

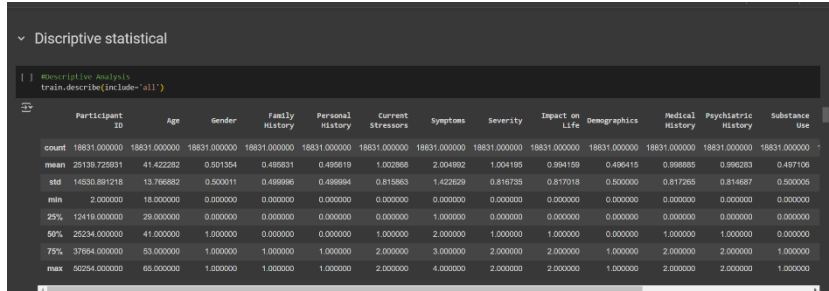
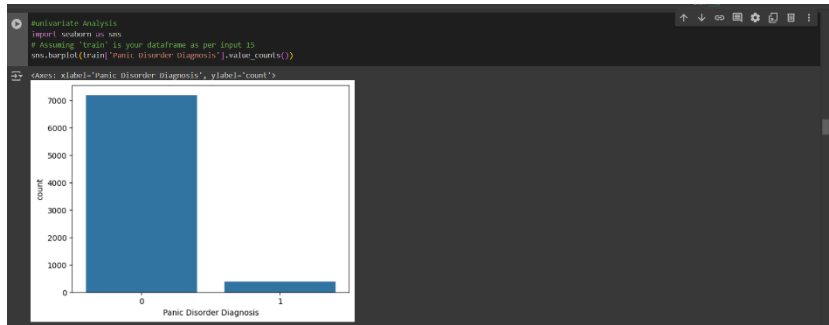
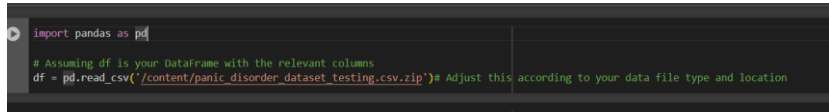
Data Collection and Preprocessing Phase

Date	15 JULY 2024
Team ID	739810
Project Title	Panic Disorder Detection
Maximum Marks	6 Marks

Data Exploration and Preprocessing Template

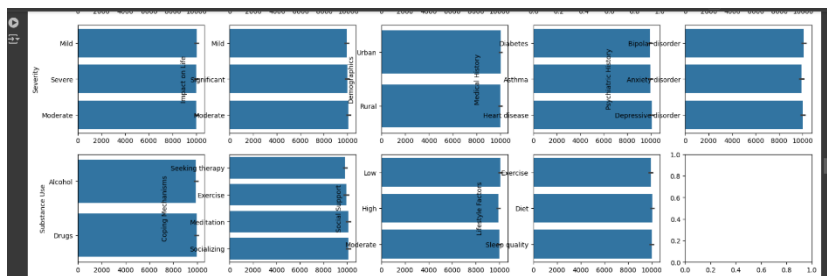
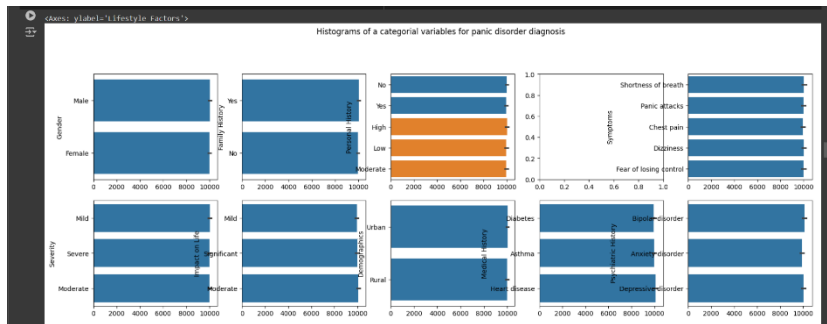
Identifies data sources, assesses quality issues like missing values and duplicates, and implements resolution plans to ensure accurate and reliable analysis.

Section	Description
Data Overview	<p>#Structure of the data: -</p> <pre>[] #Handling the missing values print('Train data shape:', train.shape) print('Test data shape:', test.shape)</pre> <p>⇒ Train data shape: (20000, 17) Test data shape: (20000, 17)</p> <p>#Descriptive Statistical: Descriptive analysis is to study the basic features of data with the statistical process. Here pandas have a worthy function called describe. With this describe function we can understand the unique, top and frequent values of categorical features. And we can find mean, std, min, max and percentile values of continuous features.</p>

	
Univariate Analysis	<p>Univariate Analysis:</p> <p>In simple words, univariate analysis is understanding the data with single feature. Here we have displayed two different graphs such as Histplot and countplot.</p> <p>Seaborn package provides a wonderful function histplot. With the help of histplot, we can find the distribution of the feature. To make multiple graphs in a single plot, we use subplot. First let's check if the data is balanced or not.</p> 
Bivariate Analysis	<p>#Bivariate Analysis:-</p> 

```
#Univariate Analysis
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns

fig, axes = plt.subplots(5, 2, figsize=(20, 10))
fig.suptitle('Histograms of a categorical variables for panic disorder diagnosis')
sns.barplot(df['Gender'], ax=axes[0,0])
sns.barplot(df['Family History'], ax=axes[0,1])
sns.barplot(df['Personal History'], ax=axes[1,0])
sns.barplot(df['Current Stressors'], ax=axes[1,1])
sns.barplot(df['Symptoms'], ax=axes[2,0])
sns.barplot(df['Severity'], ax=axes[2,1])
sns.barplot(df['Impact on Life'], ax=axes[3,0])
sns.barplot(df['Demographics'], ax=axes[3,1])
sns.barplot(df['Medical History'], ax=axes[4,0])
sns.barplot(df['Psychiatric History'], ax=axes[4,1])
sns.barplot(df['Substance Use'], ax=axes[5,0])
sns.barplot(df['Coping Mechanisms'], ax=axes[5,1])
sns.barplot(df['Social Support'], ax=axes[6,0])
sns.barplot(df['Therapeutic Factors'], ax=axes[6,1])
```



From the plot we came to know,

- Both the genders are diagnosed equally with panic disorder.
- The current stressors of the subjects are mostly high with a sleep deprived lifestyle.
- Panic disorder plays an important role in one's life and is severely affected to most of the subjects.
- The symptoms of the panic disorder are mainly 5 out of which Panic attacks are mostly observed.
- The social support provided for these subjects is also low and the coping mechanisms include seeking therapy by large number of the affected.

Visual Analysis

Visual analysis is the process of using visual representations, such as charts, plots, and graphs, to explore and understand data. It is a way to quickly identify patterns, trends, and outliers in the data, which can help to gain insights and make informed decisions.

#In this no plots are available

Outliers and Anomalies

-

Data Preprocessing Code Screenshots

Loading Data

#Loading the data

```
[ ] #Load the dataset
train = pd.read_csv("../content/panic_disorder_dataset_training.csv")
train.head()
```

	Participant ID	Age	Gender	Family History	Personal History	Current Stressors	Symptoms	Severity	Impact on Life	Demographics	Medical History	Psychiatric History	Substance Use	Coping Mechanisms	Social Support	Lifestyle Factors	Panic Disorder Diagnosis
0	1	38	Male	No	Yes	Moderate	Shortness of breath	Mild	Mild	Rural	Diabetes	Bipolar disorder	NaN	Socializing	High	Sleep quality	0.0
1	2	51	Male	No	No	High	Panic attacks	Mild	Mild	Urban	Asthma	Anxiety disorder	Drugs	Exercise	High	Sleep quality	0.0
2	3	32	Female	Yes	No	High	Panic attacks	Mild	Significant	Urban	Diabetes	Depressive disorder	NaN	Seeking therapy	Moderate	Exercise	0.0
3	4	64	Female	No	No	Moderate	Chest pain	Moderate	Moderate	Rural	Diabetes	NaN	NaN	Meditation	High	Exercise	0.0
4	5	31	Male	Yes	No	Moderate	Panic attacks	Mild	Moderate	Rural	Asthma	NaN	Drugs	Seeking therapy	Low	Sleep quality	0.0

```
test = pd.read_csv("../content/panic_disorder_dataset_testing.csv")
test.head()
```

	Participant ID	Age	Gender	Family History	Personal History	Current Stressors	Symptoms	Severity	Impact on Life	Demographics	Medical History	Psychiatric History	Substance Use	Coping Mechanisms	Social Support	Lifestyle Factors	Panic Disorder Diagnosis
0	1	41	Male	Yes	No	High	Shortness of breath	Mild	Mild	Urban	Diabetes	Bipolar disorder	Alcohol	Seeking therapy	Low	Exercise	0
1	2	20	Female	Yes	No	Low	Shortness of breath	Mild	Significant	Urban	Asthma	Anxiety disorder	Drugs	Exercise	High	Diet	0
2	3	32	Male	Yes	Yes	High	Panic attacks	Severe	Mild	Rural	Heart disease	Bipolar disorder	Drugs	Meditation	Moderate	Exercise	0
3	4	41	Female	Yes	Yes	Moderate	Shortness of breath	Moderate	Significant	Urban	Heart disease	Anxiety disorder	NaN	Exercise	High	Sleep quality	0
4	5	36	Female	Yes	No	High	Chest pain	Severe	Significant	Rural	Asthma	Depressive disorder	NaN	Seeking therapy	Low	Exercise	0

Handling Missing Data

```
#Handling the missing values
print('Train data shape:', train.shape)
print('Test data shape:', test.shape)
```

Train data shape: (20000, 17)
Test data shape: (20000, 17)

```
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20000 entries, 0 to 19999
Data columns (total 17 columns):
 #   Column                Non-Null Count  Dtype
---  -
 0   Participant ID         20000 non-null  int64
 1   Age                   20000 non-null  int64
 2   Gender                20000 non-null  object
 3   Family History         20000 non-null  object
 4   Personal History       20000 non-null  object
 5   Current Stressors      20000 non-null  object
 6   Symptoms               20000 non-null  object
 7   Severity               20000 non-null  object
 8   Impact on Life         20000 non-null  object
 9   Demographics           20000 non-null  object
10   Medical History        14999 non-null  object
11   Psychiatric History    17011 non-null  object
12   Substance Use          13183 non-null  object
13   Coping Mechanisms      20000 non-null  object
14   Social Support         20000 non-null  object
15   Lifestyle Factors      20000 non-null  object
16   Panic Disorder Diagnosis 20000 non-null  int64
dtypes: int64(1), object(14)
memory usage: 2.6+ MB
```

	 <p>For checking the null values, <code>. isnull()</code> function is used. To sum those null values we use <code>. sum()</code> function. From the below image we found that there are no null values present in our dataset. So we can skip handling the missing values step.</p>
Data Transformation	-
Feature Engineering	-
Save Processed Data	-