x = np.array(space).reshape(-1, 1)  
y = np.array(price)
```

Split the Data into Training and Testing Sets

```
In [163... 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, random_state=42)
```

Print each split data

```
In [166... print("x_train:\n", x_train)  
print("\nx_test:\n", x_test)  
print("\ny_train:\n", y_train)  
print("\ny_test:\n", y_test)
```

```
x_train:
[[1780]
 [1000]
 [1080]
 ...
 [2120]
 [ 380]
 [3130]]

x_test:
[[2070]
 [2900]
 [3770]
 ...
 [2434]
 [2250]
 [2570]]

y_train:
[325000. 257000. 228500. ... 575000. 245000. 315000.]

y_test:
[ 365000.  865000. 1040000. ... 285000.  605000.  356500.]
```

Fitting simple linear regressin to the Train the Model

```
In [169... from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(x_train, y_train)
```

Out[169...

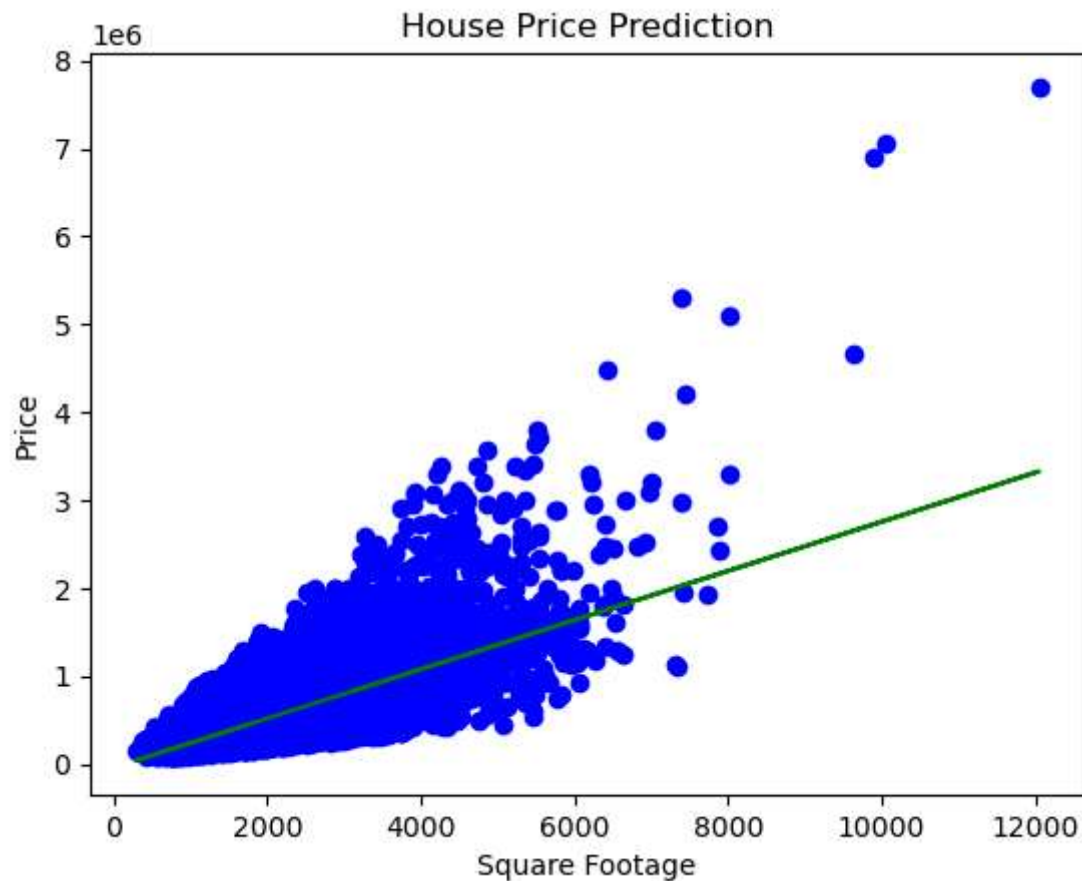
```
LinearRegression
LinearRegression()
```

Prediction of the price

```
In [172... pred = regressor.predict(x_test)
```

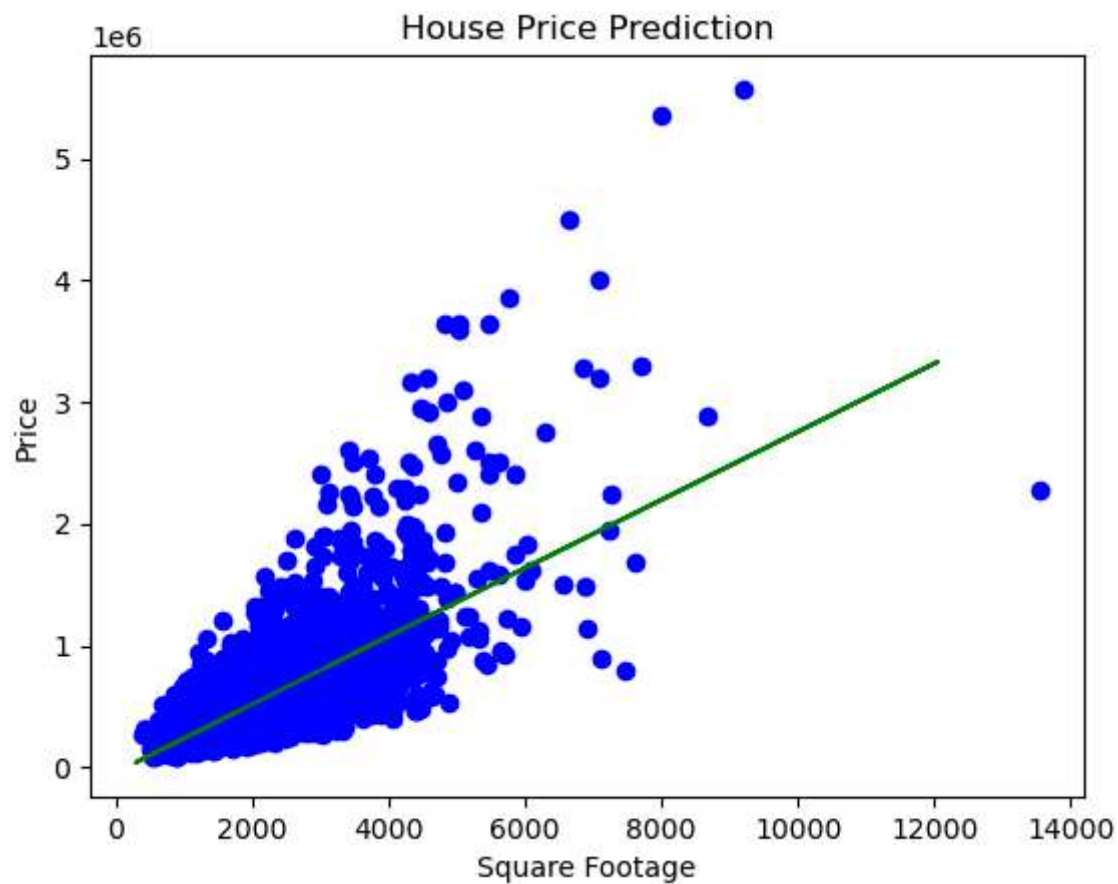
Visualizing the train results

```
In [175... plt.scatter(x_train, y_train, color='blue')  
plt.plot(x_train, regressor.predict(x_train), color='green')  
plt.title('House Price Prediction')  
plt.xlabel('Square Footage')  
plt.ylabel('Price')  
plt.show()
```



Visualizing the test result

```
In [178... plt.scatter(x_test, y_test, color = 'blue')  
plt.plot(x_train, regressor.predict(x_train), color = 'green')  
plt.title('House Price Prediction')  
plt.xlabel('Square Footage')  
plt.ylabel('Price')  
plt.show()
```



```
In [182... new_sqft = np.array([[2800]])  
predicted_price = regressor.predict(new_sqft)
```



```
print(f"Predicted price for house with 1076 sq ft: ₹{predicted_price[0]:,.2f}")
```

Predicted price for house with 1076 sq ft: ₹740,980.25

```
In [184... print(f'Predicted price for house with 1800 sq ft: ₹{predicted_price[0]:,.2f}')
```

Predicted price for house with 1800 sq ft: ₹740,980.25

```
In [188... print(f'Predicted price for flat with 2000 sq ft: ₹{predicted_price[0]:,.2f}')
```

Predicted price for flat with 2000 sq ft: ₹740,980.25

In []: