GRIP - MAY23 @ The Sparks Foundation

Name:- Hitesh Yadav

Task1 :- Prediction using Supervised ML

Problem Statement:-

- Predict the percentage of student based on the no. of study hours.
- What will be predicted score if a student studies for 9.25hr/day?

Importing basic Libraries

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import warnings
warnings.filterwarnings("ignore")
%matplotlib inline
```

Importing dataset

In [3]:

```
df = pd.read_csv("https://raw.githubusercontent.com/AdiPersonalWorks/Random/master/student_scores%20-%20student_scores.csv
df.head()
```

Out[3]:

	Hours	Scores
0	2.5	21
1	5.1	47
2	3.2	27
3	8.5	75
4	3.5	30

EDA

In [4]:

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25 entries, 0 to 24
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   Hours   25 non-null     float64
 1   Scores  25 non-null     int64  
dtypes: float64(1), int64(1)
memory usage: 528.0 bytes
```

In [5]:

```
df.describe()
```

Out[5]:

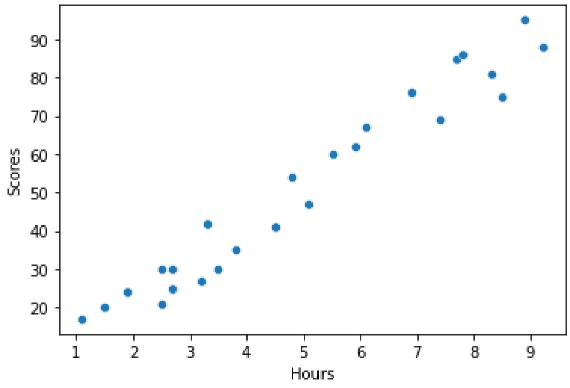
	Hours	Scores
count	25.000000	25.000000
mean	5.012000	51.480000
std	2.525094	25.286887
min	1.100000	17.000000
25%	2.700000	30.000000
50%	4.800000	47.000000
75%	7.400000	75.000000
max	9.200000	95.000000

In [6]:

```
sns.scatterplot("Hours","Scores",data  = df)
```

Out[6]:

```
<AxesSubplot:xlabel='Hours', ylabel='Scores'>
```



We can see there is the linear relationship between the features.

Splitting the data

In [8]:

```
X = df.drop("Scores",axis = 1)
Y = df['Scores']
```

In [9]:

```
# splitting the data in the training and testing.

from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(X,Y,test_size = 0.25,random_state=41)
```

Importing the linear regression model.

In [10]:

```
from sklearn.linear_model import LinearRegression
```

In [11]:

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

Out[11]:

```
LinearRegression()
```

In [12]:

```
print(lr.intercept_)
```

```
-0.21056271803429638
```

In [13]:

```
print(lr.coef_)
```

```
[10.1718186]
```

In [14]:

```
# checking the performance of model.
y_predict = lr.predict(x_test)
```

Evaluating the model

In [15]:

```
from sklearn.metrics import mean_absolute_error
from sklearn.metrics import mean_squared_error
from sklearn.metrics import r2_score
```

In [16]:

```
# mean absolute error
mean_absolute_error(y_test,y_predict)
```

Out[16]:

```
5.184007280448968
```

In [17]:

```
# mean squared error
mean_squared_error(y_test,y_predict)
```

Out[17]:

```
28.19312322015463
```

In [18]:

```
# R2 score
r2_score(y_test,y_predict)
```

Out[18]:

```
0.9628119134869286
```

In [19]:

```
# the accuracy for our project is 96.2%.
```

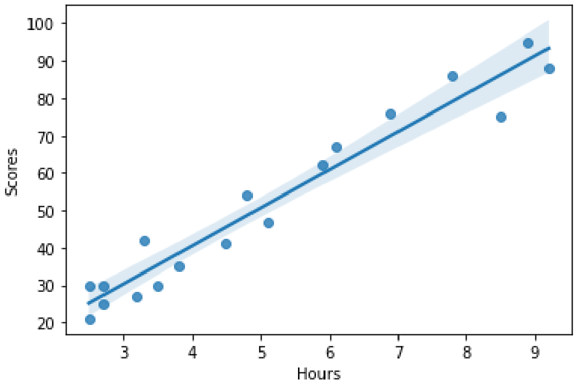
Plotting regression line.

In [20]:

```
sns.regplot(x_train,y_train)
```

Out[20]:

```
<AxesSubplot:xlabel='Hours', ylabel='Scores'>
```



In [22]:

```
# prediction of the student studying 9.25hrs/day
hrs = [[9.25]]
predict = lr.predict(hrs)
predict
```

Out[22]:

```
array([93.87875929])
```

The student studying 9.25hrs / day will likely to score 93.8%.

In [ ]: