```python
#Import libraries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

```python
path = r"/content/student_info.csv"
df  = pd.read_csv(path)
```

```python
df.head()
```

	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

```python
df.tail()
```

	study_hours	student_marks
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

```python
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 200 entries, 0 to 199
Data columns (total 2 columns):
 #   Column         Non-Null Count  Dtype
---  ------         --------------  -----
 0   study_hours    195 non-null    float64
 1   student_marks  200 non-null    float64
dtypes: float64(2)
memory usage: 3.2 KB
```

```python
df.describe()
```

	study_hours	student_marks
count	195.000000	200.00000
mean	6.995949	77.93375
std	1.253060	4.92570
min	5.010000	68.57000
25%	5.775000	73.38500
50%	7.120000	77.71000
75%	8.085000	82.32000
max	8.990000	86.99000

```python
plt.scatter(x =df.study_hours, y = df.student_marks)
plt.xlabel("Students Study Hours")
plt.ylabel("Students marks")
plt.title("Scatter Plot of Students Study Hours vs Students marks")
plt.show()
```

Scatter Plot of Students Study Hours vs Students marks
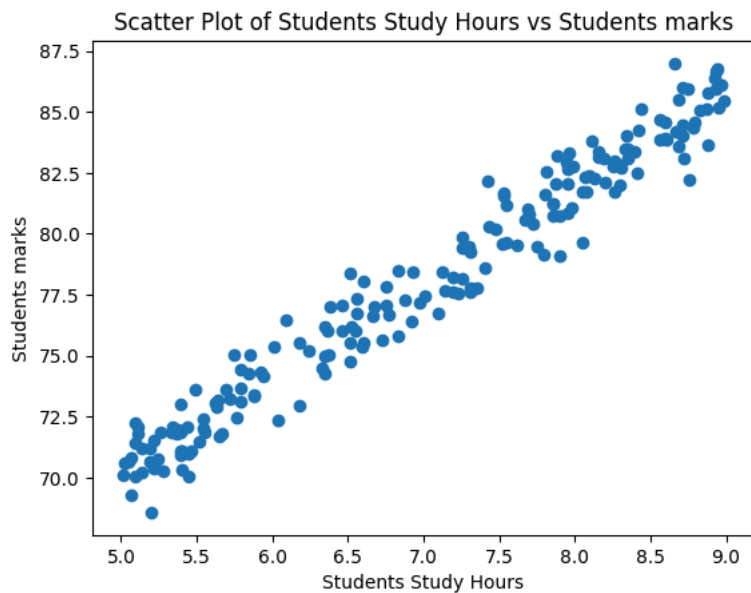

```
df.isnull().sum()
```

```
study_hours      5
student_marks    0
dtype: int64
```

```
df.mean()
```

```
study_hours       6.995949
student_marks    77.933750
dtype: float64
```

```
df2 = df.fillna(df.mean())
```

```
df2.isnull().sum()
```

```
study_hours      0
student_marks    0
dtype: int64
```

```
df2.head()
```

	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

```
X = df2.drop("student_marks", axis = "columns")
y = df2.drop("study_hours", axis = "columns")
print("shape of X = ", X.shape)
print("shape of y = ", y.shape)
```

```
shape of X =  (200, 1)
shape of y =  (200, 1)
```

```
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)
print("shape of X_train = ", X_train.shape)
print("shape of y_train = ", y_train.shape)
print("shape of X_test = ", X_test.shape)
print("shape of y_test = ", y_test.shape)
```

```
shape of X_train =  (160, 1)
shape of y_train =  (160, 1)
shape of X_test =  (40, 1)
shape of y_test =  (40, 1)
```

```python
# y = m * x + c
from sklearn.linear_model import LinearRegression
lr = LinearRegression()
```

```python
lr.fit(X_train,y_train)
```

```
    ▾ LinearRegression
     LinearRegression()
```

```python
lr.coef_
```

```
    array([[3.93571802]])
```

```python
lr.intercept_
```

```
    array([50.44735504])
```

```python
m = 3.93
c = 50.44
y  = m * 4 + c
y
```

```
    66.16
```

```python
lr.predict([[4]])[0][0].round(2)
```

```
    /usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but LinearRegression
      warnings.warn(
    66.19
```

```python
y_pred  = lr.predict(X_test)
y_pred
```

```
    array([[83.11381458],
           [78.9025963 ],
           [84.57003024],
           [85.82946001],
           [84.72745896],
           [80.75238377],
           [72.84159055],
           [71.66087515],
           [73.23516235],
           [71.66087515],
           [73.47130543],
           [76.38373677],
           [73.23516235],
           [73.58937697],
           [82.95638585],
           [70.40144538],
           [73.23516235],
           [78.74516758],
           [75.55723598],
           [82.68088559],
           [76.65923703],
           [70.48015974],
           [74.77009238],
           [77.98143645],
           [85.59331693],
           [82.56281405],
           [76.42309395],
           [85.0423164 ],
           [78.39095296],
           [81.38209865],
           [81.73631327],
           [83.15317176],
           [82.20859943],
           [81.10659839],
           [73.58937697],
           [71.1492318 ],
           [71.89701823],
           [81.53952737],
           [72.60544747],
           [71.93637541]])
```

```python
pd.DataFrame(np.c_[X_test, y_test, y_pred], columns = ["study_hours", "student_marks_original","student_marks_predicted"])
```

	study_hours	student_marks_original	student_marks_predicted
0	8.300000	82.02	83.113815
1	7.230000	77.55	78.902596
2	8.670000	84.19	84.570030
3	8.990000	85.46	85.829460
4	8.710000	84.03	84.727459
5	7.700000	80.81	80.752384
6	5.690000	73.61	72.841591
7	5.390000	70.90	71.660875
8	5.790000	73.14	73.235162
9	5.390000	73.02	71.660875
10	5.850000	75.02	73.471305
11	6.590000	75.37	76.383737
12	5.790000	74.44	73.235162
13	5.880000	73.40	73.589377
14	8.260000	81.70	82.956386
15	5.070000	69.27	70.401445
16	5.790000	73.64	73.235162
17	7.190000	77.63	78.745168
18	6.380000	77.01	75.557236
19	8.190000	83.08	82.680886
20	6.660000	76.63	76.659237
21	5.090000	72.22	70.480160
22	6.180000	72.96	74.770092
23	6.995949	76.14	77.981436
24	8.930000	85.96	85.593317
25	8.160000	83.36	82.562814
26	6.600000	78.05	76.423094
27	8.790000	84.60	85.042316
28	7.100000	76.76	78.390953
29	7.860000	81.24	81.382099
30	7.950000	80.86	81.736313

```
lr.score(X_test,y_test)
```

```
0.9514124242154466
```
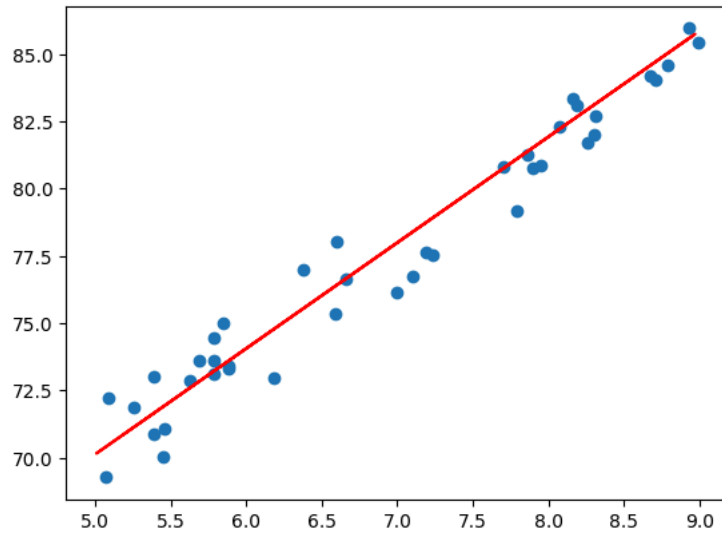| 33 | 7.790000 | 79.17 | 81.106598 |

```
plt.scatter(X_train,y_train)
```

⊋▾ <matplotlib.collections.PathCollection at 0x787eb446a9b0>

```python
plt.scatter(X_test, y_test)
plt.plot(X_train, lr.predict(X_train), color = "r")
```

⊋▾ [<matplotlib.lines.Line2D at 0x787eb41b38b0>]


```python
import joblib
joblib.dump(lr, "student_mark_predictor.pkl")
```

⊋▾ ['student_mark_predictor.pkl']

```python
model = joblib.load("student_mark_predictor.pkl")
```

```python
model.predict([[5]])[0][0]
```

⊋▾ /usr/local/lib/python3.10/dist-packages/sklearn/base.py:439: UserWarning: X does not have valid feature names, but LinearRegression
        warnings.warn(
     70.12594512018406

```python
import joblib
joblib.dump(lr, "student_mark_predictor.pkl")
```

⊋▾ ['student_mark_predictor.pkl']

```python
from google.colab import files
files.download("student_mark_predictor.pkl")
```

⊋▾