Samsumg Innovation Campus 2024 Ai / Universidad de Monterey UDEM, México, Abril 2025

Panic Disorder Diagnosis with Machine Learning Techniques

David Ricardo Santiago Alcocer
Angel Jovan Vargas Morales
Israel Serrano De La Torre
Abraham Alvarado Padilla

Figure 1: *Graphical Representation of a panic disorder.*

In this work, various Machine Learning techniques were employed to predict, based on multiple features, whether a person could be diagnosed with a *Panic Disorder*. The main objective is to facilitate the detection of cases that require medical and psychological treatment.

Among the most notable results is an XGBoost model in which an ***accuracy of*** **0.99** was achieved. Moreover, the number of *false negatives* was 0, which is crucial in a medical diagnosis context, as it prevents positive cases from going unnoticed.

A hierarchical clustering algorithm was also tested using the *Gower distance metric*, followed by a visual analysis using T-SNE. With this technique, a ***silhouette score of 0.27*** was achieved, indicating an adequate separation between the formed groups.

Additionally, other methods were tested, whose results were slightly inferior in terms of performance metrics. The following sections will present a detailed analysis of each of these approaches.

1 Panic Disorder

Panic disorder is a diagnosis classified under anxiety-related illnesses that has seen a notable increase in recent years. A person with this condition experiences frequent and random panic attacks, which severely affect their quality of life. A panic attack is defined as ”an abrupt surge of intense fear or discomfort” and may include physical symptoms such as palpitations, accelerated heartbeat or a pounding heart sensation, sweating, trembling or shaking, sensations of shortness of breath or choking, chest pain or discomfort, nausea or abdominal distress, dizziness, unsteadiness, light-headedness or faintness, chills or hot flashes, paresthesias (numbness or tingling sensations), derealization (feelings of unreality) or depersonalization (feeling detached from oneself), fear of losing control or ”going crazy,” and fear of dying.

These attacks may occur very frequently—several times per day—or more sporadically, a few times per year. However, they occur without warning, which can be very dangerous depending on the activity the person is engaged in at the time of the episode [1].

1.1 Why Machine Learning for this Diagnosis?

Since the main manifestation of panic disorder is panic attacks, it is a challenging diagnosis to obtain and is often misclassified. This is because this symptom frequently appears in most anxiety and depression-related conditions. Therefore, the most accurate diagnosis currently available is the one described in the Diagnostic and Statistical Manual of Mental Disorders, DSM-5. However, this requires a process of medical and analytical follow-up of the patient that can take months or even years, allowing the disorder to progress harmfully while the diagnosis remains uncertain.

In such cases, physical complications may arise, such as permanent damage to the cardiovascular system, heart attacks, heart disease, or even hypertension. In addition, there are many social and psychological consequences, including social isolation, depression, alcohol or substance abuse, and in extreme cases, the disorder may lead a person to develop suicidal thoughts.

This is why treatment for this condition is of vital importance, starting with a diagnosis that is both accurate and time-efficient, as treatment must begin as soon as possible.

In recent years, technology has revolutionized the way humans carry out daily and research activities. Different branches of artificial intelligence have helped many processes become more accurate, less error-prone, and much faster. Thus, the aim is to leverage various Machine Learning techniques to build and improve models capable of predicting and classifying this diagnosis. This would allow it to be given appropriate weight during evaluation, with the goal of speeding up the process and enabling more effective and timely treatment of the disorder [2].

2 About the Data

For the analysis and creation of the prediction models, data were taken from a repository on the ”Kaggle” website, specifically the ***Panic Disorder Detection Dataset***. Two files from this dataset were used. The first contains a total of 100,000 instances and 17 columns (16 features and 1 target). From this file, 80,000 entries were used for training and 20,000 for testing and evaluating the models. Subsequently, a second file containing 20,000 instances with the same columns was used to evaluate the model with new data. The following is a description of each feature in the dataset:

1. ***Participant ID***: Sequential identifier, irrelevant for prediction purposes.

2. ***Age***: Numerical feature indicating the age of the participants.

3. ***Gender***: Binary categorical feature indicating whether the person is male or female.

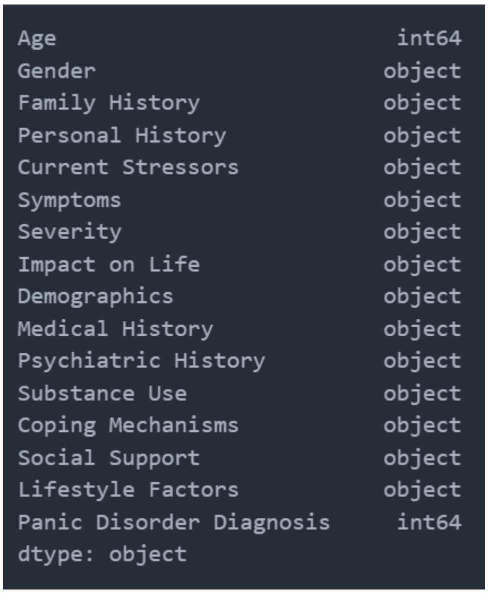
4. ***Family History***: Binary categorical feature indicating whether the patient has a family member with the same or a similar condition.

5. ***Personal History***: Binary categorical feature indicating whether the patient has had or currently has a similar condition.
6. ***Current Stressors***: Ordinal categorical feature with three levels indicating the patient's current level of stress.
7. ***Symptoms***: Categorical feature indicating the most relevant physical symptom experienced by the patient.
8. ***Severity***: Ordinal categorical feature with three levels indicating the severity level of the recorded symptom.
9. ***Impact on Life***: Ordinal categorical feature with three levels indicating the level of impact the condition has on daily life.
10. ***Demographics***: Binary categorical feature representing the type of environment in which the patient lives.
11. ***Medical History***: Categorical feature listing general medical conditions for each patient.
12. ***Psychiatric History***: Categorical feature listing any psychiatric conditions the patient has had.
13. ***Substance Use***: Categorical feature indicating the type of harmful substance used by the patient (limited to alcohol and drugs).
14. ***Coping Mechanisms***: Categorical feature listing the method each patient uses to reduce stress and stay relaxed.
15. ***Social Support***: Ordinal categorical feature with three levels indicating the level of social support (parents, friends, partner, health professionals) available to the patient.
16. ***Lifestyle Factors***: Categorical feature listing habits that negatively affect the patient's health.
17. ***Panic Disorder Diagnosis***: Target variable with binary values — ”Yes” if the patient has been diagnosed with panic disorder and ”No” otherwise.

As additional information, the dataset owner is **Muhammad Shahid**, and it is licensed under GPL 2.

3 Exploratory Data Analysis

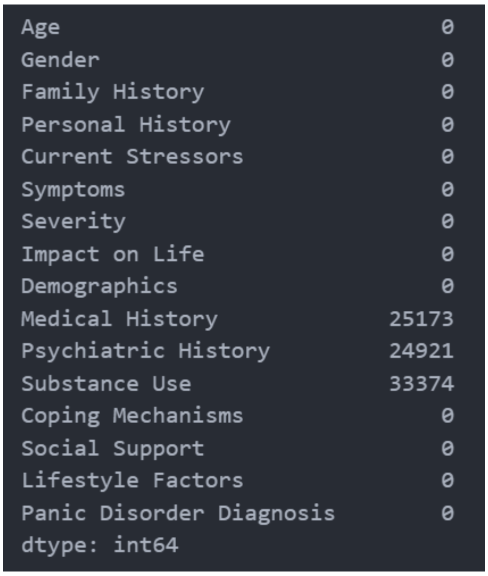
The first step is to verify that the data types are appropriate in order to avoid issues during future operations.



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

Figure 2: *Data type stored by each feature.*

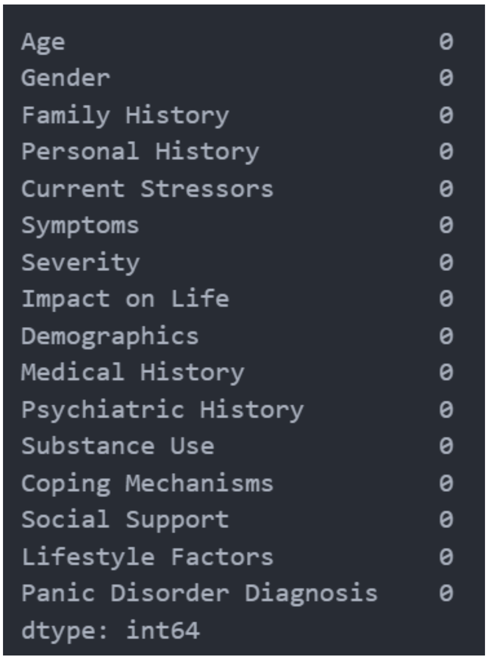
Next, we analyze whether the dataset contains any missing values.



```
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             25173
Psychiatric History         24921
Substance Use               33374
Coping Mechanisms               0
Social Support                  0
Lifestyle Factors               0
Panic Disorder Diagnosis        0
dtype: int64
```

Figure 3: *Missing values across different features.*

To address this issue, missing values were filled using the mode of each categorical feature that contained them. This approach was chosen because some of the tests and distribution plots to be performed do not allow for missing values. Once this problem was resolved, the dataset no longer contained null 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             0
Psychiatric History         0
Substance Use               0
Coping Mechanisms           0
Social Support              0
Lifestyle Factors           0
Panic Disorder Diagnosis    0
dtype: int64
```

Figure 4: *Null values issue resolved.*

Due to the nature of the dataset, most features contain categorical information. Therefore, the most useful tools for analyzing the data are histograms and category distributions. The analysis began with the distribution of the classes in the target variable:

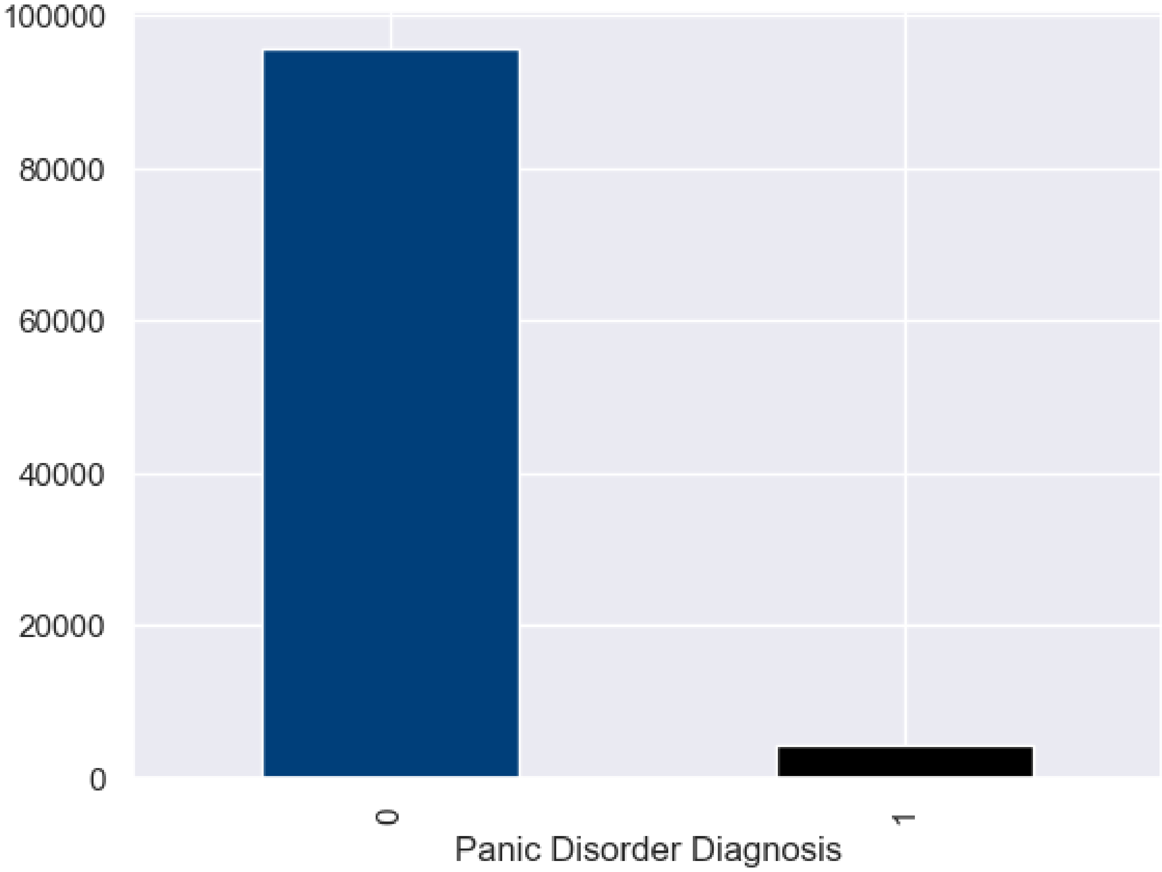
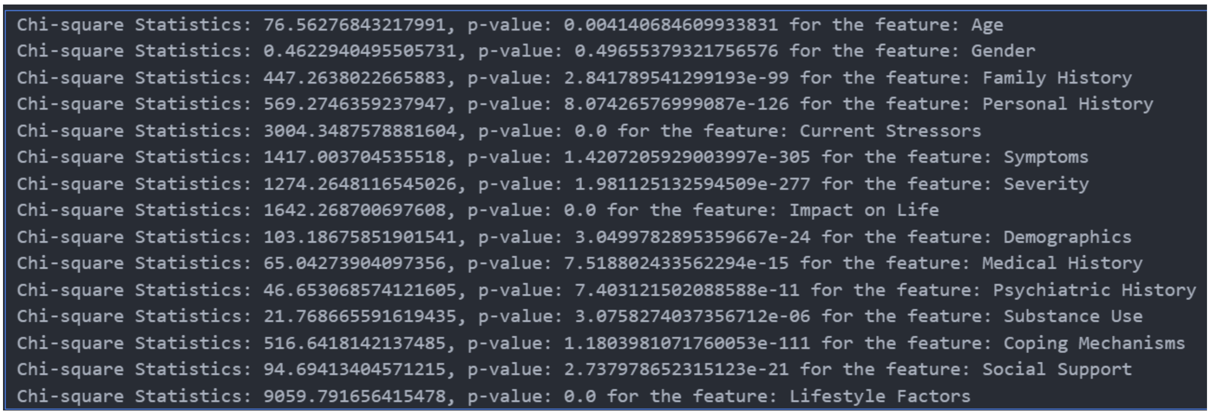


Figure 5: *Class distribution of the target variable.*

As shown, there is a significant class imbalance, which can lead to issues during model training. Most likely, the model will be biased when classifying diagnoses. This problem is addressed later using different techniques, depending on the model being applied.

Displaying a categorical distribution for each individual feature would not be ideal. Therefore, a statistical *Chi-squared* test was conducted to determine which features are most statistically related to the target. The results are as follows:



```
Chi-square Statistics: 76.56276843217991, p-value: 0.004140684609933831 for the feature: Age
Chi-square Statistics: 0.4622940495505731, p-value: 0.49655379321756576 for the feature: Gender
Chi-square Statistics: 447.2638022665883, p-value: 2.841789541299193e-99 for the feature: Family History
Chi-square Statistics: 569.2746359237947, p-value: 8.07426576999087e-126 for the feature: Personal History
Chi-square Statistics: 3004.3487578881604, p-value: 0.0 for the feature: Current Stressors
Chi-square Statistics: 1417.003704535518, p-value: 1.4207205929003997e-305 for the feature: Symptoms
Chi-square Statistics: 1274.2648116545026, p-value: 1.981125132594509e-277 for the feature: Severity
Chi-square Statistics: 1642.268700697608, p-value: 0.0 for the feature: Impact on Life
Chi-square Statistics: 103.18675851901541, p-value: 3.0499782895359667e-24 for the feature: Demographics
Chi-square Statistics: 65.04273904097356, p-value: 7.518802433562294e-15 for the feature: Medical History
Chi-square Statistics: 46.653068574121605, p-value: 7.403121502088588e-11 for the feature: Psychiatric History
Chi-square Statistics: 21.768665591619435, p-value: 3.0758274037356712e-06 for the feature: Substance Use
Chi-square Statistics: 516.6418142137485, p-value: 1.1803981071760053e-111 for the feature: Coping Mechanisms
Chi-square Statistics: 94.69413404571215, p-value: 2.737978652315123e-21 for the feature: Social Support
Chi-square Statistics: 9059.791656415478, p-value: 0.0 for the feature: Lifestyle Factors
```

Figure 6: *Chi-squared and P values from the statistical test.*

In this type of statistical test, the null hypothesis H_0 states that the variables are not statistically related. Therefore, a P-value greater than 0.05 would indicate that a feature may be discarded. This is the case for the *Gender* column, which will later receive special treatment depending on the model being used.

Once these results are obtained, the following figures show the categorical distributions of the four features that are most statistically related to the target variable.

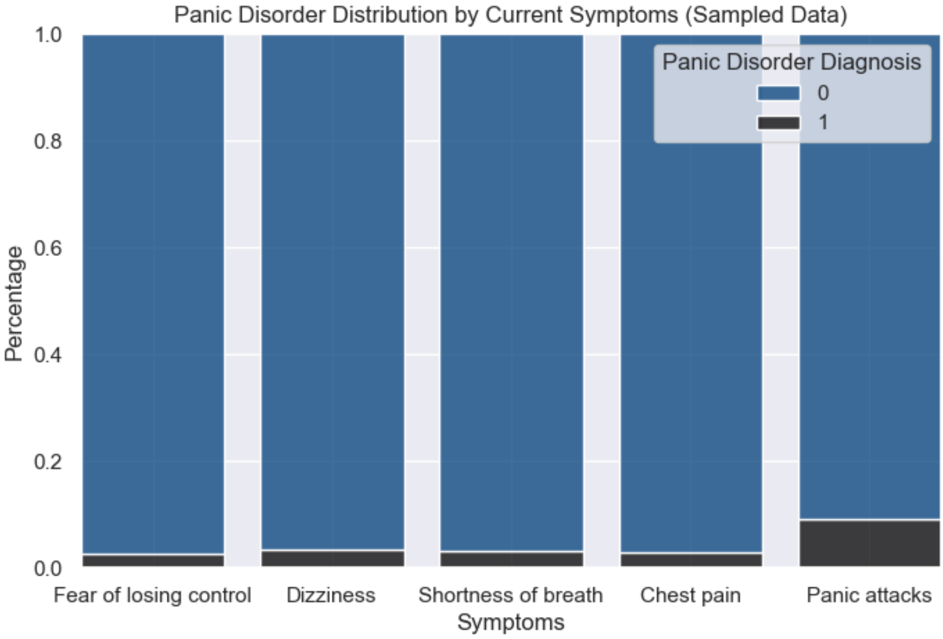


Figure 7: *Percentage of total values vs. symptom categories. The black bar represents the percentage of individuals in that category who were diagnosed with panic disorder.*

This plot shows that a higher percentage of individuals with a positive diagnosis suffer from panic attacks compared to other symptoms. This is somewhat expected, as panic attacks are the main manifestation of panic disorder. However, it is important to note that this symptom does not appear in all cases of a positive diagnosis.

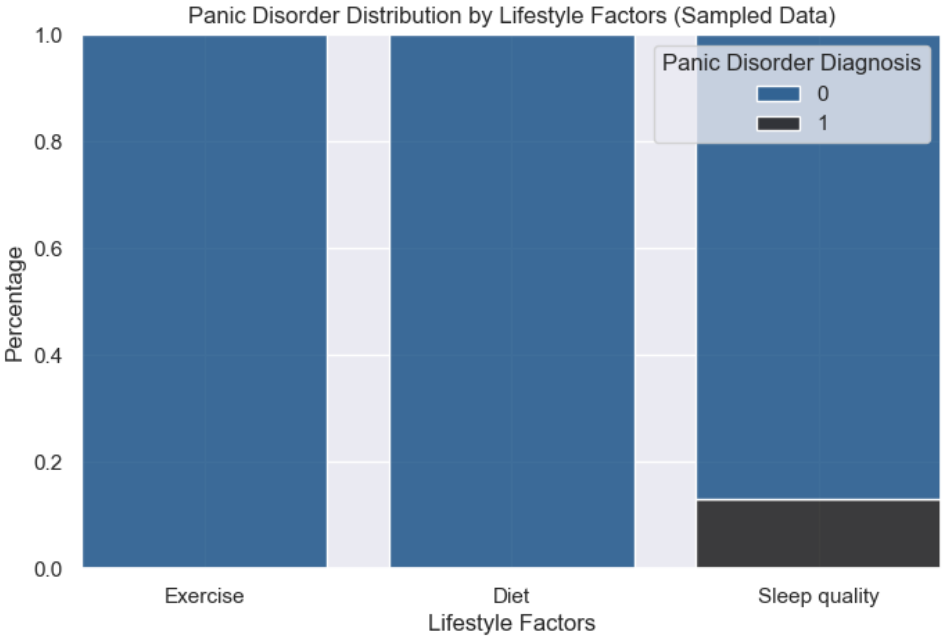


Figure 8: *Percentage of total values vs. lifestyle habit categories. The black bar represents the percentage of individuals in that category who were diagnosed with panic disorder.*

It is evident that all individuals with a positive diagnosis for panic disorder reported having poor sleep habits, either sleeping very few hours and/or having very low-quality sleep. This suggests that sleep problems may be a highly influential factor when the model makes predictions.

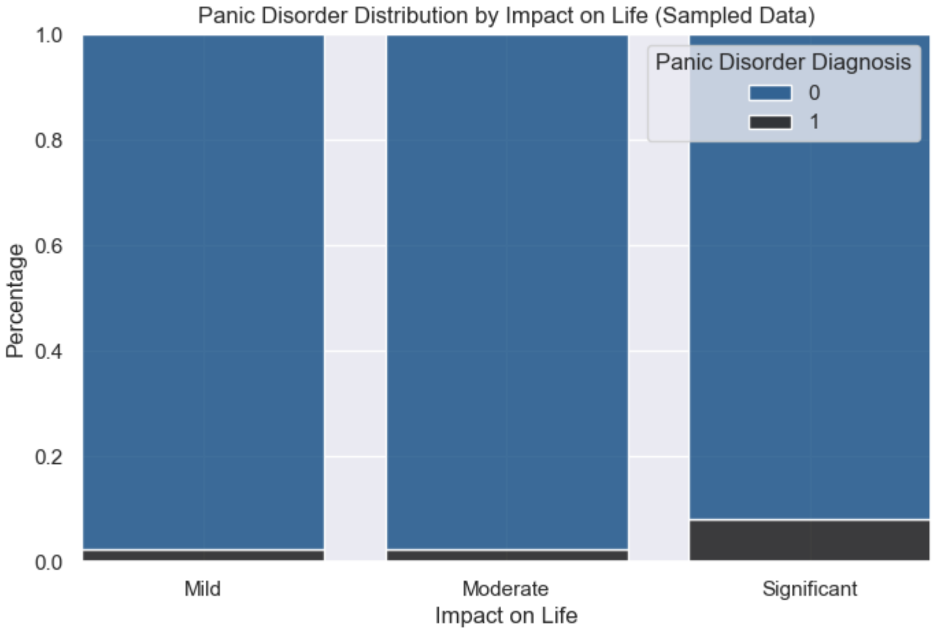


Figure 9: *Percentage of total values vs. impact on life categories. The black bar represents the percentage of individuals in that category who were diagnosed with panic disorder.*

Finally, the same class distribution is shown using the *Current Stressors* categories.

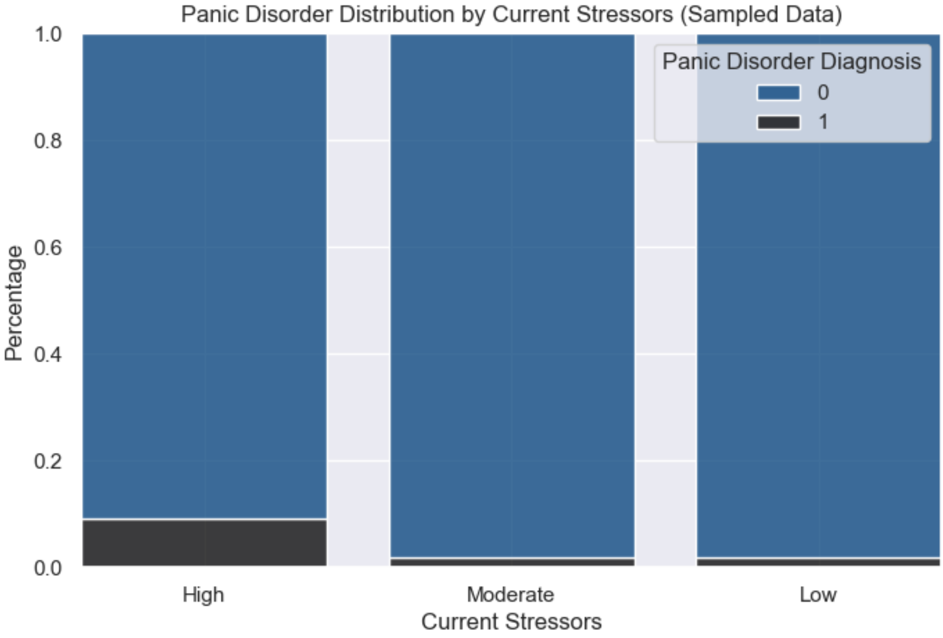


Figure 10: *Percentage of total values vs. stress level categories. The black bar represents the percentage of individuals in that category who were diagnosed with panic disorder.*

It can be observed that individuals who rate themselves as having a high level of stress tend to be those with a positive diagnosis. However, it is noteworthy that some individuals are also diagnosed despite reporting a lower level of stress. This confirms that the disorder is difficult to detect and can often be confused with other psychological conditions. Simply reducing the level of stress is not a guarantee of whether or not someone has a panic disorder.

Finally, a correlation matrix of the features is shown. For this, all features (except for the ID, which was removed) were converted into *dummy variables*, as numerical values are required to compute correlations.

There is not much to highlight from this matrix; there is almost no significant correlation between the variables, except for certain symptoms that are logically related to each other. However, it was concluded that there is no relationship strong enough to imply a significant change.

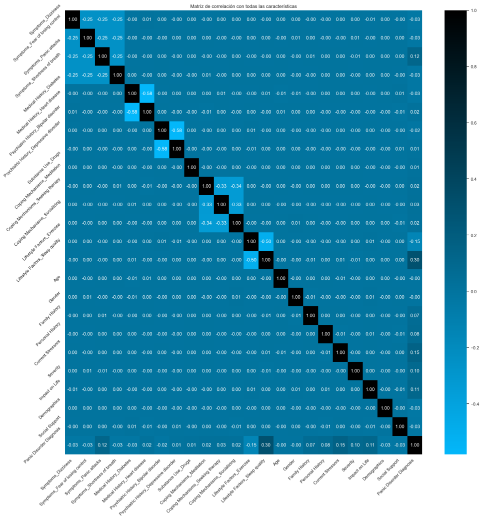


Figure 11: *Correlation between features (with dummy variables)*

4 Classification Models

The first implementation of a Machine Learning model is done using two classification models, as the main goal is to determine whether a person has a positive diagnosis for panic disorder. Thus, a classification approach is ideal.

To begin, the problem is approached with a *Random Forest* model in order to establish a baseline that can later be improved. Based on these observations, a second model using *XGBoost* is then optimized, achieving better results.

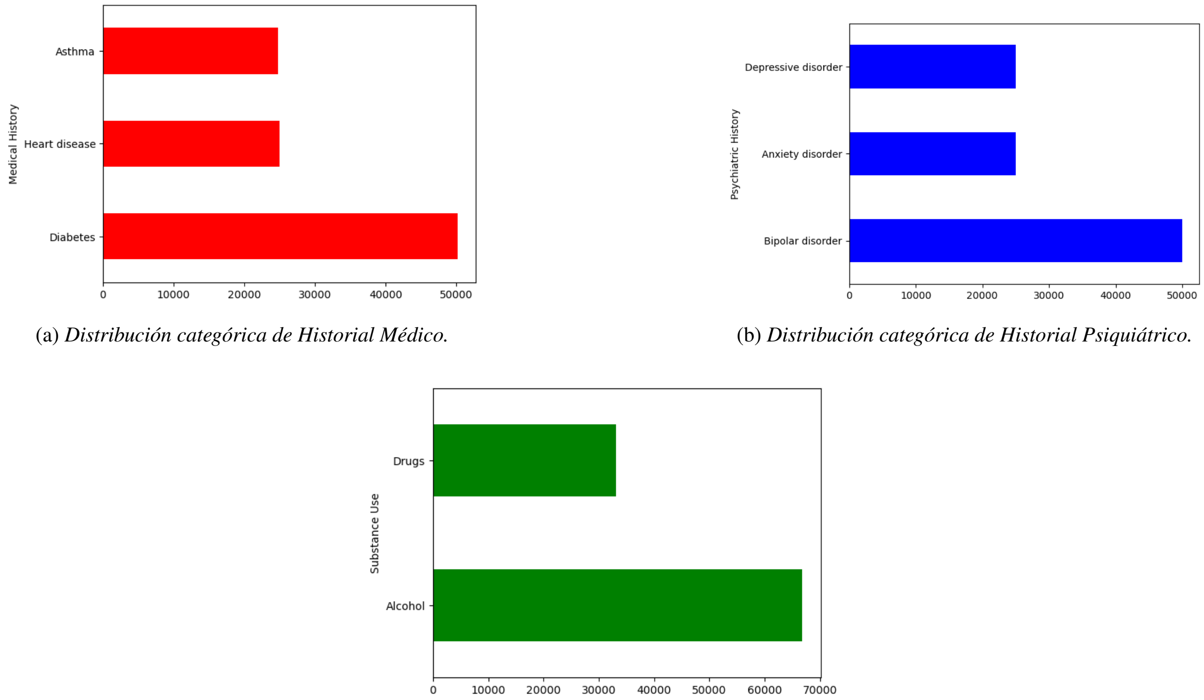
4.1 Random Forest

The first model trained for classification was a Random Forest. This choice was made because many classification models are based on decision trees. The goal was to start with a very general model that could serve as a baseline for further optimization and performance improvement. Additionally, this type of model allows for better *feature engineering* through the use of a *Gini Importance* plot, which measures the total reduction in Gini impurity and helps determine which features are more or less useful for classification.

4.1.1 Data Preprocessing

Each feature in the dataset was analyzed to determine whether any of them would be irrelevant to the model. As a result, the *Participant ID* feature was discarded, as it is merely a numerical identifier and has no predictive relevance. At this point, 15 features and 1 target variable remained.

Once this process was complete, the issue of missing values was addressed in the same way described in *Section 3*, i.e., missing values were filled using the mode of each categorical feature. However, an observation was made at this stage that is later addressed in the next model: after replacing the missing values, a potential bias may have been introduced favoring certain categories, as the previous distribution (before filling missing values) was more balanced.



(a) *Distribución categórica de Historial Médico.*

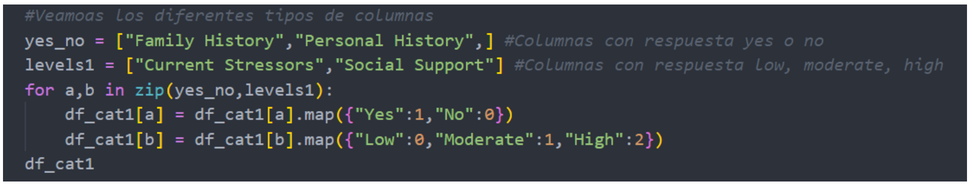
(b) *Distribución categórica de Historial Psiquiátrico.*

(c) *Distribución categórica del Uso de Sustancias.*

Figure 12: *Distribuciones categóricas de las 3 características con valores nulos.*

Next, the categorical variables are encoded, as it is important that all features have numerical values in order to apply the model.

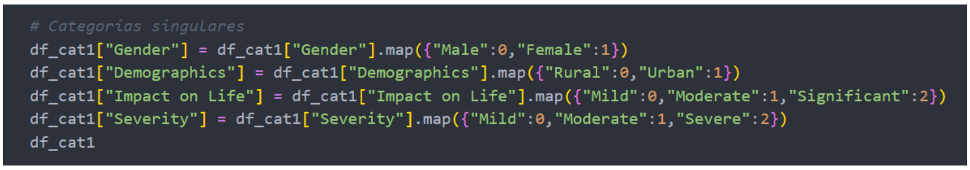
First, features with the same type of responses were identified—specifically those with: [Yes, No] and [Low, Moderate, High]:



```python
#Veamoas los diferentes tipos de columnas
yes_no = ["Family History","Personal History",] #Columnas con respuesta yes o no
levels1 = ["Current Stressors","Social Support"] #Columnas con respuesta low, moderate, high
for a,b in zip(yes_no,levels1):
    df_cat1[a] = df_cat1[a].map({"Yes":1,"No":0})
    df_cat1[b] = df_cat1[b].map({"Low":0,"Moderate":1,"High":2})
df_cat1
```

Figure 13: *yes_no: features with binary categories* "Yes" *and* "No"*, levels1: features sharing the same category names, df_cat1: name of the* DataFrame*.*

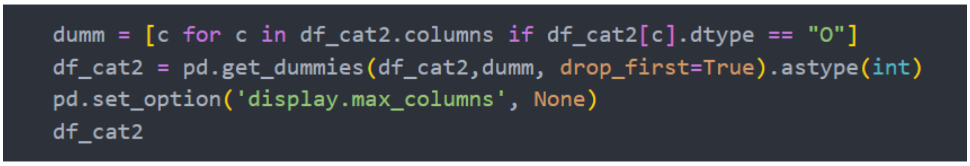
There are several features that contain their own specific categories, making it impractical to use loops for efficient encoding. Therefore, each of these variables was mapped manually.



```python
# Categorías singulares
df_cat1["Gender"] = df_cat1["Gender"].map({"Male":0,"Female":1})
df_cat1["Demographics"] = df_cat1["Demographics"].map({"Rural":0,"Urban":1})
df_cat1["Impact on Life"] = df_cat1["Impact on Life"].map({"Mild":0,"Moderate":1,"Significant":2})
df_cat1["Severity"] = df_cat1["Severity"].map({"Mild":0,"Moderate":1,"Severe":2})
df_cat1
```

Figure 14: *Mapping of different categories to numerical values for each variable.*

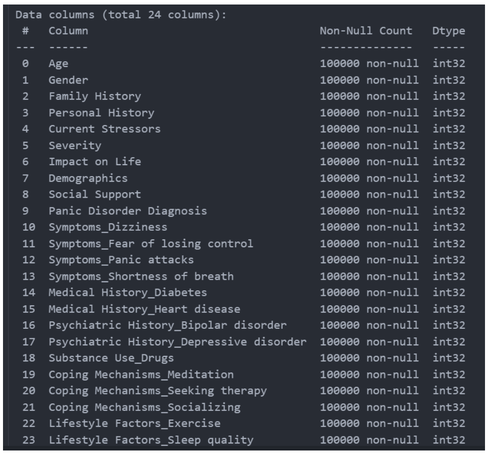
Finally, the *get_dummies* method from the pandas library was used for the remaining features that could not be encoded through the previous methods.



```python
dumm = [c for c in df_cat2.columns if df_cat2[c].dtype == "O"]
df_cat2 = pd.get_dummies(df_cat2,dumm, drop_first=True).astype(int)
pd.set_option('display.max_columns', None)
df_cat2
```

Figure 15: *Dummy encoding for the remaining features. df_cat2: name of the DataFrame used.*

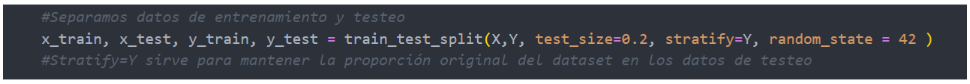
This ensures that all variables share the same data type, which must be numerical *(int32)*.



```
Data columns (total 24 columns):
 #   Column                                 Non-Null Count   Dtype
---  ------                                 --------------   -----
 0   Age                                    100000 non-null  int32
 1   Gender                                 100000 non-null  int32
 2   Family History                         100000 non-null  int32
 3   Personal History                       100000 non-null  int32
 4   Current Stressors                      100000 non-null  int32
 5   Severity                               100000 non-null  int32
 6   Impact on Life                         100000 non-null  int32
 7   Demographics                           100000 non-null  int32
 8   Social Support                         100000 non-null  int32
 9   Panic Disorder Diagnosis               100000 non-null  int32
 10  Symptoms_Dizziness                     100000 non-null  int32
 11  Symptoms_Fear of losing control        100000 non-null  int32
 12  Symptoms_Panic attacks                 100000 non-null  int32
 13  Symptoms_Shortness of breath           100000 non-null  int32
 14  Medical History_Diabetes               100000 non-null  int32
 15  Medical History_Heart disease          100000 non-null  int32
 16  Psychiatric History_Bipolar disorder   100000 non-null  int32
 17  Psychiatric History_Depressive disorder 100000 non-null  int32
 18  Substance Use_Drugs                    100000 non-null  int32
 19  Coping Mechanisms_Meditation           100000 non-null  int32
 20  Coping Mechanisms_Seeking therapy      100000 non-null  int32
 21  Coping Mechanisms_Socializing          100000 non-null  int32
 22  Lifestyle Factors_Exercise             100000 non-null  int32
 23  Lifestyle Factors_Sleep quality        100000 non-null  int32
```

Figure 16: *Data types of all feature columns.*

The issue of class imbalance in the target variable remains unresolved. To address it, the data were split into two parts: one containing all features, and another containing only the target variable. These were then divided into a training set and an evaluation set.



```
#Separamos datos de entrenamiento y testeo
x_train, x_test, y_train, y_test = train_test_split(X,Y, test_size=0.2, stratify=Y, random_state = 42 )
#Stratify=Y sirve para mantener la proporción original del dataset en los datos de testeo
```

Figure 17: *Data split for training and evaluation.*

As shown in the figure, 80% of the data (80,000 instances) was used for training, and 20% (20,000 instances) was used for evaluation. Additionally, the parameter $Stratify = Y$ was applied to preserve the class distribution in the target variable, which facilitates proper class balancing.

4.1.2 Class Balancing with SMOTE-ENN

To address the class imbalance, the ***SMOTE-ENN*** technique was employed—*Synthetic Minority Oversampling Technique (SMOTE)* combined with *Edited Nearest Neighbors (ENN)*. This method merges two approaches: oversampling and undersampling.

The first part (SMOTE) is used to generate new samples for the minority class. Rather than duplicating existing entries, it creates new data points based on a "`neighborhood`" formed by the closest neighbors of each point. This helps increase the proportion of the minority class without introducing too much noise from repeated values.

The second part (ENN) acts inversely on both classes. It eliminates noisy observations by applying a k-nearest neighbors classification; if the majority class in the neighborhood differs from the class of the observation, that observation is removed [4].

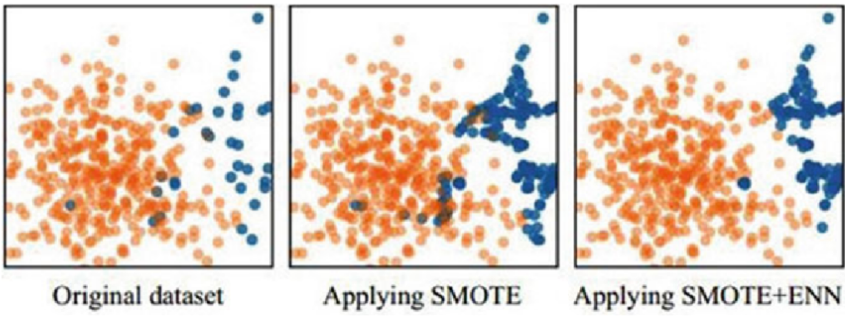
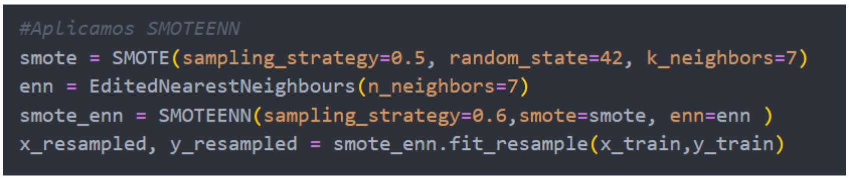


Figure 18: *Graphical representation of the SMOTE-ENN process.*

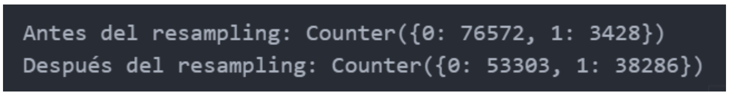
The implementation of this technique allows for the adjustment of certain parameters. In this case, the oversampling ratio of the minority class relative to the majority class was set to 0.5. This avoids generating more observations than necessary, which helps preserve the interpretability of the results.



```
#Aplicamos SMOTEENN
smote = SMOTE(sampling_strategy=0.5, random_state=42, k_neighbors=7)
enn = EditedNearestNeighbours(n_neighbors=7)
smote_enn = SMOTEENN(sampling_strategy=0.6,smote=smote, enn=enn )
x_resampled, y_resampled = smote_enn.fit_resample(x_train,y_train)
```

Figure 19: *Implementation of SMOTE-ENN in code.*

As a result, a class distribution was obtained with *0.58* for negative cases and *0.42* for positive cases, which is much more acceptable compared to the original proportion.



```
Antes del resampling: Counter({0: 76572, 1: 3428})
Después del resampling: Counter({0: 53303, 1: 38286})
```

Figure 20: *Class count before and after applying SMOTE-ENN. 0: negative diagnosis, 1: positive diagnosis.*

4.1.3 Implementation and Results

In the first model, only two parameters were modified: the number of trees (`n_estimators`), which was set to 283 (the square root of 80,000 observations), and the random state (`random_state`), which was set to 42. With this configuration, a confusion matrix was obtained with a total of 2 false negatives and 1,178 false positives.

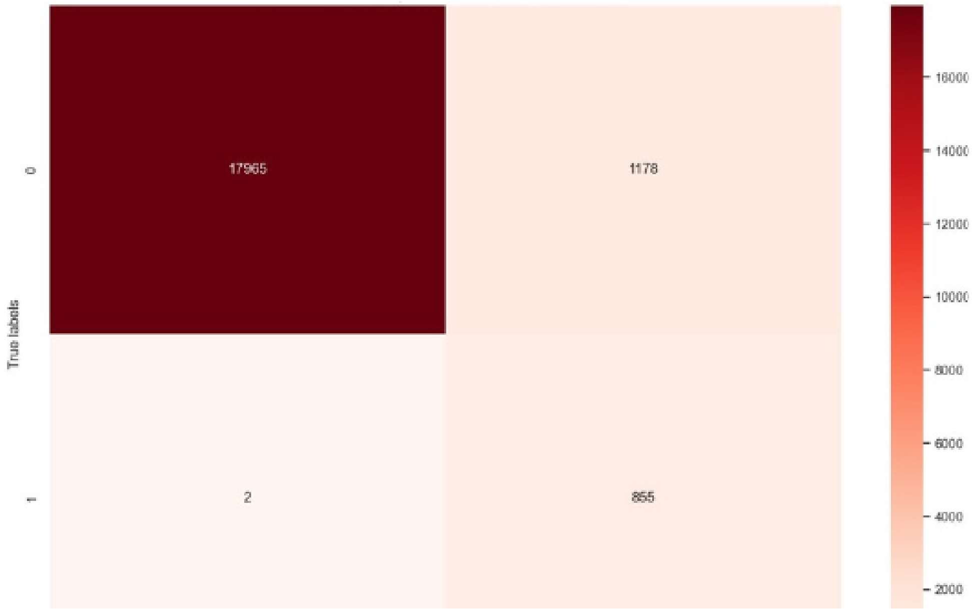
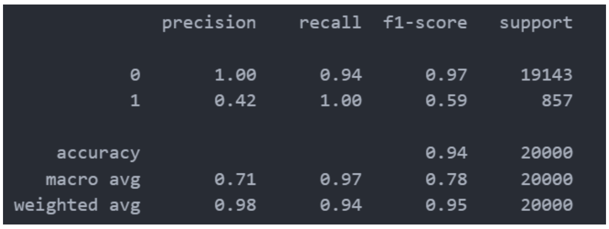


Figure 21: *Confusion matrix of the first Random Forest model.*

To better analyze the results, a classification report was generated.



```
              precision    recall  f1-score   support

           0       1.00      0.94      0.97     19143
           1       0.42      1.00      0.59       857

    accuracy                           0.94     20000
   macro avg       0.71      0.97      0.78     20000
weighted avg       0.98      0.94      0.95     20000
```

Figure 22: *Classification report for the first Random Forest model.*

Although the model achieved a high accuracy (0.94), there is still much room for improvement, especially in terms of *precision*, where one of the classes scored very low. This is due to the high number of false positives.

One of the advantages of this model is the ability to determine which features are most important in the classification of observations. This is particularly useful for informing future *feature engineering* decisions.

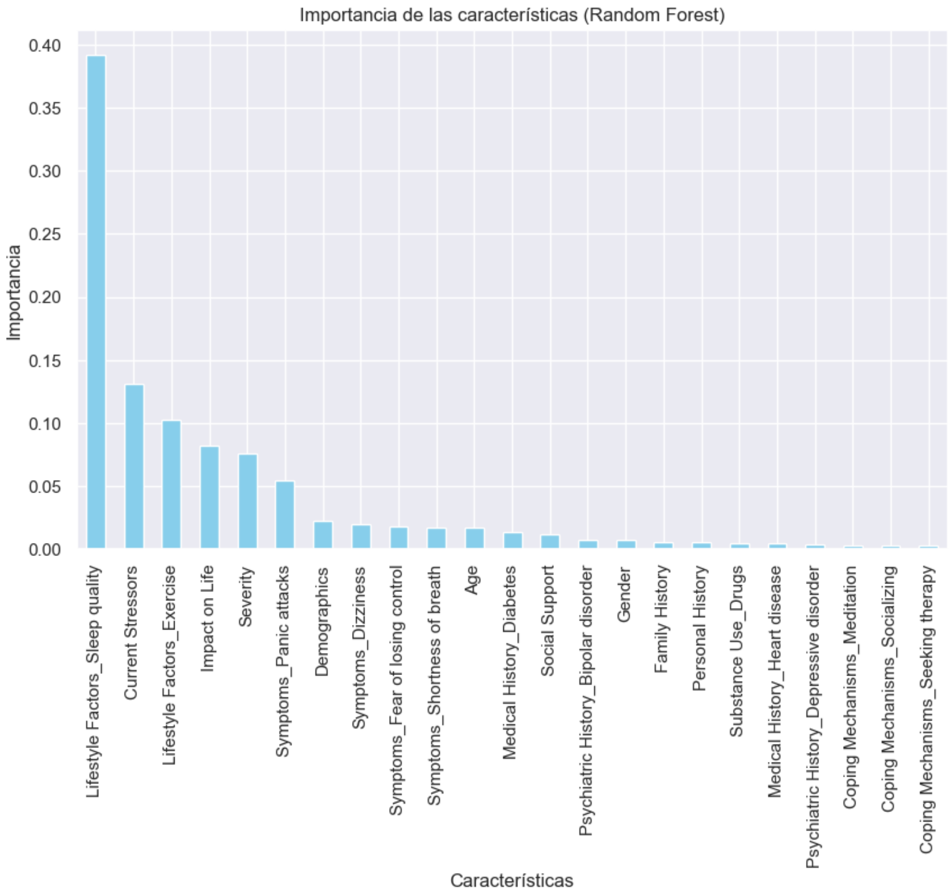
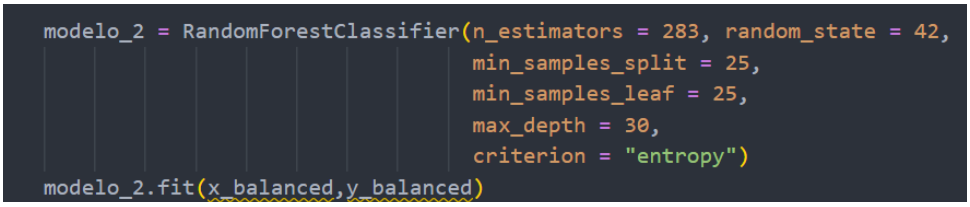


Figure 23: *Feature importance for classification, calculated using Gini importance and sorted from most (left) to least (right) relevant.*

Based on this analysis, several features were discarded in an effort to improve or simplify the model. From the *Gender* variable onward, these columns were removed, and a second Random Forest model was tested with more hyperparameters adjusted—particularly those influencing the behavior of the individual trees.



```
modelo_2 = RandomForestClassifier(n_estimators = 283, random_state = 42,
                                  min_samples_split = 25,
                                  min_samples_leaf = 25,
                                  max_depth = 30,
                                  criterion = "entropy")
modelo_2.fit(x_balanced,y_balanced)
```

Figure 24: *Creation and training of the second Random Forest model, showing the modified hyperparameters.*

It is also shown that in this case the `entropy` criterion was used, as it yielded better results, which are presented in the following figure.

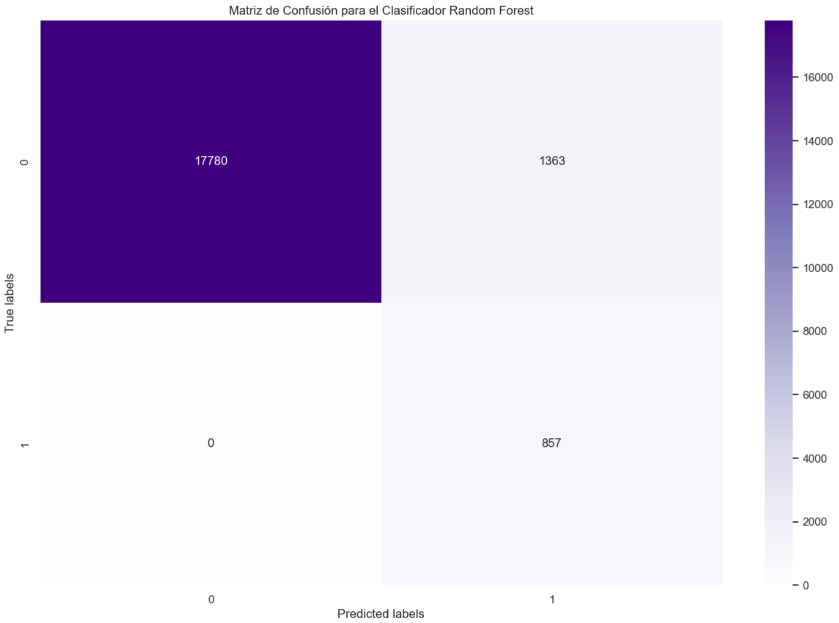
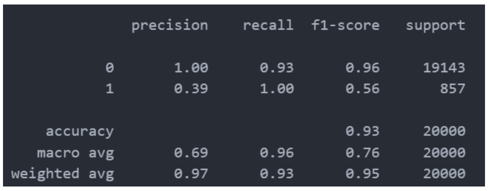


Figure 25: *Confusion matrix of the second Random Forest model.*

An improvement achieved is that the number of **false negatives was reduced to 0**, which is of utmost importance in a medical diagnostic model. However, it is evident that the number of false positives increased. A classification report was also generated to better compare the results.



```
              precision    recall  f1-score   support

           0       1.00      0.93      0.96     19143
           1       0.39      1.00      0.56       857

    accuracy                           0.93     20000
   macro avg       0.69      0.96      0.76     20000
weighted avg       0.97      0.93      0.95     20000
```

Figure 26: *Classification report of the second Random Forest model.*

In this case, the results were worse in terms of *precision* and *accuracy*, as the classification errors increased.

To better interpret the performance, a *Precision-Recall curve* was plotted, which is more suitable for imbalanced datasets. Additionally, the *AUC* metric was computed, with a value of 0.64, indicating that the model performs better than random guessing, although this value is somewhat low and could be improved.

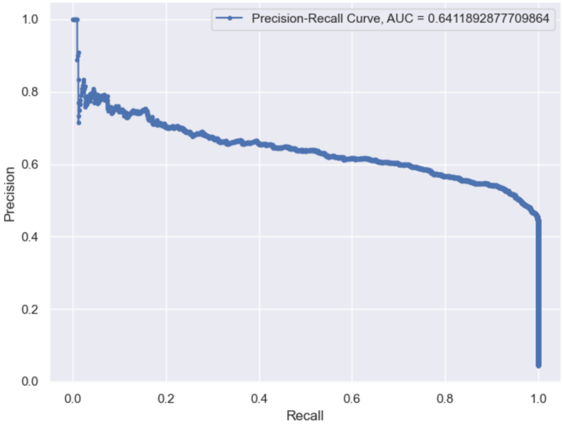


Figure 27: *Precision-Recall curve with an AUC value of 0.64.*

As shown, as *Recall* increases, *Precision* decreases. This is because the model becomes less strict when identifying positive cases, increasing the number of true positives but at the cost of more false positives.

4.1.4 Discussion

The first model implemented to solve the proposed problem performed adequately. It served as a baseline and produced standard results that provided useful *insights*. For example, based on Figure 23, we now know which features are likely to be most important in future models. Poor sleep quality, high and persistent stress levels, and little or no physical activity are key determinants that may increase the probability of a positive panic disorder diagnosis.

However, there is still much room for improvement. The current metrics classify the model as a standard baseline. In addition, synthetic sample generation can affect classification accuracy—while these samples are not generated randomly, the disorder remains rare, making proper classification inherently difficult. Once the complexity of the data and the areas for improvement were understood, more complex models were chosen for further development.

4.2 XGBoost

XGBoost is a tree-based model that incorporates greater complexity by optimizing a loss function calculated from the errors of previous trees (*tree boosting*). It includes several hyperparameters that allow the model to better adapt to the data.

First of all, XGBoost uses adaptive `neighborhoods`, allowing it to apply different degrees of flexibility in different regions of the input space. This results in an automatic *feature selection* process, eliminating errors that may arise from manually excluding columns. Additionally, the model applies a penalty to each tree it generates. Consequently, the trees can vary significantly in the number of nodes, meaning each tree can be very different from the others. This enables the model to assign more importance to certain classifications depending on the types of trees used.

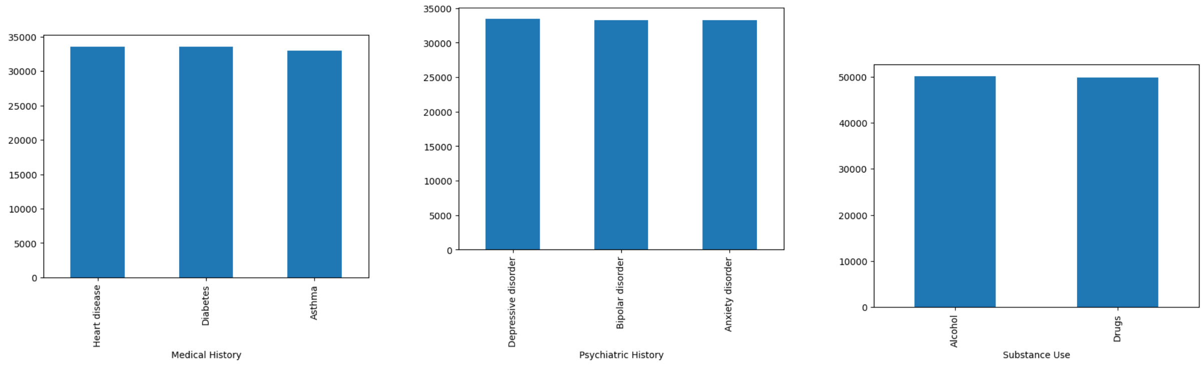
In this way, the class imbalance problem in the *target* variable can be addressed without the need for oversampling or undersampling techniques[5].

4.2.1 Data Preprocessing

For this model, a highly complex or specific preprocessing is not necessary, as the model itself can adapt or handle various data conditions on its own. Nevertheless, certain actions were taken that are likely to lead to better results.

As in the previous model, the *Participant ID* feature was removed, since it again has no predictive importance. A change from the Random Forest model was then introduced: the missing categorical values were filled with random values from all possible categories. This was done because the original distribution of these categories was fairly balanced, and this approach avoids giving the model a bias toward a category with a disproportionately high mode.

The following image displays the distribution of the categories in the columns that originally contained null values, after applying the actions mentioned above.



(a) *Categorical distribution of Medical History.*

(b) *Categorical distribution of Psychiatric History.*

(c) *Categorical distribution of Substance Use.*

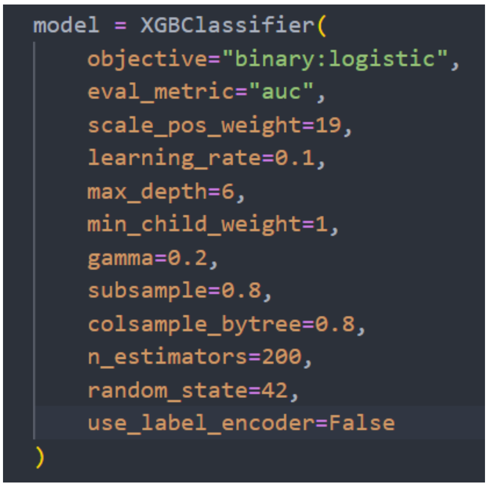
Figure 28: *Categorical distributions of the three features that had missing values, after filling.*

Regarding data types, it is essential that all variables be numeric in order for the model to work properly. Therefore, it was decided that the best approach would be to apply *one-hot encoding (dummies)* to all categorical features. Although this results in a total of 41 columns plus 1 target, the dataset still contains 80,000 training samples, so this dimensionality is not expected to have a negative impact.

Finally, the data were split in the same manner as with the previous model (Random Forest): all predictor variables were stored in `X` and the target in `Y`. The dataset was then divided into 80% for training and 20% for evaluation using the same parameters as before. For this reason, no visualization is shown, as it would be identical to *Figure 17*.

4.2.2 Implementation and Results

Once the data were ready, the model was implemented.



```
model = XGBClassifier(
    objective="binary:logistic",
    eval_metric="auc",
    scale_pos_weight=19,
    learning_rate=0.1,
    max_depth=6,
    min_child_weight=1,
    gamma=0.2,
    subsample=0.8,
    colsample_bytree=0.8,
    n_estimators=200,
    random_state=42,
    use_label_encoder=False
)
```

Figure 29: *Implementation of the XGBoost model with some explicitly specified parameters.*

Some hyperparameters were specified to help the model better adapt to the data. First, the model was instructed to perform binary classification (i.e., whether or not an individual has panic disorder), since this type of algorithm (boosting trees) can also be used for regression tasks, which is not the case in this project. Then, the evaluation metric `auc` was used at each iteration to ensure the predictions were not random.

One of the most important parameters is `scale_pos_weight`, which indicates that the target variable is imbalanced. This value was calculated as follows:

$$\text{ratio} = \frac{\text{\# of majority class samples}}{\text{\# of minority class samples}} = \frac{95000}{5000} = 19. \tag{1}$$

This automatically addresses the class imbalance problem, as the trees that classify the minority class (1) will be given much more weight than those classifying the majority class (0).

From this point on, the rest of the parameters are related to the internal structure of the trees generated by the model. For example, `gamma` is a regularization parameter that controls the minimum loss reduction required to make a split at a tree node.

With these configurations, the results obtained were as follows:

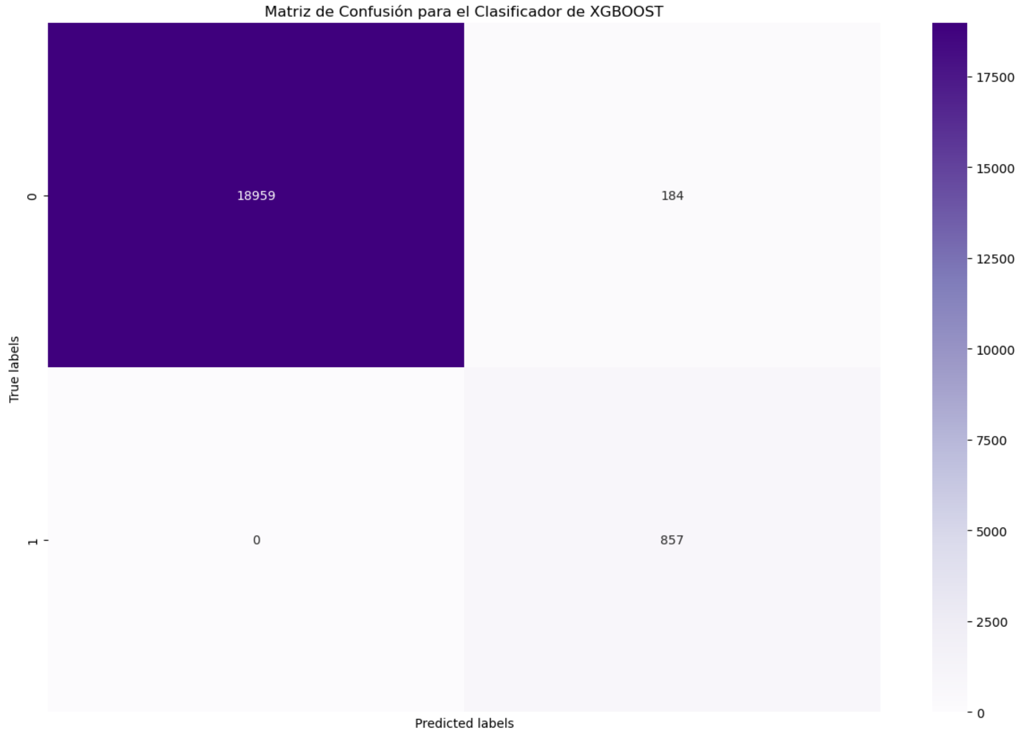
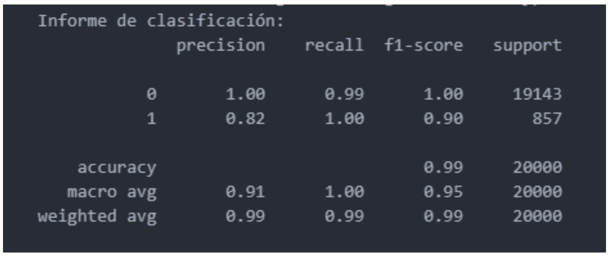


Figure 30: *Confusion matrix of the XGBoost model with the 20,000 evaluation samples.*

As shown, the results are significantly better than the previous model. Although **both models achieved 0 false negatives**, this model **reduced the number of false positives by approximately 86%**. To better evaluate the model, the classification report is shown below.



```
Informe de clasificación:
              precision    recall  f1-score   support

           0       1.00      0.99      1.00     19143
           1       0.82      1.00      0.90       857

    accuracy                           0.99     20000
   macro avg       0.91      1.00      0.95     20000
weighted avg       0.99      0.99      0.99     20000
```

Figure 31: *Classification report corresponding to the confusion matrix in Figure 30.*

An ***accuracy* of 0.99** was achieved, and for Class 1, the model reached a *precision* of 0.82 — not perfect, but highly acceptable.

To further validate the model, the second file from the *dataset* referenced in Section 2 (About the Data) was used. This file contains 20,000 additional instances that the model had not seen before. These were preprocessed in the same way as the training data and then evaluated with the model. The results are shown below.

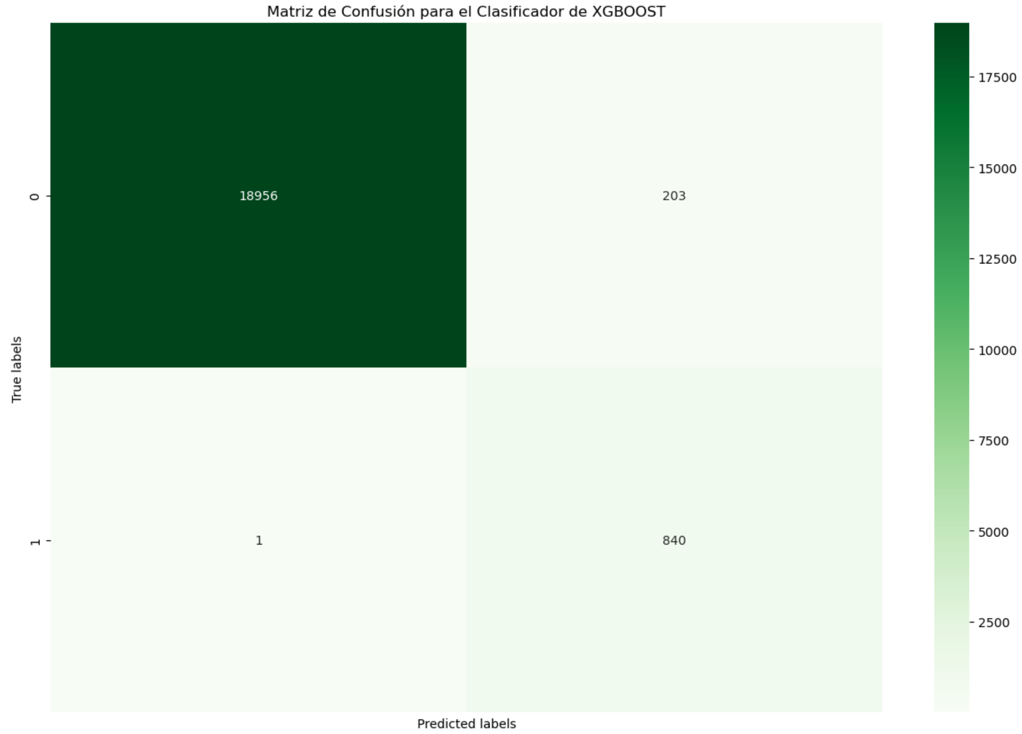
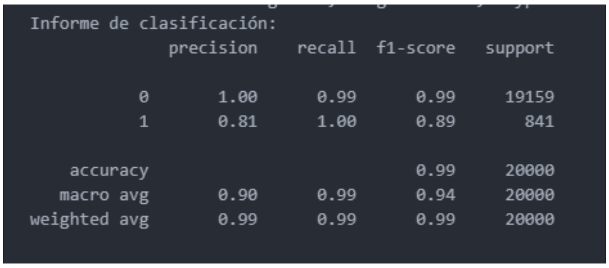


Figure 32: *Confusion matrix for the 20,000 additional samples from the second dataset file.*

Additionally, the classification report for these results is presented in the following figure.



```
Informe de clasificación:
              precision    recall  f1-score   support

           0       1.00      0.99      0.99     19159
           1       0.81      1.00      0.89       841

    accuracy                           0.99     20000
   macro avg       0.90      0.99      0.94     20000
weighted avg       0.99      0.99      0.99     20000
```

Figure 33: *Classification report for the confusion matrix of the model applied to the 20,000 test samples from the dataset.*

In this case, the model produced 1 false negative and 203 false positives, resulting in a 1% reduction in *precision*. This may be due to the presence of various different profiles among individuals with a positive diagnosis, which can lead even a more complex and adaptive model to make classification errors. Moreover, despite the 20,000 new samples, the same class imbalance in the target variable is observed, confirming once again that this condition (panic disorder) is uncommon.

4.2.3 Discussion

The improvement in this model is significant. However, there are still 184 instances that were incorrectly classified. To investigate the reason behind this, the 184 samples were stored in a separate *dataset*, and the categorical distributions of all their features were visualized and compared to the distributions of the true negative cases in order to detect any particular patterns that might explain the misclassification.

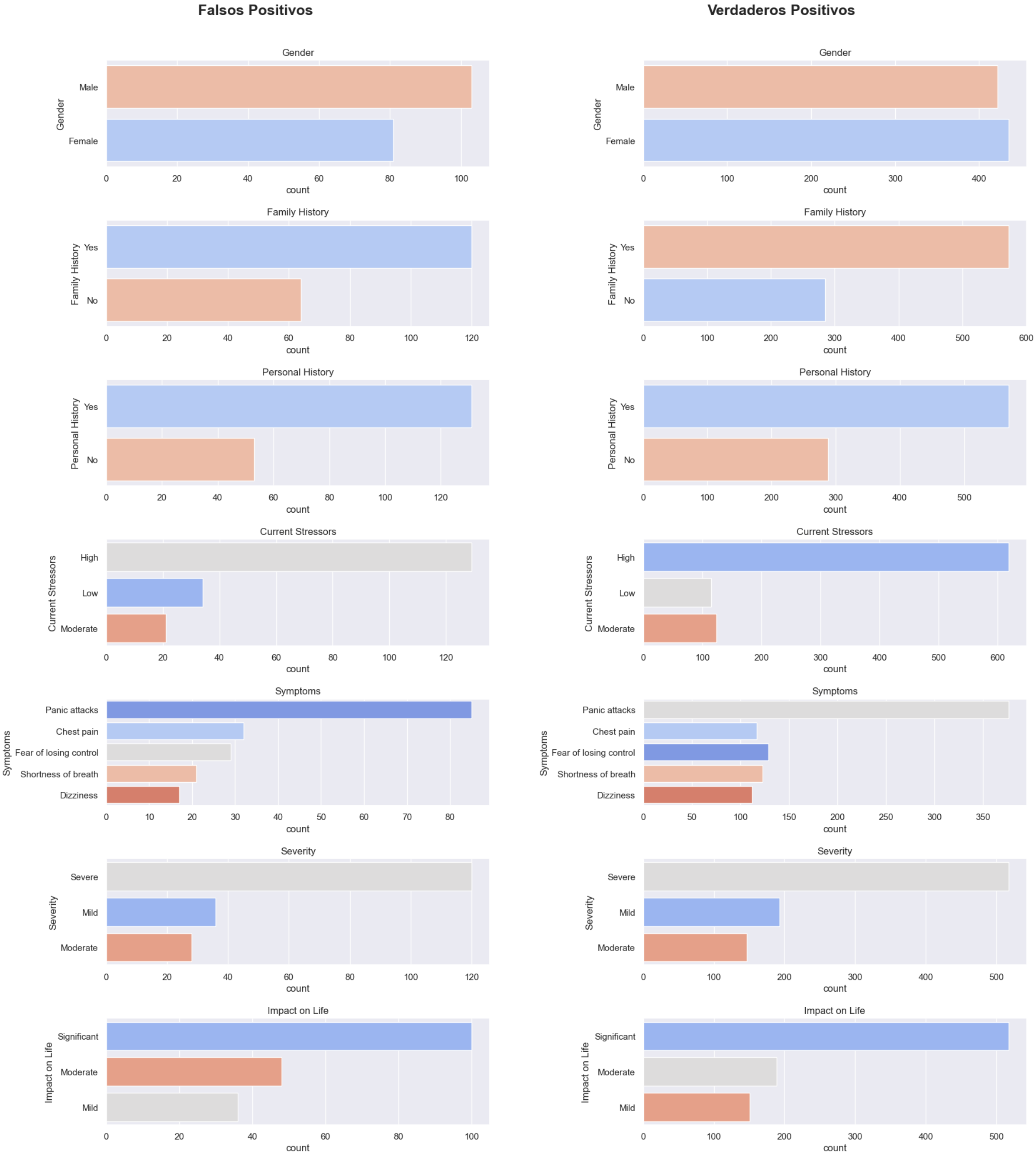


Figure 34: *First part of the categorical feature distributions for false positives (left) and true negatives (right).*

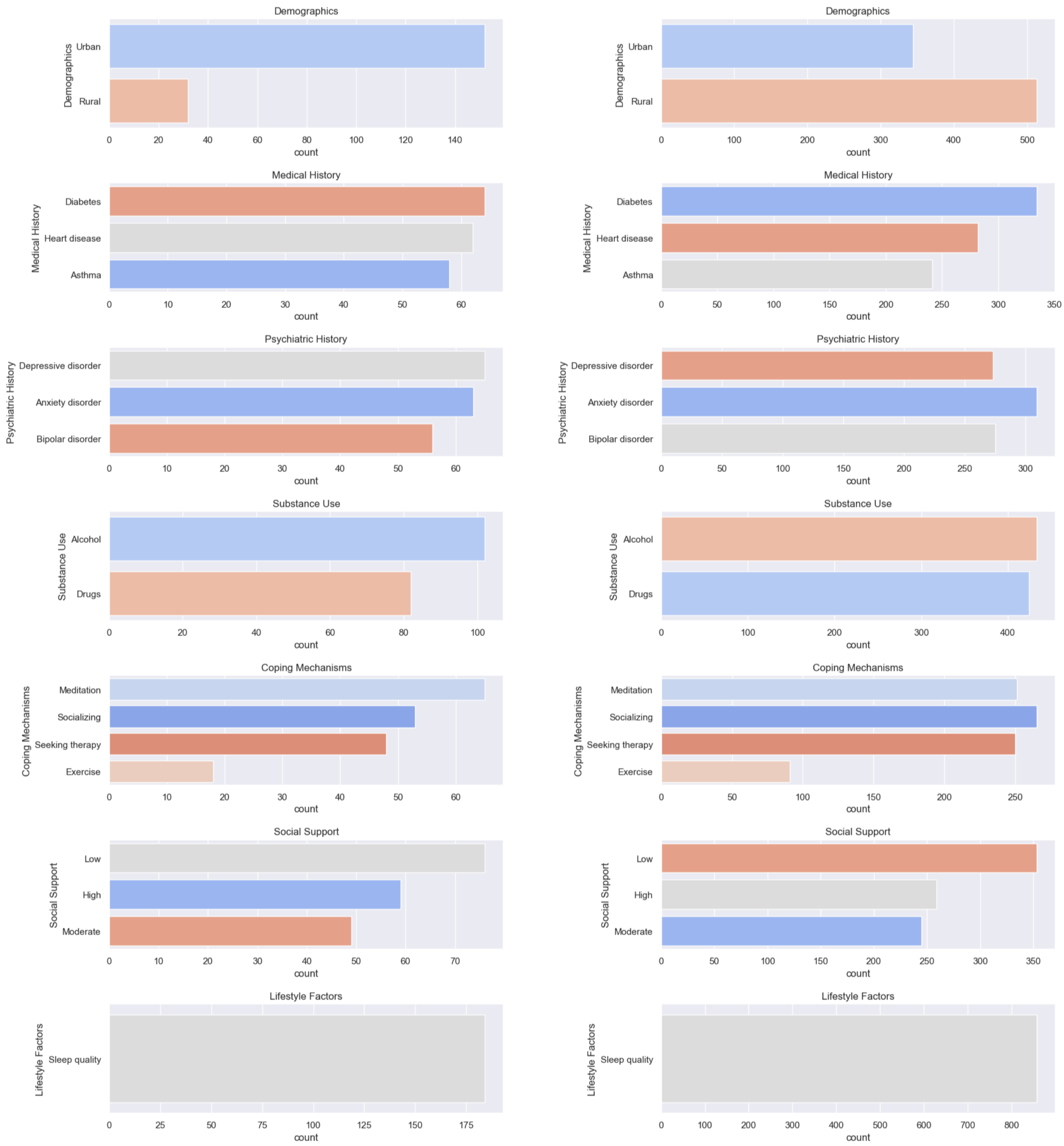


Figure 35: *Second part of the categorical feature distributions for false positives (left) and true negatives (right).*

Unfortunately, no feature of significant importance was found that clearly differentiates the classes. Some details may suggest something relevant, for instance, individuals with panic disorder tend to have higher stress levels—typically between moderate and high. They also report more frequently feeling that they are *losing control*.

Moreover, in this group (compared to the false positives), a psychiatric history of *anxiety disorder* is more prevalent than *depression disorder*, likely because panic attacks occur more frequently, which increases anxiety. Another notable difference is that individuals with a positive diagnosis are more likely to cope using social mechanisms (such as help from family and friends), whereas in the false positive group, alternative methods like meditation are more commonly used. This possibly indicates that seeking alternative ways to relax and disconnect may be more beneficial.

What stands out the most is that a greater number of individuals diagnosed with panic disorder live in rural areas compared to those who do not (the latter are mostly from urban areas). This is likely due to the fact that in larger and more urbanized cities, there are more and better ways to manage mental disorders—such as access to psychiatric hospitals, mental health professionals, pharmacies with medications, as well as a variety of options for distraction and wellness like parks, exercise facilities, and social groups.

That said, and returning to the point raised at the beginning of the section, no clear anomaly was detected that would explain why the model misclassified these cases. It is also possible that some of the false positives are due to misdiagnosis, since traditional medical assessments are not immune to errors—and this has always happened and will continue to happen.

5 Logistic Regression

Due to the structure of the data, the only plausible solutions are those provided by classification models, as the objective is simply to determine whether a person does or does not have a positive panic disorder diagnosis. For this reason, two tree-based classification models were previously chosen: *Random Forest* and *XGBoost*. However, these are not the only models capable of predicting binary classes.

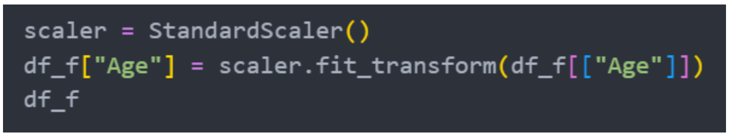
Thus, a logistic regression model was implemented. This technique can classify data without the use of decision trees or gradient boosting. Logistic regression works similarly to linear regression, with the difference being that it predicts a quantity known as the *log(odds)*, which is later mapped to a different space (probabilities), and using a threshold, it determines whether a data point belongs to one class or the other.

5.1 Data Processing

The data processing for this model was fairly straightforward and similar to the previous model. As in all cases, the *Participant ID* column was removed because it is not relevant. Missing values were then filled with random samples from their respective categories. One-hot encoding (*dummies*) was performed on all categorical features (excluding age, for obvious reasons), resulting in a total of 41 columns (not including the target) and 100,000 instances.

No visualizations were included in this section since the steps are identical to those shown in previous sections.

The only new and different step in this model was the standardization of the *Age* feature. Since this is a numerical variable, it is important to scale it appropriately because regression models are very sensitive to numerical scales. For this purpose, the *StandardScaler* class from the *sklearn* library was used to bring the values into a range that does not vary as drastically and would not negatively affect the model's training.



```
scaler = StandardScaler()
df_f["Age"] = scaler.fit_transform(df_f[["Age"]])
df_f
```

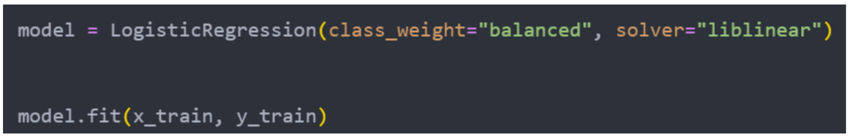
Figure 36: *Implementation of the* `StandardScaler` *class to standardize the numerical* `Age` *feature.*

Finally, the data were split into training and test sets, with 80% allocated for training and 20% for testing.

5.2 Model and Results

In this case, a fairly simple logistic regression model was used, with only two parameters modified. The first was `class_weight`, which uses the target class distribution to adjust the weights inversely proportional to the frequency of each class. This helps address the class imbalance issue in the target variable.

The second parameter was `solver`, which determines the optimization method used for minimizing the cost function.



```
model = LogisticRegression(class_weight="balanced", solver="liblinear")

model.fit(x_train, y_train)
```

Figure 37: *Logistic Regression model implemented using SKlearn.*

Using this model, the following results were obtained:

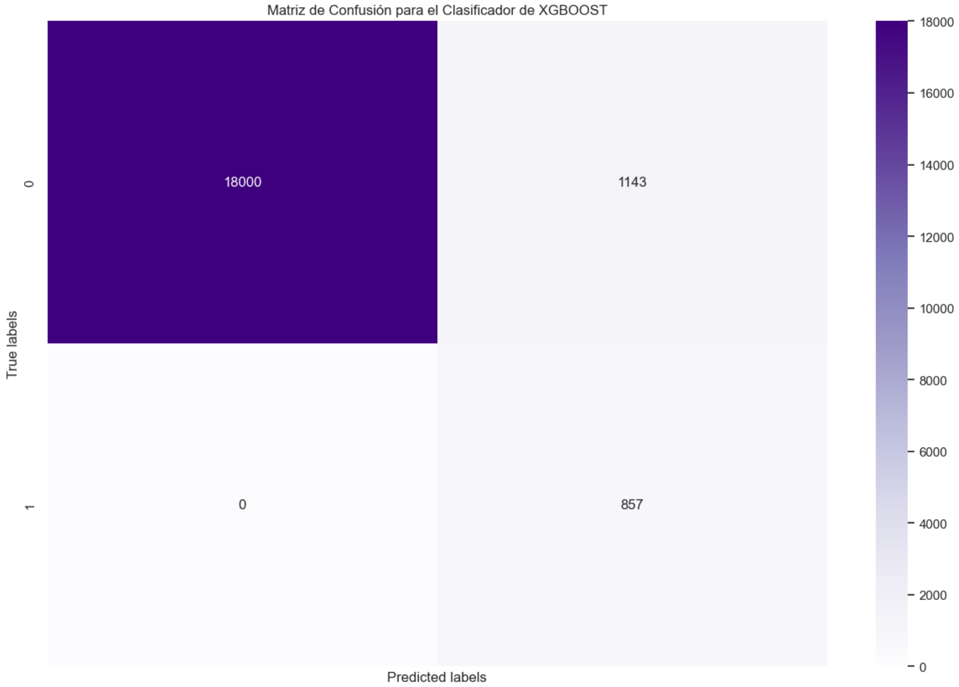
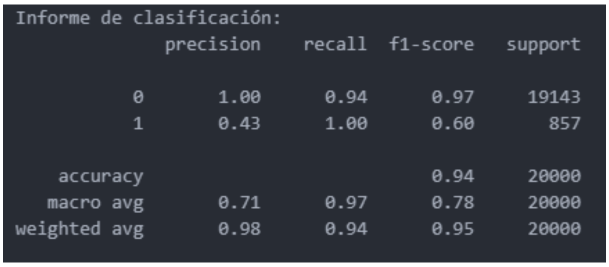


Figure 38: *Confusion matrix of the logistic regression model.*

The number of false positives is similar to that obtained with the *Random Forest* model. For further insight, the classification report corresponding to the previous matrix is presented below.



```
Informe de clasificación:
              precision    recall  f1-score   support

           0       1.00      0.94      0.97     19143
           1       0.43      1.00      0.60       857

    accuracy                           0.94     20000
   macro avg       0.71      0.97      0.78     20000
weighted avg       0.98      0.94      0.95     20000
```

Figure 39: *Classification report of the confusion matrix for the first logistic regression model.*

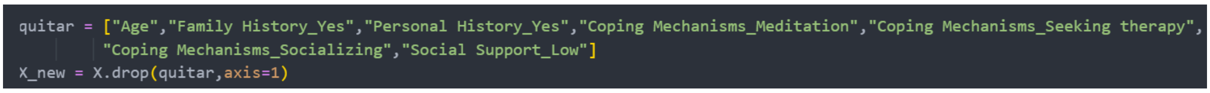
It is more clearly evident that the model makes many errors when classifying individuals who do not have a positive diagnosis but are mistakenly labeled as such.

To better interpret the results and identify opportunities for model improvement, the weights (coefficients) of each feature were compiled into a table.

Table 1: *Table of coefficients generated by the logistic regression model. The* ***intercept*** *feature is an intrinsic model value and does not carry interpretative significance.*

Feature	Logistic Coefficient
Lifestyle Factors_Sleep quality	9.072931
Symptoms_Panic attacks	3.636693
Current Stressors_High	2.989532
Impact on Life_Significant	2.010862
Severity_Severe	1.925982
Coping Mechanisms_Seeking therapy	0.042235
Social Support_Low	0.03392
Coping Mechanisms_Socializing	-0.00591
Coping Mechanisms_Meditation	-0.00836
Age	-0.02779
Personal History_Yes	-0.03777
Family History_Yes	-0.06834
Psychiatric History_Anxiety disorder	-0.58242
Medical History_Diabetes	-0.61185
Medical History_Asthma	-0.62282
Psychiatric History_Bipolar disorder	-0.64032
Medical History_Heart disease	-0.65627
Psychiatric History_Depressive disorder	-0.66819
Demographics_Rural	-0.75594
Substance Use_Alcohol	-0.9031
Social Support_Moderate	-0.90899
Gender_Male	-0.93385
Gender_Female	-0.95709
Substance Use_Drugs	-0.98783
Social Support_High	-1.01587
Demographics_Urban	-1.13499
Symptoms_Shortness of breath	-1.31073
Symptoms_Fear of losing control	-1.36972
Symptoms_Dizziness	-1.40574
Symptoms_Chest pain	-1.44143
Family History_No	-1.82259
Severity_Moderate	-1.84266
Personal History_No	-1.85316
Intercept	-1.89093
Coping Mechanisms_Exercise	-1.91889
Impact on Life_Mild	-1.94717
Impact on Life_Moderate	-1.95462
Severity_Mild	-1.97426
Current Stressors_Low	-2.36131
Current Stressors_Moderate	-2.51915
Lifestyle Factors_Diet	-5.41441
Lifestyle Factors_Exercise	-5.54946

With this information, *feature engineering* was performed, as some features had coefficients very close to zero—indicating that they have little or no relevance to the target prediction. For this reason, the following variables were removed:



```
quitar = ["Age","Family History_Yes","Personal History_Yes","Coping Mechanisms_Meditation","Coping Mechanisms_Seeking therapy",
          "Coping Mechanisms_Socializing","Social Support_Low"]
X_new = X.drop(quitar,axis=1)
```

Figure 40: *Python code showing the selected features to be removed and the creation of a new dataframe excluding those variables.*

Subsequently, the *threshold* used for predictions on the evaluation data was adjusted. Since this is a rare disorder, it is logical to assume that most people will not have it. Therefore, the *threshold* was increased to 0.6. The results are shown below:

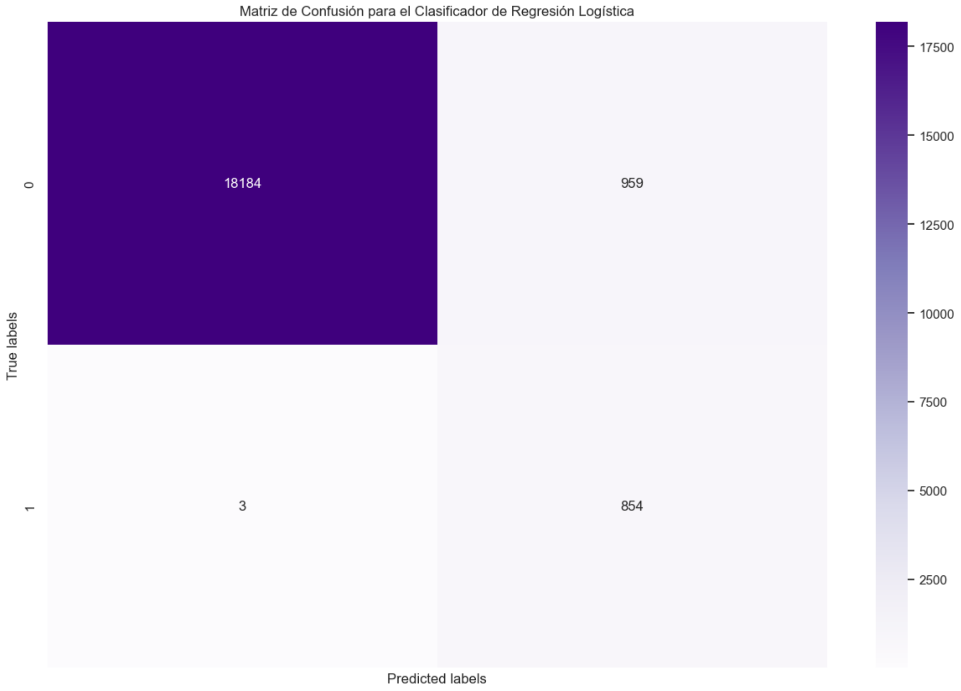


Figure 41: *Confusion matrix after applying feature engineering and increasing the prediction threshold to 0.6.*

With these adjustments, **false positives were reduced by approximately 16%**; however, the number of false negatives increased to 3, which is undesirable in a medical diagnostic model. Therefore, this modification was not considered a substantial improvement.

5.3 Discussion

Overall, the performance of the logistic regression model was fairly average. That is, it did not achieve high nor particularly low metrics—it actually performed worse than the previous model (XGBoost). Nonetheless, the most beneficial contribution from this section was the information provided in *Table 1*, which contains the logistic coefficient for each feature—i.e., the importance of each variable in determining how the model makes its classifications.

If the logistic coefficient has a positive value, it means that the feature contributes to a negative classification (i.e., the patient does not have panic disorder). On the other hand, if the coefficient is negative, the feature is important for a positive classification (i.e., the patient does have panic disorder). Furthermore, the absolute value of the coefficient is also meaningful—the larger it is, the more relevant the feature is for predicting the *target*.

Based on this, we can conclude and confirm that one of the most important factors in diagnosing panic disorder is sleep quality—it has the highest coefficient value. Additionally, the coefficient is negative, indicating that poor sleep quality is strongly associated with negative psychological outcomes. In other words, poor sleep habits may lead to or exacerbate panic disorder symptoms.

Alongside this feature are others such as severe panic attack symptoms and high stress levels, which likely suggests that a combination of long-term stress and poor habits (especially sleep-related) increases the likelihood of developing this psychological condition. Of course, additional factors—outside the scope of this study—can trigger panic attacks and should also be considered.

Conversely, the lowest entries in *Table 1* correspond to features that help protect individuals from developing panic disorder. These have positive logistic coefficients and, in some cases, high absolute values. Such variables include healthy habits like physical activity, good diet, low stress levels, etc. In other words, maintaining a healthy lifestyle (through exercise, nutrition, social support, etc.) and a low-stress environment may help prevent or manage mental health conditions like panic disorder.

6 Clustering Algorithm

In this section, an unsupervised learning method is implemented to discover patterns or groupings in the data that are not directly visible. That is, the goal is for the algorithm to cluster individuals with similar characteristics into different groups, with the intention of drawing conclusions and making inferences.

6.1 Hierarchical Clustering Algorithm with Gower Distance

As previously mentioned, the majority of the columns in the *dataset* are categorical. This means it is not feasible to compute purely numerical distances between observations for each feature. Many clustering algorithms (such as *k-means*, for instance) require this type of distance in order to group the data meaningfully.

For this reason, the first attempt involved using the *k-modes* algorithm, which is an extension of *k-means* that works specifically with categorical data. However, the results were not satisfactory—no meaningful or well-separated clusters were obtained.

A possible cause for this is the lack of consideration for the different types of categorical features in the dataset. Some are binary, others are ordinal (i.e., they have a defined order of importance), and others are nominal (unordered categories).

For this reason, the use of the *Gower Distance* was considered. Gower distance is the most popular and commonly used method to measure dissimilarity between observations with mixed data types (numerical and categorical). For numerical features, it computes the absolute difference between each pair of values; for categorical features, it assigns a value of 0 if the values are the same and 1 if they are different. It then averages the distances across all features and stores them in a matrix[6].

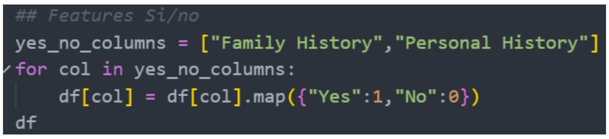
Once these values are obtained, a clustering algorithm that can work with precomputed distances—such as Gower—is needed. In this case, a hierarchical agglomerative clustering algorithm was used. These types of algorithms begin by treating each data point as its own cluster and then iteratively merge the closest pairs (based on the chosen distance metric) until only a single cluster containing all the data remains[7].

With this approach and the preprocessing applied to the data, the results were much more interpretable and acceptable.

6.2 Data Preprocessing

The first part of the process is identical to previous sections: the *Participant ID* column was removed, and missing values were filled using random observations from the same variables.

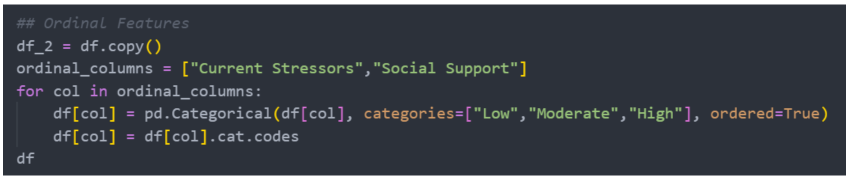
Next, categorical variables were encoded based on the type of value they contained. First, binary features with only two possible values (*Yes* and *No*) were processed. These were encoded as 1 (*Yes*) and 0 (*No*), since the responses resemble a binary activation state (on/off) without any intermediate value.



```
## Features Si/no
yes_no_columns = ["Family History","Personal History"]
for col in yes_no_columns:
    df[col] = df[col].map({"Yes":1,"No":0})
df
```

Figure 42: *Encoding of features with binary Yes/No values.*

Next, ordinal categories were handled. These are non-numeric features that nevertheless imply a ranking or order. For these, the lowest category was assigned a value of 0, the middle one a 1, and the highest a 2.



```
## Ordinal Features
df_2 = df.copy()
ordinal_columns = ["Current Stressors","Social Support"]
for col in ordinal_columns:
    df[col] = pd.Categorical(df[col], categories=["Low","Moderate","High"], ordered=True)
    df[col] = df[col].cat.codes
df
```

Figure 43: *Encoding of ordinal categorical features (0 = lowest, 1 = middle, 2 = highest).*

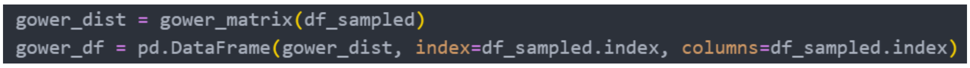
Finally, features that did not fall into any of the above types were left unchanged (i.e., no one-hot encoding was applied), since Gower distance is capable of handling raw categorical values directly.

This resulted in a dataset containing a combination of numeric and categorical variables.

Note: It is important to mention that the *Age* feature was discarded from the beginning, as it could introduce bias due to the high frequency of certain age values.

6.3 Model and Results

Before implementing the model, the pairwise distances between observations needed to be computed, as these are required for the algorithm to work. The distance computation was carried out as follows:

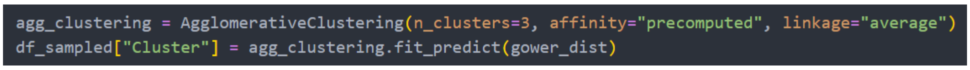


```
gower_dist = gower_matrix(df_sampled)
gower_df = pd.DataFrame(gower_dist, index=df_sampled.index, columns=df_sampled.index)
```

Figure 44: *Creation of the Gower distance matrix and conversion to a DataFrame.*

It is extremely important to note that using this process with the full set of $100,000$ data points would generate a $100,000 \times 100,000$ matrix, containing 1×10^{10} distance values—one for every pair of data points. This would be impossible to store on the computing equipment available. For this reason, a sample of 5,000 data points was selected, which results in approximately 25 million values—much more manageable.

Once the distance matrix was generated, the hierarchical clustering algorithm was implemented.

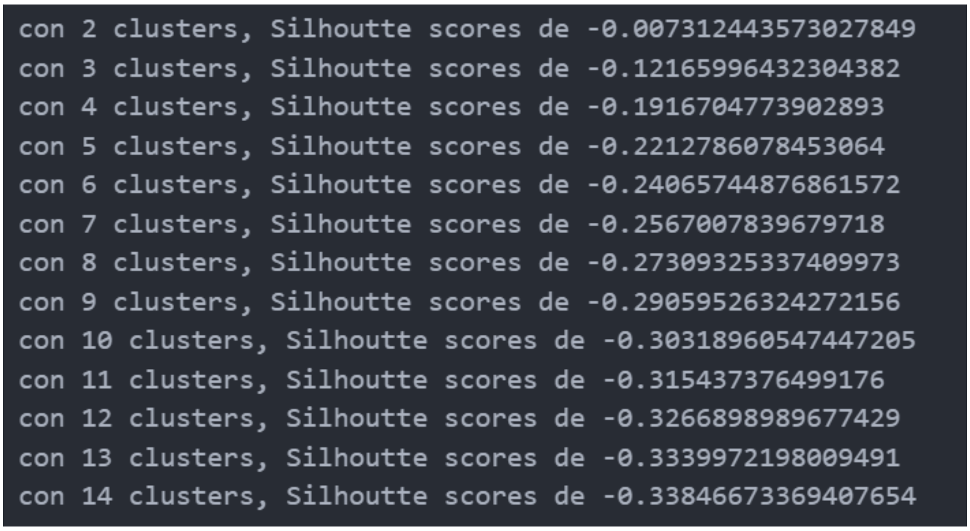


```
agg_clustering = AgglomerativeClustering(n_clusters=3, affinity="precomputed", linkage="average")
df_sampled["Cluster"] = agg_clustering.fit_predict(gower_dist)
```

Figure 45: *Model implementation using 3 clusters, a custom distance metric, and average linkage.*

To evaluate the quality of the clusters, the *Silhouette Score* was used. This coefficient measures how well each observation fits within its assigned cluster compared to others. It ranges from -1 to 1, with higher values indicating better-defined clusters[8].

Generally, values above 0.25 are considered acceptable. However, in this case, the *Silhouette Scores* obtained were extremely low—even negative—indicating that there is no meaningful pattern to cluster the data. Instead, the clustering appears random, resulting in significant overlap between clusters.



```
con 2 clusters, Silhoutte scores de -0.007312443573027849
con 3 clusters, Silhoutte scores de -0.12165996432304382
con 4 clusters, Silhoutte scores de -0.191670477390289З
con 5 clusters, Silhoutte scores de -0.2212786078453064
con 6 clusters, Silhoutte scores de -0.24065744876861572
con 7 clusters, Silhoutte scores de -0.2567007839679718
con 8 clusters, Silhoutte scores de -0.27309325337409973
con 9 clusters, Silhoutte scores de -0.29059526324272156
con 10 clusters, Silhoutte scores de -0.30318960547447205
con 11 clusters, Silhoutte scores de -0.315437376499176
con 12 clusters, Silhoutte scores de -0.3266898989677429
con 13 clusters, Silhoutte scores de -0.3339972198009491
con 14 clusters, Silhoutte scores de -0.33846673369407654
```

Figure 46: *Silhouette scores for different values of k (number of clusters).*

The most likely reason for the previously poor clustering performance is that many features had little relevance in terms of separating the data—rather than helping to distinguish groups, they *confused* the model, leading to significant overlap. A possible explanation is the presence of many similar instances due to the way similarity is computed, especially for categorical features, which can only take values of 0 or 1.

To address this issue, *feature engineering* was applied. The goal was to retain only the most relevant features and eliminate those that add noise.

The columns were filtered using a *Chi-squared test*, which was already discussed in a previous section. In this case, the goal was to identify statistically significant relationships between each feature and the cluster to which each instance was assigned (see the code in *Figure 45*). To perform this test, all variables were converted into *dummies*, since numerical values are required. The results are presented below.

Table 2: *Chi-squared test results to assess the statistical relationship between features and cluster assignments.*

Feature	Chi2 Score	p-value
Family History	745.386	1.4E162
Substance Use_Drugs	575.2788	1.2E125
Substance Use_Alcohol	572.0662	6E125
Demographics_Urban	488.1902	9.2E107
Demographics_Rural	468.2953	2E102
Personal History	107.8773	3.76E24
Medical History_Heart disease	18.76245	8.43E06
Current Stressors	11.95478	0.002535
Coping Mechanisms_Socializing	11.09095	0.002605
Coping Mechanisms_Meditation	11.60818	0.003015
Symptoms_Chest pain	10.02914	0.006641
Psychiatric History_Bipolar disorder	9.449938	0.008871
Medical History_Asthma	7.416063	0.024598
Gender_Female	7.387951	0.024873
Gender_Male	7.253252	0.026606
Lifestyle Factors_Sleep quality	5.536747	0.062764
Psychiatric History_Anxiety disorder	5.319069	0.069981
Severity	5.094804	0.078285
Symptoms_Dizziness	4.417204	0.109854

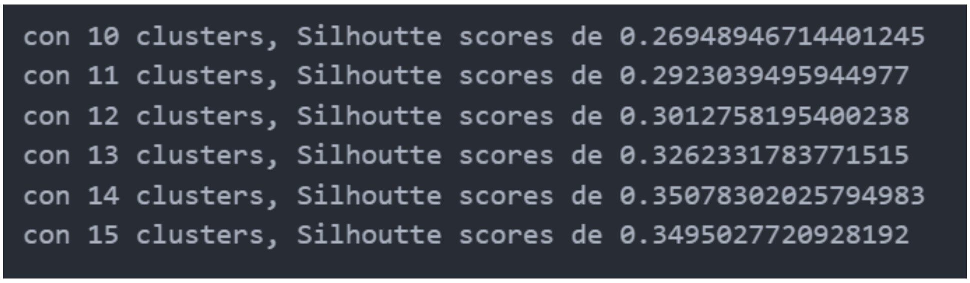
Coping Mechanisms_Exercise	4.180086	0.123682
Impact on Life	3.477422	0.175747
Psychiatric History_Depressive disorder	3.460441	0.176993
Symptoms_Shortness of breath	3.209465	0.200943
Medical History_Diabetes	2.818075	0.244378
Social Support	2.469809	0.290863
Lifestyle Factors_Exercise	2.22088	0.329414
Coping Mechanisms_Seeking therapy	1.430721	0.489016
Symptoms_Panic attacks	1.141133	0.565205
Lifestyle Factors_Diet	0.912581	0.63363
Symptoms_Fear of losing control	0.489648	0.782842

After analyzing the results, the following features were selected:

- *Family History*
- *Demographics*
- *Personal History*
- *Gender*
- *Impact on Life*
- *Current Stressors*
- *Substance Use*

These were the features most strongly associated with the assigned *clusters*.

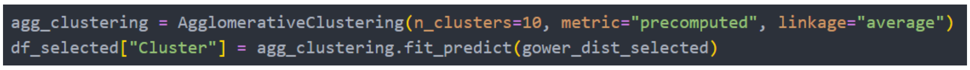
The silhouette values were recomputed after applying this feature selection strategy, and the results were as follows:



```
con 10 clusters, Silhoutte scores de 0.26948946714401245
con 11 clusters, Silhoutte scores de 0.2923039495944977
con 12 clusters, Silhoutte scores de 0.3012758195400238
con 13 clusters, Silhoutte scores de 0.3262331783771515
con 14 clusters, Silhoutte scores de 0.35078302025794983
con 15 clusters, Silhoutte scores de 0.3495027720928192
```

Figure 47: *Silhouette values for different numbers of clusters (k) after applying feature engineering.*

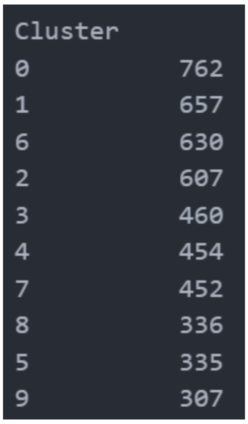
As seen, the values improved significantly—all of them now exceed the commonly accepted threshold. Therefore, clustering was performed using $K = 10$, as interpretability becomes increasingly complex with a larger number of clusters.



```
agg_clustering = AgglomerativeClustering(n_clusters=10, metric="precomputed", linkage="average")
df_selected["Cluster"] = agg_clustering.fit_predict(gower_dist_selected)
```

Figure 48: *Implementation of the hierarchical clustering model after feature engineering.*

Once a cluster was assigned to each data point, the distribution of the clusters was plotted to confirm that they were not generated randomly (i.e., ensuring that they do not all have the same size or distribution).



```
Cluster
0        762
1        657
6        630
2        607
3        460
4        454
7        452
8        336
5        335
9        307
```

Figure 49: *Distribution of the clusters. The left column shows the cluster number, and the right column indicates the number of data points in each cluster (sample of 5,000, not the full dataset).*

One useful method for interpreting these clusters is to visualize the most frequent features (modes) within each group. The following table summarizes the most common values for each feature in every cluster.

Table 3: *Summary of the most frequent (mode) values for each feature within each cluster.*

Cluster	Family History	Demographics	Personal History	Gender	Impact on Life	Current Stressors	Substance Use
0	1	Urban	0	Male	0	1	Drugs
1	0	Rural	1	Male	2	1	Alcohol
2	0	Urban	1	Male	0	1	Drugs
3	1	Rural	1	Male	0	0	Alcohol
4	1	Rural	0	Female	0	0	Alcohol
5	0	Rural	0	Female	2	0	Drugs
6	0	Urban	0	Male	2	2	Alcohol
7	1	Urban	1	Female	0	1	Drugs
8	0	Rural	0	Male	2	1	Alcohol
9	1	Rural	1	Male	2	1	Drugs

6.4 Visualization with T-SNE

One of the most important steps in working with clusters is visualizing them to gain better interpretability. However, in most cases, the dataset contains more than two dimensions, which makes a simple *scatter plot* infeasible. For this reason, dimensionality reduction techniques for visualizing high-dimensional data are used.

One of the best-known methods is *PCA (Principal Component Analysis)*, which maps the data into a lower-dimensional space using the most significant linear combinations of the original features. However, PCA is not suitable in this context due to the dominance of categorical variables in the dataset, since it requires numerical input.

As a result, a technique called *T-SNE (t-Distributed Stochastic Neighbor Embedding)* was selected. T-SNE is capable of preserving most of the local structure in high-dimensional data while also revealing meaningful global structure, such as the presence of clusters with varying densities.

This technique transforms Euclidean distances into conditional probabilities that reflect similarity between points. The similarity between a point **a** and a point **b** is interpreted as the conditional probability that **a** would choose **b** as its neighbor if neighbors were selected in proportion to their probability density under a t-distribution centered at **a** [9].

A parameter called *perplexity* is also used, which helps preserve the local structure by balancing attention between local and global aspects of the data. This may result in some points not being mapped clearly, depending on the setting. The final T-SNE visualization is shown below.

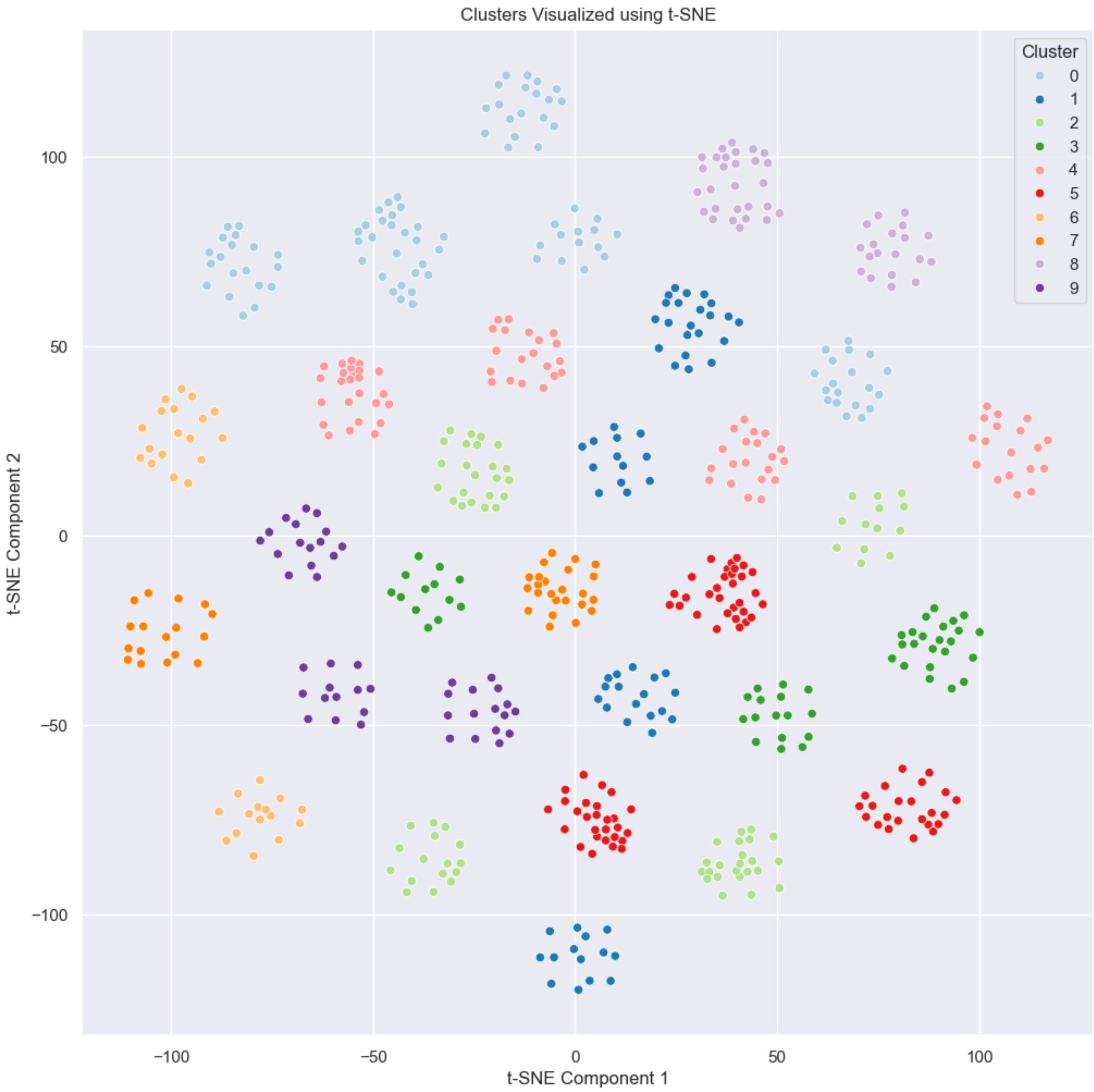


Figure 50: *T-SNE visualization of the clusters.*

This visualization presents several peculiarities. First, we observe that the number of displayed data points is lower (as previously explained, due to the *perplexity* parameter). Second, the clusters appear to be split into separate groups within the same plot. This may be due to several factors. One possibility is that T-SNE was unable to perfectly capture local relationships within clusters, as the method is highly sensitive to noise or complex structures. Another possible explanation is the presence of subgroups within the same clusters—although this is unlikely, given that all clusters appear to be clearly segmented.

Nonetheless, it is important to emphasize that the position of each point within the plot is not meaningful—the axes in T-SNE visualizations do not carry any direct interpretation. What truly matters is that each data point has been assigned to a cluster, enabling further interpretation.

6.5 Discussion

To interpret the results of the clustering analysis, the following figures present the categorical distribution of features within each cluster.

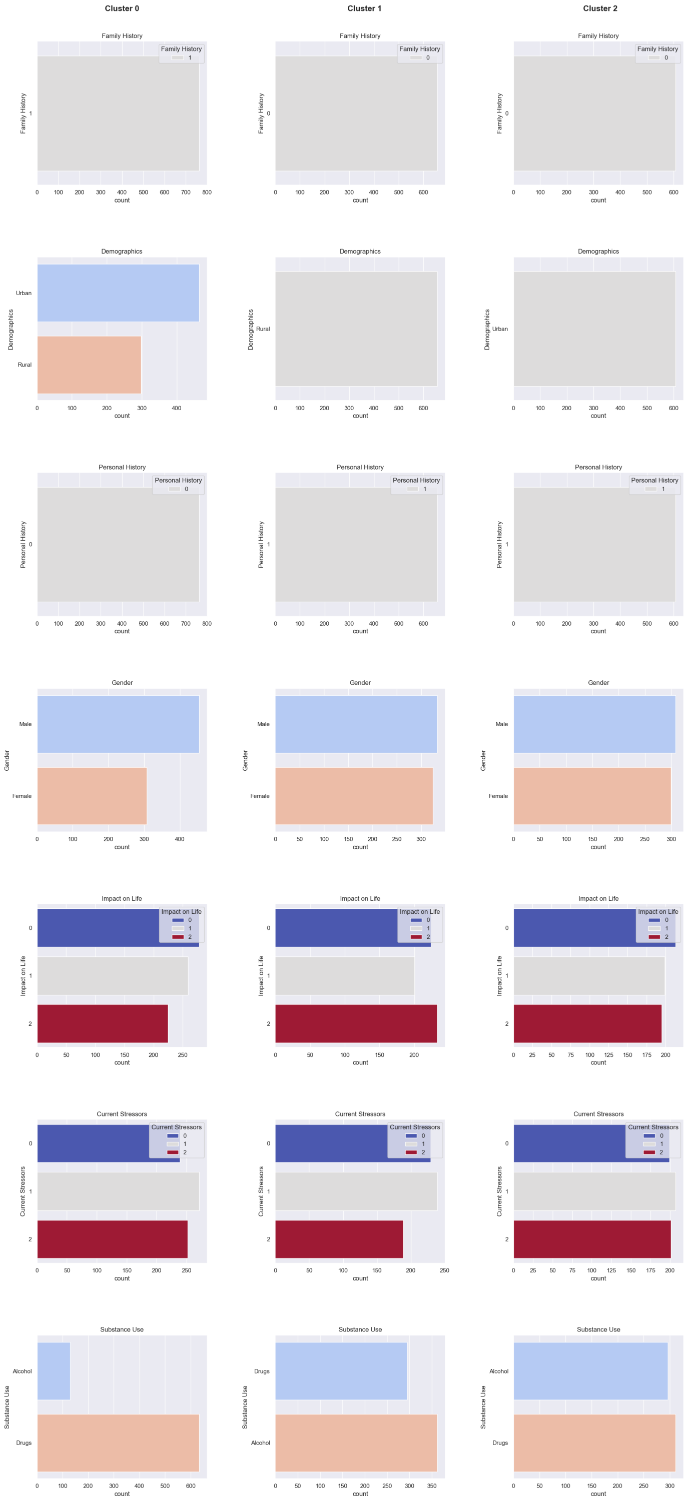


Figure 51: *Categorical distribution of features in Clusters 0, 1, and 2.*

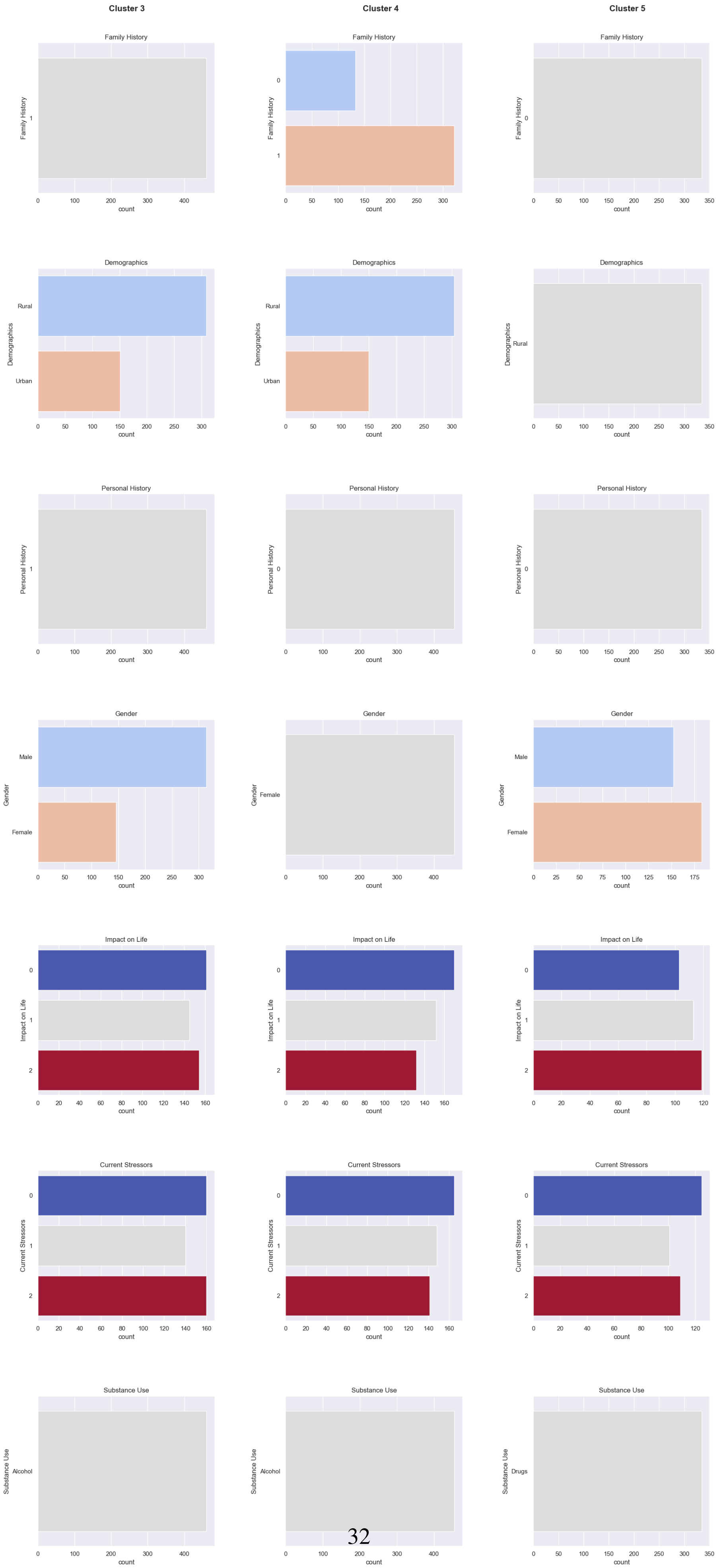


Figure 52: *Categorical distribution of features in Clusters 3, 4, and 5.*

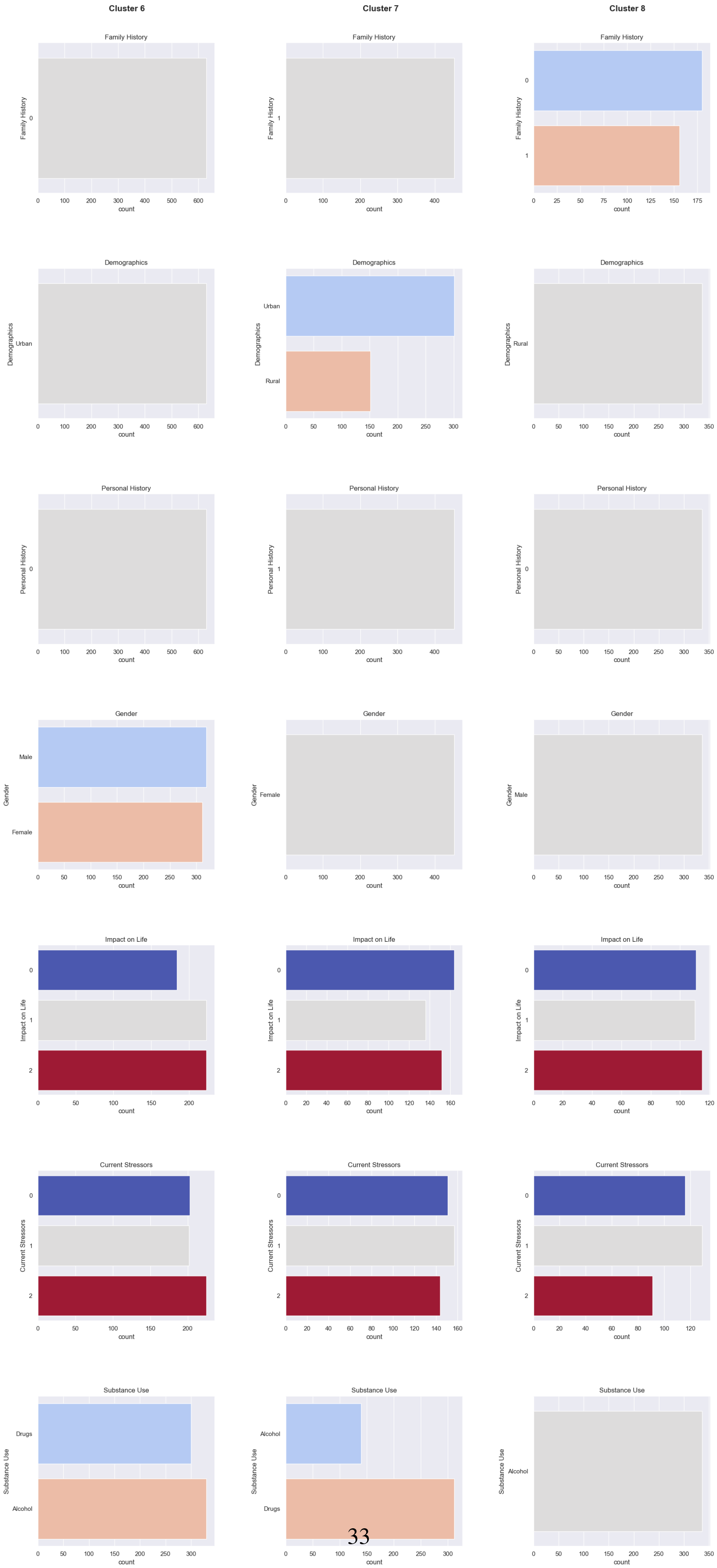


Figure 53: *Categorical distribution of features in Clusters 6, 7, and 8.*

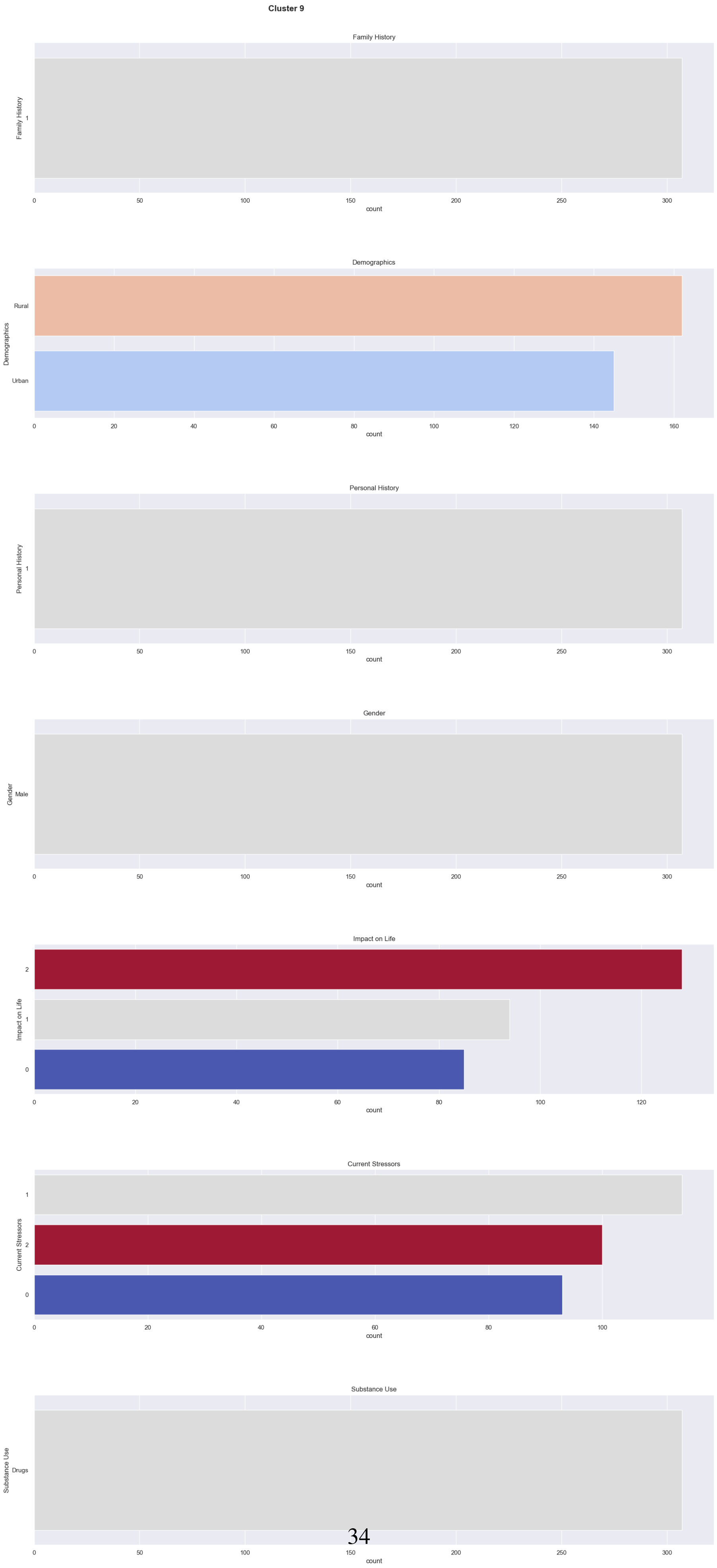


Figure 54: *Categorical distribution of features in Cluster 9.*

After analyzing the distributions of the clusters, the following group profiles were identified:

- **Cluster 0: Urban individuals with family predisposition and substance use**
  - This group likely experiences significant family-related problems, as their relatives have mental health conditions. Most of them consume drugs, possibly due to easier access in urban environments.
- **Cluster 1: Rural population with similar diagnoses and alcohol use**
  - These individuals have psychological conditions similar to panic disorder and also use harmful substances. However, since they live in rural areas where access to drugs is more limited, they tend to resort to alcohol.
- **Cluster 2: Urban individuals with similar diagnoses and substance use**
  - This group includes people who have experienced similar mental health disorders, although their family members have not. This suggests they deal with common, everyday stressors. They use substances moderately and are fairly balanced across urban and rural areas.
- **Cluster 3: Rural individuals with personal and family history who consume alcohol**
  - Similar to Cluster 1, this group includes individuals whose relatives suffer from panic-like conditions, and who themselves have been diagnosed with similar issues. All reside in rural areas and consume alcohol.
- **Cluster 4: Rural women who consume alcohol and have similar medical diagnoses**
  - This cluster consists of women from rural areas who all consume alcohol and generally have psychological conditions similar to panic disorder.
- **Cluster 5: Rural individuals who use drugs**
  - This group contains men and women who use harmful substances like drugs, but none have a history of mental illness. This suggests that they likely consume drugs recreationally, as their stress levels are moderate.
- **Cluster 6: Urban individuals who tend to consume harmful substances**
  - Similar to Cluster 5, but based entirely in urban areas. They show balanced stress levels and consume both drugs and alcohol evenly. This suggests they may use substances occasionally for leisure, as neither they nor their relatives have medical diagnoses.
- **Cluster 7: Women who use drugs and have personal and family medical history**
  - This group consists of women likely undergoing high stress and with genetic predisposition to mental disorders, as both they and their family members have related diagnoses. For this reason, they turn to drugs—more accessible in urban areas—for relief.
- **Cluster 8: Rural men who consume alcohol**
  - A cluster of men primarily located in rural areas who all consume alcohol. However, none of them have a related mental health diagnosis, indicating that most consume alcohol casually.
- **Cluster 9: Men with personal and family diagnoses who use drugs**
  - This final group consists exclusively of men, geographically balanced, all with personal and family histories of similar mental conditions. All use drugs—most likely to cope with their disorders.

7 General Conclusions

After implementing different classification, regression, and clustering models, the following conclusions were drawn:

- Proper and problem-specific preprocessing leads to significantly better results and facilitates improved data interpretation.
- Based on the exploratory data analysis and the implementation of a *Random Forest* model, it was determined that factors such as **sleep quality**, **stress level**, and **symptoms related to panic attacks** are strongly associated with a higher probability of receiving a panic disorder diagnosis.
- Due to the difficulty of diagnosing this condition and its low prevalence, the *Random Forest* model struggled to balance evaluation metrics. Specifically, as *recall* increased, *precision* decreased. Nevertheless, the model achieved an ***accuracy of 0.93***, with **0 false negatives and 1363 false positives**, meaning that out of 20,000 test instances, **18,637 were correctly classified**.
- It was demonstrated that the best way to address class imbalance in the target variable is by using model-intrinsic parameters, which allows the model to adapt to the imbalance ratio while preserving the original dataset. This approach was used in the *XGBoost* model, which achieved an ***accuracy of 0.99***, with **0 false negatives and 184 false positives**, correctly classifying **19,816 out of 20,000** test instances.
- In investigating the remaining misclassifications, it was concluded that most false positives and false negatives may result from medical misdiagnoses. Additionally, factors such as seeking social support, practicing stress-reduction techniques (e.g., meditation or exercise), likely contribute to mitigating the impact of such disorders.
- The implementation of a logistic regression model did not outperform the previously tested models, reaching an ***accuracy of 0.94***, with **3 false negatives and 959 false positives**. However, the most valuable insight from this model lies in *Table 1*, which provides the logistic coefficients of each feature, thereby identifying the most relevant variables for classification. In this case, poor sleep habits and a lifestyle that promotes high stress levels were found to increase the likelihood of panic disorder. Conversely, habits such as regular exercise, a healthy diet, and effective stress management were associated with a lower risk of such psychological disorders.
- Through the use of hierarchical clustering, which yielded a ***silhouette score of 0.27***, it was possible to identify several vulnerable population groups. Among the most notable were:
  - **Rural populations who consume alcohol and present related mental health diagnoses**.
  - **Women with a tendency toward drug use and both personal and familial medical histories**.
  - **Men who consume drugs and also have positive personal and family histories of psychological conditions**.

  Furthermore, it was found that individuals with similar mental health diagnoses (either personal or familial) are more likely to consume drugs than alcohol. On the other hand, individuals without such diagnoses tend to consume alcohol more frequently than drugs. This may suggest that some people use substances as a way to cope with or alleviate the symptoms of their psychological conditions.

References

[1] Bhandari, P. (2017). Stress and your health. In U.S. National Library of Medicine. StatPearls Publishing. https://www.ncbi.nlm.nih.gov/books/NBK430973/ .

[2] Mayo Clinic. (2023). Panic attacks and panic disorder – Symptoms and causes. Recuperado de https://www.mayoclinic.org/diseases-conditions/panic-attacks/symptoms-causes/syc-20376021.

[3] Azeem, M. S. (2023). Panic Disorder Detection Dataset [Data set]. Kaggle. https://www.kaggle.com/datasets/muhammadshahidazeem/panic-disorder-detection-dataset.

[4] Autor desconocido. (2019). Imbalanced Classification in Python: SMOTE-ENN Method. Towards Data Science. Recuperado de https://towardsdatascience.com/imbalanced-classification-in-python-smote-enn-method-db5db06b8d50/.

[5] Bølviken, E. (2017). Predicting the Presence of Panic Disorder in Patients with Chest Pain: A Machine Learning Approach (Tesis de maestría). Universidad Noruega de Ciencia y Tecnología. Recuperado de https://ntnuopen.ntnu.no/ntnu-xmlui/bitstream/handle/11250/2433761/16128_FULLTEXT.pdf.

[6] Autores desconocidos. (Fecha desconocida). Título del artículo. SSRN. Recuperado de https://download.ssrn.com/jbi/a063b9c7-66e9-46b4-ab0b-b045add1632a-meca.pdf,

[7] Müllner, D. (2011). Modern Hierarchical, Agglomerative Clustering Algorithms. arXiv preprint arXiv:1109.2378. Recuperado de https://arxiv.org/pdf/1109.2378.

[8] Řezanková, H. (2018). Different Approaches to the Silhouette Coefficient Calculation in Cluster Evaluation. En Proceedings of the 21st International Scientific Conference AMSE. Kutná Hora, República Checa. Recuperado de http://www.amse-conference.eu/wp-content/uploads/2018/09/AMSE_2018_Rezankova.pdf.

[9] van der Maaten, L., & Hinton, G. (2008). Visualizing Data using t-SNE. Journal of Machine Learning Research, 9, 2579-2605. Recuperado de http://www.jmlr.org/papers/volume9/vandermaaten08a/vandermaaten08a.pdf.