Theory :

In statistics, linear regression is a linear approach to modeling the relationship between a scalar response (or dependent variable) and one or more explanatory variables (or independent variables). The case of one explanatory variable is called simple linear regression. For more than one explanatory variable, the process is called multiple linear regression.[1] This term is distinct from multivariate linear regression, where multiple correlated dependent variables are predicted, rather than a single scalar variable.

Import all the necessary machine learning libraries

In [ ]:

```
import pandas as pd
import numpy as np
import scipy as sp
import matplotlib.pyplot as plt
import sklearn
from sklearn import linear_model
```

Import and read the csv dataset using panda : "pd.read_csv()"

In [2]:

```
df=pd.read_csv("~/Desktop/homeprices.csv")
```

In [3]:

```
df
```

Out[3]:

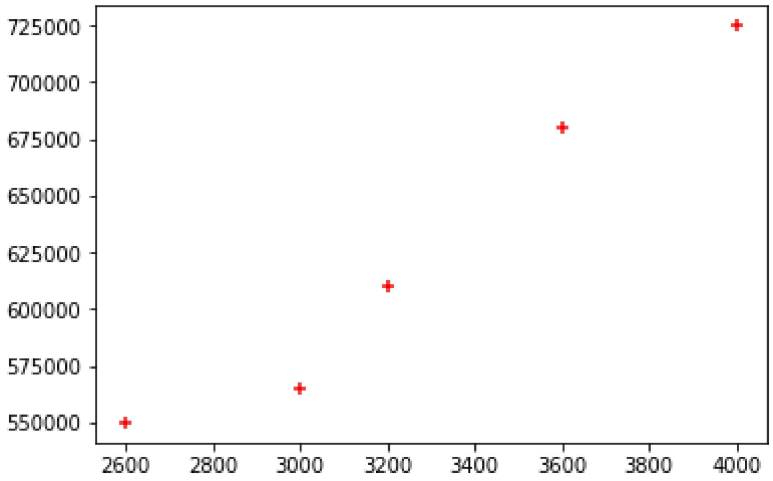
	area	price
0	2600	550000
1	3000	565000
2	3200	610000
3	3600	680000
4	4000	725000

In [4]:

```
plt.scatter(df.area,df.price,color='red',marker = '+')
```

Out[4]:

```
<matplotlib.collections.PathCollection at 0x1fc66b2b0b8>
```



Build the linear regression model

In [5]:

```
model = sklearn.linear_model.LinearRegression()
```

In [6]:

```
model
```

Out[6]:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=F
alse)
```

Fit the data to the model built

In [7]:

```
model.fit(df[['area']],df.price)
```

Out[7]:

```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=F
alse)
```

In [8]:

```
df['area']
```

Out[8]:

```
0    2600
1    3000
2    3200
3    3600
4    4000
Name: area, dtype: int64
```

In [9]:

```
type(df[['area']])
```

Out[9]:

```
pandas.core.frame.DataFrame
```

In [10]:

```
type(df['area'])
```

Out[10]:

```
pandas.core.series.Series
```

In [11]:

```
df.area
```

Out[11]:

```
0    2600
1    3000
2    3200
3    3600
4    4000
Name: area, dtype: int64
```

In [12]:

```
type(df.area)
```

Out[12]:

```
pandas.core.series.Series
```

In [13]:

```
df
```

Out[13]:

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

In [22]:

```
d = pd.read_csv('~/Desktop/areas.csv')
```

In [ ]:

```
Predict the result using predict function on the model
```

In [15]:

```
p = model.predict(d)
```

In [16]:

```
p
```

Out[16]:

```
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 [17]:

```
d['price']=p
```

In [18]:

```
d
```

Out[18]:

	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

Write the predicted values to a new csv format file

In [19]:

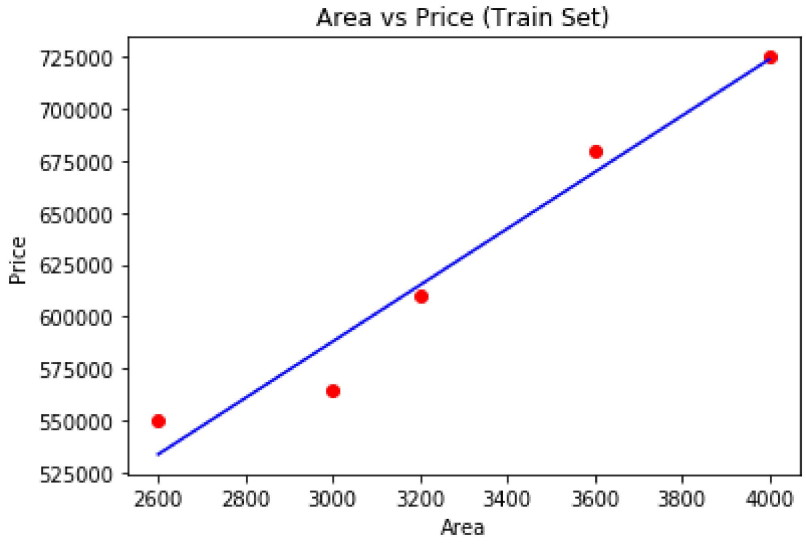
```
d.to_csv("newpred.csv")
```

In [20]:

```
X= df.iloc[:,:-1].values
Y=df.iloc[:,1].values
```

In [21]:

```
plt.scatter(X,Y,color='red')
plt.plot(X,model.predict(X),color='blue')
plt.title('Area vs Price (Train Set)')
plt.xlabel('Area')
plt.ylabel('Price')
plt.show()
```

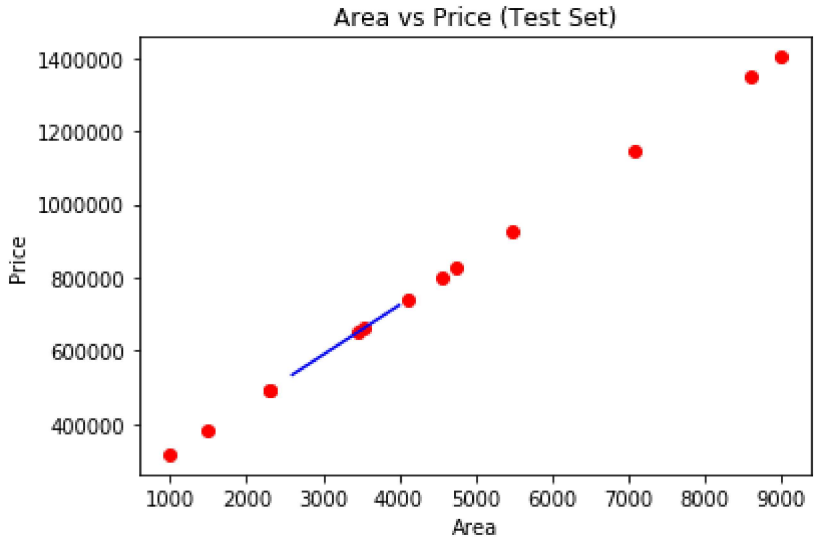


In [23]:

```
regressor = sklearn.linear_model.LinearRegression()
regressor.fit(X,Y)
pred=regressor.predict(d)
```

In [24]:

```
plt.scatter(d,pred,color='red')
plt.plot(X,model.predict(X),color='blue')
plt.title('Area vs Price (Test Set)')
plt.xlabel('Area')
plt.ylabel('Price')
plt.show()
```



Conclusion : In this experiment we learnt how to build a linear regression model using machine learning libraries such as sklearn. Also we used the fit and predict functions to fit our data in the model and to predict the output results of our model. Also we used the matplotlib functions to plot the linear regression graph.