GRIP: THE SPARKS FOUNDATION

DATA Science and business Analytics intership

Author: Sourabh Mishra

In [35]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

In [36]:
```
url="http://bit.ly/w-data"
data=pd.read_csv(url)
```

In [37]:
```
data.head()   # to view the first 5 rows of the data
```

Out[37]:

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

In [38]:
```
data.info()   # to see if there are any missing values
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25 entries, 0 to 24
Data columns (total 2 columns):
 #   Column  Non-Null Count  Dtype  
---  ------  --------------  -----  
 0   Hours   25 non-null     float64
 1   Scores  25 non-null     int64  
dtypes: float64(1), int64(1)
memory usage: 528.0 bytes
```

In [39]:
```
data.describe()  # to see the summary statistics of the data
```

Sourabh893/data-Science-: SPARKS F | Home | Downloads | predicting_student_scores - Jupyt

localhost:8888/notebooks/predicting_student_scores.ipynb

jupyter predicting_student_scores Last Checkpoint: 21 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

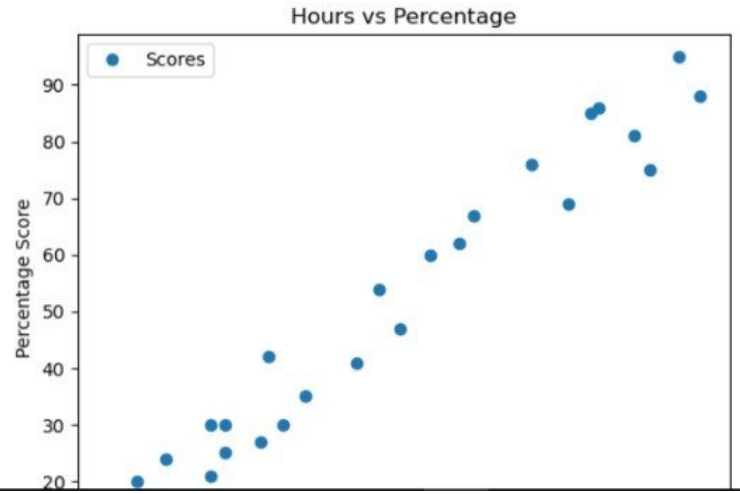
Run Code

In [39]:
```
data.describe()  # to see the summary statistics of the data
```

Out[39]:

	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 [40]:
```
data.plot(x='Hours', y='Scores', style='o')
plt.title('Hours vs Percentage')
plt.xlabel('Hours Studied')
plt.ylabel('Percentage Score')
plt.show()
```



Sourabh893/data-Science-: SPARKS F | Home | Downloads | predicting_student_scores - Jupyt

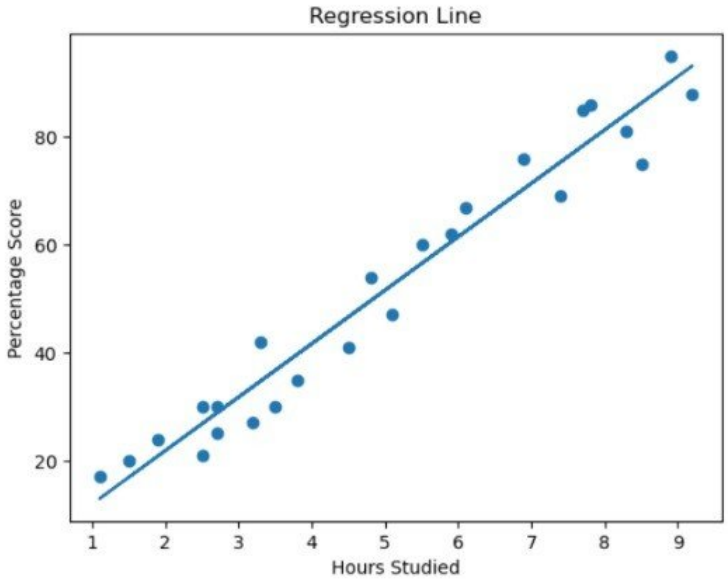
localhost:8888/notebooks/predicting_student_scores.ipynb

jupyter predicting_student_scores Last Checkpoint: 21 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

Run Code

In [45]:
```
line = regressor.coef_*X+regressor.intercept_
plt.scatter(X, y)
plt.plot(X, line)
plt.title('Regression Line')
plt.xlabel('Hours Studied')
plt.ylabel('Percentage Score')
plt.show()
```



In [46]:
```
hours = [[9.25]]
predicted_score = regressor.predict(hours)
print("Number of study hours: {}".format(hours))
print("Predicted Score = {}".format(predicted_score[0]))
```

```
Number of study hours: [[9.25]]
Predicted Score = 93.69173248737538
```

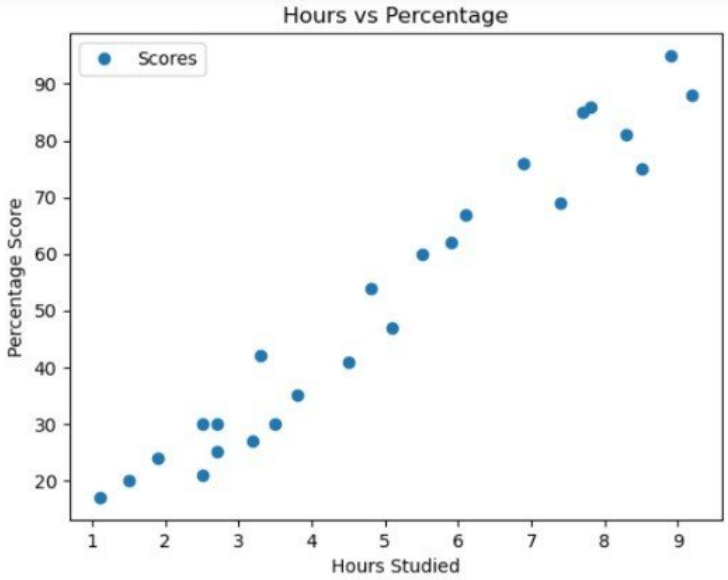
Sourabh893/data-Science-: SPARKS F | Home | Downloads | predicting_student_scores - Jupyt

localhost:8888/notebooks/predicting_student_scores.ipynb

jupyter predicting_student_scores Last Checkpoint: 21 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

Run Code



In [41]:
```
X = data.iloc[:, :-1].values
y = data.iloc[:, 1].values
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 [42]:
```
from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

Out[42]: LinearRegression()

In [43]:
```
y_pred = regressor.predict(X_test)
```

In [44]:
```
df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
```

Sourabh893/data-Science-: SPARKS F | Home | Downloads | predicting_student_scores - Jupyt

localhost:8888/notebooks/predicting_student_scores.ipynb

jupyter predicting_student_scores Last Checkpoint: 21 hours ago (autosaved) Logout

File Edit View Insert Cell Kernel Widgets Help Not Trusted Python 3 (ipykernel)

Run Code

In [43]:
```
y_pred = regressor.predict(X_test)
```

In [44]:
```
df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
df
```

Out[44]:

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

In [45]:
```
line = regressor.coef_*X+regressor.intercept_
plt.scatter(X, y)
plt.plot(X, line)
plt.title('Regression Line')
plt.xlabel('Hours Studied')
plt.ylabel('Percentage Score')
plt.show()
```