**THE SPARKS FOUNDATION - INTERNSHIP

*DATA SCIENCE AND BUSINESS ANALYTICS INTERN

**NAME : PRIYANSHU SHUKLA

TASK 1 : PREDICTION USING SUPERVISED MACHINE LEARNING

PROBLEM STATEMENT : The given dataset contains the score of students with respect to their study time. It is required to perform EDA and simple linear regression on the dataset to find out the score of a student who studies 9.5 hours/day

Data used : http://bit.ly/w-data.

In [42]:

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

In [43]:

```
data_load = pd.read_excel("Dataset2.xlsx")
```

In [44]:

```
print("Successfully imported data into console" )
```

```
Successfully imported data into console
```

In [45]:

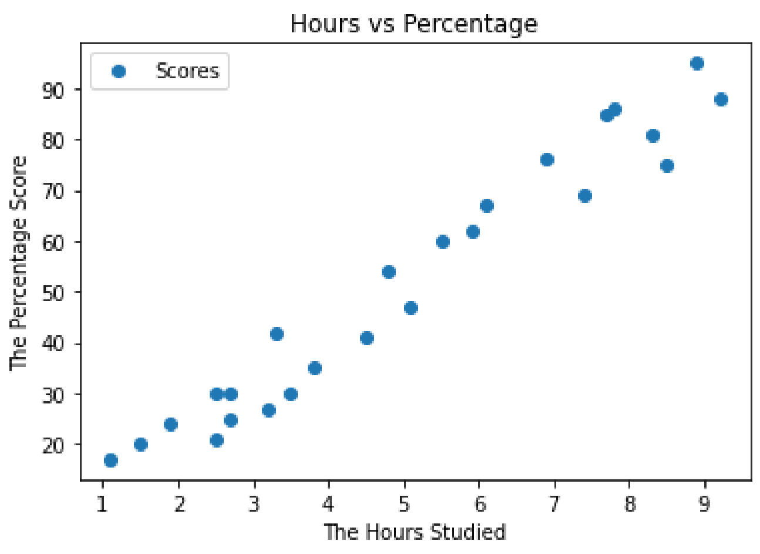
```
data_load.head(25)
```

Out[45]:

	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 [46]:

```
data_load.plot(x='Hours', y='Scores', style='o')
plt.title('Hours vs Percentage')
plt.xlabel('The Hours Studied')
plt.ylabel('The Percentage Score')
plt.show()
```



In [47]:

```
X = data_load.iloc[:, :-1].values
y = data_load.iloc[:, 1].values
```

In [48]:

```
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, random_state=0)
```

In [49]:

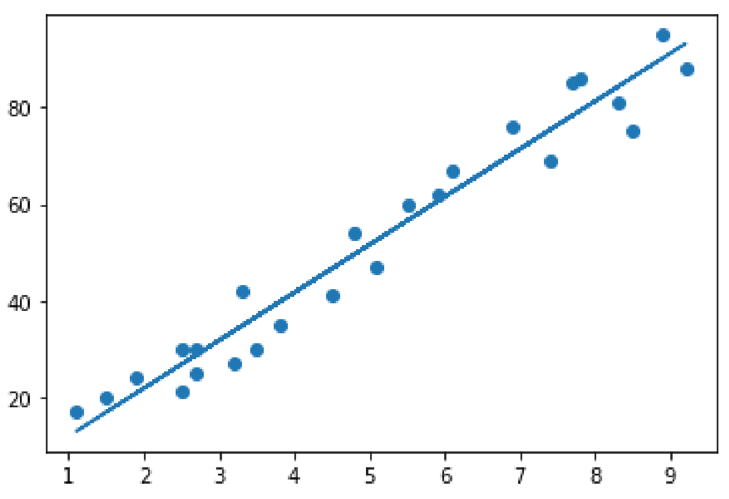
```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)

print("Training ... Completed !.")
```

```
Training ... Completed !.
```

In [50]:

```
line = regressor.coef_*X+regressor.intercept_
plt.scatter(X, y)
plt.plot(X, line);
plt.show()
```



In [51]:

```
print(X_test)
y_pred = regressor.predict(X_test)
```

```
[[1.5]
 [3.2]
 [7.4]
 [2.5]
 [5.9]]
```

In [52]:

```
df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
df
```

Out[52]:

	Actual	Predicted
0	20	16.884145
1	27	33.732261
2	69	75.357018
3	30	26.794801
4	62	60.491033

In [53]:

```
hours = [[9.25]]
own_pred = regressor.predict(hours)
print("Number of hours = {}".format(hours))
print("Prediction Score = {}".format(own_pred[0]))
```

```
Number of hours = [[9.25]]
Prediction Score = 93.69173248737538
```

In [54]:

```
#mean absolute error of the model
import sklearn as sk
from sklearn import metrics
print('Mean Absolute Error:', metrics.mean_absolute_error(y_test,y_pred))
```

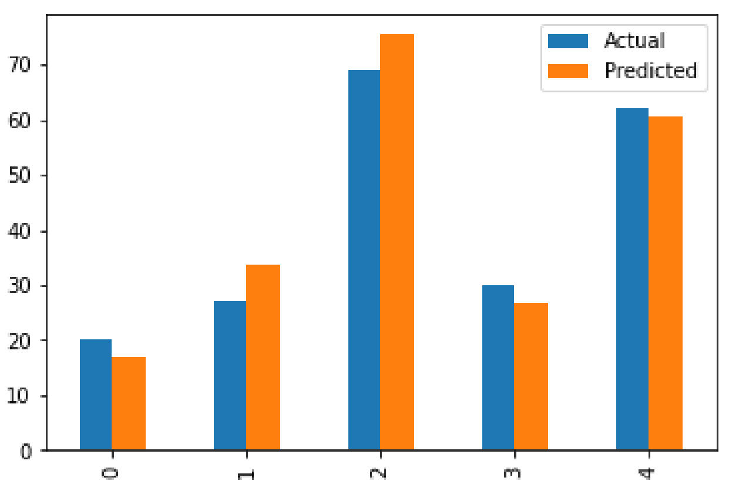
```
Mean Absolute Error: 4.183859899002975
```

In [55]:

```
df.plot(kind='bar')
```

Out[55]:

```
<AxesSubplot:>
```



In [ ]: