Student Marks Prediction using Python

The dataset I am using for the student marks prediction task is downloaded from Kaggle. Now let's start with this task by importing the necessary Python libraries and dataset:

```
import numpy as np
import pandas as pd
import plotly.express as px
from sklearn.model_selection import train_test_split
from sklearn.linear_model import LinearRegression

data = pd.read_csv("/content/Student_Marks.csv")
print(data.head(10))
```

```
   number_courses  time_study   Marks
0               3       4.508  19.202
1               4       0.096   7.734
2               4       3.133  13.811
3               6       7.909  53.018
4               8       7.811  55.299
5               6       3.211  17.822
6               3       6.063  29.889
7               5       3.413  17.264
8               4       4.410  20.348
9               3       6.173  30.862
```

So there are only three columns in the dataset. The marks column is the target column as we have to predict the marks of a student.

Now before moving forward, let's have a look at whether this dataset contains any null values or not:

```
print(data.isnull().sum())
```

```
number_courses    0
time_study        0
Marks             0
dtype: int64
```

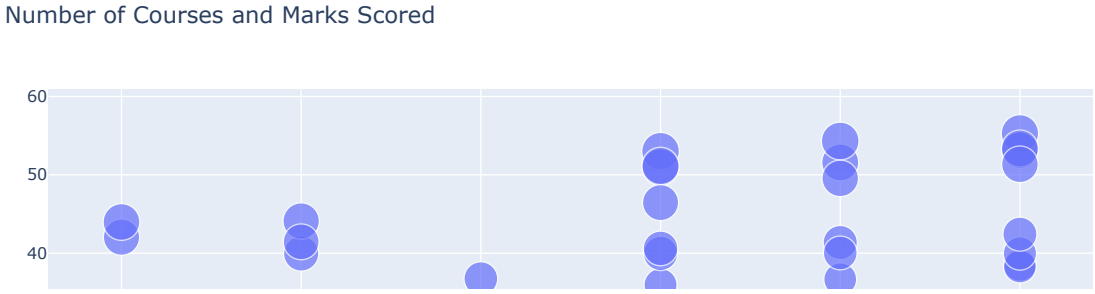
The dataset is ready to use because there are no null values in the data. There is a column in the data containing information about the number of courses students have chosen. Let's look at the number of values of all values of this column:

```
data["number_courses"].value_counts()
```

```
3    22
4    21
6    16
8    16
7    15
5    10
Name: number_courses, dtype: int64
```

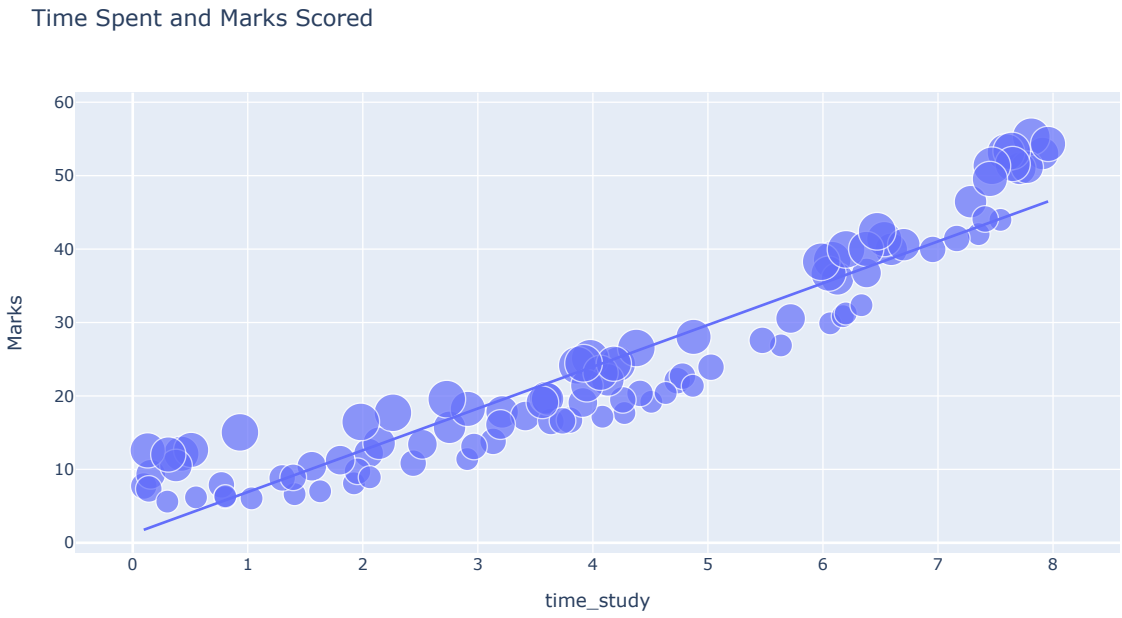
So there are a minimum of three and a maximum of eight courses students have chosen. Let's have a look at a scatter plot to see whether the number of courses affects the marks of a student:

```
figure = px.scatter(data_frame=data, x = "number_courses",
                    y = "Marks", size = "time_study",
                    title="Number of Courses and Marks Scored")
figure.show()
```



According to the above data visualization, we can say that the number of courses may not affect the marks of a student if the student is studying for more time daily. So let's have a look at the relationship between the time a studied daily and the marks scored by the student:

```
figure = px.scatter(data_frame=data, x = "time_study",
                    y = "Marks", size = "number_courses",
                    title="Time Spent and Marks Scored", trendline="ols")
figure.show()
```



You can see that there is a linear relationship between the time studied and the marks obtained. This means the more time students spend studying, the better they can score.

Now let's have a look at the correlation between the marks scored by the students and the other two columns in the data:

```
correlation = data.corr()
print(correlation["Marks"].sort_values(ascending=False))
```

```
Marks             1.000000
time_study        0.942254
number_courses    0.417335
Name: Marks, dtype: float64
```

So the time_studied column is more correlated with the marks column than the other column.

Student Marks Prediction Model

Now let's move to the task of training a machine learning model for predicting the marks of a student. Here, I will first start by splitting the data into training and test sets:

```
x = np.array(data[["time_study", "number_courses"]])
y = np.array(data["Marks"])
xtrain, xtest, ytrain, ytest = train_test_split(x, y,
```

```
                                                test_size=0.2,
                                                random_state=42)
```

Now I will train a machine learning model using the linear regression algorithm:

```
model = LinearRegression()
model.fit(xtrain, ytrain)
model.score(xtest, ytest)
```

```
     0.9459936100591211
```

Now let's test the performance of this machine learning model by giving inputs based on the features we have used to train the model and predict the marks of a student:

```
# Features = [["time_study", "number_courses"]]
features = np.array([[4.508, 3]])
model.predict(features)
```

```
     array([22.30738483])
```

So this is how you can predict the marks of a student with machine learning using Python.

Summary

So this is how you can solve the problem of student marks prediction with machine learning. It is a good regression problem for data science beginners as it is easy to solve and understand. I hope you liked this article on Student marks prediction with machine learning using Python. Feel free to ask valuable questions in the comments section below.