The Sparks Foundation - Data Science & Business Analytics Internship

TASK 1 - Prediction using Supervised Machine Learning

In [ ]:

```
In this task it is required to predict the percentage of a student on the basis of number of hours studie

Steps:
Step 1 - Importing the dataset
Step 2 - Visualizing the dataset
Step 3 - Data preparation
Step 4 - Training the algorithm
Step 5 - Visualizing the model
Step 6 - Making predcitions
Step 7 - Evaluating the model
```

Author : BHAMA SRI D

STEP 1 : IMPORTING THE DATA SET

In this step, we will import the dataset through the link with the help of pandas library and then we will observe the data

In [12]:

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

In [13]:

```
# Reading data from remote link
url = "https://raw.githubusercontent.com/AdiPersonalWorks/Random/master/student_scores%20-%20student_scor
s_data = pd.read_csv(url)
print("Data imported successfully")

s_data.head(10)
```

```
Data imported successfully
```

Out[13]:

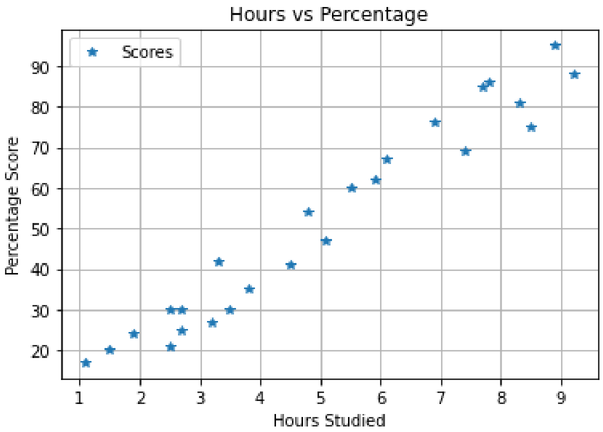
	Hours	Scores
0	2.5	21
1	5.1	47
2	3.2	27
3	8.5	75
4	3.5	30
5	1.5	20
6	9.2	88
7	5.5	60
8	8.3	81
9	2.7	25

STEP 2 : VISUALIZING THE DATASET

In this we will plot the dataset to check whether we can observe any relation between the two variables or not

In [27]:

```
# Plotting the distribution of scores
s_data.plot(x='Hours', y='Scores', style='*')
plt.title('Hours vs Percentage')
plt.xlabel('Hours Studied')
plt.ylabel('Percentage Score')
plt.grid()
plt.show()
```



STEP 3 : DATA PREPARATION

In [15]:

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

In [16]:

```
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)
```

STEP 4 : TRAINING THE ALGORITHM

In [ ]:

```
We have splited our data into training and testing sets, and now we will train our Model.
```

In [17]:

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

print("Training complete.")
```

```
Training complete.
```

STEP 5 : VISUALIZING THE MODEL

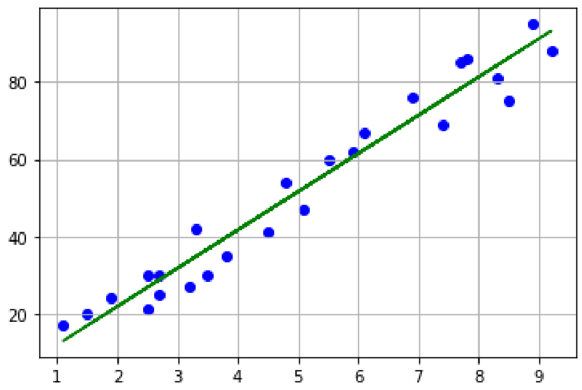
In [ ]:

```
After training the model, now its time to visualize it.
```

In [41]:

```
# Plotting the regression line
line = regressor.coef_*X+regressor.intercept_

# Plotting for the test data
plt.scatter(X,y,color='blue')
plt.plot(X, line,color='green');
plt.grid()
plt.show()
```



STEP 6 : MAKING PREDICTION

In [ ]:

```
Now that we have trained our algorithm, it's time to make some predictions.
```

In [19]:

```
print(X_test) # Testing data - In Hours
y_pred = regressor.predict(X_test) # Predicting the scores
```

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

In [39]:

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

Out[39]:

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

In [21]:

```
# You can also test with your own data
hours = 9.25
own_pred = regressor.predict([[hours]])
print("No of Hours = {}".format(hours))
print("Predicted Score = {}".format(own_pred[0]))
```

```
No of Hours = 9.25
Predicted Score = 93.69173248737539
```

STEP 7 : EVALUATING THE MODEL

In [ ]:

```
In the last step, we are going to evaluate our trained model by calculating mean absolute error
```

In [22]:

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

```
Mean Absolute Error: 4.183859899002982
```

In [ ]: