Task 1 - The Sparks Foundation

Name - Smita Rautmare

predict the percentage of student based on number of study hours

prediction using supervised ML

In [61]:
```
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
```

In [62]:
```
import warnings
warnings.filterwarnings('ignore')
```

In [63]:
```
path="http://bit.ly/w-data"
data = pd.read_csv(path)
data.head()
```

Out[63]:

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

In [64]:
```
data.info()
```

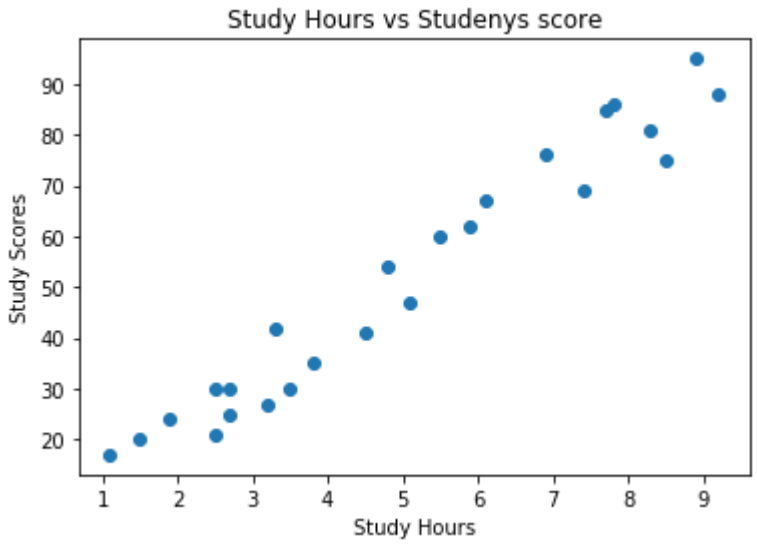
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 25 entries, 0 to 24
Data columns (total 2 columns):
Hours     25 non-null float64
Scores    25 non-null int64
dtypes: float64(1), int64(1)
memory usage: 528.0 bytes
```

In [65]:
```
data.describe()
```

Out[65]:

	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 [66]:
```
plt.scatter(x=data.Hours,y=data.Scores)
plt.xlabel("Study Hours")
plt.ylabel("Study Scores")
plt.title("Study Hours vs Studenys score")
plt.show()
```



In [67]:
```
x=data.drop("Scores",axis="columns")
y=data.drop("Hours",axis="columns")
print("shape of x",x.shape)
print("shape of y",y.shape)
```

```
shape of x (25, 1)
shape of y (25, 1)
```

In [68]:
```
#training a model
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=51)
```

In [69]:
```
print("shape of X train",X_train.shape)
print("shape of Y train",X_train.shape)
print("shape of X test",X_test.shape)
print("shape of Y test",X_test.shape)
```

```
shape of X train (20, 1)
shape of Y train (20, 1)
shape of X test (5, 1)
shape of Y test (5, 1)
```

Training a model

In [70]:
```
from sklearn.linear_model import LinearRegression
l=LinearRegression()
```

In [71]:
```
l.fit(X_train,Y_train)
```

Out[71]:
```
LinearRegression(copy_X=True, fit_intercept=True, n_jobs=None, normalize=False)
```

In [72]:
```
l.coef_
```

Out[72]:
```
array([[9.56433743]])
```

In [73]:
```
l.intercept_
```

Out[73]:
```
array([2.70197068])
```

In [74]:
```
y_pred=l.predict(X_test)
y_pred
```

Out[74]:
```
array([[55.30582657],
       [76.34736893],
       [68.69589898],
       [82.08597139],
       [28.52568176]])
```

In [75]:
```
pd.DataFrame(np.c_[X_test,Y_test,y_pred], columns=["Hours","Scores","scores_predi
```

Out[75]:

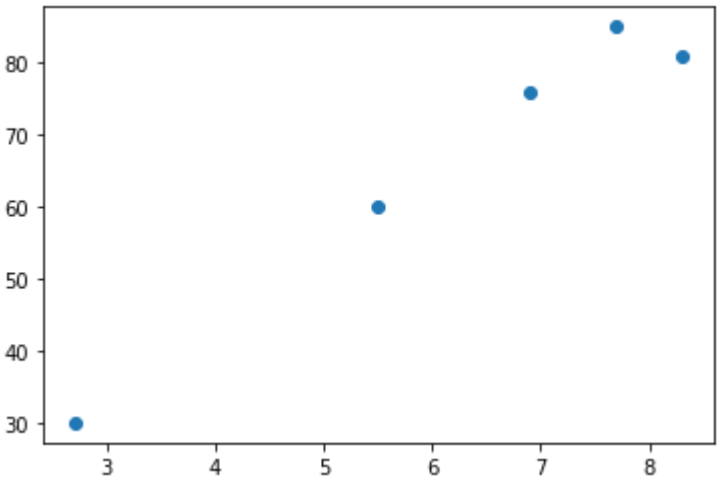
	Hours	Scores	scores_predict
0	5.5	60.0	55.305827
1	7.7	85.0	76.347369
2	6.9	76.0	68.695899
3	8.3	81.0	82.085971
4	2.7	30.0	28.525682

In [76]:
```
l.score(X_test,Y_test)
```

Out[76]:
```
0.9238518102278777
```

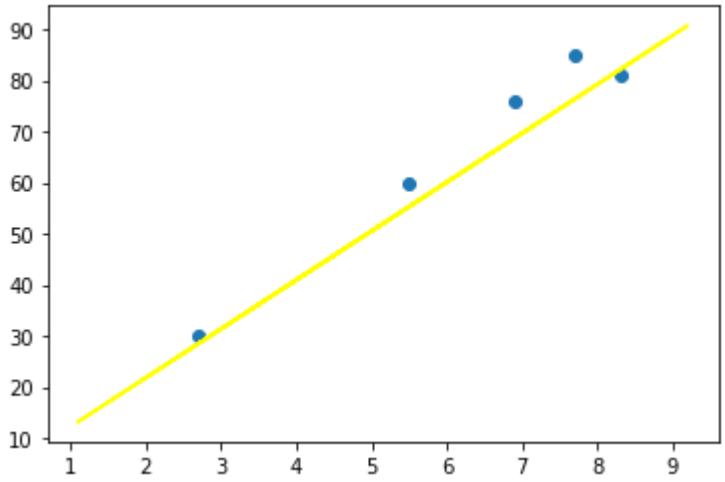
In [77]:
```
plt.scatter(X_test,Y_test)
```

Out[77]:
```
<matplotlib.collections.PathCollection at 0x2c170434448>
```



In [78]:
```
plt.scatter(X_test,Y_test)
plt.plot(X_train,l.predict(X_train),color="yellow")
```

Out[78]:
```
[<matplotlib.lines.Line2D at 0x2c17044e4c8>]
```



In [79]:
```
import joblib
joblib.dump(l,"Students_marks_Predictor.pkl")
```

Out[79]:
```
['Students_marks_Predictor.pkl']
```

In [80]:
```
model=joblib.load("Students_marks_Predictor.pkl")
```

In [81]:
```
model.predict([[9.25]])
```

Out[81]:
```
array([[91.17209195]])
```

In [ ]:
```

```