

GRIP : THE SPARKS FOUNDATION

DATA SCIENCE AND BUISNESS ANALYTICS

Prediction using Supervised ML

Task 1 : 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 : Md Saqib intern at at The Spark Foundation

#GRIPJULY21

1) : IMPORT REQUIRED MODULE

In [1]:

```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

2) : Import Linear Resgression from Scikit-learn 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: 528.0 bytes
```

In [8]:

```
X=data.iloc[:,1]
Y=data.iloc[:,1:]
print(X.head())    # Independent Variable
print(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(X,Y,test_size=0.33)
```

In [10]:

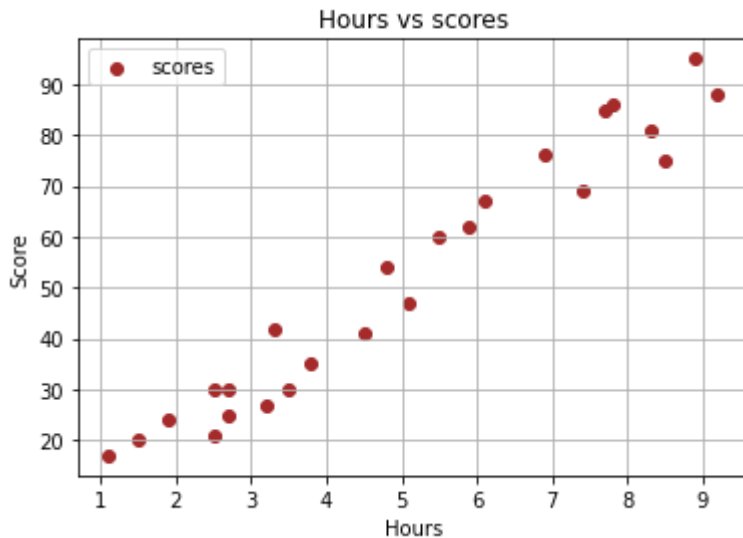
```
print("Train Size: ",len(x_train),len(y_train))
print("Test Size: ",len(x_test),len(y_test))
```

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

5) : DRAW INITIAL GRAPH

In [11]:

```
pt.scatter(X,Y,color='brown',label='scores')
pt.title("Hours vs scores")
pt.xlabel("Hours")
pt.ylabel("Score")
pt.legend()
pt.grid()
pt.show()
```



6) : Training the Algorithm

In [12]:

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

Out[12]:

LinearRegression()

7) : Predict the test case

In [13]:

```
y_predict=model.predict(x_test)
```

In [14]:

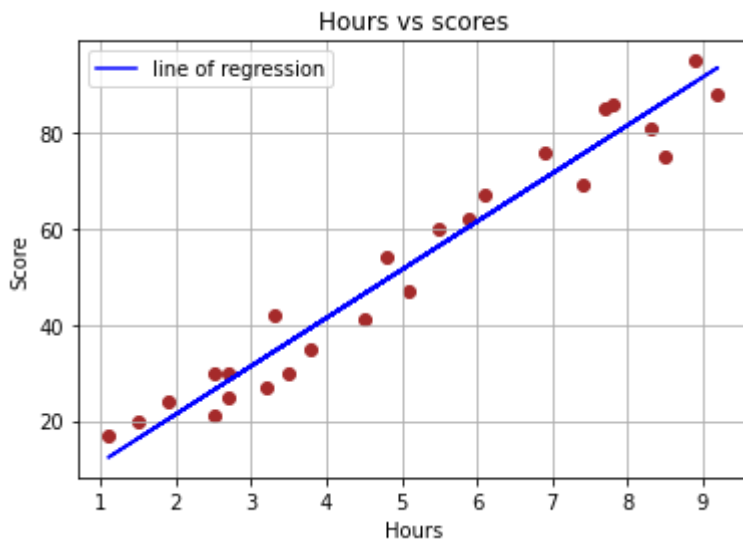
```
intercept=model.intercept_      # Intercept value(c)
slope=model.coef_               # Slope Value(m)
line=slope*X+intercept          # Linear Line(y=mx+c)
print("SLOPE= ",slope)
print("INTERCEPT= ",intercept)
```

```
SLOPE= [[10.02915586]]
INTERCEPT= [1.31121867]
```

8) : DRAW FINAL GRAPH WITH LINEAR REGRESSION BEST FITTED LINE

In [15]:

```
pt.scatter(X,Y,color='brown')
pt.plot(X,line,color='blue',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 [16]:

```
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: 94.08 Score

9) : ACCURACY CHECK OF LINEAR MODEL

In [17]:

```
from sklearn.metrics import mean_squared_error,mean_absolute_error,r2_score # import nece
```

In [18]:

```
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: 36.02623986613959
mean_absolute_error: 5.524215243765806
r2_score: 0.9565961829313823
```

THANKYOU