Task1: Predicting the Score based on the number of hours studied

Importing Necessary Libraries

In [45]:
```
import pandas as pd
import numpy as np
from sklearn import linear_model
import matplotlib.pyplot as plt
from sklearn.model_selection import train_test_split
from sklearn.metrics import mean_squared_error
```

Importing Data

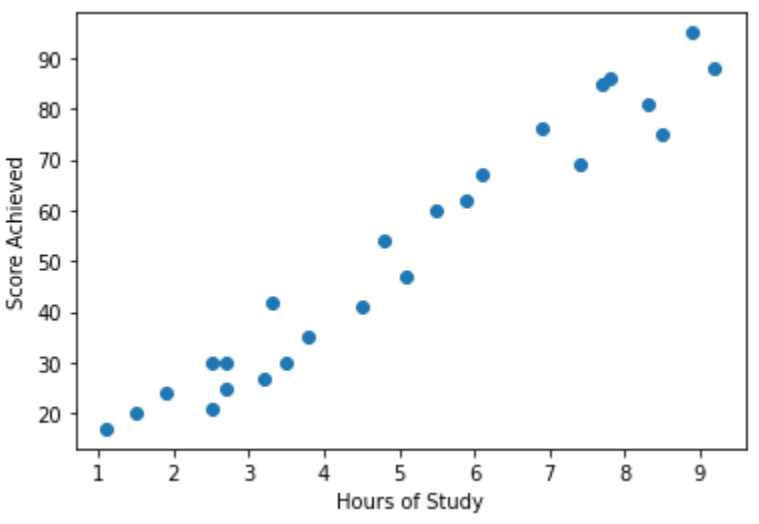
In [3]:
```
df = pd.read_csv("https://raw.githubusercontent.com/AdiPersonalWorks/Random/master/stu
df
```

Out[3]:

	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

Visualizing the Data

In [14]:
```
x = df['Hours']
y = df['Scores']
plt.scatter(x, y)
plt.xlabel('Hours of Study')
plt.ylabel('Score Achieved')
plt.show()
```



Defining attributes and target for data splitting

In [17]:
```
x = df[['Hours']]
y = df['Scores']
xtrain, xtest, ytrain, ytest = train_test_split(x, y, test_size = 0.2, random_state =
```

Model Development

In [18]:
```
reg = linear_model.LinearRegression()
```

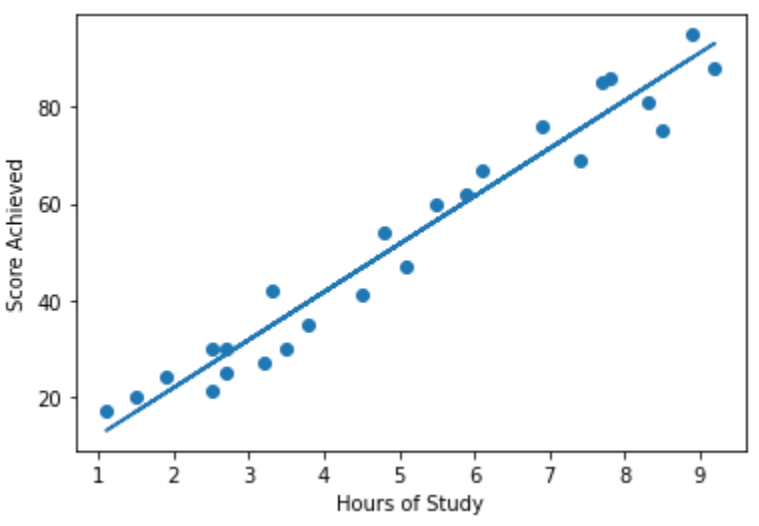
Model Training

In [19]:
```
reg.fit(xtrain,ytrain)
```

Out[19]:
```
LinearRegression()
```

Constructing the Regressor Line

In [23]:
```
line = reg.coef_*x + reg.intercept_
plt.scatter(x, y)
plt.plot(x, line);
plt.xlabel('Hours of Study')
plt.ylabel('Score Achieved')
plt.show()
```



Actual Data v/s Predicted Data

In [44]:
```
ypred = reg.predict(xtest)
df = pd.DataFrame({'Actual': ytest, 'Predicted': ypred})
df
```

Out[44]:

	Actual	Predicted
5	20	16.884145
2	27	33.732261
19	69	75.357018
16	30	26.794801
11	62	60.491033

Model Evaluation

Coefficient of determination or R squared Value

In [8]:
```
reg.score(x,y)
```

Out[8]:
```
0.9529481969048356
```

Predicting Score for someone who studies for 9.25 hours

In [11]:
```
reg.predict(np.array([[9.25]]))
```

```
c:\users\user\appdata\local\programs\python\python39\lib\site-packages\sklearn\base.p
y:450: UserWarning: X does not have valid feature names, but LinearRegression was fitt
ed with feature names
  warnings.warn(
```

Out[11]:
```
array([92.90985477])
```