Linear Regression with one variable

Create a Linear regression Model that can predict the score of student based on the number of hour student studied.

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("student_scores.csv")
```

Dataset describe

The dataset contains 2 columns (Hours and Scores)

Hours : Hours spend by student studying for the test

Scores: Score received by student for the corresponding study hour

In [3]:

```
df.head()
```

Out[3]:

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

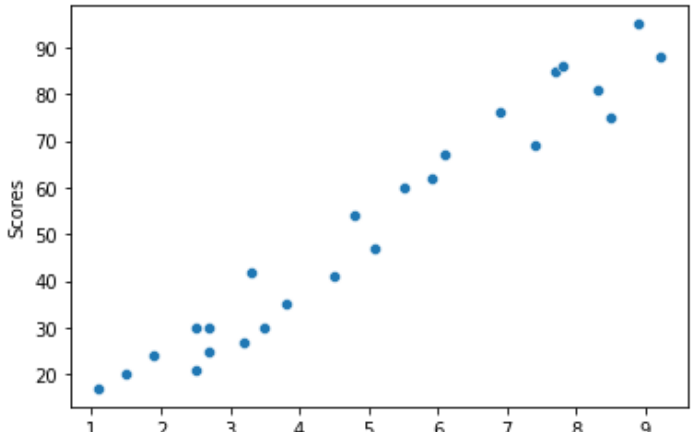
In [3]:

```
#scatter plot to visualize the distribution of the data. We can clearly see that there is a clear +ve correlation.

sns.scatterplot(x='Hours',y='Scores',data=df)
```

Out[3]:

```
<AxesSubplot:xlabel='Hours', ylabel='Scores'>
```



Hours

In [4]:

```
X = df['Hours']
y = df['Scores']
print(X.shape)
print(y.shape)
```

```
(25,)
(25,)
```

In [5]:

```
from sklearn.linear_model import LinearRegression
```

In [6]:

```
from sklearn.model_selection import train_test_split
```

In [7]:

```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.3, random_state=10
1)
```

In [8]:

```
#Creating linear model
model = LinearRegression()
```

In [9]:

```
model.fit(X_train.values.reshape(-1, 1),y_train)
```

Out[9]:

```
LinearRegression()
```

In [10]:

```
#We have to reshape as the model is not designed to automatically convert the shape of th
e data, but it can do for multi-feature data.
test_predictions=model.predict(X_test.values.reshape(-1,1))
```

In [11]:

```
#Error metrices

from sklearn.metrics import mean_absolute_error,mean_squared_error
MAE = mean_absolute_error(y_test,test_predictions)
MSE = mean_squared_error(y_test,test_predictions)
RMSE = np.sqrt(MSE)
```

In [12]:

```
print(MAE)
```

```
5.864954643694258
```

In [13]:

```
print(MSE)
```

```
41.63486564552886
```

In [14]:

```
print(RMSE)
```

```
6.452508476982333
```

In [15]:

```
model.coef_
```

Out[15]:

```
array([9.96651548])
```

In [16]:

```
# Model predict on random/unseen data

#1 - score when student study for 5 hr
model.predict([[5]])
```

Out[16]:

```
array([51.93507047])
```

In [17]:

```
#2 - score when student study for 4 hr
model.predict([[4]])
```

Out[17]:

```
array([41.96855499])
```

In [19]:

```
#3 - score when student study for 1 hr
model.predict([[1]])
```

Out[19]:

```
array([12.06900855])
```

In [ ]:

```

```