In [1]:

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

In [2]:

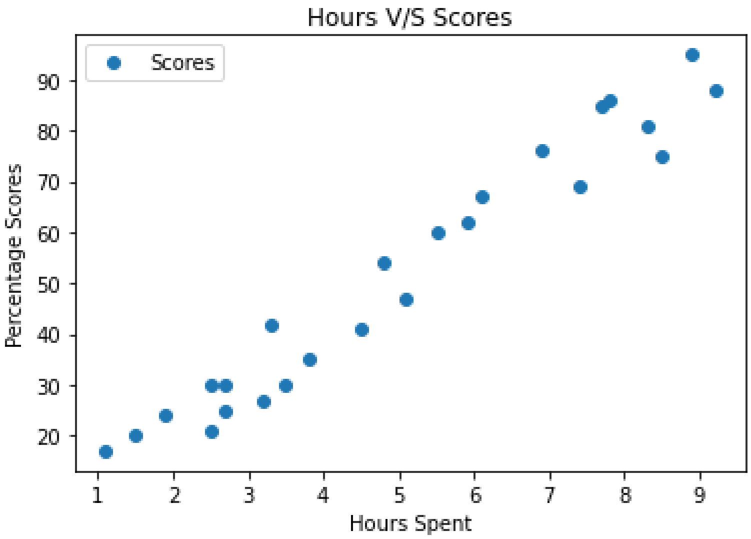
```
data_url = "http://bit.ly/w-data"
data = pd.read_csv(data_url)
```

In [ ]:

```
# data.head()
data.info()
data.describe()
```

In [10]:

```
data.plot(x = 'Hours', y = 'Scores', style = 'o')
plt.title('Hours V/S Scores')
plt.xlabel('Hours Spent')
plt.ylabel('Percentage Scores')
plt.show()
```



In [6]:

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

In [7]:

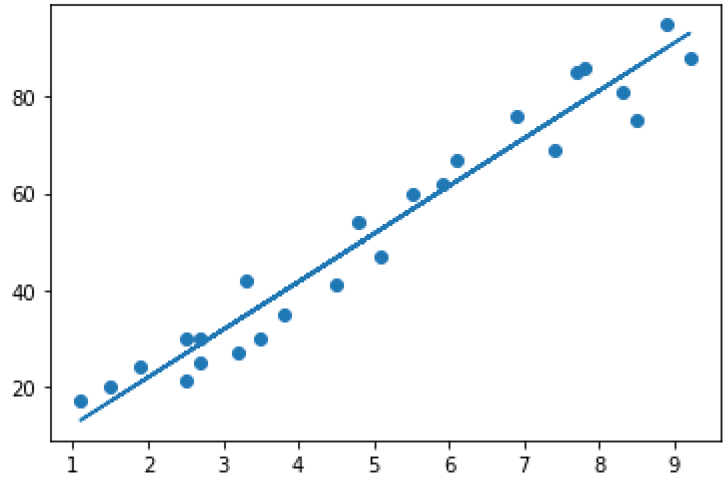
```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size = 0.2, random_state
from sklearn.linear_model import LinearRegression
lr_model = LinearRegression()
lr_model.fit(X_train, y_train)

print('Model Trained Successfully')
lr_model.coef_, lr_model.intercept_
#Plotting line
line = lr_model.coef_*X + lr_model.intercept_
```

```
Model Trained Successfully
```

In [8]:

```
plt.scatter(X, y)
plt.plot(X, line)
plt.show()
predictions = lr_model.predict(X_test)
print('Values Predicted')
```



```
Values Predicted
```

In [9]:

```
data_frame = pd.DataFrame({'Actual Value': y_test, 'Predicted Value': predictions})
data_frame
arr = np.array([9.25])
hour_studied = arr.reshape(-1,1)
pred_score = lr_model.predict(hour_studied)
print('If a student studied 9.25 hrs/day the score would be :',pred_score[0])
from sklearn.metrics import mean_absolute_error
print('Mean Absolute Error for our trained algorithm is: ',mean_absolute_error(y_test,
```

```
If a student studied 9.25 hrs/day the score would be : 93.69173248737538
Mean Absolute Error for our trained algorithm is:  4.183859899002975
```

In [ ]: