Task 1:Prediction using Supervised Machine Learning

Problem statement:

In this regression task we will predict the percentage of marks that a student is expected to score based upon the number of hours they studied. This is a simple linear regression task as it involves just two variables.

Author: Yadnesh Ingole

Data Science and Business Analytics internship at Sparks Foundations.

Importing the libraries

In [2]:

```
#Importing all the libraries
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
```

Importing the dataset

In [3]:

```
#Importing the data using pandas library
url = "http://bit.ly/w-data"
dataset= pd.read_csv(url)
print("Data imported successfully")

dataset.head(10)
```

```
Data imported successfully
```

Out[3]:

	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

In [61]:

```
#using describe method for showing statistics of the dataset
dataset.describe()
```

Out[61]:

	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 [62]:

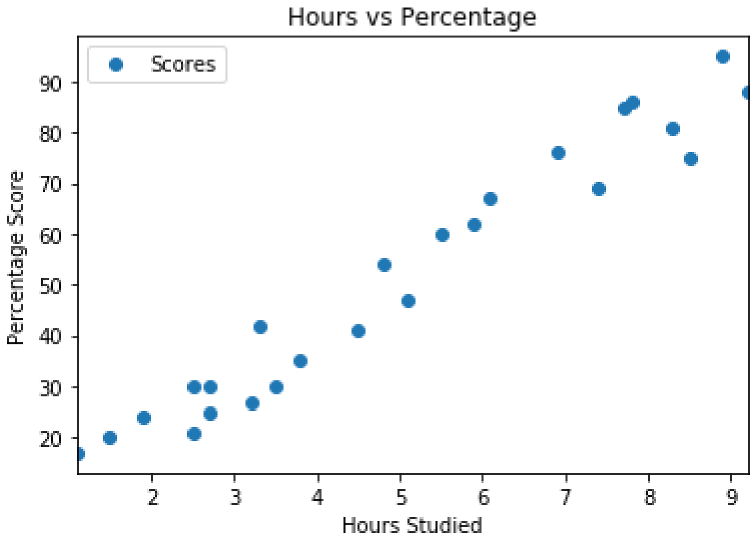
```
#To summarize the information on dataset
dataset.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25 entries, 0 to 24
Data columns (total 2 columns):
Hours     25 non-null float64
Scores    25 non-null int64
dtypes: float64(1), int64(1)
memory usage: 480.0 bytes
```

Plotting the scatter plot

In [60]:

```
#Plotting the dataset on 2-D graph
dataset.plot(x='Hours', y='Scores', style='o')
plt.title('Hours vs Percentage')
plt.xlabel('Hours Studied')
plt.ylabel('Percentage Score')
plt.show()
```



Preprocessing the dataset

In [19]:

```
#Dividing the dataset into indepedent variable(input)X and dependent variable Output(Y)

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

In [21]:

```
print(X)
```

```
[[2.5]
 [5.1]
 [3.2]
 [8.5]
 [3.5]
 [1.5]
 [9.2]
 [5.5]
 [8.3]
 [2.7]
 [7.7]
 [5.9]
 [4.5]
 [3.3]
 [1.1]
 [8.9]
 [2.5]
 [1.9]
 [6.1]
 [7.4]
 [2.7]
 [4.8]
 [3.8]
 [6.9]
 [7.8]]
```

In [22]:

```
print(y)
```

```
[21 47 27 75 30 20 88 60 81 25 85 62 41 42 17 95 30 24 67 69 30 54 35 76
 86]
```

Splitting the dataset into training set and testing set

In [24]:

```
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)
```

Training the Simple Linear Regression model on training set

In [26]:

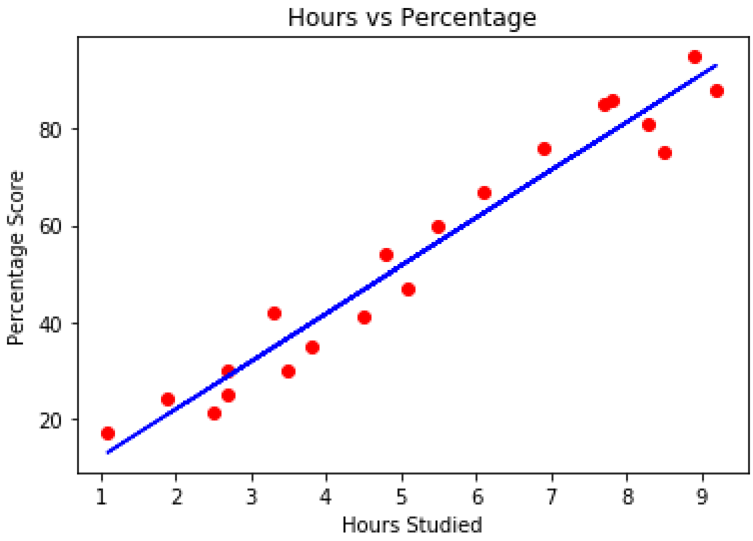
```
from sklearn.linear_model import LinearRegression
regressor=LinearRegression()
regressor.fit(X_train,y_train)
print('Training is complete')
```

```
Training is complete
```

Visualize the training set result

In [35]:

```
plt.scatter(X_train,y_train,color='Red')
plt.plot(X_train,regressor.predict(X_train),color='blue')
plt.title('Hours vs Percentage')
plt.xlabel('Hours Studied')
plt.ylabel('Percentage Score')
plt.show()
```



Predicting the test set result

In [37]:

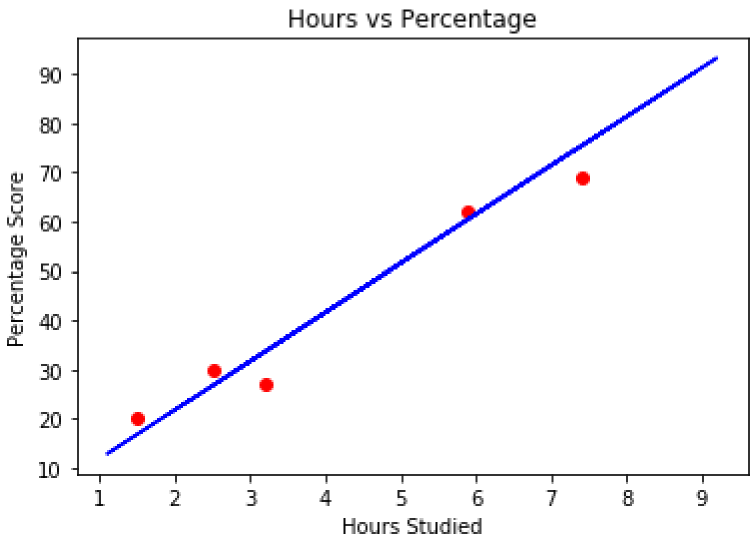
```
print(X_test)
y_pred=regressor.predict(X_test)
print(y_pred)
```

```
[[1.5]
 [3.2]
 [7.4]
 [2.5]
 [5.9]]
[16.88414476 33.73226078 75.357018   26.79480124 60.49103328]
```

Visulizing the test set result

In [41]:

```
plt.scatter(X_test,y_test,color='Red')
plt.plot(X_train,regressor.predict(X_train),color='blue')
plt.title('Hours vs Percentage')
plt.xlabel('Hours Studied')
plt.ylabel('Percentage Score')
plt.show()
```



Comparing the actual vs predicted result

In [42]:

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

Out[42]:

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

Evaluating model performance using R2_score

In [49]:

```
from sklearn.metrics import r2_score
r2_score(y_test,y_pred)
```

Out[49]:

```
0.9454906892105356
```

Predicting the Score percentage if student studies for 9.25hrs/day

In [59]:

```
# You can also test with your own data
hours = [9.25]
result = regressor.predict([hours])
print("Predicted Score for 9.25 hours/day = {}".format(round(result[0],2)))
```

```
Predicted Score for 9.25 hours/day = 93.69
```