Linear regression

- General equation of any straight line
- y=m (x) + c
- y - y_axis
- m - gradient or slope
- x - x_axis
- c - intercept

Steps

- 1) Import libraries
- 2) Import the train data file and clean the data
- 3) Separate features data
- 4) Separate Labels data
- 5) Create a linear Regression model
- 6) Fit the data into the model
- 7) Prediction result

In [1]:

```
# 1) Import libraries
import pandas as pd
import numpy as np
from sklearn import linear_model
import matplotlib.pyplot as plt
```

In [2]:

```
# 2) Import train data file
train_data = pd.read_csv('homeprices.csv')
```

In [3]:

```
train_data
```

Out[3]:

	area	price
0	2600	550000
1	3000	565000
2	3200	610000
3	3600	680000
4	4000	725000

In [4]:

```
# 3) Separate features data
Feature = train_data.drop('price',axis='columns')
```

In [5]:

```
Feature
```

Out[5]:

```
   area
0  2600
1  3000
2  3200
3  3600
4  4000
```

In [6]:

```
# 4) Separate Labels data
Label = train_data[['price']]
```

In [7]:

```
Label
```

Out[7]:

```
    price
0  550000
1  565000
2  610000
3  680000
4  725000
```

In [8]:

```
# 5) Creat linear regretion model
lin_reg = linear_model.LinearRegression()
```

In [9]:

```
# 6) Fit the data into model
lin_reg.fit(Feature,Label)
```

Out[9]:

```
LinearRegression()
```

In [10]:

```
# 7) Prediction result (enter the area)
area_input=input('Enter the square feets:')
lin_reg.predict([[area_input]])
# -----------------------------------------------------------------------------------------
```

```
Enter the square feets:7000

C:\Users\Gopi\anaconda3\lib\site-packages\sklearn\base.py:450: UserWarnin
g: X does not have valid feature names, but LinearRegression was fitted wi
th feature names
  warnings.warn(
C:\Users\Gopi\anaconda3\lib\site-packages\sklearn\base.py:566: FutureWarni
ng: Arrays of bytes/strings is being converted to decimal numbers if dtype
='numeric'. This behavior is deprecated in 0.24 and will be removed in 1.1
(renaming of 0.26). Please convert your data to numeric values explicitly
instead.
  X = check_array(X, **check_params)
```

Out[10]:

```
array([[1131130.1369863]])
```

In [ ]:

E D A

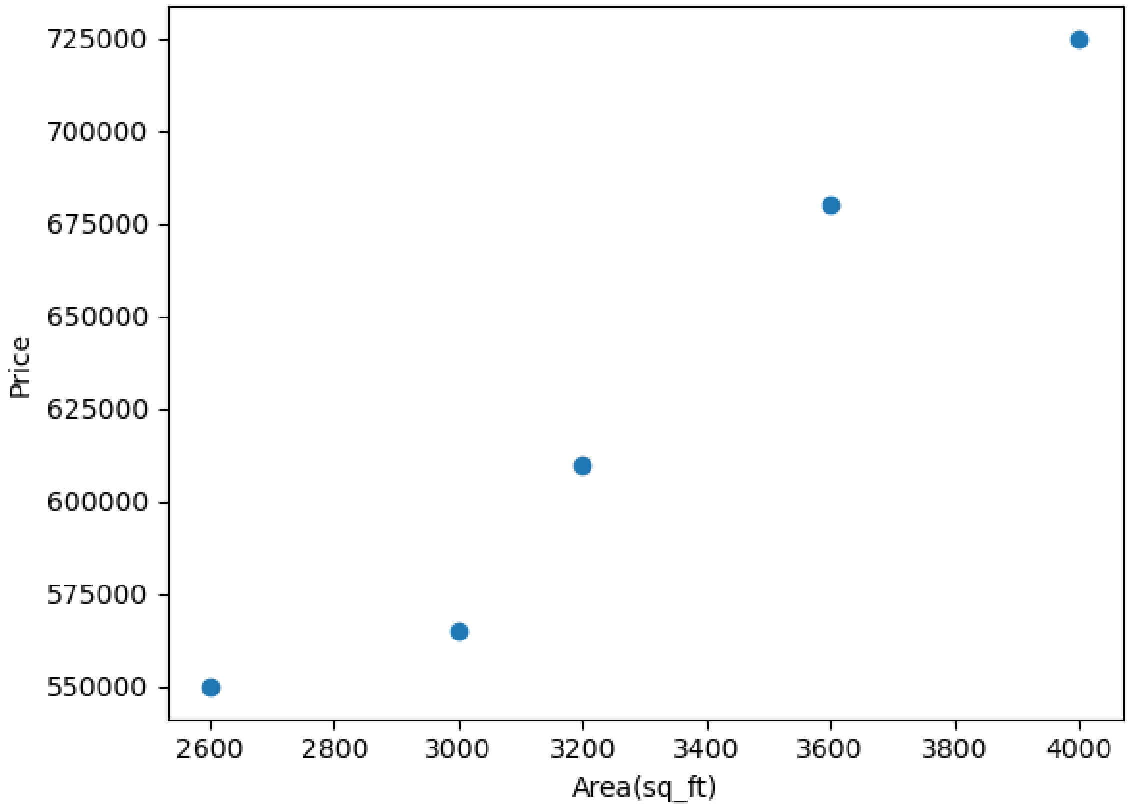
- EDA just for view the data

In [11]:

```
%matplotlib inline
plt.xlabel('Area(sq_ft)')
plt.ylabel('Price')
plt.scatter(train_data['area'],train_data['price'])
```

Out[11]:

```
<matplotlib.collections.PathCollection at 0x271fdfecfd0>
```

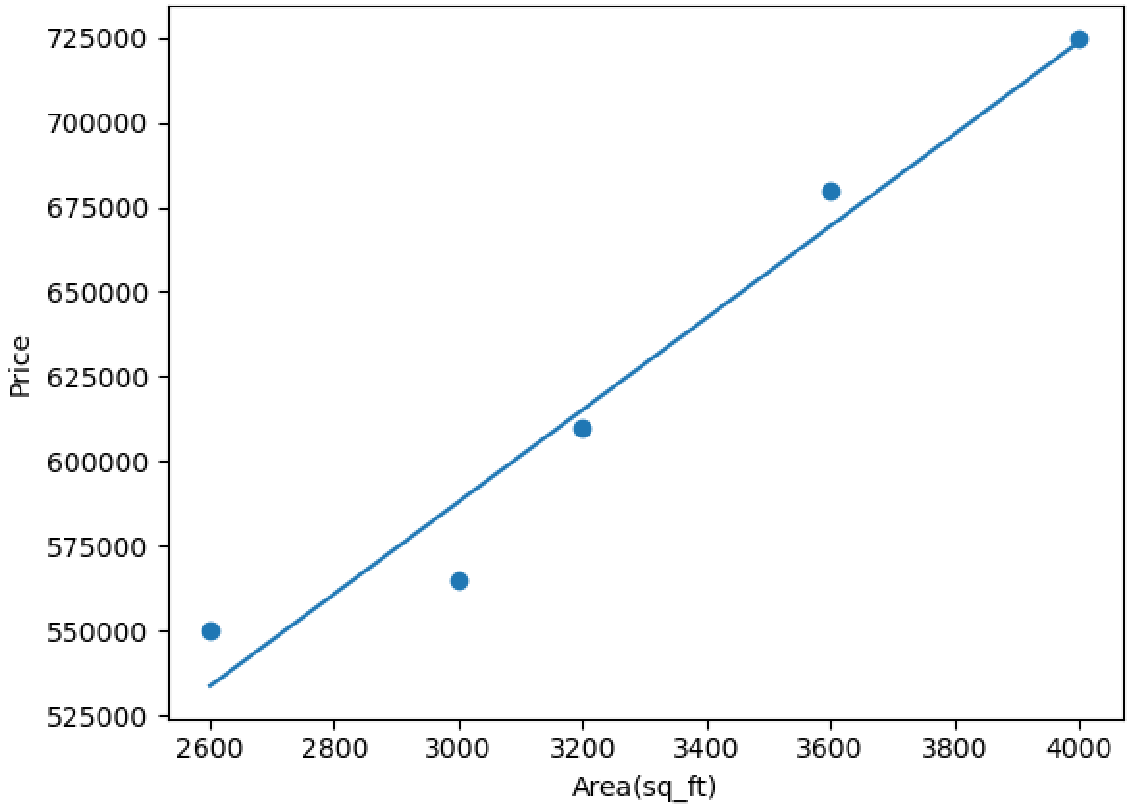


In [12]:

```
# Linear regresession model result
%matplotlib inline
plt.xlabel('Area(sq_ft)')
plt.ylabel('Price')
plt.scatter(train_data.area,train_data.price)
plt.plot(train_data['area'],lin_reg.predict(train_data[['area']])) # Result
```

Out[12]:

```
[<matplotlib.lines.Line2D at 0x271fe056be0>]
```



Test the model for the formula

In [13]:

```
# Take 'm' value
lin_reg.coef_
```

Out[13]:

```
array([[135.78767123]])
```

In [14]:

```
# Take 'c' value
lin_reg.intercept_
```

Out[14]:

```
array([180616.43835616])
```

In [15]:

```
# y=m*x+c
y=135.78767123*float(area_input)+180616.43835616432
y
```

Out[15]:

```
1131130.1369661642
```

In [ ]:

```

```

TEST data

In [16]:

```
# Import areas
test_data = pd.read_csv('areas.csv')
```

In [17]:

```
test_data
```

Out[17]:

	area
0	1000
1	1500
2	2300
3	3540
4	4120
5	4560
6	5490
7	3460
8	4750
9	2300
10	9000
11	8600
12	7100

In [18]:

```
# Prediction result
result=lin_reg.predict(test_data)
```

In [19]:

```
result
```

Out[19]:

```
array([ 316404.10958904,  384297.94520548,  492928.08219178,
        661304.79452055,  740061.64383562,  799808.21917808,
        926090.75342466,  650441.78082192,  825607.87671233,
        492928.08219178, 1402705.47945205, 1348390.4109589 ,
       1144708.90410959])
```

In [20]:

```
# Creat new columns
test_data['price']=result
```

In [21]:

```
test_data
```

Out[21]:

	area	price
0	1000	3.164041e+05
1	1500	3.842979e+05
2	2300	4.929281e+05
3	3540	6.613048e+05
4	4120	7.400616e+05
5	4560	7.998082e+05
6	5490	9.260908e+05
7	3460	6.504418e+05
8	4750	8.256079e+05
9	2300	4.929281e+05
10	9000	1.402705e+06
11	8600	1.348390e+06
12	7100	1.144709e+06

In [ ]:

```
```