R Squared & Adjusted R Squared

Evaluating Regression Model

Import Libraries

In [1]:

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

Load Dataset from Local Directory

In [2]:

```
from google.colab import files
uploaded = files.upload()
```

Choose File **No file selected**

Upload widget is only available when the cell has been executed in the current browser session. Please rerun this cell to enable.

```
Saving dataset1.csv to dataset1.csv
```

Load Dataset

In [4]:

```
dataset = pd.read_csv('dataset1.csv')
```

Load Summarize

In [5]:

```
print(dataset.shape)
print(dataset.head(5))
```

```
(10, 2)
   area  price
0  1000   2245
1  2000   4575
2  3000   6874
3  4000   8878
4  5000  10589
```

Visualize Dataset

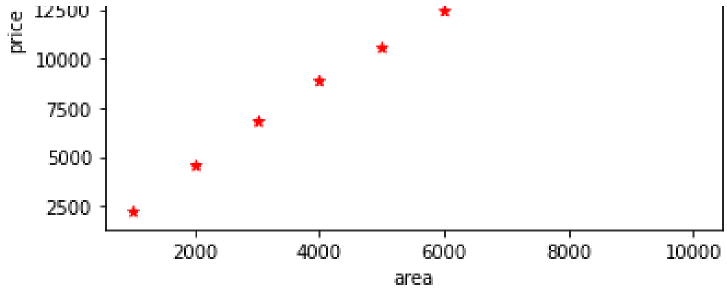
In [6]:

```
plt.xlabel('area')
plt.ylabel('price')
plt.scatter(dataset.area, dataset.price, color = 'red', marker = '*')
```

Out[6]:

```
<matplotlib.collections.PathCollection at 0x7f827346f910>
```

20000
17500
15000



Segregate Dataset into Input X & Output Y

In [7]:

```
X = dataset.drop('price', axis = 'columns')
X
```

Out[7]:

	area
0	1000
1	2000
2	3000
3	4000
4	5000
5	6000
6	7000
7	8000
8	9000
9	10000

In [8]:

```
Y = dataset.price
Y
```

Out[8]:

```
0     2245
1     4575
2     6874
3     8878
4    10589
5    12457
6    14785
7    16785
8    18958
9    20789
Name: price, dtype: int64
```

Splitting Dataset for Testing our Model

In [9]:

```
from sklearn.model_selection import train_test_split
x_train,x_test, y_train, y_test = train_test_split(X, Y, test_size = 0.20, random_state
= 0)
```

Train Dataset using Linear Regression

In [10]:

```
model = LinearRegression()
model = LinearRegression()
model.fit(x_train, y_train)
```

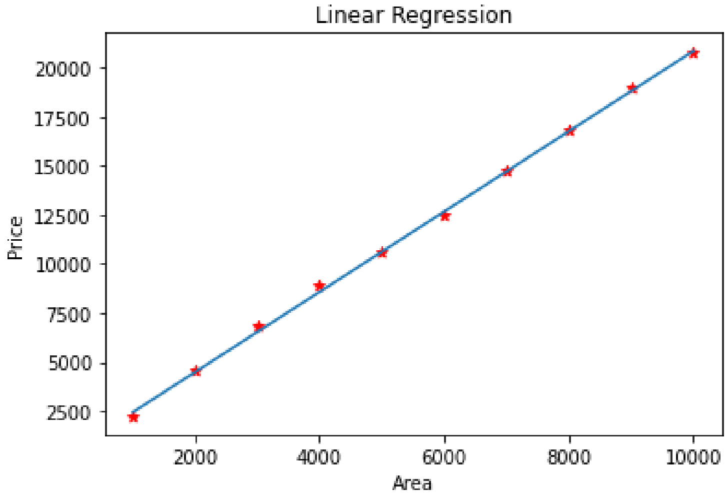
Out[10]:

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

Visualizing Linear Regression results

In [11]:

```
plt.scatter(X, Y, color = 'red', marker = '*')
plt.plot(X, model.predict(X))
plt.title("Linear Regression")
plt.xlabel("Area")
plt.ylabel("Price")
plt.show()
```



R Squared = 1 - (SSR/SST)

Where,

SSR = Sum of Squared Residuals

SST = Sum of Squared Total Adjusted R Squared = 1-[(1 - RSquared)*((n-1)/(n-p-1))]

R - Squared Score

In [12]:

```
rsquared = model.score(x_test, y_test)
print(rsquared)
```

```
0.9980555305079885
```

Adjusted R Squared of the Model

In [13]:

```
n = len(dataset)   #Length of Total Dataset
p = len(dataset.columns)-1   #Length of Features
adjr = 1-(1-rsquared)*(n-1)/(n-p-1)
print(adjr)
```

```
0.997812471821487
```

Prediction

In [14]:

```
x = 6500
LandAreainSqFt = [[x]]
```

```
PredictionmodelResult = model.predict(LandAreainSqFt)
print(PredictionmodelResult)
```

```
[13687.72504892]
```