PEKHAM SEAL

Task 1: In this regression task we will predict the percentage of marks that a student is expected to score on the basis of the number of hours they studied. This is a two variable simple linear regression task.

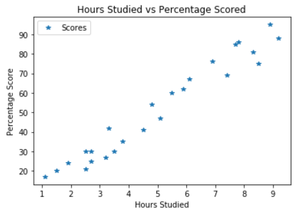
In [1]:
```
# Importing all libraries required in this notebook
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

In [5]:
```
# Reading data from remote link
url = "https://raw.githubusercontent.com/AdiPersonalWorks/Random/master/student_scores%20-%20student_scores.csv"
data = pd.read_csv(url)
#to see if the data is successfully imported
data
```

Out[5]:

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



From the graph above, we see that there is a positive linear relation between the number of hours studied and percentage of score secured.

Dividing the data into attributes and labels

In [7]:
```
X = data.iloc[:, :-1].values
y = data.iloc[:, 1].values
```

In [8]:
```
#splitting the data into training and test datasets using Scikit-Learn
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)
```

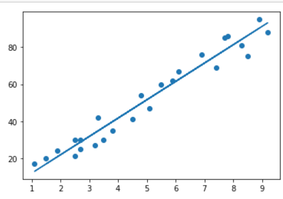
In [10]:
```
#training the Algorithm
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)

print("Training Done")
```

```
Training Done
```

In [11]:
```
# Plotting the regression line
line = regressor.coef_*X+regressor.intercept_
```

In [12]:
```
# Plotting for the test data
plt.scatter(X,y)
plt.plot(X, line);
plt.show()
```



Making Predictions

In [13]:
```
# Testing data
print(X_test)
# Predicting the scores
y_pred = regressor.predict(X_test)
```

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

In [14]:
```
# Comparison of Actual vs Predicted
df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
df
```

Out[14]:

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

In [15]:
```
#testing with own data
hours = 8.77
own_pred = regressor.predict([[hours]])
print("No of Hours = {}".format(hours))
print("Predicted Score = {}".format(own_pred[0]))
```

```
No of Hours = 8.77
Predicted Score = 88.93461737666709
```

In [16]:
```
from sklearn import metrics
print('Mean Absolute Error:',
      metrics.mean_absolute_error(y_test, y_pred))
```

```
Mean Absolute Error: 4.183859899002975
```