Predictive Modeling of Panic Disorder Based on User Profile

Deborupa Pal

[1]Indiana University-Purdue University, Indianapolis-46202, USA

[fnsama, debpal, pumishra, nekowth, saivadal, famatul] @iu.edu

Abstract. The study focuses on the development of a predictive model for panic disorder leveraging user profiles. Panic disorder is a prevalent mental health condition, and understanding its manifestation in individuals is crucial for early detection and intervention. The research employs a dataset containing diverse user profiles and associated panic disorder diagnoses. Machine learning techniques, including feature engineering and model training, are applied to discern patterns and relationships between user attributes and the presence of panic disorder. The predictive model aims to offer valuable insights into the factors influencing the likelihood of panic disorder development. The study contributes to the advancement of mental health analytics, providing a foundation for proactive identification and targeted support for individuals at risk of panic disorder.

Keywords: Panic Disorder; Mental health; Reporting; Kaggle database; Machine learning; Predictive model

1. Project Scope

1.1 Introduction

Panic disorder (PD) is a common mental disorder with a lifetime prevalence of about 1.6%-3.5% worldwide. Its main characteristic is the fear of recurrent panic attacks (PAs) and loss of control, which leads to functional impairment. Patients suffering from PD often make frequent visits to the emergency department before formal diagnosis and psychoeducation. Functional impairment of PD can be avoidant behavior in terms of crowds, open spaces, traffic vehicles, or stressful situations. Severe PD cases may become homebound. Accurate PA prediction may help clinicians to provide appropriate, timely treatment and to optimize personalized medicine (Tsai et al. 2022). Early detection and intervention are critical for symptom management and long-term outcome improvement (Craske & Barlow, 2014).

Machine learning is a technique that aims to construct systems that can improve through experience by using advanced statistical and probabilistic techniques. It is believed to be a significantly useful tool to help in predicting mental health. It allows many researchers to acquire important information from the data, provide personalized experiences, and develop automated intelligent systems. The widely used algorithms in the field of machine learning such as support vector machine, random forest, and artificial neural networks have been utilized to forecast and categorize future events (Chung & Teo, 2022).

The goal of this study is to create a predictive model for panic disorder using user profile data, which includes demographic, health, and lifestyle factors. To identify patterns and relationships

between these attributes and the presence of panic disorder, machine learning techniques will be used (Nemesure et al., 2021).
The proposed model seeks to provide valuable insights into the factors influencing the likelihood of panic disorder development by analyzing user profiles, potentially paving the way for personalized preventive measures and improved mental health outcomes.

1.2. Aim

To develop a predictive model that can accurately predict whether an individual is likely to be diagnosed with panic disorder based on their user profile information.

1.3. Research Question

To what extent can machine learning models effectively predict the likelihood of an individual being diagnosed with panic disorder using user profile information, and how this information contributes to the predictive accuracy of the model?

The following two hypotheses are based on our research question:
Null Hypothesis - Panic Disorder Diagnosis is not related to Age, Gender, Family History, Personal History, Current Stressors, Symptoms, Severity, Impact on Life, Demographics, Medical History, Psychiatric History, Substance Abuse, Coping Mechanism, Social Support, and Lifestyle Factors.

Alternate Hypothesis - Panic Disorder Diagnosis is related to Age, Gender, Family History, Personal History, Current Stressors, Symptoms, Severity, Impact on Life, Demographics, Medical History, Psychiatric History, Substance Abuse, Coping Mechanism, Social Support, and Lifestyle Factors.

1.4. Purpose

Predictive modeling of panic disorder through machine learning aims to detect and intervene early in individuals at risk, improving mental health outcomes. By analyzing user profiles, personalized treatment plans can be devised, optimizing therapeutic approaches. Resource allocation in healthcare can be enhanced, directing interventions to those with a higher likelihood of developing panic disorder. This approach contributes to destigmatizing mental health issues, providing an objective and data-driven perspective. Additionally, it aids in public health planning, identifying demographic and environmental factors associated with panic disorder. The integration of technology, particularly machine learning, showcases its potential in mental health assessments. Ethical considerations, including data privacy and consent, must be prioritized in projects involving mental health data. Collaboration with mental health professionals is crucial for accurate and ethical model implementation. Ultimately, it seeks to contribute to a more informed, proactive, and compassionate approach to mental health. the results of this project aim to contribute to a more informed, proactive, and compassionate approach to mental health, potentially revolutionizing how we understand, prevent, and address panic disorder.

2. Methodology

2.1 Steps of the Project

- Data Cleaning and Extraction:
  Looking for missing values and tackling them using mode and duplicate values.
  Indexed relevant columns and scaled the data.

- Data Visualization:
  Used Mat plot Lib for correlation matrix.
  Plotted the count graph to understand the target column.
  Box plot to visualize the numeric data vs the target variable.
  Used seaborn for plotting the confusion matrix of the different models.

- Fitting Machine Learning Models:
  We chose Logistic regression as our outcomes were expected to be binary. Decision trees and Random Forest are versatile machine-learning models capable of performing both regression and classification tasks. XGBoost is designed for efficient and scalable training of machine learning models.

- Performing Analysis of the Models:
  Calculated the accuracy of each model and concluded which is the best among them.

- Rejected the Null Hypothesis.

3. Data Collection and Import

One of the most crucial steps in creating a machine-learning model is data collection. After filtering the dataset from Kaggle to the past year, resulting in the finding of the Panic Disorder dataset. Two CSV files were found—one for testing and the other for training. The dataset consists of 16 columns and 12,000 rows overall. Python was utilized to view and comprehend our data.
The files were uploaded into Jupyter Hub, for processing with Python.
The dataset used is: https://www.kaggle.com/datasets/muhammadshahidazeem/panic-disorder-detection-dataset/data

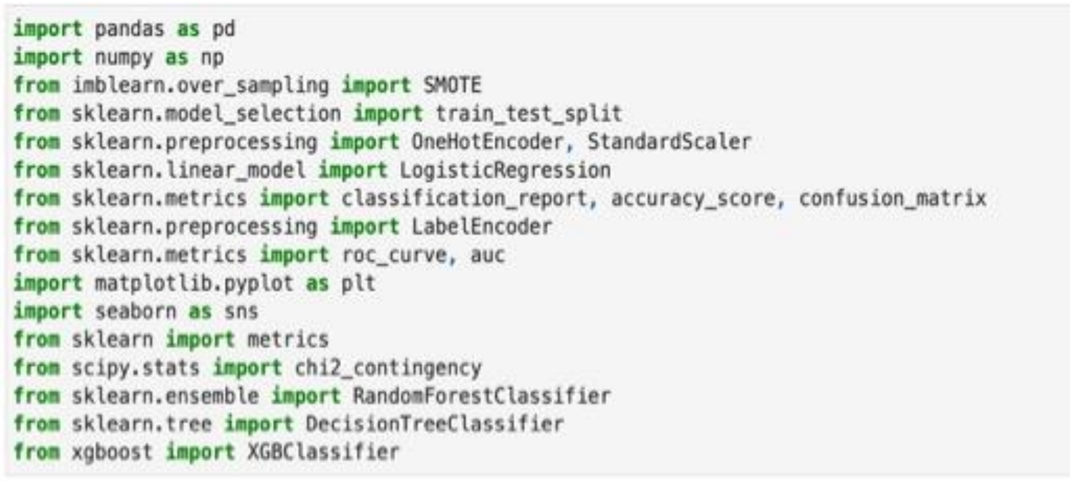
3.1. Panic Disorder Dataset Details

Participant ID: A unique identifier for each participant in the survey or study. Typically, this is a number that has no intrinsic meaning other than identifying a unique individual.
Age: The age of the participant at the time of the survey or study.
Gender: The gender of the participant.
Family History: Information about the presence of panic disorder or other mental health disorders in the participant's family.
Personal History: The personal history of the participant, likely referring to past events or conditions relevant to mental health.

Current Stressors: Current stressors in the participant's life that might affect mental health.
Symptoms: Symptoms reported by the participant
Severity: Symptom severity reported by the participant.
Impact on Life: Measures the degree to which the symptoms affect the participant's daily life, whether at work, in relationships, or in other areas.
Demographics: Demographic information about the participant.
Medical History: The general medical history of the participant.
Psychiatric History: The psychiatric history of the participant.
Substance Use: Information about the participant's use of substances.
Coping Mechanisms: The coping strategies the participant uses to deal with stress or the symptoms of panic disorder.
Social Support: The amount and quality of social support available to the participant, such as friends, family, or support groups.
Lifestyle Factors: Factors related to the participant's lifestyle that might affect mental health, such as sleep patterns, diet, and physical exercise.
Panic Disorder Diagnosis: The diagnosis of panic disorder likely indicates whether the participant has been diagnosed with the disorder or not.

3.2. Python Libraries Imported

Pandas: Pandas is a Python software package designed for data analysis and manipulation. It provides procedures and data structures for working with time series and numerical tables.
NumPy: Large, multi-dimensional arrays and matrices are supported by NumPy, a library for the Python programming language that also offers an extensive range of advanced mathematical operations that may be performed on these arrays.
Scikit-learn Package: Scikit-learn is a free Python machine-learning library. It has a number of algorithms, including XGBoost, Random Forest, Decision Tree, and Logistic Regression.
Confusion Matrix: A confusion matrix is a summary tool used to assess how well a classification algorithm performs. If there are more than two classes in your dataset or if there are unequal numbers of observations in each class, classification accuracy alone may be deceptive.
Classification Report: The model's precision, recall, F1, and support scores are shown by the classification report visualizer.
Accuracy Score: It's a metric that's used to assess how well the classification model performs. It represents the ratio of correctly predicted instances to the total number of instances in a dataset. The accuracy score is often expressed as a percentage.



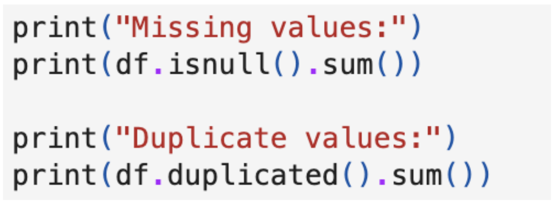
```
import pandas as pd
import numpy as np
from imblearn.over_sampling import SMOTE
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import OneHotEncoder, StandardScaler
from sklearn.linear_model import LogisticRegression
from sklearn.metrics import classification_report, accuracy_score, confusion_matrix
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import roc_curve, auc
import matplotlib.pyplot as plt
import seaborn as sns
from sklearn import metrics
from scipy.stats import chi2_contingency
from sklearn.ensemble import RandomForestClassifier
from sklearn.tree import DecisionTreeClassifier
from xgboost import XGBClassifier
```

4. Data Preprocessing

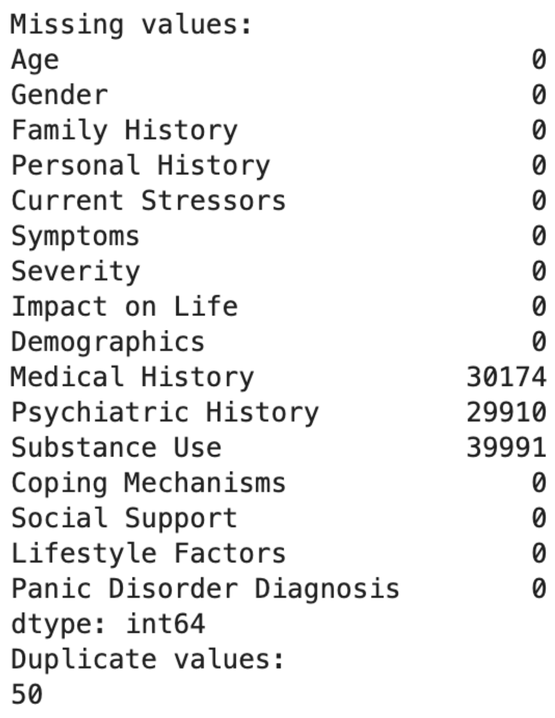
In order to assess and provide useful results, data collection entails gathering task-specific information utilizing specified variables. However, some data might be noisy, which means that some of the values could be erroneous, incomplete, or incorrect. To guarantee correct results, it is therefore imperative to process the data before analysis. Data transformation, data cleaning, and data selection are all included in data pre-processing.
Data pre-processing is the process of preparing raw data so that it may be used with a machine learning model. To construct a machine learning model, this first and crucial step is essential.

4.1. Data cleaning

The performance of our machine learning model relies on the quality of the data it receives. Therefore, an essential preliminary task was to clean the data before using it for model training. In this process, we examined the dataset for any instances of missing values, typically indicated as 'NaN' or 'None.' The absence of data can stem from issues like data corruption or incomplete recording. Managing these missing values is of utmost importance during dataset preprocessing, especially since many machine learning algorithms cannot effectively handle datasets with gaps in information.



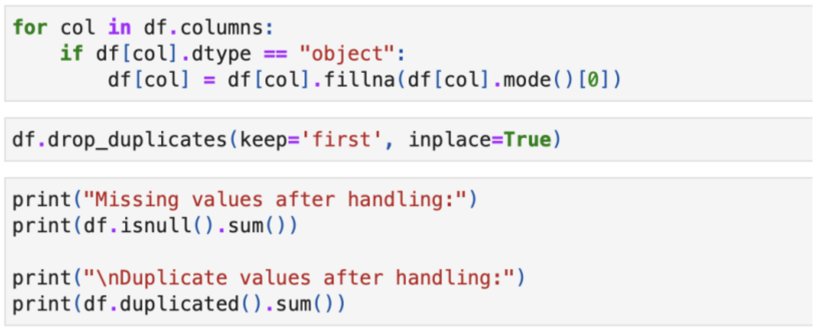
```
print("Missing values:")
print(df.isnull().sum())

print("Duplicate values:")
print(df.duplicated().sum())
```



```
Missing values:
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             30174
Psychiatric History         29910
Substance Use               39991
Coping Mechanisms               0
Social Support                  0
Lifestyle Factors               0
Panic Disorder Diagnosis        0
dtype: int64
Duplicate values:
50
```

Figure: 1 Missing values in the dataset

The figure shows the number of missing values in each column of a patient data set with the highest occurrences found in the 'Medical History,' 'Psychiatric History,' and 'Substance Use' columns. Additionally, 50 duplicate entries were identified. To address these data quality issues, a systematic data handling process was implemented. The script employed imputation methods, specifically filling missing values in categorical ('object') columns with the mode (most frequent value). Subsequently, duplicate entries were removed to ensure data integrity. The final inspection of the dataset demonstrated a significant reduction in missing values and the elimination of duplicate entries, enhancing the overall quality and reliability of the data for subsequent analyses. This systematic approach ensures that the dataset is more robust and suitable for meaningful insights and interpretations in the context of patient-related research or analysis.



```python
for col in df.columns:
    if df[col].dtype == "object":
        df[col] = df[col].fillna(df[col].mode()[0])
```

```python
df.drop_duplicates(keep='first', inplace=True)
```

```python
print("Missing values after handling:")
print(df.isnull().sum())

print("\nDuplicate values after handling:")
print(df.duplicated().sum())
```

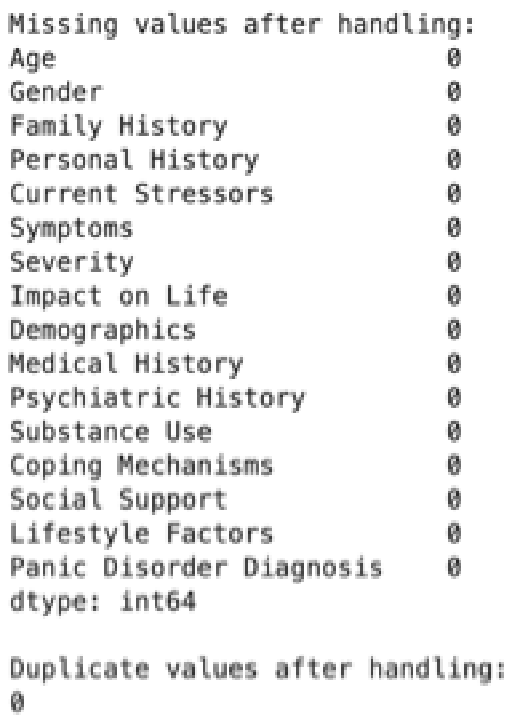

```
Missing values after handling:
Age                         0
Gender                      0
Family History              0
Personal History            0
Current Stressors           0
Symptoms                    0
Severity                    0
Impact on Life              0
Demographics                0
Medical History             0
Psychiatric History         0
Substance Use               0
Coping Mechanisms           0
Social Support              0
Lifestyle Factors           0
Panic Disorder Diagnosis    0
dtype: int64

Duplicate values after handling:
0
```

Figure: 2 Missing values after handling in the dataset

4.2. Data Preprocessing Using Label Encoding

In the process of preparing our data for machine learning tasks, one crucial step is handling categorical variables. Categorical variables are those that represent categories rather than numerical values, and many machine learning algorithms require numerical inputs. To address this, we employed the LabelEncoder from the scikit-learn library to transform categorical data into a format suitable for modeling.
For each column in our Data Frame, we examined the data type. If the data type was identified as "object," indicating a categorical variable, the LabelEncoder was applied to transform the categorical labels into numerical representations. One-hot encoding is a method that transforms categorical variables into binary vectors, providing a more suitable representation for certain machine learning algorithms.

```
categorical_cols = df.select_dtypes(include=['object']).columns
```

```
label_encoder = LabelEncoder()
for column in df.columns:
    if df[column].dtype == "object":
        df[column] = label_encoder.fit_transform(df[column])
```

```
encoder = OneHotEncoder(drop='first', sparse_output=False)
X_encoded = pd.DataFrame(encoder.fit_transform(df[categorical_cols]), columns=encoder.get_feature_names_out(categorical_cols))
```

5. Statistical Analysis
5.1. Chi-square Test

The Chi-square test was used to examine possible significant associations between Panic Disorder and each independent feature in the datasets, aiming to identify key features for subsequent predictive modeling. A significance level of $p < 0.05$ was chosen, indicating the rejection of the null hypothesis and signifying a notable association between the groups. The chi-square contingency test in Python was employed for this analysis.

Variable	p-value	Reject Null
Gender	0.8857650238866819	No
Family History	4.997152155282144e-119	Yes
Personal History	1.0726121649379595e-146	Yes
Current Stressors	0.0	Yes
Symptoms	0.0	Yes
Severity	0.0	Yes
Impact on Life	0.0	Yes
Demographics	6.780394662469932e-29	Yes
Medical History	1.308869350466466e-21	Yes
Psychiatric History	3.002406393432515e-13	Yes
Substance Use	4.1757990269588947e-07	Yes
Coping Mechanisms	5.947016904751294e-136	Yes
Social Support	5.569840505560957e-22	Yes
Lifestyle Factors	0.0	Yes

Table3. Chi-Square Testing Results

The variable "Gender" was excluded from further analysis due to its non-significant p-value ($p = 0.886$), suggesting that gender differences may not be a significant predictor for panic disorder in this study.

6. Data Visualisation
6.1. Count Plot

To gain insights into the distribution of classes within our target variable, 'Panic Disorder Diagnosis', we employed data visualization techniques. Specifically, we utilized the seaborn library to create a count plot, offering a clear representation of the frequency of each class.

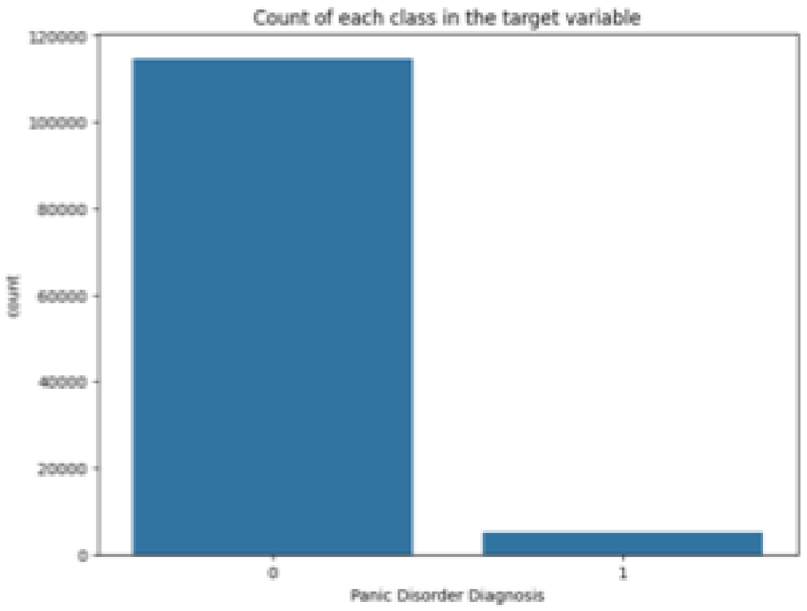
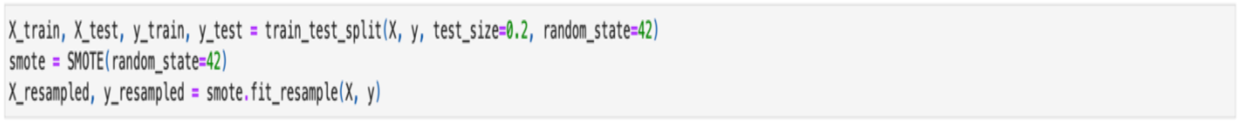


Figure4. *Figure5. Count Plot for Panic Disorder Diagnosis*

The generated count plot visually illustrates the distribution of classes in the 'Panic Disorder Diagnosis' variable. Each bar on the plot signifies a distinct class, and the bar's height corresponds to the frequency or count of occurrences of that particular class within the dataset.

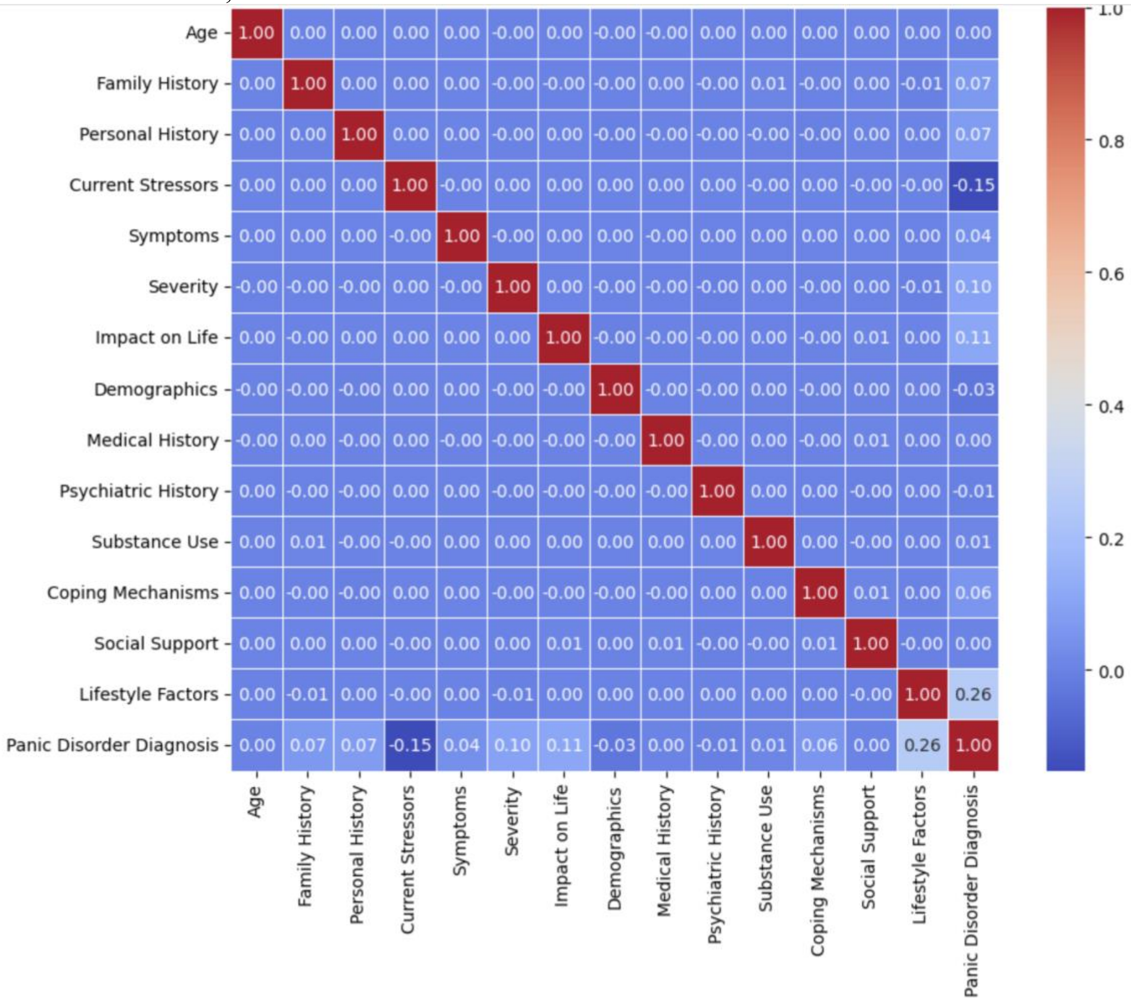
This visualization is instrumental in understanding the balance or imbalance in the distribution of classes, which is essential information for machine learning tasks, especially for classification problems. An imbalanced distribution may influence the performance of classification models, and this plot serves as a valuable exploratory tool in our data analysis process.
This dataset is highly imbalanced. Due to the potential impact of class imbalance on the performance of classification models, we conducted a **Synthetic Minority Over-sampling Technique (SMOTE)** analysis to address this issue. SMOTE involves generating synthetic instances of the minority class to balance the class distribution.


```
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)
smote = SMOTE(random_state=42)
X_resampled, y_resampled = smote.fit_resample(X, y)
```

The provided code implements oversampling through the SMOTE algorithm to rectify class imbalance in the training data. This process involves addressing the disparity in class distribution within the original dataset by oversampling the minority class. By doing so, the code aims to enhance the performance of machine learning models, mitigating bias toward the majority class and ultimately improving overall accuracy and predictive capability.

6.2. Correlation Matrix (Heatmap)

A correlation matrix is a tabular representation illustrating the correlation coefficients between pairs of variables, providing a measure of the strength of the relationship between them. The correlation coefficient, ranging from -1 to 1, signifies a perfect negative correlation at -1, a perfect positive correlation at 1, and no correlation at 0.



	Age	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
Age	1.00	0.00	0.00	0.00	0.00	-0.00	0.00	-0.00	-0.00	0.00	0.00	0.00	0.00	0.00	0.00
Family History	0.00	1.00	0.00	0.00	0.00	-0.00	-0.00	-0.00	0.00	-0.00	0.01	-0.00	0.00	-0.01	0.07
Personal History	0.00	0.00	1.00	0.00	0.00	-0.00	0.00	-0.00	-0.00	-0.00	-0.00	-0.00	0.00	0.00	0.07
Current Stressors	0.00	0.00	0.00	1.00	-0.00	0.00	0.00	0.00	0.00	0.00	-0.00	0.00	-0.00	-0.00	-0.15
Symptoms	0.00	0.00	0.00	-0.00	1.00	-0.00	0.00	0.00	-0.00	0.00	0.00	0.00	0.00	0.00	0.04
Severity	-0.00	-0.00	-0.00	0.00	-0.00	1.00	0.00	-0.00	-0.00	-0.00	0.00	-0.00	0.00	-0.01	0.10
Impact on Life	0.00	-0.00	0.00	0.00	0.00	0.00	1.00	-0.00	-0.00	-0.00	0.00	-0.00	0.01	0.00	0.11
Demographics	-0.00	-0.00	-0.00	0.00	0.00	-0.00	-0.00	1.00	-0.00	-0.00	0.00	-0.00	0.00	0.00	-0.03
Medical History	-0.00	0.00	-0.00	0.00	-0.00	-0.00	-0.00	-0.00	1.00	-0.00	0.00	-0.00	0.01	0.00	0.00
Psychiatric History	0.00	-0.00	-0.00	0.00	0.00	-0.00	-0.00	-0.00	-0.00	1.00	0.00	0.00	-0.00	0.00	-0.01
Substance Use	0.00	0.01	-0.00	-0.00	0.00	0.00	0.00	0.00	0.00	0.00	1.00	0.00	-0.00	0.00	0.01
Coping Mechanisms	0.00	-0.00	-0.00	0.00	0.00	-0.00	-0.00	-0.00	-0.00	0.00	0.00	1.00	0.01	0.00	0.06
Social Support	0.00	0.00	0.00	-0.00	0.00	0.00	0.01	0.00	0.01	-0.00	-0.00	0.01	1.00	-0.00	0.00
Lifestyle Factors	0.00	-0.01	0.00	-0.00	0.00	-0.01	0.00	0.00	0.00	0.00	0.00	0.00	-0.00	1.00	0.26
Panic Disorder Diagnosis	0.00	0.07	0.07	-0.15	0.04	0.10	0.11	-0.03	0.00	-0.01	0.01	0.06	0.00	0.26	1.00

Figure: 6 Heatmap

The generated heatmap provides a visual depiction of the correlation matrix, where the color of each cell reflects the strength and direction of the correlation between variable pairs. The color spectrum, spanning from cool tones for negative correlation to warm tones for positive correlation, aids in discerning patterns of association.

Analysis of the correlation matrix reveals that the diagnosis of panic disorder exhibits strong correlations with personal history, current stressors, symptoms, severity, and impact on life. Additionally, there are moderate correlations with age, gender, family history, psychiatric history, substance use, coping mechanisms, social support, and lifestyle factors.

6.3. Boxplot Analysis

```
In [18]: plt.figure(figsize=(12, 8))
         for i, col in enumerate(df.select_dtypes(include='number').columns):
             plt.subplot(2, 3, min(i+1, 6))
             sns.boxplot(x='Panic Disorder Diagnosis', y=col, data=df)
         plt.tight_layout()
```

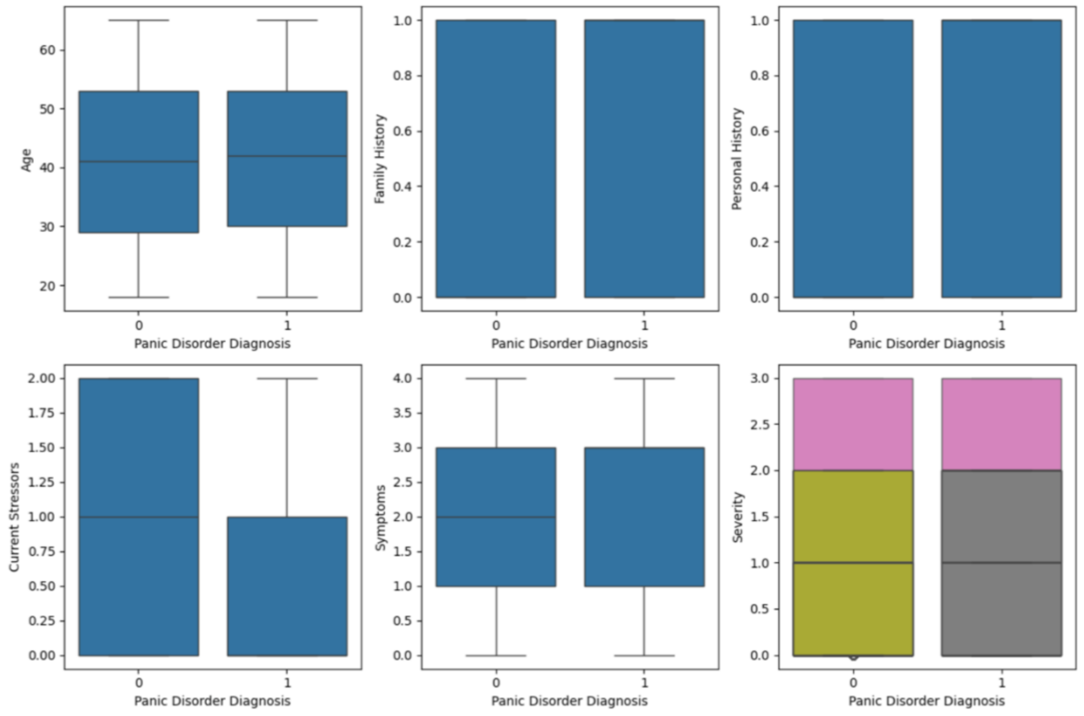


Figure: 7 Box Plot Analysis

In this code snippet, a grid of boxplots is generated using the ‘matplotlib’ and ‘seaborn’ libraries in Python. The code iterates over the numerical columns of a DataFrame (df) that are presumably loaded with data, filtering them using select_dtypes (include='number'). For each numerical column, the code creates a subplot using plt.subplot (2, 3, min (i+1, 6)), organizing the subplots in a 2x3 grid. The min (i+1, 6) is used to ensure that the loop does not create more than six subplots, as there might be more than six numerical columns.
Within each subplot, a boxplot is drawn using sns.boxplot(x='Panic Disorder Diagnosis', y=col, data=df). This visual representation displays the distribution of data for each numerical column based on the values in the 'Panic Disorder Diagnosis' column. Each boxplot provides information about the median, quartiles, and potential outliers for the corresponding numerical variable.
The plt.tight_layout() call is used to improve the layout of the subplots, preventing them from overlapping and ensuring better readability.

7. Machine Learning and Model Testing

We have trained four different machine learning models, namely:

1. Logistic Regression
2. Random Forest Classifier
3. Decision Tree
4. XGBoost

These algorithms were used to predict and analyze results and compare these algorithms based on their performance.

7.1. LOGISTIC REGRESSION

Logistic Regression, a commonly employed classification algorithm, is designed to model the likelihood of a binary outcome. Logistic regression model is created using the LogisticRegression() function.

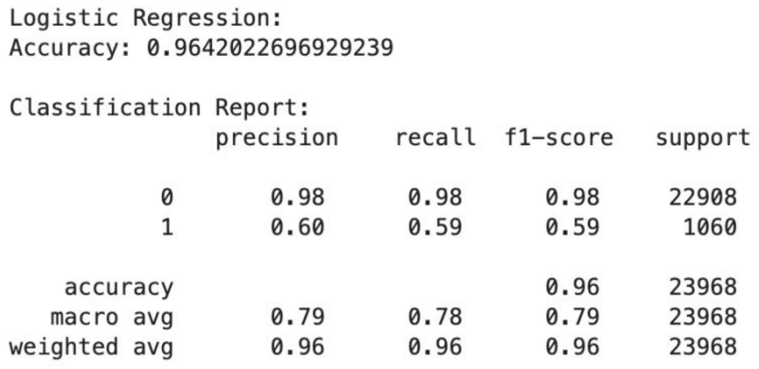
This model was fit to the resampled and scaled training data (X_resampled_scaled) with corresponding target labels (y_resampled).
To implement the Logistic Regression using Python, five steps:

- Data Pre-processing step
- Fitting Logistic Regression to the Training set
- Predicting the test result
- Test accuracy of the result (Creation of Confusion matrix)
- Visualizing the test set result.

0.9642, or 96.42%, indicates the overall proportion of correct predictions made by the model. This is a high accuracy, suggesting the model is generally effective in distinguishing between the two classes.

Classification Report

The classification report provides a more detailed breakdown of the model's performance for each class. The report provides various metrics for each class (in this case, class 0, and class 1) and overall metrics.



```
Logistic Regression:
Accuracy: 0.9642022696929239

Classification Report:
              precision    recall  f1-score   support

           0       0.98      0.98      0.98     22908
           1       0.60      0.59      0.59      1060

    accuracy                           0.96     23968
   macro avg       0.79      0.78      0.79     23968
weighted avg       0.96      0.96      0.96     23968
```

Figure: 8 Classification Report for Logistic Regression

Precision: Precision, defined as the ratio of correctly predicted positive observations to the total predicted positives, is 0.98 for class 0, indicating a 98% accuracy when predicting this class. However, for class 1, precision is 0.60, suggesting a lower accuracy in predictions for this class.

Recall: Recall, also known as sensitivity or true positive rate, represents the ratio of correctly predicted positive observations to all observations in the actual class. For class 0, recall is 0.98, indicating the model captures 98% of instances for this class. In contrast, for class 1, recall is 0.59, signifying the model captures 59% of instances for this class.
F1-Score: The F1-score, a weighted average of precision and recall that considers both false positives and false negatives, is 0.98 for class 0 and 0.59 for class 1.
This report reveals that the model performed well in the majority class (class 0) with high precision, recall, and F1-score. However, the performance for the minority class (class 1) was lower, indicating that the model struggled to accurately identify instances belonging to this class.

Receiver Operating Characteristic (ROC) Curve

The ROC curve visually displays how the true positive rate (TPR) and false positive rate (FPR) trade-off at different thresholds, while the area under the ROC curve (AUC) serves as a metric for the model's capacity to distinguish between classes. The AUC, calculated using the roc_curve and auc functions, was employed to assess the model's overall performance. When plotted alongside a diagonal line representing random guessing using matplotlib, the Logistic Regression model in this instance achieved an AUC of 0.98, denoting strong discriminative ability.

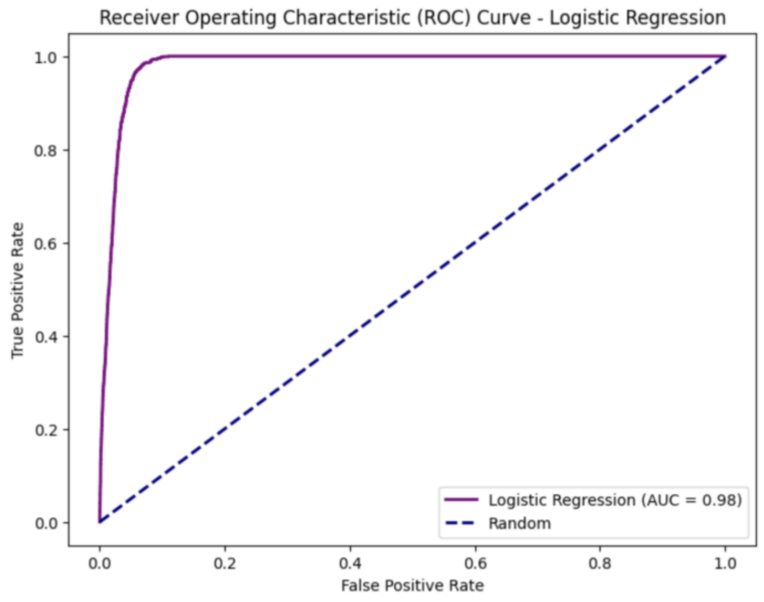


Figure: 9 ROC Curve for Logistic Regression

Confusion Matrix

A confusion matrix was created to illustrate how well the model performed in terms of accurately predicting true positives, true negatives, false positives, and false negatives.

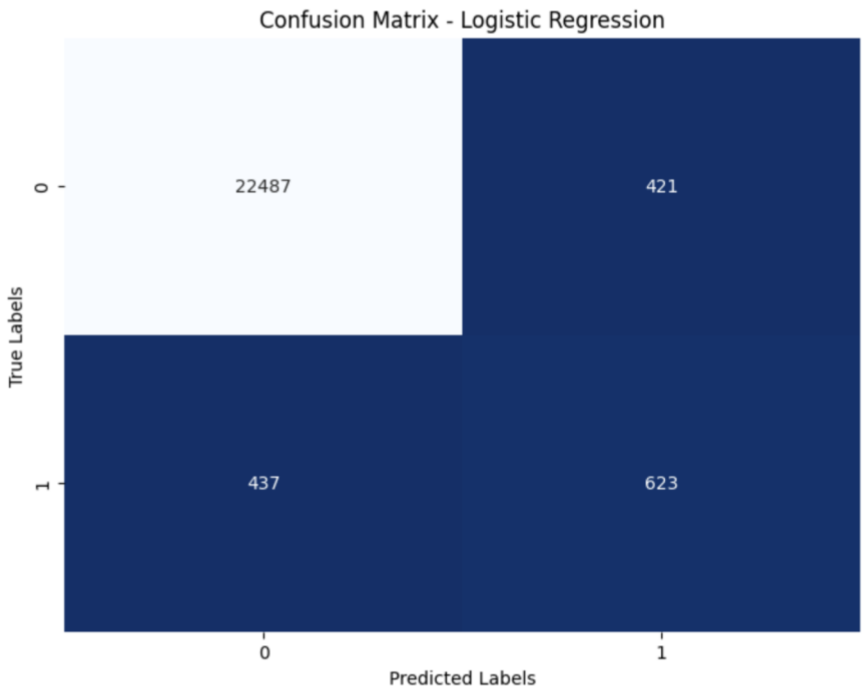


Figure: 10 Confusion Matrix for Logistic Regression

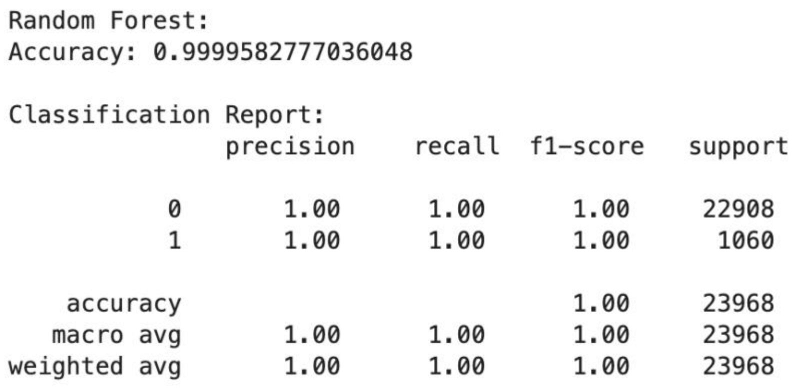
True Positives (TP): 623 - Correctly identified positive cases (panic disorder diagnosed).
False Positives (FP): 421 - Incorrectly identified positive cases (healthy individuals diagnosed with panic disorder).
True Negatives (TN): 224887 - Correctly identified negative cases (healthy individuals diagnosed as healthy).
False Negatives (FN): 437 - Incorrectly identified negative cases (individuals with panic disorder diagnosed as healthy).

7.2. RANDOM FOREST CLASSIFIER

Random Forest stands as a versatile machine-learning technique proficient in handling regression and classification assignments. Additionally, it encompasses procedures for dimensional reduction, addressing missing values, outlier values, and other crucial steps in data exploration, exhibiting commendable performance. This method belongs to the ensemble learning category, wherein a collection of weak models collaborates to create a robust and effective model.

Classification Report

The Random Forest model achieved exceptional performance on the test dataset, demonstrating an outstanding accuracy of 99.99%. This indicates that the model correctly classified nearly all instances, showcasing its superior ability to learn complex patterns and relationships within the data. The classification report further emphasizes this remarkable performance.



```
Random Forest:
Accuracy: 0.9999582777036048

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     22908
           1       1.00      1.00      1.00      1060

    accuracy                           1.00     23968
   macro avg       1.00      1.00      1.00     23968
weighted avg       1.00      1.00      1.00     23968
```

Figure: 11 Classification Report for Random Forest

This report reveals an ideal scenario where the model perfectly identified all instances of both the majority and minority class. This result suggests that the Random Forest model effectively learned the underlying structure of the data and accurately captured the true relationships between features and outcomes.

ROC Curve

The AUC of the ROC curve is 0.98, indicating that the classifier has excellent diagnostic ability.

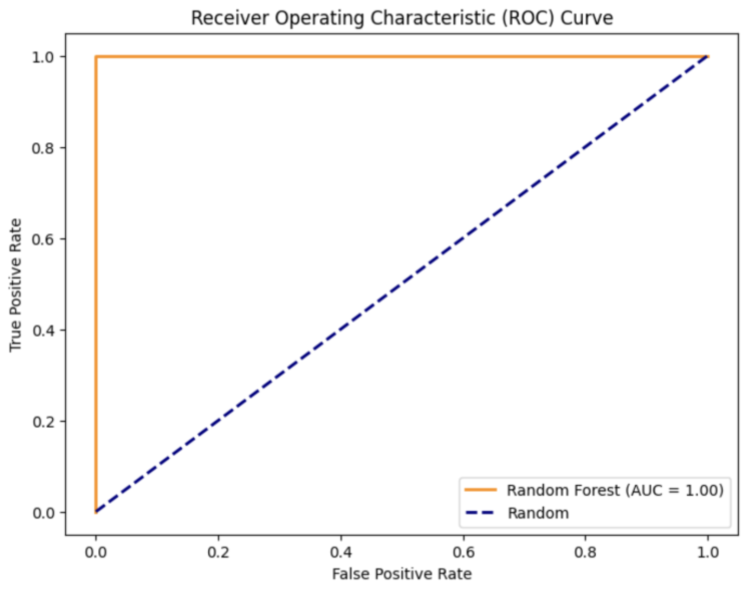


Figure: 12 ROC Curve for Random Forest

Confusion Matrix

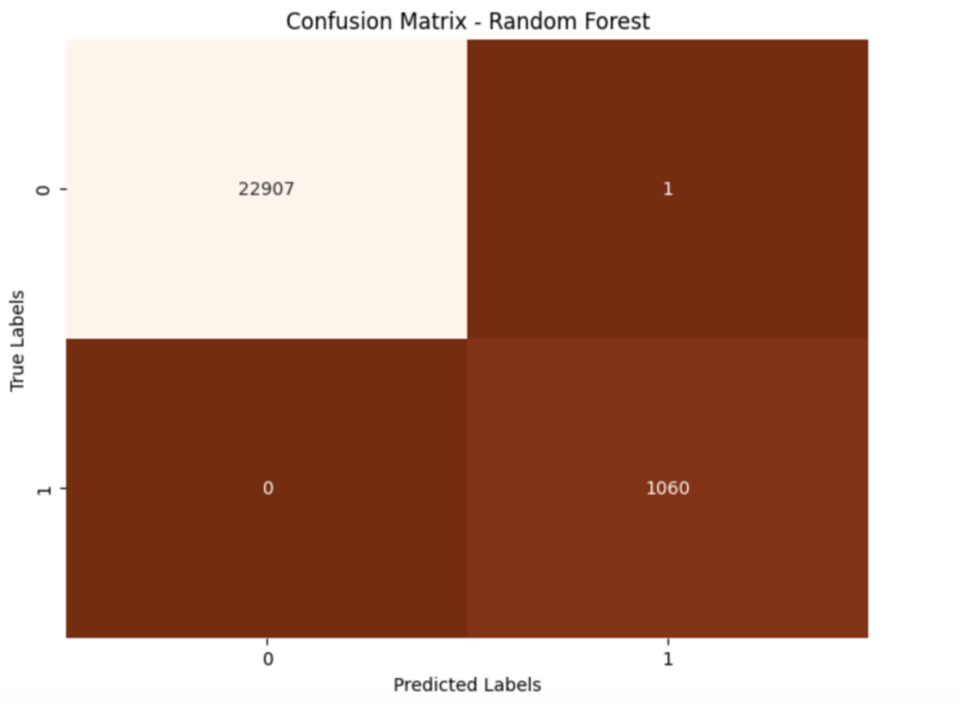
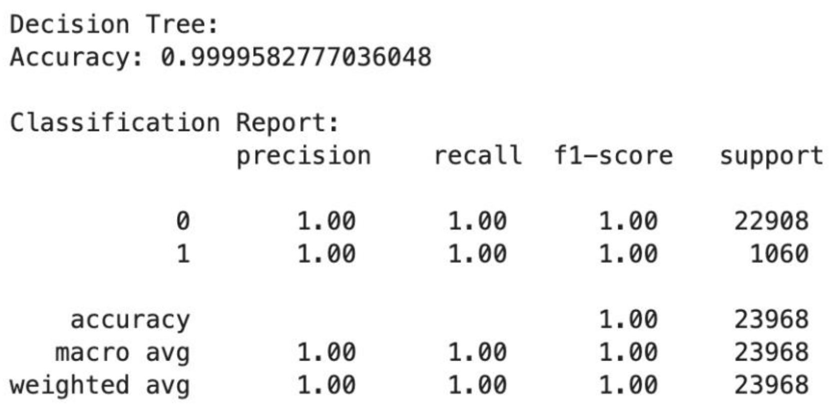


Figure: 13 Confusion Matrix for Random Forest

Based on the confusion matrix for panic disorder classification, the following can be observed:
True positives (TP): 1060 cases of panic disorder were correctly predicted.
True negatives (TN): 22,908 cases without panic disorder were correctly predicted.
False positives (FP): 1 case without panic disorder was incorrectly predicted as having panic disorder.
False negatives (FN): 0 cases of panic disorder were incorrectly predicted as not having panic disorder.
These results suggest that the classifier has excellent performance for panic disorder classification, with a precision of 1.00, recall of 1.00, and F1-score of 1.00. This means that the classifier can accurately identify instances of panic disorder with a very high degree of confidence.

7.3. DECISION TREE

A decision tree serves as a supervised machine learning algorithm applicable to both classification and regression tasks, presenting a straightforward and efficient visual representation for decision-making processes.



```
Decision Tree:
Accuracy: 0.9999582777036048

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     22908
           1       1.00      1.00      1.00      1060

    accuracy                           1.00     23968
   macro avg       1.00      1.00      1.00     23968
weighted avg       1.00      1.00      1.00     23968
```

Figure: 14 Classification Report for Decision Tree

The overall accuracy of the model is extremely high, with a value of approximately 99.996%. This implies that the model correctly predicted the class labels for almost all instances in the test dataset.

Classification Report

Precision, Recall, and F1-Score for Class 0 (Negative Class):
Precision: 100% - All instances predicted as class 0 were actually class 0.
Recall: 100% - The model identified all instances of class 0 correctly.
F1-Score: 100% - The harmonic mean of precision and recall is also perfect.
Precision, Recall, and F1-Score for Class 1 (Positive Class):
Precision: 100% - All instances predicted as class 1 were actually class 1.
Recall: 100% - The model identified all instances of class 1 correctly.
F1-Score: 100% - The harmonic mean of precision and recall is also perfect.

ROC Curve

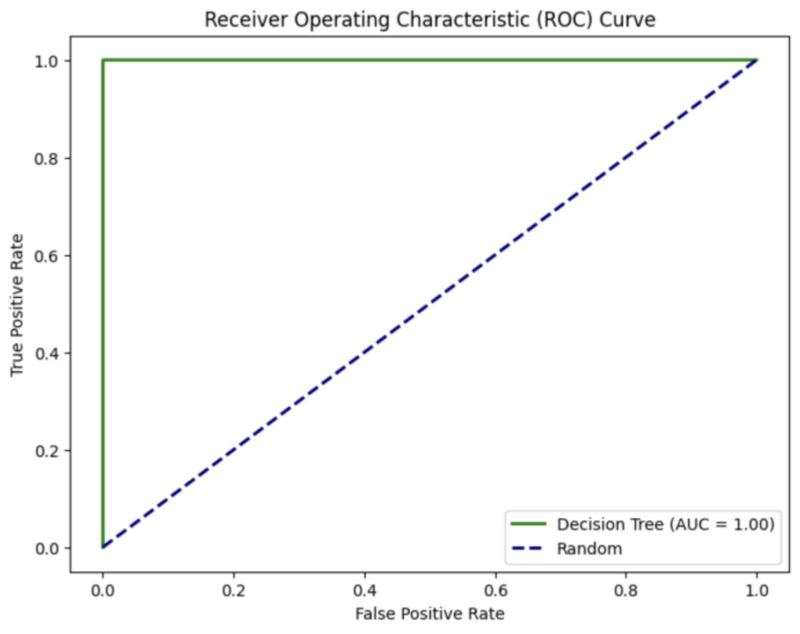


Figure: 15 ROC Curve for Decision Tree

The ROC curve for the decision tree model in the image shows that the model has perfect performance, with an AUC of 1.00. This means that the model can perfectly distinguish between positive and negative cases at all classification thresholds. In other words, the ROC curve interprets the relationship between the false positive rate and the decision tree. It shows that the decision tree model can correctly classify all positive cases without incorrectly classifying any negative cases.

Confusion Matrix

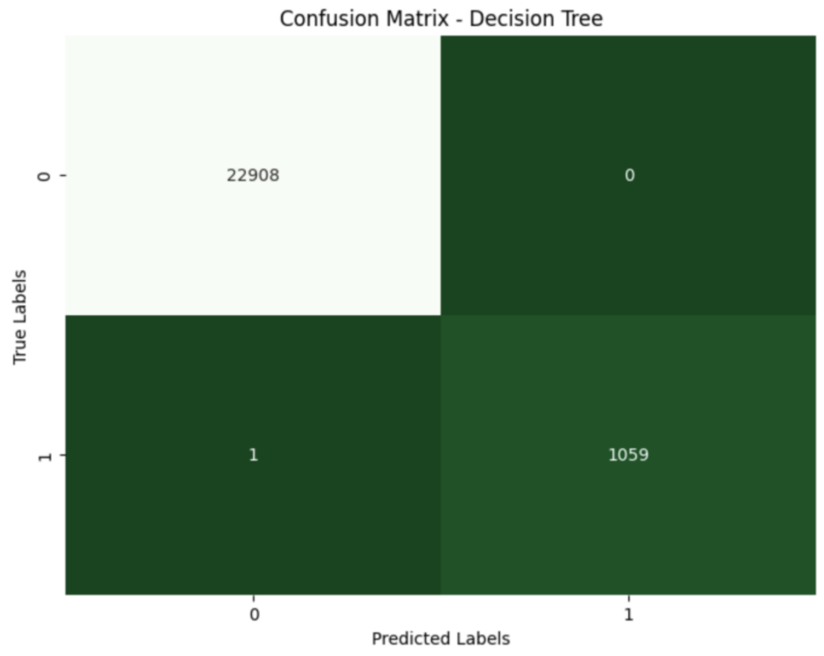


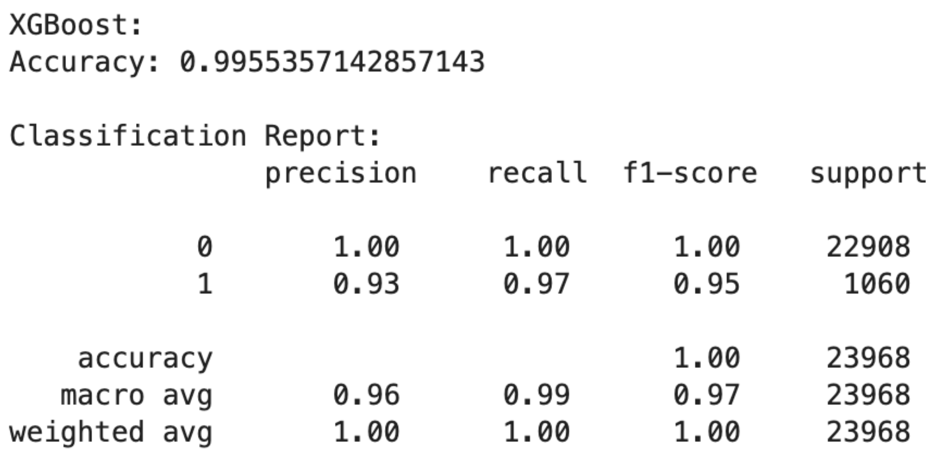
Figure: 16 Confusion Matrix for Decision Tree

True Positive (TP): The number of instances that are actually positive (class 1) and are correctly predicted as positive. In this case, there are 1,059 instances.
True Negative (TN): The number of instances that are actually negative (class 0) and are correctly predicted as negative. In this case, there are 22,908 instances.
False Positive (FP): The number of instances that are actually negative but are incorrectly predicted as positive. In this case, there are 0 instances. (This is an ideal scenario, indicating no false positives.)
False Negative (FN): The number of instances that are actually positive but are incorrectly predicted as negative. In this case, there is 1 instance.

7.4. XGBOOST

XGBoost is a specialized distributed gradient boosting library crafted for effective and scalable training of machine learning models. It operates as an ensemble learning technique, amalgamating predictions from various weak models to generate a more robust prediction.

The XGBoost model achieved an overall accuracy of about 99.55%, signifying its accurate prediction of class labels for most instances in the test dataset.



```
XGBoost:
Accuracy: 0.9955357142857143

Classification Report:
              precision    recall  f1-score   support

           0       1.00      1.00      1.00     22908
           1       0.93      0.97      0.95      1060

    accuracy                           1.00     23968
   macro avg       0.96      0.99      0.97     23968
weighted avg       1.00      1.00      1.00     23968
```

Figure: 17 Classification Report for XGBoost

Classification Report

<u>Precision, Recall, and F1-Score for Class 0 (Negative Class):</u>
Precision: 100% - All instances predicted as class 0 were actually class 0.
Recall: 100% - The model identified all instances of class 0 correctly.
F1-Score: 100% - The harmonic mean of precision and recall is perfect.
<u>Precision, Recall, and F1-Score for Class 1 (Positive Class):</u>
Precision: 93% - Of all instances predicted as class 1, 93% were actually class 1.
Recall: 97% - The model identified 97% of the instances of class 1 correctly.
F1-Score: 95% - The harmonic mean of precision and recall is quite high.

ROC Curve

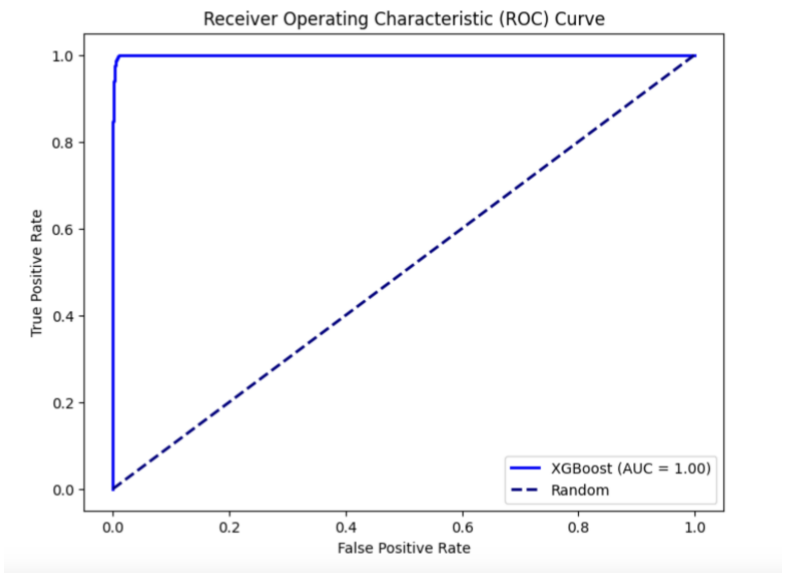


Figure: 18 ROC Curve for XGBoost

The ROC curve shows that the model can predict panic disorder with high accuracy. The AUC of 0.92 means that the model can correctly distinguish between patients with and without panic disorder in 92% of cases. This impressive performance indicates the model's potential utility in assisting clinicians with panic disorder diagnosis. This implies the model's capability to aid clinicians in diagnosing panic disorder. However, it is important to be aware that the model is slightly less sensitive than specific, and it may miss some patients who do have panic disorder.

Confusion Matrix

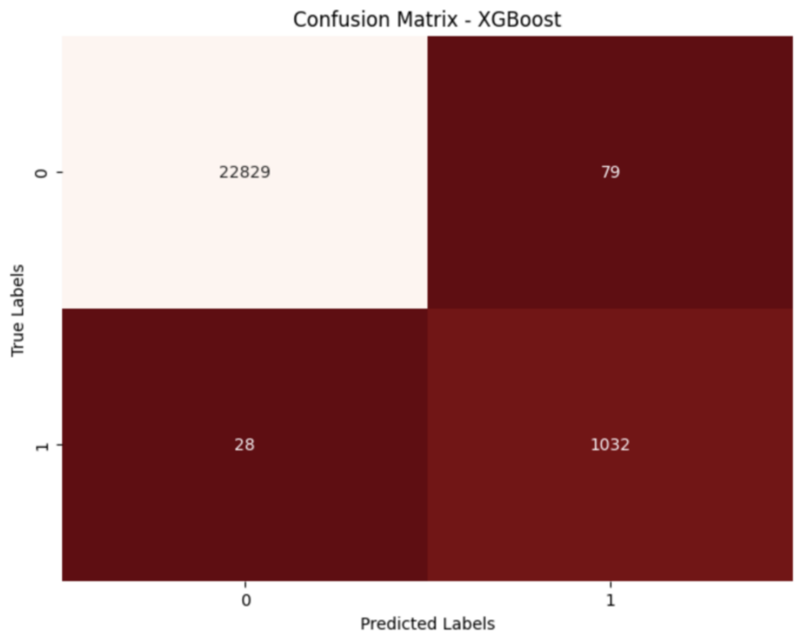


Figure: 19 Confusion Matrix for XGBoost

True Negative (TN): 22,829 instances were correctly predicted as not having panic disorder.
False Positive (FP): 79 instances were incorrectly predicted as having panic disorder when they don't.
False Negative (FN): 28 instances were incorrectly predicted as not having panic disorder when they do.
True Positive (TP): 1,032 instances were correctly predicted as having panic disorder.

8. Results

<u>Evaluating Model Performance</u>
The decision to choose a particular model depends on the specific requirements of the application. Random Forest and Decision Tree demonstrate exceptional performance, especially in scenarios where precision, recall, and F1-score are critical. Logistic Regression and XGBoost provide a balance between accuracy and interpretability.

Models	**Logistic Regression**	**Random Forest**	**XGBoost**	**Decision Tree**
Accuracy	0.964	0.999	0.995	0.999

F1-score	0.59	1.0	0.95	1
Precision	0.60	1.0	0.93	1
Recall	0.59	1.0	0.97	1
AUC	0.98	0.886948	0.9856	0.99955

Table: 20 Evaluating Model Performance

9. Hypothesis Testing

- The chi-square test was employed to test the hypothesis due to the presence of categorical data in the dataset.
- We rejected the null hypothesis as all the factors expect gender had a significant impact on Panic Disorder Diagnosis.
- Based on the p values the factors that are highly risky for Panic Disorder are:
    - Current Stressors
    - Severity
    - Impact on Life
    - Symptoms
    - Lifestyle Factors

10. Conclusions

- Hypothesis testing supported the rejection of the null hypothesis.
- All models performed exceptionally well, but Random Forest and Decision Tree achieved the highest accuracy, while Logistic Regression and XGBoost have slightly lower accuracy.
- Random Forest and Decision Tree models are overfitting the data as indicated by their high accuracy and perfect precision, recall, and F1-score.
- Logistic Regression, while having a lower accuracy, provides a more balanced view with reasonable precision, recall, and F1-score.
- Consideration should be given to the trade-off between model complexity and interpretability.
- Further fine-tuning and validation on an independent dataset may be necessary to confirm the robustness of the models.

11. Limitations

- Data Constraints: The quality and quantity of the dataset may impact the predictive model's accuracy and generalizability. Incomplete or biased data can lead to limitations in capturing diverse user profiles and panic disorder manifestations. This was combatted by SMOTE analysis.

- Model Interpretability: Complex models, such as XGBoost, may lack interpretability, posing challenges in understanding the rationale behind predictions. This limitation may hinder user and professional trust in the model's outcomes.

- Algorithmic Sensitivity: Sensitivity to hyperparameter tuning in machine learning algorithms can be a limitation. Finding the optimal configuration for model performance is a complex task that may affect the reliability of predictions.

- Real-world Applicability: The model's generalization to real-world scenarios may be limited. External factors influencing panic disorder, like life events or changes in societal dynamics, may not be fully captured in the dataset.

12. Project Challenges and Successes

- Feature Engineering Complexity: Selecting relevant features and avoiding overfitting or underfitting is challenging. The complexity of user profiles and the dynamic nature of mental health add difficulty in determining the most influential factors.

- Privacy and Confidentiality Concerns: Safeguarding user privacy while working with sensitive mental health data presents a constant challenge. Maintaining the confidentiality of individuals' information requires meticulous handling throughout the project.

- Algorithmic and Computational Hurdles: The computational intensity of training models like XGBoost can be challenging, particularly with limited resources. Efficiently managing computational demands is crucial for scalability and practical implementation.

- Dynamic Nature of Mental Health: The evolving nature of mental health conditions, including panic disorder, poses challenges in creating a model that remains relevant over time. Regular updates and adaptations may be necessary to accommodate emerging patterns.

- Ensuring Model Robustness: Model validation and ensuring robustness against various scenarios are ongoing challenges. The unpredictability of mental health conditions requires a thorough examination of the model's performance in diverse situations.

- Model Accuracy and Selection: The project successfully applied and analyzed multiple machine learning models, leading to the identification of the best-performing model based on accuracy. This success contributes to the reliability of predictions.

- Comprehensive Data Visualization: Effective use of Matplotlib and Seaborn for visualizing data, including correlation matrices and confusion matrices, demonstrates a thorough understanding of the dataset and the models' performance.

- Null Hypothesis Rejection: Successfully rejecting the null hypothesis indicates that the developed predictive model provides meaningful insights, contributing to the advancement of mental health analytics.

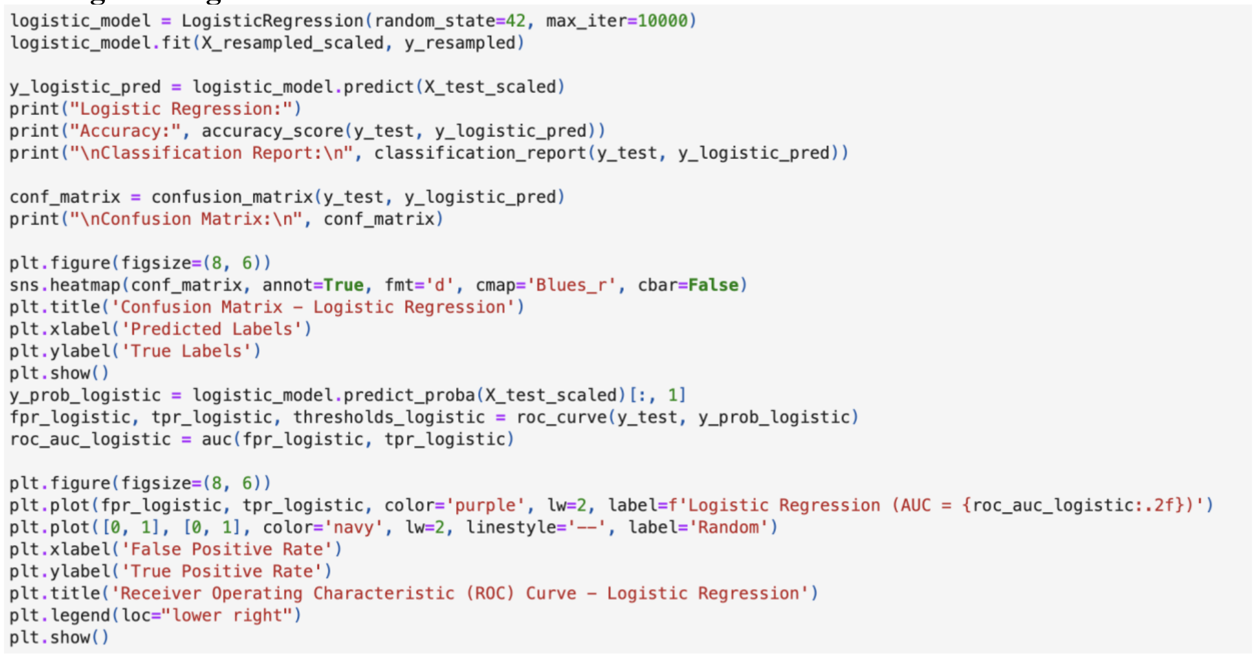
- Contributions to Early Detection: The project's focus on proactive identification and targeted support for individuals at risk of panic disorder is a success, contributing positively to early detection and intervention in mental health.

- Methodological Rigor: The methodology, covering data cleaning, extraction, visualization, and model fitting, reflects a systematic and rigorous approach to predictive modeling in the context of mental health.

Appendix

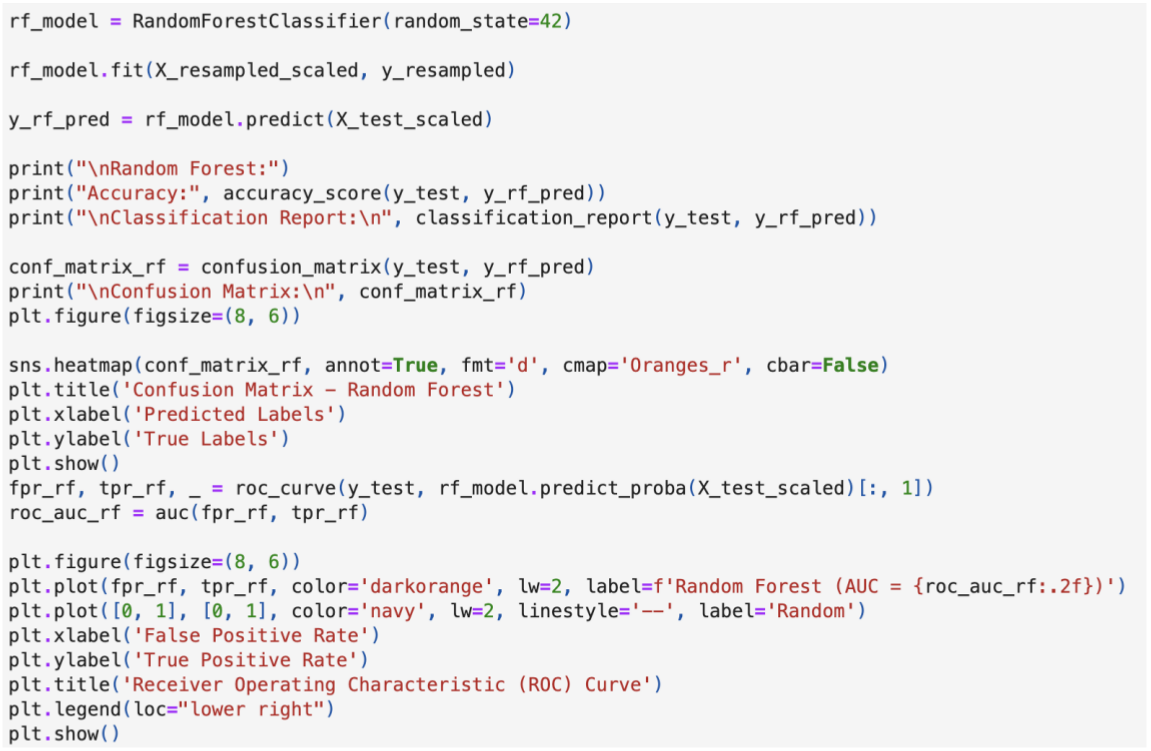
Codes for the models, confusion matrix, and ROC curve

1. Logistic Regression



```
logistic_model = LogisticRegression(random_state=42, max_iter=10000)
logistic_model.fit(X_resampled_scaled, y_resampled)

y_logistic_pred = logistic_model.predict(X_test_scaled)
print("Logistic Regression:")
print("Accuracy:", accuracy_score(y_test, y_logistic_pred))
print("\nClassification Report:\n", classification_report(y_test, y_logistic_pred))

conf_matrix = confusion_matrix(y_test, y_logistic_pred)
print("\nConfusion Matrix:\n", conf_matrix)

plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix, annot=True, fmt='d', cmap='Blues_r', cbar=False)
plt.title('Confusion Matrix - Logistic Regression')
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')
plt.show()
y_prob_logistic = logistic_model.predict_proba(X_test_scaled)[:, 1]
fpr_logistic, tpr_logistic, thresholds_logistic = roc_curve(y_test, y_prob_logistic)
roc_auc_logistic = auc(fpr_logistic, tpr_logistic)

plt.figure(figsize=(8, 6))
plt.plot(fpr_logistic, tpr_logistic, color='purple', lw=2, label=f'Logistic Regression (AUC = {roc_auc_logistic:.2f})')
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--', label='Random')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve - Logistic Regression')
plt.legend(loc="lower right")
plt.show()
```

2. Random Forest



```
rf_model = RandomForestClassifier(random_state=42)

rf_model.fit(X_resampled_scaled, y_resampled)

y_rf_pred = rf_model.predict(X_test_scaled)

print("\nRandom Forest:")
print("Accuracy:", accuracy_score(y_test, y_rf_pred))
print("\nClassification Report:\n", classification_report(y_test, y_rf_pred))

conf_matrix_rf = confusion_matrix(y_test, y_rf_pred)
print("\nConfusion Matrix:\n", conf_matrix_rf)
plt.figure(figsize=(8, 6))

sns.heatmap(conf_matrix_rf, annot=True, fmt='d', cmap='Oranges_r', cbar=False)
plt.title('Confusion Matrix - Random Forest')
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')
plt.show()
fpr_rf, tpr_rf, _ = roc_curve(y_test, rf_model.predict_proba(X_test_scaled)[:, 1])
roc_auc_rf = auc(fpr_rf, tpr_rf)

plt.figure(figsize=(8, 6))
plt.plot(fpr_rf, tpr_rf, color='darkorange', lw=2, label=f'Random Forest (AUC = {roc_auc_rf:.2f})')
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--', label='Random')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve')
plt.legend(loc="lower right")
plt.show()
```

3. Decision Tree

```python
dt_model = DecisionTreeClassifier(random_state=42)

dt_model.fit(X_resampled_scaled, y_resampled)

y_dt_pred = dt_model.predict(X_test_scaled)

print("\nDecision Tree:")
print("Accuracy:", accuracy_score(y_test, y_dt_pred))
print("\nClassification Report:\n", classification_report(y_test, y_dt_pred))

conf_matrix_dt = confusion_matrix(y_test, y_dt_pred)
print("\nConfusion Matrix:\n", conf_matrix_dt)

plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix_dt, annot=True, fmt='d', cmap='Greens_r', cbar=False)
plt.title('Confusion Matrix - Decision Tree')
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')
plt.show()

fpr_dt, tpr_dt, _ = roc_curve(y_test, dt_model.predict_proba(X_test_scaled)[:, 1])
roc_auc_dt = auc(fpr_dt, tpr_dt)

plt.figure(figsize=(8, 6))
plt.plot(fpr_dt, tpr_dt, color='green', lw=2, label=f'Decision Tree (AUC = {roc_auc_dt:.2f})')
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--', label='Random')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve')
plt.legend(loc="lower right")
plt.show()
```

4. XGBoost

```python
from xgboost import XGBClassifier

xgb_model = XGBClassifier()

xgb_model.fit(X_resampled_scaled, y_resampled)

y_xgb_pred = xgb_model.predict(X_test_scaled)

print("\nXGBoost:")
print("Accuracy:", accuracy_score(y_test, y_xgb_pred))
print("\nClassification Report:\n", classification_report(y_test, y_xgb_pred))

conf_matrix_xgb = confusion_matrix(y_test, y_xgb_pred)
print("\nConfusion Matrix:\n", conf_matrix_xgb)

plt.figure(figsize=(8, 6))
sns.heatmap(conf_matrix_xgb, annot=True, fmt='d', cmap='Reds_r', cbar=False)
plt.title('Confusion Matrix - XGBoost')
plt.xlabel('Predicted Labels')
plt.ylabel('True Labels')
plt.show()

fpr_xgb, tpr_xgb, _ = roc_curve(y_test, xgb_model.predict_proba(X_test_scaled)[:, 1])
roc_auc_xgb = auc(fpr_xgb, tpr_xgb)

plt.figure(figsize=(8, 6))
plt.plot(fpr_xgb, tpr_xgb, color='blue', lw=2, label=f'XGBoost (AUC = {roc_auc_xgb:.2f})')
plt.plot([0, 1], [0, 1], color='navy', lw=2, linestyle='--', label='Random')
plt.xlabel('False Positive Rate')
plt.ylabel('True Positive Rate')
plt.title('Receiver Operating Characteristic (ROC) Curve')
plt.legend(loc="lower right")
plt.show()
```

References

Chung, J., & Teo, J. (2022). Mental Health Prediction Using Machine Learning: Taxonomy, Applications, and Challenges. *Applied Computational Intelligence and Soft Computing, 2022*, 9970363. https://doi.org/10.1155/2022/9970363

Craske, M. G., & Barlow, D. H. (2014). *Mastering your anxiety and worry: A six-week program for overcoming panic disorder, agoraphobia, and generalized anxiety disorder.* New York, NY: Guilford Publications.

Nemesure, M. D., Heinz, M. V., Huang, R., & Jacobson, N. C. (2021). Predictive modeling of depression and anxiety using electronic health records and a novel machine learning approach with artificial intelligence. *Scientific Reports*, *11*(1). https://doi.org/10.1038/s41598-021-81368-4

Tsai, C. H., Chen, P. C., Liu, D. S., Kuo, Y. Y., Hsieh, T. T., Chiang, D. L., Lai, F., & Wu, C. T. (2022). Panic Attack Prediction Using Wearable Devices and Machine Learning: Development and Cohort Study. *JMIR Medical Informatics*, *10*(2), e33063. https://doi.org/10.2196/33063