

TASK 2 TO EXPLORE SUPERVISED MACHINE LEARNING

```
In [1]: import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

```
In [9]: dataset = pd.read_csv('D:\student_scores.csv')
```

```
In [10]: dataset.shape
```

```
Out[10]: (25, 2)
```

```
In [11]: dataset.head()
```

```
Out[11]:
```

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

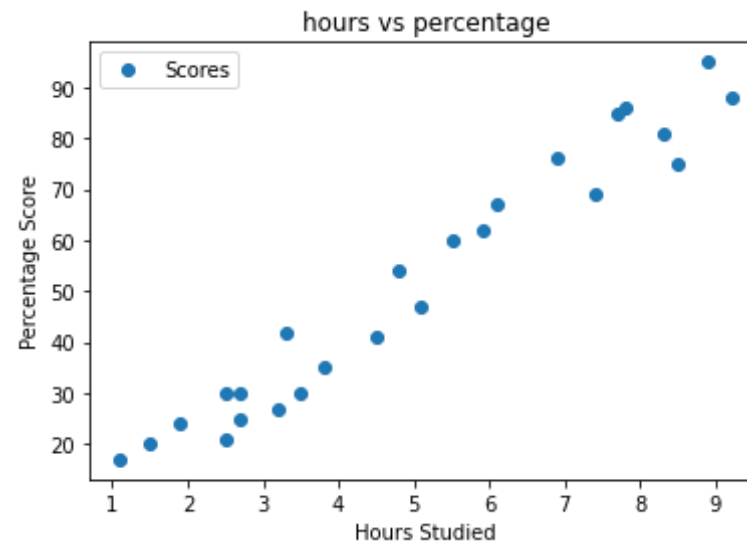
```
In [12]: dataset.describe()
```

```
Out[12]:
```

	Hours	Scores
count	25.000000	25.000000

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



```
In [19]: X=dataset.iloc[:, :-1].values
Y=dataset.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)
```

TRAINING THE LINEAR REGRESSION MODEL

```
In [21]: from sklearn.linear_model import LinearRegression
regressor = LinearRegression()
regressor.fit(X_train, y_train)
```

Out[21]: LinearRegression()

```
In [22]: print(regressor.intercept_)
```

2.018160041434669

```
In [23]: print(regressor.coef_)
```

[9.91065648]

PREDICTING THE RESULT USING TEST DATA

```
In [24]: y_pred = regressor.predict(X_test)
```

```
In [25]: df = pd.DataFrame({'Actual': y_test, 'Predicted': y_pred})
df
```

Out[25]:

	Actual	Predicted
0	20	16.884145
1	27	33.732261

	Actual	Predicted
2	69	75.357018
3	30	26.794801
4	62	60.491033

MEAN ABSOLUTE ERROR AND MEAN SQUARED ERROR

```
In [26]: from sklearn import metrics
print('Mean Absolute Error:', metrics.mean_absolute_error(y_test, y_pred))
print('Mean Squared Error:', metrics.mean_squared_error(y_test, y_pred))
print('Root Mean Squared Error:', np.sqrt(metrics.mean_squared_error(y_test, y_pred)))
```

Mean Absolute Error: 4.18385989900298
Mean Squared Error: 21.598769307217413
Root Mean Squared Error: 4.647447612100368

TAKING INPUT AND GIVING ANSWER AS OUTPUT

```
In [31]: hours = float(input(' Enter the hours of study:- '))
predicted = regressor.predict([[hours]])
print(" If a student studies for = {}".format(hours), "hours, then his/her predicted score is = {}".format(predicted))
```

Enter the hours of study:- 9.5
If a student studies for = 9.5 hours, then his/her predicted score is = [96.16939661]%

In []:

In []:

In []: