Data Science and Business Analytics (GRIP May21)

Author: Ravi charan Baraka

# Task 1 : Prediction using supervised ML

Problem statement:

```
Predict the percentage of a student based on the number of study hours if a student studies for 9.25 hrs/ day.
```

importing the required libraries

In [2]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
```

importing data set

In [3]:
```
#reading the data using pandas
df=pd.read_csv("http://bit.ly/w-data")
print("Data imported Successfully")
df.head()
```

```
Data imported Successfully
```

Out[3]:

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

Understanding Data

In [4]:
```
df.describe()
```

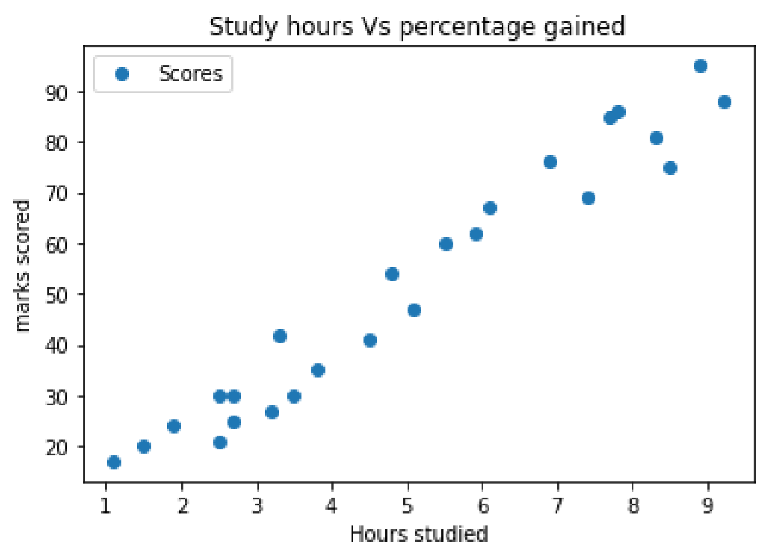
Out[4]:

	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 [5]:
```
df.shape
```

Out[5]:
```
(25, 2)
```

In [41]:
```
#plotting the distribution of scores
df.plot(x='Hours',y='Scores',style='o')
plt.title('Study hours Vs percentage gained')
plt.xlabel('Hours studied')
plt.ylabel('marks scored')
plt.show()
```



From the graph above, we can clearly see that there is a positive linear relation between the number of hours studied and percentage of score.

Cleaning the Data

In [7]:
```
df.isnull().sum()
```

Out[7]:
```
Hours     0
Scores    0
dtype: int64
```

preparing the data

In [8]:
```
x=df.iloc[:,:-1].values
y=df.iloc[:,1].values
```

In [33]:
```
#split the data for training and validation
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.20,random_state=0)
print("splitting is done")
```

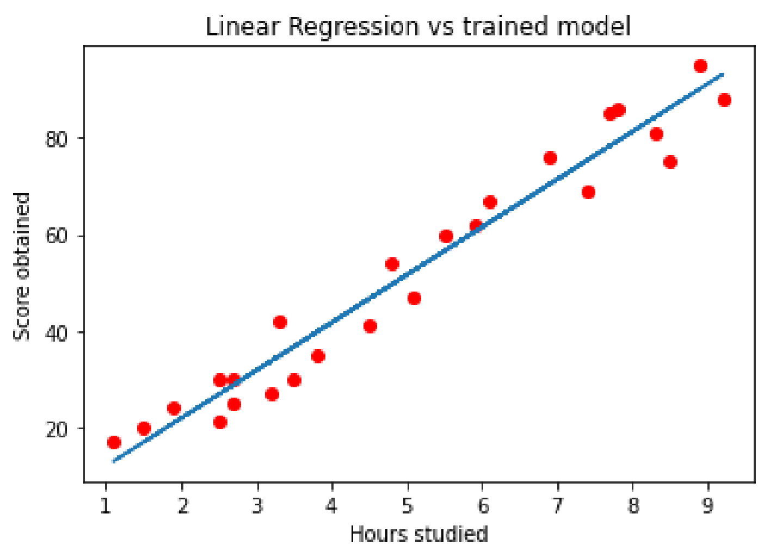
```
splitting is done
```

In [34]:
```
#training the algorithm(model)
from sklearn.linear_model import LinearRegression
model=LinearRegression()
model.fit(x_train,y_train)
print("Training Complete")
```

```
Training Complete
```

In [40]:
```
#plotting regression line
line=model.coef_*x+model.intercept_

# Plotting for the test data
plt.scatter(x, y,c="red")
plt.title('Linear Regression vs trained model')
plt.xlabel('Hours studied')
plt.ylabel('Score obtained')
plt.plot(x, line);
plt.show()
```



Predicting values

In [22]:
```
y_pred = model.predict(X_test)
```

In [24]:
```
y_pred
```

Out[24]:
```
array([16.88414476, 33.73226078, 75.357018  , 26.79480124, 60.49103328])
```

Comparing Actual Vs Predicted

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

In [26]:
```
compare
```

Out[26]:

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

In [29]:
```
#testing the accuracy of Model
result=model.score(X_test,y_test)
print(result)
```

```
0.9454906892105356
```

Solution for given problem statement:

In [30]:
```
hours=9.25
prediction=model.predict([[hours]])
print(prediction)
```

```
[93.69173249]
```

Evaluating the Model

In [31]:
```
from sklearn import metrics
print('Mean Absolute Error:',
      metrics.mean_absolute_error(y_test, y_pred))
```

```
Mean Absolute Error: 4.183859899002975
```

Conclusion:

For a student studying 9.25Hrs a day , the model predicts his score as 93.6917

In [ ]: