student-performance-prediction-22

September 4, 2025

[1]:
```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt

import warnings
warnings.filterwarnings("ignore")
```

[2]:
```
data = pd.read_csv('/kaggle/input/
 ↪student-performance-multiple-linear-regression/Student_Performance.csv')
```

[3]:
```
# Display Top 5 Rows of the DataFrame
data.head()
```

[3]:
```
   Hours Studied  Previous Scores Extracurricular Activities  Sleep Hours  \
0              7               99                        Yes            9
1              4               82                         No            4
2              8               51                        Yes            7
3              5               52                        Yes            5
4              7               75                         No            8

   Sample Question Papers Practiced  Performance Index
0                                 1               91.0
1                                 2               65.0
2                                 2               45.0
3                                 2               36.0
4                                 5               66.0
```

[5]:
```
data.shape
```

[5]:
```
(10000, 6)
```

[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
```

[7]:
```
data.isnull().sum()
```

[7]:
```
Hours Studied                       0
Previous Scores                     0
Extracurricular Activities          0
Sleep Hours                         0
Sample Question Papers Practiced    0
Performance Index                   0
dtype: int64
```

[10]:
```
# Check For Duplicates
data.duplicated().sum()
```

[10]:
```
127
```

[11]:
```
# Drop duplicates
data.drop_duplicates(keep='first', inplace=True)
```

[12]:
```
data.shape
```

[12]:
```
(9873, 6)
```

[13]:
```
# Quick Statistics
data.describe()
```

[13]:
```
       Hours Studied  Previous Scores  Sleep Hours  \
count    9873.000000      9873.000000  9873.000000
mean        4.992100        69.441102     6.531652
std         2.589081        17.325601     1.697683
min         1.000000        40.000000     4.000000
25%         3.000000        54.000000     5.000000
50%         5.000000        69.000000     7.000000
75%         7.000000        85.000000     8.000000
max         9.000000        99.000000     9.000000

       Sample Question Papers Practiced  Performance Index
count                       9873.000000        9873.000000
mean                           4.583004          55.216651
std                            2.867202          19.208570
```

```
min                              0.000000          10.000000
25%                              2.000000          40.000000
50%                              5.000000          55.000000
75%                              7.000000          70.000000
max                              9.000000         100.000000
```

```
[14]: data.head()
```

```
[14]:    Hours Studied  Previous Scores Extracurricular Activities  Sleep Hours  \
      0              7               99                        Yes            9
      1              4               82                         No            4
      2              8               51                        Yes            7
      3              5               52                        Yes            5
      4              7               75                         No            8

         Sample Question Papers Practiced  Performance Index
      0                                 1               91.0
      1                                 2               65.0
      2                                 2               45.0
      3                                 2               36.0
      4                                 5               66.0
```

```
[15]: data['Extracurricular Activities'].value_counts()
```

```
[15]: Extracurricular Activities
      No     4986
      Yes    4887
      Name: count, dtype: int64
```

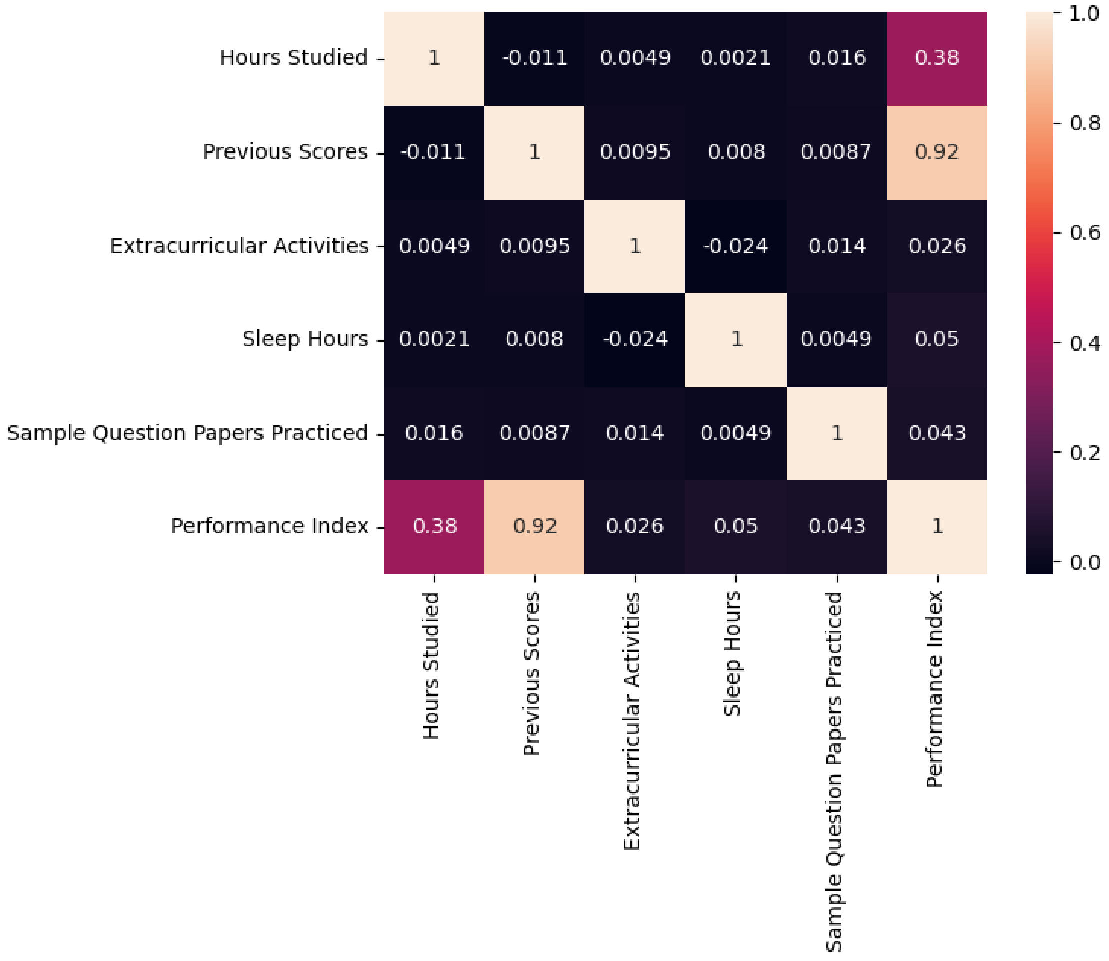
```
[16]: data['Extracurricular Activities'] = data['Extracurricular Activities'].
       ↪map({'Yes':1, 'No':0})
```

```
[17]: data.head()
```

```
[17]:    Hours Studied  Previous Scores  Extracurricular Activities  Sleep Hours  \
      0              7               99                           1            9
      1              4               82                           0            4
      2              8               51                           1            7
      3              5               52                           1            5
      4              7               75                           0            8

         Sample Question Papers Practiced  Performance Index
      0                                 1               91.0
      1                                 2               65.0
      2                                 2               45.0
      3                                 2               36.0
      4                                 5               66.0
```

[18]:
```
sns.heatmap(data.corr(), annot=True )
```

[18]:
```
<Axes: >
```



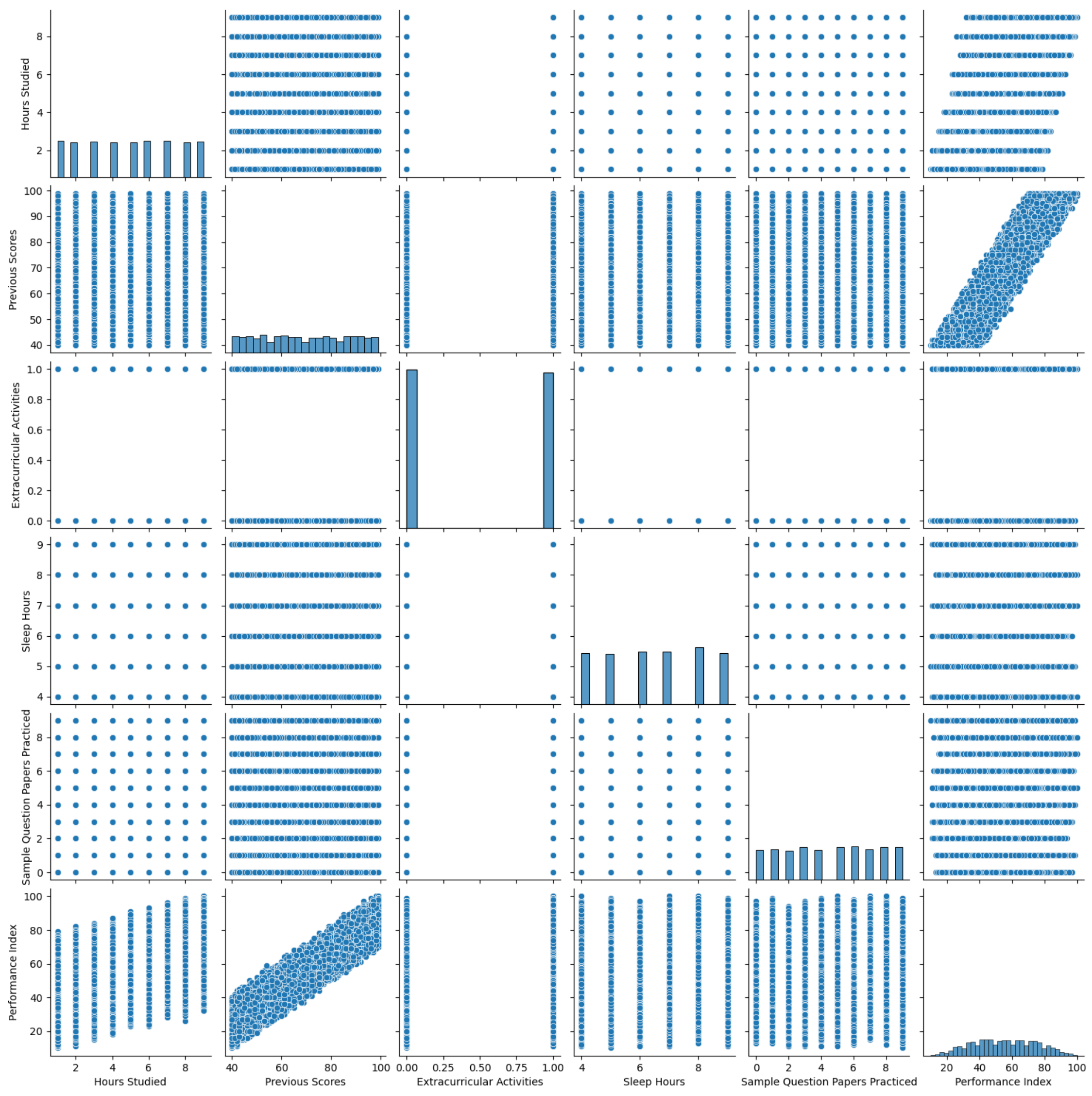
[19]:
```
data.corr()
```

[19]:
```
                                  Hours Studied  Previous Scores  \
Hours Studied                          1.000000        -0.010676
Previous Scores                       -0.010676         1.000000
Extracurricular Activities             0.004899         0.009534
Sleep Hours                            0.002131         0.007975
Sample Question Papers Practiced       0.015740         0.008719
Performance Index                      0.375332         0.915135

                                  Extracurricular Activities  Sleep Hours  \
Hours Studied                                       0.004899     0.002131
Previous Scores                                     0.009534     0.007975
```

```
Extracurricular Activities                     1.000000    -0.024008
Sleep Hours                                   -0.024008     1.000000
Sample Question Papers Practiced               0.013839     0.004907
Performance Index                              0.026075     0.050352

                                  Sample Question Papers Practiced  \
Hours Studied                                             0.015740
Previous Scores                                           0.008719
Extracurricular Activities                                0.013839
Sleep Hours                                               0.004907
Sample Question Papers Practiced                          1.000000
Performance Index                                         0.043436

                                  Performance Index
Hours Studied                              0.375332
Previous Scores                            0.915135
Extracurricular Activities                 0.026075
Sleep Hours                                0.050352
Sample Question Papers Practiced           0.043436
Performance Index                          1.000000
```

[20]:
```
sns.pairplot(data)
```

[20]:
```
<seaborn.axisgrid.PairGrid at 0x7d092317ead0>
```

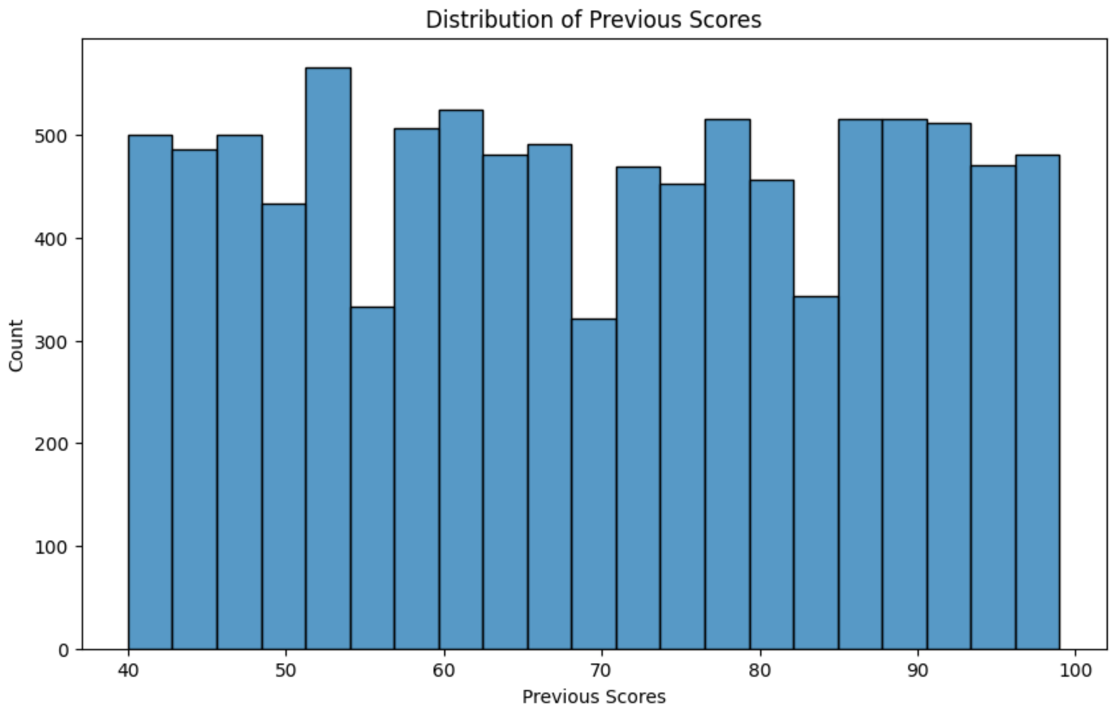


```
[21]: data.head()
```

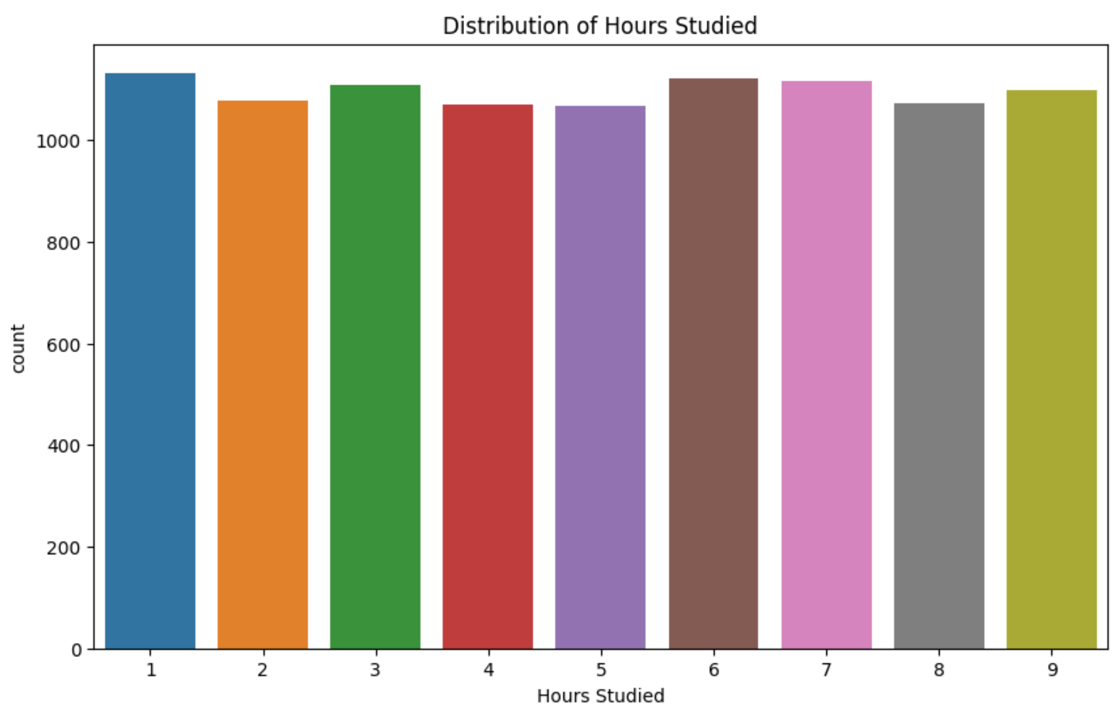
```
[21]:    Hours Studied  Previous Scores  Extracurricular Activities  Sleep Hours  \
      0              7               99                           1            9
      1              4               82                           0            4
      2              8               51                           1            7
      3              5               52                           1            5
      4              7               75                           0            8

         Sample Question Papers Practiced  Performance Index
      0                                 1               91.0
      1                                 2               65.0
      2                                 2               45.0
```

```
3                                   2               36.0
4                                   5               66.0
```

[23]:
```
plt.figure(figsize=(10,6))
sns.histplot(x='Previous Scores',data=data)
plt.title('Distribution of Previous Scores')
plt.show()
```



[24]:
```
plt.figure(figsize=(10,6))
sns.countplot(x='Hours Studied',data=data)
plt.title('Distribution of Hours Studied')
plt.show()
```



[25]:
```
X = data.drop('Performance Index', axis=1)
y = data['Performance Index']
```

[26]:
```
from sklearn.model_selection import train_test_split
```

[27]:
```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2,␣
 ↪random_state=42)
```

[28]:
```
from sklearn.linear_model import LinearRegression
```

[29]:
```
lr = LinearRegression()
lr.fit(X_train, y_train)
```

[29]:
```
LinearRegression()
```

[30]:
```
y_pred = lr.predict(X_test)
```

[31]:
```
from sklearn.metrics import r2_score
```

[32]:
```
print(r2_score(y_test, y_pred))
```

```
0.9884301209927054
```

[33]:
```
before_fs_r2 = r2_score(y_test, y_pred)
```

```
[35]: before_fs_r2
```

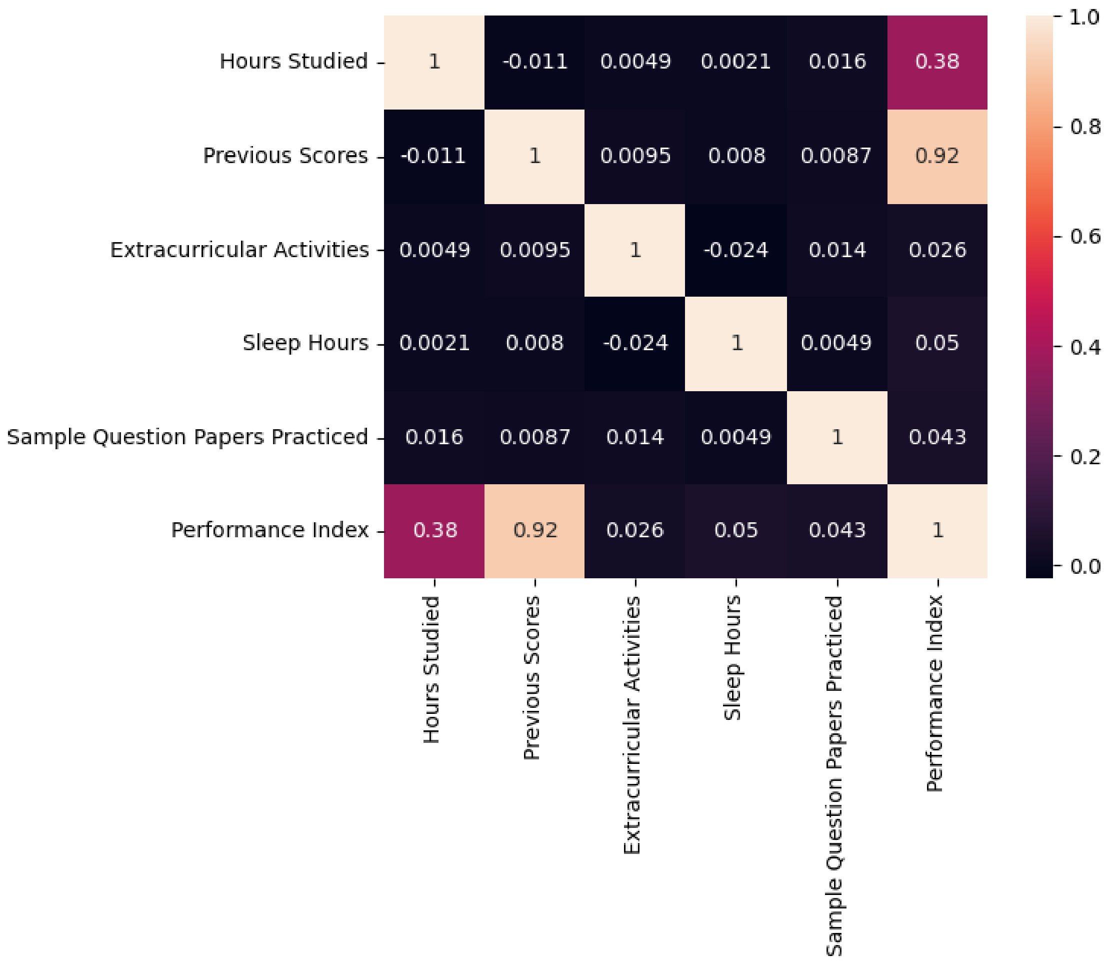
```
[35]: 0.9884301209927054
```

```
[36]: data.head()
```

```
[36]:    Hours Studied  Previous Scores  Extracurricular Activities  Sleep Hours  \
      0              7               99                           1            9
      1              4               82                           0            4
      2              8               51                           1            7
      3              5               52                           1            5
      4              7               75                           0            8

         Sample Question Papers Practiced  Performance Index
      0                                 1               91.0
      1                                 2               65.0
      2                                 2               45.0
      3                                 2               36.0
      4                                 5               66.0
```

```
[38]: sns.heatmap(data.corr(), annot=True)
```

```
[38]: <Axes: >
```



[37]:
```
df = data.drop(['Extracurricular Activities','Sleep Hours', 'Sample Question␣
↪Papers Practiced'], axis=1)
```

[39]:
```
df
```

[39]:
```
      Hours Studied  Previous Scores  Performance Index
0                 7               99               91.0
1                 4               82               65.0
2                 8               51               45.0
3                 5               52               36.0
4                 7               75               66.0
…               …              …                 …
9995              1               49               23.0
9996              7               64               58.0
9997              6               83               74.0
9998              9               97               95.0
9999              7               74               64.0
```

```
[9873 rows x 3 columns]
```

```
[40]: X_df = df.drop('Performance Index', axis=1)
      y_df = df['Performance Index']
```

```
[41]: X_train, X_test, y_train, y_test = train_test_split(X_df, y_df, test_size=0.2,␣
       ↪random_state=42)
```

```
[42]: lr_f = LinearRegression()
      lr_f.fit(X_train, y_train)
```

```
[42]: LinearRegression()
```

```
[43]: yf_pred = lr_f.predict(X_test)
```

```
[44]: after_fs_r2 = r2_score(y_test, yf_pred)
```

```
[45]: print( after_fs_r2)
```

```
0.9850233951895029
```

```
[46]: Result = pd.Series({'Before Feature Selection':before_fs_r2,'After Feature␣
       ↪Selection':after_fs_r2 })
```

```
[47]: Result
```

```
[47]: Before Feature Selection    0.988430
      After Feature Selection     0.985023
      dtype: float64
```

```
[50]: from sklearn.preprocessing import PolynomialFeatures
      from sklearn.metrics import r2_score


      degree = 2
      poly = PolynomialFeatures(degree=degree)
      X_train_poly = poly.fit_transform(X_train)
      X_test_poly = poly.transform(X_test)


      y_pred = poly_model.predict(X_test_poly)
      print("R² Score:", r2_score(y_test, y_pred))
```

```
R² Score: 0.9850343841670226
```

```
[ ]:
```