

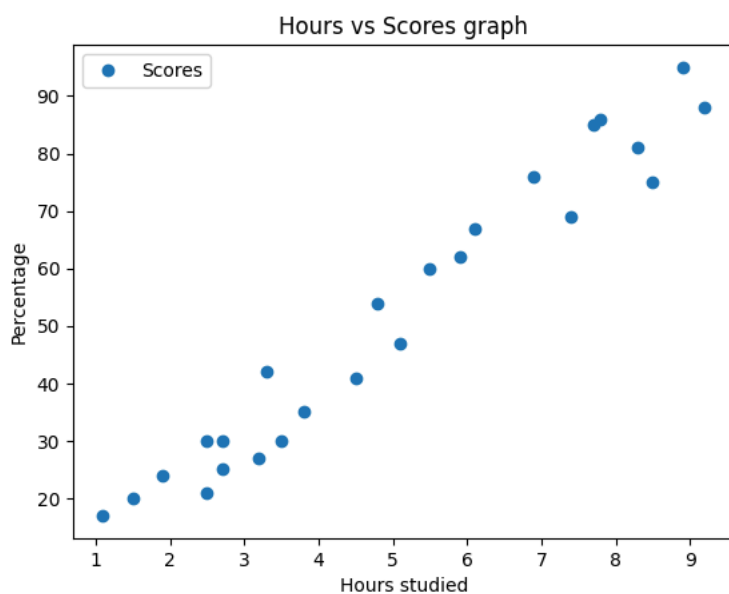
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

```
#importing the data
import pandas as pd
data="http://bit.ly/w-data"
im_data=pd.read_csv(data)
print("Data imported")
im_data.head()
```

Data imported

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

```
#plotting the points of scores vs hours on a graph
import matplotlib.pyplot as plt
im_data.plot(x="Hours",y="Scores",style="o")
plt.title("Hours vs Scores graph")
plt.xlabel("Hours studied")
plt.ylabel("Percentage")
plt.show()
```



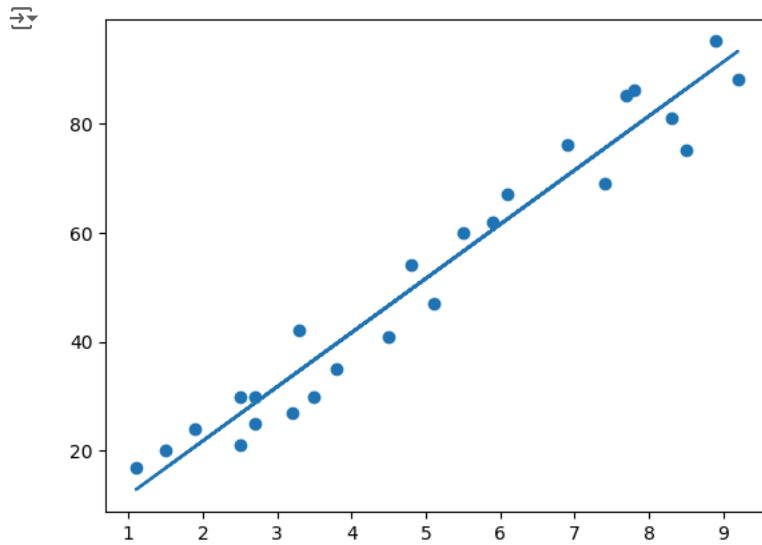
```
x=im_data.iloc[:, :-1].values
y=im_data.iloc[:, 1].values
```

```
#splitting the dta into train and test sets using train_test_split() method
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 model
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(x_train,y_train)
print("Training has been completed.")
```

Training has been completed.

```
#Regression line
line= lm.coef_*x + lm.intercept_
#plotting test data and regression line
plt.scatter(x,y)
plt.plot(x,line)
plt.show()
```



```
#predicting using the trained algorithm
print(x_test)
Y=lm.predict(x_test)
```

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

```
#comparing the actual values to our predicted values
df=pd.DataFrame({"Actual value":y_test,"Predicted value":Y})
df
```

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

```
#Testing what would be the predicted score
hours=6
pred = lm.predict([[hours]])
print ("Number of hours:{}".format(hours))
print ("Predicted Score:{}".format(pred[0]))
```

```
Number of hours:6
Predicted Score:61.482098925288106
```

```
#calculating the Mean absolute error
from sklearn import metrics
print("Mean Absolute Error:", metrics.mean_absolute_error(y_test,Y))
```

```
Mean Absolute Error: 4.183859899002982
```

