Linear Regression

Predict the percentage of marks of an student based on the number of study hours

1. Importing Required Libraries

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
%matplotlib inline
```

2. Importing Dataset from url

In [2]:

```
url= 'http://bit.ly/w-data'
dataset= pd.read_csv(url)
dataset.head()
```

Out[2]:

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

In [3]:

```
dataset.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
```

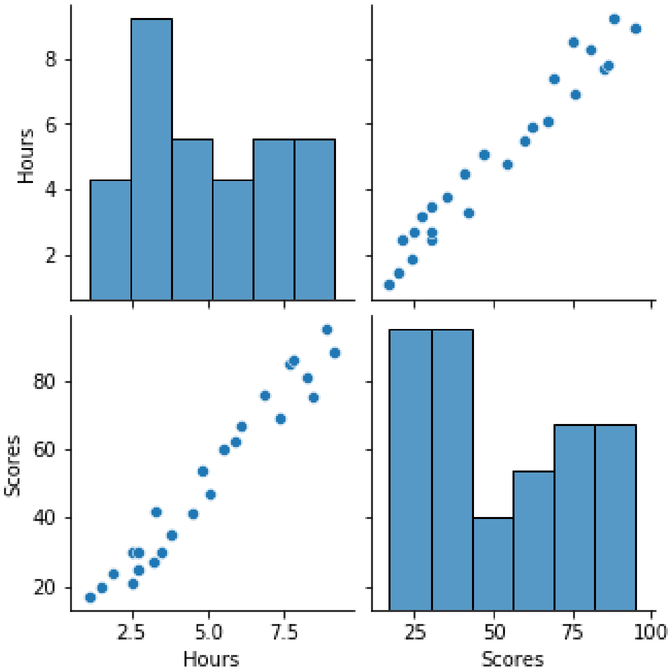
3. Visualizing Dataset

In [4]:

```
sns.pairplot(dataset)
```

Out[4]:

```
<seaborn.axisgrid.PairGrid at 0x7f962cd63550>
```



4. Splitting Dataset into Deoendent Variable (y) and Independent Variable (x)

In [5]:

```
x= dataset.iloc[:,0:1]
x.head()
```

Out[5]:

	Hours
0	2.5
1	5.1
2	3.2
3	8.5
4	3.5

In [6]:

```
y= dataset.iloc[:,1:]
y.head()
```

Out[6]:

	Scores
0	21
1	47
2	27
3	75
4	30

5. Splitting Training and Testing Dataset

In [7]:

```
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test= train_test_split(x,y,test_size=0.2)
print(x_train.shape, y_train.shape, x_test.shape, y_test.shape)
```

```
(20, 1) (20, 1) (5, 1) (5, 1)
```

6. Fitting Training Data in Linear Regression Model

In [8]:

```
from sklearn.linear_model import LinearRegression
model= LinearRegression()
model.fit(x_train, y_train)
predictions= model.predict(x_test)
```

7. Comparing Predicted Data with Testing Data

In [9]:

```
comparison= pd.DataFrame(np.c_[y_test, predictions],columns=['Original Score','Prediction
s'])
comparison
```

Out[9]:

	Original Score	Predictions
0	75.0	85.991095
1	86.0	79.050911
2	42.0	34.435440
3	27.0	33.443985
4	54.0	49.307263

1. Mean Squared Error
2. Mean Absolute Error
3. Root Mean Squared Error

In [10]:

```
from sklearn import metrics
print('MSE:',metrics.mean_squared_error(y_test, predictions))
print('MAE:',metrics.mean_absolute_error(y_test, predictions))
print('RMSE:',np.sqrt(metrics.mean_squared_error(y_test, predictions)))
```

```
MSE: 57.972661064210115
MAE: 7.32829323626246
RMSE: 7.61397800523551
```

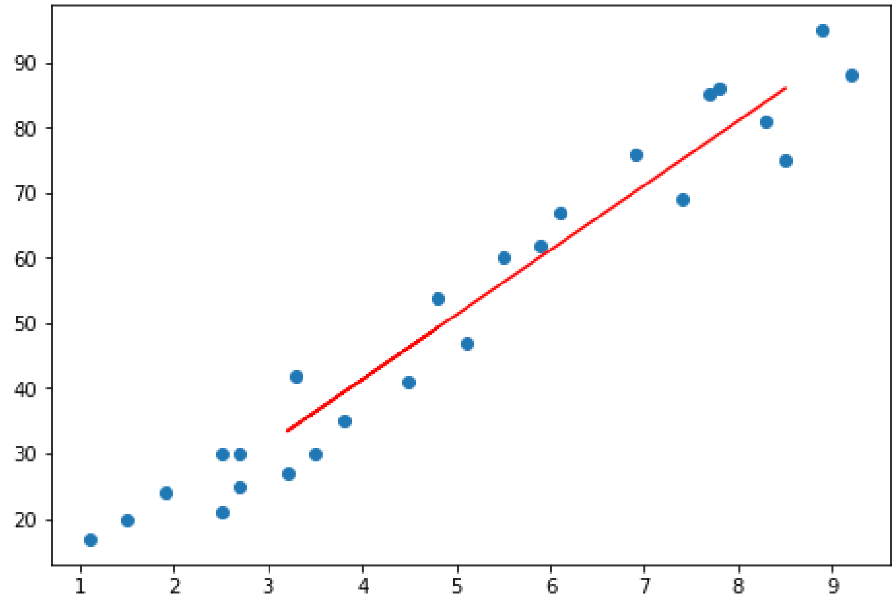
8. Regression Line

In [11]:

```
fig= plt.figure()
axes= fig.add_axes([0,0,1,1])
axes.scatter(x,y)
axes.plot(x_test,predictions,color='red')
```

Out[11]:

```
[<matplotlib.lines.Line2D at 0x7f9622204e10>]
```



9. Calculating Score for stuying 9.5 hrs/day

In [12]:

```
A= model.predict([[9.5]])
print('If student studies 9.5 hrs/day he would get {} percentage.'.format(A[0][0]))
```

```
If student studies 9.5 hrs/day he would get 95.90564441862209 percentage.
```