THE SPARKS FOUNDATION

#GRIPAPR22

Author : Devarapalli Ruthwik Reddy

TASK-1 Prediction using Supervised ML

In this task we will predict the percentage of marks that a student is expected to score based on the number of hours they studied and we will also predict the percentage of a student if he studies for 9.25hrs/day

This is a simple regression task as it invovles only two variables.

In [3]:

```
# importing all required libraries for this task
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [5]:

```
#import dataset
students_data=pd.read_csv("http://bit.ly/w-data")
print(students_data.head())
```

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

In [7]:

```
students_data.size
```

Out[7]:

```
50
```

In [8]:

```
students_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
```

From above output we can see that there are no Null values in data set. so we need not to bother about existence of null values.

In [9]:

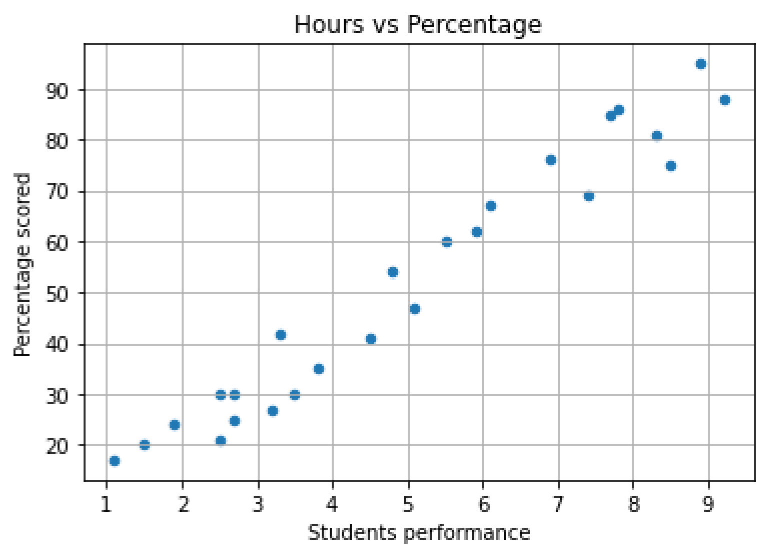
```
students_data.describe()
```

Out[9]:

	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 [104…

```
students_data.plot(kind="scatter",x="Hours",y="Scores",title="Hours vs Percentage",xlabel="Students performance",ylabel="Percentage scored")
plt.grid()
```



From the above graph we can coclude that there is a positive linear relation (proportionality) between the number of hours studied and percentage of score secured by the student.

Splitting the data

In [101…

```
# divide independent and dependent data
x=students_data.iloc[:,:-1].values#hours
y=students_data.iloc[:,-1].values#score
```

In [81]:

```
#split data into train and test sets.
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)
print(len(x_test),len(x_train))
print(len(y_test),len(y_train))
```

```
5 20
5 20
```

Training the model

In [87]:

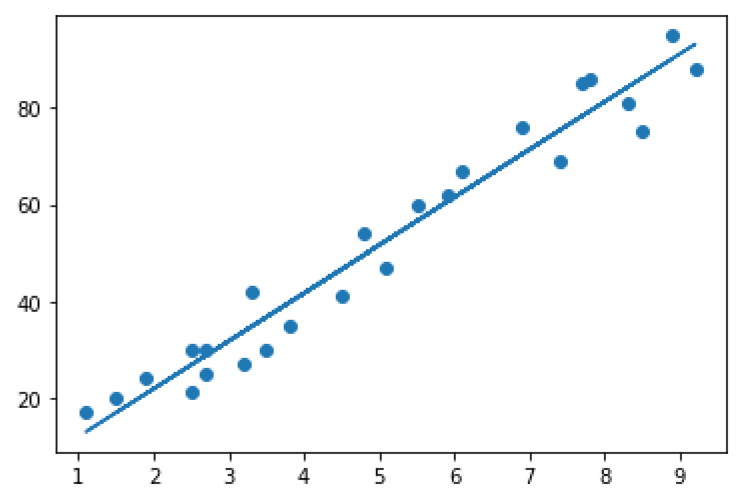
```
from sklearn.linear_model import LinearRegression
reg=LinearRegression()
reg.fit(x_train,y_train)
print("------Model trained------")
```

```
------Model trained------
```

In [92]:

```
#plot regression line of form mx+c
line=reg.coef_*x+reg.intercept_

#plot for test data
plt.scatter(x,y)
plt.plot(x,line)
plt.show()
```



Making predictions

In [93]:

```
y_pred=reg.predict(x_test)#predicting the scores with help of x_test data
```

In [96]:

```
#comapre actual vs predicted values
dataframe=pd.DataFrame({'Actual':y_test,'Predicted':y_pred})
dataframe
```

Out[96]:

	Actual	Predicted
0	20	16.884145
1	27	33.732261
2	69	75.357018
3	30	26.794801
4	62	60.491033

Predicted score of student if he studies for 9.25hrs/day.

In [99]:

```
hours=9.25
own_pred=reg.predict([[hours]])
print(f"no of hours = {hours}")
print(f"Predicted Score = {own_pred[0]}")
```

```
no of hours = 9.25
Predicted Score = 93.69173248737538
```

Evaluating model

In [100…

```
from sklearn import metrics
print("Mean Absolute Error:",metrics.mean_absolute_error(y_test,y_pred))
```

```
Mean Absolute Error: 4.183859899002975
```

Smaller the value of mean absolute error lesser the chances of error

Conclusion:

If a student studies for 9.25hrs/day he will be scoring 93.69%.