In [1]:
```
#import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
import random
from sklearn.linear_model import LinearRegression
from sklearn.linear_model import Ridge,Lasso
from sklearn.model_selection import train_test_split
```

In [2]:
```
#access dataset
df=pd.read_csv('Documents\\student_info.csv')
df
```

Out[2]:

	study_hours	student_marks
0	6.83	78.50
1	6.56	76.74
2	NaN	78.68
3	5.67	71.82
4	8.67	84.19
...	...	...
195	7.53	81.67
196	8.56	84.68
197	8.94	86.75
198	6.60	78.05
199	8.35	83.50

200 rows × 2 columns

In [3]:
```
#check null value
df.isnull().sum()
```

Out[3]:
```
study_hours      5
student_marks    0
dtype: int64
```

In [4]:
```
#calculate mean
df.mean()
```

Out[4]:
```
study_hours       6.995949
student_marks    77.933750
dtype: float64
```

In [5]:
```
#fill null value
df2=df.fillna(df.mean())
df2
```

Out[5]:

	study_hours	student_marks
0	6.830000	78.50
1	6.560000	76.74
2	6.995949	78.68
3	5.670000	71.82
4	8.670000	84.19
...	...	...
195	7.530000	81.67
196	8.560000	84.68
197	8.940000	86.75
198	6.600000	78.05
199	8.350000	83.50

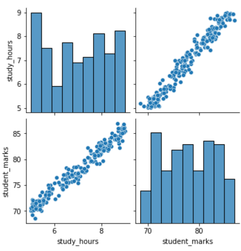
200 rows × 2 columns

In [6]:
```
#check null value
df2.isnull().sum()
```

Out[6]:
```
study_hours      0
student_marks    0
dtype: int64
```

In [7]:
```
#data visualization using seaborn
sns.pairplot(df2)
```

Out[7]:
```
<seaborn.axisgrid.PairGrid at 0x1e3f7f76f70>
```



In [8]:
```
X=df2.drop('student_marks',axis=1)
y=df2['student_marks']
```

In [9]:
```
#train-test-split data
X_train,X_test,y_train,y_test=train_test_split(X,y,test_size=0.2,random_state=51)
```

In [10]:
```
X_train.shape
```

Out[10]:
```
(160, 1)
```

In [11]:
```
X_test.shape
```

Out[11]:
```
(40, 1)
```

In [12]:
```
lr=LinearRegression()
lr.fit(X_train,y_train)
lr.score(X_test,y_test)
```

Out[12]:
```
0.9514124242154464
```

In [13]:
```
#improving accuracy using ridge
rd=Ridge()
rd.fit(X_train,y_train)
rd.score(X_test,y_test)
```

Out[13]:
```
0.9518132972505241
```

In [19]:
```
#incresing alpha value for more accuracy
rd1=Ridge(alpha=13)
rd1.fit(X_train,y_train)
rd1.score(X_test,y_test)
```

Out[19]:
```
0.9538194744711458
```

In [20]:
```
#predicting X_test
rd1.predict(X_test)
```

Out[20]:
```
array([82.85556311, 78.85925499, 84.23746405, 85.43262162, 84.38685874,
       80.61464267, 73.10755919, 71.98709897, 73.48104593, 71.98709897,
       73.70513797, 76.46893985, 73.48104593, 73.81718399, 82.70616841,
       70.7919414 , 73.48104593, 78.70986029, 75.6846177 , 82.4447277 ,
       76.73038057, 70.86663875, 74.93764421, 77.98510448, 85.20852957,
       82.33268167, 76.50628852, 84.68564814, 78.37372223, 81.21222145,
       81.54835952, 82.89291178, 81.99654361, 80.95078073, 73.81718399,
       71.5015662 , 72.21119101, 81.36161615, 72.88346714, 72.24853968])
```

In [21]:
```
y_test
```

Out[21]:
```
148    82.02
104    77.55
4      84.19
7      85.46
192    84.03
160    80.81
118    73.61
58     70.90
190    73.14
174    73.02
23     75.02
10     75.37
115    74.44
86     73.40
67     81.70
68     69.27
177    73.64
171    77.63
128    77.01
14     83.08
82     76.63
50     72.22
45     72.96
31     76.14
176    85.96
21     83.36
198    78.05
89     84.60
35     76.76
36     81.24
113    80.86
121    82.69
99     82.30
162    79.17
79     73.34
131    71.86
65     70.06
13     80.76
85     72.87
42     71.10
Name: student_marks, dtype: float64
```

In [22]:
```
#this model gives 95% accuracy
```

In [ ]:
```

```