

```
In [21]: import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression as LR
from sklearn.model_selection import train_test_split as TTS
from sklearn.metrics import mean_squared_error as MSE, mean_absolute_error as MAE, r2_score as R2
```

```
In [5]: # Load dataset
df=pd.read_csv(r'D:\Intern\SkillCraft Intern\House pred.csv')
```

```
In [6]: df.head()
```

```
Out[6]:
```

	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_above	sqft_basement	yr_built
0	3	1.50	1340	7912	1.5	0	0	3	1340	0	195
1	5	2.50	3650	9050	2.0	0	4	5	3370	280	192
2	3	2.00	1930	11947	1.0	0	0	4	1930	0	196
3	3	2.25	2000	8030	1.0	0	0	4	1000	1000	196
4	4	2.50	1940	10500	1.0	0	0	4	1140	800	197

```
In [7]: df.tail()
```

```
Out[7]:
```

	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	sqft_above	sqft_basement	yr_built
4595	3	1.75	1510	6360	1.0	0	0	4	1510	0	
4596	3	2.50	1460	7573	2.0	0	0	3	1460	0	
4597	3	2.50	3010	7014	2.0	0	0	3	3010	0	
4598	4	2.00	2090	6630	1.0	0	0	3	1070	1020	
4599	3	2.50	1490	8102	2.0	0	0	4	1490	0	

```
In [8]: df.isnull().sum()
```

```
Out[8]: bedrooms      0
bathrooms      0
sqft_living     0
sqft_lot       0
floors         0
waterfront     0
view           0
condition      0
sqft_above     0
sqft_basement  0
yr_built       0
price          0
dtype: int64
```

In [9]: `df.describe()`

Out[9]:

	bedrooms	bathrooms	sqft_living	sqft_lot	floors	waterfront	view	condition	
count	4600.000000	4600.000000	4600.000000	4.600000e+03	4600.000000	4600.000000	4600.000000	4600.000000	4
mean	3.400870	2.160815	2139.346957	1.485252e+04	1.512065	0.007174	0.240652	3.451739	1
std	0.908848	0.783781	963.206916	3.588444e+04	0.538288	0.084404	0.778405	0.677230	
min	0.000000	0.000000	370.000000	6.380000e+02	1.000000	0.000000	0.000000	1.000000	
25%	3.000000	1.750000	1460.000000	5.000750e+03	1.000000	0.000000	0.000000	3.000000	1
50%	3.000000	2.250000	1980.000000	7.683000e+03	1.500000	0.000000	0.000000	3.000000	1
75%	4.000000	2.500000	2620.000000	1.100125e+04	2.000000	0.000000	0.000000	4.000000	2
max	9.000000	8.000000	13540.000000	1.074218e+06	3.500000	1.000000	4.000000	5.000000	9

In [11]: `# Split into Independent and Dependent Values`

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

In [12]: `# Data Split Into Train And Test`

```
x_train,x_test,y_train,y_test=TTS(x,y,test_size=0.3,random_state=203)
```

In [13]: `lm=LR()`

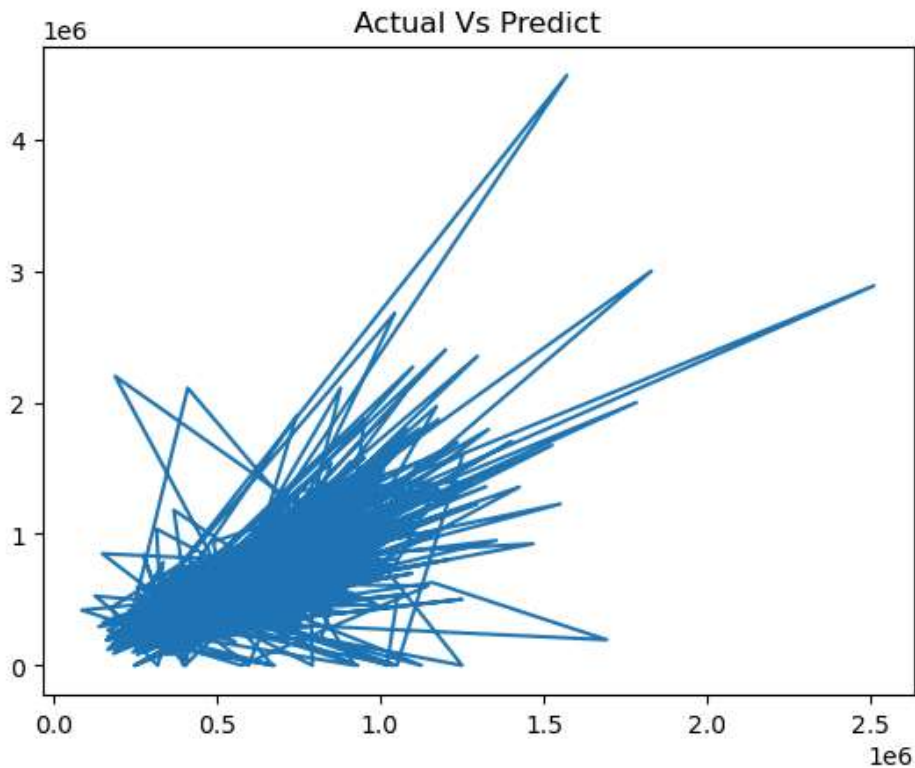
```
lm.fit(x_train,y_train)
```

Out[13]:

```
LinearRegression
LinearRegression()
```

In [14]: `y_pred=lm.predict(x_test)`

```
In [20]: # Visualize
plt.plot(y_pred,y_test)
plt.title('Actual Vs Predict')
plt.show()
```



```
In [22]: # evaluate its Performance Using Metrics
mse=MSE(y_pred,y_test)
mae=MAE(y_pred,y_test)
rs=RS(y_pred,y_test)
print(f"Mean Squared Error = {mse}")
print(f"Mean Absolute Error = {mae}")
print(f"R2 Score = {rs}")
```

Mean Squared Error = 63187196340.68652
Mean Absolute Error = 158397.51460542338
R2 Score = -0.04511278779316674

```
In [23]: #Enter the New Data to Predict
import numpy as np
```

```
In [27]: new_data=np.array([[3,2,3400,7910,2,1,0,3,1300,200,2000]])
print(f"Predicted House Price = {lm.predict(new_data)[0]}")
```

Predicted House Price = 934260.4190417998

C:\Users\nikhil\anaconda3\Lib\site-packages\sklearn\base.py:464: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
warnings.warn(