

Microsoft Excel ribbon showing the following tabs: File, Home, Insert, Formulas, Data, Review, View, and Help. The ribbon is currently set to the 'Home' tab, which includes sections for Clipboard, Font, Paragraph, Styles, Cells, and Editing.

The spreadsheet displays a dataset with two columns: 'area' and 'price'. The data is organized into rows, with the first row (row 1) serving as the header. The 'area' column contains values ranging from 1000 to 10000, and the 'price' column contains corresponding values ranging from 2245 to 20789.

area	price
1000	2245
2000	4575
3000	6874
4000	8878
5000	10589
6000	12457
7000	14785
8000	16785
9000	18958
10000	20789

The spreadsheet is titled 'dataset1' in the bottom-left corner. A small video feed of a person is visible in the bottom-right corner of the screen.

## Import Libraries

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

## Load Dataset from Local Directory

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



```
[3] dataset = pd.read_csv('dataset1.csv')
```

Load Summarize

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

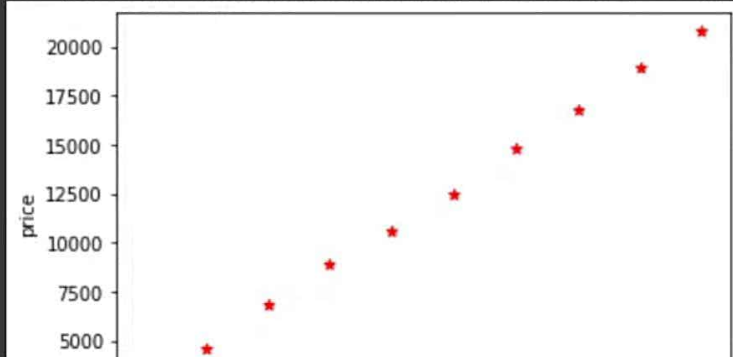
```
(10, 2)
  area  price
0  1000   2245
1  2000   4575
2  3000   6874
3  4000   8878
```

0s completed at 19:06



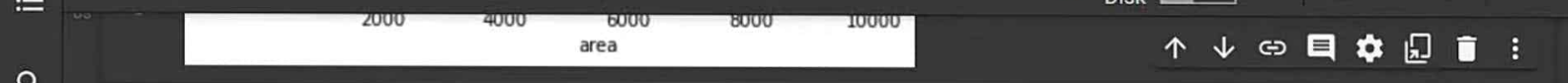
```
plt.xlabel('area')
plt.ylabel('price')
plt.scatter(dataset.area, dataset.price, color='red', marker='*')
```

<matplotlib.collections.PathCollection at 0x7fa060a22e50>



✓ 0s completed at 19:07





## Segregate Dataset into Input X & Output Y

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

```
[ ] Y = dataset.price
Y
```

## Splitting Dataset for Testing our Model

✓ 0s completed at 19:07



colab.research.google.com/drive/1ry0raEyThl-qIH51v5V4yfoykfb-.8A-#scrollTo=nKmEySI1poV\_

colab.research.google.com wants to  
Show notifications  
Allow Block

SessionModelUsingRSquaredAdjust... ☆ Comment Share Settings Profile

Runtime Tools Help

+ Code + Text RAM Disk Editing

```
[8] 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)
```

▼ Training Dataset using Linear Regression


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

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

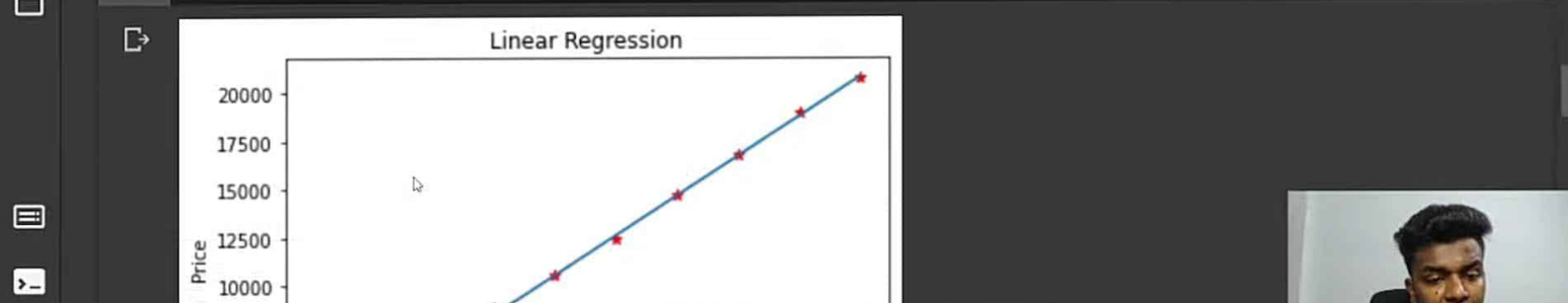
▼ Visualizing Linear Regression results

0s completed at 19:08

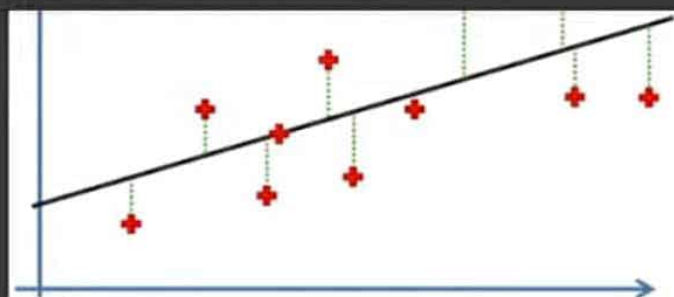
Type here to search 28°C Mostly cloudy



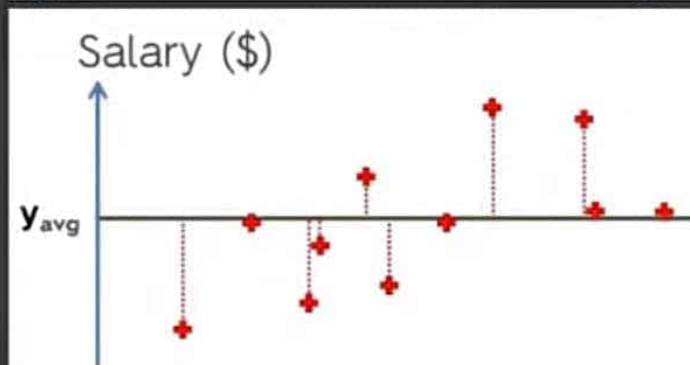
```
plt.scatter(X,Y, color="red",marker='*')  
plt.plot(X, model.predict(X))  
plt.title("Linear Regression")  
plt.xlabel("Area")  
plt.ylabel("Price")  
plt.show()
```



+ Code + Text



$$\text{SUM } (y_i - \hat{y}_i)^2 \rightarrow \min$$



$$SS_{\text{tot}} = \text{SUM } (y_i - y_{\text{avg}})^2$$

R Squared =  $1 - (\text{SSR} / \text{SST})$

where, SSR = Sum of Squared Residuals

SST = Sum of Squared Total

Adjusted R Squared =  $1 - [(1 - \text{R Squared}) * ((n-1) / (n-p-1))]$



R Squared =  $1 - (SSR / SST)$

where, SSR = Sum of Squared Residuals

SST = Sum of Squared Total

Adjusted R Squared =  $1 - [(1 - R \text{ Squared}) * ((n-1) / (n-p-1))]$

### R-Squared Score

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

0.9980555305079885

### Adjusted R Squared of the Model

0s completed at 19:15



```
[12] 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

```
x=6500
LandAreainSqFt=[[x]]
PredictedmodelResult = model.predict(LandAreainSqFt)
print(PredictedmodelResult)
```

[13687.72504892]

0s completed at 19:15

