A Data-constrained Clinical Decision Support System for Mental Health: Architecture, Implementation and Evaluation

Juan Francisco Paoli*, Lucía Coll, Parag Chatterjee and María F. Pollo-Cattaneo

Universidad Tecnológica Nacional, Buenos Aires, Argentina

Abstract

Mental health care poses unique challenges for clinical decision-making due to the complexity, heterogeneity, and often subjective nature of mental health disorders. While clinical decision support systems (CDSSs) have shown considerable success in other medical domains, their adoption in mental health remains limited. Existing tools are frequently condition-specific, rely on narrow datasets, and rarely integrate multiple machine learning (ML) algorithms within a single framework—reducing their applicability in diverse real-world scenarios. There is, therefore, a pressing need for an integrative CDSS capable of operating effectively under varying data availability conditions, while making advanced ML techniques accessible to clinicians and patients. In order to meet this need, this work introduces the development of a modular CDSS architecture focused on mental health including exploratory data analysis, model training, and prediction functionalities by presenting a user-friendly graphical user interface (GUI). The system uses well-known ML algorithms like Random-Forest, K-Nearest Neighbors, Gradient Boosting, and Support Vector Machine, and evaluates them across three publicly available datasets. Results show, after validating with multiple performance metrics and cross-validation, that binary classification tasks achieved consistently better metrics than multiclass classification, with Random Forest and Gradient Boosting standing out. Despite these promising results, some limitations persist, especially in dataset quality and the shortage of non-synthetic, traceable mental health data. Future work should include enhancing dataset quality and trying advanced ML models to improve the results. All in all, the findings exhibit the potential of CDSSs combined with ML algorithms to better mental health decision-making for both professionals and patients.

Keywords

Clinical Decision Support System, Mental Health, Machine Learning

1. Introduction

A clinical decision support system (CDSS) consists of software designed to assist healthcare professionals in their decision-making process, by using a clinical knowledge base to align patient-specific characteristics, thus allowing professionals to generate personalized assessment and treatment recommendations. CDSSs are able to give guideline-based interventions, drug interaction alerts, risk assessments, and recommendations at the point of care, with the goal of improving overall clinical effectiveness and reducing preventable errors and costs [1, 2]. It is relevant to highlight that these systems are intended to complement rather than replace clinical professionals, enhancing decision-making with information. In the context of mental healthcare, CDSSs assist professionals such as psychologists, psychiatrists, and therapists in diagnosing conditions and adapting treatment strategies. Due to their ability to notice patterns in complex data at early stages [1], these systems show substantial potential for identifying possible concerns and allowing the discoveryof more about a patient's health.

Even though CDSSs have demonstrated their numerous benefits in various healthcare settings [3], their use in the field of mental health remains underexplored. Because of the diversity of mental health disorders, existing CDSSs in this field mostly focus on specific conditions, which limits their applicability. There is, therefore, a demand for a generic, integrative tool capable of incorporating multiple machine

ICAIW 2025: Workshops at the 8th International Conference on Applied Informatics 2025, October 8–11, 2025, Ben Guerir, Morocco

*Corresponding author.

jpaoli@frba.utn.edu.ar (J. F. Paoli); lcoll@frba.utn.edu.ar (L. Coll); parag@frba.utn.edu.ar (P. Chatterjee); flo.pollo@gmail.com (M. F. Pollo-Cattaneo)

0009-0005-5077-4332 (J. F. Paoli); 0009-0007-3999-9826 (L. Coll); 0000-0001-6760-4704 (P. Chatterjee); 0000-0003-4197-3880 (M. F. Pollo-Cattaneo)

learning (ML) algorithms to address diverse clinical scenarios and make sophisticated models usable in routine practice.

A key barrier in this field is the limited availability of high-quality data. Small sample sizes—both in training and testing—are frequently reported, which hampers model generalizability and robustness. As highlighted in prior work, ML-based CDSSs in mental health depend heavily on comprehensive, high-quality datasets [4, 5], making data scarcity a critical challenge.

Typically, a CDSS consists of three primary components:

- Knowledge base — a repository containing structured clinical knowledge, datasets, or domain-specific rules.
- Decision-support engine — the inference mechanism that applies stored knowledge to patient-specific data to generate predictions or recommendations.
- Graphical user interface (GUI) — the interactive layer that enables clinicians and, in some cases, patients to engage with the system.

Despite the clear definition of these components, the development of ML-driven CDSSs for mental health has lagged, primarily due to the aforementioned data constraints. The present work addresses this gap by developing a general-purpose, ML-based CDSS for mental health that adheres to this three-component architecture and is explicitly designed to function under limited-data conditions. The system incorporates two GUIs: one for healthcare professionals, supporting data exploration, feature relationship analysis, and model training; and a second interface for both professionals and patients, enabling predictions from pre-trained models—thereby separating training and prediction workflows.

By following best practices in ML model development and evaluation, this study aims to demonstrate that, even in data-constrained environments, meaningful insights and functional decision-support tools for mental health can be realized.

2. Methods

To achieve the study's objectives, the methodology was structured into four main components: system architecture, machine learning models, evaluation procedures, and datasets.

First, the system architecture was designed as a modular framework comprising three cooperating layers:

- Exploration layer — enables clinicians to perform interactive exploratory data analysis (EDA) and visualize key patterns in the data.
- Training module — cleans and analyzes data, builds machine learning models, and stores the resulting artifacts.
- Prediction module — loads stored artifacts to provide real-time predictions.

The hand-off between the Training and Prediction modules occurs via a shared file system, ensuring a straightforward, reproducible, and maintainable interface.

Second, the machine learning models implemented in the system utilize well-established algorithms widely adopted in clinical research, including Random Forest, K-Nearest Neighbors (KNN), Gradient Boosting, and Support Vector Machine (SVM) for classification tasks [6], as well as Linear Regression for regression tasks. All models are evaluated within a uniform pipeline employing 5-fold cross-validation [7], allowing direct comparison under identical conditions. The best-performing model, along with all necessary information to reproduce it, is persisted for future inference.

Third, the evaluation framework addresses both predictive quality and platform-level performance. Predictive quality is assessed using standard classification and regression metrics to ensure comparability across problems and models. Platform performance is monitored by recording execution times, usage counts, and other operational metrics to detect performance drifts, assess scalability, and optimize resource utilization. Additionally, simple dataset descriptors (e.g., record count, feature count) are tracked to identify structural changes early.

Finally, the datasets used to validate the CDSS were drawn from three publicly available sources [8, 9, 10], chosen to represent diversity in dataset size, target type, and feature composition. These datasets range from slightly over 100 records to several thousand and are fully traceable via Digital Object Identifiers (DOIs). This diversity enables the evaluation of both the modeling pipeline and system infrastructure under varied conditions without introducing unnecessary complexity.

The following subsections expand on each methodological component in greater detail.

2.1. Architecture

The proposed CDSS is a unified platform operationally divided into three modules with distinct roles:

- Exploration module — facilitates in-depth understanding of the dataset through interactive EDA and visualization tools.
- Training module — trains and saves machine learning models, producing artifacts such as trained weights, preprocessing pipelines, and metadata.
- Prediction module — generates predictions for a selected feature based on user-provided input data, relying entirely on the artifacts produced by the Training module.

The Prediction module cannot function without the trained models and their associated metadata. Communication between the Training and Prediction modules occurs exclusively through a shared file system, which serves as the contract for exchanging models, variables, and configuration files. By contrast, the Exploration and Training modules are integrated into the same web application, enabling direct communication via internal function calls.

This separation of components, illustrated in Figure 1, ensures modularity, simplifies maintenance, and enables independent scaling of prediction services without affecting the exploration or training functionalities.

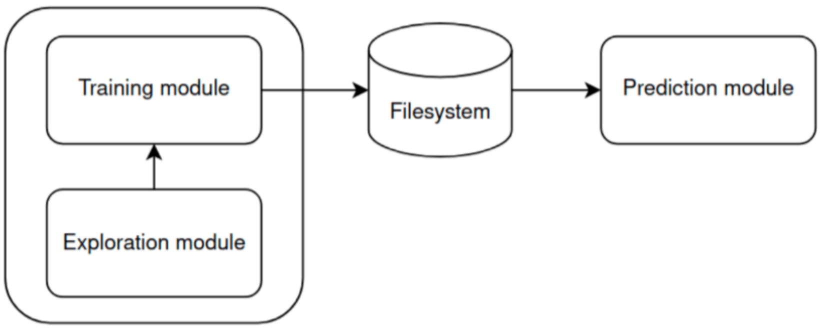

Figure 1: Overall system's architecture

2.1.1. Exploration module

The Exploration Module enables users to gain an in-depth understanding of the dataset by providing insights into feature distributions and interrelationships through interactive data exploration techniques.

Data Loading and Preprocessing: The module accepts data in CSV or XLS formats. The preprocessing steps involve removing timestamp and identification columns, and rows with missing values, while also converting to lowercase all characters and eliminating unwanted symbols, thus normalizing text. To ensure consistency in subsequent analyses these steps are repeated every time the dataset is modified.

Exploration and Visualization: Utilizing data science libraries, the module generates an exploratory data analysis (EDA) report that includes descriptive statistics, variable distributions, and the relationship between two features. Users may select any two features for visualization, with the variable types determining the plot type: Bar plot if both features are categorical and Box plot if at least one feature is numerical. This module aims to help users with the task of identifying data patterns and potential predictive relationships, prior to model development.

2.1.2. Training module

The Training module builds, evaluates, and saves predictive models based on the preprocessed dataset. It is organized into two main phases:

Machine Learning Model Training: First, a target variable is selected for prediction, after which the dataset is divided into two subsets, one for training and the other one for testing. Depending on the target type (categorical or numerical), suitable machine learning algorithms are applied. To internally validate candidate models, a 5-fold cross-validation loop is embedded within the training pipeline. Performance is evaluated using multiple metrics, including F1-score, accuracy, and confusion matrix analysis, allowing the analysis of the models and identifying the most suitable for the task.

Model Saving: Based on the evaluation results, users may select one or more models to be stored into the shared file system. Along with the serialized model, a JSON file is created containing metadata such as variable names, performance metrics, and training duration. This ensures replicability and grants seamless integration with the Prediction Module.

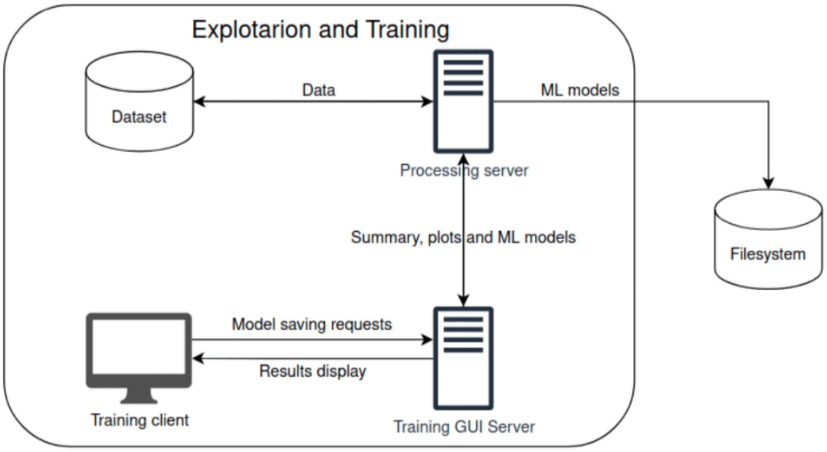

Figure 2: Architecture of the Exploration and Training modules

2.1.3. Prediction module

The Prediction module generates outcomes built on previously trained and saved models. It consists of four key steps:

- Loading Previously Trained Models — The module recovers serialized models and related metadata from the shared file system and loads the variables that are required for prediction.
- User Data Input — Users provide values for the relevant input variables. Depending on the data type, categorical fields are presented as dropdown menus with predefined options, while numerical fields accept direct numeric input.
- Prediction — Using the user-provided input, the loaded model generates a real-time prediction.
- Results Display — Prediction results are presented in a clear, interpretable, and user-friendly format, enabling immediate use in clinical decision-making.

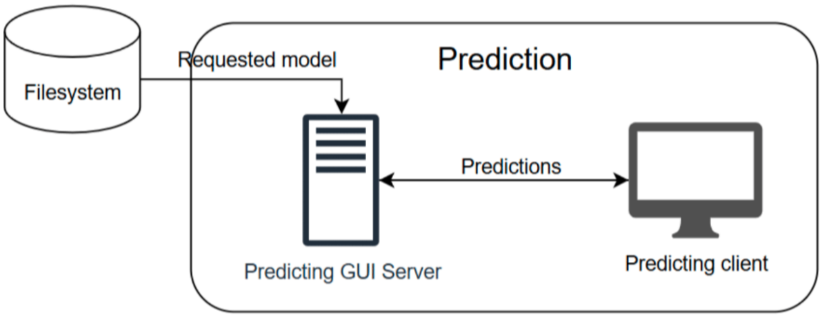

Figure 3: Architecture of the Prediction module.

2.2. Machine Learning Models

To support the functionality of the Prediction Module, the CDSS incorporates widely recognized machine learning algorithms commonly employed in healthcare classification tasks [6]:

- K-Nearest Neighbors (KNN) — valued for its simplicity and effectiveness, KNN predicts outcomes by calculating the distance between the query point and training samples, assigning the label most common among the K nearest neighbors.
- Random Forest — a supervised ensemble method composed of multiple decision trees, which reduces overfitting and generally outperforms single decision tree models.
- Gradient Boosting — applicable to both classification and regression tasks, this ensemble method iteratively improves decision trees by applying a gradient descent approach to minimize prediction errors.
- Support Vector Machine (SVM) — suitable for both classification and regression, SVM identifies the optimal hyperplane that maximizes separation between classes in the feature space.

For regression tasks, several regression models were also implemented. However, due to the lack of publicly available mental health datasets with reliable numerical target regression variables, it was not externally tested.

2.3. Metrics

The evaluation framework employed a diverse set of metrics, organized into three main categories: model quality metrics, system performance metrics, and dataset descriptors.

Model Quality Metrics: The selection of metrics depended on the prediction task [8]. For regression models, we computed the Mean Absolute Error (MAE), Mean Squared Error (MSE), coefficient of determination (R^2), adjusted R^2, and Mean Absolute Percentage Error (MAPE) [9]. Binary classification models were evaluated using the Receiver Operating Characteristic (ROC) curve, Area Under the ROC Curve (ROC-AUC), Brier Score, F1-score, and confusion matrix. For multiclass classification, accuracy, F1-score, and confusion matrix were considered. To ensure robustness and comparability across candidates, all models were validated using 5-fold cross-validation.

System Performance Metrics: Operational performance was measured through metrics such as latency and execution times, more precisely, the average and most recent times for graph generation, dataset summary execution, and model training. Tracking these metrics allows the early detection of performance degradation and enables early resolution of technical issues.

Dataset Descriptors: In order to maintain visibility over dataset structure and complexity, basic indicators such as the total number of columns and rows were tracked. While simple, these measures help in identifying unexpected structural changes, including data loss or the unforeseen appearance of new features.

2.4. Datasets

The datasets were chosen to be representative of different demographic groups, variable types, and psychological or behavioral scenarios. This was done to ensure that the clinical decision support system (CDSS) addressed a broad spectrum of mental health–related prediction scenarios. This diversity was essential to evaluate the generalizability of the system across heterogeneous data sources, a crucial property for a CDSS intended for real-world clinical environments. Additionally, using datasets that portrayed distinct mental health indicators—ranging from depression symptoms to behavioral adaptations during lockdown—allowed the system to be tested on tasks reflecting both acute and chronic conditions, therefore aligning with the primary objective of improving mental health monitoring and support.

To design, train, and validate the CDSS, we drew on three publicly available datasets, as detailed in Table 1. Their selection was driven by three complementary criteria:

1. Sample size range — spanning from 105 to 3,487 records, enabling the evaluation of the system's computational scalability under varying data volumes.
2. Diversity of predictive variables — incorporating categorical, binary, and numerical features to enhance the generalizability of the predictive models and facilitate comparative analysis across different data types.
3. Traceability and reproducibility — ensured by the presence of Digital Object Identifiers (DOIs), confirming that the datasets are authentic, non-synthetic, and citable in scientific research.

Each dataset represents a distinct domain within mental and behavioral health, offering complementary perspectives:

- Raw Dataset.csv — predominantly categorical variables derived from student self-reports of psychological and behavioral symptoms. The prediction task targets a categorical variable measuring how frequently students experienced feelings of being "down, depressed, or hopeless" during a semester.
- sahar.xlsx — a mix of binary and numerical variables describing behavioral, psychological, and physiological states. The target variable is binary ("irritable"), indicating the presence or absence of irritability.
- Resilience_CleanOnly_v1.csv — categorical and numerical variables assessing behavioral and lifestyle changes during lockdown periods. The selected target variable, "r6_change_fct3," categorically classifies changes in alcohol consumption during that period.

Table 1
Summary of the public datasets used to train and evaluate the CDSS, including dimensionality (rows × columns), prediction task type, target variable, and identifier (DOI).

Dataset file	Columns x Rows	Category	Selected Target Variable
[10] Raw Dataset.csv	33x2028	Categorical	In a semester, how often have you been feeling down, depressed, or hopeless
[11] sahar.xlsx	20x105	Binary	irritable
[12] Resilience_CleanOnly_v1.csv	38x3487	Categorical	r6_change_fct3 (change in alcohol intake during lockdown)

3. Results

In this section, we present the outcomes from the implementation and evaluation of the clinical decision support system (CDSS). First, we elaborate on the operational details of the Training and Prediction modules, highlighting their functionalities and interaction through the filesystem using Pickle for model serialization. Finally, we summarize the performance metrics obtained for each dataset, facilitating a comparative evaluation of the machine learning models applied.

3.1. Modules

The Training and Prediction modules of the CDSS were developed in Python 3.11, a widely adopted language for machine learning applications [13]. Both modules, to simplify ML models exchange between them, implement Pickle, a library that serializes Python objects to the filesystem and then reloads them. Additionally, the Training module generates and stores a JSON file containing detailed metadata, including the name, type, and potential values for each feature used in predictive models. This metadata grants accurate data entry and consistent feature encoding during predictions.

3.1.1. Exploration module

The Exploration module makes use of multiple Python libraries: NumPy and pandas for data retrieval and preprocessing, YDataProfiling, matplotlib, and Seaborn for exploratory data analysis (EDA), Flask for web application development, and scikit-learn for machine learning model training.

After loading a dataset, the system provides a complete summary of the data, with reports such as feature names, data types, value distributions, and correlations between all feature pairs. Users are also able to select any two features and generate a plot that compares them using the "Compare Features" functionality, allowing them to visualize the relationship as either a bar plot (if both features are categorical) or a box plot (if at least one feature is numerical). This interactive analysis allows users to better understand the dataset structure and potential predictive relationships prior to model training.

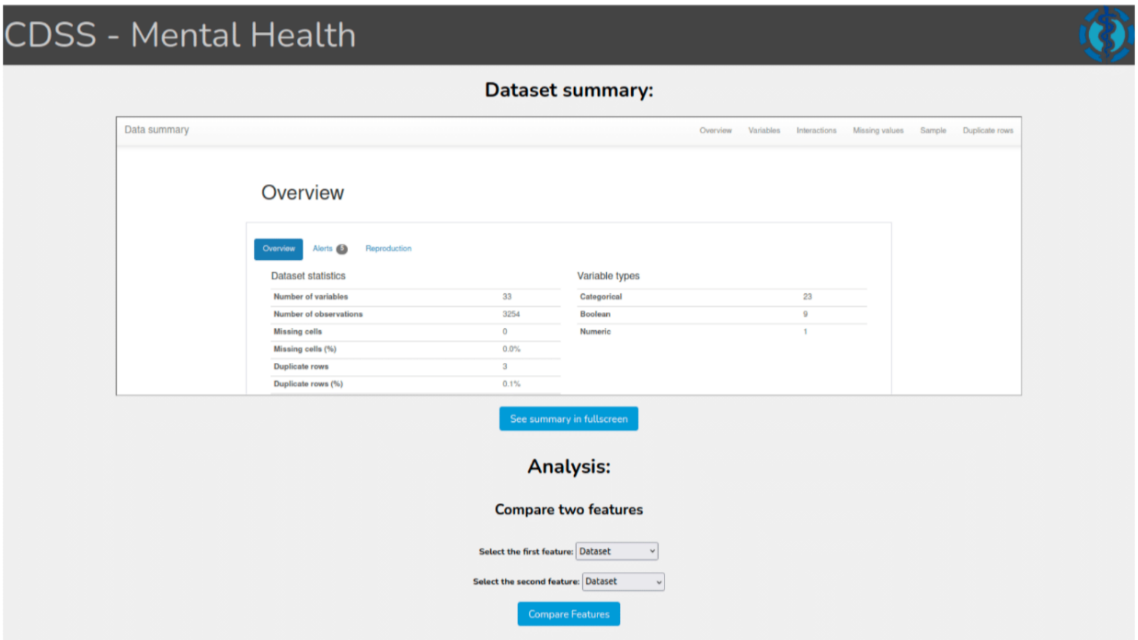

Figure 4: The exploratory data analysis graphical user interface.

3.1.2. Training module

In addition to the libraries used by the exploration component, the training component also utilized scikit-learn for the training and use of machine learning models. Following the exploratory analysis, the training component enables the user to select one of the variables available in order to predict it using machine learning algorithms.

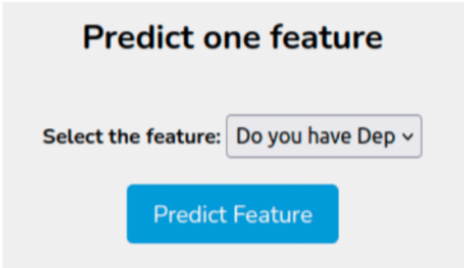

Figure 5: Selection of the predicted feature.

After selecting the feature and pressing the "predict feature" button, all models are trained (depending on the type of the feature and the number of possible values) and the user is taken to a different webpage that showcases the metrics for the different models, in which the user can press "save model" to save the model in the filesystem for the prediction component to use it.

Trained machine Learning model: GradientBoosting.
Training time: 36.187 segundos.
Accuracy for selected feature: 69.18%

Confusion Matrix (Total Samples: 305)

	Predicted: 2 - more than half the days	Predicted: 0 - not at all	Predicted: 1 - several days	Predicted: 3 - nearly every day
Actual: 2 - more than half the days	21	12	0	0
Actual: 0 - not at all	5	71	14	5
Actual: 1 - several days	0	19	34	19
Actual: 3 - nearly every day	0	4	16	85

Save model

Trained machine Learning model: RandomForest.
Training time: 3.25 segundos.
Accuracy for selected feature: 68.52%

Confusion Matrix (Total Samples: 305)

	Predicted: 2 - more than half the days	Predicted: 0 - not at all	Predicted: 1 - several days	Predicted: 3 - nearly every day
Actual: 2 - more than half the days	20	13	0	0
Actual: 0 - not at all	4	73	7	11
Actual: 1 - several days	0	25	28	19
Actual: 3 - nearly every day	0	6	11	88

Save model

Figure 6: Comparison of the models.

3.1.3. Prediction module

On the other hand, the prediction component employs the pandas library for encoding categorical variables and Gradio to develop the user-friendly web interface.

The user interface for this component is simpler than the previous one, since it's aimed to also be used by non-technical users. It is formed by a single form, in which the user can:

1. Select one of the previously saved models.
2. Complete the form with all the information required by the original dataset, except for the predicted feature.
3. Press "predict" to use the model to predict the feature with the given values.

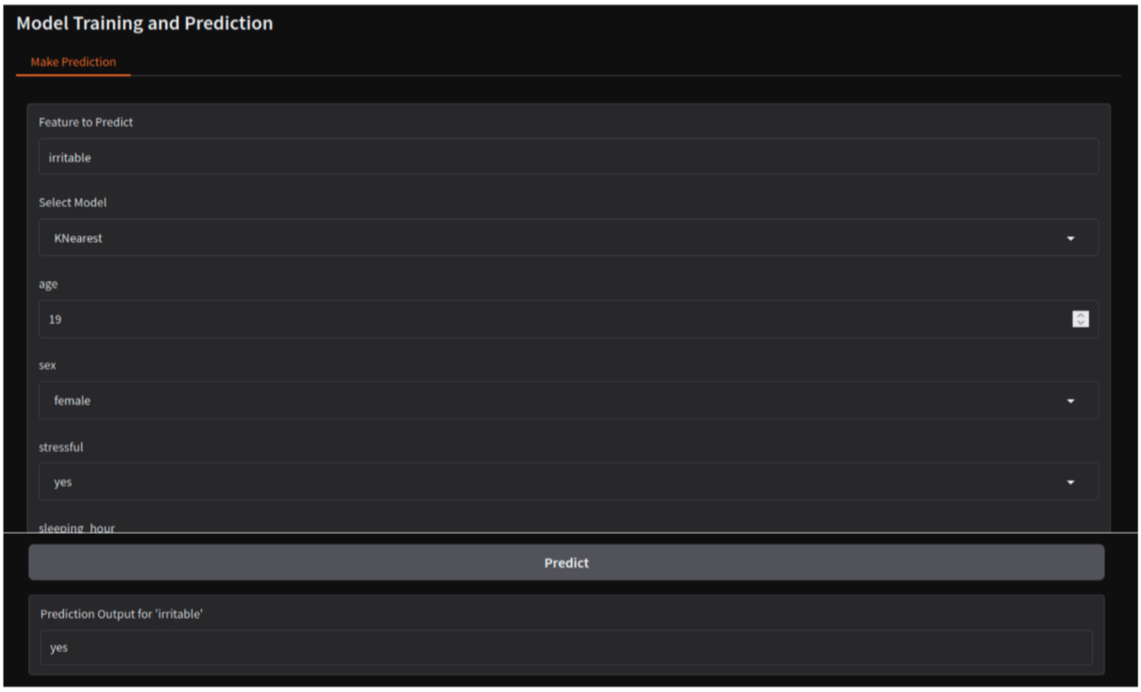

Figure 7: The prediction component's GUI

3.2. Models and their metrics for each dataset

All datasets underwent five training cycles for each model, with five-fold cross-validation executed to ensure robust evaluation. This scheme was selected because it offers a balanced bias–variance trade-off for our sample sizes: every observation is used for testing and for training, yielding stable performance estimates without the excessive computational cost of higher-k folds and avoiding the high variance of a single hold-out split. In the following sections, we describe each dataset individually, highlighting the classification problems addressed and the metrics obtained for each model.

3.2.1. Raw Dataset.csv

The Raw Dataset.csv is designed to address a multiclass classification problem. Post-preprocessing, no rows or columns were excluded. The ROC-AUC metric is not applicable to this dataset since it contains more than two classes, requiring alternative metrics such as accuracy, F1-score, and confusion matrices for effective evaluation. Table 2 presents the metrics obtained for this dataset.

Table 2
Summary of the metrics derived from dataset “Raw Dataset.csv”

Metric		KNearest	Random Forest	Gradient Boosting	SVM
Mean training time		0.82519 s	3.794 s	28.1938 s	2.3628 s
Last training time		2.344 s	3.235 s	24.781 s	2.797 s
F1 Score		0.64184	0.67417	0.68858	0.66321
Accuracy		0.64918	0.68535	0.6918	0.66557
Confusion Matrix (Total sample: 305; Classes: More than half the days, not at all, several days, nearly every day		[[23 10 0 0] [12 67 7 9] [3 27 27 15] [6 6 12 81]]	[[20 13 0 0] [4 73 7 11] [0 25 28 19] [0 6 11 88]]	[[21 12 0 0] [5 71 14 5] [0 19 34 19] [0 4 16 85]]	[[20 13 0 0] [3 69 16 7] [0 19 32 21] [0 5 18 82]]
Cross validation	Accuracy mean	0.60301	0.64715	0.65934	0.66396
	F1 Score mean	0.59866	0.63966	0.6548	0.65944

3.2.2. sahar.xlsx

The sahar.xlsx dataset focuses on a binary classification task within a relatively small dataset. Post-preprocessing, this dataset comprises 20 columns and 104 rows, with only one row removed. Table 3 presents the metrics obtained for this dataset.

Table 3
Summary of the metrics derived from dataset “sahar.xlsx”

Metric		KNearest	Random Forest	Gradient Boosting	SVM
Mean training time		0.1628 s	1.4688 s	0.631 s	0.1374 s
Last training time		0.171 s	1.469 s	0.672 s	0.14 s
F1 Score		0.93	0.93	0.88	0.78
Accuracy		0.93333	0.93333	0.875	0.77778
Confusion Matrix(Total sample: 11; Classes: Yes, No)		[[3 1] [0 7]]	[[3 1] [0 7]]	[[2 2] [0 7]]	[[0 4] [0 7]]
ROC-AUC		0.875	0.875	0.75	0.5
Brier Score		0.07071	0.11485	0.18478	-
Cross validation	Accuracy mean	0.76433	0.78538	0.79532	0.76374
	F1 Score mean	0.8466	0.86478	0.86992	0.86586
	Brier Score mean	-0.18207	-0.14802	-0.13268	-

3.2.3. Resilience_CleanOnly_v1.csv

The Resilience_CleanOnly_v1.csv dataset addresses a multiclass classification issue centered around resilience outcomes. Following preprocessing, the dataset consists of 33 columns and 3254 rows, with 5 columns and 233 rows removed. As with the Raw Dataset.csv, the ROC-AUC metric is not applicable here due to the multiclass classification nature. Table 4 presents the metrics obtained for this dataset.

Table 4
Summary of the metrics derived from dataset “Resilience_CleanOnly_v1.csv”

Metric		KNearest	Random Forest	Gradient Boosting	SVM
Mean training time		0.7622 s	5.3344 s	17.3216 s	7.2376 s
Last training time		0.687 s	5.234 s	17.687 s	9.578 s
F1 Score		0.46354	0.6132	0.5902	0.4366
Accuracy		0.46421	0.66258	0.62372	0.58896
Confusion Matrix (Total sample: 489; Classes: increase, no change, decrease)		[[14 13 50] [34 27 63] [60 42 186]]	[[13 5 59] [2 43 79] [2 18 268]]	[[15 9 53] [6 45 73] [13 30 245]]	[[0 0 77] [0 0 124] [0 0 288]]
Cross validation	Accuracy mean	0.47776	0.64629	0.60289	0.56636
	F1 Score mean	0.46992	0.5856	0.56792	0.40983

4. Conclusions

The evaluation of the proposed clinical decision support system (CDSS) yielded several important insights. Prediction quality varied remarkably across all datasets. Raw Dataset.csv exhibited moderate accuracy and F1-scores, with Gradient Boosting slightly outperforming other models. On the other hand, sahar.xlsx achieved high accuracy and F1-scores but displayed variability in ROC–AUC, suggesting a generally good fit with some instability, probably due to its small sample size. In contrast, Resilience_CleanOnly_v1.csv showed the lowest accuracies and F1-scores across all models, despite its comparable size and fewer classes than Raw Dataset.csv. This likely indicates that data noise and complexity are critical factors when determining model performance, rather than just dataset size and class count.

These comparisons across the datasets also revealed that binary classification tasks (sahar.xlsx) generally yield higher performance metrics than multiclass tasks (Raw Dataset.csv and Resilience_CleanOnly_v1.csv). Also, class distribution and dataset complexity significantly affect predictive effectiveness, as imbalanced or heterogeneous features often reduce model performance. Across all tasks, both Random Forest and Gradient Boosting consistently outperformed simpler models such as K-Nearest Neighbors and SVM, particularly for more challenging datasets. These results highlight the importance of selecting and tuning models based on dataset characteristics and classification complexity.

Several limitations were identified, primarily in regard to dataset availability and quality. One of the most important constraints was the scarcity of reliable, traceable, non-synthetic datasets in mental health, which excluded training regression models and limited our evaluation to classification tasks. Future work should focus on obtaining or generating high-quality, traceable datasets—including those suitable for regression—to improve CDSS robustness. Further research should also explore a broader range of machine learning approaches, including advanced methods such as artificial neural networks and unsupervised learning techniques. Evaluating these alternative models could provide deeper insights into their relative strengths and limitations, facilitating the selection of the most effective model for specific mental health applications.

Overall, this study highlights the potential of machine learning in developing CDSSs for mental health. Despite challenges related to data availability and complexity, the results demonstrate that carefully designed systems can provide valuable predictive insights to support both clinical professionals and patients.

Acknowledgements

The authors acknowledge Cloudgenia for their technical assistance in this work.

Declaration on Generative AI

The authors have not employed any Generative AI tools.

References

[1] L.-A. Bertie, E. A. McDermott, J. C. Quiroz, J. L. Hudson, A scoping review of clinical decision support systems for child and adolescent mental health, Current Psychology 44 (2025) 2785–2804.
[2] F. Tong, R. Lederman, S. D'Alfonso, Clinical decision support systems in mental health: A scoping review of health professionals' experiences, International Journal of Medical Informatics (2025) 105881.
[3] D. Gnjidic, S. N. Hilmer, F. M. Blyth, V. Naganathan, L. Waite, M. J. Seibel, A. J. McLachlan, R. G. Cumming, D. J. Handelsman, D. G. Le Couteur, Polypharmacy cutoff and outcomes: five or more medicines were used to identify community-dwelling older men at risk of different adverse outcomes, Journal of clinical epidemiology 65 (2012) 989–995.

[4] D. Wu, J. An, S. Nan, Y. She, H. Duan, N. Deng, A knowledge-based clinical decision support system for personalized health examination items in china: design and evaluation, BMC Medical Informatics and Decision Making 25 (2025) 183.
[5] F. Velosa, H. Florez, Edge solution with machine learning and open data to interpret signs for people with visual disability., in: ICAI workshops, 2020, pp. 15–26.
[6] W. Alsabhan, A. Alfadhly, Effectiveness of machine learning models in diagnosis of heart disease: a comparative study, Scientific Reports 15 (2025) 24568.
[7] Coursera Staff, What is cross-validation in machine learning?, 2025.
[8] C. Miller, T. Portlock, D. M. Nyaga, J. M. O'Sullivan, A review of model evaluation metrics for machine learning in genetics and genomics, Frontiers in Bioinformatics 4 (2024) 1457619.
[9] A. De Myttenaere, B. Golden, B. Le Grand, F. Rossi, Mean absolute percentage error for regression models, Neurocomputing 192 (2016) 38–48.
[10] M. Syeed, A. Rahman, L. Akter, K. Fatema, R. Khan, M. Karim, M. Hossain, M. Uddin, Mhp (anxiety, stress, depression) dataset of university students, 2024.
[11] I. Faeck Jaafar, S. Ahmed Mehdi, A.-H. M. Fawzi, Isolation insult during covid-19 pandemic on the psychological status of medical students, Journal of Clinical Laboratory Analysis 36 (2022) e24702.
[12] B. Beaglehole, J. Williman, C. Bell, J. Stanley, M. Jenkins, P. Gendall, J. Hoek, C. Rapsey, S. Every-Palmer, Thriving in a pandemic: Determinants of excellent wellbeing among new zealanders during the 2020 covid-19 lockdown; a cross-sectional survey, PloS one 17 (2022) e0262745.
[13] S. Raschka, J. Patterson, C. Nolet, Machine learning in python: Main developments and technology trends in data science, machine learning, and artificial intelligence, Information 11 (2020) 193.