Prediction using supervised machine learning
Task done by SHARAD PARMAR

In this task based upon the number of hours studied we will predict the percentage of marks that the student is expected to score.

In [34]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
from sklearn import linear_model
```

In [3]:
```
df=pd.read_csv("task1.csv")
```

In [16]:
```
df
```

Out[16]:

	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 [6]:
```
df.shape
```

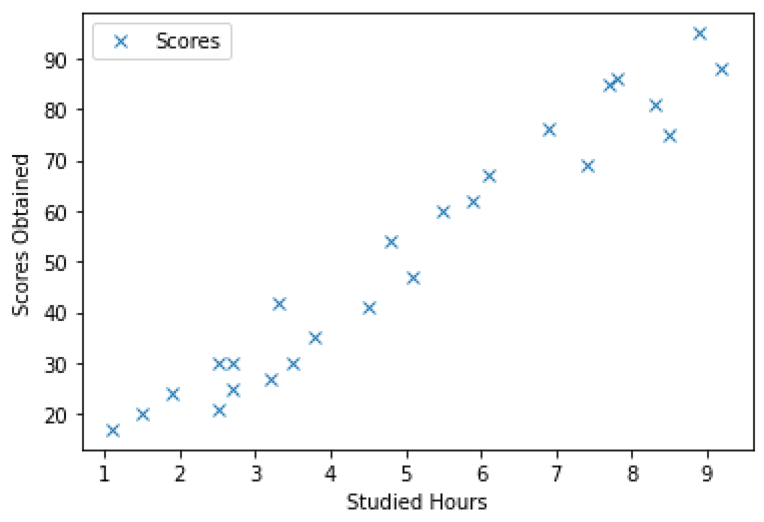
Out[6]: (25, 2)

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

Out[7]:

	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 [11]:
```
df.plot(x="Hours",y="Scores",style='x')
plt.xlabel("Studied Hours")
plt.ylabel("Scores Obtained")
plt.show()
```



In [12]:
```
#feature and target variables
X = df.Hours
y = df.Scores
```

In [13]:
```
#data splitting into test and train set
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)
```

In [14]:
```
X_train = np.array(X_train).reshape(-1,1)
X_test = np.array(X_test).reshape(-1,1)
y_train = np.array(y_train).reshape(-1,1)
y_test = np.array(y_test).reshape(-1,1)
```

In [21]:
```
#Training Model
from sklearn.linear_model import LinearRegression
reg = LinearRegression()
reg.fit(X_train, y_train)
```

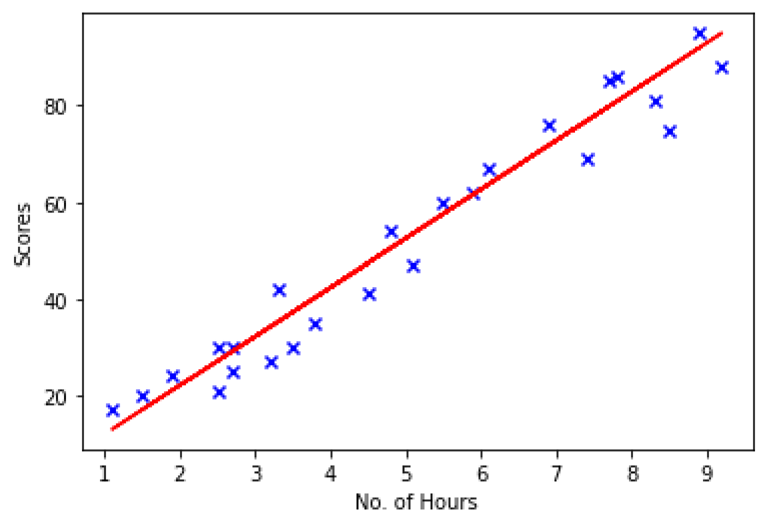
Out[21]: LinearRegression()

In [27]:
```
reg.score(X_test, y_test)
```

Out[27]: 0.9009408195020412

In [35]:
```
plt.scatter(data.Hours, data.Scores, marker = 'x', color = 'blue')
plt.plot(df.Hours, reg.predict(df[['Hours']]), color = 'red')  #plotting
the line of best fit
plt.xlabel('No. of Hours')
plt.ylabel('Scores')
```

Out[35]: Text(0, 0.5, 'Scores')



In [30]:
```
#evaluation of model

from sklearn import metrics
predictions = reg.predict(X_test)
print('Mean absolute error:', metrics.mean_absolute_error(y_test, predic
tions))
print('Mean squared error:', metrics.mean_squared_error(y_test, predicti
ons))
print('Root mean squared error:', np.sqrt(metrics.mean_squared_error(y_t
est, predictions)))
```

```
Mean absolute error: 6.956521568010078
Mean squared error: 59.21361573445986
Root mean squared error: 7.69503838420965
```

Making predictions

In [33]:
```
#making predictions
hrs_inp = float(input("Enter hours studied: "))
y_pred = reg.predict([[hrs_inp]])
s = str(y_pred)
print("Predicted Score: {}" .format(s[2:-2]))
```

```
Enter hours studied: 9.25
Predicted Score: 95.52803251
```