WELCOME TO SATYAPRAKASH'S REGRESSION MODEL

FOLLOW ME ON LINKEDIN TO GET MORE MODEL

```
from google.colab import drive
drive.mount('/content/drive')
```

```
Drive already mounted at /content/drive; to attempt to forcibly remount, call drive.m
```

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

```
df=pd.read_csv("/content/drive/MyDrive/home price.csv")
```

```
df.head()
```

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

```
%matplotlib inline
plt.xlabel('area(sqr ft)')
plt.ylabel('price(USD$)')
plt.scatter(df.area,df.Price,color='blue',marker='*')
```

```
<matplotlib.collections.PathCollection at 0x7fd5b91bdfd0>
```


```
reg =linear_model.LinearRegression()
reg.fit(df[['area']],df.Price)
```

```
LinearRegression()
```



```
reg.predict([[3300]])
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning: X does not h
  "X does not have valid feature names, but"
array([628715.75342466])
```

```
reg.predict([[2600]])
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning: X does not h
  "X does not have valid feature names, but"
array([533664.38356164])
```

```
reg.predict([[3000]])
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning: X does not h
  "X does not have valid feature names, but"
array([587979.45205479])
```

```
reg.predict([[3200]])
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning: X does not h
  "X does not have valid feature names, but"
array([615136.98630137])
```

```
reg.predict([[3600]])
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning: X does not h
  "X does not have valid feature names, but"
array([669452.05479452])
```

```
reg.predict([[4000]])
```

```
/usr/local/lib/python3.7/dist-packages/sklearn/base.py:451: UserWarning: X does not h
  "X does not have valid feature names, but"
array([723767.12328767])
```

```
reg.coef_
```

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

```
reg.intercept_
```

```
180616.43835616432
```

```
#y=m*x+c
135.78767123*3300+180616.43835616432
```

```
628715.7534151643
```

```
d=pd.read_csv("/content/drive/MyDrive/area.csv")
```

```
d
```

	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

```
reg.predict(d)
```

```
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])
```

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

```
d['Prices'] = p
```

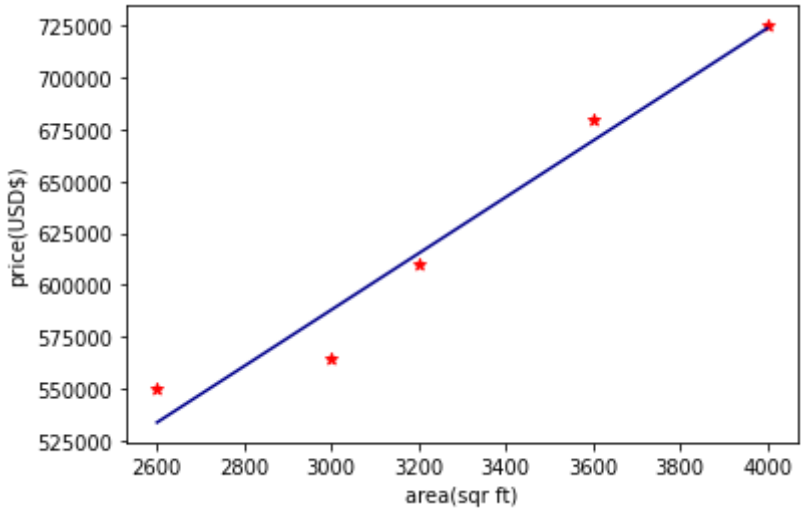
d

	area	Prices
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

```
d.to_csv("prediction.csv",index=False)
```

```
%matplotlib inline
plt.xlabel('area(sqr ft)')
plt.ylabel('price(USD$)')
plt.scatter(df.area,df.Price,color='red',marker='*')
plt.plot(df.area,reg.predict(df[['area']]),color="darkblue")
```

```
[<matplotlib.lines.Line2D at 0x7fd5b8c26e50>]
```




```
from sklearn.metrics import r2_score
```

```
y_actual=[550000,565000,610000,680000,725000]
y_predict=[533664.38356164,587979.45205479,615136.98630137,669452.05479452,723767.12328767
r2 = r2_score(y_actual,y_predict)
r2
```

```
0.9584301138199518
```

```
#Higher r2 value indicates better result
#in worse case you will get negative value.
#if r2 value is 0 the model will give same result always
```