In [2]:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
import warnings
warnings.filterwarnings("ignore")
```

Load dataset

In [3]:

```
df = pd.read_csv(r"C:\Users\s323\Desktop\Gatherings\Data Science\Datasets\student_
```

In [4]:

```
df.head()
```

Out[4]:

	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

In [5]:

```
df.tail()
```

Out[5]:

	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

In [7]:

```
df.shape
```

Out[7]:

```
(200, 2)
```

EDA

In [9]:

```
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
```

In [10]:

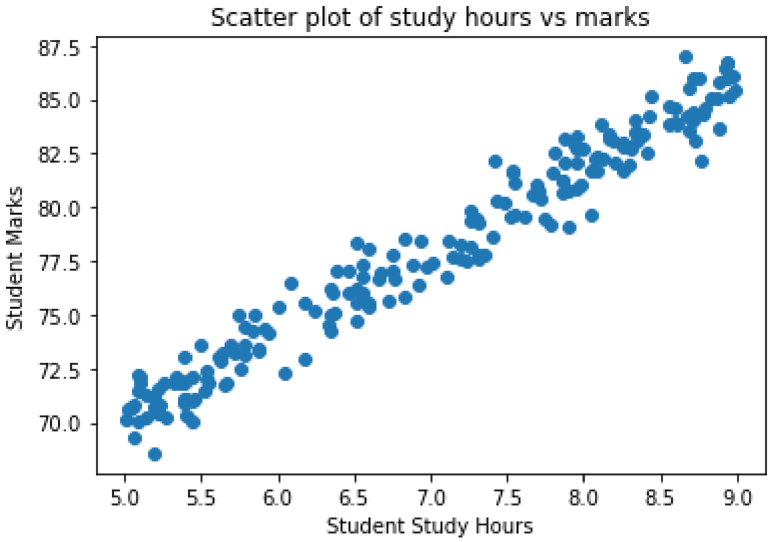
```
df.describe()
```

Out[10]:

	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

In [15]:

```
plt.scatter(x=df.study_hours,y=df.student_marks)
plt.xlabel("Student Study Hours")
plt.ylabel("Student Marks")
plt.title("Scatter plot of study hours vs marks")
plt.show()
```



Preparation of data for ML algorithm

- Cleasining of data

In [17]:

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

Out[17]:

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

In [21]:

```
df.mean()
```

Out[21]:

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

In [22]:

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

In [23]:

```
df2.isnull().any()
```

Out[23]:

```
study_hours      False
student_marks    False
dtype: bool
```

Split our dataset

In [24]:

```
X = df2.drop("student_marks",axis=1)
Y = df2["student_marks"]
```

In [25]:

```
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)
```

Linear Regression

In [27]:

```
from sklearn.linear_model import LinearRegression
lin_reg=LinearRegression()
```

In [29]:

```
lin_reg.fit(x_train,y_train)
```

Out[29]:

```
LinearRegression()
```

In [32]:

```
lin_reg.coef_
# m
```

Out[32]:

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

In [34]:

```
lin_reg.intercept_
# c
```

Out[34]:

```
50.44735503694244
```

In [35]:

```
lin_reg.score(x_test,y_test)
```

Out[35]:

```
0.9514124242154464
```

Make predictions

In [41]:

```
y_hat=lin_reg.predict(x_test)
```

Accuracy - fine tune model

In [42]:

```
from sklearn.metrics import mean_squared_error,mean_absolute_error,r2_score
```

In [43]:

```
r2_score(y_test,y_hat)
```

Out[43]:

```
0.9514124242154464
```

In [45]:

```
mean_squared_error(y_test, y_hat)
```

Out[45]:

```
1.1080039417516496
```

In [46]:

```
mean_absolute_error(y_test,y_hat)
```

Out[46]:

```
0.8780690208883186
```

Save ML model

```
In [47]: import joblib
         #joblib - model saves our linear regression model
         joblib.dump(lin_reg,"student_marks_predictor.pkl")

Out[47]: ['student_marks_predictor.pkl']
```

```
In [48]: model=joblib.load("student_marks_predictor.pkl")
```

```
In [51]: model.predict([[5]])

Out[51]: array([70.12594512])
```

Launch, Monitor, Maintain our system

- Last Journey in a ML model

```
In [ ]:
```