In [1]:

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

In [12]:

```
#Importing Dataset
url = "http://bit.ly/w-data"
data = pd.read_csv(url)
```

In [4]:

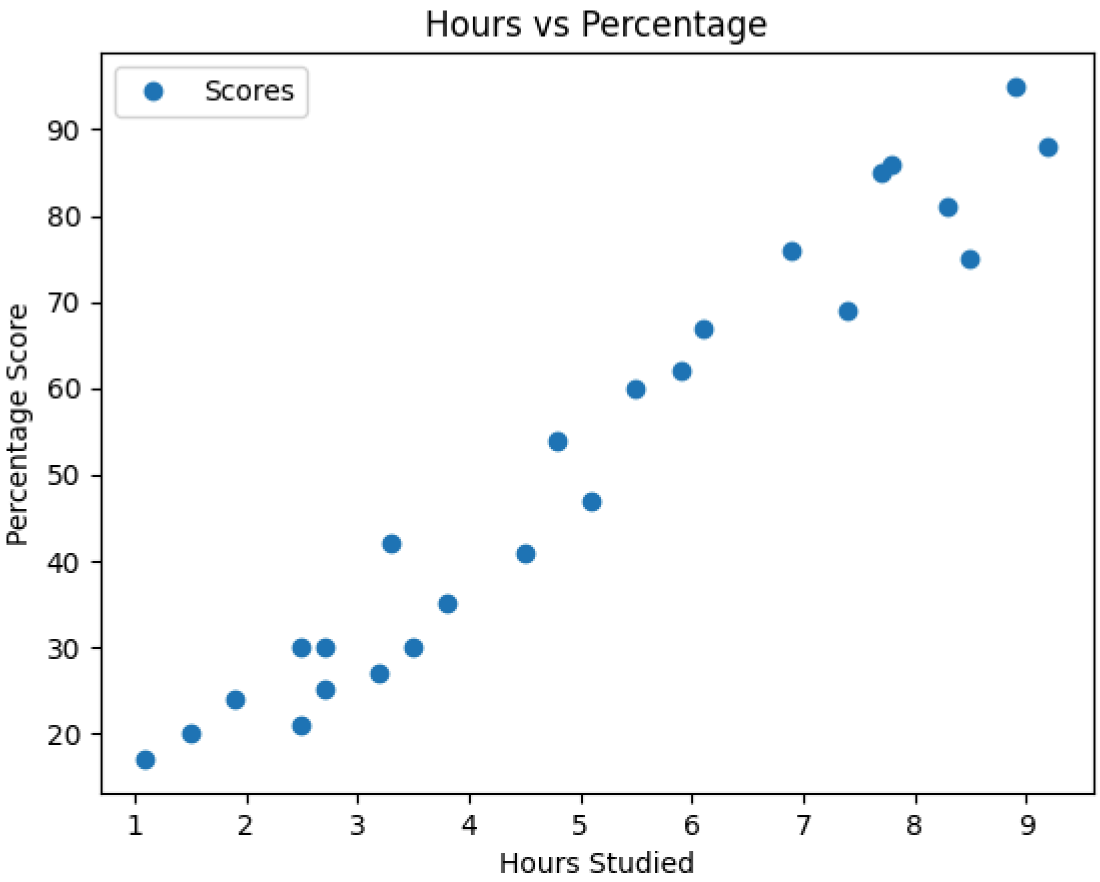
```
data.head()
```

Out[4]:

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

In [5]:

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



In [13]:

```
#Setting Attributes and Labels
X = data.iloc[:, :-1].values
y = data.iloc[:, 1].values
```

In [14]:

```
#Splitting the dataset to train and test sets
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 [15]:

```
#Training the Model
from sklearn.linear_model import LinearRegression
```

```
model = LinearRegression()
model.fit(X_train, y_train)
```

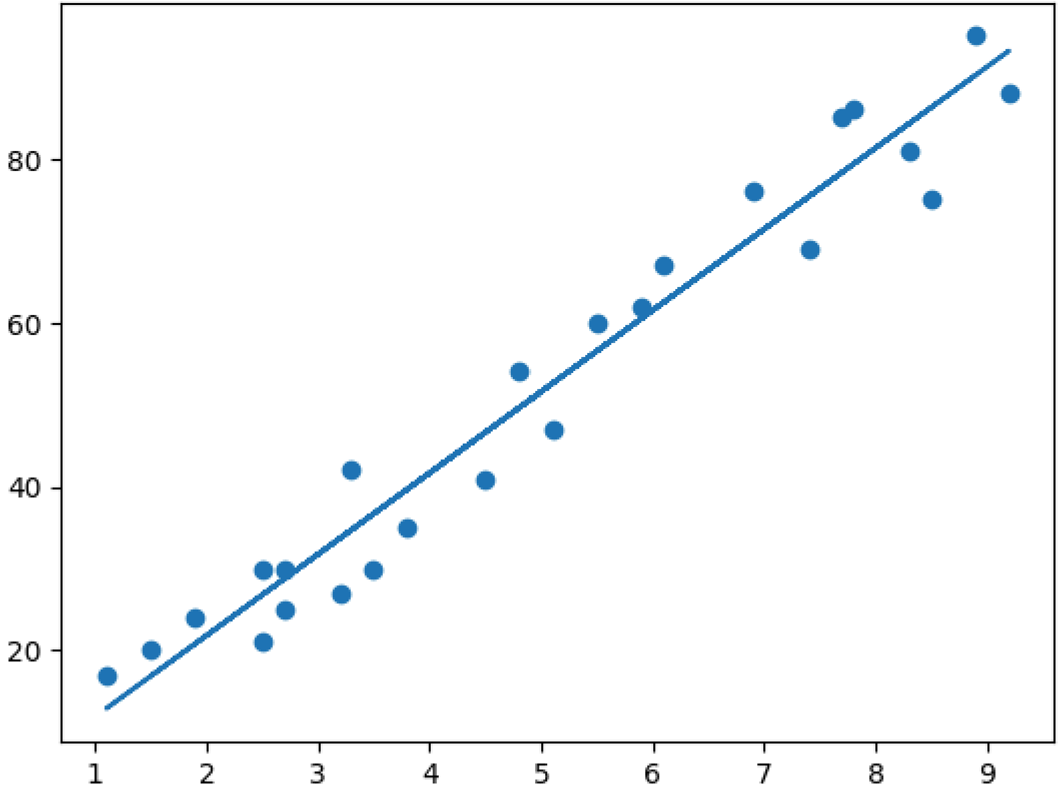
Out[15]:

```
▾ LinearRegression
LinearRegression()
```

In [16]:

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

# Plotting for the test data
plt.scatter(X, y)
plt.plot(X, line);
plt.show()
```



In [21]:

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

```
Mean Absolute Error: 4.183859899002975
```

In [19]:

```
#Making Prediction
y_pred = model.predict(X_test) # Predicting the scores
```

In [20]:

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

Out[20]:

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

In [25]:

```
#Predicted score if a student studies for 9.25 hrs/day
duration = [[9.25]]
prediction = model.predict(duration)
print("No of Hours = {}".format(duration))
print("Predicted Score = {}".format(prediction[0]))
```

```
No of Hours = [[9.25]]
Predicted Score = 93.69173248737535
```