* importing required libraries

In [108…

```
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
from pandas_profiling import ProfileReport
```

* using student_score dataset

In [174…

```
data=pd.read_csv("student_scores.csv")
```

In [116…

```
data.head()
```

Out[116…

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

In [117…

```
data.describe()
```

Out[117…

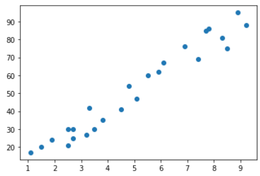
	Hours	Scores
count	25.000000	25.000000
mean	5.012000	51.480000
std	2.525094	25.286887
min	1.100000	17.000000
25%	2.700000	30.000000
50%	4.800000	47.000000
75%	7.400000	75.000000
max	9.200000	95.000000

In [120…

```
plt.scatter(data["Hours"],data["Scores"])
```

Out[120…

```
<matplotlib.collections.PathCollection at 0x2ca29a06190>
```



- There is linear co-relation between data

* Training data

In [186…

```
x=data.iloc[:,:1]
x
```

Out[186…

	Hours
0	2.5
1	5.1
2	3.2
3	8.5
4	3.5
5	1.5
6	9.2
7	5.5
8	8.3
9	2.7
10	7.7
11	5.9
12	4.5
13	3.3
14	1.1
15	8.9
16	2.5
17	1.9
18	6.1
19	7.4
20	2.7
21	4.8
22	3.8
23	6.9
24	7.8

In [159…

```
y=data["Scores"]
y
```

Out[159…

```
0     21
1     47
2     27
3     75
4     30
5     20
6     88
7     60
8     81
9     25
10    85
11    62
12    41
13    42
14    17
15    95
16    30
17    24
18    67
19    69
20    30
21    54
22    35
23    76
24    86
Name: Scores, dtype: int64
```

In [168…

```
regr=LinearRegression()
```

In [187…

```
regr.fit(x,y)
```

Out[187…

```
LinearRegression()
```

In [188…

```
regr.coef_
```

Out[188…

```
array([9.77580339])
```

In [189…

```
regr.intercept_
```

Out[189…

```
2.483673405373196
```

* MAking prediction

In [198…

```
for i in data["Hours"]:
    print(regr.predict([[i]]))
```

```
[26.92318188]
[52.3402707]
[33.76624426]
[85.57800223]
[36.69898527]
[17.14737849]
[92.4210646]
[56.25059205]
[83.62284155]
[28.87834256]
[77.7573951]
[60.16091341]
[46.47478866]
[34.74382459]
[13.23705714]
[89.48832358]
[26.92318188]
[21.05769985]
[62.11607409]
[74.8246185]
[28.87834256]
[49.40752968]
[39.63172629]
[69.9367168]
[78.73493985]
```

* Taking input and getting predicted output

In [199…

```
inp=int(input("Enter a study hours"))
regr.predict([[inp]])
```

```
Enter a study hours7
```

Out[199…

```
array([70.91429714])
```

In [ ]: