

In [7]:

```
import numpy as np
import pandas as pd
import matplotlib as plt
from matplotlib import pyplot as plt
import seaborn as sns
plt.style.use('seaborn-white')#Set the matplotlib stylesheet
```

In [8]:

```
sco=pd.read_csv('score.csv')
```

In [54]:

```
sco.head()
```

Out[54]:

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

In [52]:

```
sco.shape
```

Out[52]:

```
(25, 2)
```

In [11]:

sco.info

Out[11]:

	<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	

In [12]:

sco.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

```

In [13]:

```
sco.isnull()
```

Out[13]:

	Hours	Scores
0	False	False
1	False	False
2	False	False
3	False	False
4	False	False
5	False	False
6	False	False
7	False	False
8	False	False
9	False	False
10	False	False
11	False	False
12	False	False
13	False	False
14	False	False
15	False	False
16	False	False
17	False	False
18	False	False
19	False	False
20	False	False
21	False	False
22	False	False
23	False	False
24	False	False

In [14]:

```
sco.sum()
```

Out[14]:

```
Hours      125.3
Scores     1287.0
dtype: float64
```

In [15]:

```
sco.isnull().sum()
```

Out[15]:

```
Hours      0  
Scores     0  
dtype: int64
```

Exploratory Data Analysis(EDA)

In [29]:

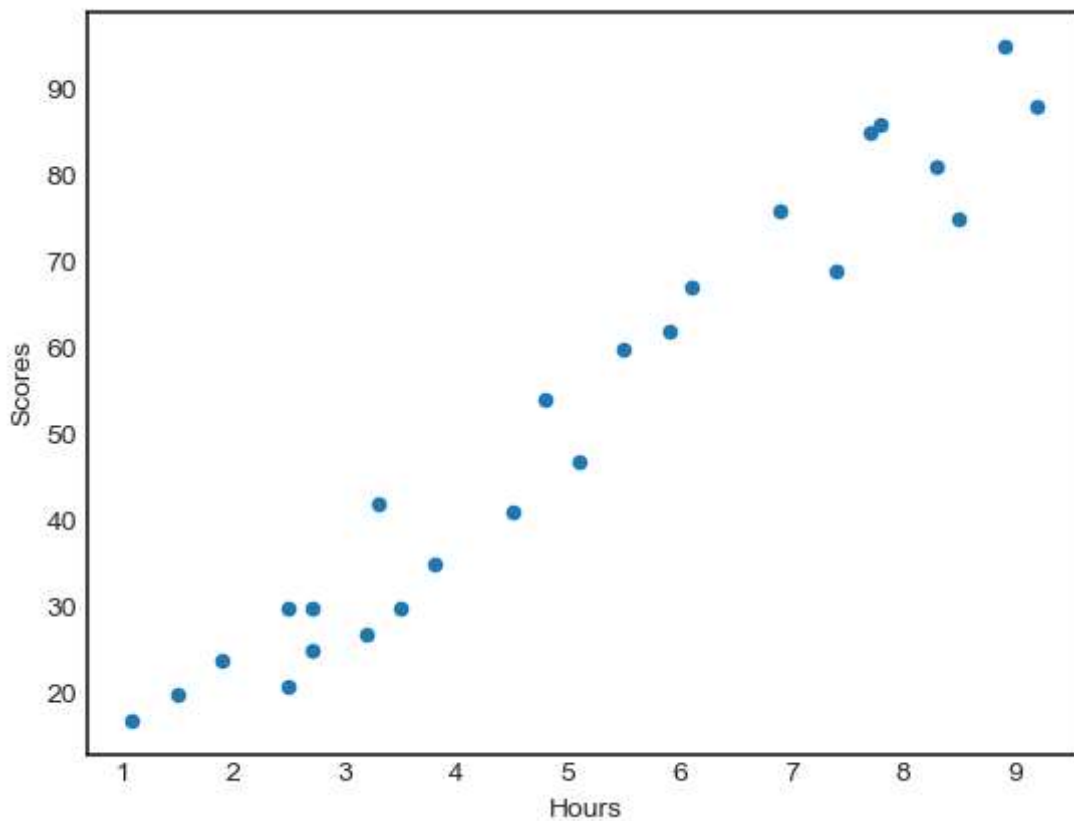
```
import matplotlib.pyplot as plt  
import seaborn as sns
```

In [79]:

```
sco.plot(kind='scatter',x='Hours',y='Scores')
```

Out[79]:

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



In [34]:

```
feature=sco[["Hours"]] #independent variable (X)  
target=sco[["Scores"]] #dependent variable (Y)
```

In [37]:

```
sco.iloc[:,1:]  
feature.head()
```

Out[37]:

	Hours
0	2.5
1	5.1
2	3.2
3	8.5
4	3.5

In [38]:

```
target.head()
```

Out[38]:

0	21
1	47
2	27
3	75
4	30

Name: Scores, dtype: int64

In [41]:

```
pd.set_option('display.precision',1)  
sco.describe()
```

Out[41]:

	Hours	Scores
count	25.0	25.0
mean	5.0	51.5
std	2.5	25.3
min	1.1	17.0
25%	2.7	30.0
50%	4.8	47.0
75%	7.4	75.0
max	9.2	95.0

In [62]:

```
from sklearn.model_selection import train_test_split  
X_train,X_test,y_train,y_test=train_test_split(feature,target,test_size=0.3,random_state=
```

In [63]:

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

In [64]:

```
model_hr = LinearRegression() #creating an instantiate (or creating an object)
```

In [65]:

```
model_hr.fit(X_train,y_train)
```

Out[65]:

```
LinearRegression()
```

In [66]:

```
hr_score=model_hr.score(X_test,y_test)*100 # we multiply score by 100 to get percentage
```

In [67]:

```
print("Linear regression model's score --> ", int(hr_score),"%")
```

```
Linear regression model's score --> 95 %
```

In [71]:

```
y_pred = model_hr.predict(X_test)
```

In [72]:

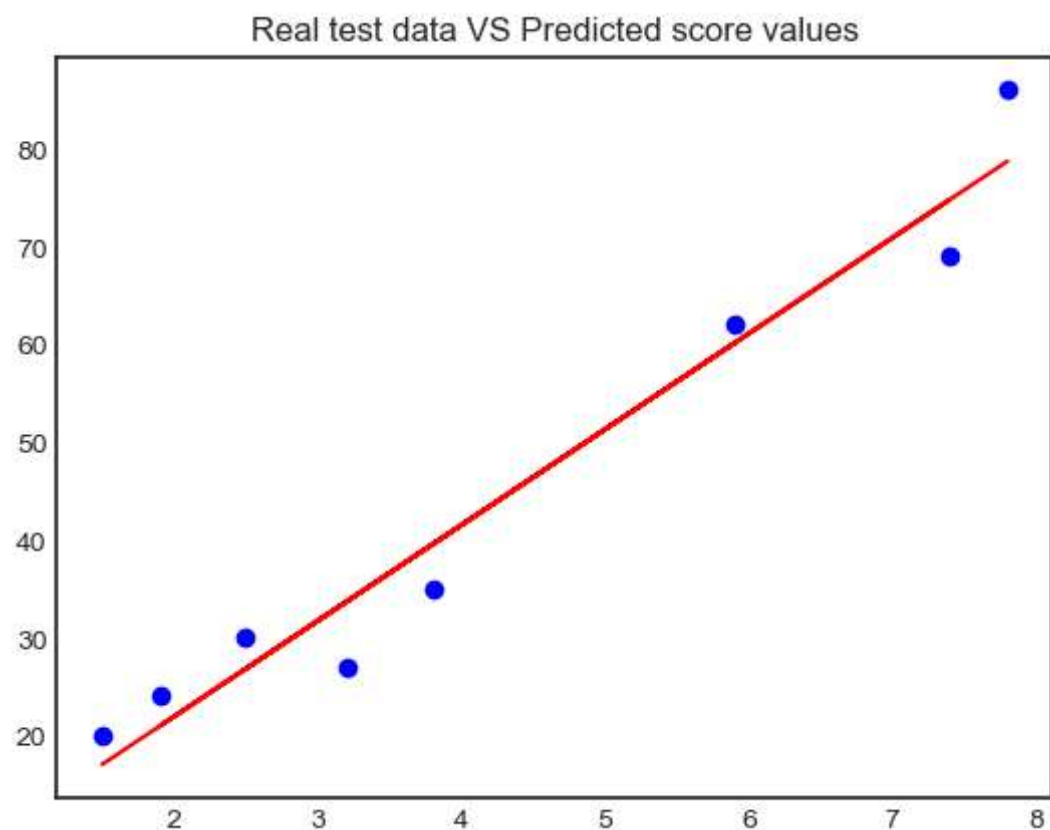
```
y_pred
```

Out[72]:

```
array([17.05366541, 33.69422878, 74.80620886, 26.8422321 , 60.12335883,  
       39.56736879, 20.96909209, 78.72163554])
```

In [75]:

```
plt.title("Real test data VS Predicted score values")
plt.scatter(X_test, y_test, color='blue', label="Real test data")
plt.plot(X_test, y_pred, color='red', label="Predicted score values")
plt.show()
```



In []: