In [1]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [2]:
```
df=pd.read_csv(r'C:\DATA\task_1.csv')
df
```

Out[2]:

	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
10	7.7	85
11	5.9	62
12	4.5	41
13	3.3	42
14	1.1	17
15	8.9	95
16	2.5	30
17	1.9	24
18	6.1	67
19	7.4	69
20	2.7	30
21	4.8	54
22	3.8	35
23	6.9	76
24	7.8	86

In [3]:
```
#checking diamension and information of the data
df.shape
df.info
```

Out[3]:
```
<bound method DataFrame.info of     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
10    7.7      85
11    5.9      62
12    4.5      41
13    3.3      42
14    1.1      17
15    8.9      95
16    2.5      30
17    1.9      24
18    6.1      67
19    7.4      69
20    2.7      30
21    4.8      54
22    3.8      35
23    6.9      76
24    7.8      86>
```

Checking Missing Value

In [4]:
```
df.isnull().values.any()
```

Out[4]:
```
False
```

Exploring The Data

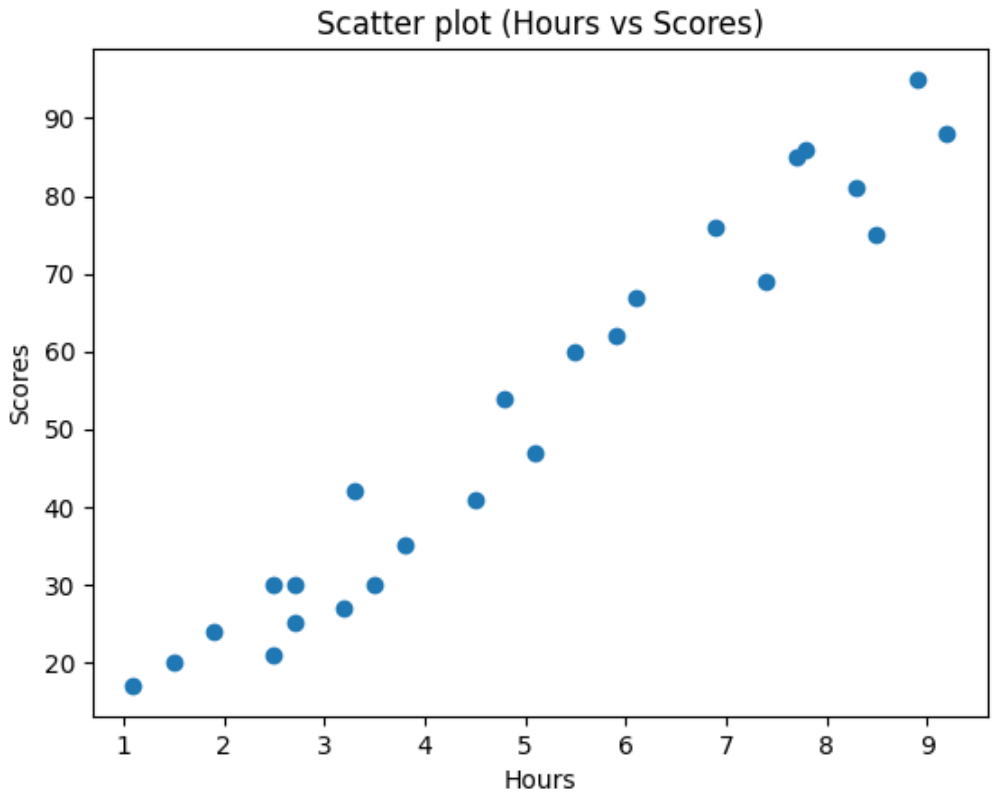
Statistical Summary

In [5]:
```
df.describe()
```

Out[5]:

	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 [6]:
```
plt.scatter(df.Hours,df.Scores)
plt.title("Scatter plot (Hours vs Scores)")
plt.ylabel("Scores")
plt.xlabel("Hours")
plt.show()
```



correlation Between Hours and Scores

In [7]:
```
df.corr()
```

Out[7]:

	Hours	Scores
Hours	1.000000	0.976191
Scores	0.976191	1.000000

Conclusion: From above scatter plot we see that most of points are linearly correlated,so we can use Simple_linear_Regression and correlation between Hours and Scores is 0.976191 so both are positively (strongly) correlated to each others

Simple Linear Regression

In [8]:
```
#splitting data into dependent(y) and independent(x) variable
x=df.iloc[:,:-1].values
y=df.iloc[:,1].values
```

In [9]:
```
# split data into train and test
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test = train_test_split(x,y,test_size=1/3,random_state=3)
```

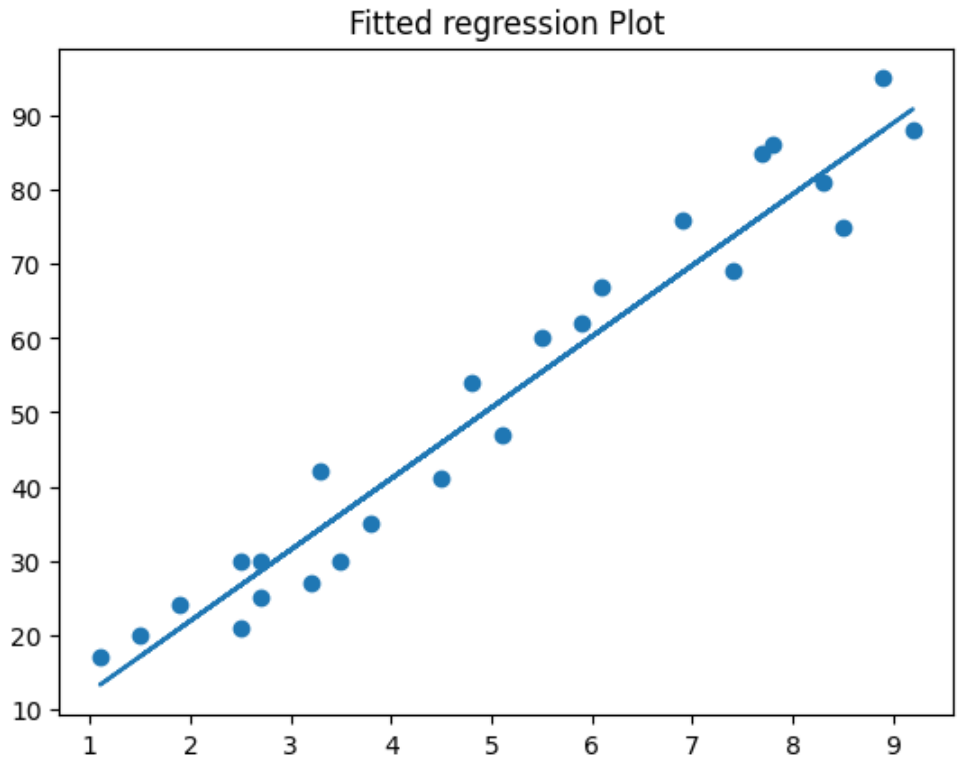
In [18]:
```
from sklearn.linear_model import LinearRegression
model=LinearRegression()
model.fit(x_train,y_train)
```

In [26]:
```
#fitted model
y_hat=model.coef_*x+model.intercept_
print("slope=",model.coef_)
print("intercept=",model.intercept_)
```

```
slope= [9.58147869]
intercept= 2.730963545948896
```

In [29]:
```
#plotting regression line
plt.scatter(x,y)
plt.plot(x,y_hat)
plt.title('Fitted regression Plot')
plt.show
```

Out[29]:
```
<function matplotlib.pyplot.show(close=None, block=None)>
```



In [28]:
```
#predicted value of y
y_pred = model.predict(x_test)
```

In [16]:
```
from sklearn.metrics import r2_score
R_square = r2_score(y_test,y_pred)
print(R_square)
```

```
0.9371605994687603
```

conclusion: 93.71% of the variability observed in the responce variable is explained by the regression model

In [ ]: