Project Report

On

Prediction of Personality Disorders using Machine Learning

"A dissertation submitted in partial fulfillment of the requirements of 8thSemester 2022 Project-II (CS-781) examination in Computer Science and Engineering of the Maulana Abul Kalam Azad University of Technology"

Submitted by

PROTYUSHA CHAUDHURI (10200119037)

SAMPURNA BISWAS (10200119021)

SUBHANKHI MAITI (10200119056)

ANKUR KUMAR (10200119009)

Under the guidance of

Prof.Swapan Kumar Mandal

Department of Computer Science and Engineering

Kalyani Government Engineering College

(Affiliated to Maulana Abul Kalam Azad University of Technology, West Bengal), Kalyani - 741235, Nadia, WB

Phone: 15826680 (PBX) Kalyani, 741235,

Nadia, West Bengal, India

Fax: 25821309

Ref. No.

Date:

Kalyani Government Engineering College

|Govt. of West Bengal|

Certificate of Approval

This is to certify that this report of B. Tech 8thSem, 2022 project, entitled **“**Prediction of Personality Disorders**”** is a record of bonafide work, carried out by Protyusha Chaudhuri, Sampurna Biswas, SubhankhiMaiti, Ankur Kumar under my supervision and guidance.In my opinion, the report in its present form is in partial fulfillment of all the requirements, as specified by the ***Kalyani Government Engineering College*** and as per regulations of the ***Maulana Abul Kalam Azad University of Technology***. In fact, it has attained the standard, necessary for submission. To the best of my knowledge, the results embodied in this report, are original in nature and worthy of incorporation in the present version of the report for the Project-II (CS-781) 8thSem B. Tech program in Computer Science and Engineering in the year 2022.

Guide / Supervisor

Prof. Swapan Kumar Mandal

Department of Computer Science and Engineering

Kalyani Government Engineering College

Examiner(s)

Head of theDepartment

Computer Science and Engineering

Kalyani Government Engineering College

ACKNOWLEDGEMENTS

We prepared this project report after the completion of our project “Prediction of Personality Disorders using Machine Learning".

We are very grateful to Prof. Swapan Kumar Mandal, our project guide(Professor of the Department of Computer Science and Engineering, Kalyani Government EngineeringCollege) who helped us to undertake the project by providing continuous support andassistance.

We would like to express our heartiest gratitude to Prof. Supriyo Banerjee (B. Tech Project Coordinator of the Department of Computer Science and Engineering, Kalyani Govt. Engineering College) and H.O.D of the Department of Computer Science and Engineering Dr. Koushik Dasgupta, Kalyani Government Engineering College, who allowed us to undertake this project.

Our special thanks to all the faculty members of Kalyani Govt. Engineering College, who rendered their help during the period of our study and project work.

Place: Kalyani

Date:

PROTYUSHA CHAUDHURI
B. Tech (CSE)
Roll No: 10200119037

SAMPURNA BISWAS
B. Tech (CSE)
Roll No: 10200119021

ANKUR KUMAR
B. Tech (CSE)
Roll No: 10200119009

SUBHANKHI MAITI
B. Tech (CSE)
Roll No: 10200119056

Abstract

A personality disorder is a type of mental disorder in which you have a rigid and unhealthy pattern of thinking, functioning, and behaving. A person with a personality disorder has trouble perceiving and relating to situations and people. This causes significant problems and limitations in relationships, social activities, work, and school.

Psychometric tests are popular and they can with the help of an algorithm fulfill the gap between a human brain and a complex problem that is hard to predict. When the intuition of the human brain is coupled with the complexities of the algorithm we may get a more accurate analysis of ourselves. This idea is simple, it puts together day-to-day information and life choices to predict the personality disorders that one may be suffering from in an accurate light. A cheaper option when paying for therapy might be costing 1000 bucks.

The most recent fifth edition of the Diagnostic and Statistical Manual of Mental Disorders (DSM-5) lists ten specific personality disorders. In this work, we have analyzed the dataset of different types of personality disorders. We aim to build a Machine Learning model which can predict whether a person might have any of the above-mentioned personality disorder

Index

1 Project Objective

1.1Study of DSM-5

The most recent fifth edition of the Diagnostic and Statistical Manual of Mental Disorders (DSM-5) lists ten specific personality disorders:

1. Paranoid personality disorder[1] is a pattern of distrust and suspiciousness such that others' motives are interpreted as malevolent.
2. Schizoid personality disorder[1] is a pattern of detachment from social relationships and a restricted range of emotional expression.
3. Schizotypal personality disorder[1] is a pattern of acute discomfort in close relationships, cognitive or perceptual distortions, and eccentricities of behaviour.
4. Antisocial personality disorder[2]** is a pattern of disregard for and violation of, the rights of others.
5. Borderline personality disorder[2] is a pattern of instability in interpersonal relationships, self-image, and affects, and marked impulsivity.
6. Histrionic personality disorder[2] is a pattern of excessive emotionality and attention seeking.
7. Narcissistic personality disorder[2] is a pattern of grandiosity, need for admiration, and lack of empathy.
8. Avoidant personality disorder[3] is a pattern of social inhibition, feelings of inadequacy, and hypersensitivity to negative evaluation.
9. Dependent personality disorder[3] is a pattern of submissive and clinging behavior related to an excessive need to be taken care of.
10. Obsessive-compulsive personality disorder[3] is a pattern of preoccupation with orderliness, perfectionism, and control.

**We have excluded the “Antisocial personality disorder” as the diagnosis of antisocial personality disorder is not given to individuals younger than 18 years and is given only if there is a history of some symptoms of conduct disorder before age 15 years. For individuals older than 18 years, a diagnosis of conduct disorder is given only if the criteria for an antisocial personality disorder are not met.

1.2Classification

We have grouped the above-mentioned 9 personality disorders into three clusters based on descriptive similarities.

Cluster A: includes paranoid, schizoid, and schizotypal personality disorders. Individuals with these disorders often appear odd or eccentric.

Cluster B: includes borderline, histrionic, and narcissistic personality disorders. Individuals with these disorders often appear dramatic, emotional, or erratic.

Cluster C: includes avoidant, dependent, and obsessive-compulsive personality disorders. Individuals with these disorders often appear anxious or fearful.

1.3 Data Analysis

Data cleaning is the process of fixing or removing incorrect, corrupted, incorrectly formatted, duplicate, or incomplete data within a dataset.

Exploratory Data Analysis (EDA):This is an approach to analyzing the data using visual techniques. It is used to discover trends, and patterns, or to check assumptions with the help of statistical summaries and graphical representations.Data cleaning is just one application of EDA

We have cleaned the existing data set using EDA and further removed extra unrequited columns. The data set contains no missing values and outliers (an observation that lies an abnormal distance from other values in a random sample from a population). The range of the dataset (the difference between the maximum and the minimum values) is from -3 to 3.

1.4 Metric analysis

In this work, we have analyzed the dataset of different types of personality disorders. We aim to build a Machine Learning model which can predict whether a person might have any of the above-mentioned personality disorders. Since this is a multiclass classification model the appropriate metrics to use would be Recall, Precision, and f1 score along with AUC score and accuracy.

Confusion Matrix: A confusion matrix[8] is a table that is used to define the performance of a classification algorithm. A confusion matrix visualizes and summarizes the performance of a classification algorithm.

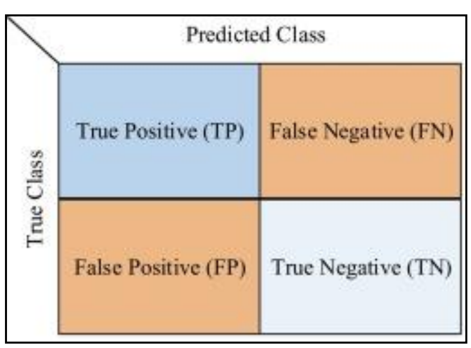

Fig 1.1 Confusion Matrix

1.4.1 Precision

Precision explains how many of the correctly predicted cases turned out to be positive. Precision is useful in cases where a False Positive (We predicted positive and it's false i.e. negative results are predicted positive) is a higher concern than False Negatives (We predicted negative and it's false i.e. positive

$$\boxed{Precision = \frac{TruePositive}{TruePositive + FalsePositive}}$$

results are predicted negative). Reducing false positives will increase the metric value.

Fig 1.2 Formula for precision

Precision for a label is defined as the number of true positives (We predicted a positive and it's true.) divided by the number of predicted positives.

1.4.2 Recall (Sensitivity)

Recall explains how many of the actual positive cases we were able to predict correctly with our model. It is a useful metric in cases where a False Negative is of higher concern than a False Positive. Reducing false positives will increase the metric value.

$$\boxed{Recall = \frac{TruePositive}{TruePositive + FalseNegative}}$$

Fig 1.3 Formula for recall

Recall for a label is defined as the number of true positives divided by the total number of actual positives.

1.4.3 F1 Score

It gives a combined idea about Precision and Recall metrics. It is the maximum when Precision is equal to Recall. This metric will balance out the recall and precision values equally.

The F1 Score is the harmonic mean of precision and recall.

$$\boxed{F1 = 2.\ \frac{Precision \ \times Recall}{Precision + Recall}}$$

Fig 1.4 Formulae for F1score

In medical cases, it is as important to check whether we raise a false alarm i.e. a person without any disorder gets diagnosed as an actual positive case(negative results are predicted positive) or a situation where a person with an actual disorder goes undiagnosed(positive results are predicted negative). Thus, we need higher precision for the first case, recall for the latter, and an F1 score to balance both of them.

1.4.4 AUC Score

AUC stands for "Area under the ROC Curve." That is, AUC measures the entire two-dimensional area underneath the entire ROC curve. AUC ranges in value from 0 to 1. A model whose predictions are 100% wrong has an AUC of 0.0; one whose predictions are 100% correct has an AUC of 1.0.

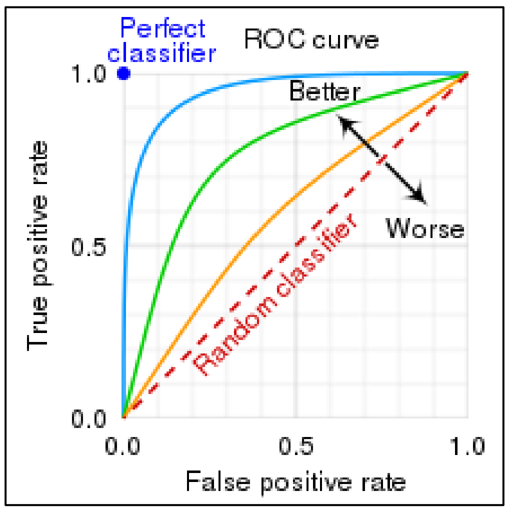

Fig. 1.5 ROC curve

1.4.5 Accuracy

Accuracy (ACC) is calculated as the number of all correct predictions divided by the total number of the dataset. The best accuracy is 1.0, whereas the worst is 0.0.

$$\boxed{\mathrm{ACC} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{TP} + \mathrm{TN} + \mathrm{FN} + \mathrm{FP}} = \frac{\mathrm{TP} + \mathrm{TN}}{\mathrm{P} + \mathrm{N}}}$$

Fig 1.6 Formulae for accuracy

2 Literature Survey

We have found the existing work in the field of personality-related studies which is the Prediction of Personality and Psychological Distress Using Natural Language Processing.

The current work introduces the study method and procedure of phase II which includes the interview questions for the five-factor model (FFM) of personality developed in phase I. This study aims to develop the interview (semi-structured) and open-ended questions for the FFM-based personality assessments, specifically designed with experts in the field of clinical and personality psychology (phase 1), and to collect the personality-related text data using the interview questions and self-report measures on personality and psychological distress (phase 2).

Self-report multiple choice questionnaires have been widely utilized to quantitatively measure one's personality and psychological constructs. Despite several strengths (e.g., brevity and utility), self-report multiple-choice questionnaires have considerable limitations in nature. With the rise of machine learning (ML) and Natural language processing (NLP), researchers in the field of psychology are widely adopting NLP to assess psychological constructs to predict human behaviors. However, there is a difference between these works and the predictions of personality disorders. It's because of the lack of connection between these being performed in computer science and psychology due to small data sets and invalidated modeling practices.

With the help of the available information and studying the report on this existing work we have worked on the prediction of personality disorders model.

The existing works in this field introduce the study method and the interview questions for the disorders using the DSM-5.

Furthermore, they aimed to examine the relationship between natural language data obtained from the interview questions, measuring the personality assessment to demonstrate the validity of the natural language-based personality disorder prediction and to make a Machine Learning model which can predict different types of personality disorders.

3 Proposed Work

Step 1 Learn Machine learning algorithms

Algorithms such as Logistic Regression, K-Nearest Neighbors, Naive Bayes, Decision Tree, Random Forest, AdaBoost, gradient boost, and XG boost. Also, principal component analysis is required (LDA, etc.) for the analysis of the data.

Deadline: 20th September

Step 2 Learn about personality disorders and their types

Researching the nine types of personality disorders. The exception is Anti-social personality disorder which couldn't be implemented due to complications in clinical processes.

Deadline: 30th September

Step 3 Identification of the column names/features in the dataset

76 survey questions i.e. a set of psychometric test questions have been prepared for the dataset whose relative answers will define the prediction of the presence of a disorder or not.

Deadline: 15th October

Step 4 Making of the dataset

The dataset is randomly generated by an algorithm as a survey data allocation may have reduced the number of cases for prediction.

Deadline: 31st October

Step 5 Logistic regression algorithms

Logistic Regression algorithm, hyperparameter tuning using GridSearch, RFE, feature selection using statistics

Deadline: 15th November

Step 6 Trees algorithms

Decision tree and random forest algorithm, hyperparameter tuning using GridSearch, RFE, feature selection using statistics

Deadline: 30th November

Step 7 Boosting, KNN, Naïve Bayes algorithms

KNN, Naïve Bayes, feature selection using statistics

Deadline: 31st December

Step 8 Neural Networks

Adaboost, Gradient Boost, XG boost algorithm, hyperparameter tuning using GridSearch, RFE, feature selection using statistics

Deadline: 31st January

Step 9 Testing on the Validation set and final model selection

Comparing the accuracy, precision, recall, and F1 score of all the algorithms to conclude which one accurately predicts the data with the least errors.

Deadline: 28th February

Step 10 Report writing

Detailed analysis report to be submitted for the project

Deadline: 31st March

4 Models

4.1 Logistic Regression

Logisticregression[4]estimates the probability of an event occurring, suchas voting or didn't vote, based on a given dataset of independent variables. Since the outcome is a probability, the dependent variable is bounded between 0 and 1.

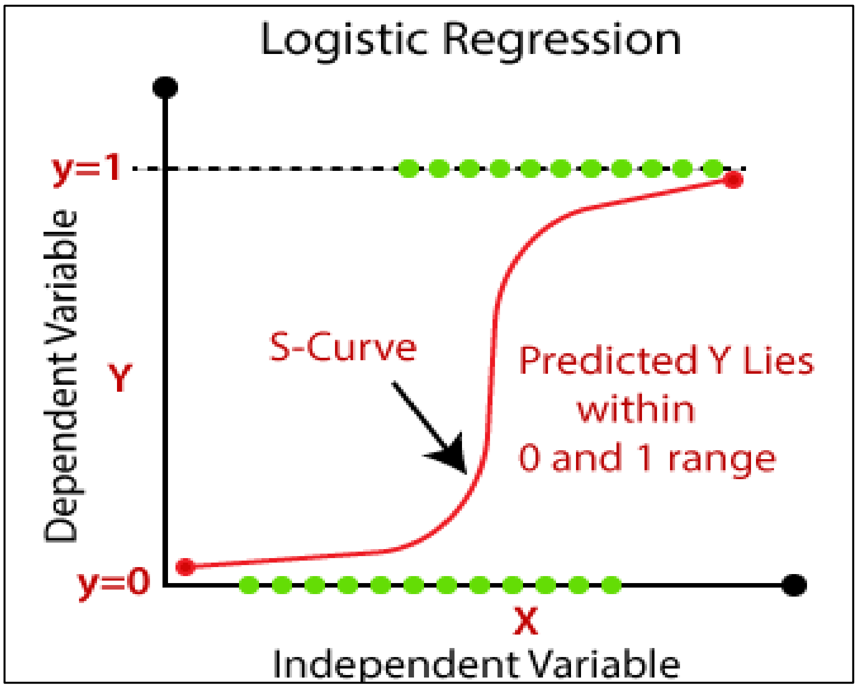

Fig 4.1 Logistic Regression Graph

Instead of least-squares, we make use of the maximum likelihood to find the best-fitting line in logistic regression. InMaximumLikelihoodEstimation[4], a probability distribution for the target variable (class label) is assumed and then a likelihood function is defined that calculates the probability of observing the outcome given the input data and the model. This function can then be optimized to find the set of parameters that result in the largest sum likelihood over the training dataset.

4.1.1 Model Building

Step 1: The target y is the Personality Disorders

The base model is made- model=linear_model.LogisticRegression()

Step 2: The next model is made with columns selected using feature selection fscols=pd.DataFrame(zip(Xtrain.columns,np.abs(model.coef_[0])))

Step 3: The next model is made with columns selected using feature selection (STATS MODEL)

P>|z| 5% is selected

Step 4: The next model is made with columns selected usingrecursive feature selection

rfemodel =feature_selection.RFE(estimator=model)

4.2 Decision Tree Model

The decision tree[5] uses the tree representation to solve the problem in which each leaf node corresponds to a class label and attributes are represented on the internal node of the tree. We can represent any Boolean function on discrete attributes using the decision tree.

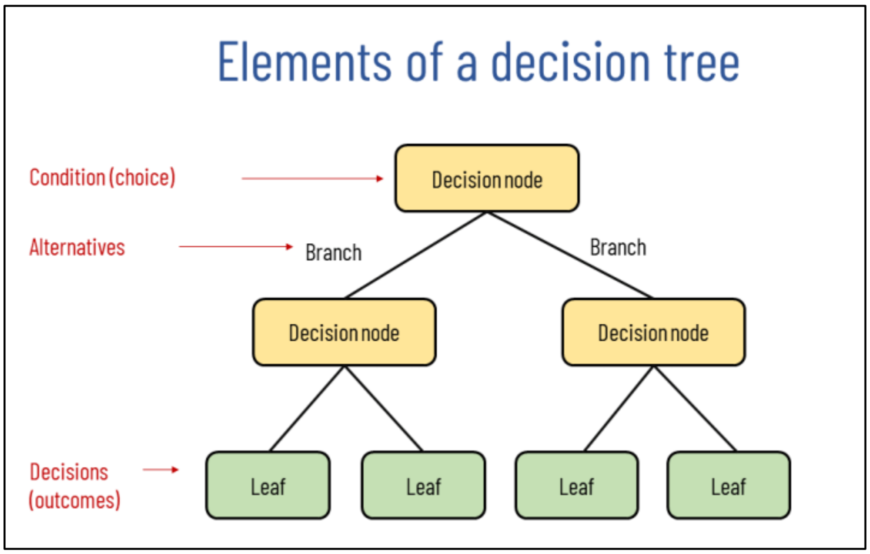

Fig 4.2A decision tree

The Gini impurity measure is one of the methods used in decision tree algorithms to decide the optimal split from a root node and subsequent splits. Gini Impurity tells us what the probability of misclassifying an observation is.

$$\begin{gathered} Ginx = p_1 \cdot (1 - p_1) + p_2 \cdot (1 - p_2) \\ equivalently, \\ Ginx = 2 \cdot p_1 p_2 \end{gathered}$$

Fig 4.3 Formula for GINI index

4.2.1 Model Building

Step 1: The target y is the Personality Disorders

The base model is made- model = tree.DecisionTreeClassifier(random_state=42)

Step 2: The next model is made with columns selected using feature selection

fscols=pd.DataFrame(zip(Xtrain.columns,np.abs(model.coef_[0])))

Step 3: The next model is made with columns selected using feature selection (STATS MODEL)

P>|z| 5% is selected

Step 4: Hyperparameters tuning is done using GridSearchCV, a cross-validation technique.

Step 5: The next model is made with columns selected using recursive feature selection

rfemodel =feature_selection.RFE(estimator=model)

4.3Random Forest

Random forests[5] are an ensemble learning[9] method that operates by constructing a multitude of decision trees in parallel at training time. The training algorithm for random forests applies the general technique of bootstrap aggregating, or bagging, to tree learners.

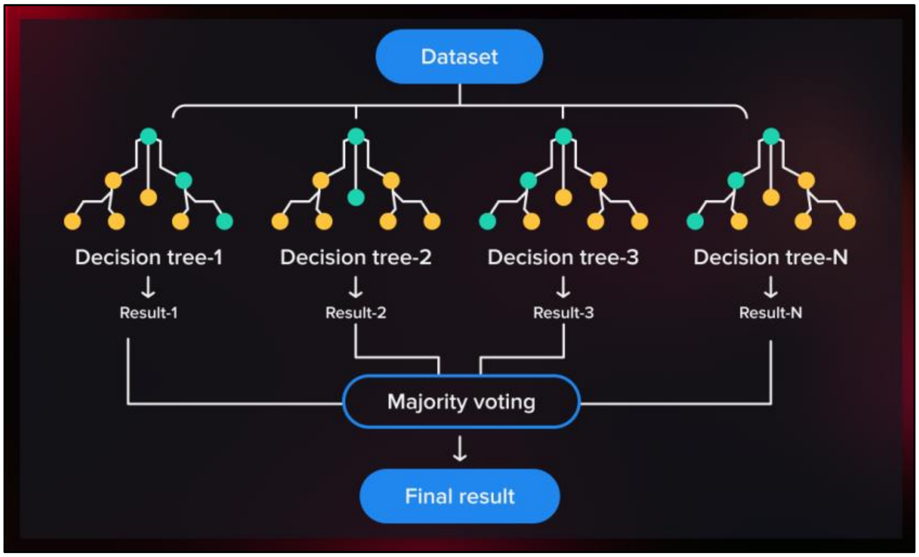

Fig 4.4Random forest using multiple decision trees

Gini Impurity[5] is also a measurement used to build a Random forest to determine how the features of a dataset should split nodes to form the tree. Information Gain[5], or IG for short, measures the reduction in entropy or surprise by splitting a dataset according to a given value of a random variable. A larger information gain suggests a lower entropy group or groups of samples and hence less surprise.

4.3.1 Model Building

Step 1: The target y is the Personality Disorders

The base model is made-model = ensemble.RandomForestClassifier(random_state=42)

Step 2: The next model is made with columns selected using feature selection (STATS MODEL)

P>|z| 5% is selected

Step 3: Hyperparameters tuning is done using GridSearchCV, a cross-validation technique.

Step 4: The next model is made with columns selected using recursive feature selection

rfemodel =feature_selection.RFE(estimator=model)

4.4 Naive Bayes

Naïve Bayes[7]is called Naïve because it assumes that the occurrence of a certain feature is independent of the occurrence of other features. It is called Bayes because it depends on the principle of Bayes' Theorem.

$$\underbrace{P(y/X)}_{\text{Posteriori}} = \frac{\overbrace{P(X/y)}^{\text{Likelihood}}\overbrace{P(y)}^{\text{Prior}}}{\underbrace{P(X)}_{\text{Predictor Prior}}}$$

Fig 4.5Bayes Theorem

Step 1: Convert the given dataset into frequency tables.

Step 2: Generate a Likelihood table by finding the probabilities of given features.

Step 3: Now, use the Bayes theorem to calculate the posterior probability.

Bernoulli Naive Bayes is a part of the Naive Bayes family. It is based on the Bernoulli Distribution[10] and accepts only binary values, i.e., 0 or 1. If the features of the dataset are binary, then we can assume that Bernoulli Naive Bayes is the algorithm to be used.

4.4.1 Model Building

Step 1: The target y is the Personality Disorders

The base model is made- model=naive_bayes.BernoulliNB()

This classifier is suitable for discrete data

Step 2: The next model is made with columns selected using feature selection

fscols=pd.DataFrame(zip(Xtrain.columns,np.abs(model.coef_[0])))

Step 3: The next model is made with columns selected using feature selection (STATS MODEL)

P>|z| 5% is selected

4.5 K-Nearest Neighbors

KNN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories.

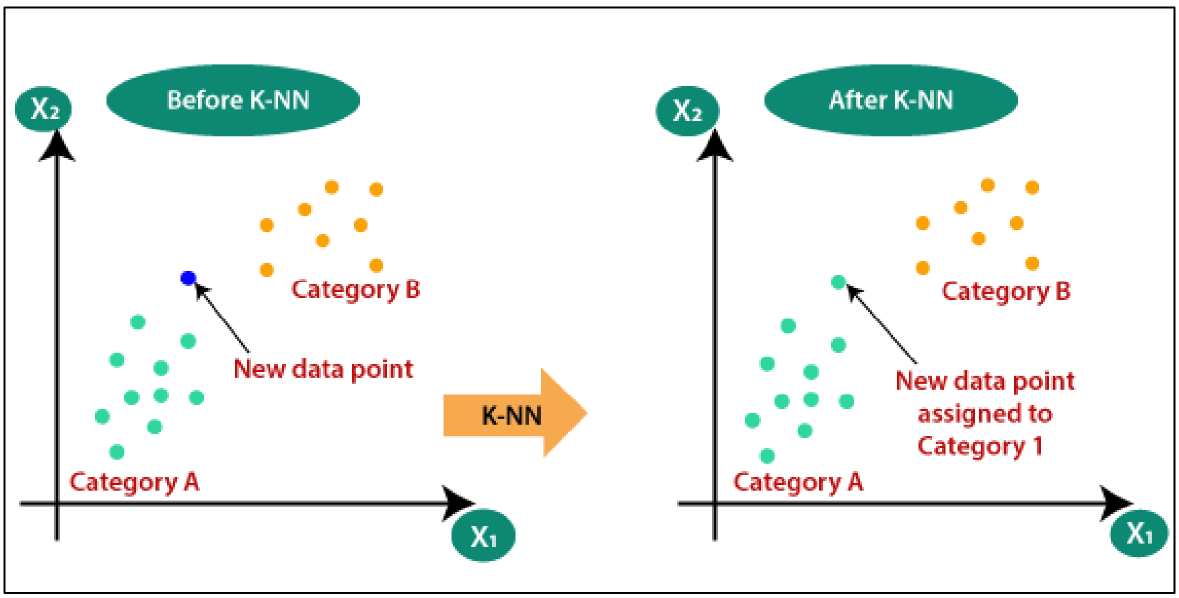

Fig 4.6 Working of K-Nearest Neighbor

KNN is a non-parametric algorithm, which means it does not make any assumptionsabout underlying data. It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset.

4.5.1 Model Building

Step 1: The target y is the Personality Disorders

The base model is made-

model = neighbors.KNeighborsClassifier(n_neighbors=3, algorithm='ball_tree')

Step 2: The next model is made with columns selected using feature selection

fscols=pd.DataFrame(zip(Xtrain.columns,np.abs(model.coef_[0])))

Step 3: The next model is made with columns selected using feature selection (STATS MODEL)

P>|z| 5% is selected

4.6 Adaboost

Adaboost[6] helps you combine multiple “weak classifiers” into a single “strong classifier”. The weak learners in Adaboost are decision trees with a single split, called decision stumps. Adaboost works by putting more weight on difficult-to-classify instances and less on those already handled well.

After training a classifier at any level, Adaboost assigns weight to each training item. Misclassified item is assigned a higher weight so that it appears in the training subset of the next classifier with a higher probability. After each classifier is trained, the weight is assigned to the classifier as well based on accuracy. The more accurate classifier is assigned a higher weight so that it will have more impact on the outcome.

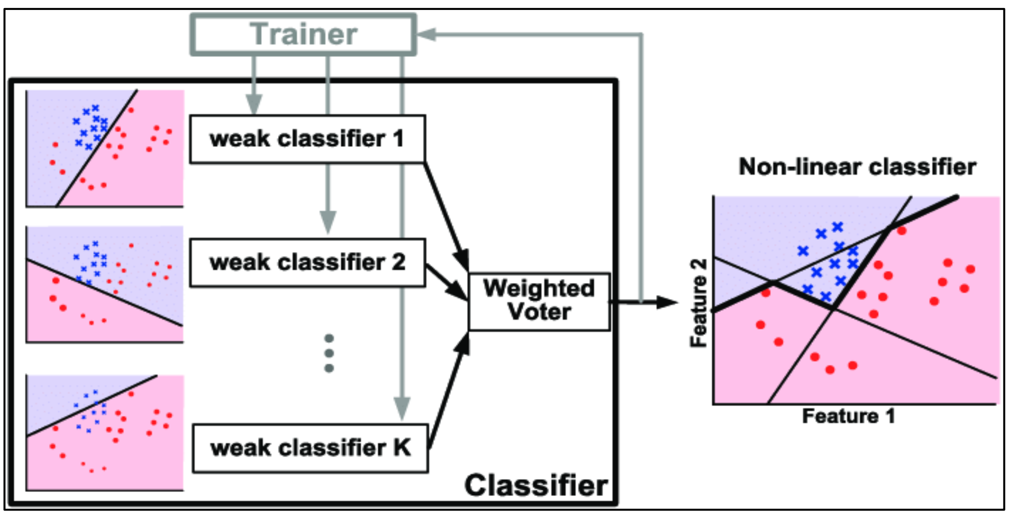

Fig 4.7Adaptive boosting working

4.6.1 Model Building

Step 1: The target y is the Personality Disorders

The base model is made-

model = ensemble.AdaBoostClassifier(random_state=42)

Step 2: The next model is made with columns selected using feature selection (STATS MODEL)

P>|z| 5% is selected

Step 4: Hyperparameters tuning is done using GridSearchCV, a cross-validation technique.

Step 5: The next model is made with columns selected using recursive feature selection

rfemodel =feature_selection.RFE(estimator=model)

4.7 Gradient Boost

In gradient boosting[6], we try to reduce the loss by adding decision trees. Also, we can minimize the error rate by cutting down the parameters. So we design the model in such a way that the addition of a tree does not change the existing tree.

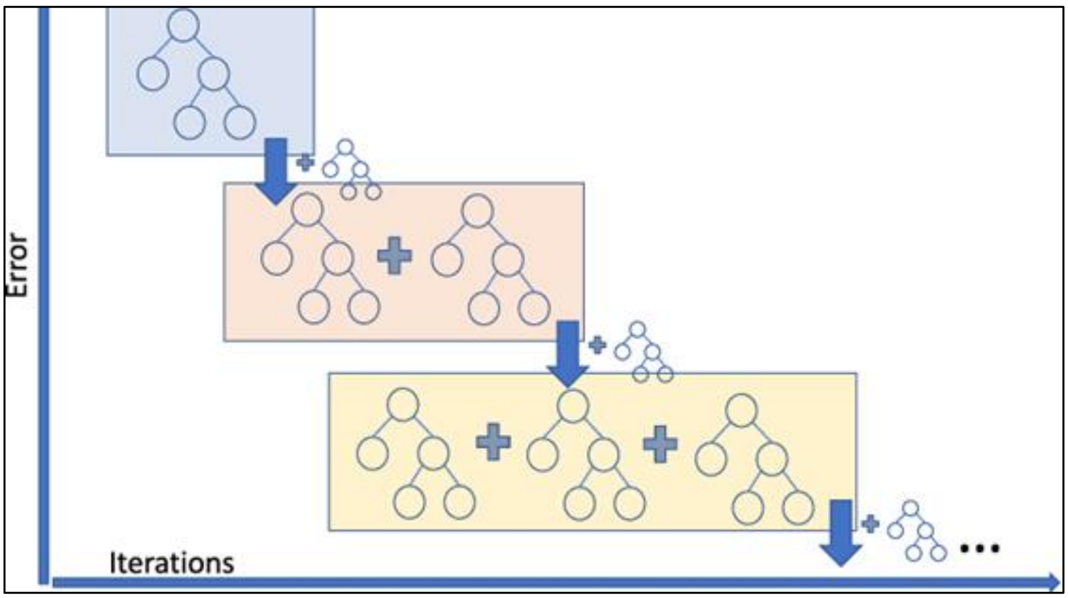

Fig 4.8Gradient Boost learning

It is a sequential ensemble learning technique where the performance of the model improves over iterations. This method creates the model in a stage-wise fashion. It infers the model by enabling the optimization of an absolute differentiable loss function[11]. As we add each weak learner, a new model is created that gives a more precise estimation of the response variable.

4.7.1 Model Building

Step 1: The target y is the Personality Disorders

The base model is made-

model = ensemble.GradientBoostingClassifier(random_state=42)

Step 2: The next model is made with columns selected using feature selection (STATS MODEL)

P>|z| 5% is selected

Step 4: Hyperparameters tuning is done using GridSearchCV, a cross-validation technique.

Step 5: The next model is made with columns selected using recursive feature selection

rfemodel =feature_selection.RFE(estimator=model)

4.8 Extreme Gradient Boost

XGBoost[6] algorithm is an extended version of the gradient boosting algorithm. It is designed to enhance the performance and speed of a model.

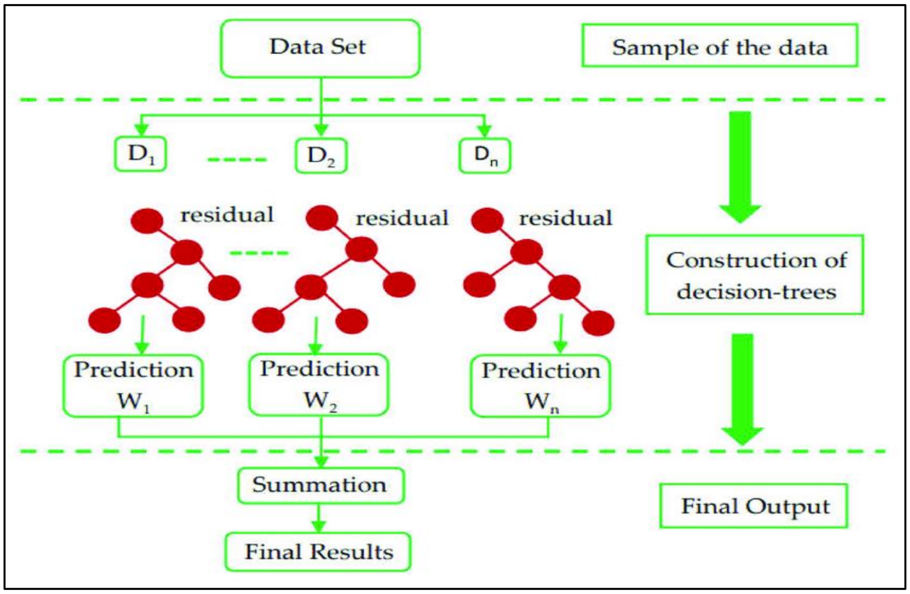

Fig 4.9 Extreme Gradient Boost working

XGBoost introduces a new metric called similarity score for node selection and splitting. Information gain gives the difference between old similarity and new similarity and thus tells how much homogeneity is achieved by splitting the node at a given point.

$$\boxed{\text{Similarity Score} = \frac{\text{Gradient}^2}{\text{Hessian} + \lambda}}$$

Fig 4.10Formulae for Similarity Score

4.8.1 Model Building

Step 1: The target y is the Personality Disorders

The base model is made-

model = xgb.XGBClassifier (random_state=42,objective='binary:logistic',eval_metric='aucpr',seed=42)

Step 2: The next model is made with columns selected using feature selection (STATS MODEL)

P>|z| 5% is selected

Step 4: Hyperparameters tuning is done using GridSearchCV, a cross-validation technique.

Step 5: The next model is made with columns selected using recursive feature selection

rfemodel =feature_selection.RFE(estimator=model)

5 Results

5.1 Logistic Regression

5.1.1 Paranoid Personality Disorder

n_features_to_select': 6

TRAIN	AUC 0.9099	Accuracy 0.9165	Precision 0.7816	Recall 0.8975	F1 0.8356
TEST	AUC 0.9236	Accuracy 0.9307	Precision 0.8053	Recall 0.9107	F1 0.8547

5.1.2 Schizoid Personality Disorder

'n_features_to_select': 15

TRAIN	AUC 0.8339	Accuracy 0.9289	Precision 0.7953	Recall 0.699	F1 0.7441
TEST	AUC 0.8359	Accuracy 0.9307	Precision 0.7111	Recall 0.7111	F1 0.7111

5.1.3 Schizotypal Personality Disorder

n_features_to_select': 8

TRAIN	AUC 0.866	Accuracy 0.9228	Precision 0.8363	Recall 0.7703	F1 0.8019
TEST	AUC 0.8375	Accuracy 0.9013	Precision 0.7622	Recall 0.7315	F1 0.7466

5.1.4 Borderline PersonalityDisorder

n_features_to_select': 10

TRAIN	AUC 0.7565	Accuracy 0.944	Precision 0.7423	Recall 0.5302	F1 0.6186
TEST	AUC 0.8081	Accuracy 0.952	Precision 0.7963	Recall 0.6324	F1 0.7049

5.1.5 Histrionic PersonalityDisorder

'n_features_to_select': 7

TRAIN	AUC 0.8499	Accuracy 0.9155	Precision 0.7678	Recall 0.7483	F1 0.7579

TEST	AUC 0.8807	Accuracy 0.936	Precision 0.8333	Recall 0.7955	F1 0.814

5.1.6 Narcissistic PersonalityDisorder

'n_features_to_select': 13

TRAIN	AUC 0.682	Accuracy 0.9593	Precision 0.7768	Recall 0.3702	F1 0.5014
TEST	AUC 0.6686	Accuracy 0.96	Precision 0.8235	Recall 0.3415	F1 0.4828

5.1.7 Avoidant Personality Disorder

'n_features_to_select': 17

TRAIN	AUC 0.8317	Accuracy 0.9311	Precision 0.7989	Recall 0.6924	F1 0.7419
TEST	AUC 0.8295	Accuracy 0.928	Precision 0.7789	Recall 0.6916	F1 0.7327

5.1.8 Dependent Personality Disorder

'n_features_to_select': 13

TRAIN	AUC 0.829	Accuracy 0.9282	Precision 0.7772	Recall 0.6905	F1 0.7313
TEST	AUC 0.824	Accuracy 0.9213	Precision 0.7723	Recall 0.6842	F1 0.7256

5.1.9 Obsessive-Compulsive Personality Disorder

'n_features_to_select': 11

TRAIN	AUC 0.7484	Accuracy 0.9435	Precision 0.759	Recall 0.5122	F1 0.6117
TEST	AUC 0.777	Accuracy 0.9493	Precision 0.8085	Recall 0.5672	F1 0.6667

5.2 Decision Tree Model

5.2.1 Paranoid Personality Disorder

n_features_to_select=5

min_samples_split=1300

TRAIN	AUC 0.9555	Accuracy 0.9362	Precision 0.7913	Recall 0.992	F1 0.8804
TEST	AUC 0.9609	Accuracy 0.9427	Precision 0.799	Recall 0.994	F1 0.8859

5.2.2 Schizoid Personality Disorder

n_features_to_select=8

min_samples_split=800

TRAIN	AUC 0.9666	Accuracy 0.9609	Precision 0.8031	Recall 0.9745	F1 0.8806
TEST	AUC 0.9545	Accuracy 0.9453	Precision 0.696	Recall 0.9667	F1 0.8093

5.2.3 Schizotypal Personality Disorder

n_features_to_select=9

min_samples_split=1200

TRAIN	AUC 0.9602	Accuracy 0.9504	Precision 0.8151	Recall 0.9768	F1 0.8887
TEST	AUC 0.9608	Accuracy 0.9533	Precision 0.8239	Recall 0.9732	F1 0.8923

5.2.4 Borderline PersonalityDisorder

n_features_to_select=9

min_samples_split=500

TRAIN	AUC 0.9601	Accuracy 0.9748	Precision 0.7995	Recall 0.9423	F1 0.8651
TEST	AUC 0.9773	Accuracy 0.9707	Precision 0.7614	Recall 0.9853	F1 0.859

5.2.5 Histrionic PersonalityDisorder

n_features_to_select=8

min_samples_split=1000

TRAIN	AUC 0.9563	Accuracy 0.9471	Precision 0.7822	Recall 0.9707	F1 0.8663
TEST	AUC 0.9603	Accuracy	Precision	Recall 0.9773	F1 0.8716

		0.9493	0.7866		

5.2.6 Narcissistic PersonalityDisorder

n_features_to_select=11

min_samples_split=300

TRAIN	AUC 0.9465	Accuracy 0.9821	Precision 0.7978	Recall 0.9064	F1 0.8486
TEST	AUC 0.98	Accuracy 0.984	Precision 0.7843	Recall 0.9756	F1 0.8696

5.2.7 Avoidant Personality Disorder

n_features_to_select=12

min_samples_split=800

TRAIN	AUC 0.9639	Accuracy 0.9593	Precision 0.7919	Recall 0.9704	F1 0.8721
TEST	AUC 0.9595	Accuracy 0.9507	Precision 0.7536	Recall 0.972	F1 0.849

5.2.8 Dependent Personality Disorder

n_features_to_select=15

min_samples_split=800

TRAIN	AUC 0.9643	Accuracy 0.9602	Precision 0.7943	Recall 0.97	F1 0.8734
TEST	AUC 0.972	Accuracy 0.9587	Precision 0.7902	Recall 0.9912	F1 0.8794

5.2.9 Obsessive-Compulsive Personality Disorder

n_features_to_select=9

min_samples_split=500

TRAIN	AUC 0.9565	Accuracy 0.9744	Precision 0.8023	Recall 0.935	F1 0.8636
TEST	AUC 0.9697	Accuracy 0.9693	Precision 0.7558	Recall 0.9701	F1 0.8497

5.3 Random Forest

5.3.1 Paranoid Personality Disorder

n_features_to_select=14

n_estimators= 70

max_features=10

min_samples_leaf=40

TRAIN	AUC 0.995	Accuracy 0.9976	Precision 1.0	Recall 0.99	F1 0.995
TEST	AUC 0.997	Accuracy 0.9987	Precision 1.0	Recall 0.994	F1 0.997

5.3.2 Schizoid Personality Disorder

n_estimators= 100

max_features=6

min_samples_leaf=35

n_features_to_select=7

TRAIN	AUC 0.9626	Accuracy 0.9565	Precision 0.7851	Recall 0.9713	F1 0.8683
TEST	AUC 0.9621	Accuracy 0.9587	Precision 0.7565	Recall 0.9667	F1 0.8488

5.3.3 Schizotypal Personality Disorder

n_estimators= 80

max_features=10

min_samples_leaf=30

n_features_to_select=16

TRAIN	AUC 0.9867	Accuracy 0.9946	Precision 1.0	Recall 0.9733	F1 0.9865
TEST	AUC 0.9857	Accuracy 0.9933	Precision 0.9932	Recall 0.9732	F1 0.9831

5.3.4 Borderline PersonalityDisorder

n_features_to_select=12

n_estimators=80

max_features=12

min_samples_leaf=50

TRAIN	AUC 0.9712	Accuracy 0.9951	Precision 1.0	Recall 0.9423	F1 0.9703
TEST AUC 0.9853	Accuracy 0.9973	Precision 1.0	Recall 0.9706	F1 0.9851	

5.3.5 Histrionic PersonalityDisorder

n_estimators= 70

max_features=7

min_samples_leaf=40

n_features_to_select=7

TRAIN	AUC 0.984	Accuracy 0.9944	Precision 1.0	Recall 0.968	F1 0.9838
TEST	AUC 0.9878	Accuracy 0.9947	Precision 0.9923	Recall 0.9773	F1 0.9847

5.3.6 Narcissistic PersonalityDisorder

n_estimators= 70

max_features=10

min_samples_leaf=40

n_features_to_select=11

TRAIN	AUC 0.943	Accuracy 0.9793	Precision 0.7653	Recall 0.9021	F1 0.8281
TEST	AUC 0.9751	Accuracy 0.9747	Precision 0.6897	Recall 0.9756	F1 0.8081

5.3.7 Avoidant Personality Disorder

n_features_to_select=10

n_estimators=50

max_features=6

min_samples_leaf=40

TRAIN	AUC 0.9834	Accuracy 0.9951	Precision 0.9983	Recall 0.9671	F1 0.9825
TEST	AUC 0.986	Accuracy 0.996	Precision 1.0	Recall 0.972	F1 0.9858

5.3.8 Dependent Personality Disorder

n_estimators= 70

max_features=5

min_samples_leaf=40

n_features_to_select=7

TRAIN	AUC 0.9606	Accuracy 0.9539	Precision 0.7661	Recall 0.97	F1 0.8561
TEST	AUC 0.9752	Accuracy 0.964	Precision 0.8129	Recall 0.9912	F1 0.8933

5.3.9 Obsessive-Compulsive Personality Disorder

n_estimators=50

max_features=7

min_samples_leaf=50

n_features_to_select=9

TRAIN	AUC 0.9518	Accuracy 0.9656	Precision 0.7388	Recall 0.935	F1 0.8254
TEST	AUC 0.969	Accuracy 0.968	Precision 0.7471	Recall 0.9701	F1 0.8442

5.4 Naive Bayes

5.4.1 Paranoid Personality Disorder

Train	AUC 0.9348	Accuracy 0.9047	Precision 0.7152	Recall 0.992	F1 0.8312
Test	AUC 0.9476	Accuracy 0.9187	Precision 0.7336	Recall 1.0	F1 0.8463

5.4.2 Schizoid Personality Disorder

Train	AUC 0.8292	Accuracy 0.9064	Precision 0.6706	Recall 0.7197	F1 0.6943
Test	AUC 0.8407	Accuracy 0.9053	Precision 0.5812	Recall 0.7556	F1 0.657

5.4.3 Schizotypal Personality Disorder

Train	AUC 0.9421	Accuracy 0.9221	Precision 0.7307	Recall 0.9756	F1 0.8356
Test	AUC 0.9283	Accuracy 0.9013	Precision 0.6744	Recall 0.9732	F1 0.7967

5.4.4 Borderline PersonalityDisorder

Train	AUC 0.5106	Accuracy 0.9155	Precision 0.7273	Recall 0.022	F1 0.0427
Test	AUC 0.5074	Accuracy 0.9107	Precision 1.0	Recall 0.0147	F1 0.029

5.4.5 Histrionic PersonalityDisorder

Train	AUC 0.936	Accuracy 0.9118	Precision 0.6731	Recall 0.9734	F1 0.7959
Test	AUC 0.9336	Accuracy 0.9053	Precision 0.6548	Recall 0.9773	F1 0.7842

5.4.6 Narcissistic PersonalityDisorder

Train	AUC 0.5245	Accuracy 0.9456	Precision 0.6	Recall 0.0511	F1 0.0941
Test	AUC 0.5115	Accuracy 0.9453	Precision 0.5	Recall 0.0244	F1 0.0465

5.4.7 Avoidant Personality Disorder

Train	AUC 0.5471	Accuracy 0.8638	Precision 0.6495	Recall 0.1036	F1 0.1787
Test	AUC 0.5296	Accuracy 0.8613	Precision 0.6364	Recall 0.0654	F1 0.1186

5.4.8 Dependent Personality Disorder

Train	AUC 0.6552	Accuracy	Precision	Recall 0.3328	F1 0.453

		0.8864	0.7092		
Test	AUC 0.6223	Accuracy 0.86	Precision 0.5818	Recall 0.2807	F1 0.3787

5.4.9 Obsessive-Compulsive Personality Disorder

Train	AUC 0.5279	Accuracy 0.9172	Precision 0.84	Recall 0.0569	F1 0.1066
Test	AUC 0.5135	Accuracy 0.9107	Precision 0.5	Recall 0.0299	F1 0.0563

5.5 K-Nearest Neighbors

5.5.1Paranoid Personality Disorder

n_neighbors=7

TRAIN	AUC 0.9563	Accuracy 0.9391	Precision 0.8003	Recall 0.9891	F1 0.8847
TEST	AUC 0.9362	Accuracy 0.924	Precision 0.763	Recall 0.9583	F1 0.8496

5.5.2 Schizoid Personality Disorder

n_neighbors=7

TRAIN	AUC 0.9463	Accuracy 0.9433	Precision 0.7398	Recall 0.9506	F1 0.8321
TEST	AUC 0.8972	Accuracy 0.912	Precision 0.5896	Recall 0.8778	F1 0.7054

5.5.3 Schizotypal Personality Disorder

n_neighbors=9

TRAIN	AUC 0.9463	Accuracy 0.9322	Precision 0.7614	Recall 0.9698	F1 0.8531
TEST	AUC 0.9208	Accuracy 0.8933	Precision 0.6575	Recall 0.9664	F1 0.7826

5.5.4 Borderline PersonalityDisorder

n_neighbors=9

TRAIN	AUC 0.9415	Accuracy 0.9591	Precision 0.6979	Recall 0.9203	F1 0.7938

TEST	AUC 0.9302	Accuracy 0.9453	Precision 0.6392	Recall 0.9118	F1 0.7515

5.5.5 Histrionic PersonalityDisorder

n_neighbors=11

TRAIN	AUC 0.9454	Accuracy 0.9334	Precision 0.7388	Recall 0.964	F1 0.8365
TEST	AUC 0.9368	Accuracy 0.9253	Precision 0.7159	Recall 0.9545	F1 0.8182

5.5.6 Narcissistic PersonalityDisorder

n_neighbors=13

TRAIN	AUC 0.8211	Accuracy 0.9647	Precision 0.6889	Recall 0.6596	F1 0.6739
TEST	AUC 0.7793	Accuracy 0.952	Precision 0.5581	Recall 0.5854	F1 0.5714

5.5.7 Avoidant Personality Disorder

n_neighbors=7

TRAIN	AUC 0.9481	Accuracy 0.9452	Precision 0.7395	Recall 0.9523	F1 0.8325
TEST	AUC 0.9362	Accuracy 0.924	Precision 0.6623	Recall 0.9533	F1 0.7816

5.5.8 Dependent Personality Disorder

n_neighbors=7

TRAIN	AUC 0.9571	Accuracy 0.9645	Precision 0.827	Recall 0.9468	F1 0.8829
TEST	AUC 0.9557	Accuracy 0.9493	Precision 0.7639	Recall 0.9649	F1 0.8527

5.5.9 Obsessive-Compulsive Personality Disorder

n_neighbors=5

TRAIN	AUC 0.9314	Accuracy 0.9598	Precision 0.7134	Recall 0.897	F1 0.7947
TEST	AUC 0.8892	Accuracy	Precision	Recall 0.8209	F1 0.7285

		0.9453	0.6548		

5.6 Adaboost

5.6.1 Paranoid Personality Disorder

n_features_to_select=6

n_estimators=10

TRAIN	AUC 0.9535	Accuracy 0.9336	Precision 0.7849	Recall 0.991	F1 0.876
TEST	AUC 0.9656	Accuracy 0.9467	Precision 0.8077	Recall 1.0	F1 0.8936

5.6.2 Schizoid Personality Disorder

n_estimators=10

n_features_to_select=8

TRAIN	AUC 0.9367	Accuracy 0.9426	Precision 0.7455	Recall 0.9283	F1 0.827
TEST	AUC 0.9293	Accuracy 0.9347	Precision 0.664	Recall 0.9222	F1 0.7721

5.6.3 Schizotypal Personality Disorder

n_estimators=10

n_features_to_select=7

TRAIN	AUC 0.9486	Accuracy 0.9332	Precision 0.7623	Recall 0.9745	F1 0.8554
TEST	AUC 0.9491	Accuracy 0.9347	Precision 0.7632	Recall 0.9732	F1 0.8555

5.6.4 Borderline PersonalityDisorder

n_features_to_select=20

n_estimators=40

TRAIN	AUC 0.6565	Accuracy 0.9341	Precision 0.78	Recall 0.3214	F1 0.4553
TEST	AUC 0.6331	Accuracy 0.9227	Precision 0.6786	Recall 0.2794	F1 0.3958

5.6.5 Histrionic PersonalityDisorder

n_estimators=10

n_features_to_select=8

TRAIN	AUC 0.9463	Accuracy 0.9296	Precision 0.7242	Recall 0.972	F1 0.83
TEST	AUC 0.9547	Accuracy 0.94	Precision 0.7544	Recall 0.9773	F1 0.8515

5.6.6 Narcissistic PersonalityDisorder

n_estimators=50

n_features_to_select=20

TRAIN	AUC 0.6867	Accuracy 0.9607	Precision 0.8091	Recall 0.3787	F1 0.5159
TEST	AUC 0.6463	Accuracy 0.9613	Precision 1.0	Recall 0.2927	F1 0.4528

5.6.7 Avoidant Personality Disorder

n_features_to_select=8

n_estimators=10

TRAIN	AUC 0.9602	Accuracy 0.9529	Precision 0.7642	Recall 0.9704	F1 0.8551
TEST	AUC 0.958	Accuracy 0.948	Precision 0.7429	Recall 0.972	F1 0.8421

5.6.8 Dependent Personality Disorder

n_estimators=20

n_features_to_select=7

TRAIN	AUC 0.9612	Accuracy 0.9548	Precision 0.7701	Recall 0.97	F1 0.8586
TEST	AUC 0.9705	Accuracy 0.956	Precision 0.7793	Recall 0.9912	F1 0.8726

5.6.9 Obsessive-Compulsive Personality Disorder

n_estimators=25

n_features_to_select=16

TRAIN	AUC 0.6588	Accuracy 0.9367	Precision 0.8623	Recall 0.3225	F1 0.4694
TEST	AUC 0.662	Accuracy 0.936	Precision 0.88	Recall 0.3284	F1 0.4783

5.7 Gradient Boost

5.7.1 Paranoid Personality Disorder

n_features_to_select=8

n_estimators=15

max_features=.33

min_samples_leaf=440

TRAIN	AUC 0.9139	Accuracy 0.9593	Precision 1.0	Recall 0.8279	F1 0.9058
TEST	AUC 0.881	Accuracy 0.9467	Precision 1.0	Recall 0.7619	F1 0.8649

5.7.2 Schizoid Personality Disorder

n_estimators=25

max_features=0.33

min_samples_leaf=400

n_features_to_select=8

TRAIN	AUC 0.9857	Accuracy 0.9958	Precision 1.0	Recall 0.9713	F1 0.9855
TEST	AUC 0.9826	Accuracy 0.9947	Precision 0.9886	Recall 0.9667	F1 0.9775

5.7.3 Schizotypal Personality Disorder

n_estimators=20

max_features=.26

min_samples_leaf=330

n_features_to_select=8

TRAIN	AUC 0.9449	Accuracy 0.9776	Precision 1.0	Recall 0.8898	F1 0.9417
TEST	AUC 0.9455	Accuracy 0.9773	Precision 0.9925	Recall 0.8926	F1 0.9399

5.7.4 Borderline PersonalityDisorder

n_features_to_select=10

n_estimators=40

max_features=.4

min_samples_leaf=350

TRAIN	AUC 0.9712	Accuracy 0.9951	Precision 1.0	Recall 0.9423	F1 0.9703
TEST	AUC 0.9853	Accuracy 0.9973	Precision 1.0	Recall 0.9706	F1 0.9851

5.7.5 Histrionic PersonalityDisorder

n_estimators=25

max_features=0.23

min_samples_leaf=330

n_features_to_select=7

TRAIN	AUC 0.984	Accuracy 0.9944	Precision 1.0	Recall 0.968	F1 0.9838
TEST	AUC 0.9878	Accuracy 0.9947	Precision 0.9923	Recall 0.9773	F1 0.9847

5.7.6 Narcissistic PersonalityDisorder

n_estimators=60

max_features=0.32

min_samples_leaf=400

n_features_to_select=12

TRAIN	AUC 0.9511	Accuracy 0.9946	Precision 1.0	Recall 0.9021	F1 0.9485
TEST	AUC 0.9878	Accuracy	Precision 1.0	Recall 0.9756	F1 0.9877

		0.9987			

5.7.7 Avoidant Personality Disorder

n_features_to_select=8

n_estimators=25

max_features=.32

min_samples_leaf=380

TRAIN	AUC 0.9834	Accuracy 0.9951	Precision 0.9983	Recall 0.9671	F1 0.9825
TEST	AUC 0.986	Accuracy 0.996	Precision 1.0	Recall 0.972	F1 0.9858

5.7.8 Dependent Personality Disorder

n_estimators=30

max_features=.26

min_samples_leaf=300

n_features_to_select=8

TRAIN	AUC 0.9842	Accuracy 0.9955	Precision 1.0	Recall 0.9684	F1 0.9839
TEST	AUC 0.9956	Accuracy 0.9987	Precision 1.0	Recall 0.9912	F1 0.9956

5.7.9 Obsessive-Compulsive Personality Disorder

n_estimators=60

max_features=.25

min_samples_leaf=370

n_features_to_select=10

TRAIN	AUC 0.9675	Accuracy 0.9944	Precision 1.0	Recall 0.935	F1 0.9664
TEST	AUC 0.9851	Accuracy 0.9973	Precision 1.0	Recall 0.9701	F1 0.9848

5.8 Extreme Gradient Boost

5.8.1 Paranoid Personality Disorder

n_features_to_select=8

n_estimators=15

min_child_weight=90

learning_rate=0.02

TRAIN	AUC 0.9806	Accuracy 0.9908	Precision 1.0	Recall 0.9612	F1 0.9802
TEST	AUC 0.9851	Accuracy 0.9933	Precision 1.0	Recall 0.9702	F1 0.9849

5.8.2 Schizoid Personality Disorder

n_estimators=20

min_child_weight=70

learning_rate=0.05

n_features_to_select=8

TRAIN	AUC 0.9435	Accuracy 0.9833	Precision 1.0	Recall 0.8869	F1 0.9401
TEST	AUC 0.9492	Accuracy 0.9867	Precision 0.9878	Recall 0.9	F1 0.9419

5.8.3 Schizotypal Personality Disorder

n_estimators=20

min_child_weight=75

learning_rate=0.03

n_features_to_select=8

TRAIN	AUC 0.9867	Accuracy 0.9946	Precision 1.0	Recall 0.9733	F1 0.9865
TEST	AUC 0.9857	Accuracy 0.9933	Precision 0.9932	Recall 0.9732	F1 0.9831

5.8.4 Borderline PersonalityDisorder

n_features_to_select=7

n_estimators=30

min_child_weight=35

learning_rate=0.06

TRAIN	AUC 0.9423	Accuracy 0.9901	Precision 1.0	Recall 0.8846	F1 0.9388
TEST	AUC 0.9338	Accuracy 0.988	Precision 1.0	Recall 0.8676	F1 0.9291

5.8.5 Histrionic PersonalityDisorder

n_estimators=30

min_child_weight=85

learning_rate=0.01

n_features_to_select=6

TRAIN	AUC 0.98	Accuracy 0.9929	Precision 1.0	Recall 0.9601	F1 0.9796
TEST	AUC 0.984	Accuracy 0.9933	Precision 0.9922	Recall 0.9697	F1 0.9808

5.8.6 Narcissistic PersonalityDisorder

n_estimators=20

min_child_weight=10

learning_rate=0.17

n_features_to_select=8

TRAIN	AUC 0.9511	Accuracy 0.9946	Precision 1.0	Recall 0.9021	F1 0.9485
TEST	AUC 0.9878	Accuracy 0.9987	Precision 1.0	Recall 0.9756	F1 0.9877

5.8.7 Avoidant Personality Disorder

n_features_to_select=6

n_estimators=20

min_child_weight=50

learning_rate=0.01

TRAIN	AUC 0.9834	Accuracy 0.9951	Precision 0.9983	Recall 0.9671	F1 0.9825
TEST	AUC 0.986	Accuracy 0.996	Precision 1.0	Recall 0.972	F1 0.9858

5.8.8 Dependent Personality Disorder

n_estimators=20

min_child_weight=50

learning_rate=0.01

n_features_to_select=6

TRAIN	AUC 0.9842	Accuracy 0.9955	Precision 1.0	Recall 0.9684	F1 0.9839
TEST	AUC 0.9956	Accuracy 0.9987	Precision 1.0	Recall 0.9912	F1 0.9956

5.8.9 Obsessive-Compulsive Personality Disorder

n_estimators=25

min_child_weight=35

learning_rate=0.07

n_features_to_select=5

TRAIN	AUC 0.9675	Accuracy 0.9944	Precision 1.0	Recall 0.935	F1 0.9664
TEST	AUC 0.9851	Accuracy 0.9973	Precision 1.0	Recall 0.9701	F1 0.9848

6 Final Models

6.1 Paranoid personality disorder

MODEL: ADA BOOST

6.1.1 Model Construction

```
model = ensemble.AdaBoostClassifier(random_state=42,n_estimators=10)

rfeobj =feature_selection.RFE(estimator=model,n_features_to_select=6)

rfeobj.fit(Xtrain,ytrain)

selected =Xtrain.columns[rfeobj.support_]

Xtrain1 = Xtrain[selected]

Xtest1 = Xtest[selected]

model = ensemble.AdaBoostClassifier(random_state=42,n_estimators=10)

model.fit(Xtrain1,ytrain)

predtrain= model.predict(Xtrain1)

predtest=model.predict(Xtest1)

print("TRAIN")

printmetric(ytrain,predtrain)

print("TEST")

printmetric(ytest,predtest)
```

6.1.2 Results

n_features_to_select=6

n_estimators=10

TRAIN	AUC 0.9535	Accuracy 0.9336	Precision 0.7849	Recall 0.991	F1 0.876
TEST	AUC 0.9656	Accuracy 0.9467	Precision 0.8077	Recall 1.0	F1 0.8936

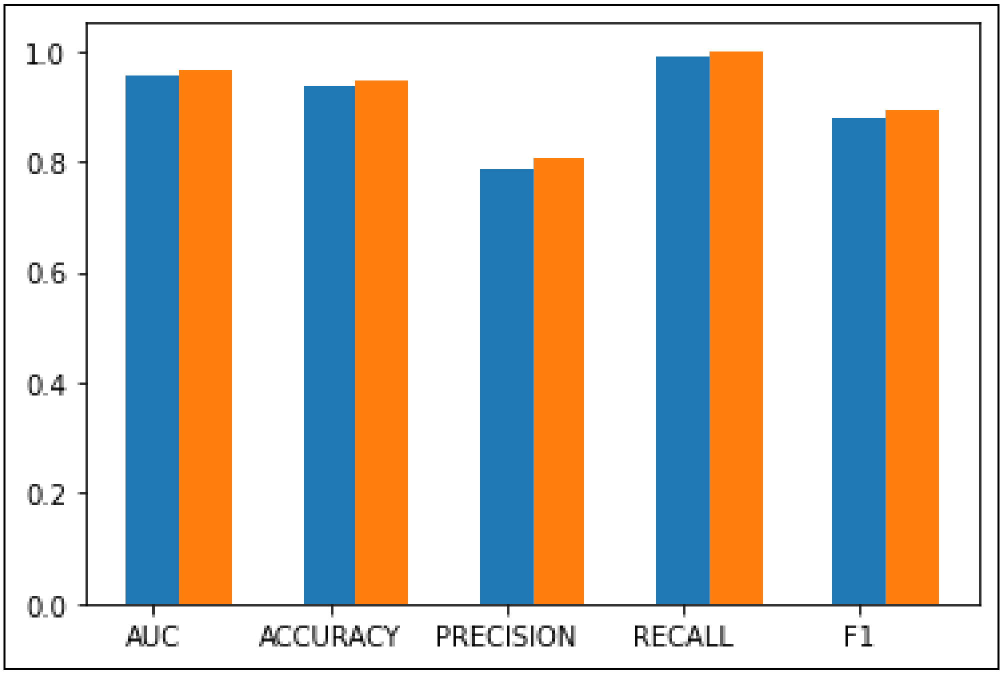

Fig 6.1 Training data and Test data comparison

Analysis: The score results are neither Overfit nor underfit. The precision is slightly lower than the recall value, indicating more false positive cases than false negatives. Overall, the F1 score is balanced. The accuracy is high with a good prediction of results and a higher AUC score that indicted how the model is significant.

6.2 Schizoid personality disorder

MODEL: XG BOOST

6.2.1 Model Construction

```
model =
xgb.XGBClassifier(random_state=42,objective='binary:logistic',eval_metric='auc',seed=42,n_estimators=
20,

min_child_weight=70,learning_rate=0.05)

rfeobj =feature_selection.RFE(estimator=model,n_features_to_select=8)

rfeobj.fit(Xtrain,ytrain)

selected =Xtrain.columns[rfeobj.support_]

Xtrain1 = Xtrain[selected]

Xtest1 = Xtest[selected]
```

```
model =
xgb.XGBClassifier(random_state=42,objective='binary:logistic',eval_metric='auc',seed=42,n_estimators=
20,min_child_weight=70,

learning_rate=0.05)

model.fit(Xtrain,ytrain)

predtrain= model.predict(Xtrain)

predtest=model.predict(Xtest)

print("TRAIN")

printmetric(ytrain,predtrain)

print("TEST")

printmetric(ytest,predtest)
```

6.2.2 Results

n_estimators=20

min_child_weight=70

learning_rate=0.05

n_features_to_select=8

TRAIN	AUC 0.9435	Accuracy 0.9833	Precision 1.0	Recall 0.8869	F1 0.9401
TEST	AUC 0.9492	Accuracy 0.9867	Precision 0.9878	Recall 0.9	F1 0.9419

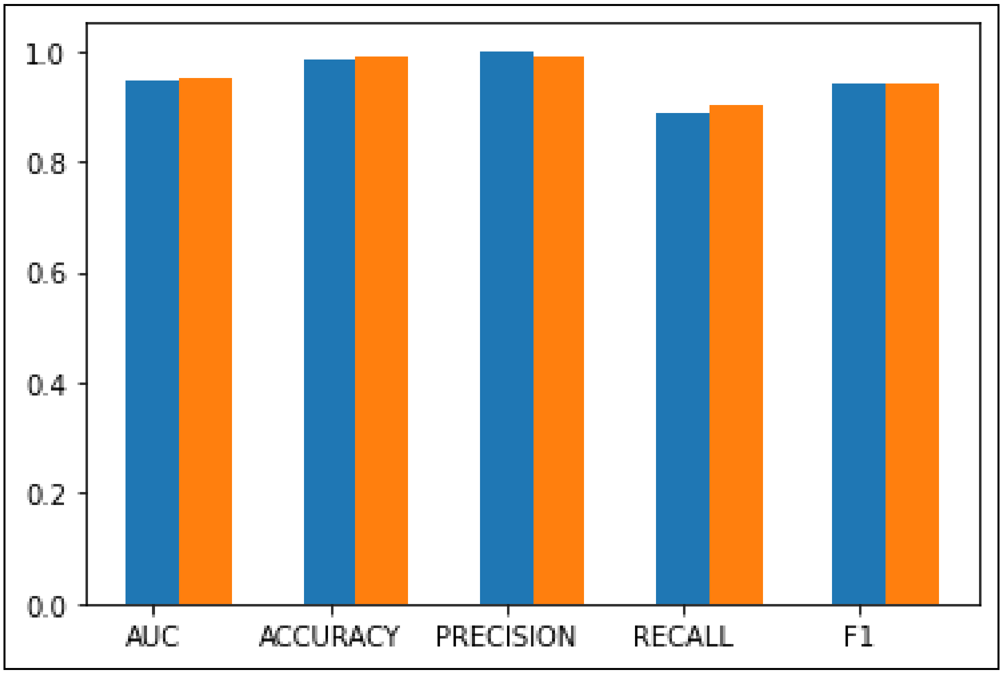

Fig 6.2 Training data and Test data comparison

Analysis: The score results are neither Overfit nor underfit. The recall is slightly lower than the precision value, indicating more false negative cases than false positives. Overall, the F1 score is balanced. The accuracy is high with a good prediction of results and a higher AUC score that indicted how the model is significant.

6.3 Schizotypal personality disorder

MODEL: GRADIENT BOOST

6.3.1 Model Construction

model = ensemble.GradientBoostingClassifier(random_state=42,n_estimators=20,max_features=0.26,min_samples_leaf=330)

rfeobj =feature_selection.RFE(estimator=model,n_features_to_select=8)

rfeobj.fit(Xtrain,ytrain)

selected =Xtrain.columns[rfeobj.support_]

Xtrain1 = Xtrain[selected]

Xtest1 = Xtest[selected]

```
model = ensemble.GradientBoostingClassifier(random_state=42,n_estimators=20,max_features=.26,min_samples_leaf=330)

model.fit(Xtrain1,ytrain)

predtrain= model.predict(Xtrain1)

predtest=model.predict(Xtest1)

print("TRAIN")

printmetric(ytrain,predtrain)

print("TEST")

printmetric(ytest,predtest)
```

6.3.2 Results

n_estimators=20

max_features=.26

min_samples_leaf=330

n_features_to_select=8

TRAIN	AUC 0.9449	Accuracy 0.9776	Precision 1.0	Recall 0.8898	F1 0.9417
TEST	AUC 0.9455	Accuracy 0.9773	Precision 0.9925	Recall 0.8926	F1 0.9399

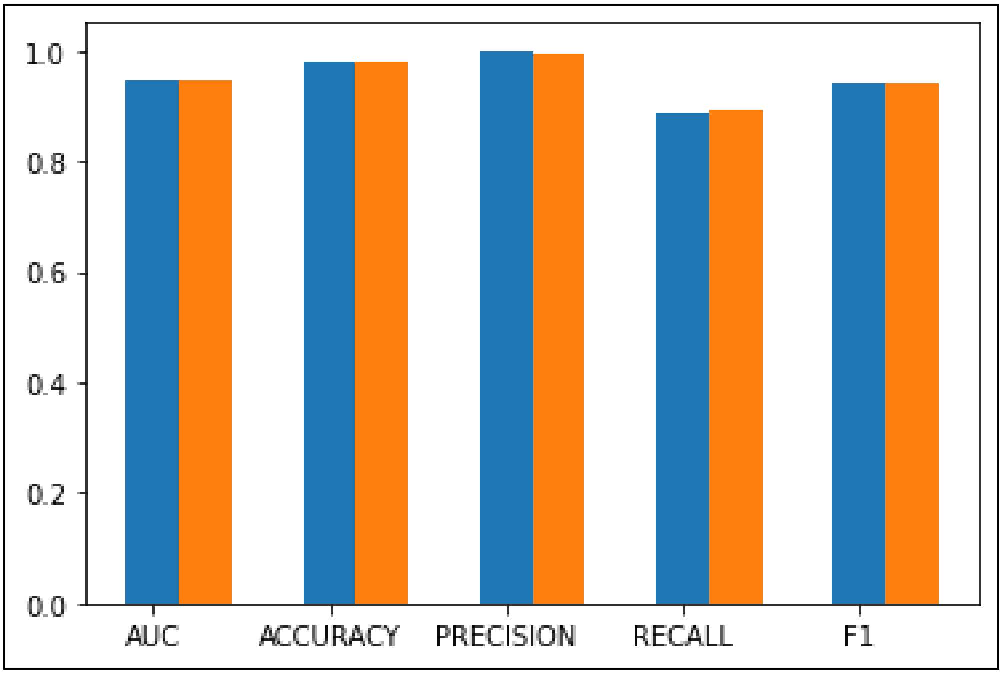

Fig 6.3 Training data and Test data comparison

Analysis: The score results are neither Overfit nor underfit. The recall is slightly lower than the precision value, indicating more false negative cases than false positives. Overall, the F1 score is balanced. The accuracy is high with a good prediction of results and a higher AUC score that indicted how the model is significant.

6.4 Borderline personality disorder

MODEL: XG BOOST

6.4.1 Model Construction

model = xgb.XGBClassifier(random_state=42,objective='binary:logistic',eval_metric='auc',seed=42,n_estimators=30,

min_child_weight=35,learning_rate=0.06)

rfeobj =feature_selection.RFE(estimator=model,n_features_to_select=7)

rfeobj.fit(Xtrain,ytrain)

selected =Xtrain.columns[rfeobj.support_]

Xtrain1 = Xtrain[selected]

Xtest1 = Xtest[selected]

model =
xgb.XGBClassifier(random_state=42,objective='binary:logistic',eval_metric='auc',seed=42,n_estimators=
30,

min_child_weight=35,learning_rate=0.06)

model.fit(Xtrain,ytrain)

predtrain= model.predict(Xtrain)

predtest=model.predict(Xtest)

print("TRAIN")

printmetric(ytrain,predtrain)

print("TEST")

printmetric(ytest,predtest)

6.4.2 Results

n_features_to_select=7

n_estimators=30

min_child_weight=35

learning_rate=0.06

TRAIN	AUC 0.9423	Accuracy 0.9901	Precision 1.0	Recall 0.8846	F1 0.9388
TEST	AUC 0.9338	Accuracy 0.988	Precision 1.0	Recall 0.8676	F1 0.9291

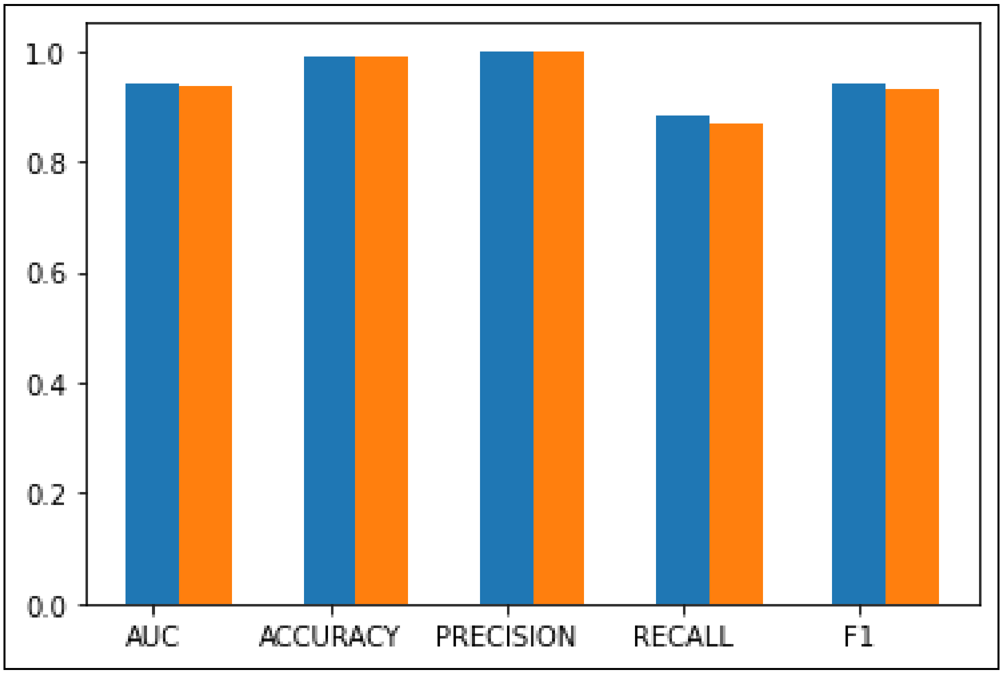

Fig 6.4 Training data and Test data comparison

Analysis: The score results are neither Overfit nor underfit. The recall is slightly lower than the precision value, indicating more false negative cases than false positives. Overall, the F1 score is balanced. The accuracy is high with a good prediction of results and a higher AUC score that indicted how the model is significant.

6.5 Histrionic personality disorder

MODEL: ADA BOOST

6.5.1 Model Construction

model = ensemble.AdaBoostClassifier(random_state=42,n_estimators=10)

rfeobj =feature_selection.RFE(estimator=model,n_features_to_select=8)

rfeobj.fit(Xtrain,ytrain)

selected =Xtrain.columns[rfeobj.support_]

Xtrain1 = Xtrain[selected]

Xtest1 = Xtest[selected]

model = ensemble.AdaBoostClassifier(random_state=42,n_estimators=10)

model.fit(Xtrain1,ytrain)

```
predtrain= model.predict(Xtrain1)

predtest=model.predict(Xtest1)

print("TRAIN")

printmetric(ytrain,predtrain)

print("TEST")

printmetric(ytest,predtest)
```

6.5.2 Results

n_estimators=10

n_features_to_select=8

TRAIN	AUC 0.9463	Accuracy 0.9296	Precision 0.7242	Recall 0.972	F1 0.83
TEST	AUC 0.9547	Accuracy 0.94	Precision 0.7544	Recall 0.9773	F1 0.8515

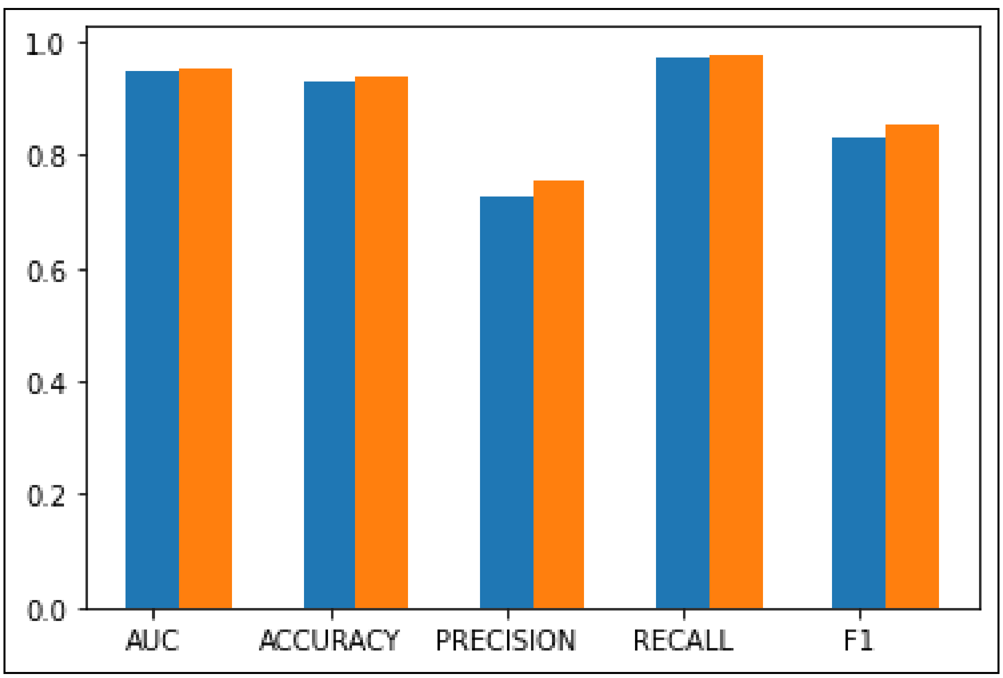

Fig 6.5 Training data and Test data comparison

Analysis: The score results are neither Overfit nor underfit. The precision is lower than the recall value, indicating more false positive cases than false negatives. Overall, the F1 score is balanced. The accuracy is high with a good prediction of results and a higher AUC score that indicted how the model is significant.

6.6 Narcissistic personality disorder

MODEL: DECISION TREE

6.6.1 Model Construction

```
model = tree.DecisionTreeClassifier(random_state=42,min_samples_split=300)

rfemodel =feature_selection.RFE(estimator=model)

pdict={'n_features_to_select':[11,12,13]} # dict with key hyperparameter name and value is a list

gridobj = model_selection.GridSearchCV(estimator=rfemodel,scoring='f1',param_grid=pdict,cv=5,

n_jobs=-1,return_train_score=True)

gridobj.fit(Xtrain,ytrain)

bestmodel =gridobj.best_estimator_

x=Xtrain.columns[bestmodel.support_]

Xtrain1 = Xtrain[x]

Xtest1 = Xtest[x]

model = tree.DecisionTreeClassifier(random_state=42,min_samples_split=300)

model.fit(Xtrain1,ytrain)

trainpred=model.predict(Xtrain1)

testpred=model.predict(Xtest1)

print("TRAIN")

printmetric(ytrain,trainpred)

print("TEST")

printmetric(ytest,testpred)
```

6.6.2 Results

n_features_to_select=11

min_samples_split=300

TRAIN	AUC 0.9465	Accuracy 0.9821	Precision 0.7978	Recall 0.9064	F1 0.8486

TEST	AUC 0.98	Accuracy 0.984	Precision 0.7843	Recall 0.9756	F1 0.8696

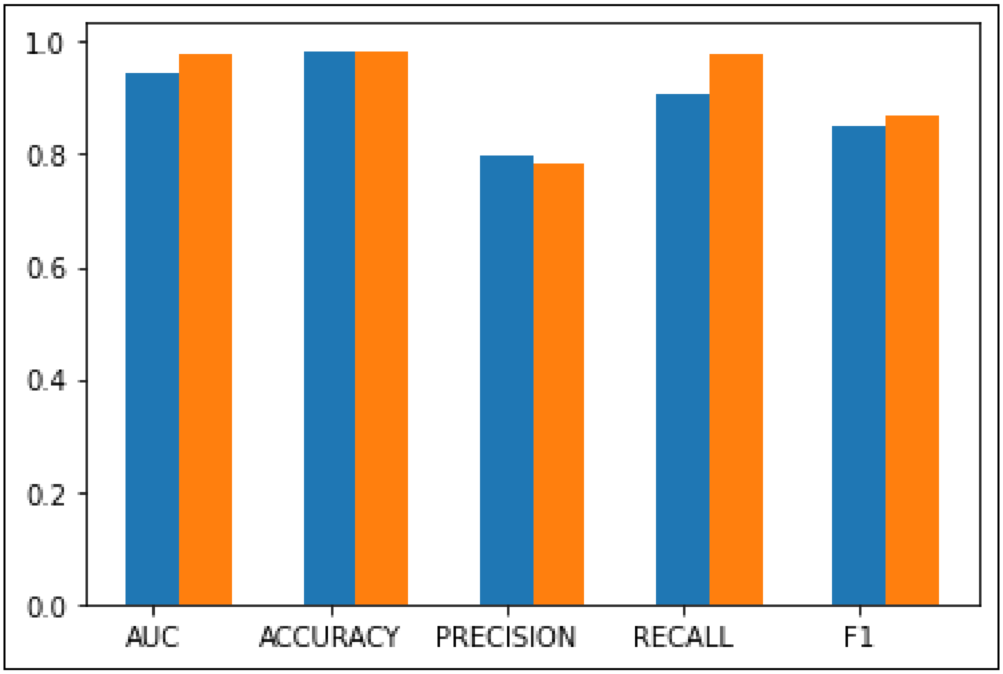

Fig 6.6 Training data and Test data comparison

Analysis: The score results are neither Overfit nor underfit. The precision is slightly lower than the recall value, indicating more false positive cases than false negatives. Overall, the F1 score is balanced. The accuracy is high with a good prediction of results and a higher AUC score that indicted how the model is significant.

6.7 Avoidant personality disorder

MODEL: ADA BOOST

6.7.1 Model Construction

model = ensemble.AdaBoostClassifier(random_state=42,n_estimators=10)

rfeobj =feature_selection.RFE(estimator=model,n_features_to_select=8)

rfeobj.fit(Xtrain,ytrain)

selected =Xtrain.columns[rfeobj.support_]

Xtrain1 = Xtrain[selected]

Xtest1 = Xtest[selected]

```
model = ensemble.AdaBoostClassifier(random_state=42,n_estimators=10)

model.fit(Xtrain1,ytrain)

predtrain= model.predict(Xtrain1)

predtest=model.predict(Xtest1)

print("TRAIN")

printmetric(ytrain,predtrain)

print("TEST")

printmetric(ytest,predtest)
```

6.7.2 Results

n_features_to_select=8

n_estimators=10

TRAIN	AUC 0.9602	Accuracy 0.9529	Precision 0.7642	Recall 0.9704	F1 0.8551
TEST	AUC 0.958	Accuracy 0.948	Precision 0.7429	Recall 0.972	F1 0.8421

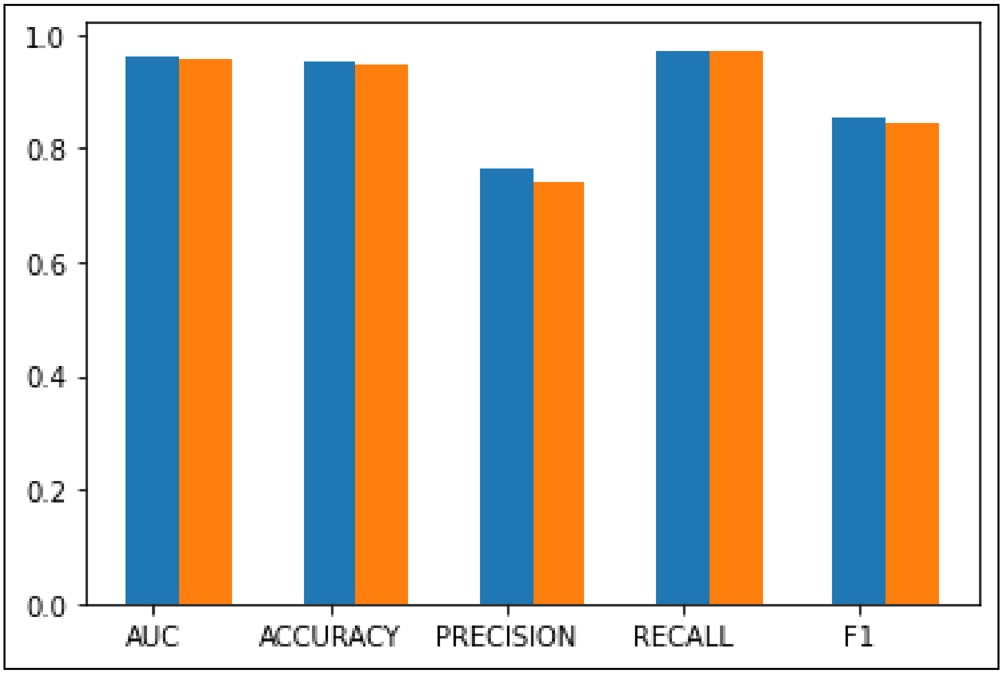

Fig 6.7 Training data and Test data comparison

Analysis: The score results are neither Overfit nor underfit. The precision is slightly lower than the recall value, indicating more false positive cases than false negatives. Overall, the F1 score is balanced. The accuracy is high with a good prediction of results and a higher AUC score that indicted how the model is significant.

6.8 Dependent personality disorder

MODEL: RANDOM FOREST

6.8.1 Model Construction

```
model = ensemble.RandomForestClassifier(random_state=42,n_estimators= 50,max_features=6,

min_samples_leaf=50)

rfeobj =feature_selection.RFE(estimator=model,n_features_to_select=7)

rfeobj.fit(Xtrain,ytrain)

selected =Xtrain.columns[rfeobj.support_]

Xtrain1 = Xtrain[selected]

Xtest1 = Xtest[selected]

model = ensemble.RandomForestClassifier(random_state=42,n_estimators= 70,max_features=5,

min_samples_leaf=40)

model.fit(Xtrain1,ytrain)

predtrain= model.predict(Xtrain1)

predtest=model.predict(Xtest1)

print("TRAIN")

printmetric(ytrain,predtrain)

print("TEST")

printmetric(ytest,predtest)
```

6.8.2 Results

n_estimators= 70

max_features=5

min_samples_leaf=40

n_features_to_select=7

TRAIN	AUC 0.9606	Accuracy 0.9539	Precision 0.7661	Recall 0.97	F1 0.8561
TEST	AUC 0.9752	Accuracy 0.964	Precision 0.8129	Recall 0.9912	F1 0.8933

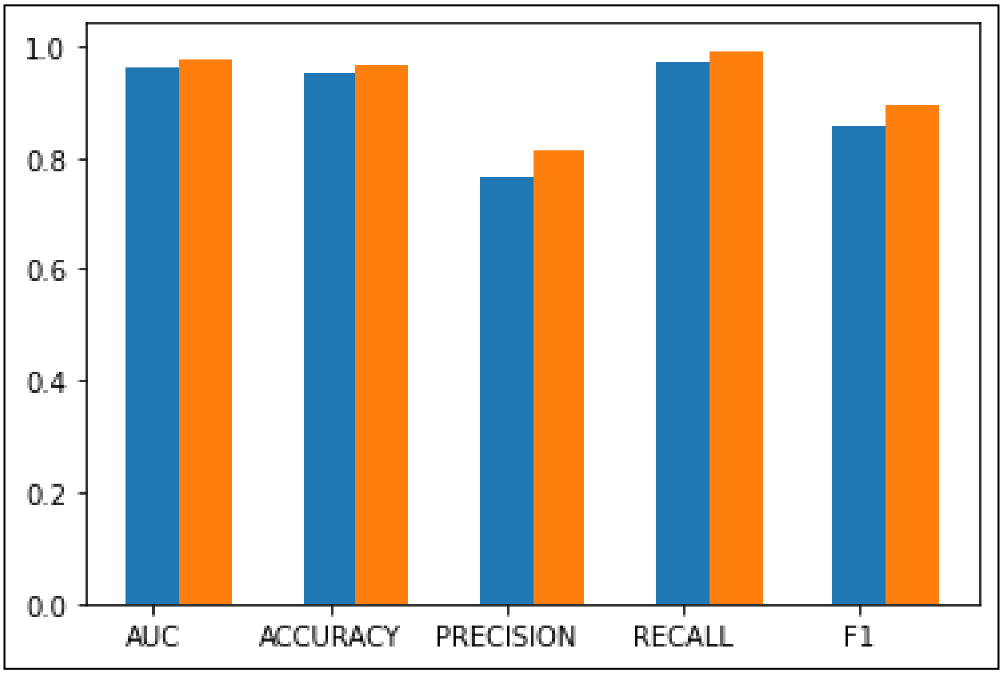

Fig 6.8 Training data and Test data comparison

Analysis: The score results are neither Overfit nor underfit. The precision is slightly lower than the recall value, indicating more false positive cases than false negatives. Overall, the F1 score is balanced. The accuracy is high with a good prediction of results and a higher AUC score that indicted how the model is significant.

6.9 Obsessive-compulsive personality disorder

MODEL: XG BOOST

6.9.1 Model Construction

model = xgb.XGBClassifier(random_state=42,objective='binary:logistic',eval_metric='auc',seed=42,n_estimators=25,

min_child_weight=35,

learning_rate=0.07)

```
rfeobj =feature_selection.RFE(estimator=model,n_features_to_select=5)

rfeobj.fit(Xtrain,ytrain)

selected =Xtrain.columns[rfeobj.support_]

Xtrain1 = Xtrain[selected]

Xtest1 = Xtest[selected]

model = xgb.XGBClassifier(random_state=42,objective='binary:logistic',eval_metric='auc',seed=42,n_estimators=25,

min_child_weight=35,

learning_rate=0.07)

model.fit(Xtrain,ytrain)

predtrain= model.predict(Xtrain)

predtest=model.predict(Xtest)

print("TRAIN")

printmetric(ytrain,predtrain)

print("TEST")

printmetric(ytest,predtest)
```

6.9.2 Results

n_estimators=25

min_child_weight=35

learning_rate=0.07

n_features_to_select=5

TRAIN	AUC 0.9675	Accuracy 0.9944	Precision 1.0	Recall 0.935	F1 0.9664
TEST	AUC 0.9851	Accuracy 0.9973	Precision 1.0	Recall 0.9701	F1 0.9848

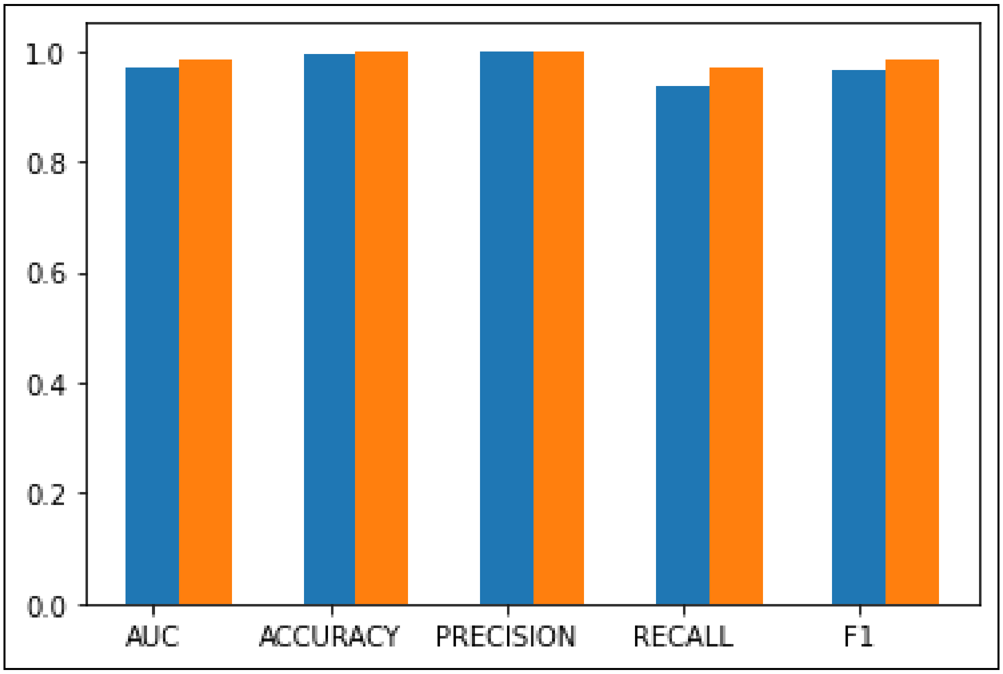

Fig 6.9 Training data and Test data comparison

Analysis: The score results are neither Overfit nor underfit. The recall is slightly lower than the precision value, indicating more false negative cases than false positives. Overall, the F1 score is balanced. The accuracy is high with a good prediction of results and a higher AUC score that indicted how the model is significant.

7 Conclusions

Currently, there are no other public datasets available on this problem, and the work done so far has given some empirical proof, that we may be able to make a Machine Learning model which can detect whether a person has any sort of Personality disorder or not. The cleaned data set seems a good resource to train and test our proposed model as it has no outliers and missing data.

We, while doing this project, are learning about the applications of Machine Learning, in solving real-world problems.We have learned about the need for quality data collection, and data cleaning and we will be learning about the uses of Python libraries in Machine Learning algorithms, creating and training a classification model from scratch.We understood how to measure model performance precision, recall, and f1-score. We also learned about cross-validation concepts.

8 Future Scope

1. Building comprehensive and diverse datasets that include reliable diagnostic information and behavioural dataset related to personality disorders can be used for better accuracy.
2. Identifying better features or variables associated with personality disorders and selecting appropriate psychological assessment measures such as behavioural patterns, linguistic cues, or other relevant data sources can help to get better results.
3. Thoroughly validating the prediction models using independent datasets and diverse populations across different settings and cultural contexts can help ensure this model's reliability and applicability.
4. Taking better measurements to get an authentic response dataset for 'Antisocial personality disorder' which is excluded from this model, can help include the prediction of all the personality disorders.
5. Finally, to make this technology available for medical use a framework needs to be designed.

9 References

[1]Blashfield, R.K., and Reynolds, S.M., An invisible college view of the DSM-5 personality disorder classification. *Journal of Personality Disorders*, *26*(6), p.649, 2012.

[2]Blashfield, R.K., and Reynolds, S.M., An invisible college view of the DSM-5 personality disorder classification. *Journal of Personality Disorders*, *26*(6), p.659, 2012.

[3]Blashfield, R.K., and Reynolds, S.M., An invisible college view of the DSM-5 personality disorder classification. *Journal of Personality Disorders*, *26*(6), p.672, 2012.

[4]Bishop, Christopher M. Pattern Recognition and Machine Learning. New York: Springer, Linear Models for Classification, p.205, 2006.

[5]Trevor Hastie, Robert Tibshirani, Jerome H. Friedman, The Elements of Statistical Learning, Edition 2.0, Springer, p.295, 2009.

[6]Trevor Hastie, Robert Tibshirani, Jerome H. Friedman, The Elements of Statistical Learning, Edition 2.0, Springer, p.337, 2009.

[7]Trevor Hastie, Robert Tibshirani, Jerome H. Friedman, The Elements of Statistical Learning, Edition 2.0, Springer, p.210, 2009.

[8]Trevor Hastie, Robert Tibshirani, Jerome H. Friedman, The Elements of Statistical Learning, Edition 2.0, Springer, p.310, 2009.

[9]Trevor Hastie, Robert Tibshirani, Jerome H. Friedman, The Elements of Statistical Learning, Edition 2.0, Springer, p.605, 2009.

[10] Kevin P. Murphy, Machine Learning: A Probabilistic Perspective, Cambridge, Mass. [u.a.]: MIT Press, p.32, 2013.

[11] Trevor Hastie, Robert Tibshirani, Jerome H. Friedman, The Elements of Statistical Learning, Edition 2.0, Springer, p.346, 2009.