Experiment No: 01

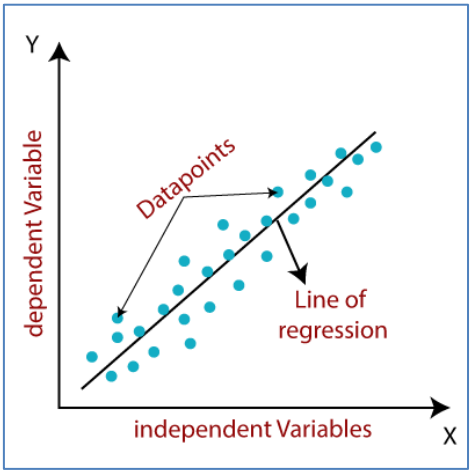
● **Aim: To implement Supervised Learning using Linear regression algorithm.**

● **Theory:**

Linear regression

Linear regression algorithm shows a linear relationship between a dependent (y) and one or more independent (y) variables, hence called as linear regression. Since linear regression shows the linear relationship, which means it finds how the value of the dependent variable is changing according to the value of the independent variable.

The linear regression model provides a sloped straight line representing the relationship between the variables. Consider the below image:



Mathematically, we can represent a linear regression as:

$$y= a0+a1x+ \varepsilon$$

Here,

Y= Dependent Variable (Target Variable)

X= Independent Variable (predictor Variable)

a0= intercept of the line (Gives an additional degree of freedom)

a1 = Linear regression coefficient (scale factor to each input value).

Types of Linear Regression

Simple Linear Regression:

If a single independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Simple Linear Regression.

Multiple Linear regression:

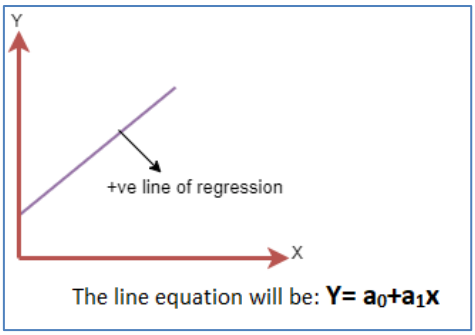
If more than one independent variable is used to predict the value of a numerical dependent variable, then such a Linear Regression algorithm is called Multiple Linear Regression.

Linear Regression Line:

A linear line showing the relationship between the dependent and independent variables is called a regression line. A regression line can show two types of relationship.

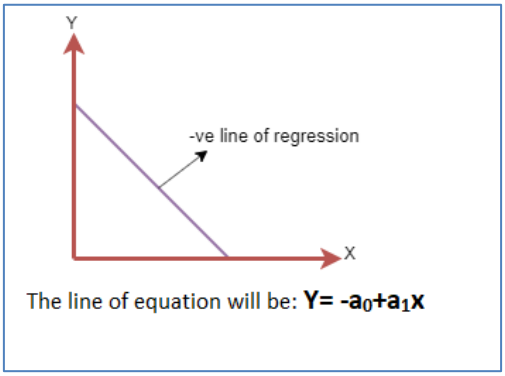
Positive Linear Relationship:

If the dependent variable increases on the Y-axis and independent variable increases on X-axis, then such a relationship is termed as a Positive linear relationship.



Negative Linear Relationship:

If the dependent variable decreases on the Y-axis and independent variable increases on the X-axis, then such a relationship is called a negative linear relationship.



Simple Linear Regression Model:

The Simple Linear Regression model can be represented using the below equation:

$$y= a0+a1x+ \varepsilon$$

Where,

a0= It is the intercept of the Regression line (can be obtained putting x=0)

a1= It is the slope of the regression line, which tells whether the line is increasing or decreasing.

ε = The error term. (For a good model it will be negligible)

Implementation:

jupyter Linear Regression single value Last Checkpoint: 09/09/2023 (autosaved)

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

Run Code

In [15]:
```
import pandas as pd
import numpy as np
import matplotlib.pylab as plt
from sklearn import linear_model
```

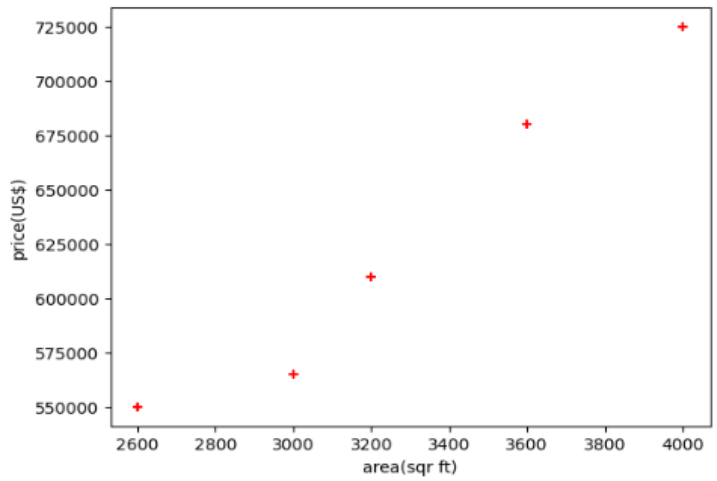
In [13]:
```
df = pd.read_csv("homeprices.csv")
df
```

Out[13]:

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

In [16]:
```
%matplotlib inline
plt.xlabel('area(sqr ft)')
plt.ylabel('price(US$)')
plt.scatter(df.area,df.price ,color ='red' ,marker='+')
```

Out[16]: <matplotlib.collections.PathCollection at 0x2ca35ae9210>



In [17]:
```
reg =linear_model.LinearRegression()
reg.fit(df[['area']],df.price)
```

In [19]:
```
reg.predict([[3300]])
```

```
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py:439: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```

Out[19]: array([628715.75342466])

In [20]:
```
reg.coef_
```

Out[20]: array([135.78767123])

In [21]:
```
reg.intercept_
```

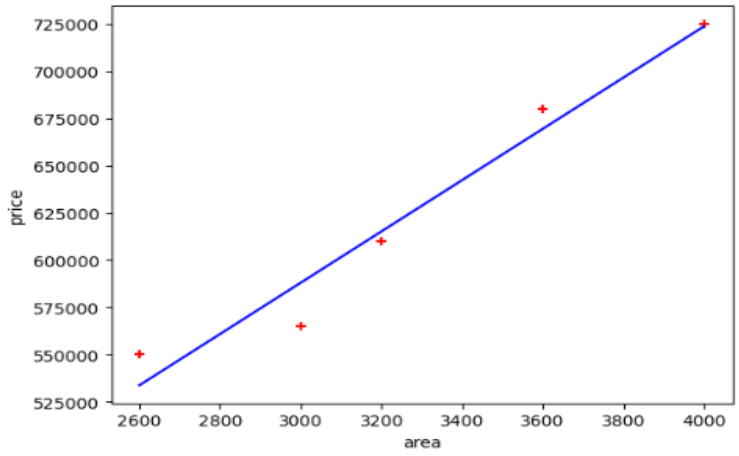
Out[21]: 180616.43835616432

Y = m * X + b (m is coefficient and b is intercept)

In [22]:
```
135.78767123*3300+180616.43835616432
```

Out[22]: 628715.7534151643

Out[46]: [<matplotlib.lines.Line2D at 0x2ca3cfb02d0>]



new_df = df.drop('price',axis='columns')

new_df

```
In [10]: price = df.price
         price
```

```
Out[10]: 0    550000
         1    565000
         2    610000
         3    680000
         4    725000
         Name: price, dtype: int64
```

```
In [12]: #Create linear regression object
         reg = linear_model.LinearRegression()
         reg.fit(new_df,price)
```

```
Out[12]: LinearRegression()
```

In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

(1) Predict price of a home with area = 3300 sqr ft

```
In [27]: reg.predict([[5000]])
```

```
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py:439: UserWarning: X does not have valid feature names, but LinearReg
ression was fitted with feature names
  warnings.warn(
```

```
Out[27]: array([859554.79452055])
```

```
In [28]: reg.coef_
```

```
Out[28]: array([135.78767123])
```

```
In [29]: reg.intercept_
```

```
Out[29]: 180616.43835616432
```

Y = m * X + b (m is coefficient and b is intercept)

```
In [26]: 5000*135.78767123 + 180616.43835616432
```

```
Out[26]: 628715.7534151643
```

(1) Predict price of a home with area = 5000 sqr ft

```
In [18]: reg.predict([[5000]])
```

```
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py:439: UserWarning: X does not have valid feature names, but LinearReg
ression was fitted with feature names
  warnings.warn(
```

```
Out[18]: array([859554.79452055])
```

Generate CSV file with list of home price predictions

```
In [20]: p = reg.predict(area_df)
         p
```

```
Out[20]: 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])
```

Multiple Linear Regression:

MLR equation:

In Multiple Linear Regression, the target variable(Y) is a linear combination of multiple predictor variables x1, x2, x3, ...,xn. Since it is an enhancement of Simple Linear Regression, so the same is applied for the multiple linear regression equation, the equation becomes:

Y= b₀+b₁x₁+ b₂x₂+ b<sub>3

Where,

Y= Output/Response variable

b0, b1, b2, b3 , bn....= Coefficients of the model.

x1, x2, x3, x4,...= Various Independent/feature variable .

Implementation:

jupyter Linear Regression multiple value Last Checkpoint: 09/17/2023 (autosaved)

Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

Run Code

In [3]:
```
import pandas as pd
import numpy as np
from sklearn import linear_model
```

In [4]:
```
df = pd.read_csv('homeprices.csv')
df
```

Out[4]:

	area	bedrooms	age	price
0	2600	3.0	20	550000
1	3000	4.0	15	565000
2	3200	NaN	18	610000
3	3600	3.0	30	595000
4	4000	5.0	8	760000
5	4100	6.0	8	810000

Data Preprocessing: Fill NA values with median value of a column

In [5]:
```
import math
median_bedrooms = math.floor(df.bedrooms.median())
median_bedrooms
```

Out[5]: 4

In [6]:
```
df.bedrooms = df.bedrooms.fillna(median_bedrooms)
df
```

Out[6]:

	area	bedrooms	age	price
0	2600	3.0	20	550000
1	3000	4.0	15	565000
2	3200	4.0	18	610000
3	3600	3.0	30	595000
4	4000	5.0	8	760000
5	4100	6.0	8	810000

In [7]:
```
reg = linear_model.LinearRegression()
reg.fit(df[['area','bedrooms','age']],df.price)
```

Out[7]:
```
LinearRegression()
```
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [8]:
```
reg.coef_
```
Out[8]:
```
array([  112.06244194, 23388.88007794, -3231.71790863])
```

In [9]:
```
reg.intercept_
```
Out[9]:
```
221323.00186540408
```

Find price of home with 3000 sqr ft area, 3 bedrooms, 40 year old

In [12]:
```
reg.predict([[3000, 3, 40]])
```
```
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py:439: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```
Out[12]:
```
array([498408.25158031])
```

In [14]:
```
112.06244194*3000+23388.88007794*3+-3231.71790863*40+221323.00186540408
```
Out[14]:
```
498408.2515740241
```

Find price of home with 2500 sqr ft area, 4 bedrooms, 5 year old

In [15]:
```
reg.predict([[2500, 4, 5]])
```
```
C:\ProgramData\anaconda3\Lib\site-packages\sklearn\base.py:439: UserWarning: X does not have valid feature names, but LinearRegression was fitted with feature names
  warnings.warn(
```
Out[15]:
```
array([578876.03748933])
```

In [ ]:

● Conclusion:

Linear regression is a versatile tool used in various fields, including finance, economics, and science, for tasks such as predicting stock prices, housing values, and experimental outcomes. It's a valuable technique for understanding relationships within data and making accurate predictions.