3.Data Collection and Preprocessing Phase

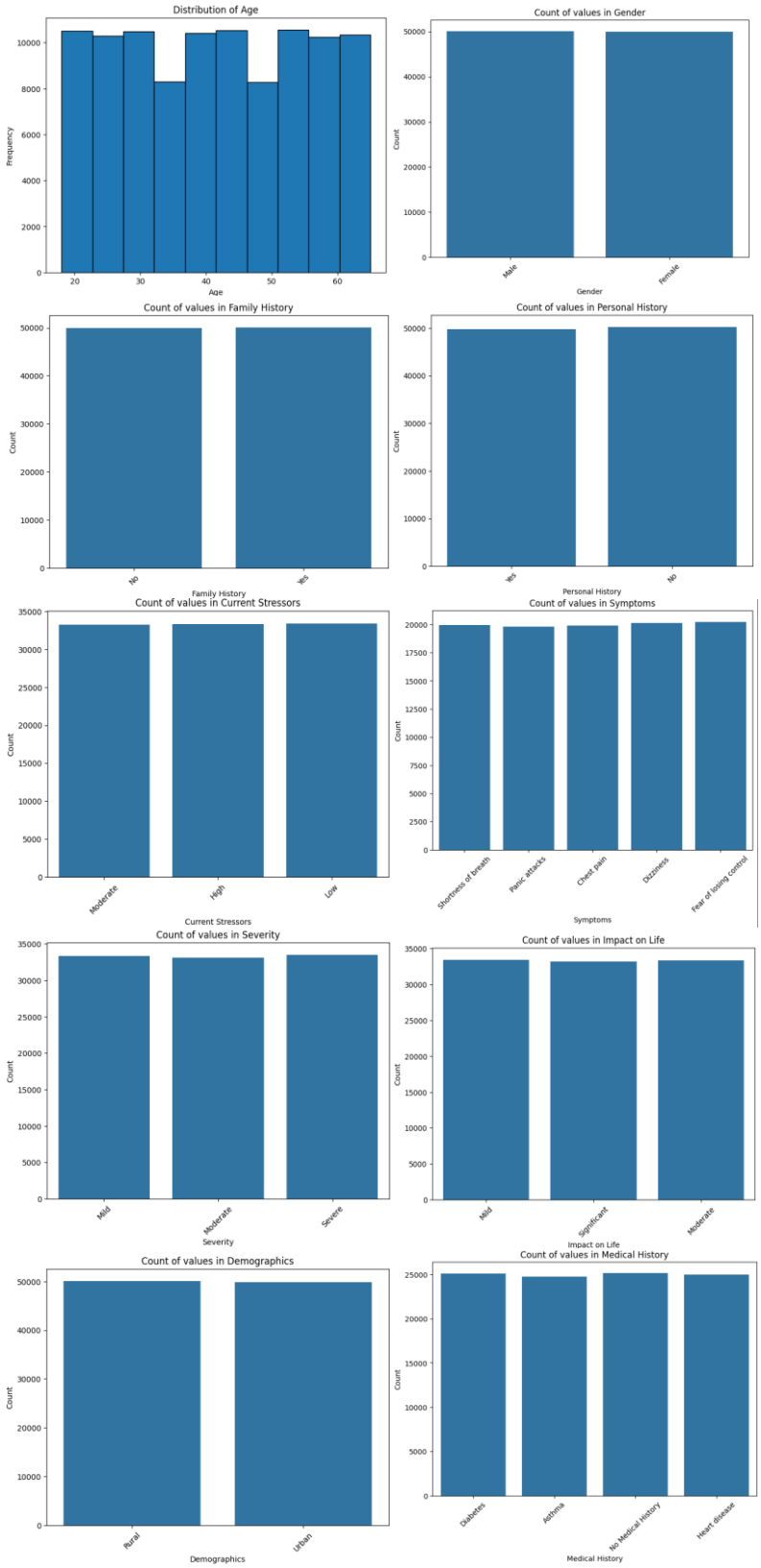
Date	06 July 2024
Team ID	SWTID1720017249
Project Title	Panic Disorder Detection
Maximum Marks	6 Marks

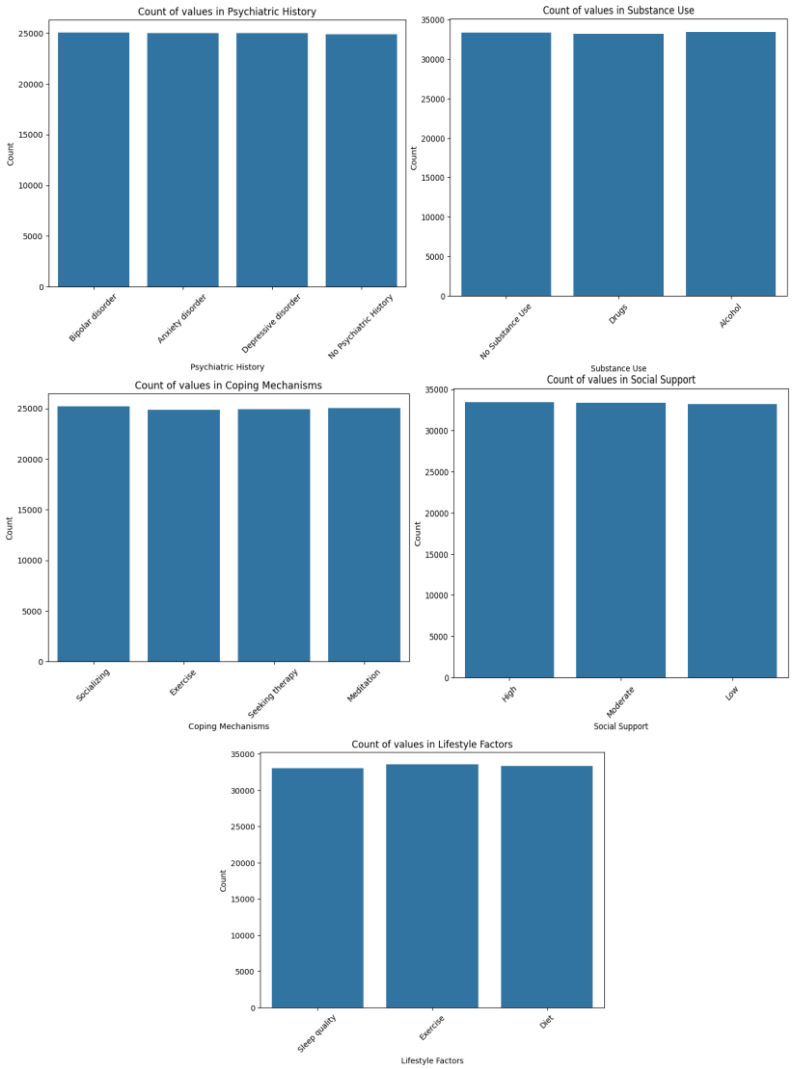
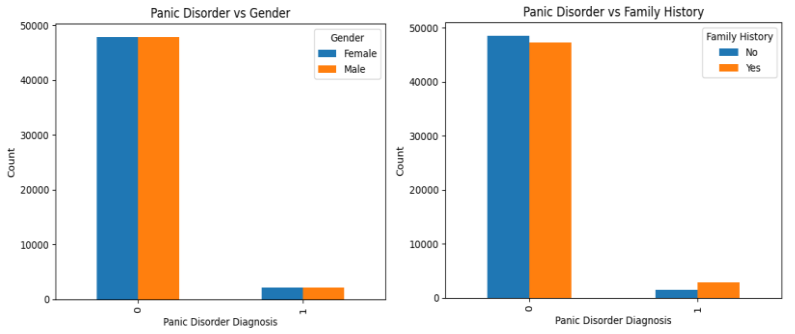
3.3 Data Exploration and Preprocessing :

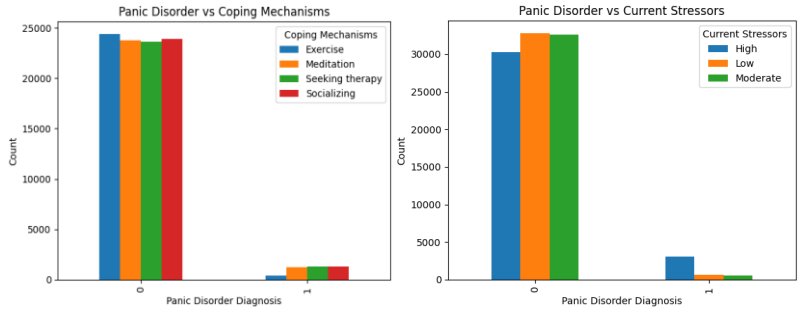
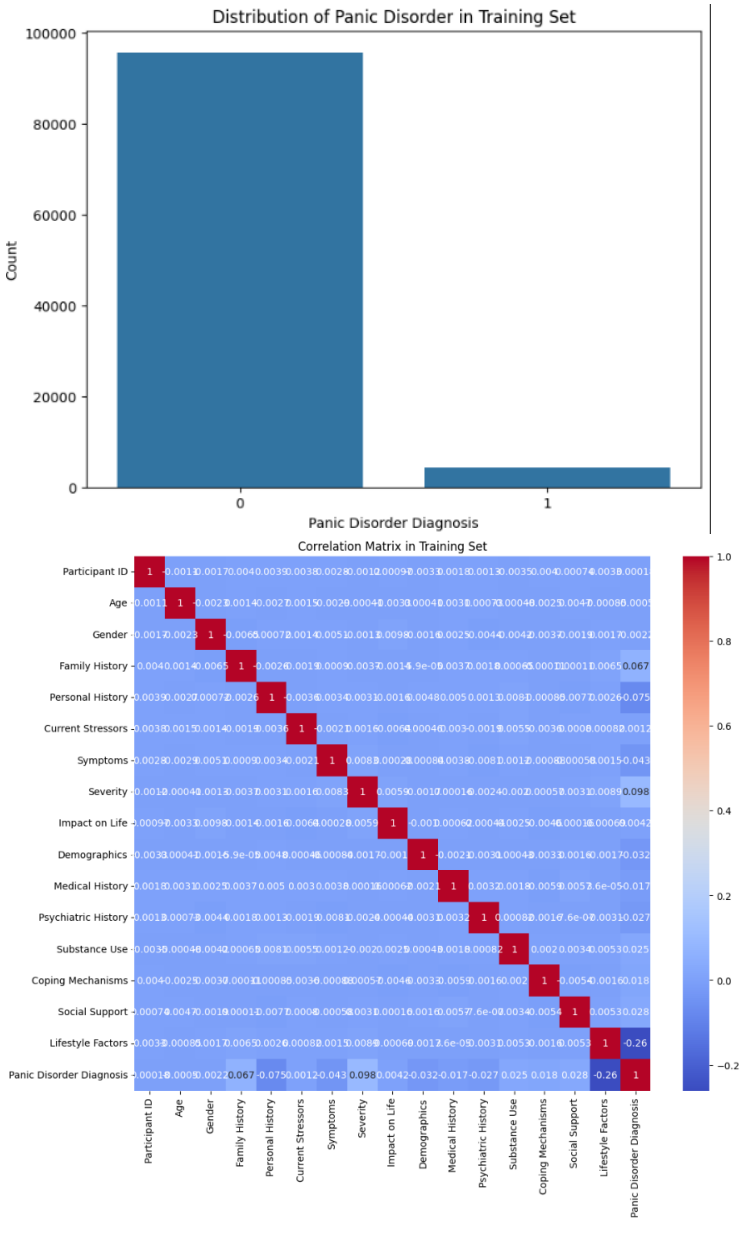
Dataset variables will be statistically analyzed to identify patterns and outliers, with Python employed for preprocessing tasks like normalization and feature engineering. Data cleaning will address missing values and outliers, ensuring quality for subsequent analysis and modeling, and forming a strong foundation for insights and predictions..

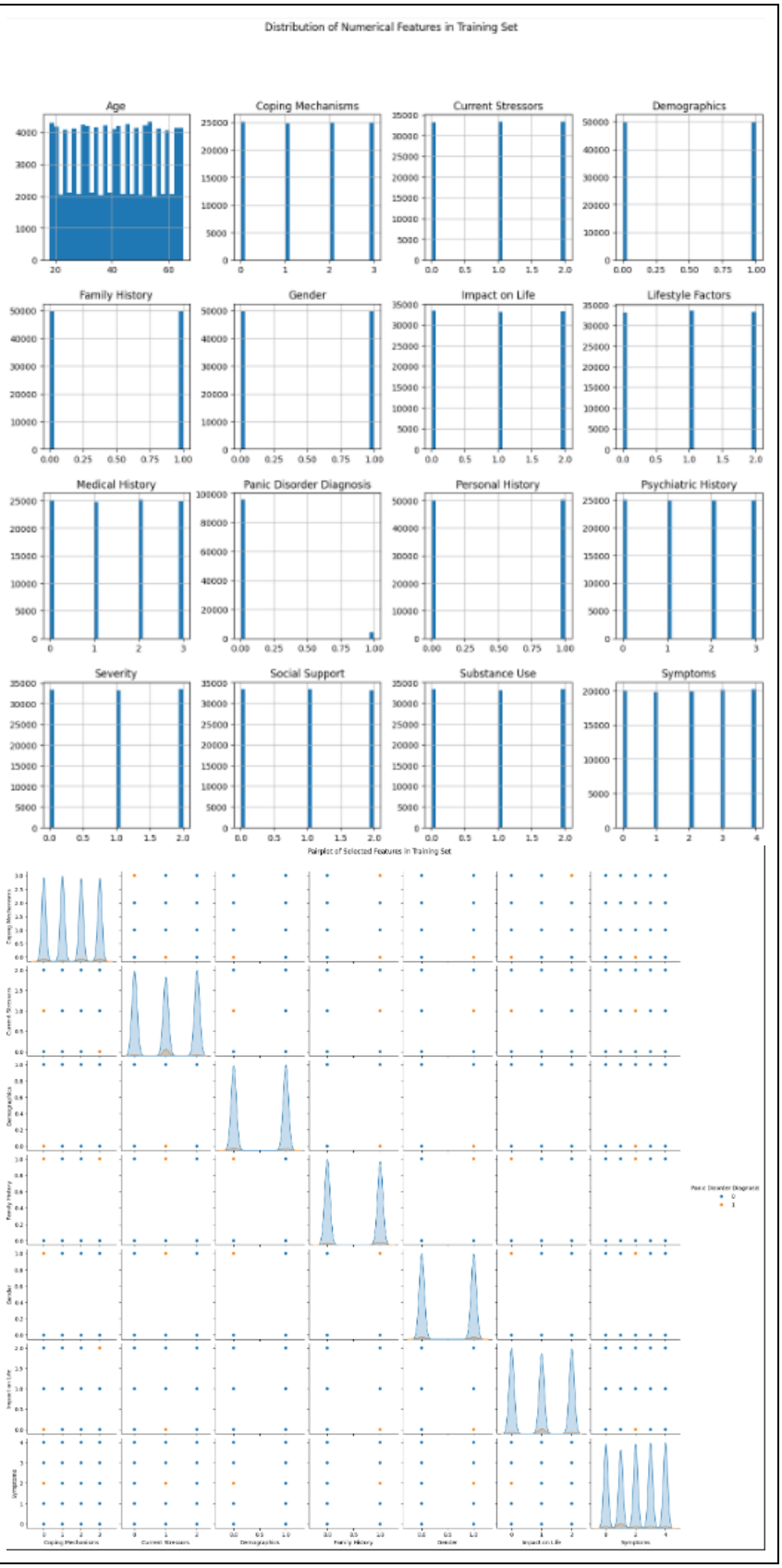
Section	Description
Data Overview	Dimension: 100000 rows x 17 columns Descriptive statistics:

	Participant ID	Age	Gender	Family History	Personal History	Current Stressors	Symptoms	Severity	Impact on Life	Demographics	Medical History	Psychiatric History	Substance Use
count	100000.000000	100000.000000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000	100000
unique	NaN	NaN	2	2	2	3	5	3	3	2	4	4	3
top	NaN	NaN	Male	Yes	No	Low	Fear of losing control	Severe	Mild	Rural	No Medical History	Bipolar disorder	Alcohol
freq	NaN	NaN	50052	50042	50210	33409	20236	33510	33452	50087	25173	25058	33445
mean	50000.500000	41.454300	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
std	28867.657797	13.839204	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
min	1.000000	18.000000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
25%	25000.750000	29.000000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
50%	50000.500000	41.000000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
75%	75000.250000	53.000000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
max	100000.000000	65.000000	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Univariate Analysis	

	
Bivariate Analysis	

<table>
<tr>
<td></td>
<td> 
</td>
</tr>
<tr>
<td>Multivariate Analysis</td>
<td> 
</td>
</tr>
</table>


Distribution of Numerical Features in Training Set
Age
Coping Mechanisms
Current Stressors
Demographics
Family History
Gender
Impact on Life
Lifestyle Factors
Medical History
Panic Disorder Diagnosis
Personal History
Psychiatric History
Severity
Social Support
Substance Use
Symptoms
Pairplot of Selected Features in Training Set

Outliers and Anomalies	`train.isnull().sum() #Determining`
	
	Participant ID 0
	Age 0
	Gender 0
	Family History 0
	Personal History 0
	Current Stressors 0
	Symptoms 0
	Severity 0
	Impact on Life 0
	Demographics 0
	Medical History 25173
	Psychiatric History 24921
	Substance Use 33374
	Coping Mechanisms 0
	Social Support 0
	Lifestyle Factors 0
	Panic Disorder Diagnosis 0
	dtype: int64
	
	`test.isnull().sum() #Determining`
	
	Participant ID 0
	Age 0
	Gender 0
	Family History 0
	Personal History 0
	Current Stressors 0
	Symptoms 0
	Severity 0
	Impact on Life 0
	Demographics 0
	Medical History 5001
	Psychiatric History 4989
	Substance Use 6617
	Coping Mechanisms 0
	Social Support 0
	Lifestyle Factors 0
	Panic Disorder Diagnosis 0
	dtype: int64

Data Preprocessing Code Screenshots

Loading Data

```
train=pd.read_csv('panic_disorder_dataset_training.csv')
train.head() #print the first 5 rows of the training dataset
```

	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
1	2	51	Male	No	No	High	Panic attacks	Mild	Mild	Urban	Asthma	Anxiety disorder	Drugs	Exercise	High	Sleep quality	0
2	3	32	Female	Yes	No	High	Panic attacks	Mild	Significant	Urban	Diabetes	Depressive disorder	NaN	Seeking therapy	Moderate	Exercise	0
3	4	64	Female	No	No	Moderate	Chest pain	Moderate	Moderate	Rural	Diabetes	NaN	NaN	Meditation	High	Exercise	0
4	5	31	Male	Yes	No	Moderate	Panic attacks	Mild	Moderate	Rural	Asthma	NaN	Drugs	Seeking therapy	Low	Sleep quality	0

Handling Missing Data

```
train.replace('None', pd.NA, inplace=True)
train['Medical History'].fillna('No Medical History', inplace=True)
train['Psychiatric History'].fillna('No Psychiatric History', inplace=True)
train['Substance Use'].fillna('No Substance Use', inplace=True)

test.replace('None', pd.NA, inplace=True)
test['Medical History'].fillna('No Medical History', inplace=True)
test['Psychiatric History'].fillna('No Psychiatric History', inplace=True)
test['Substance Use'].fillna('No Substance Use', inplace=True)
```

Data Transformation

```
le = {}
for column in train.columns:
    if train[column].dtype=='object':
        le[column] = {}
        c = 0
        for i in train[column].unique():
            le[column][i] = c
            c += 1
        train[column] = train[column].map(le[column])
le = {}
for column in test.columns:
  if test[column].dtype == object:
    le[column] = {}
    c = 0
    for i in test[column].unique():
      le[column][i] = c
      c += 1
    test[column] = test[column].map(le[column])
```

Feature Engineering

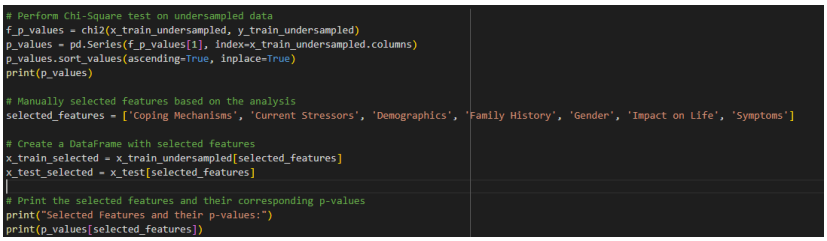
```
x_train=train.iloc[:,1:-1] #Dependent variables of the training dataset
y_train=train.iloc[:,-1] #Independent variables of the training dataset
x_test=test.iloc[:,1:-1] #Dependent variables of the testing dataset
y_test=test.iloc[:,-1] #Independent variables of the testing dataset

class_0_indices = x_train[y_train == 0].index #negative class
class_1_indices = x_train[y_train == 1].index #positive class

undersample_indices = np.random.choice(class_0_indices, size=class_1_indices.shape[0], replace=False)

x_train_undersampled = pd.concat([x_train.loc[undersample_indices], x_train.loc[class_1_indices]])
y_train_undersampled = pd.concat([y_train.loc[undersample_indices], y_train.loc[class_1_indices]])

# Print before and after balancing
print("Before balancing", Counter(y_train))
print("After balancing", Counter(y_train_undersampled))
```

Save Processed Data	

```python
# Perform Chi-Square test on undersampled data
f_p_values = chi2(x_train_undersampled, y_train_undersampled)
p_values = pd.Series(f_p_values[1], index=x_train_undersampled.columns)
p_values.sort_values(ascending=True, inplace=True)
print(p_values)

# Manually selected features based on the analysis
selected_features = ['Coping Mechanisms', 'Current Stressors', 'Demographics', 'Family History', 'Gender', 'Impact on Life', 'Symptoms']

# Create a DataFrame with selected features
x_train_selected = x_train_undersampled[selected_features]
x_test_selected = x_test[selected_features]

# Print the selected features and their corresponding p-values
print("Selected Features and their p-values:")
print(p_values[selected_features])
```