Author : Pooja Gandhi

Task1:Prediction using Supervised ML

Simple Linear Regression using Python Scikit Learn

Step 1:Importing the dataset

In [1]:

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

In [2]:

```
data_load = pd.read_csv("student_scores.csv")
print("Successfully imported data into console" )
```

```
Successfully imported data into console
```

In [3]:

```
data_load.head(6)
```

Out[3]:

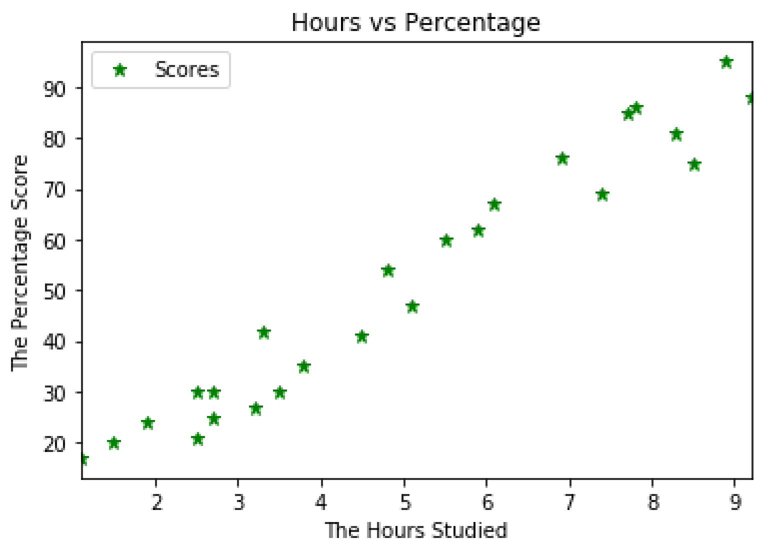
	Hours	Scores
0	2.5	21
1	5.1	47
2	3.2	27
3	8.5	75
4	3.5	30
5	1.5	20

step2:Visualizing the dataset

we will plot the datasetand check if there is any relation between the variables

In [4]:

```
data_load.plot(x='Hours', y='Scores', style='*', color='green', markersize=7)
plt.title('Hours vs Percentage')
plt.xlabel('The Hours Studied')
plt.ylabel('The Percentage Score')
plt.show()
```



Step3: Data preparation

In this step we will divide the data into inputs and outputs.And then we will divide the whole dataset into 2parts-testing data and training data

In [5]:

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

In [6]:

```
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)
```

Step4:Training the Algorithm

we have split our data into training and testing sets ,and now is finally the time to train our algorithm.

In [8]:

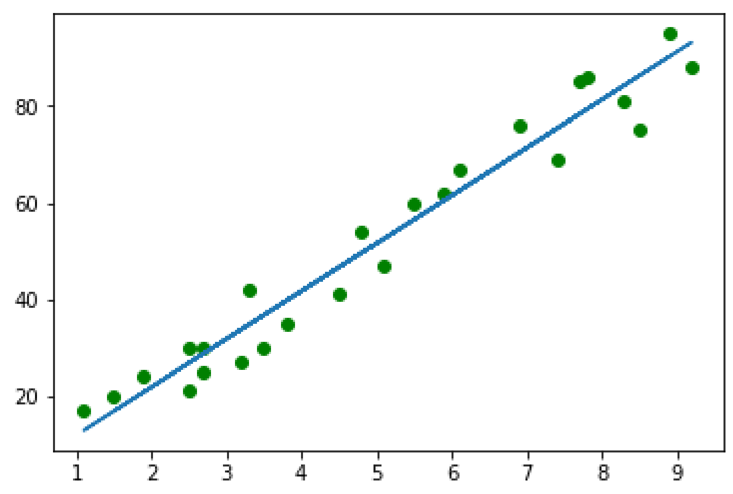
```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)

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

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

Step5:Visualizing the model

In [9]:

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



Step6:Making Predictions

After training the algorithm,it's time to make predictions

In [10]:

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

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

In [11]:

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

Out[11]:

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

In [12]:

```
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 = 93.6917324874
```

If a student studies for 9.25 hours then the predicted score is 93.69173248

Step4:Evaluating the model

Final step, we are going to evaluate our trained model by calculating mean absolute error.

In [14]:

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

```
('Mean Absolute Error:', 4.183859899002975)
```

Thankyou!

In [ ]:

```

```