

Supervised Learning: Linear Regression

Linear Regression is a machine learning algorithm based on supervised learning. Linear regression performs the task to predict a dependent variable value (y) based on a given independent variable (x).

Steps to build a Machine learning Model:

1. import dataset
2. visualizing the dataset
3. Data preparation
4. Training the algorithm
5. Visualizing the model
6. Making Prediction.

importing dataset

In []:

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

In [2]:

```
df=pd.read_csv('https://raw.githubusercontent.com/AdiPersonalWorks/Random/master/student_scores.csv')
df
```

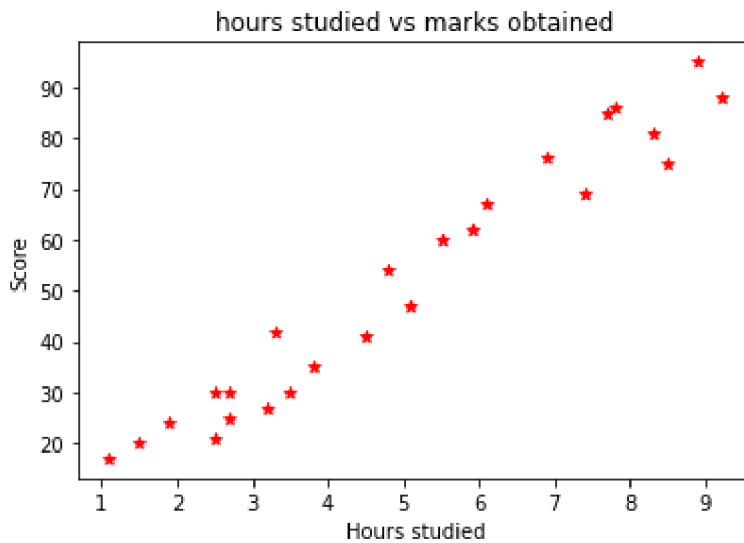
Out[2]:

	Hours	Scores
0	2.5	21
1	5.1	47
2	3.2	27
3	8.5	75
4	3.5	30
5	1.5	20
6	9.2	88
7	5.5	60
8	8.3	81
9	2.7	25
10	7.7	85
11	5.9	62
12	4.5	41
13	3.3	42
14	1.1	17
15	8.9	95
16	2.5	30
17	1.9	24
18	6.1	67
19	7.4	69
20	2.7	30
21	4.8	54
22	3.8	35
23	6.9	76
24	7.8	86

Visualizing the dataset

In [8]:

```
plt.title("hours studied vs marks obtained")
plt.scatter(df.Hours,df.Scores,color='red',marker='*')
plt.xlabel('Hours studied')
plt.ylabel('Score')
plt.show()
```



Data Perparation

In [14]:

```
#y=mx+c
..
lr=linear_model.LinearRegression()
lr.fit(df[['Hours']],df.Scores)
```

Out[14]:

LinearRegression()

In [15]:

```
#y(prediction output)=m(slope)x(Hours)+c(intercept)
m=lr.coef_
m
```

Out[15]:

array([9.77580339])

In [16]:

```
c=lr.intercept_  
c
```

Out[16]:

2.48367340537321

In [55]:

```
predicted_score=m*df['Hours']+c
```

Training the Algorithm

In []:

```
def predicting_score(hours):  
    total_hours=hours  
    score=m*hours+c  
    return total_hours,score
```

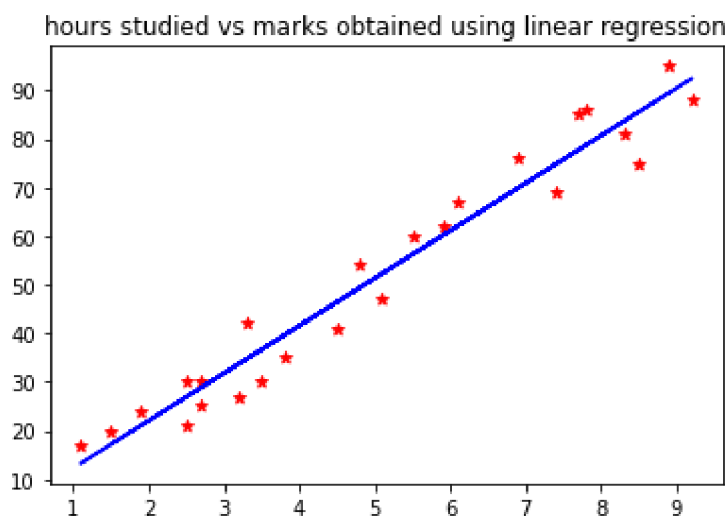
Visulaizing Model

In [57]:

```
plt.title('hours studied vs marks obtained using linear regression')  
plt.scatter(df.Hours,df.Scores,color='red',marker='*',label='data distribution')  
plt.plot(df.Hours,predicted_score,color='blue')
```

Out[57]:

[<matplotlib.lines.Line2D at 0xa963e38>]



Making Predictions

In [52]:

```
pr=predicting_score(9.25)
print('if student studies for',pr,'will be the score')
```

if student studies for (9.25, array([92.90985477])) will be score

Evaluating the model

In [54]:

```
from sklearn.metrics import mean_absolute_error
print('mean absolute error:',mean_absolute_error(df.Scores,predicted_score))
```

mean absolute error: 4.972805319866375