

# Author: VIVEK KUMAR NIGAM

DATA SCIENCE & BUSINESS ANALYTICS @ TSF

## GRIP\_NOV'21 TASK\_01

SUPERVISED MACHINE LEARNING USING LINEAR REGRESSION

```
In [149]: import pandas as pd
import os
import sklearn as skl
import matplotlib.pyplot as plt
```

```
In [150]: os.getcwd()
```

```
Out[150]: 'C:\\Users\\VIVEK\\Desktop\\TSF-INTERN'
```

```
In [151]: os.chdir("C:\\Users\\VIVEK\\Desktop\\TSF-INTERN")
```

```
In [152]: os.getcwd()
```

```
Out[152]: 'C:\\Users\\VIVEK\\Desktop\\TSF-INTERN'
```

```
In [153]: sd = pd.read_csv("student_scores - student_scores.csv")
```

```
In [154]: sd.shape
```

```
Out[154]: (25, 2)
```

```
In [155]: sd.info()
```

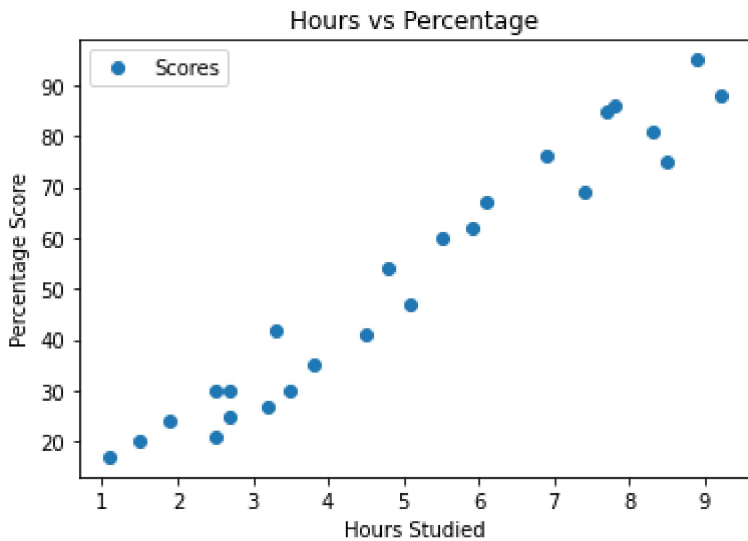
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25 entries, 0 to 24
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype  
---  -
 0   Hours   25 non-null      float64
 1   Scores  25 non-null      int64   
dtypes: float64(1), int64(1)
memory usage: 528.0 bytes
```

```
sd.describe()
```

Out[156]:

	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 [157]: # Plotting
sd.plot(x='Hours', y='Scores', style='o')
plt.title('Hours vs Percentage')
plt.xlabel('Hours Studied')
plt.ylabel('Percentage Score')
plt.show()
```



## Plot shows linear relation

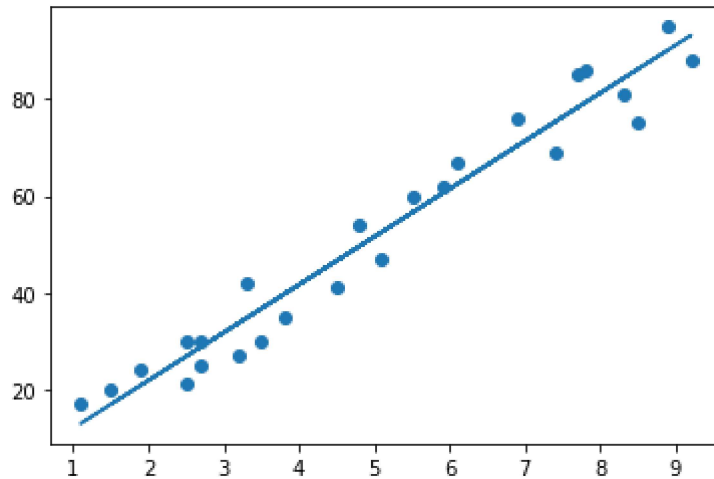
```
In [158]: X = sd.iloc[:, :-1].values
          y = sd.iloc[:, 1].values
```

[illegible]

```
In [160]: from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

Out[160]: LinearRegression()

```
In [161]: l = regressor.coef_*X+regressor.intercept_
plt.scatter(X, y)
plt.plot(X, l);
plt.show()
```



```
In [175]: print(X_test)
y_pred = regressor.predict(X_test)
```

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

```
In [163]: df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
df
```

Out[163]:

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

```
In [169]: hrs = [[9.25]]
own_pred = regressor.predict(hrs)
print("No of Hours =", hrs)
print("Predicted Score =", own_pred[0])
```

```
No of Hours = [[9.25]]
Predicted Score = 93.69173248737538
```

```
In [165]: from sklearn import metrics  
print('Mean Absolute Error:',metrics.mean_absolute_error(y_test, y_pred))
```

Mean Absolute Error: 4.183859899002975