**Name : Kesia Sara Koshy**

# Prediction Using Supervised ML

## Aim:

To predict the percentage of a student based on the no.of study hours.

What will be predicted score if a student studies for 9.25 hrs/ day?

## What is ML?

Machine Learning is a subset of Artificial Intelligence, it may be defined as the process or study of computer algorithms in such a way that it helps to predict a potential decision or optimum decision based on data provided and scenarios.

## Supervised ML

Supervised Machine Learning is a method where the models are trained using labeled data, it needs supervision to train the model.

## Procedure & Analysis

The first step is to import the libraries that are required for the implementation of the code operation. Here we import the Pandas to import and analyze data, NumPy to perform the multi-dimensional operation, and matplotlib to perform graphical plot into the context.

In [1]:
```
# Importing all libraries required in this procedure
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:
```
data = pd.read_csv('TSF.csv')
data.head(10)  #loading the data to perform the desired operation
```

Out[2]:

| | Hours | Scores |
|---|---|---|
| **0** | 2.5 | 21 |
| **1** | 5.1 | 47 |
| **2** | 3.2 | 27 |
| **3** | 8.5 | 75 |
| **4** | 3.5 | 30 |
| **5** | 1.5 | 20 |
| **6** | 9.2 | 88 |
| **7** | 5.5 | 60 |
| **8** | 8.3 | 81 |
| **9** | 2.7 | 25 |

In [3]:
```
data.shape #To understand the data
```

Out[3]:
```
(25, 2)
```

From the above operation, we can see that there are 25 rows and 2 columns

In [4]:
```
#To check if there are any null values
data.isnull().sum()
```

Out[4]:
```
Hours     0
Scores    0
dtype: int64
```

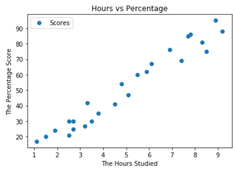
From the above function, we can see that there are no null values. Which conclude that our data is a cleaned data.

In [5]:
```
data.describe() #this function allows us to get its summary statistics
```

Out[5]:

| | 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 [6]:
```
data.plot(x='Hours', y='Scores', style='o')
plt.title('Hours vs Percentage')
plt.xlabel('The Hours Studied')
plt.ylabel('The Percentage Score')
plt.show()
```




From the above graph, we can see a linear relationship between the two variables. Therefore, it suggests us a simple linear regression model.

## To perform a Simple Linear Regression model

In [8]:
```
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
```

In [9]:
```
#splitting the data
X = data.iloc[:, :-1].values
y = data.iloc[:, 1].values

# Spliting the Data in two
X_train, X_test, y_train, y_test = train_test_split(X, y, random_state = 0)
```

In [10]:
```
regression = LinearRegression()
regression.fit(X_train, y_train)
```

Out[10]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=1, normalize=False)
```

**To predict the percentage**

In [11]:
```
pred_y = regression.predict(X_test)
prediction = pd.DataFrame({'Hours': [i[0] for i in X_test], 'Predicted Marks': [k for k in p
red_y]})
prediction
```

Out[11]:

| | Hours | Predicted Marks |
|---|---|---|
| **0** | 1.5 | 16.844722 |
| **1** | 3.2 | 33.745575 |
| **2** | 7.4 | 75.500624 |
| **3** | 2.5 | 26.786400 |
| **4** | 5.9 | 60.588106 |
| **5** | 3.8 | 39.710582 |
| **6** | 1.9 | 20.821393 |

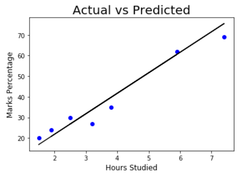
**Comparing predicted marks with actual marks**

In [12]:
```
compare_scores = pd.DataFrame({'Actual Marks': y_test, 'Predicted Marks': pred_y})
compare_scores
```

Out[12]:

| | Actual Marks | Predicted Marks |
|---|---|---|
| **0** | 20 | 16.844722 |
| **1** | 27 | 33.745575 |
| **2** | 69 | 75.500624 |
| **3** | 30 | 26.786400 |
| **4** | 62 | 60.588106 |
| **5** | 35 | 39.710582 |
| **6** | 24 | 20.821393 |

In [13]:
```
# Visvalising the above
plt.scatter(x=X_test, y=y_test, color='blue')
plt.plot(X_test, pred_y, color='Black')
plt.title('Actual vs Predicted', size=20)
plt.ylabel('Marks Percentage', size=12)
plt.xlabel('Hours Studied', size=12)
plt.show()
```




## What will be the predicted score of a student if he/she studies for 9.25 hrs/ day?

In [14]:
```
hours = [9.25]
answer = regression.predict([hours])
print("Score = {}".format(round(answer[0],3)))
```

```
Score = 93.893
```

Therefore we can say that, if a student studies for 9 hrs and 25 mins a day, he/she is likely to score 93.893 marks.