

**Author- VALIVETI.SRIRAM SAI**

**Data Science & Business Analytics Internship-The Sparks Foundation**

### **Task-1 -Prediction using Supervised ML**

**The Aim of the task is to predict the percentage of an student based on the number of study ho supervised machine learning algorithm**

**STEP-1 : Importing the DataSet** Here we would be importing all the libraries required in this notebook.

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

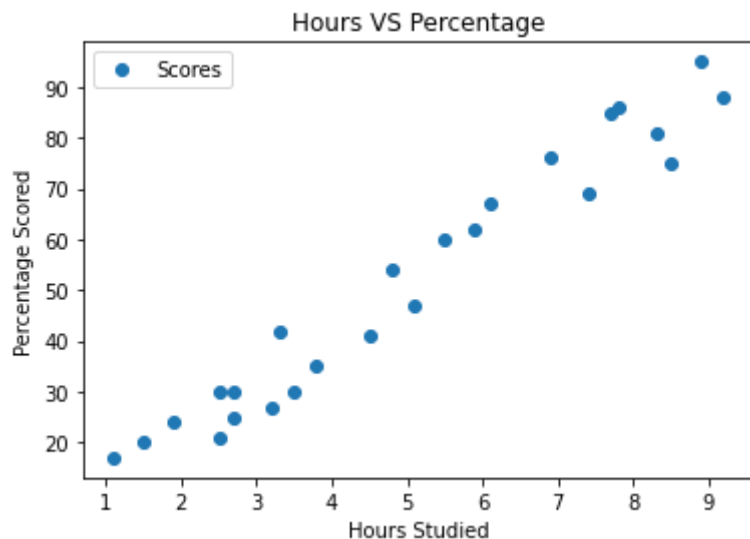
```
url="http://bit.ly/w-data"
data=pd.read_csv(url)
print("Data has Imported Successfully")
data.head(10)
```

☞ Data has Imported Successfully

	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

**STEP-2 : Visualizing the Data**

```
data.plot(x="Hours",y="Scores",style='o')
plt.title('Hours VS Percentage')
plt.xlabel('Hours Studied')
plt.ylabel('Percentage Scored')
plt.show()
```



From the above Step There is a relationship between the variables

### STEP-3 : Preparation of Data

```
x=data.iloc[:, :-1].values  
y=data.iloc[:, 1].values
```

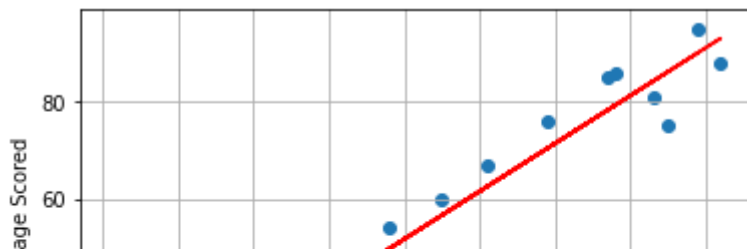
```
from sklearn.model_selection import train_test_split  
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.2,random_state=0)
```

### STEP-4 : Training The Algorithm

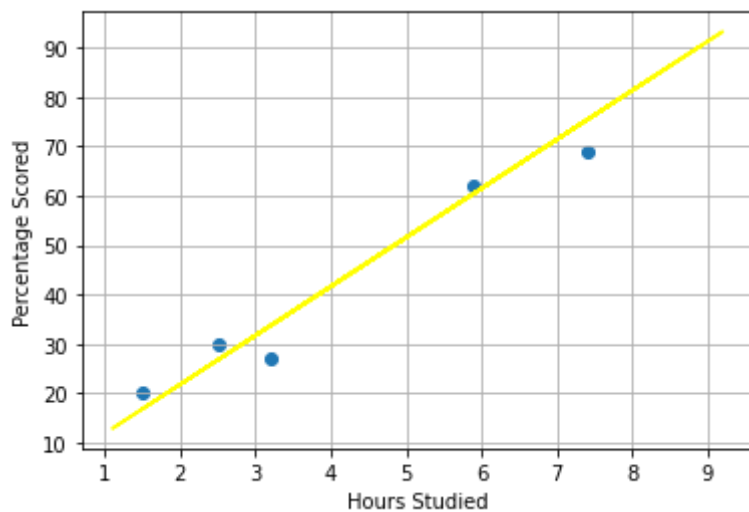
```
from sklearn.linear_model import LinearRegression  
regressor=LinearRegression()  
regressor.fit(x_train,y_train)  
print("Traing Completed...")
```

Traing Completed...

```
line=regressor.coef_*x+regressor.intercept_  
plt.scatter(x_train,y_train)  
plt.plot(x,line,color='red')  
plt.xlabel('Hours Studied')  
plt.ylabel('Percentage Scored')  
plt.grid()  
plt.show()
```



```
plt.scatter(x_test,y_test)
plt.plot(x,line,color='yellow')
plt.xlabel('Hours Studied')
plt.ylabel('Percentage Scored')
plt.grid()
plt.show()
```



### STEP-5 : Making Predictions

```
print(x_test) #Testing the data in hours
y_pred=regressor.predict(x_test) #Predicting the scores
```

```
[[1.5]
 [3.2]
 [7.4]
 [2.5]
 [5.9]]
```

```
df=pd.DataFrame({'Actual':y_test,'Predicted':y_pred})
df
```

**Actual Predicted**

Testing with the own data

1 27 33.732261

```
hours=9.5
own_pred=regressor.predict([[hours]])
print("No of Hours ={}".format(hours))
print("Predicted Score={}".format(own_pred[0]))
```

```
No of Hours =9.5
Predicted Score=96.16939660753593]
```

**STEP-6 : Evaluating the Data**

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

```
The Mean Absolute Error: 4.183859899002982
```

✓ 0s completed at 5:04 PM

