TASK-1 Prediction_using_Supervised_ML Sparks Foundation

Name: Ashishkumar Trada

In [1]:
```
# importing libraries

import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
%matplotlib inline
```

In [2]:
```
# Extracting data
data = pd.read_csv('https://raw.githubusercontent.com/AdiPersonalWorks/Random/master/student_scores%20-%20student_scores.csv')
```

In [3]:
```
data.head(10)
```

Out[3]:

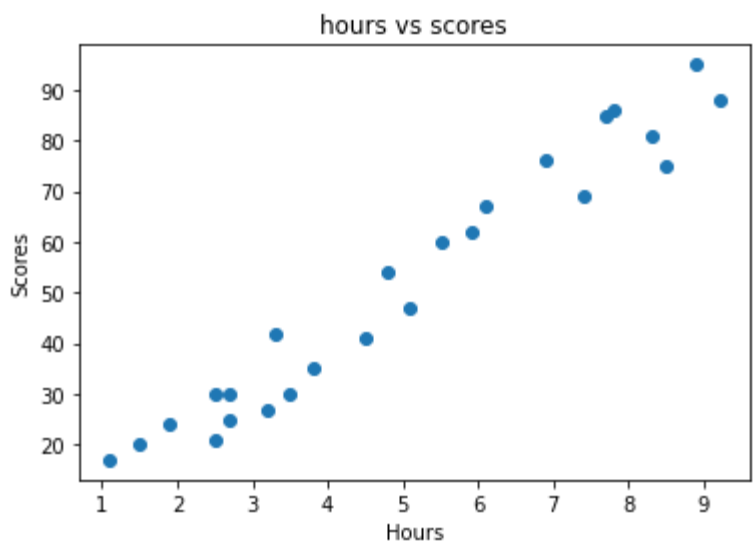
	Hours	Scores
0	2.5	21
1	5.1	47
2	3.2	27
3	8.5	75
4	3.5	30
5	1.5	20
6	9.2	88
7	5.5	60
8	8.3	81
9	2.7	25

In [4]:
```
print('shape: ', data.shape)
```

```
shape:  (25, 2)
```

In [5]:
```
# data ploating

plt.scatter(data.Hours, data.Scores)
plt.title('hours vs scores')
plt.xlabel('Hours')
plt.ylabel('Scores')
plt.show()
```



In [6]:
```
X = data.iloc[:, 0]
Y = data.iloc[:, 1]
```

In [7]:
```
# preparing train and test data

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 [8]:
```
x_train = x_train.values.reshape((-1,1))
```

In [9]:
```
x_test = x_test.values.reshape((-1,1))
```

Linear Regression

In [10]:
```
from sklearn.linear_model import LinearRegression
```

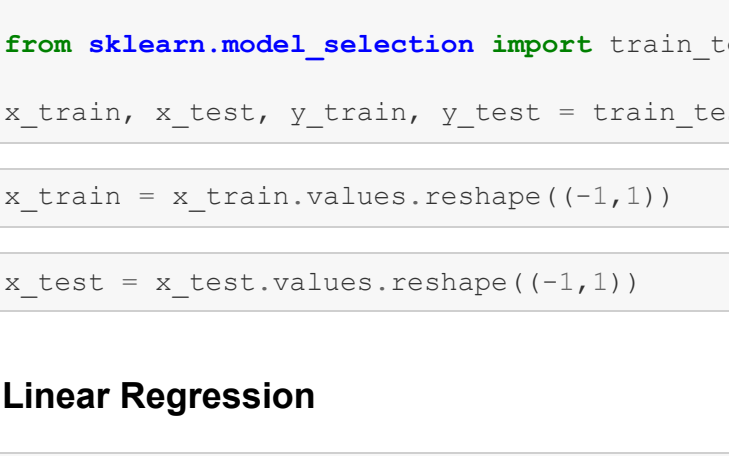
In [11]:
```
# tarining of data

lr = LinearRegression()
lr.fit(x_train, y_train)
```

Out[11]:
```
LinearRegression()
```

In [12]:
```
# line
y = lr.coef_ * X + lr.intercept_
```

In [13]:
```
#  ploating of line
plt.scatter(data.Hours, data.Scores)
plt.plot(X, y)
plt.show()
```



Prediction of model on test data

In [14]:
```
y_pred = lr.predict(x_test)
```

In [15]:
```
pd.DataFrame({'actual': y_test, 'predictions':y_pred})
```

Out[15]:

	actual	predictions
5	20	16.884145
2	27	33.732261
19	69	75.357018
16	30	26.794801
11	62	60.491033

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

In [16]:
```
hrs = 9.25
prediction_score = lr.predict([[hrs]])
print('if a student studies for 9.25 hrs/ day \n predicted score = ', prediction_score)
```

```
if a student studies for 9.25 hrs/ day
 predicted score =  [93.69173249]
```

Model Evaluation

In [17]:
```
from sklearn.metrics import r2_score, mean_squared_error
```

In [18]:
```
print("MSE: ", mean_squared_error(y_test, y_pred))
print('r2_score: ', r2_score(y_test, y_pred))
```

```
MSE:  21.598769307217406
r2_score:  0.9454906892105355
```