project-1

February 21, 2024

```
[ ]: LOGISTIC REGRESSION BASED ON PROJECTS
```

```
[ ]: Students Performance based on Logistic Regression
```

```
[96]: from sklearn.linear_model import LogisticRegression
      from sklearn.metrics import accuracy_score
      from sklearn.model_selection import train_test_split
```

```
[5]: import pandas as pd
```

```
[97]: df=pd.read_csv(r"C:\Users\micro\Downloads\LR_Student_Performance.csv")
      df
```

```
[97]:       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
      …              …               …                          …           …
      9995              1               49                        Yes            4
      9996              7               64                        Yes            8
      9997              6               83                        Yes            8
      9998              9               97                        Yes            7
      9999              7               74                         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
      …                                 …                …
      9995                                 2               23.0
      9996                                 5               58.0
      9997                                 5               74.0
      9998                                 0               95.0
```

```
9999                                  1               64.0

[10000 rows x 6 columns]
```

[40]:
```
x=df[['Hours Studied']]
y=df['Previous Scores']
from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.
 ↪4,random_state=100)
from sklearn.linear_model import LogisticRegression
model=LogisticRegression()
model
```

[40]:
```
LogisticRegression()
```

[41]:
```
x_train,x_test,y_train,y_test=train_test_split(x,y,test_size=0.
 ↪2,random_state=101)
```

[42]:
```
model.fit(x_train,y_train)
```

[42]:
```
LogisticRegression()
```

[43]:
```
y_pred=model.predict(x_test)
y_pred
```

[43]:
```
array([54, 54, 86, …, 54, 54, 86], dtype=int64)
```

[44]:
```
y_test
```

[44]:
```
6676    55
6421    98
9834    57
8492    71
9982    51
        ..
4441    65
4166    65
2567    62
8527    64
406     42
Name: Previous Scores, Length: 2000, dtype: int64
```

[45]:
```
from sklearn.metrics import accuracy_score
```

[47]:
```
import numpy as np
```

```
[98]: acc=accuracy_score(y_test,np.round(y_pred))
      acc
```

```
[98]: 0.015
```

```
[99]: inputdata=[[17]]
      prediction=model.predict(inputdata)
      prediction
```

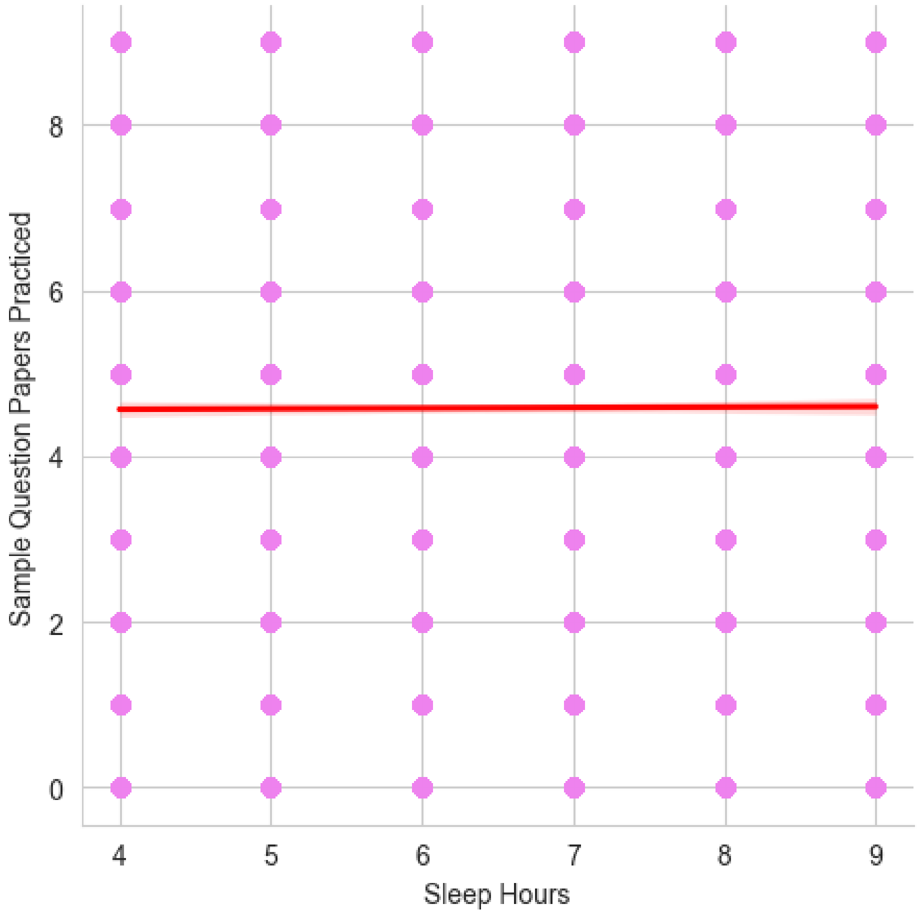
```
C:\Users\micro\AppData\Local\Programs\Python\Python312\Lib\site-
packages\sklearn\base.py:493: UserWarning: X does not have valid feature names,
but LogisticRegression was fitted with feature names
  warnings.warn(
```

```
[99]: array([56], dtype=int64)
```

```
[100]: from sklearn.metrics import mean_squared_error
       mse=mean_squared_error(y_test,y_pred)
       mse
```

```
[100]: 557.3895
```

```
[70]: import seaborn as sns
      import matplotlib.pyplot as plt
      sns.lmplot(x="Sleep Hours",y="Sample Question Papers␣
        ↪Practiced",data=df,scatter_kws={"color":'violet' },line_kws={'color':"red"})
      sns.set_style('whitegrid')
      ax=plt.gca()
      plt.gca()
      plt.gca().set_facecolor('white')
```



[55]:
```
#To check duplicate values
duplicate_rows=df.duplicated()
df[duplicate_rows].sum()
```

[55]:
```
Hours Studied
642
Previous Scores
8865
Extracurricular Activities
NoYesNoYesYesNoNoNoNoNoYesNoYesNoNoNoYesYesYes…
Sleep Hours
819
Sample Question Papers Practiced
585
Performance Index
7094.0
dtype: object
```

[57]:
```
x=df.drop("Hours Studied",axis=1)
x
```

[57]:
```
      Previous Scores Extracurricular Activities  Sleep Hours  \
0                  99                        Yes            9
1                  82                         No            4
2                  51                        Yes            7
3                  52                        Yes            5
4                  75                         No            8
…                 …                         …           …
9995               49                        Yes            4
9996               64                        Yes            8
9997               83                        Yes            8
9998               97                        Yes            7
9999               74                         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
…                                 …                 …
9995                                 2               23.0
9996                                 5               58.0
9997                                 5               74.0
9998                                 0               95.0
9999                                 1               64.0

[10000 rows x 5 columns]
```

[58]:
```
x.shape
```

[58]:
```
(10000, 5)
```

[101]:
```
y=df.drop("Sleep Hours",axis=1)
y
```

[101]:
```
      Hours Studied  Previous Scores Extracurricular Activities  \
0                 7               99                        Yes
1                 4               82                         No
2                 8               51                        Yes
3                 5               52                        Yes
4                 7               75                         No
…              …              …                         …
9995              1               49                        Yes
9996              7               64                        Yes
```

```
9997              6              83                          Yes
9998              9              97                          Yes
9999              7              74                           No

      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
…                                  …                 …
9995                                 2               23.0
9996                                 5               58.0
9997                                 5               74.0
9998                                 0               95.0
9999                                 1               64.0

[10000 rows x 5 columns]
```

[64]:
```
print("Before dropping duplicate:",df.shape)
df.drop_duplicates()
print("After dropping duplicate:",df.shape)
```

```
Before dropping duplicate: (10000, 6)
After dropping duplicate: (10000, 6)
```

[66]:
```
response=df["Extracurricular Activities"]
response.dtype
```

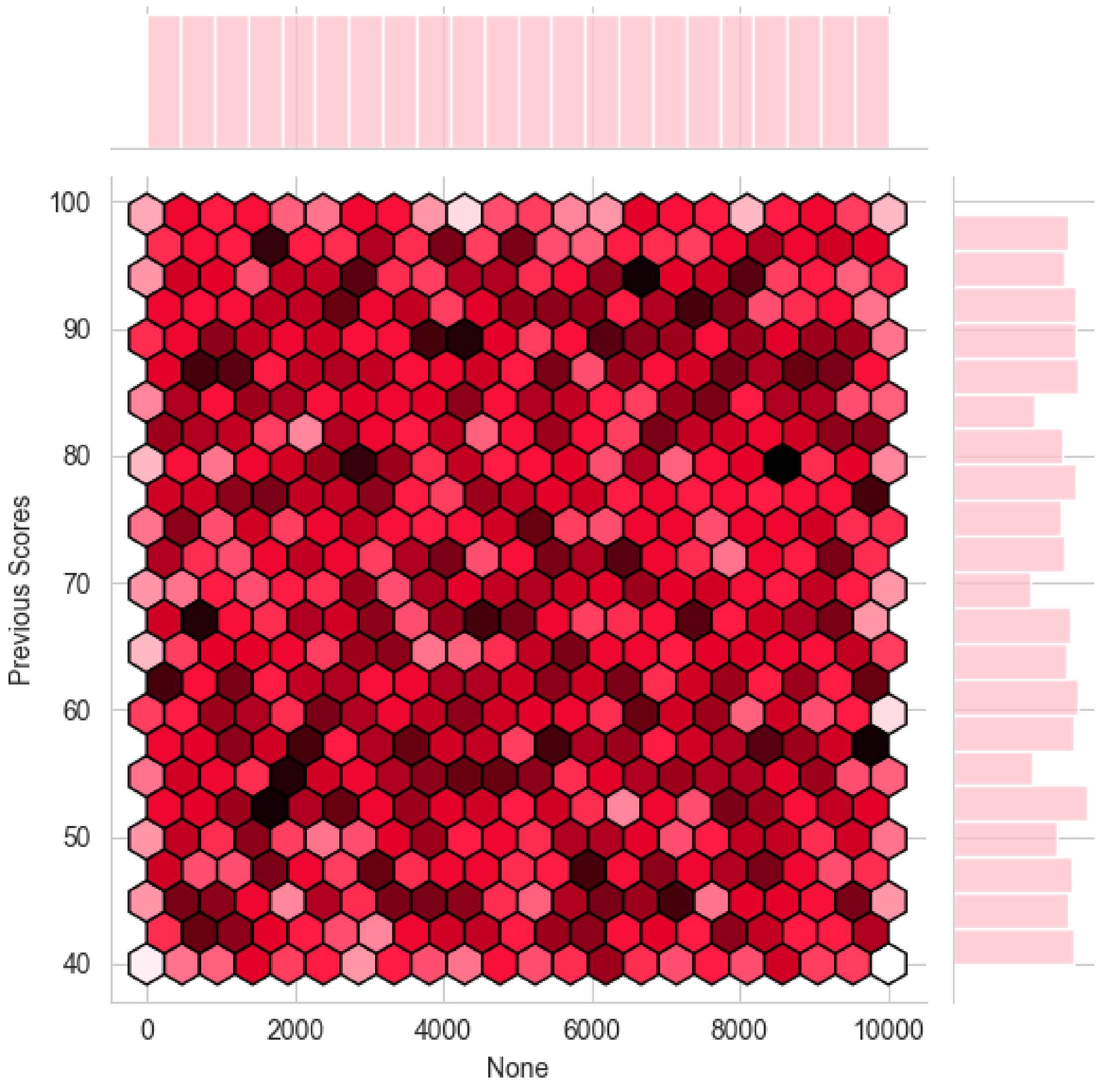
[66]:
```
dtype('O')
```

[68]:
```
response=df["Performance Index"]
response.dtype
```

[68]:
```
dtype('float64')
```

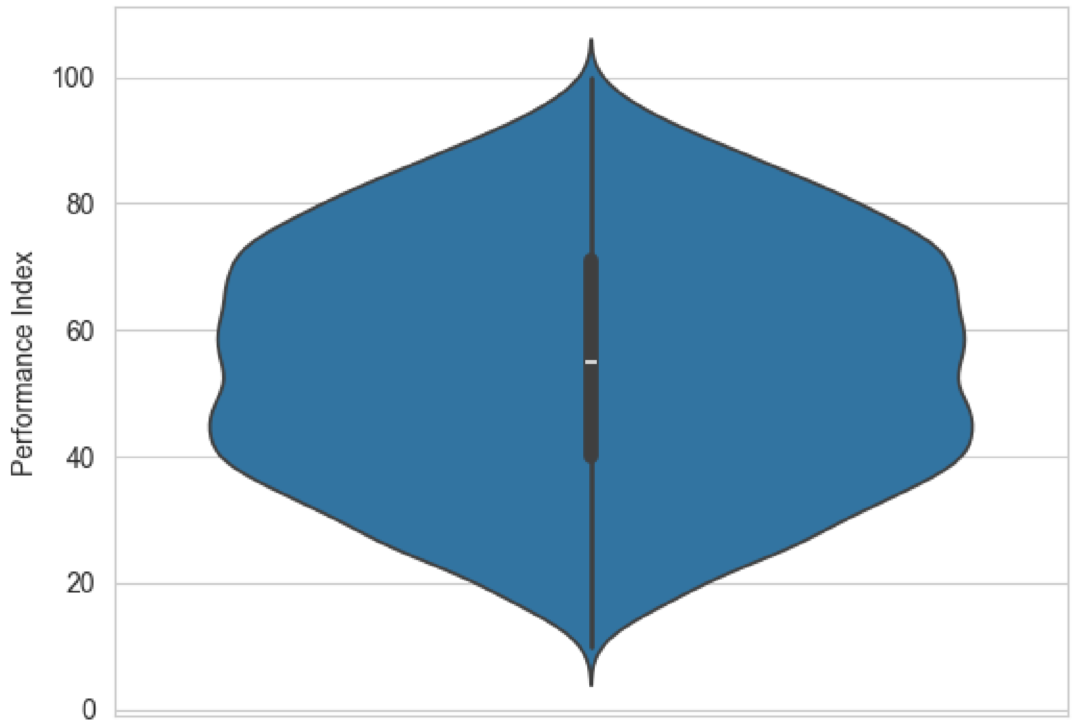
[75]:
```
sns.jointplot(x=response.index,y="Previous␣
 ↪Scores",data=df,kind='hex',color='pink',edgecolor='black')
```

[75]:
```
<seaborn.axisgrid.JointGrid at 0x26d970a52e0>
```



[76]:
```
sns.violinplot(response)
```

[76]:
```
<Axes: ylabel='Performance Index'>
```



```
[91]: ma=(df["Performance Index"]).max()
      ma
```

```
[91]: 100.0
```

```
[92]: mi=(df["Performance Index"]).min()
      mi
```

```
[92]: 10.0
```

```
[93]: (df['Performance Index']==mi).sum()
```

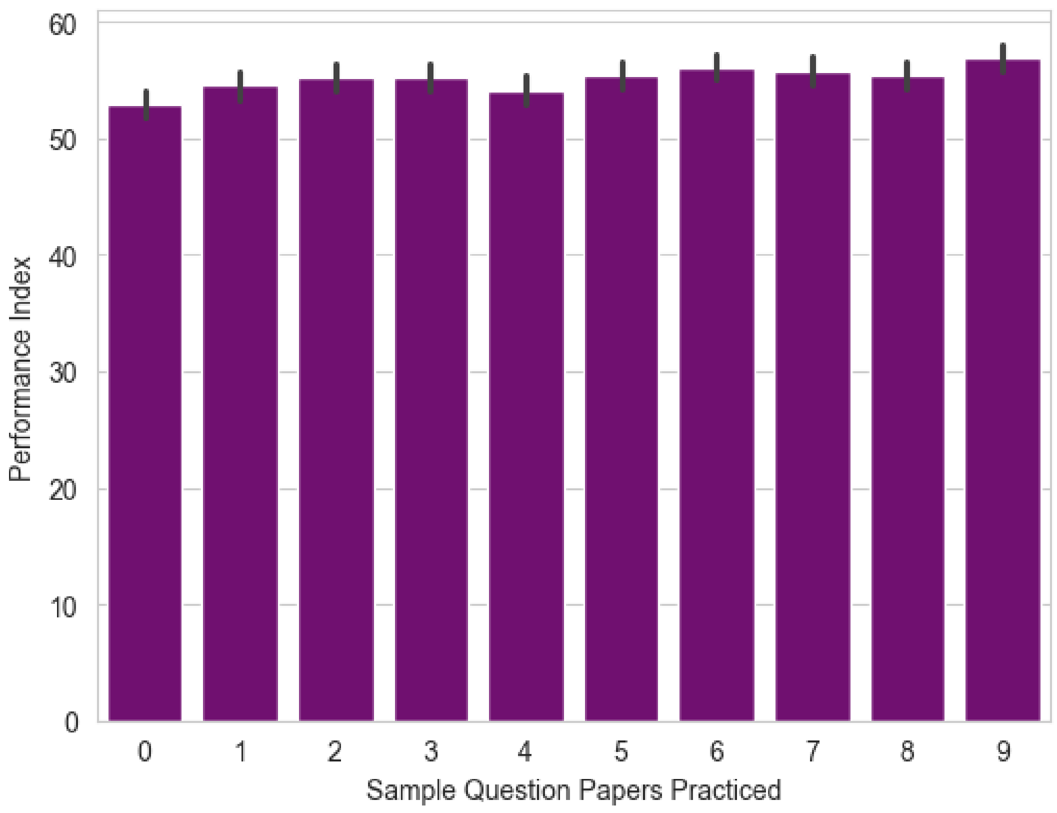
```
[93]: 1
```

```
[94]: (df['Hours Studied']).unique()
```

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

```
[81]: import seaborn as sns
      sns.barplot(y="Performance Index",x="Sample Question Papers␣
       ↪Practiced",data=df,color='purple')
```

```
[81]: <Axes: xlabel='Sample Question Papers Practiced', ylabel='Performance Index'>
```



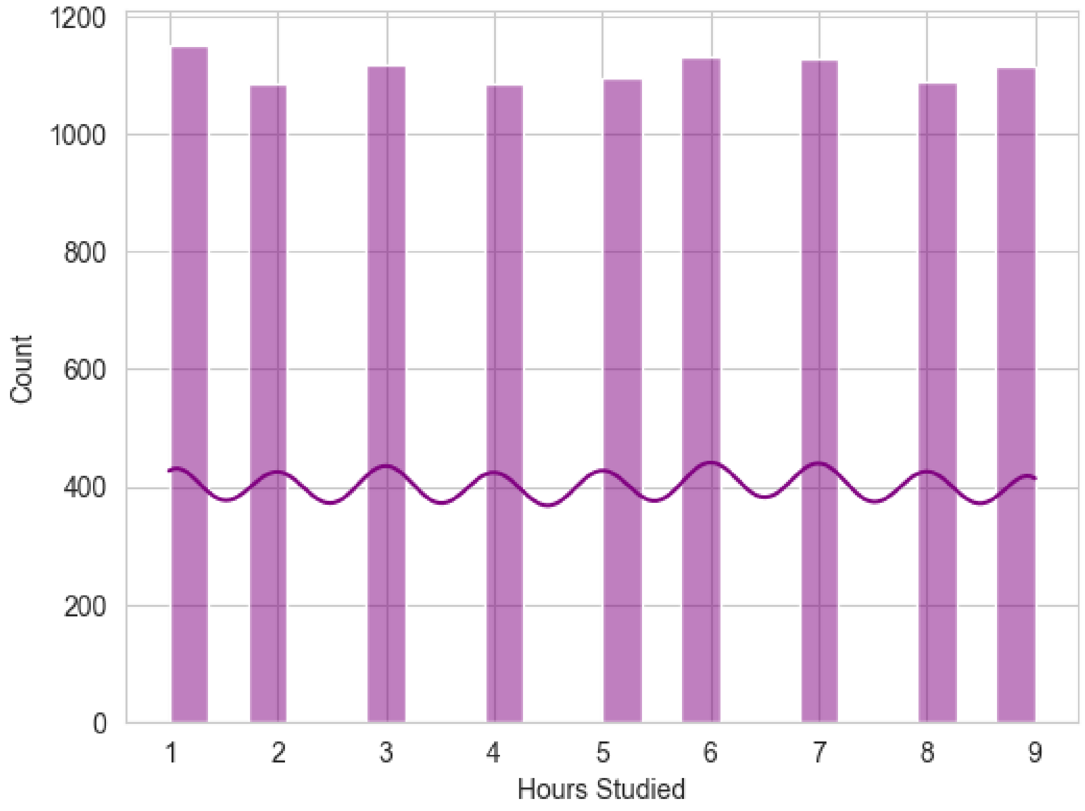
```
[25]: df.head()
```

```
[25]:    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
```

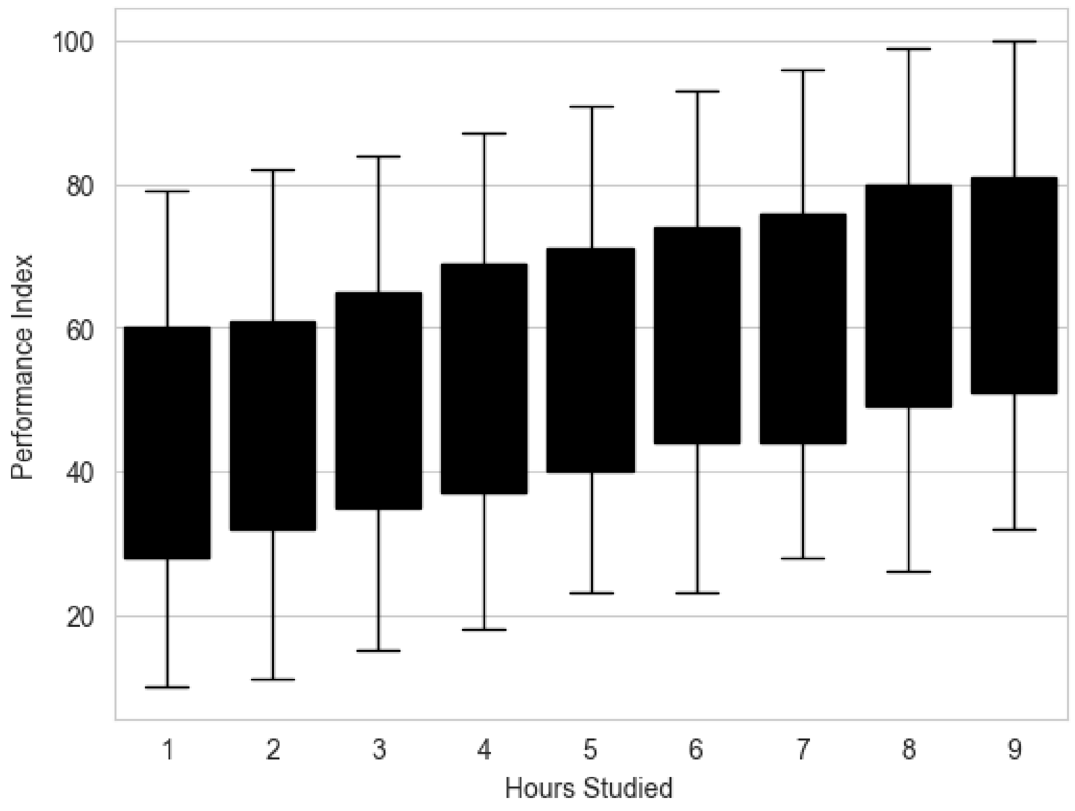
```
[83]: import seaborn as sns
      x=df['Hours Studied']
      sns.histplot(x,color='purple',kde=True)
```

```
[83]: <Axes: xlabel='Hours Studied', ylabel='Count'>
```



```
[88]: sns.boxplot(x=df['Hours Studied'],y=df['Performance Index'],color='black')
```

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



```
[89]: x=df[['Hours Studied', 'Previous Scores', 'Extracurricular Activities',
             'Sleep Hours', 'Sample Question Papers Practiced']]
      y=df['Performance Index']
      df.head()
```

```
[89]:    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
```

```
[95]: df['Extracurricular Activities']=df['Extracurricular Activities'].apply(lambda␣
       ↪x: 1 if x=="Yes"else 0)
```

```
df.head()
```

```
[95]:    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
```

```
[28]: df.info
```

```
[28]: <bound method DataFrame.info of       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
      …               …               …                         …          …
      9995              1               49                        Yes            4
      9996              7               64                        Yes            8
      9997              6               83                        Yes            8
      9998              9               97                        Yes            7
      9999              7               74                         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
      …                                  …               …
      9995                                 2               23.0
      9996                                 5               58.0
      9997                                 5               74.0
      9998                                 0               95.0
      9999                                 1               64.0

      [10000 rows x 6 columns]>
```

[33]:
```
df.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
```

[34]:
```
df.isna()
```

[34]:
```
      Hours Studied  Previous Scores  Extracurricular Activities  Sleep Hours  \
0             False            False                       False        False
1             False            False                       False        False
2             False            False                       False        False
3             False            False                       False        False
4             False            False                       False        False
…                …              …                         …           …
9995          False            False                       False        False
9996          False            False                       False        False
9997          False            False                       False        False
9998          False            False                       False        False
9999          False            False                       False        False

      Sample Question Papers Practiced  Performance Index
0                                False              False
1                                False              False
2                                False              False
3                                False              False
4                                False              False
…                                 …               …
9995                             False              False
9996                             False              False
9997                             False              False
9998                             False              False
9999                             False              False

[10000 rows x 6 columns]
```

[36]:
```
df.isna().sum()
```

[36]:
```
Hours Studied                       0
Previous Scores                     0
Extracurricular Activities          0
Sleep Hours                         0
Sample Question Papers Practiced    0
Performance Index                   0
dtype: int64
```

[37]:
```
type(df)
```

[37]:
```
pandas.core.frame.DataFrame
```

[14]:
```
# To train the algorithm
clf.fit(x_train,y_train)
```

[14]:
```
LogisticRegression()
```

[15]:
```
y_pred=clf.predict(x_test)
y_pred
```

[15]:
```
array([0, 0, 0, 2, 1, 2, 1, 1, 2, 0, 2, 0, 0, 2, 2, 1, 1, 1, 0, 2, 1, 0,
       1, 1, 1, 1, 1, 2, 0, 0])
```

[ ]:
```

```