GRIP : THE SPARKS FOUNDATION

DATA SCIENCE AND BUISNESS ANALYTICS

Prediction using Supervised ML

TASK1: Predict the percentage of an student based on the no. of study hours.
This is a simple linear regression task as it involves just 2 variables.

AUTHOR: Ali Ahmed Ansari intern at at The Spark Foundation

#GRIPJULY21

1)- IMPORT NECESSARY MODULE SUCH AS numpy,pandas and sklearn

In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as pt
```

2)- Import Linear Resgression from sklearn module

In [2]:
```
from sklearn.linear_model import LinearRegression
```

In [3]:
```
from sklearn.model_selection import train_test_split
```

3)- Import Dataset

In [4]:
```
data=pd.read_csv("https://raw.githubusercontent.com/AdiPersonalWorks/Random/master/student_
```

In [5]:
```
data.head()
```

Out[5]:

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

In [6]:
```
data.describe()    # DESCRIPTION
```

Out[6]:

	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 [7]:

```
data.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: 464.0 bytes
```

In [8]:

```
data_X=data.iloc[:,:1]
data_Y=data.iloc[:,1:]
print(data_X.head())   # Independent Variable
print(data_Y.head())   # Dependent Variable
```

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

4) - SPLITTING THE TRAIN AND TEST SAMPLES

In [9]:

```
x_train,x_test,y_train,y_test=train_test_split(data_X,data_Y,test_size=0.33)
```

In [10]:

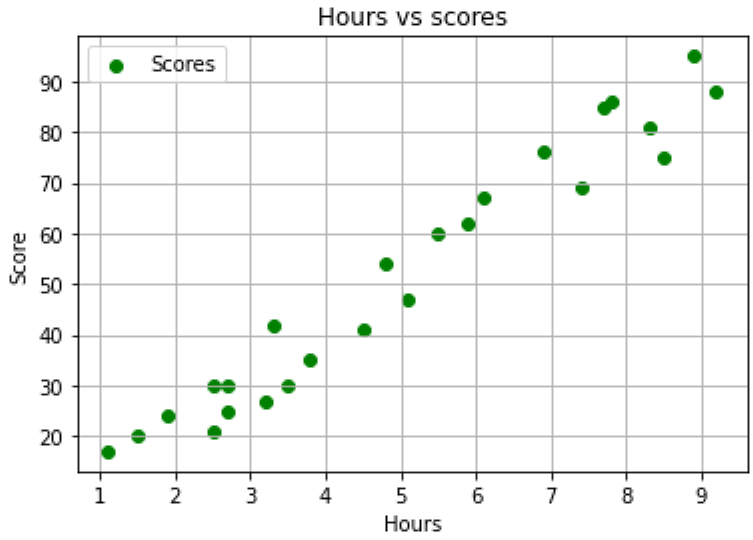
```
print("Train Size: ",len(x_train),len(y_train))
print("Test Size: ",len(x_test),len(y_test))
```

```
Train Size:  16 16
Test Size:  9 9
```

5)- DRAW INITIAL GRAPH

In [11]:

```
pt.scatter(data_X,data_Y,color='green',label='Scores')
pt.title("Hours vs scores")
pt.xlabel("Hours")
pt.ylabel("Score")
pt.legend()
pt.grid()
pt.show()
```



6)- Create Linear Regression Model

```
In [12]: model=LinearRegression()
```

7)- Fit the Model(Training The Model)

```
In [13]: model.fit(x_train,y_train)

Out[13]: LinearRegression()
```

8)- Predict the Test case

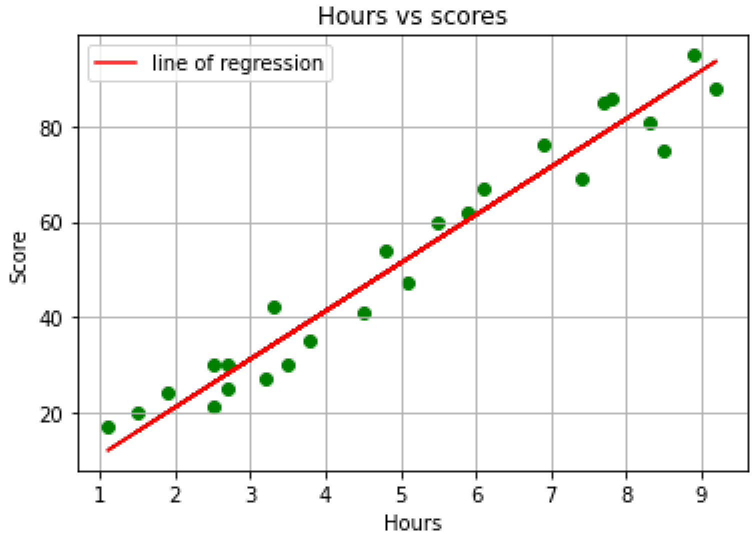
```
In [14]: y_predict=model.predict(x_test)
```

```
In [15]: intercept=model.intercept_     # Intercept value(c)
         slope=model.coef_              # Slope Value(m)
         line=slope*data_X+intercept         # Linear Line(y=mx+c)
         print("SLOPE= ",slope)
         print("INTERCEPT= ",intercept)

         SLOPE=  [[10.09834587]]
         INTERCEPT=  [0.83286837]
```

9)- DRAW FINAL GRAPH WITH LINEAR REGRESSION BEST FITTED LINE

```
In [16]: pt.scatter(data_X,data_Y,color='green')
         pt.plot(data_X,line,color='red',label='line of regression')
         pt.title("Hours vs scores")
         pt.xlabel("Hours")
         pt.ylabel("Score")
         pt.legend()
         pt.grid()
         pt.show()
```



QUESTION)- What will be predicted score if a student studies for 9.25 hrs/ day?

```
In [17]: answer=model.predict([[9.25]])
         print("If the student study 9.25 hours they will get: ",round(float(answer),2)," Score")

         If the student study 9.25 hours they will get:  94.24  Score
```

10)- ACCURACY CHECK OF LINEAR MODEL

```
In [18]: from sklearn.metrics import mean_squared_error,mean_absolute_error,r2_score   # import nece
```

```
In [19]: print("mean squared error: ",mean_squared_error(y_test,y_predict))
         print("mean_absolute_error: ",mean_absolute_error(y_test,y_predict))
         print("r2_score: ",r2_score(y_test,y_predict))

         mean squared error:  19.236167186546638
         mean_absolute_error:  4.181039692156442
         r2_score:  0.9624800245109257
```