Data Collection and Preprocessing Phase

Date	09 July 2024
Team ID	SWTID1720243396
Project Title	Panic Disorder Detection
Maximum Marks	6 Marks

Section	Description
Data Overview	

```
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
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

Next steps: Generate code with train | View recommended plots

```
[10] 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	38	Female	Yes	No	High	Chest pain	Severe	Significant	Rural	Asthma	Depressive disorder	NaN	Seeking therapy	Low	Exercise	0

Next steps: Generate code with test | View recommended plots

```
[11] print("Training set", train.shape)
     print("Testing set", test.shape)

Training set (100000, 17)
Testing set (20000, 17)
```

After Undersampling

	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
count	8570.000000	8570.000000	8570.000000	8570.000000	8570.000000	8570.000000	8570.000000	8570.000000	8570.000000	8570.000000	8570.000000	8570.000000	8570.000000	8570.000000	8570.000000	8570.000000
mean	41.379347	0.491365	0.581447	0.593466	1.282497	1.855776	1.190665	1.200000	0.470478	1.405018	1.433722	0.945041	1.534306	0.951342	0.517036	0.500000
std	13.808046	0.499955	0.493351	0.491215	0.823641	1.365161	0.831354	0.829861	0.499157	1.092158	1.098746	0.807908	1.157380	0.823190	0.768745	0.500029
min	18.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
25%	29.000000	0.000000	0.000000	0.000000	1.000000	1.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
50%	41.000000	0.000000	1.000000	1.000000	2.000000	2.000000	1.000000	1.000000	0.000000	1.000000	1.000000	1.000000	2.000000	1.000000	0.000000	0.500000
75%	53.000000	1.000000	1.000000	1.000000	2.000000	3.000000	2.000000	2.000000	1.000000	2.000000	2.000000	2.000000	3.000000	2.000000	1.000000	1.000000
max	65.000000	1.000000	1.000000	1.000000	2.000000	4.000000	2.000000	2.000000	1.000000	3.000000	3.000000	2.000000	3.000000	2.000000	2.000000	1.000000

```
train.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 100000 entries, 0 to 99999
Data columns (total 17 columns):
 #   Column                    Non-Null Count   Dtype
---  ------                    --------------   -----
 0   Participant ID            100000 non-null  int64
 1   Age                       100000 non-null  int64
 2   Gender                    100000 non-null  object
 3   Family History            100000 non-null  object
 4   Personal History          100000 non-null  object
 5   Current Stressors         100000 non-null  object
 6   Symptoms                  100000 non-null  object
 7   Severity                  100000 non-null  object
 8   Impact on Life            100000 non-null  object
 9   Demographics              100000 non-null  object
 10  Medical History           74827 non-null   object
 11  Psychiatric History       75079 non-null   object
 12  Substance Use             66626 non-null   object
 13  Coping Mechanisms         100000 non-null  object
 14  Social Support            100000 non-null  object
 15  Lifestyle Factors         100000 non-null  object
 16  Panic Disorder Diagnosis  100000 non-null  int64
dtypes: int64(3), object(14)
memory usage: 13.0+ MB
```

```
test.info()
```

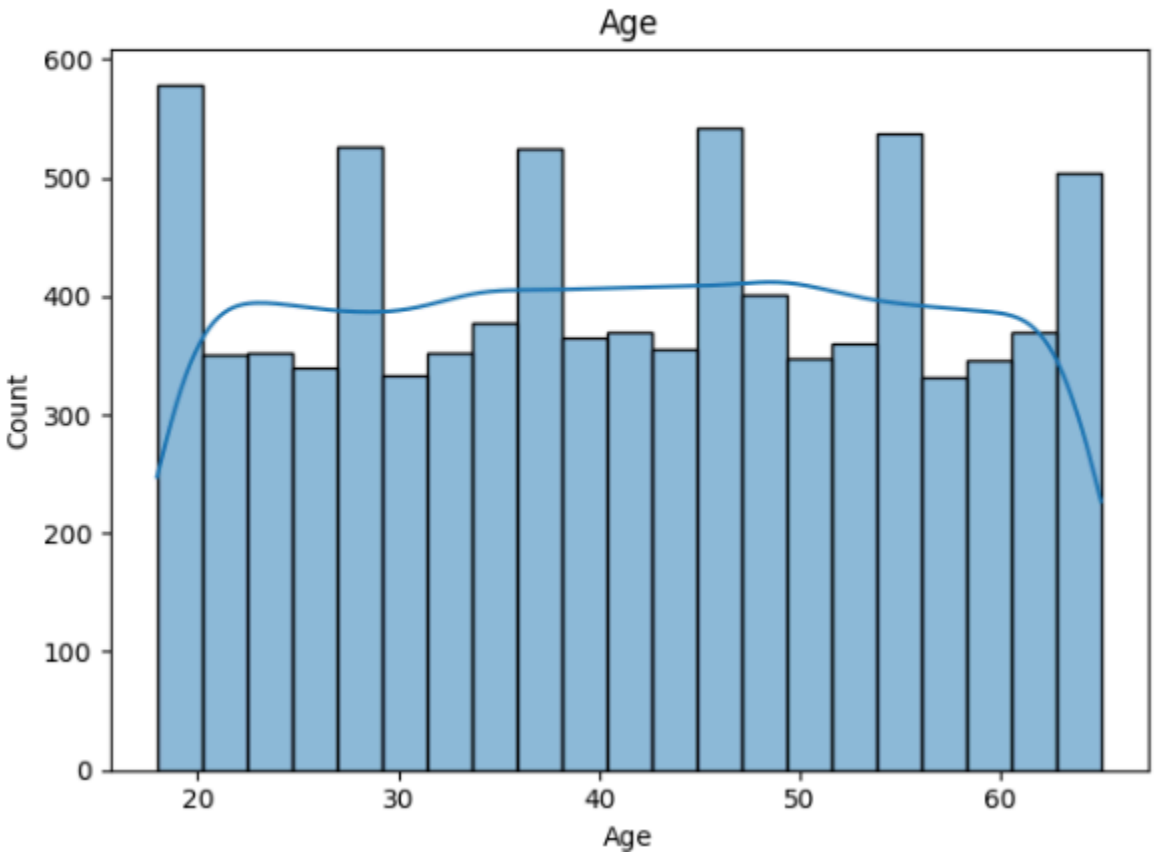
```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 20000 entries, 0 to 19999
Data columns (total 17 columns):
 #   Column                    Non-Null Count  Dtype
---  ------                    --------------  -----
 0   Participant ID            20000 non-null  int64
 1   Age                       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       15011 non-null  object
 12  Substance Use             13383 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(3), object(14)
memory usage: 2.6+ MB
```

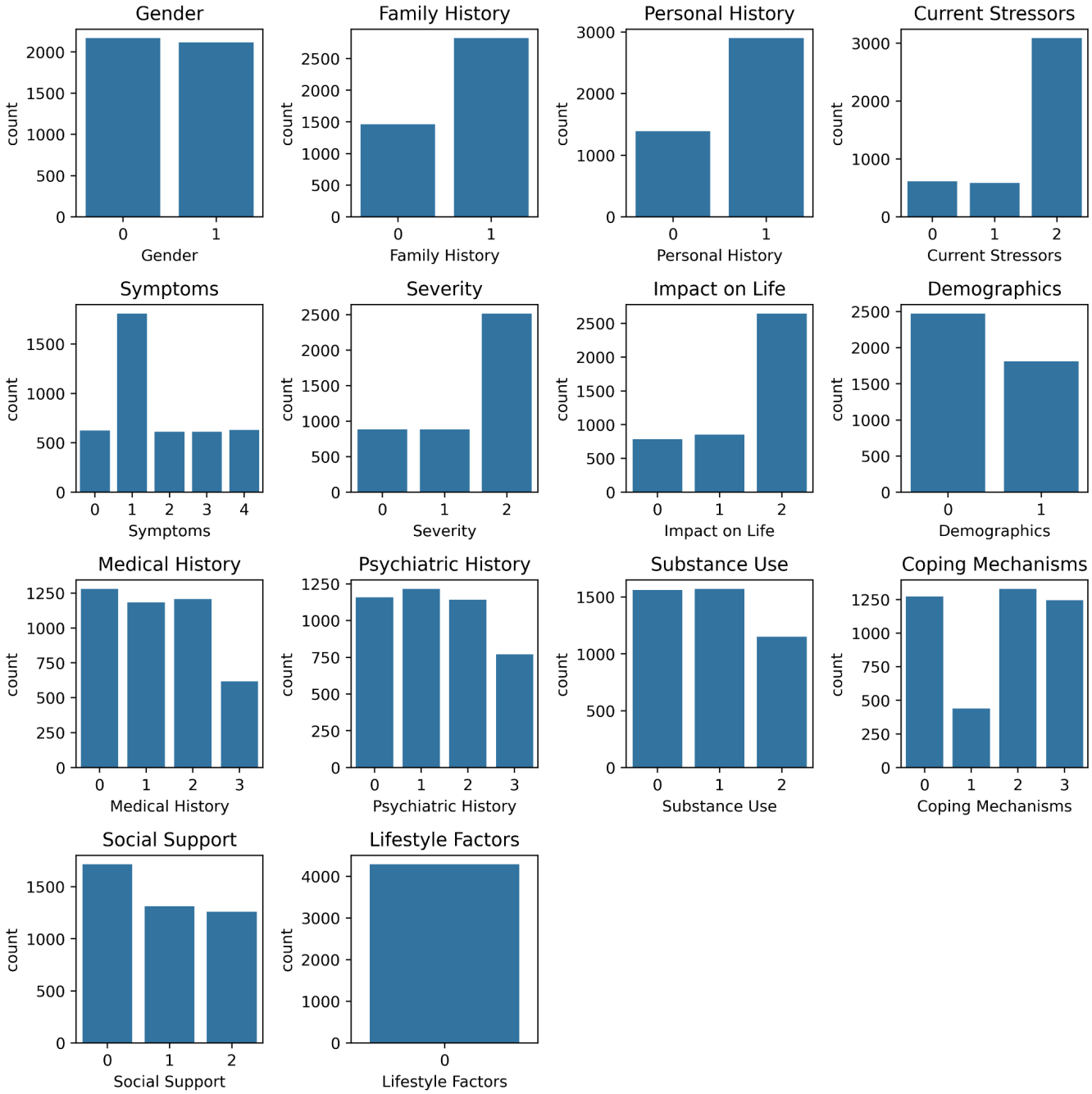
```
train.columns
```

```
Index(['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'],
      dtype='object')
```

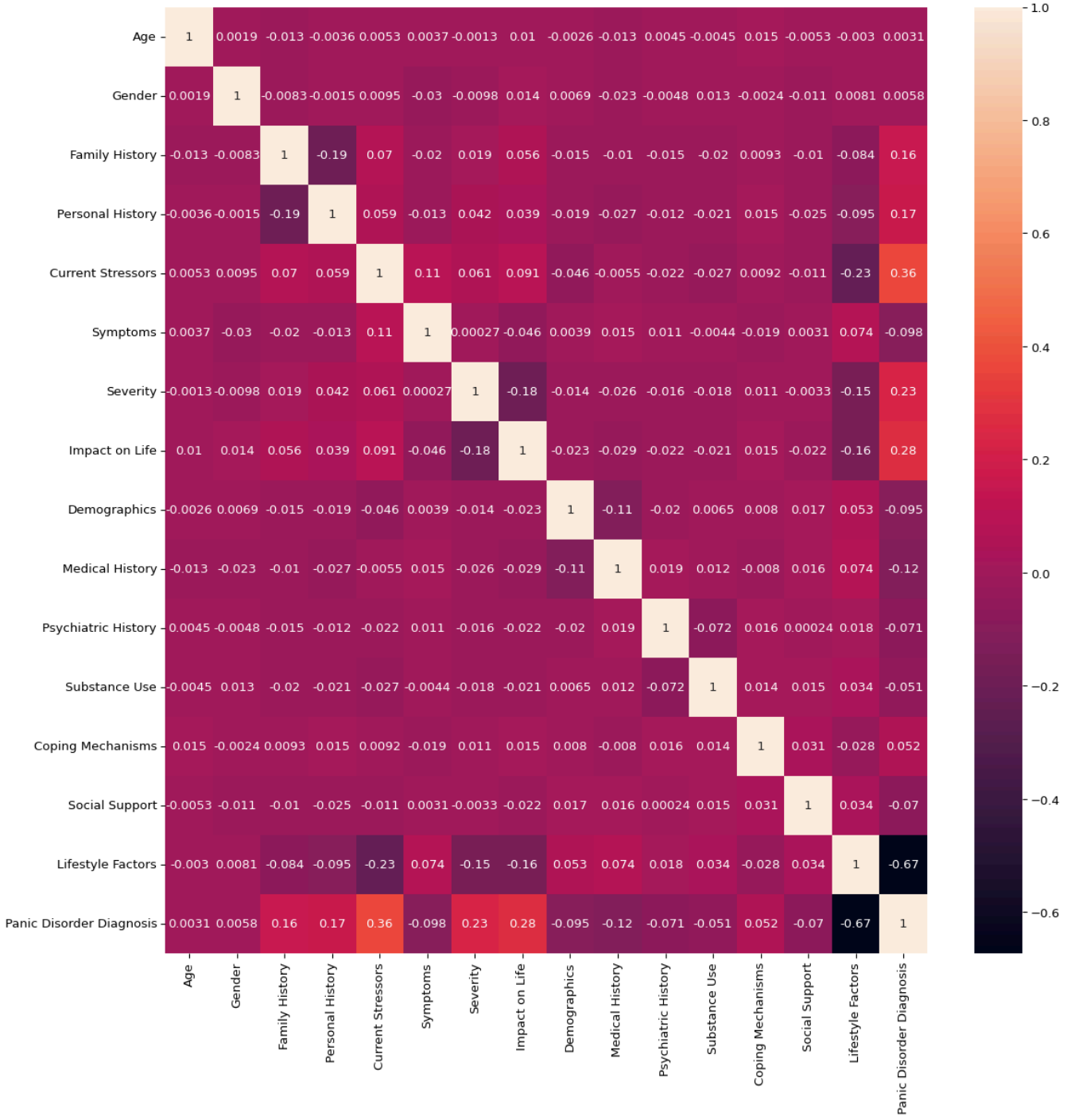
Univariate Analysis	``` cols = ["Age"] plt.boxplot(train["Age"]) plt.title("Box Plot of Age") plt.tight_layout() plt.show() ``` 

```
sns.histplot(train["Age"],kde=True)
plt.title("Age")
plt.tight_layout()
plt.show()
```



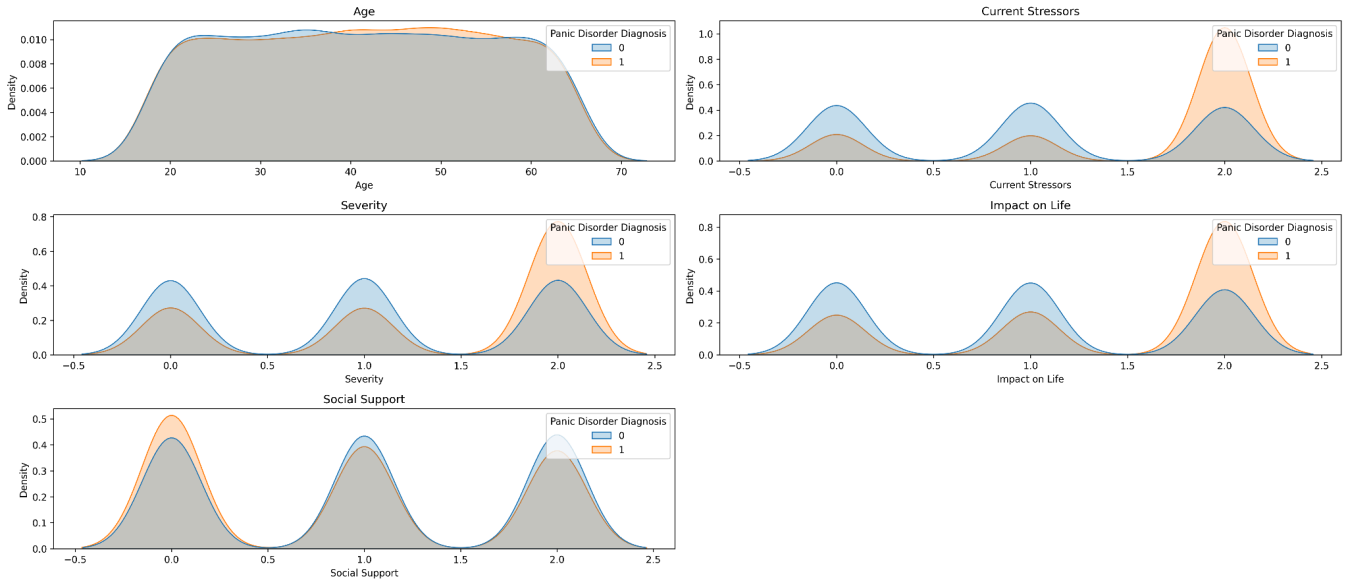


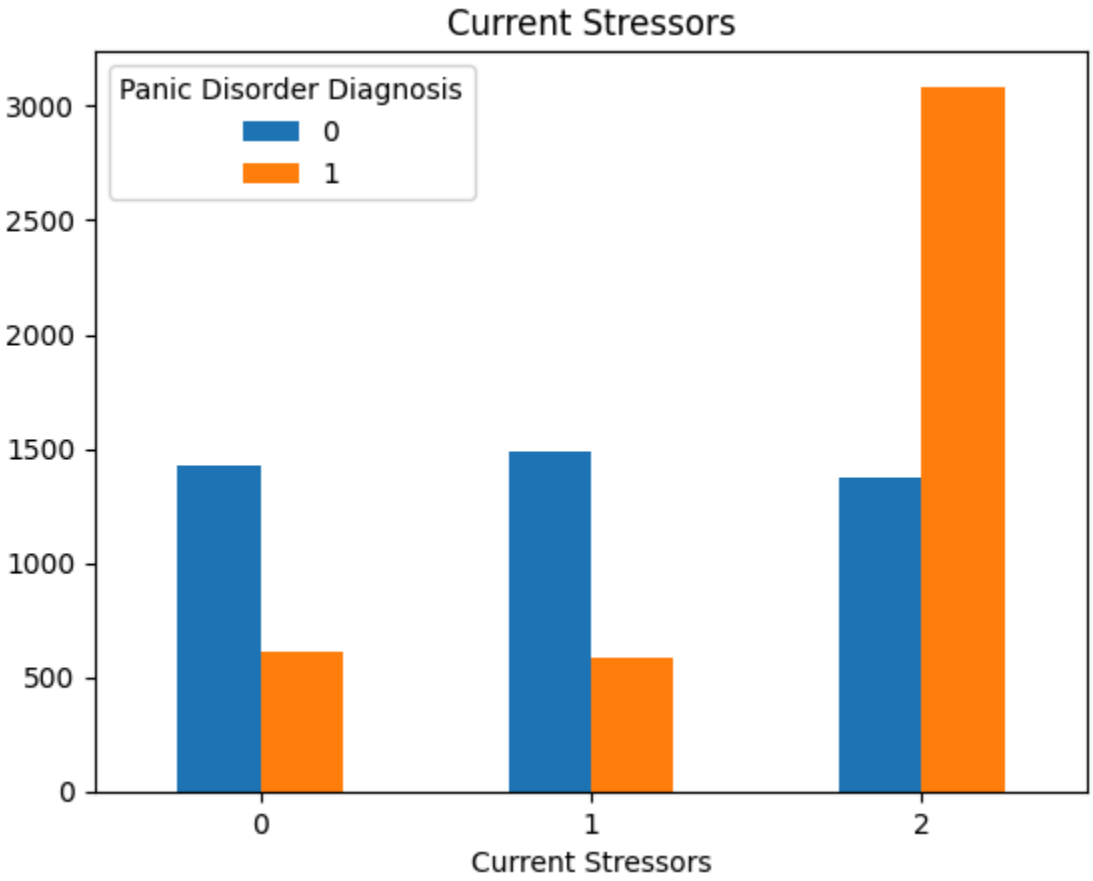
Bivariate Analysis

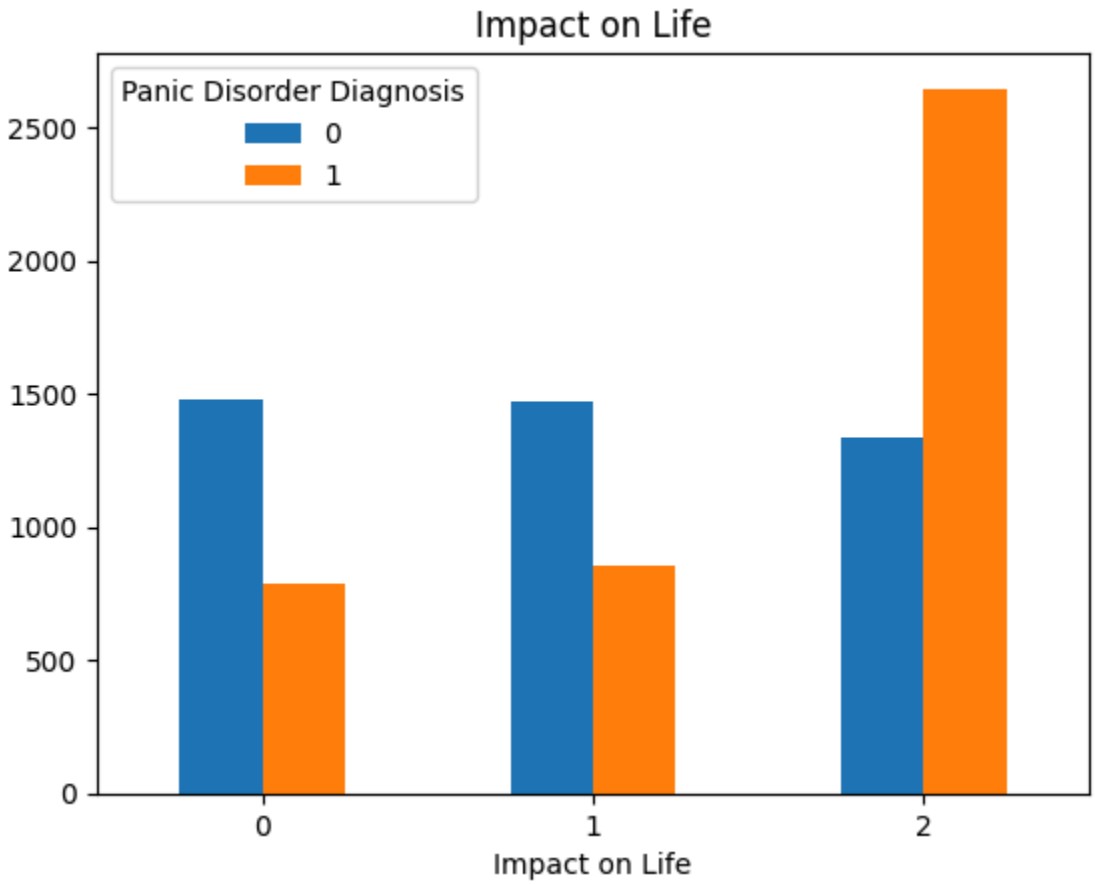

Age 1 0.0019 -0.013 -0.0036 0.0053 0.0037 -0.0013 0.01 -0.0026 -0.013 0.0045 -0.0045 0.015 -0.0053 -0.003 0.0031
Gender 0.0019 1 -0.0083 -0.0015 0.0095 -0.03 -0.0098 0.014 0.0069 -0.023 -0.0048 0.013 -0.0024 -0.011 0.0081 0.0058
Family History -0.013 -0.0083 1 -0.19 0.07 -0.02 0.019 0.056 -0.015 -0.01 -0.015 -0.02 0.0093 -0.01 -0.084 0.16
Personal History -0.0036 -0.0015 -0.19 1 0.059 -0.013 0.042 0.039 -0.019 -0.027 -0.012 -0.021 0.015 -0.025 -0.095 0.17
Current Stressors 0.0053 0.0095 0.07 0.059 1 0.11 0.061 0.091 -0.046 -0.0055 -0.022 -0.027 0.0092 -0.011 -0.23 0.36
Symptoms 0.0037 -0.03 -0.02 -0.013 0.11 1 0.00027 -0.046 0.0039 0.015 0.011 -0.0044 -0.019 0.0031 0.074 -0.098
Severity -0.0013 -0.0098 0.019 0.042 0.061 0.00027 1 -0.18 -0.014 -0.026 -0.016 -0.018 0.011 -0.0033 -0.15 0.23
Impact on Life 0.01 0.014 0.056 0.039 0.091 -0.046 -0.18 1 -0.023 -0.029 -0.022 -0.021 0.015 -0.022 -0.16 0.28
Demographics -0.0026 0.0069 -0.015 -0.019 -0.046 0.0039 -0.014 -0.023 1 -0.11 -0.02 0.0065 0.008 0.017 0.053 -0.095
Medical History -0.013 -0.023 -0.01 -0.027 -0.0055 0.015 -0.026 -0.029 -0.11 1 0.019 0.012 -0.008 0.016 0.074 -0.12
Psychiatric History 0.0045 -0.0048 -0.015 -0.012 -0.022 0.011 -0.016 -0.022 -0.02 0.019 1 -0.072 0.016 0.00024 0.018 -0.071
Substance Use -0.0045 0.013 -0.02 -0.021 -0.027 -0.0044 -0.018 -0.021 0.0065 0.012 -0.072 1 0.014 0.015 0.034 -0.051
Coping Mechanisms 0.015 -0.0024 0.0093 0.015 0.0092 -0.019 0.011 0.015 0.008 -0.008 0.016 0.014 1 0.031 -0.028 0.052
Social Support -0.0053 -0.011 -0.01 -0.025 -0.011 0.0031 -0.0033 -0.022 0.017 0.016 0.00024 0.015 0.031 1 0.034 -0.07
Lifestyle Factors -0.003 0.0081 -0.084 -0.095 -0.23 0.074 -0.15 -0.16 0.053 0.074 0.018 0.034 -0.028 0.034 1 -0.67
Panic Disorder Diagnosis 0.0031 0.0058 0.16 0.17 0.36 -0.098 0.23 0.28 -0.095 -0.12 -0.071 -0.051 0.052 -0.07 -0.67 1
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
1.0
0.8
0.6
0.4
0.2
0.0
−0.2
−0.4
−0.6

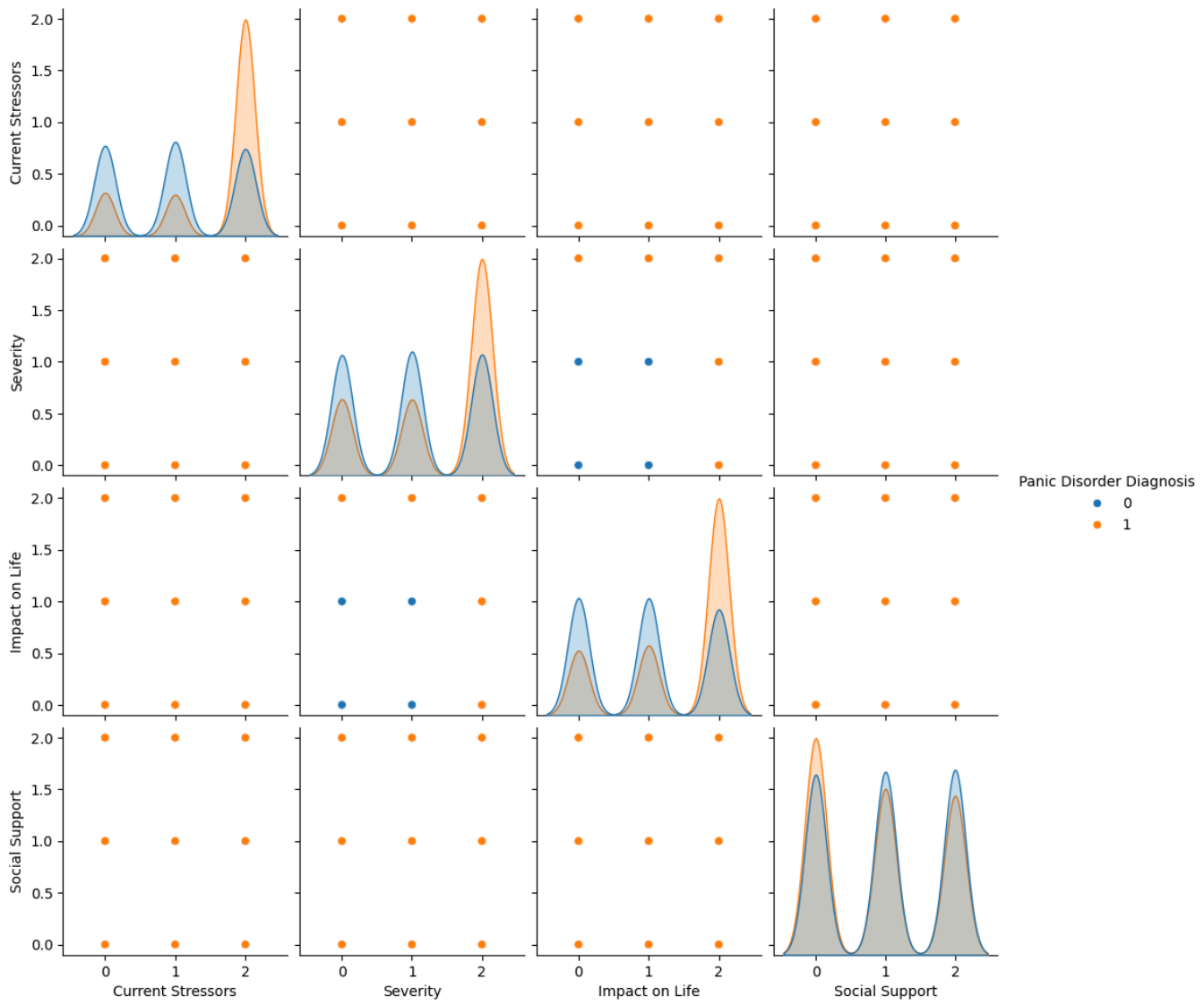
```
train.groupby('Panic Disorder Diagnosis').agg("mean")[["Age","Current Stressors","Severity","Impact on Life","Social Support"]]
```

Panic Disorder Diagnosis	Age	Current Stressors	Severity	Impact on Life	Social Support
0	41.337223	0.988331	1.00140	0.966628	1.008868
1	41.421470	1.576663	1.37993	1.433372	0.893816



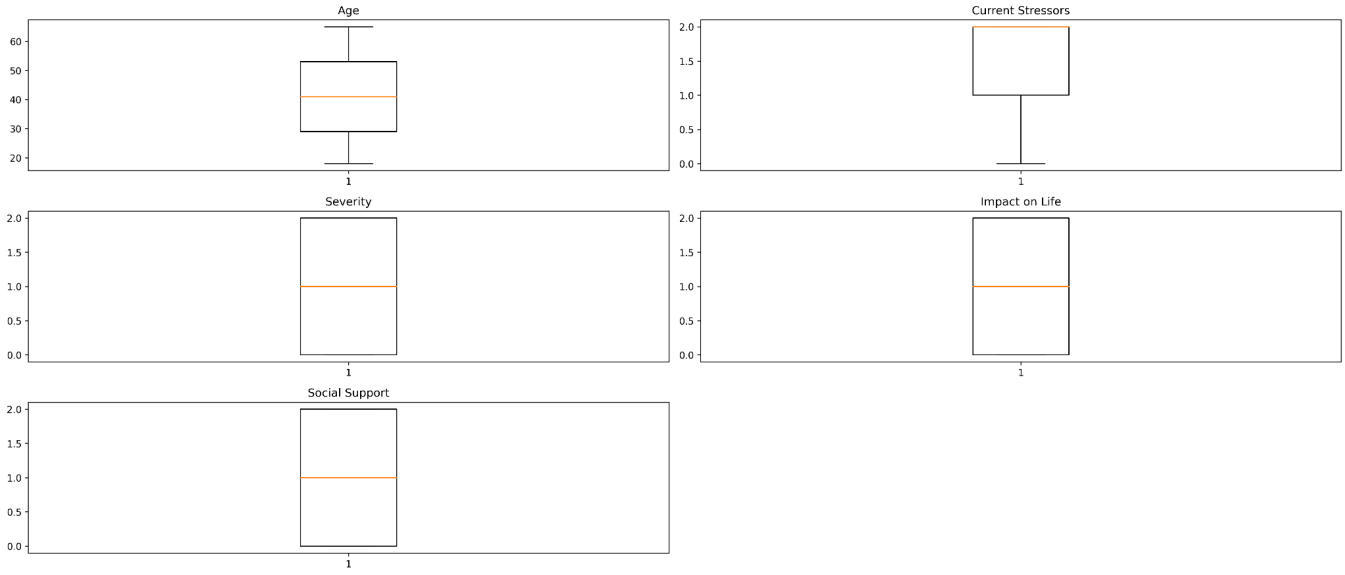

Current Stressors
Panic Disorder Diagnosis
0
1
3000
2500
2000
1500
1000
500
0
0
1
2
Current Stressors


Impact on Life
Panic Disorder Diagnosis
0
1
2500
2000
1500
1000
500
0
0
1
2
Impact on Life

Multivariate Analysis	
Outliers and Anomalies	Filtering out Outliers An outlier is an observation that lies an abnormal distance from other values in a random sample from a population. In a sense, this definition leaves it up to the analyst (or a consensus process) to decide what will be considered abnormal. We can detect/handling outliers through 2 major detections: 1. IQR Method 2. Z-Score We will be applying all three methods one by one effectively analyse our outlier data

```
skewvalue=train.skew(axis=0)
skewvalue
```

```
Age                          0.000628
Gender                       0.002080
Family History              -0.001680
Personal History             0.008400
Current Stressors            0.000735
Symptoms                    -0.010331
Severity                    -0.002668
Impact on Life               0.004354
Demographics                 0.003480
Medical History             -0.004513
Psychiatric History          0.002568
Substance Use               -0.003543
Coping Mechanisms            0.001535
Social Support              -0.004758
Lifestyle Factors           -0.004749
Panic Disorder Diagnosis     4.514710
dtype: float64
```



No outliers present

Data Preprocessing Code Screenshots

Loading Data	

```
od.download("https://www.kaggle.com/datasets/muhammadshahidazeem/panic-disorder-detection-dataset")
```

```
Please provide your Kaggle credentials to download this dataset. Learn more: http://bit.ly/kaggle-creds
Your Kaggle username: arvasugupta
Your Kaggle Key: ··········
Dataset URL: https://www.kaggle.com/datasets/muhammadshahidazeem/panic-disorder-detection-dataset
Downloading panic-disorder-detection-dataset.zip to ./panic-disorder-detection-dataset
100%|██████████| 1.50M/1.50M [00:00<00:00, 2.66MB/s]
```

Loading the Datasets

```
input_test = ('./panic-disorder-detection-dataset/panic_disorder_dataset_testing.csv')
input_train = ('./panic-disorder-detection-dataset/panic_disorder_dataset_training.csv')
test = pd.read_csv(input_test)
train = pd.read_csv(input_train)
```

Handling Missing Data

```
print(train.isnull().values.sum())
train[train.isnull().any(axis=1)]
```

```
83468
```

	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
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
5	6	38	Male	Yes	Yes	Moderate	Dizziness	Moderate	Significant	Urban	NaN	Bipolar disorder	Alcohol	Seeking therapy	Moderate	Exercise	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
99994	99995	24	Male	Yes	No	High	Dizziness	Moderate	Moderate	Rural	NaN	Bipolar disorder	Drugs	Seeking therapy	Low	Exercise	0
99995	99996	22	Male	Yes	No	High	Chest pain	Mild	Mild	Rural	Heart disease	NaN	NaN	Socializing	Low	Diet	0
99996	99997	57	Female	No	Yes	Low	Panic attacks	Severe	Mild	Rural	Heart disease	Depressive disorder	NaN	Meditation	High	Diet	0
99997	99998	20	Male	Yes	No	Moderate	Panic attacks	Severe	Moderate	Rural	Heart disease	Bipolar disorder	NaN	Seeking therapy	Low	Exercise	0
99999	100000	18	Male	No	No	Low	Panic attacks	Severe	Mild	Rural	Diabetes	NaN	NaN	Meditation	Moderate	Exercise	0

62560 rows × 17 columns

```
print(test.isnull().values.sum())
test[test.isnull().any(axis=1)]
```

16607

	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
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
5	6	23	Female	Yes	Yes	Moderate	Panic attacks	Moderate	Moderate	Rural	NaN	Anxiety disorder	NaN	Seeking therapy	Low	Exercise	0
7	8	64	Male	No	No	High	Shortness of breath	Mild	Mild	Rural	Heart disease	NaN	NaN	Socializing	High	Exercise	0
8	9	49	Male	No	No	Low	Dizziness	Severe	Moderate	Urban	NaN	NaN	Alcohol	Exercise	High	Diet	0
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
19992	19993	38	Female	Yes	No	Moderate	Dizziness	Moderate	Significant	Rural	Heart disease	NaN	Drugs	Socializing	High	Diet	0
19993	19994	62	Male	Yes	No	Low	Fear of losing control	Mild	Significant	Urban	NaN	NaN	Drugs	Seeking therapy	Low	Sleep quality	0
19994	19995	27	Male	Yes	Yes	Moderate	Dizziness	Mild	Moderate	Urban	NaN	Anxiety disorder	Drugs	Seeking therapy	High	Diet	0
19995	19996	31	Female	Yes	Yes	High	Chest pain	Moderate	Moderate	Rural	Heart disease	Bipolar disorder	NaN	Exercise	Moderate	Sleep quality	0
19998	19999	28	Male	No	Yes	Moderate	Dizziness	Mild	Significant	Rural	Heart disease	Anxiety disorder	NaN	Meditation	Moderate	Sleep quality	0

12437 rows × 17 columns

```
print(train.drop(labels = ['Medical History','Psychiatric History','Substance Use'],axis = 1).isnull().values.sum())
train[train.drop(labels = ['Medical History','Psychiatric History','Substance Use'],axis = 1).isnull().any(axis=1)]
```

0

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

```
print(test.drop(labels = ['Medical History','Psychiatric History','Substance Use'],axis = 1).isnull().values.sum())
test[test.drop(labels = ['Medical History','Psychiatric History','Substance Use'],axis = 1).isnull().any(axis=1)]
```

0

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

```
# Dealing with Multivariate Ordinal(with missing valus)
## Note: There are two paths. Coding mising value as another vairables, thus removing the missing value(no need of imputation methods). OR we could not mark the
## mising value and deal with remaining with imputation methods

## Rationally we should go by first and mark missing value, because missing values in these three columns have logical reason. There was no option to leave as blank or
## possibly no option to mark as "other" or "none". Hence assuming it as other is most rational decision.
## Thus this will also solve the issue of missing value


train.fillna("Others",inplace=True,axis = 1)
test.fillna("Others",inplace=True,axis = 1)
train["Medical History"] = train["Medical History"].replace({'Diabetes':0,'Asthma':1,'Heart disease':2,'Others':3}).astype('int64')
train["Psychiatric History"] = train["Psychiatric History"].replace({'Bipolar disorder':0,'Anxiety disorder':1,'Depressive disorder':2,'Others':3}).astype('int64')
train["Substance Use"] = train["Substance Use"].replace({'Drugs':0,'Alcohol':1,'Others':2}).astype('int64')


test["Medical History"] = test["Medical History"].replace({'Diabetes':0,'Asthma':1,'Heart disease':2,'Others':3}).astype('int64')
test["Psychiatric History"] = test["Psychiatric History"].replace({'Bipolar disorder':0,'Anxiety disorder':1,'Depressive disorder':2,'Others':3}).astype('int64')
test["Substance Use"] = test["Substance Use"].replace({'Drugs':0,'Alcohol':1,'Others':2}).astype('int64')
```

Data Transformation

```
print(train.drop(labels = ['Participant ID','Age','Panic Disorder Diagnosis'],axis = 1).apply(lambda col: col.unique()))
```

```
Gender                                                [Male, Female]
Family History                                             [No, Yes]
Personal History                                           [Yes, No]
Current Stressors                                [Moderate, High, Low]
Symptoms              [Shortness of breath, Panic attacks, Chest pai...
Severity                                    [Mild, Moderate, Severe]
Impact on Life                         [Mild, Significant, Moderate]
Demographics                                           [Rural, Urban]
Medical History                 [Diabetes, Asthma, nan, Heart disease]
Psychiatric History   [Bipolar disorder, Anxiety disorder, Depressiv...
Substance Use                                    [nan, Drugs, Alcohol]
Coping Mechanisms     [Socializing, Exercise, Seeking therapy, Medit...
Social Support                                   [High, Moderate, Low]
Lifestyle Factors                   [Sleep quality, Exercise, Diet]
dtype: object
```

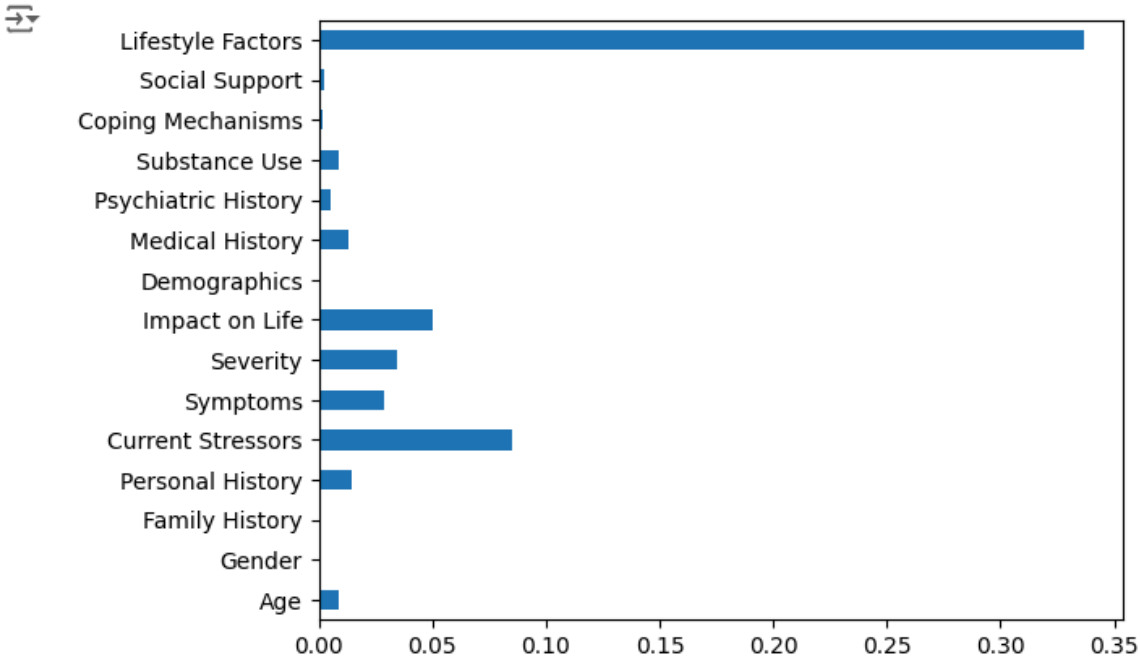
```
train['Gender']=train['Gender'].replace({'Male':0,'Female':1}).astype('int64')
train['Family History']=train['Family History'].replace({'No':0,'Yes':1}).astype('int64')
train['Personal History']=train['Personal History'].replace({'No':0,'Yes':1}).astype('int64')
train['Demographics']=train['Demographics'].replace({'Rural':0,'Urban':1}).astype('int64')

test['Gender']=test['Gender'].replace({'Male':0,'Female':1}).astype('int64')
test['Family History']=test['Family History'].replace({'No':0,'Yes':1}).astype('int64')
test['Personal History']=test['Personal History'].replace({'No':0,'Yes':1}).astype('int64')
test['Demographics'] = test['Demographics'].replace({'Rural':0,'Urban':1}).astype('int64')
```

```
#Dealing with Multivariate Nominal
train["Symptoms"] = train['Symptoms'].replace({'Shortness of breath':0,'Panic attacks':1,'Chest pain':2,'Dizziness':3,'Fear of losing control':4}).astype('int64')
train["Coping Mechanisms"] = train['Coping Mechanisms'].replace({'Socializing':0,'Exercise':1,'Seeking therapy':2,'Meditation':3}).astype('int64')
train["Lifestyle Factors"] = train['Lifestyle Factors'].replace({'Sleep quality':0,'Exercise':1,'Diet':2}).astype('int64')

test["Symptoms"] = test["Symptoms"].replace({'Shortness of breath':0,'Panic attacks':1,'Chest pain':2,'Dizziness':3,'Fear of losing control':4}).astype('int64')
test["Coping Mechanisms"] = test["Coping Mechanisms"].replace({'Socializing':0,'Exercise':1,'Seeking therapy':2,'Meditation':3}).astype('int64')
test["Lifestyle Factors"] = test["Lifestyle Factors"].replace({'Sleep quality':0,'Exercise':1,'Diet':2}).astype('int64')
```

```
# Dealing with Multivariate Ordinal(without missing values)
test["Current Stressors"] = test["Current Stressors"].replace({'Low':0,'Moderate':1,'High':2}).astype('int64')
test["Severity"] = test["Severity"].replace({'Mild':0,'Moderate':1,'Severe':2}).astype('int64')
test["Impact on Life"] = test["Impact on Life"].replace({'Mild':0,'Moderate':1,'Significant':2}).astype('int64')
test["Social Support"] = test["Social Support"].replace({'Low':0,'Moderate':1,'High':2}).astype('int64')

train["Current Stressors"] = train["Current Stressors"].replace({'Low':0,'Moderate':1,'High':2}).astype('int64')
train["Severity"] = train["Severity"].replace({'Mild':0,'Moderate':1,'Severe':2}).astype('int64')
train["Impact on Life"] = train["Impact on Life"].replace({'Mild':0,'Moderate':1,'Significant':2}).astype('int64')
train["Social Support"] = train["Social Support"].replace({'Low':0,'Moderate':1,'High':2}).astype('int64')
```

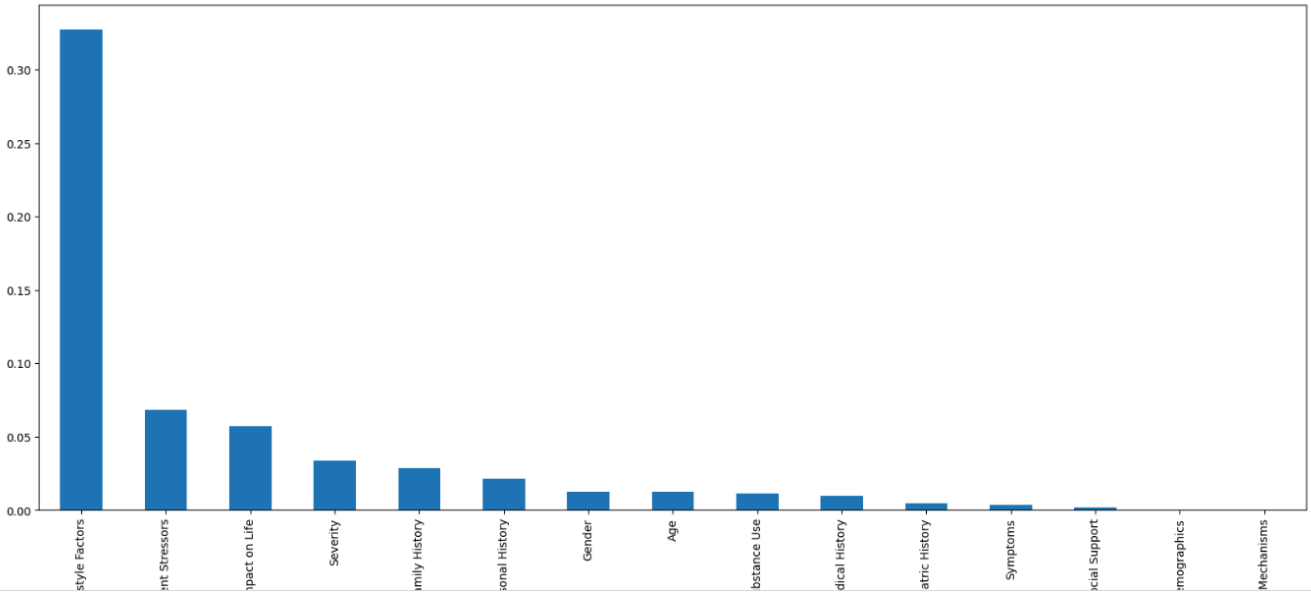
Feature Engineering

```
importnaces = mutual_info_classif(train_resampled,train_resampled_pred)
feat_importances = pd.Series(importnaces, index=train_resampled.columns[0:len(train_resampled.columns)])
feat_importances.plot(kind='barh')
plt.show()
```




```
ig = mutual_info_regression(train_resampled, train_resampled_pred)
mutual_info = pd.Series(ig)
mutual_info.index = train_resampled.columns
mutual_info.sort_values(ascending=False)
mutual_info.sort_values(ascending=False).plot.bar(figsize=(20, 8))
```

```
<Axes: >
```

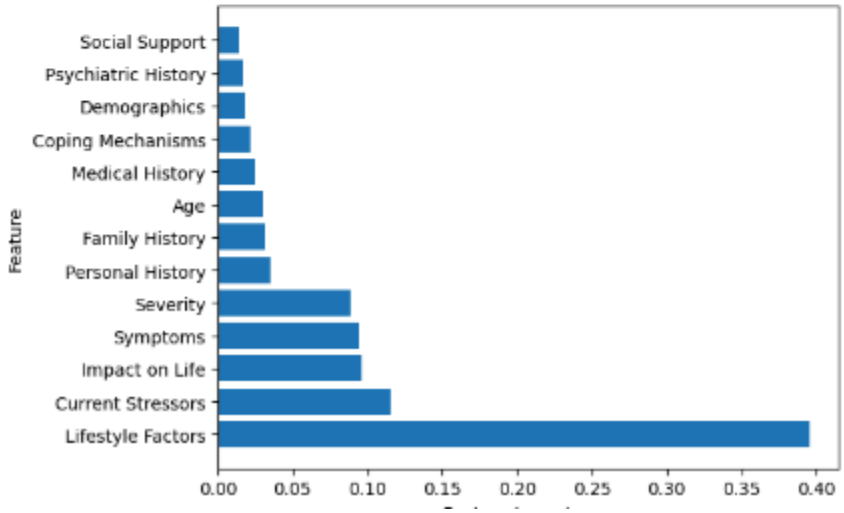


```
estimator = ExtraTreesRegressor(random_state=42)

rfecv = RFECV(estimator, min_features_to_select = 1)

# Fit the data
rfecv.fit(train_resampled, train_resampled_pred)

# Get integer index of the features selected
feature_index = rfecv.get_support(indices = True)

# Get a mask of the features selected
feature_mask = rfecv.support_

# Get selected feature names
feature_names = rfecv.get_feature_names_out()

# Get the number of features retained
feature_number = rfecv.n_features_

# Get results
results = pd.DataFrame(rfecv.cv_results_)

# Get RFECV score
rfecv_score = rfecv.score(train_resampled, train_resampled_pred)

# Print feature number, names and score
print('Original feature number:', len(train_resampled.columns))
print('Optimal feature number:', feature_number)
print('Selected features:', feature_names)
print('Score:', rfecv_score)
```

```
Original feature number: 15
Optimal feature number: 13
Selected features: ['Family History' 'Personal History' 'Current Stressors' 'Symptoms'
 'Severity' 'Impact on Life' 'Demographics' 'Medical History'
 'Psychiatric History' 'Substance Use' 'Coping Mechanisms'
 'Social Support' 'Lifestyle Factors']
Score: 1.0
```

```
rfc = RandomForestClassifier(n_estimators=100, random_state=42)

# Fit the model to the training data
rfc.fit(train_resampled,train_resampled_pred)

# Get feature importances from the trained model
importances = rfc.feature_importances_

# Sort the feature importances in descending order
indices = np.argsort(importances)[::-1]


# Select the top 10 features
num_features = feature_number
top_indices = indices[:num_features]
top_importances = importances[top_indices]


# Print the top 10 feature rankings
print("Top 10 feature rankings:")
for f in range(num_features):  # Use num_features instead of 10
    print(f"{f+1}. {train_resampled.columns[indices[f]]}: {importances[indices[f]]}")
# Plot the top 10 feature importances in a horizontal bar chart
plt.barh(range(num_features), top_importances, align="center")
plt.yticks(range(num_features), train_resampled.columns[top_indices])
plt.xlabel("Feature Importance")
plt.ylabel("Feature")
plt.show()
```

```
Top 10 feature rankings:
1. Lifestyle Factors: 0.3951861333418734
2. Current Stressors: 0.11496112541044148
3. Impact on Life: 0.09568726517618337
4. Symptoms: 0.09472545539798145
5. Severity: 0.08875364959757988
6. Personal History: 0.03465034506542778
7. Family History: 0.03176352882965312
8. Age: 0.029991146809159583
9. Medical History: 0.025036888246801142
10. Coping Mechanisms: 0.02124736125853044
11. Demographics: 0.01863970343579124
12. Psychiatric History: 0.01640311773014501
13. Social Support: 0.01376213679756068
```



Save Processed Data	-