In [1]:
```
import  pandas  as  pd
import  numpy  as  np
import  pickle
import  warnings
warnings . filterwarnings ( 'ignore' )
```

In [3]:
```
data = pd . read_csv ( "C:/Users/gonab/OneDrive/Desktop/student data.csv" )
data
```

Out[3]:

	Hours Studied	Previous Scores	Extracurricular Activities	Sleep Hours	Sample Question Papers Practiced	Performance Index
0	7	99	Yes	9	1	91.0
1	4	82	No	4	2	65.0
2	8	51	Yes	7	2	45.0
3	5	52	Yes	5	2	36.0
4	7	75	No	8	5	66.0
...	...	...	...	...	...	...
9995	1	49	Yes	4	2	23.0
9996	7	64	Yes	8	5	58.0
9997	6	83	Yes	8	5	74.0
9998	9	97	Yes	7	0	95.0
9999	7	74	No	8	1	64.0

10000 rows × 6 columns

In [4]:
```
data . isnull (). sum ()
```

Out[4]:
```
Hours Studied 0
Previous Scores 0
Extracurricular Activities 0
Sleep Hours 0
Sample Question Papers Practiced 0
Performance Index 0
dtype: int64
```

In [5]:
```
data . describe ()
```

Out[5]:

	Hours Studied	Previous Scores	Sleep Hours	Sample Question Papers Practiced	Performance Index
count	10000.000000	10000.000000	10000.000000	10000.000000	10000.000000
mean	4.992900	69.445700	6.530600	4.583300	55.224800
std	2.589309	17.343152	1.695863	2.867348	19.212558
Min	1.000000	40.000000	4.000000	0.000000	10.000000
25%	3.000000	54.000000	5.000000	2.000000	40.000000
50%	5.000000	69.000000	7.000000	5.000000	55.000000
75%	7.000000	85.000000	8.000000	7.000000	71.000000
max	9.000000	99.000000	9.000000	9.000000	100.000000

In [6]:
```
data . info ()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 6 columns):
 # Column Non-Null Count Dtype
--- ------ ------ -------- -----
 0 Hours Studied 10000 non-null int64
 1 Previous Scores 10000 non-null int64
 2 Extracurricular Activities 10000 non-null object
 3 Sleep Hours 10000 non-null int64
 4 Sample Question Papers Practiced 10000 non-null int64
 5 Performance Index 10000 non-null float64
dtypes: float64(1), int64(4), object(1)
memory usage: 468.9+ KB
```

In [7]:
```
data . shape
```

Out[7]:
```
(10000, 6)
```

In [8]:
```
data . head ()
```

Out[8]:

	Hours Studied	Previous Scores	Extracurricular Activities	Sleep Hours	Sample Question Papers Practiced	Performance Index
0	7	99	Yes	9	1	91.0
1	4	82	No	4	2	65.0
2	8	51	Yes	7	2	45.0
3	5	52	Yes	5	2	36.0
4	7	75	No	8	5	66.0

In [9]:
```
data . tail ()
```

Out[9]:

	Hours Studied	Previous Scores	Extracurricular Activities	Sleep Hours	Sample Question Papers Practiced	Performance Index
9995	1	49	Yes	4	2	23.0
9996	7	64	Yes	8	5	58.0
9997	6	83	Yes	8	5	74.0
9998	9	97	Yes	7	0	95.0
9999	7	74	No	8	1	64.0

In [33]:
```
data [ 'Performance Index' ]. unique ()
```

Out[33]:
```
array([ 91., 65., 45., 36., 66., 61., 63., 42., 69., 84., 73., 27.,
        33., 68., 43., 67. , 70., 30., 71., 85., 57., 35., 49., 83., 74.
        , 39., 58., 47., 60., 32., 64., 54., 17 .,
        53., 75., 52., 78., 38., 98., 87., 41., 81., 15., 88.,
        95., 29., 21., 76., 25., 34., 50., 56., 82., 23., 46., 92., 77., 8
6., 44.,
        94., 40., 100., 31., 26., 18., 51. ,
        72., 16., 28., 89., 48., 37., 62., 59., 19., 79., 22., 10.,
        90., 80., 24., 20., 96 ., 55., 97., 12., 93., 14.,
        99., 11., 13.])
```

In [36]:
```
data [ 'Hours Studied' ]. unique ()
```

Out[36]:
```
array([7, 4, 8, 5, 3, 6, 2, 1, 9], dtype=int64)
```

In [35]:
```
data . groupby ([ 'Extracurricular Activities' ]). count ()
```

Out[35]:

Extracurricular Activities	Hours Studied	Previous Scores	Sleep Hours	Sample Question Papers Practiced	Performance Index
0	5052	5052	5052	5052	5052
1	4948	4948	4948	4948	4948

In [11]:
```
data [ 'Extracurricular Activities' ] =  data [ 'Extracurricular Activities
data . head ()
```

Out[11]:

	Hours Studied	Previous Scores	Extracurricular Activities	Sleep Hours	Sample Question Papers Practiced	Performance Index
0	7	99	1	9	1	91.0
1	4	82	0	4	2	65.0
2	8	51	1	7	2	45.0
3	5	52	1	5	2	36.0
4	7	75	0	8	5	66.0

In [12]:
```
data . corr ()[ 'Performance Index' ]
```

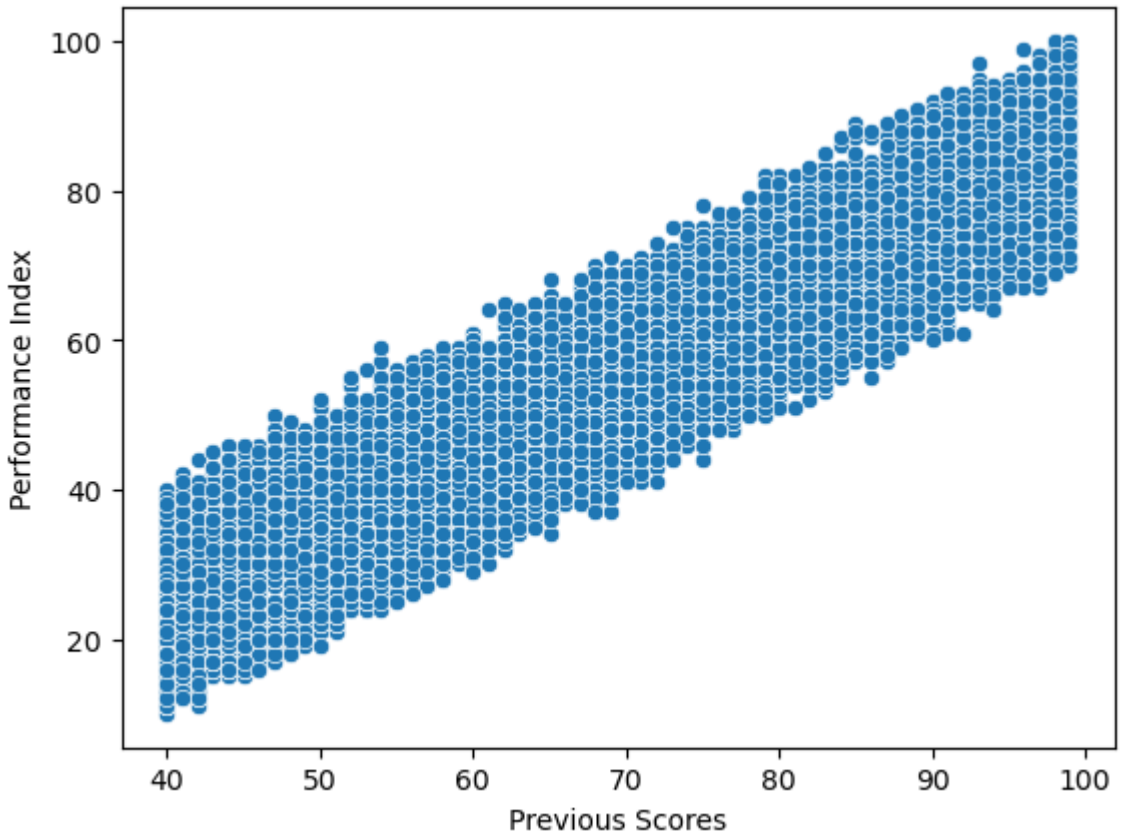
Out[12]:
```
Hours Studied                       0.373730
Previous Scores                     0.915189
Extracurricular Activities          0.024525
Sleep Hours                         0.048106
Sample Question Papers Practiced    0.043268
Performance Index                   1.000000
Name: Performance Index, dtype: float64
```

plotting

In [13]:
```
import seaborn as sns
import matplotlib.pyplot as plt
```

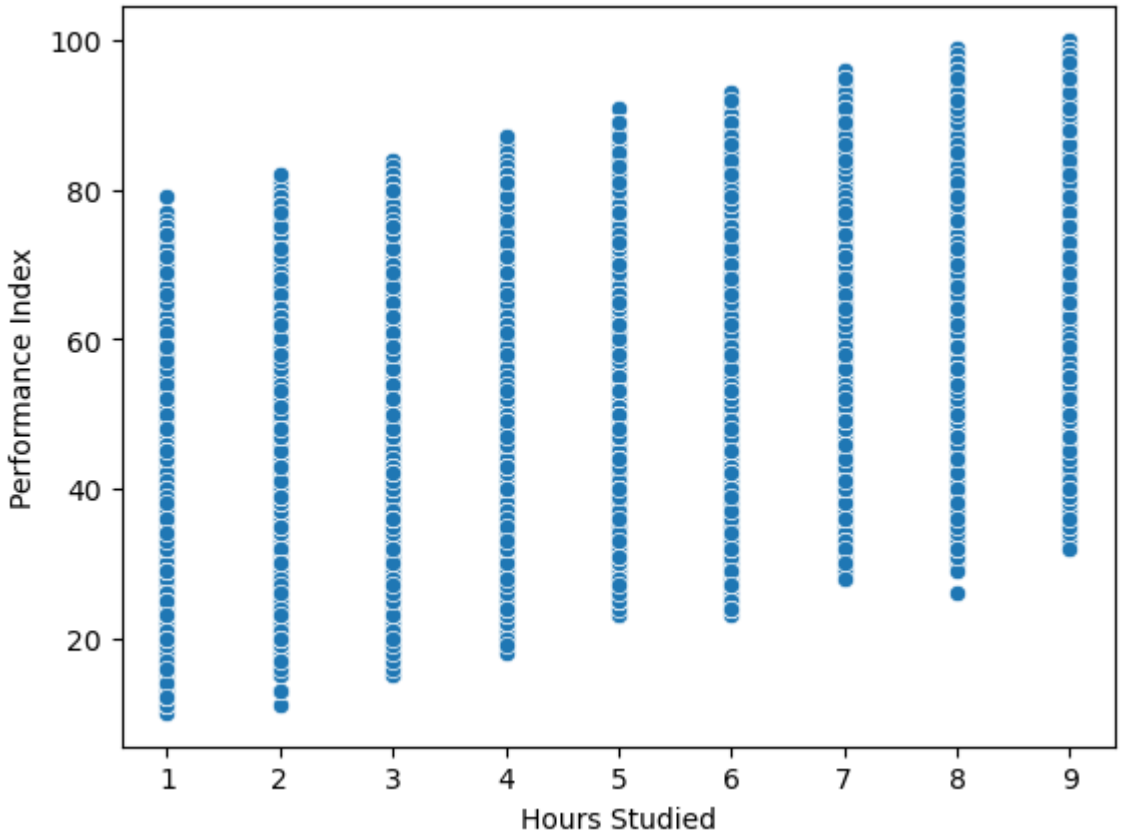
In [14]:
```
sns.scatterplot(data=data,x='Previous Scores',y='Performance Index')
```

Out[14]:
```
<Axes: xlabel='Previous Scores', ylabel='Performance Index'>
```



In [16]:
```
sns.scatterplot(data=data,x='Hours Studied',y='Performance Index')
```

Out[16]:
```
<Axes: xlabel='Hours Studied', ylabel='Performance Index'>
```



In [17]:
```
data=pd.get_dummies(data,dtype=int)
data.head()
```

Out[17]:

	Hours Studied	Previous Scores	Extracurricular Activities	Sleep Hours	Sample Question Papers Practiced	Performance Index
0	7	99	1	9	1	91.0
1	4	82	0	4	2	65.0
2	8	51	1	7	2	45.0
3	5	52	1	5	2	36.0
4	7	75	0	8	5	66.0

modelling

In [19]:
```
x= data.drop("Performance Index",axis=1)
y = data['Performance Index']
x
```

Out[19]:

	Hours Studied	Previous Scores	Extracurricular Activities	Sleep Hours	Sample Question Papers Practiced
0	7	99	1	9	1
1	4	82	0	4	2
2	8	51	1	7	2
3	5	52	1	5	2
4	7	75	0	8	5
...	...	...	...	...	...
9995	1	49	1	4	2
9996	7	64	1	8	5
9997	6	83	1	8	5
9998	9	97	1	7	0
9999	7	74	0	8	1

10000 rows × 5 columns

In [20]:
```
y
```

Out[20]:
```
0       91.0
1       65.0
2       45.0
3       36.0
4       66.0
        ...
9995    23.0
9996    58.0
9997    74.0
9998    95.0
9999    64.0
Name: Performance Index, Length: 10000, dtype: float64
```

In [24]:
```
from sklearn.model_selection import train_test_split
x_train, x_test, y_train, y_test = train_test_split( x, y, test_size=0.3, r
x_test.head()
```

Out[24]:

	Hours Studied	Previous Scores	Extracurricular Activities	Sleep Hours	Sample Question Papers Practiced
1977	5	74	1	8	7
3880	3	97	1	5	1
52	6	81	0	9	9
2551	9	67	0	4	9
2246	5	79	1	4	1

In [25]:
```
from sklearn.linear_model import LinearRegression
reg=LinearRegression()
reg . fit ( x_train , y_train )
```

Out[25]:
```
LinearRegression()
```
In a Jupyter environment, please rerun this cell to show the HTML representation or trust the notebook.
On GitHub, the HTML representation is unable to render, please try loading this page with nbviewer.org.

In [26]:
```
ypred = reg . predict ( x_test )
ypred
```

Out[26]:
```
array([61.33752771, 76.50352646, 71.549652 , ..., 47.77402083,
       54.48246304, 33.93339642])
```

In [27]:
```
from  sklearn . metrics  import  r2_score
r2_score ( y_test , ypred )
```

Out[27]:
```
0.9888882665528521
```

In [28]:
```
from  sklearn . metrics  import  mean_squared_error
mean_squared_error ( ypred , y_test )
```

Out[28]:
```
4.19157982492363
```

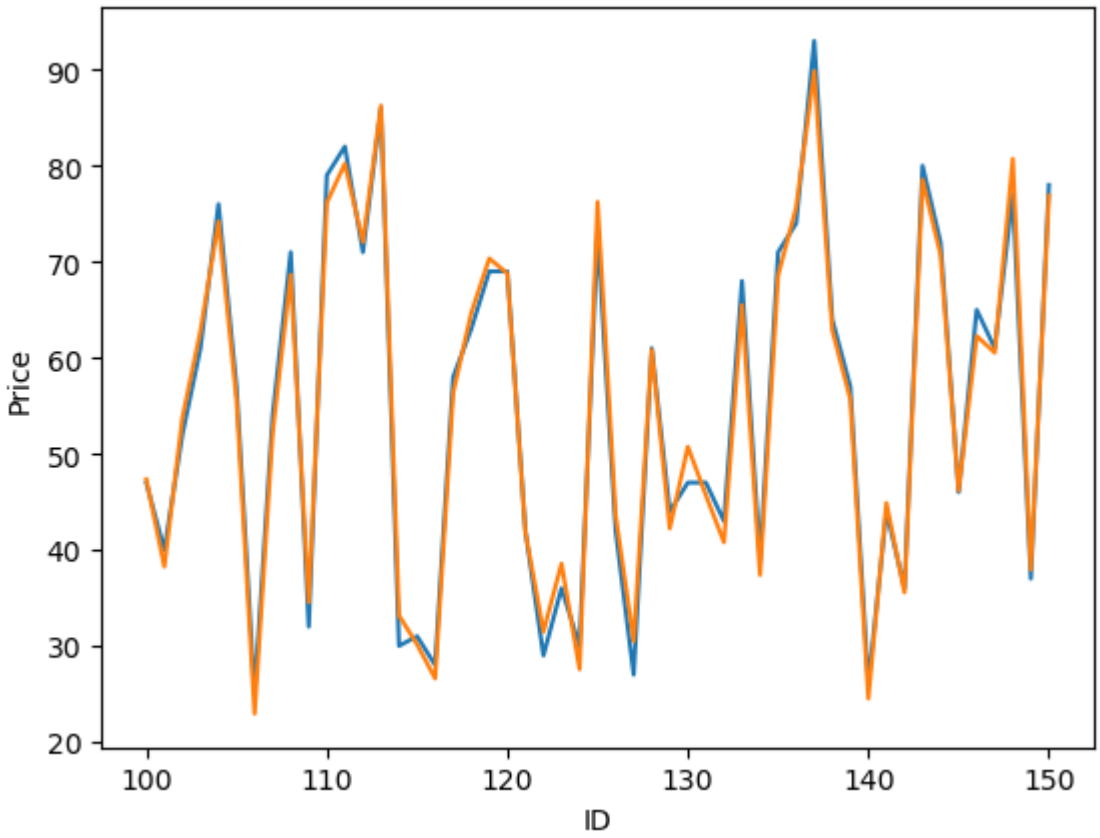
In [29]:
```
Results = pd . DataFrame ( columns = [ 'Price' , 'Predicted' ])
Results [ 'Price' ] = y_test
Results [ 'Predicted' ] = ypred
Results = Results . reset_index ()
Results [ 'ID' ] = Results . index
Results . head ( 15 )
```

Out[29]:

	index	Price	Predicted	ID
0	1977	66.0	61.337528	0
1	3880	77.0	76.503526	1
2	52	75.0	71.549652	2
3	2551	64.0	63.489681	3
4	2246	62.0	63.411925	4
5	270	42.0	45.117783	5
6	601	48.0	47.345758	6
7	2441	74.0	74.799626	7
8	3286	28.0	28.384095	8
9	2967	41.0	45.320545	9
10	4712	42.0	41.904840	10
11	9032	45.0	41.576707	11
12	1787	43.0	43.240778	12
13	1698	29.0	27.695841	13
14	3225	80.0	80.275661	14

In [30]:
```
sns . lineplot ( x = 'ID' , y = 'Price' , data = Results . loc [ 100 : 150
sns . lineplot ( x = 'ID' , y = 'Predicted' , data = Results . loc [ 100 :
plt . plot ()
```

Out[30]: []



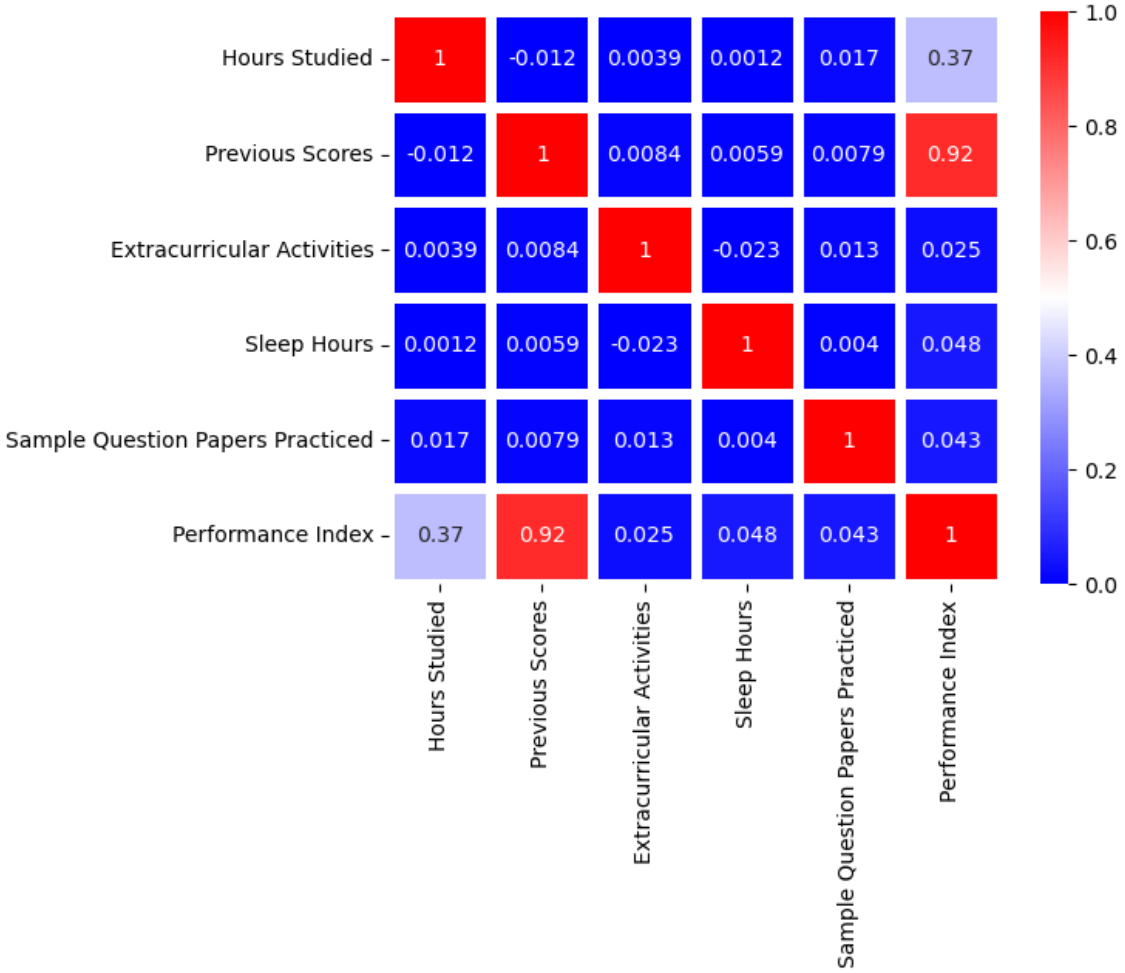
In [31]:
```
cor_mat = data . corr ()
cor_mat
```

Out[31]:

	Hours Studied	Previous Scores	Extracurricular Activities	Sleep Hours	Sample Question Papers Practiced	Performance Index
Hours Studied	1.000000	-0.012390	0.003873	0.001245	0.017463	0.373730
Previous Scores	-0.012390	1.000000	0.008369	0.005944	0.007888	0.915189
Extracurricular Activities	0.003873	0.008369	1.000000	-0.023284	0.013103	0.024525
Sleep Hours	0.001245	0.005944	-0.023284	1.000000	0.003990	0.048106
Sample Question Papers Practiced	0.017463	0.007888	0.013103	0.003990	1.000000	0.043268
Performance Index	0.373730	0.915189	0.024525	0.048106	0.043268	1.000000

In [32]:
```
sns . heatmap ( cor_mat , vmax = 1 , vmin = 0 , annot = True , linewidths =
```

Out[32]: <Axes: >



In [ ]:
```

```