This notebook demonstrates a linear regression model that predicts student scores based on the number of hours they study. The dataset contains two columns: `Hours` (number of hours studied) and `Scores` (scores obtained by students). The goal is to build a simple linear regression model to understand the relationship between study hours and scores, and to predict scores for given study hours.

In [30]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings('ignore')
```

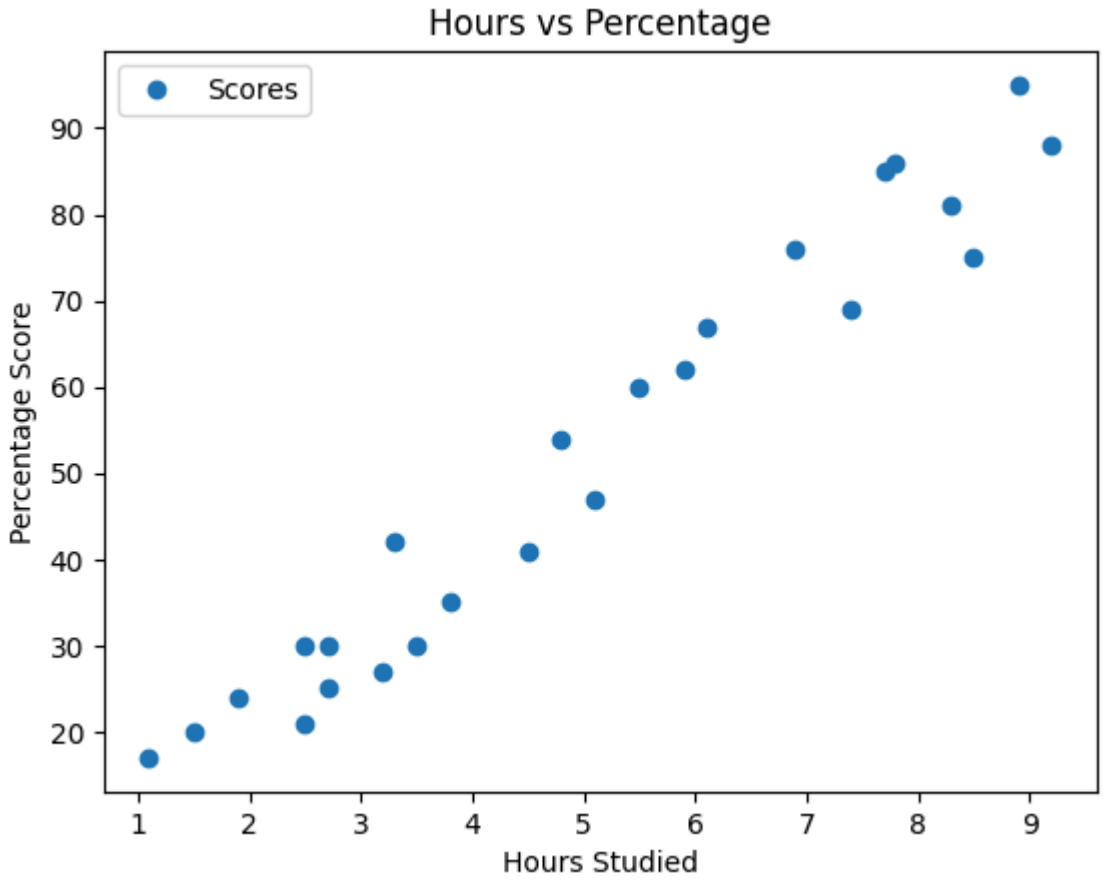
In [31]:
```
#Importing Dataset
url = "http://bit.ly/w-data"
data = pd.read_csv(url)
```

In [32]:
```
data.head()
```

Out[32]:

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

In [33]:
```
# 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()
```



The scatter plot visualization reveals a clear direct proportionality between study hours and scores

In [34]:
```
#Setting Attributes and Labels
X = data.iloc[:, :-1].values
y = data.iloc[:, 1].values
```

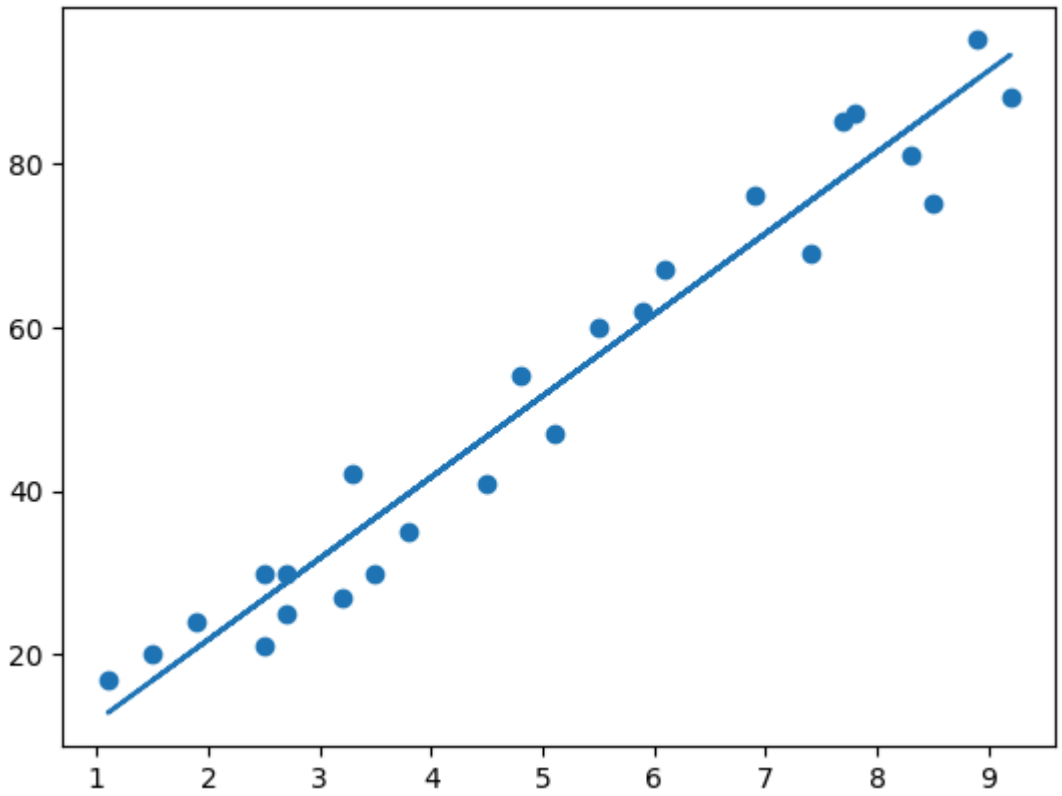
In [35]:
```
#Splitting the dataset to train and test sets
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(X, y,
                            test_size=0.2, random_state=0)
```

In [36]:
```
#Training the Model
from sklearn.linear_model import LinearRegression
model = LinearRegression()
model.fit(X_train, y_train)
```

Out[36]:
```
LinearRegression
LinearRegression()
```

In [37]:
```
# 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 [38]:
```
from sklearn.metrics import mean_absolute_error, mean_squared_error, r2_score
mae = mean_absolute_error(y_test, y_pred)
mse = mean_squared_error(y_test, y_pred)
rmse = mean_squared_error(y_test, y_pred, squared=False)
r2 = r2_score(y_test, y_pred)
print(f"Mean Absolute Error (MAE): {mae:.2f}")
print(f"Mean Squared Error (MSE): {mse:.2f}")
print(f"Root Mean Squared Error (RMSE): {rmse:.2f}")
print(f"R^2 Score: {r2:.2f}")
```

```
Mean Absolute Error (MAE): 4.18
Mean Squared Error (MSE): 21.60
Root Mean Squared Error (RMSE): 4.65
R^2 Score: 0.95
```

The R^2 Score of 0.95 demonstrates that the model explains 95% of the variance in student scores, indicating an excellent fit to the data. Overall, these metrics suggest that the linear regression model is highly effective in predicting student performance based on study hours.

In [39]:
```
#Making Prediction
y_pred = model.predict(X_test) # Predicting the scores
```

In [40]:
```
# Comparing Actual vs Predicted
df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
df
```

Out[40]:

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

In [41]:
```
#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
```

In [ ]: