

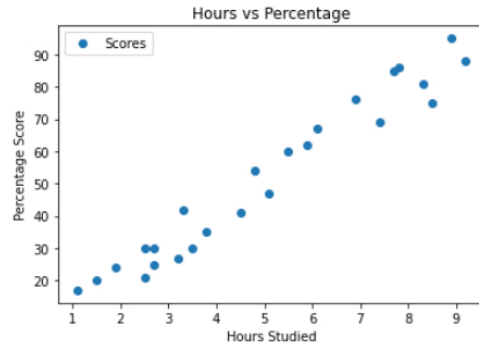
```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn import metrics
```

```
In [2]: dataset = pd.read_csv('student-scores.csv')
dataset.shape
dataset.head()
dataset.describe()
```

```
Out[2]:
```

	Hours	Scores
count	25.000000	25.000000
mean	5.012000	51.480000
std	2.525094	25.286887
min	1.100000	17.000000
25%	2.700000	30.000000
50%	4.800000	47.000000
75%	7.400000	75.000000
max	9.200000	95.000000

```
In [3]: dataset.plot(x='Hours', y='Scores', style='o')
plt.title('Hours vs Percentage')
plt.xlabel('Hours Studied')
plt.ylabel('Percentage Score')
plt.show()
```



```
In [4]: X = dataset.iloc[:, :-1].values
Y = dataset.iloc[:, 1].values
```

```
In [5]: trainX, testX, trainY, testY = train_test_split(X, Y, test_size=0.2, random_state=0)
```

```
In [6]: regressor = LinearRegression()
regressor.fit(trainX, trainY)
```

```
Out[6]:
```

LinearRegression
LinearRegression()

```
In [7]: print(regressor.intercept_)

2.0181600414346974
```

```
In [8]: print(regressor.coef_)

[9.91065648]
```

```
In [9]: predY = regressor.predict(testX)
df = pd.DataFrame({'Actual': testY, 'Predicted': predY})
df
```

```
Out[9]:
```

Actual	Predicted
--------	-----------

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

```
In [10]: print('Mean Absolute Error:', metrics.mean_absolute_error(testY, predY))
print('Mean Squared Error:', metrics.mean_squared_error(testY, predY))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(testY, predY)))
```

```
Mean Absolute Error: 4.183859899002975
Mean Squared Error: 21.598769307217406
Root Mean Squared Error: 4.647447612100367
```

In []: