1 Name- Ranjeeta Kumari

2 The Sparks Foundation- Data Analysis and Business Analytics

3 Task 1 : Prediction using Supervised ML

4 To Predict the percentage of an student based on the number of study hours they studied.

[1]:

```
# Importing all libraries required in this notebook
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

5 Reading and importing the data

[2]:

```
url = "http://bit.ly/w-data"
data = pd.read_csv(url)
print("Data imported successfully")

data.head(10)
```

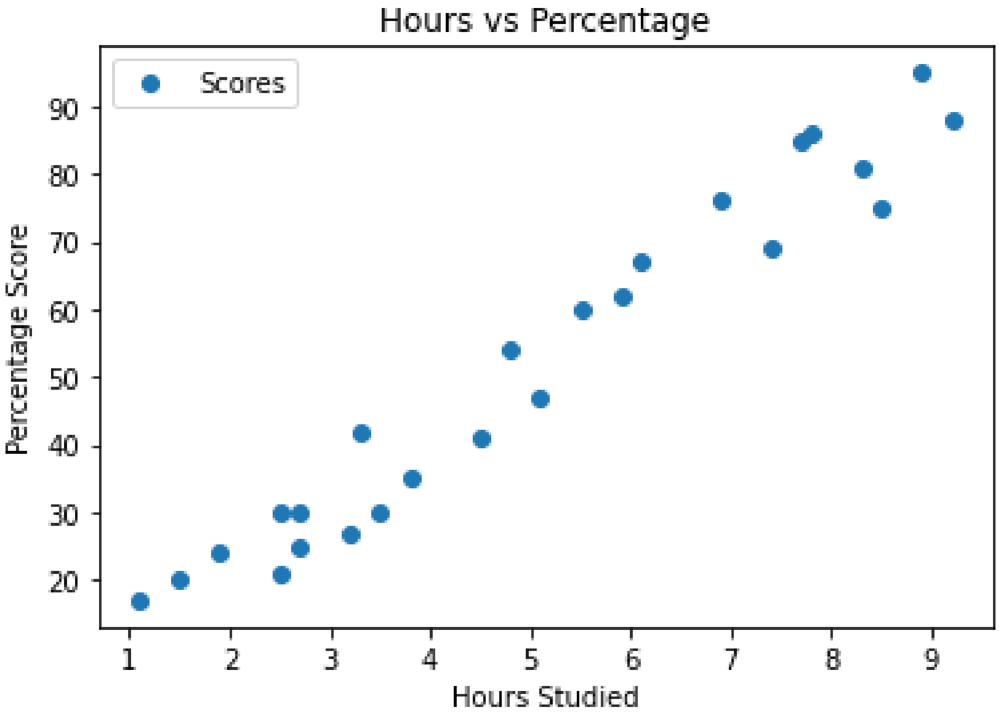
```
Data imported successfully
```

[2]:

```
   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
```

6 Plotting graph for analyzing the data

[3]:
```
# Plotting the graphs for analyzing the data set
data.plot(x='Hours', y='Scores', style='o')
plt.title('Hours vs Percentage')
plt.xlabel('Hours Studied')
plt.ylabel('Percentage Score')
plt.show()
```



7 Graph depicts that as number of study hours increases percentage of score of students increases.

8 Data preparation for model

[4]:
```
X = data.iloc[:, :-1].values
y = data.iloc[:, 1].values
```

9 Dividing data for training and testing of the model

[5]:
```
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)
```

10 Training the algorithm

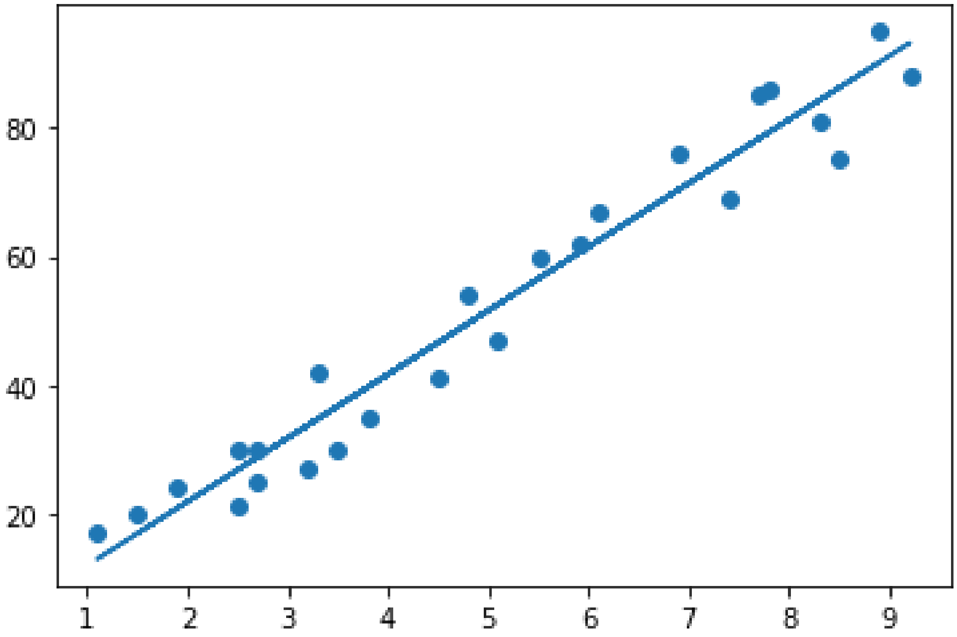
[6]:
```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)

print("Training complete.")
```

```
Training complete.
```

[7]:
```
# Plotting the regression line
line = regressor.coef_*X+regressor.intercept_

# Plotting for the test data
plt.scatter(X, y)
plt.plot(X, line);
plt.show()
```



11 Testing the model

[8]:
```
print(X_test)
print("Prediction of Score")
y_pred = regressor.predict(X_test)
print(y_pred)
```

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

12 Checking the accuracy of the model

[9]:
```
df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
df
```

[9]:
```
   Actual  Predicted
0      20  16.884145
1      27  33.732261
2      69  75.357018
3      30  26.794801
4      62  60.491033
```

13 Score of a student when he studies 9.5 hrs/day

[10]:
```
hours = [[9.25]]
pred = regressor.predict(hours)
print(pred)
```

```
[93.69173249]
```

14 Evaluating the model

[11]:
```
from sklearn import metrics
print('Mean Absolute Error:',
      metrics.mean_absolute_error(y_test, y_pred))
```

```
Mean Absolute Error: 4.183859899002982
```