# Task 1: PREDICTION USING SUPERVISED ML

To predict the percentage of marks of the students based on the number of hours they have studied.

AUTHOR : KRUNAL RAJU CHAUDHAR

In [1]:
```
# IMPORT THE REQUIRED LIBRARIES
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression
from sklearn.metrics import mean_absolute_error
```

In [2]:
```
# reading the data
data = pd.read_csv ('https://raw.githubusercontent.com/AdiPersonalWorks/Random/mast
data.head(10)
```

Out[2]:

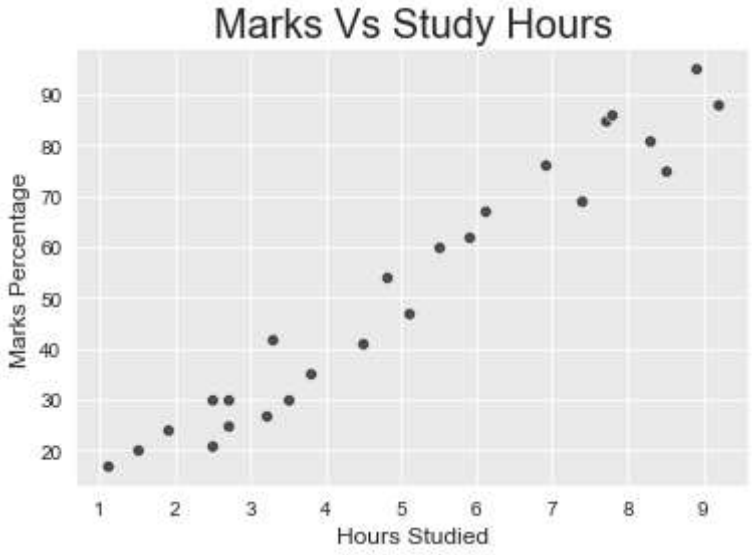
| | Hours | Scores |
|---|---|---|
| **0** | 2.5 | 21 |
| **1** | 5.1 | 47 |
| **2** | 3.2 | 27 |
| **3** | 8.5 | 75 |
| **4** | 3.5 | 30 |
| **5** | 1.5 | 20 |
| **6** | 9.2 | 88 |
| **7** | 5.5 | 60 |
| **8** | 8.3 | 81 |
| **9** | 2.7 | 25 |

In [3]:
```
# Check if there any null value in dataset
data.isnull == True
```

Out[3]:
```
False
```

There is no null value in the dataset so now we can visualize our data.

In [4]:
```
sns.set_style('darkgrid')
sns.scatterplot(y= data['Scores'], x= data['Hours'], color="red")
plt.title('Marks Vs Study Hours',size=20)
plt.ylabel('Marks Percentage', size=12)
plt.xlabel('Hours Studied', size=12)
plt.show()
```




# From the above scatter plot there looks to be correlation between the 'Marks Percentage' and 'Hours Studied', Let's plot a regression line to confirm the correlation.

Training the model

1) Splitting the data

In [5]:
```
# Defining X and y from the Data
X = data.iloc[:, :-1].values
y = data.iloc[:, 1].values

# Spliting the Data in two
train_X, val_X, train_y, val_y = train_test_split(X, y, random_state = 0)
```

2) Filling the data into the model

----------Model Trained----------

In [6]:
```
regression = LinearRegression()
regression.fit(train_X, train_y)
print("**********Model Trained**********")
```

```
**********Model Trained**********
```

# Predicting the Percentage of Marks

In [7]:
```
pred_y = regression.predict(val_X)
prediction = pd.DataFrame({'Hours': [i[0] for i in val_X], 'Predicted Marks': [k fo
prediction
```

Out[7]:

| | 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 the Predicted Marks with the Actual Marks

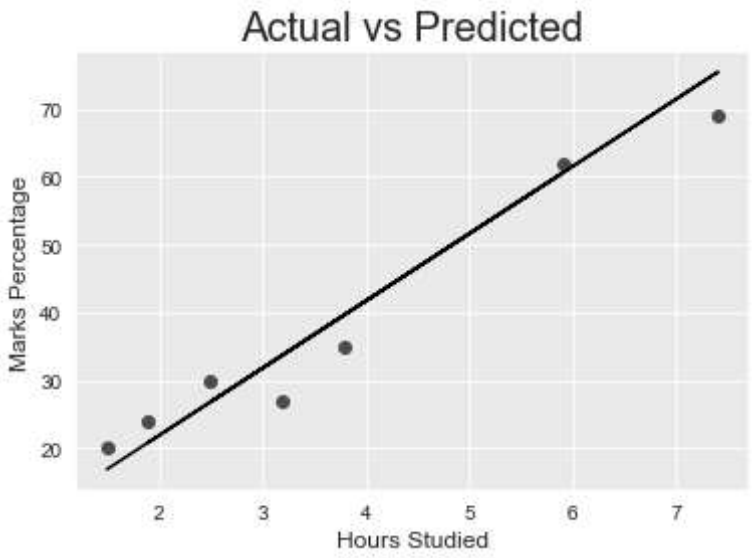
In [8]:
```
compare_scores = pd.DataFrame({'Actual Marks': val_y, 'Predicted Marks': pred_y})
compare_scores
```

Out[8]:

| | 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 |

# Visually Comparing the Predicted Marks with the Actual Marks

In [9]:
```
plt.scatter(x=val_X, y=val_y, color='red')
plt.plot(val_X, 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()
```




# Evaluating the Model

In [10]:
```
# Calculating the accuracy of the model
print('Mean absolute error: ',mean_absolute_error(val_y,pred_y))
```

```
Mean absolute error:  4.130879918502486
```

Small value of Mean absolute error states that the chances of error or wrong forecasting through the model are very less.

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

In [11]:
```
hours = [9.25]
answer = regression.predict([hours])
print("Score = {}".format(round(answer[0],3)))
```

```
Score = 93.893
```

According to the regression model if a student studies for 9.25 hours a day he/she is likely to score 93.89 marks.