In [ ]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.linear_model import LinearRegression
```

Load Dataset

In [ ]:
```
data=pd.read_csv("E:\data.csv")
```

In [ ]:
```
data.head()
```

Out[ ]:

	date	price	bedrooms	bathrooms	sqft_living	area	floors	waterfront	view	condition	sqft_above	sqft_basement	yr
0	2014-05-02 00:00:00	313000.0	3.0	1.50	1340	7912	1.5	0	0	3	1340	0	
1	2014-05-02 00:00:00	2384000.0	5.0	2.50	3650	9050	2.0	0	4	5	3370	280	
2	2014-05-02 00:00:00	342000.0	3.0	2.00	1930	11947	1.0	0	0	4	1930	0	
3	2014-05-02 00:00:00	420000.0	3.0	2.25	2000	8030	1.0	0	0	4	1000	1000	
4	2014-05-02 00:00:00	550000.0	4.0	2.50	1940	10500	1.0	0	0	4	1140	800	

In [ ]:
```
data.tail()
```

Out[ ]:

	date	price	bedrooms	bathrooms	sqft_living	area	floors	waterfront	view	condition	sqft_above	sqft_basem
4595	2014-07-09 00:00:00	308166.666667	3.0	1.75	1510	6360	1.0	0	0	4	1510	
4596	2014-07-09 00:00:00	534333.333333	3.0	2.50	1460	7573	2.0	0	0	3	1460	
4597	2014-07-09 00:00:00	416904.166667	3.0	2.50	3010	7014	2.0	0	0	3	3010	
4598	2014-07-10 00:00:00	203400.000000	4.0	2.00	2090	6630	1.0	0	0	3	1070	1
4599	2014-07-10 00:00:00	220600.000000	3.0	2.50	1490	8102	2.0	0	0	4	1490	

In [ ]:
```
data.describe()
```

Out[ ]:

	price	bedrooms	bathrooms	sqft_living	area	floors	waterfront	view	condition	sq
count	4.600000e+03	4600.000000	4600.000000	4600.000000	4.600000e+03	4600.000000	4600.000000	4600.000000	4600.000000	460
mean	5.519630e+05	3.400870	2.160815	2139.346957	1.485252e+04	1.512065	0.007174	0.240652	3.451739	182
std	5.638347e+05	0.908848	0.783781	963.206916	3.588444e+04	0.538288	0.084404	0.778405	0.677230	86
min	0.000000e+00	0.000000	0.000000	370.000000	6.380000e+02	1.000000	0.000000	0.000000	1.000000	37
25%	3.228750e+05	3.000000	1.750000	1460.000000	5.000750e+03	1.000000	0.000000	0.000000	3.000000	119
50%	4.609435e+05	3.000000	2.250000	1980.000000	7.683000e+03	1.500000	0.000000	0.000000	3.000000	159
75%	6.549625e+05	4.000000	2.500000	2620.000000	1.100125e+04	2.000000	0.000000	0.000000	4.000000	230
max	2.659000e+07	9.000000	8.000000	13540.000000	1.074218e+06	3.500000	1.000000	4.000000	5.000000	941

In [ ]:
```
data.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 4600 entries, 0 to 4599
Data columns (total 18 columns):
 #   Column         Non-Null Count  Dtype  
---  ------         --------------  -----  
 0   date           4600 non-null   object 
 1   price          4600 non-null   float64
 2   bedrooms       4600 non-null   float64
 3   bathrooms      4600 non-null   float64
 4   sqft_living    4600 non-null   int64  
 5   area           4600 non-null   int64  
 6   floors         4600 non-null   float64
 7   waterfront     4600 non-null   int64  
 8   view           4600 non-null   int64  
 9   condition      4600 non-null   int64  
 10  sqft_above     4600 non-null   int64  
 11  sqft_basement  4600 non-null   int64  
 12  yr_built       4600 non-null   int64  
 13  yr_renovated   4600 non-null   int64  
 14  street         4600 non-null   object 
 15  city           4600 non-null   object 
 16  statezip       4600 non-null   object 
 17  country        4600 non-null   object 
dtypes: float64(4), int64(9), object(5)
memory usage: 647.0+ KB
```

In [ ]:
```
data = data.filter(['area', 'price'])
```

Load Summarize

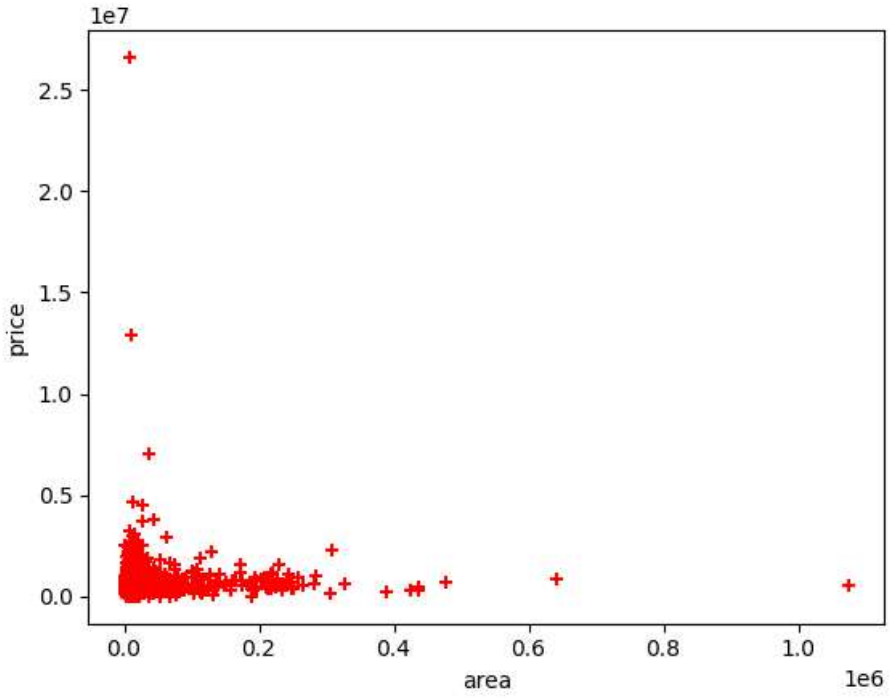
In [ ]:
```
print(data.shape)
print(data.head(5))
```

```
(4600, 2)
    area      price
0   7912   313000.0
1   9050  2384000.0
2  11947   342000.0
3   8030   420000.0
4  10500   550000.0
```

Visualize Data

In [ ]:
```
plt.xlabel("area")
plt.ylabel("price")
plt.scatter(data.area,data.price,color='red',marker='+')
```

Out[ ]:
```
<matplotlib.collections.PathCollection at 0x14770998f50>
```



Segregate Dataset into Input X and Output Y

In [ ]:
```
X=data.drop('price',axis='columns')
X
```

Out[ ]:

	area
0	7912
1	9050
2	11947
3	8030
4	10500
...	...
4595	6360
4596	7573
4597	7014
4598	6630
4599	8102

4600 rows × 1 columns

In [ ]:
```
Y=data.price
Y
```

Out[ ]:
```
0       3.130000e+05
1       2.384000e+06
2       3.420000e+05
3       4.200000e+05
4       5.500000e+05
            ...
4595    3.081667e+05
4596    5.343333e+05
4597    4.169042e+05
4598    2.034000e+05
4599    2.206000e+05
Name: price, Length: 4600, dtype: float64
```

Training Datset using Linear Regression

In [ ]:
```
model = LinearRegression()
model.fit(X,Y)
```

Out[ ]:
```
▾ LinearRegression
LinearRegression()
```

Predicted Price for Land sq.ft of Custom Values

In [ ]:
```
x=20000
LandAreainSqFt=[[x]]
PredictedPrice=model.predict(LandAreainSqFt)
print(f"Predicted Price of a Land of {x} SqFt is: ",PredictedPrice)
```

```
Predicted Price of a Land of 20000 SqFt is:  [556043.48480864]
C:\Users\pr12-\AppData\Local\Packages\PythonSoftwareFoundation.Python.3.11_qbz5n2kfra8p0\LocalCache\local-packages\Python311\site-packages\sklearn\base.py:464: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

Checking if the Model is Right

In [ ]:
```
m=model.coef_
print("Slope of the line is: ",m)
```

```
Slope of the line is:  [0.79271668]
```

In [ ]:
```
b=model.intercept_
print("Intercept of the line is: ",b)
```

```
Intercept of the line is:  540189.1512959745
```

In [ ]:
```
y=m*x+b
print(f"Price of a Land of {x} SqFt is: ",y)
```

```
Price of a Land of 20000 SqFt is:  [556043.48480864]
```