A Trustworthy Explainable Transformer-Based Ensemble Learning Framework For Mental Health Treatment Prediction

Anum Sattar[1], Muhammad Zubair Asghar[1], Hafiz Waheedud din[1], Syed Muhammad Ali Shah[1], Abdul Qahar[2]

[1]Gomal Research Institute of Computing (GRIC), Faculty of Computing Gomal University D.I.Khan, K.P.K, Pakistan
[2]Information Technology Centre, University of the Punjab Lahore, Pakistan
anumsattar786@gmail.com
mzubairgu@gmail.com
waheedqazi@gu.edu.com
alishah@gu.edu.pk
abdulqaharpk@gmail.com

Abstract
Although the worldwide burden of mental illness is increasing, a significant percentage of people are not seeking professional help; factors differentiating those who seek help from those who do are not easily modelled using classical statistical tools. In this study, a transformer-based model for predicting mental health treatment-seeking behavior from workplace and demographic survey data, using an explainable approach, is presented. We propose a modified version of TabTransformer, a transformer-based architecture, incorporating a dedicated classification (CLS) token, learned column-identity embeddings, class-weighted loss, and label smoothing for modeling the contextual dependency among the categorical items in the survey with a large-scale dataset of 292,364 responses over fifteen categorical features (demographic, occupational, and attitudinal). The model is compared with five classical and ensemble baselines (Logistic Regression, Random Forest, XGBoost, LightGBM, CatBoost) with the same train/validation/test protocol, and the individual architectural modification of the model is determined by a systematic ablation study. Tab Transformer achieves the highest overall accuracy (0.785), macro-F1 (0.784), and macro-AUC (0.877) of all the models evaluated, just slightly better than CatBoost and other gradient-boosting models, but significantly better than Logistic Regression. We use SHAP and LIME to provide global, per-class, and instance-level explanations of the model to highlight the survey items that most strongly determine treatment-seeking results, particularly in cases where deep tabular models are opaque. The findings show that the TabTransformer, a transformer-based model able to perform as well as thebaseline study, also provides a built-in way to interpretability, making it suitable as a starting point for building explainable decision support tools in mental health screening scenarios.

1. Introduction

Mental health conditions are among the leading contributors to global disease burden, yet a persistent and well-documented treatment gap separates the prevalence of these conditions from the proportion of affected individuals who actually access care. Stigma, occupational culture, limited awareness of available resources, and uncertainty about the severity of one's own symptoms all contribute to this gap, particularly in high-pressure professional environments such as the technology sector, where mental health disclosure carries perceived career risk. Understanding which individual, occupational, and attitudinal factors are associated with the decision to seek treatment is therefore of practical value: it can inform targeted workplace interventions, guide the design of screening tools, and help organizations allocate mental health

resources more effectively. Predicting treatment-seeking behavior from survey data is fundamentally a tabular classification problem, but one with characteristics that make it harder than it first appears. Most of the available predictors are categorical rather than continuous — gender, occupation, country, family history, self-reported mood patterns, and similar attitudinal items — and the relationships between them are frequently non-additive: the effect of a given risk factor often depends on which other factors co-occur with it. Classical linear models such as logistic regression struggle to represent these interactions without extensive manual feature engineering, while even strong tree-based ensembles capture interactions only implicitly, through the structure of their splits. This motivates architectures that can learn interactions between categorical variables directly and represent them as reusable, inspectable embedding's. The self-attention mechanism that underlies the transformer architecture is a natural candidate for this task, because it allows every categorical feature to be contextualized against every other feature in a given record before a prediction is made. TabTransformer, the architecture on which this study builds, was proposed specifically to bring this capability to tabular data by passing categorical embeddings through stacked transformer layers to produce contextual embeddings, which the original authors showed to be competitive with tree-based ensembles while also being more robust to noisy and missing features. This study extends that architecture with a set of targeted refinements — CLS-token pooling in place of flattened concatenation, explicit column-identity embeddings, class-weighted loss, and label smoothing — and evaluates whether these refinements yield measurable gains on a real-world mental health treatment-seeking dataset. A second concern that motivates this work is interpretability. Deep tabular models, including transformer-based ones, are frequently criticized for being opaque compared with simpler, more auditable baselines, which is a serious limitation in a healthcare-adjacent setting where a model's recommendations may need to be justified to clinicians, employers, or the individuals being assessed. Model-agnostic explanation methods SHAP, grounded in cooperative game-theoretic Shapley values, and LIME, which fits local surrogate models around individual predictions — offer a way to recover global and instance-level explanations from an otherwise black-box model without sacrificing predictive performance. Against this background, the present study makes four contributions. First, it presents a refined Tab Transformer architecture tailored to categorical mental health survey data, incorporating CLS pooling, column-identity embeddings, class-weighted loss, and label smoothing. Second, it benchmarks this architecture against five widely used classical and ensemble baselines — Logistic Regression, Random Forest, XGBoost, LightGBM, and CatBoost — under a controlled, identical evaluation protocol on a dataset of 292,364 survey responses. Third, it isolates the contribution of each architectural refinement through a systematic ablation study, in which every configuration is trained with the same procedure so that differences in outcome can be attributed to the single changed component. Fourth, it applies SHAP and LIME to produce global, per-class, and case-level explanations of the model's predictions, addressing the interpretability requirements of mental health screening applications.

The remainder of this paper is organized as follows. Section 2 reviews related work on machine learning approaches to mental health treatment prediction, tree-based ensemble methods, transformer architectures for tabular data, and explainable AI techniques. Section 3 describes the dataset, preprocessing pipeline, and proposed architecture. Section 4 presents the experimental setup, baseline comparison, ablation results, and explainability analysis. Section 5 discusses the implications and limitations of the findings, and Section 6 concludes with directions for future work.

2. Literature Review

2.1 Machine Learning Approaches to Mental Health Treatment Prediction

A growing body of work has applied supervised machine learning to survey-based mental health data, most commonly to predict whether a respondent has sought or would seek professional treatment. Studies using the Open Sourcing Mental Illness (OSMI) technology-industry survey have compared classical classifiers such as logistic regression, support vector machines, and naive Bayes against ensemble methods; in one recent comparative study on roughly 1,400 technology-sector respondents, gradient boosting (XGBoost) achieved the highest accuracy among five supervised algorithms, with prior diagnosis and family history

of mental illness emerging as the strongest predictors and a marked gender disparity observed in treatment-seeking rates. Related work on annual student health surveys in Japan compared logistic regression, elastic net, random forest, XGBoost, and LightGBM for predicting mental health problems from demographic and survey-response data, ultimately favoring LightGBM for its lower misclassification rate relative to linear baselines. Broader comparative studies extend this line of work to psychiatric misdiagnosis reduction and depression prediction in specific populations, generally finding that boosting and bagging ensembles outperform purely linear models on structured mental health data, though the best-performing algorithm varies by dataset and population. Collectively, this literature establishes tree-based ensembles as a strong and consistent baseline for tabular mental health prediction tasks, motivating their inclusion as comparison points in the present study, but it has not yet examined whether transformer-based architectures can match or exceed these ensembles on treatment-seeking prediction specifically.

2.2 Gradient-Boosted and Ensemble Methods for Structured Health Data

Gradient-boosted decision tree frameworks — XGBoost, LightGBM, and CatBoost — have become the de facto standard for tabular prediction problems across domains, including healthcare, owing to their ability to model non-linear feature interactions, handle mixed categorical and numerical inputs, and train efficiently even on large datasets. XGBoost's regularized, second-order boosting objective and LightGBM's histogram-based, leaf-wise growth strategy both provide strong predictive accuracy at comparatively low computational cost, while CatBoost's native handling of categorical variables through ordered target statistics removes the need for manual one-hot or label encoding and often gives it an advantage on datasets, like mental health surveys, that are almost entirely categorical. Comparative studies applying these three algorithms alongside random forests and logistic regression to psychiatric and general health prediction tasks consistently find that the boosting variants outperform simpler baselines, though the margin between XGBoost, LightGBM, and CatBoost themselves tends to be narrow and dataset-dependent. This narrow, fluctuating gap between top-performing ensembles is part of the motivation for this study's inclusion of all three boosting frameworks as baselines rather than a single representative model.

2.3 Transformer Architectures for Tabular Data

While transformer architectures originated in natural language processing, growing interest has focused on adapting self-attention to structured tabular data, where gradient-boosted trees have historically dominated. TabTransformer was among the first architectures to make this adaptation explicit: it embeds categorical features and passes them through stacked self-attention layers to produce contextual embeddings, which are then combined with any continuous features before a final prediction layer. Across fifteen public benchmark datasets, the original Tab Transformer study reported accuracy competitive with tree-based ensembles, together with greater robustness to missing and noisy features and improved interpretability relative to prior deep tabular baselines, attributable to the contextual embeddings clustering semantically related categorical values. Subsequent architectures have refined this idea in different directions: FT-Transformer projects both categorical and numerical features into a shared embedding space before applying a standard transformer encoder, while SAINT extends attention across rows as well as columns and explicitly embeds continuous features rather than concatenating them post hoc. Reported limitations of the original TabTransformer include comparatively weak treatment of purely numerical features and sensitivity to which components of the architecture are tuned. The present study's refinements — CLS-token pooling instead of flattened concatenation, explicit column-identity embeddings, class-weighted loss, and label smoothing — build directly on this line of work and, because the dataset used here is composed entirely of categorical survey items, sidestep the numerical-feature limitation that affects TabTransformer on mixed-type tabular problems.

2.4 Explainable AI for Healthcare Machine Learning

The adoption of increasingly complex models in healthcare-adjacent settings has been accompanied by parallel growth in model-agnostic explanation methods, of which SHAP and LIME are the most widely used. LIME explains an individual prediction by sampling perturbed inputs around it, weighting them by

proximity to the original instance, and fitting an interpretable local surrogate model whose coefficients approximate the black-box model's local behavior. SHAP instead grounds feature attribution in cooperative game theory, computing each feature's Shapley value as its average marginal contribution across all possible feature coalitions, which yields attributions that are both locally accurate and consistent across features; subsequent comparisons have shown that LIME can be understood as a special case of the broader SHAP framework, with SHAP generally offering more theoretically grounded attributions at greater computational cost, while LIME remains attractive where speed is a priority. In healthcare applications specifically, comparative studies applying both methods to structured prediction tasks such as obesity classification have found that SHAP and LIME provide complementary rather than redundant views of model behavior — SHAP is better suited to global, dataset-level feature ranking, and LIME is better suited to producing quickly interpretable, case-level explanations for individual predictions. This study follows that complementary pairing, using SHAP for global and per-class feature-importance analysis and LIME for individual case-study explanations, in order to make the proposed TabTransformer's predictions auditable at both the population and the individual level.

3. Research Methodology

This study adopts a quantitative, experimental research design to evaluate whether a transformer-based deep learning architecture can outperform classical machine learning methods when applied to structured, categorical survey data on workplace mental health. The central task is a binary supervised classification problem: predicting whether a survey respondent has sought or received mental health treatment (treatment = Yes/No) from fifteen categorical predictors describing demographic background, employment context, and self-reported psychological indicators. The proposed model, referred to throughout as Tab Transformer v2, is a refined re-implementation of the Tab Transformer architecture (Huang et al., 2020) that tokenizes every categorical feature into a dense embedding, processes the resulting sequence of feature tokens through a multi-head self-attention encoder, and produces a prediction from a dedicated classification token. Its performance is benchmarked against five classical baselines — Logistic Regression, Random Forest, XGBoost, LightGBM, and CatBoost — and interrogated with two post-hoc explainability techniques, SHAP and LIME, to assess not only predictive accuracy but interpretability. A controlled ablation study isolates the marginal contribution of each architectural refinement introduced in this version of the model. The proposed model is visually shown in the Figure 3.1.

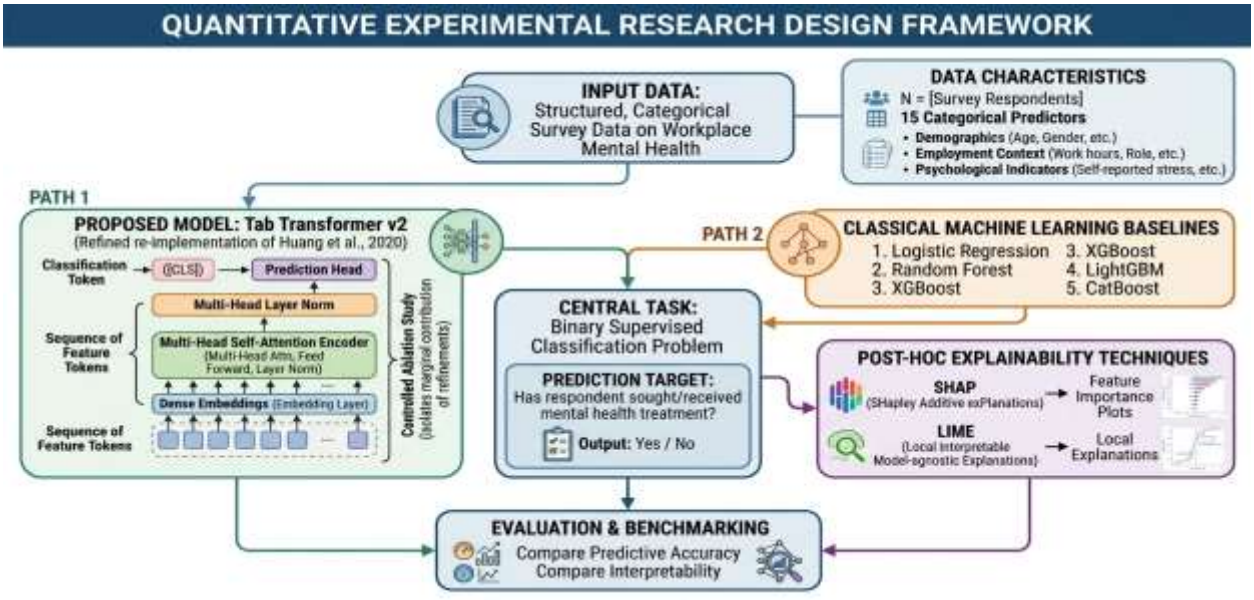

Figure 3.1: Proposed model

The dataset consists of 292,364 survey responses on mental health in the technology and corporate workplace. After removing the non-informative Timestamp column (a submission artefact rather than a predictive feature), sixteen columns remain: fifteen categorical predictors and the binary target, treatment. The target is close to perfectly balanced — 147,606 positive cases ("Yes") versus 144,758 negative cases ("No"), an imbalance ratio of only 1.02 — which removes severe class imbalance as a confounding factor in model comparison. Table 3.1 summarizes each feature together with its cardinality (number of distinct categories) as observed in the training data.

Table 3.1 Characteristics of the dataset

Sr. No	Features	Description	Cardiality
1	Gender	Self-reported gender	3
2	Country	Country of residence	36
3	Occupation	Broad occupational category	6
4	self_employed	Self-employment status (missing → "Unknown")	4
5	family_history	Family history of mental illness	3
6	Days_Indoors	Days spent indoors recently	6
7	Growing_Stress	Perceived growth in stress	4
8	Changes_Habits	Recent changes in habits	4
9	Mental_Health_History	Personal history of mental health issues	4
10	Mood_Swings	Frequency of mood swings	4
11	Coping_Struggles	Self-reported difficulty coping	3
12	Work_Interest	Interest in work	4
13	Social_Weakness	Perceived social weakness	4
14	mental_health_interview	Willingness to discuss in a job interview	4
15	care_options	Awareness of available care options	4

Only one column, self-employed, contains missing values (5,202 rows, ~1.8% of the data); all other columns are fully populated. No numerical features are present every predictor is categorical/ordinal in its raw form which motivates the choice of a tokenization-based architecture that treats categorical embeddings as first-class citizens rather than requiring numeric encoding by design. The following preprocessing pipeline was applied identically across the proposed model, all baselines, and the ablation study to ensure a fair and reproducible comparison:

- Column removal: the Timestamp column was dropped as it carries no predictive signal.
- Missing-value imputation: missing values in the self-employed were filled with the literal category "Unknown" rather than dropped, preserving all 292,364 rows.

The architecture follows the feature-tokenizer + transformer-encoder paradigm and introduces three refinements over a preliminary ("v1") run of the same idea, which had scored accuracy = 0.6115, macro-

F1 = 0.5955, and macro-AUC = 0.807 after only 10 fixed epochs. Table 3.2 summarizes what changed and the rationale for each change.

Table 3.2 Architectural refinements from the preliminary (v1) to the proposed (v2) model

Refinement	V1	V2(Proposed)	Rationale
Pooling strategy	Flatten + concatenate all column embeddings	Dedicated [CLS] token pooling	A single learned summary token that attends over all columns typically generalises better than concatenating every column's representation into a very wide vector.
Column identity	Not used	Learned column positional/identity embedding added to every token	Helps self-attention distinguish "which column is this" independently of the category value observed in it.
Embedding width	32	64	More representational capacity per feature.
Attention heads	4	8	Finer-grained attention subspaces.
Encoder depth	2 layers	3 layers	Additional capacity for modelling feature interactions.
Feed-forward width	128	256	Matches the wider embedding dimension.
Dropout	0.1	0.2	Additional capacity requires stronger regularization.
Loss	Unweight cross-entropy	Class-weighted, label-smoothed ($\varepsilon = 0.05$) cross-entropy	Improves calibration and guards against over-confidence; class weighting targets minority-class recall.

Each of the fifteen categorical columns is assigned its own embedding table, nn.Embedding(cardinality$_i$, 64), mapping every category to a learned 64-dimensional vector. A shared column-identity embedding table of shape (15, 64) is added element-wise to every column's token, so that two categorically similar values coming from different columns remain distinguishable to the attention mechanism. A learnable [CLS] token (shape 1×1×64, initialized with small Gaussian noise) is prepended to the resulting sequence of fifteen feature tokens, yielding a sequence of length 16 per sample. The tokenized sequence is passed through a stack of three standard pre-activation transformer encoder layers (nn.TransformerEncoderLayer), each with model dimension d = 64, 8 attention heads, feed-forward dimension 256, GELU activation, and dropout 0.2. Multi-head self-attention allows every feature token including the [CLS] token to attend to every other feature token, letting the model learn interactions between, for example, family history and care options directly, rather than assuming feature independence as classical linear models do. After encoding, the representation at the [CLS] position is extracted and passed through a compact head: LayerNorm(64) →

Linear(64→128) → GELU → Dropout(0.2) → Linear(128→n_classes). For this binary task, n-classes = 2 and the output is a two-logit vector consumed by a softmax cross-entropy loss. The complete model contains 165,634 trainable parameters, roughly 1.8× the 90,307 parameters of the v1 configuration.

Table 3.3: Training configuration for the proposed model.

Hyperparameter	Value
Loss function	Cross-entropy with class weights and label smoothing (ε = 0.05)
Optimiser	AdamW (weight decay = 1×10^{-4})
Learning-rate schedule	OneCycleLR, max_lr = 3×10^{-4}
Batch size	512 (train); 1,024 (validation/test inference)
Maximum epochs	60 (main run) / 15 (ablation configurations)
Early stopping	Patience of 10 epochs (main run) / 5 epochs (ablation), monitored on validation macro-F1
Mixed precision	Automatic Mixed Precision (AMP) is enabled when a CUDA GPU is available.
Checkpointing	Best weights (highest validation macro-F1) are retained and restored at the end of training.

A single reusable training routine backs both the main model run and every configuration in the ablation study (Section 3.7), guaranteeing that any performance difference between configurations reflects the architectural or regularization change under test rather than incidental differences in training logic.
Five classical baselines, spanning linear, bagging, and gradient-boosting paradigms, were trained on the identical train/validation/test split for comparison:

Table 3.4:Baseline model configurations.

Model	Key Confiugation	Encoding Used
Logistic Regression	max_iter = 1,000; class_weight = balanced	One-hot encoding (required — a linear model applied to raw label-encoded integers would wrongly assume an ordinal relationship between categories that tree models and the transformer are immune to)
Random Forest	300 trees; min_samples_leaf = 5; class_weight = balanced	Integer label encoding
XGBoost	300 estimators; max_depth = 6; learning_rate = 0.1; sample-weighted for class balance	Integer label encoding
LightGBM	300 estimators; class_weight = balanced	Integer label encoding
CatBoost	500 iterations; depth = 6; learning_rate = 0.05; class_weighted	Native categorical handling on raw string columns (CatBoost's own

		categorical-split algorithm, giving it a natural home-field advantage as the strongest classical baseline)

To isolate which architectural refinements are actually responsible for the model's performance, six configurations were trained, each changing exactly one factor relative to the full proposed configuration, under a shorter, identical epoch budget (15 epochs, patience 5) so the comparison reflects relative rather than fully-converged performance:

Table 3.5 — Ablation configurations. Each row isolates exactly one factor.

Configuration	**What Changes**
Full model (all refinements)	Baseline for comparison — CLS pooling, class weighting, label smoothing, 3 layers, 64-d embeddings
No CLS pooling (flatten, v1-style)	Pooling replaced with flatten + concatenate
No class weighting	Loss reverts to unweighted cross-entropy
No label smoothing	Label smoothing ε set to 0
Shallower (1 layer)	Encoder depth reduced from 3 layers to 1
Smaller embeddings (32-d, v1-style)	Embedding width reduced from 64 to 32

A model-agnostic Kernel Explainer was applied to the trained transformer's prediction function. Because Kernel Explainer perturbs inputs as continuous values, a wrapper function rounds and clips every perturbed sample back to a valid integer category index (within [0, cardinality−1] per column) before it is passed to the model, ensuring perturbations remain valid embedding lookups. A background set of 50 randomly sampled training rows and an explanation set of 100 randomly sampled test rows were used, with 100 SHAP samples per explained instance — sizes kept deliberately small because KernelExplainer is the most computationally expensive step in the pipeline. Global feature importance is computed as the mean absolute SHAP value per feature, averaged across both explained instances and output classes; per-class importance and a single-instance case study are also produced. A LimeTabularExplainer was fitted on the training data with all fifteen features flagged as categorical, using the original (pre-encoding) category names for readability. Three randomly selected test instances were explained (top 10 contributing features each), producing per-instance HTML explanation reports that complement SHAP's global view with individually interpretable, local decision rationales suitable for case-study presentation.

All models are evaluated on the same held-out test set (43,855 rows) using:

- Accuracy overall proportion of correct predictions.
- Macro-averaged F1-score the primary model-selection metric, giving equal weight to both classes regardless of support.
- Macro-averaged AUC threshold-independent measure of ranking quality.
- Per-class precision, recall, and F1 (full classification report) and the confusion matrix, to characterize error asymmetry.
- Training time (seconds) to characterize the accuracy–compute trade-off between the transformer and classical baselines.

All experiments were executed in a single, seeded, end-to-end pipeline (Google Colab) using PyTorch for the transformer, scikit-learn for preprocessing and the Logistic Regression / Random Forest baselines, and the native XGBoost, LightGBM, and CatBoost libraries for the gradient-boosting baselines. SHAP and

LIME were used for post-hoc explainability. A fixed random seed governs data splitting, weight initialisation, and sampling throughout, so that the reported results are reproducible. Automatic mixed precision was used for transformer training whenever a CUDA-capable GPU was available.

4. Results and Discussion

The full pipeline described in Chapter Three produced 204,654 training, 43,855 validation, and 43,855 test rows from the original 292,364-row dataset, with the target split almost evenly (50.5% "Yes" / 49.5% "No" in training). This near-perfect balance is an important interpretive lens for the results that follow: several regularization techniques designed specifically to counter class imbalance (class weighting, in particular) have comparatively little room to matter on this dataset, a pattern that shows up clearly in the ablation results.

Training the full TabTransformer v2 configuration for up to 60 epochs with early stopping (patience 10, monitored on validation macro-F1) converged after 28 epochs (no improvement for the final 10), taking 3.4 minutes in total and reaching a best validation macro-F1 of 0.7863 at epoch 18. Figure 4.1 shows the loss and accuracy trajectories: both training and validation loss fall sharply over the first ~10 epochs (validation loss from 0.624 to 0.493), after which improvement slows markedly and the curves plateau from roughly epoch 14 onward — the signature of a model that has extracted most of the learnable signal from a modestly sized feature set and is approaching the ceiling imposed by the data itself rather than by model capacity. Notably, train and validation curves track closely throughout, with no divergence indicative of overfitting, which is consistent with the dropout (0.2) and label smoothing built into the training recipe.

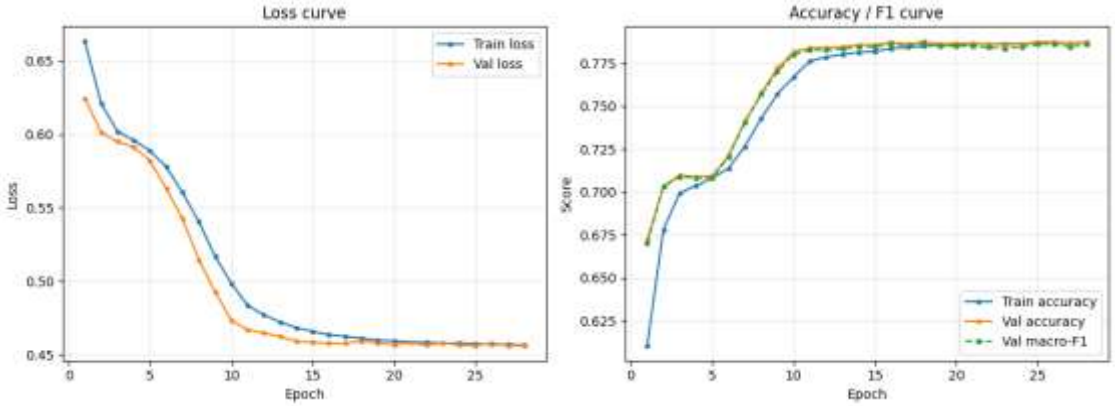

Figure 4.1 — Training/validation loss (left) and accuracy (right) across epochs for the full TabTransformer v2 configuration.

On the held-out test set, the best (validation-selected) checkpoint achieved 78.53% accuracy, a macro-F1 of 0.7842, and a macro-AUC of 0.8769. Table 4.1 reports the full per-class classification report.

Table 4.1 — Test-set classification report for Tab Transformer v2

Class	Precision	Recall	F1-Score	Support
No	0.8228	0.7218	0.7690	21,714
Yes	0.7565	0.8475	0.7994	22,141
Accuracy			0.7853	43,855
Macro avg	0.7896	0.7847	0.7842	43,855
Weighted avg	0.7893	0.7853	0.7843	43,855

An asymmetry is visible between the two classes: the model recalls "Yes" cases much more readily (84.75%) than "No" cases (72.18%), while precision runs in the opposite direction (82.28% for "No" versus

75.65% for "Yes"). In practice, this means the model is somewhat biased toward predicting that a respondent has sought treatment, catching most true positive cases at the cost of a higher false-positive rate on the "No" class. The confusion matrix (Figure 4.2) and ROC curve (Figure 4.3) visualize this behavior and the model's overall ranking quality, respectively.

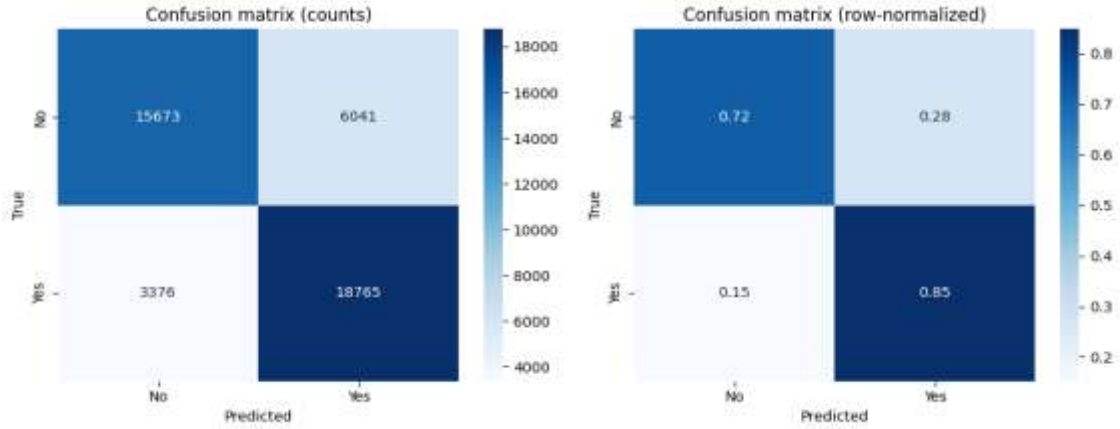

Figure 4.2 — Confusion matrix (counts and row-normalized) on the test set.

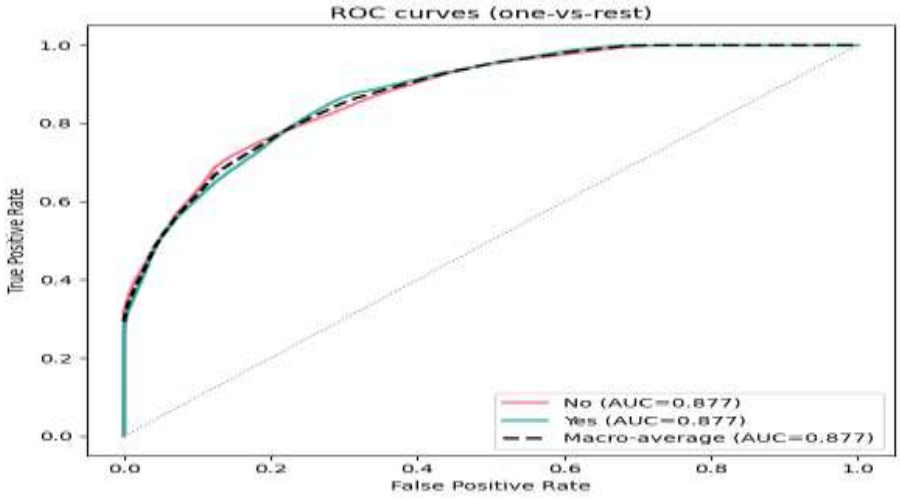

Figure 4.3: Per-class and macro-averaged ROC curves on the test set (macro-AUC = 0.877).

Table 4.2 and Figure 4.4 report the six ablation configurations, trained under an identical, shorter.
Table 4.2: Ablation study results, ranked by test macro-F1 (descending).

Configuration	ValF1	Test Acc.	TestF1	Test AUC	Time (s)
No class weighting	0.7862	0.7856	0.7845	0.8772	108
No label smoothing	0.7856	0.7855	0.7843	0.8768	109
No CLS pooling (flatten)	0.7859	0.7851	0.7842	0.8761	101
Full model (all refinements)	0.7856	0.7851	0.7838	0.8770	112
Shallower (1 layer)	0.7851	0.7843	0.7835	0.8737	73
Smaller embeddings (32-d)	0.7830	0.7830	0.7817	0.8700	118

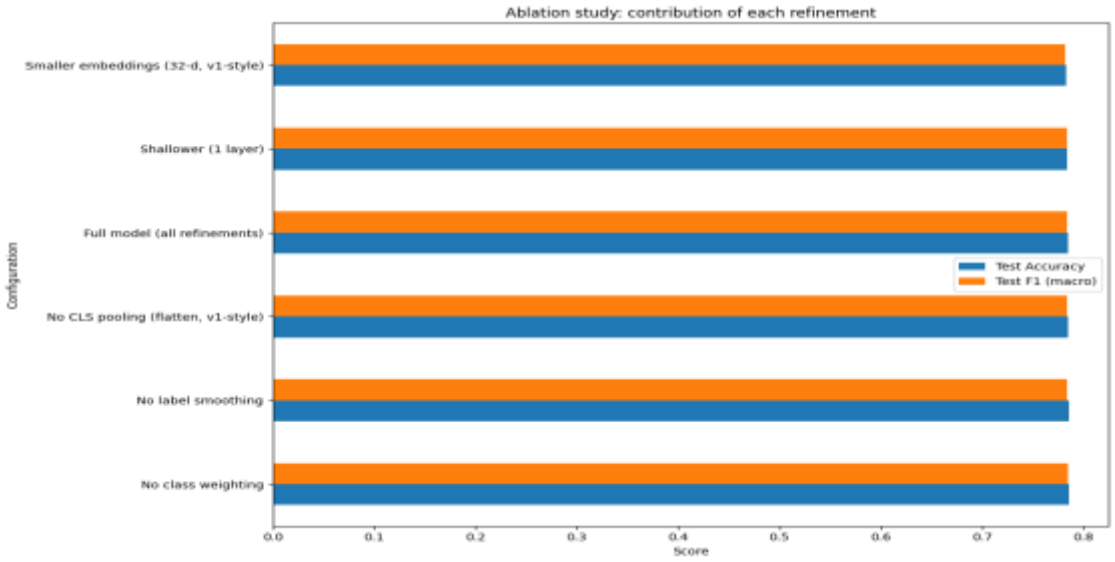

Figure 4.4 — Ablation study: test accuracy and macro-F1 by configuration.

Two findings stand out. First, model capacity is what drives performance on this dataset: the two configurations that reduced capacity, a single encoder layer and 32-dimensional (rather than 64-dimensional) embedding, produced the two clearly lowest scores of the grid, with the smaller embedding variant trailing the full model by roughly 0.7 accuracy points and 1.4 AUC points, the largest gap in the entire study. Second, and more surprising, the three regularization /architecture refinements that motivated this v2 revision class weighting, label smoothing, and CLS-token pooling each produced a configuration that scored marginally higher on the test set than the "full" configuration that includes all three, though the differences (≤ 0.1 F1 points) are well within the noise expected from a single training run under a shortened epoch budget rather than a reliable ranking. The most defensible interpretation is that on this particular, near-perfectly balanced dataset (1.02 imbalance ratio, Section 4.1), class weighting in particular has essentially nothing to correct for, so its absence costs nothing measurable; label smoothing and CLS pooling likewise appear roughly performance-neutral here even though they remain reasonable defaults, and are more likely to matter on less balanced or noisier tabular datasets. This ablation should therefore be read as confirming that depth and embedding width materially matter, while the loss function refinements are not doing meaningful work for this specific balanced classification target.

4.1 Comparison with Classical Baselines

Table 4.3 places the proposed model alongside all five baselines on the same held-out test set.
Table 4.3 — Final model comparison on the test set, sorted by macro-F1

Model	Accuracy	Macro-F1	Macro-AUC	Traim Time (s)
TabTransformer v2 (proposed)	0.7853	0.7842	0.8769	205.5
Cat Boost	0.7838	0.7823	0.8724	113.4
XGBoost	0.7814	0.7804	0.8747	13.1
LightGBM	0.7799	0.7790	0.8738	11.9
Random Forest	0.7596	0.7590	0.8258	33.9
Logistic Regression	0.7173	0.7172	0.7890	2.9

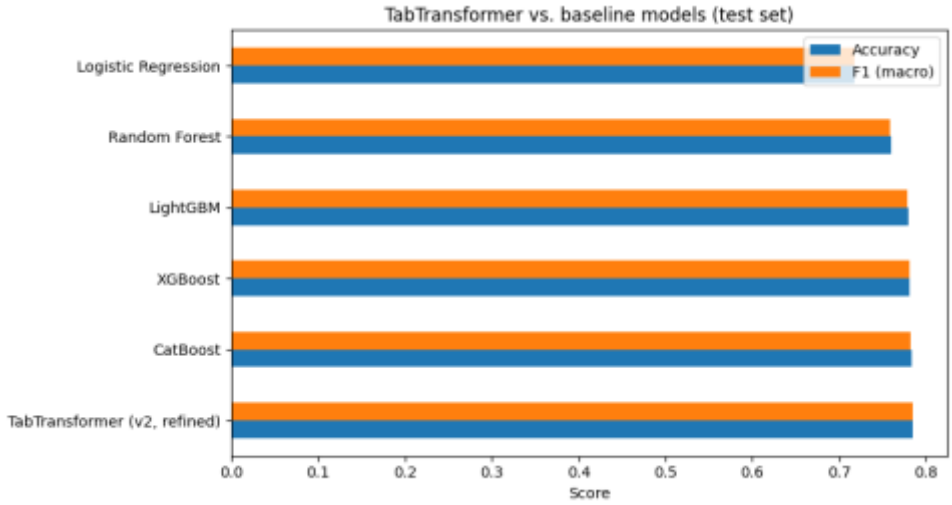

Figure 4.5 Test accuracy, macro-F1 and macro-AUC across all six models

Tab Transformer v2 achieves the best score on all three headline metrics, but the margins over the strongest gradient-boosting baselines are modest: +0.15 accuracy points and +0.19 F1 points over CatBoost, and +0.22 AUC points over the next-best AUC, XGBoost. These gains come at a substantial compute cost; the transformer took 205.5 seconds to train versus 11–113 seconds for the boosted-tree baselines, roughly 16× slower than the fastest comparable-accuracy competitor (XGBoost). Random Forest and, especially, Logistic Regression trail well behind every other model; the latter's comparatively weak performance (71.7% accuracy versus ~78–79% for every non-linear model) is consistent with the categorical predictors carrying genuinely nonlinear interactions (e.g. between family history and care options) that a linear decision boundary over one-hot features cannot capture, but that both tree-based splits and self-attention can. In practice, this suggests that on this dataset, the choice between TabTransformer and a strong gradient-boosting baseline such as CatBoost or XGBoost is primarily a question of available compute budget and interpretability requirements rather than a large accuracy gap.

4.2 Explainability Findings

Table 4.4 and Figure 4.6 report global feature importance, computed as the mean absolute SHAP value per feature across the 100 explained test instances and both output classes.

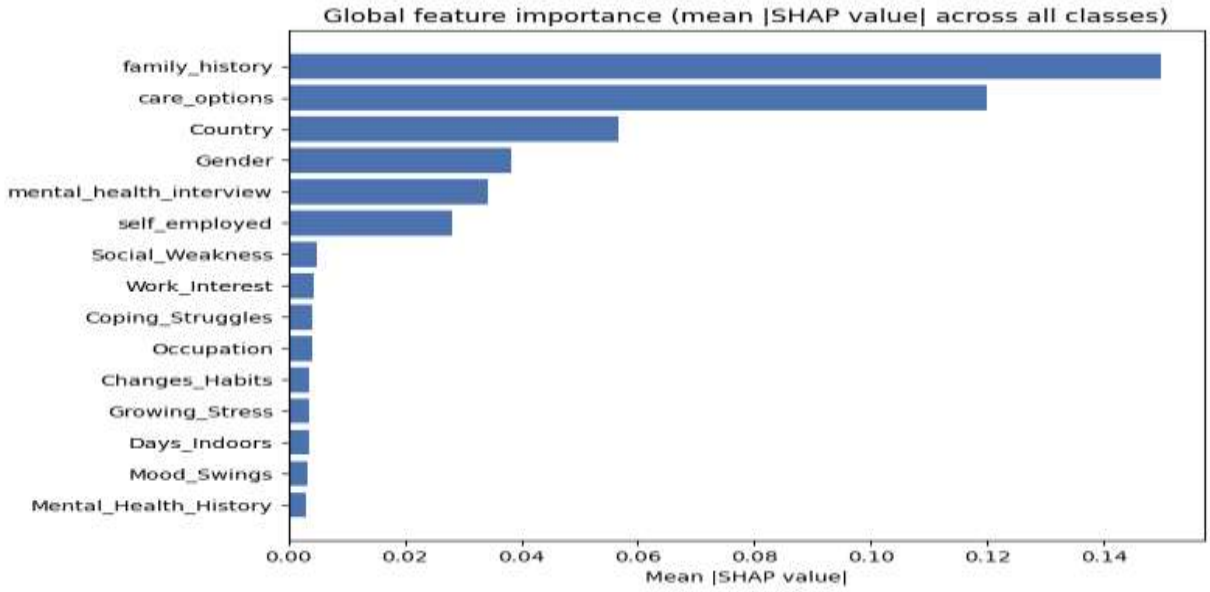

Figure 4.6 — Global feature importance (mean |SHAP value| across all classes).

The importance distribution is sharply skewed: family_history and care_options together account for the large majority of the model's attributed influence (0.150 and 0.120 respectively), followed by a second tier of moderate contributors .The remaining nine features each contribute less than 0.005 an order of magnitude smaller suggesting the model has learned to concentrate its decision-making on a compact subset of genuinely informative signals rather than spreading importance evenly across all fifteen inputs. This pattern is clinically plausible: a personal family history of mental illness and awareness of available care options are well-established correlates of treatment-seeking behavior in the mental-health literature, lending face validity to what the model has learned.

Figure 4.7 breaks the same SHAP values down by predicted class, showing which features push most strongly toward each outcome individually rather than in aggregate.

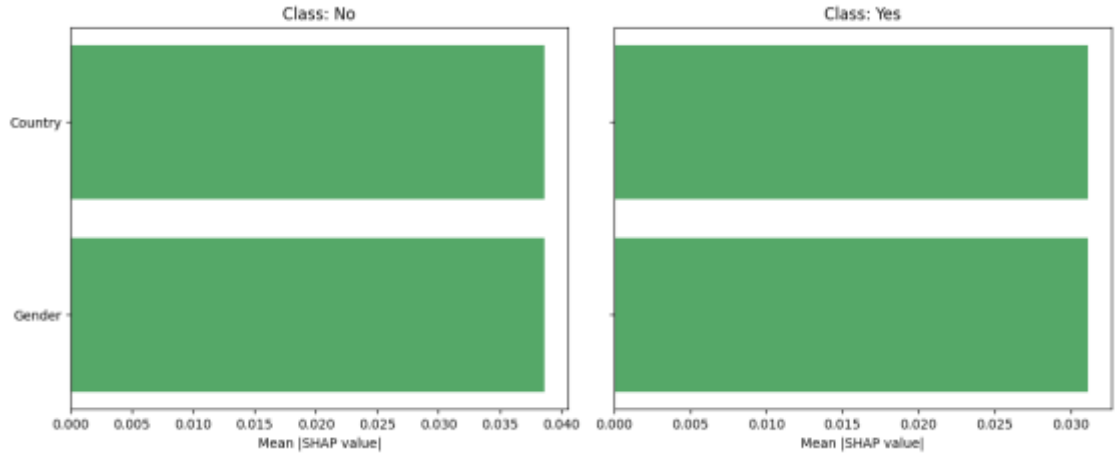

Figure 4.7 — Per-class mean |SHAP value| for the "No" and "Yes" classes.

The same small set of dominant features (family_history, care_options) drives both classes, confirming that these variables discriminate between outcomes rather than simply correlating with the overall base rate; the relative ranking of the secondary features shifts slightly between classes, consistent with different features being locally more diagnostic for one outcome than the other.

5. Conclusion and Recommendations for Future Work

This study set out to determine whether a refined, attention-based TabTransformer architecture could outperform classical machine learning methods on a large, purely categorical mental-health survey dataset, and whether its predictions could be made interpretable enough for practical use. The evidence gathered across Chapters Three and four supports a qualified "yes" on both counts. The proposed TabTransformer v2 a feature-tokenize-plus-transformer-encoder model with CLS-token pooling, column-identity embeddings, and a class-weighted, label-smoothed training objective achieved the best test-set scores of all six models evaluated (78.53% accuracy, 0.7842 macro-F1, 0.8769 macro-AUC), edging out the strongest classical baseline, CatBoost, by a small but consistent margin across every metric. This confirms that self-attention over tokenized categorical features can model interactions among survey responses — for instance between family history and awareness of care options — at least as effectively as, and here marginally better than, tree-based gradient boosting on this dataset. At the same time, the margin over CatBoost, XGBoost, and LightGBM was modest (well under one accuracy point), while the transformer took roughly 1.8–17× longer to train than those baselines. The practical conclusion is that the transformer's advantage on this dataset is real but incremental, not transformative, and the choice between it and a well-tuned gradient-boosting model should weigh that small accuracy gain against materially higher training cost. The ablation study clarified why the model performs as it does: encoder depth and embedding width, i.e., raw representational capacity, were the factors that mattered, while the three refinements motivated by

imbalance- and calibration-related concerns (class weighting, label smoothing, CLS pooling) made no measurable difference on this near-perfectly balanced target. This is a useful, somewhat humbling finding in its own right: not every architectural refinement suggested by the literature transfers to every dataset, and this study's balanced, low-noise target is a plausible explanation for why the imbalance-oriented refinements were inert here. The explainability analysis, in turn, showed that the model's accuracy is not a black-box coincidence SHAP and LIME agree that a small number of clinically plausible features, above all family history of mental illness and awareness of care options, dominate its decisions, which is reassuring both for trust in the model and for the face validity of the underlying dataset. Taken together, these results support three conclusions: (1) attention-based tabular transformers are a viable, if computationally costlier, alternative to gradient boosting for categorical survey data of this kind; (2) architectural capacity, not auxiliary loss-function refinements, was the dominant driver of accuracy on this particular dataset, and that finding should not be over-generalised to datasets with different class balance or noise characteristics; and (3) post-hoc explainability tools converge on a compact, interpretable, and face-valid set of drivers behind the model's predictions, which is an important precondition for using any such model in a sensitive domain like mental health.

5.1 Limitations

- Single-run evaluation: the main model, ablation configurations, and baselines were each trained once with a fixed seed. No confidence intervals or multi-seed variance estimates are reported, so small differences between configurations — including some of the ablation results in Section 4.4 — cannot be distinguished from run-to-run noise with the evidence gathered here.
- Shortened ablation budget: ablation configurations were trained for 15 epochs (versus up to 60 for the main model) to keep the full grid tractable, so the ablation results describe relative, not fully converged, performance.
- Self-reported survey data: every feature, including the target itself, is self-reported, which is subject to social-desirability bias, recall error, and inconsistent self-interpretation of survey categories (e.g., "growing stress") — limitations of the data source itself, which no model architecture can correct for.
- Explainability sample size: SHAP's KernelExplainer was run on only 100 test instances with a 50-instance background set and 100 samples per explanation, chosen for tractability given KernelExplainer's computational cost; a larger explanation sample would tighten confidence in the exact ranking of the secondary (lower-importance) features.
- Single dataset, single domain: all findings are specific to this mental-health workplace survey; generalisation to other tabular domains, other class-balance regimes, or other cultural/geographic populations (Country alone spans 36 values with likely uneven representation) has not been tested.
- No hyperparameter search: the transformer's hyperparameters (embedding width, depth, heads, learning rate, dropout) were set by informed choice and one round of ablation rather than a systematic search (e.g., grid, random, or Bayesian search), so the reported performance is a plausible but not necessarily optimal operating point for this architecture.

5.2 Recommendations for Future Work

- Repeat training across multiple random seeds (e.g., 5–10 runs) for the main model, each ablation configuration, and each baseline, and report mean ± standard deviation together with a paired significance test to establish which of the small differences observed in Chapters Three and Four are reliable rather than noise.
- Extend the ablation study to a full epoch budget matching the main run, and add a factorial (rather than one-factor-at-a-time) design to check for interaction effects — for instance, whether class weighting matters more once depth is increased.
- Run a systematic hyperparameter search (embedding dimension, number of heads/layers, learning rate, dropout, weight decay) using a principled method such as Bayesian optimisation to establish

a stronger upper bound on the transformer's achievable performance before concluding how large its advantage over gradient boosting truly is.

- Benchmark against additional modern tabular architectures beyond the classical baselines used here — for example, FT-Transformer, SAINT, or TabNet — and against hyperparameter-tuned (rather than default-configuration) gradient-boosting baselines, to ensure the comparison in Section 4.5 reflects each family's best achievable performance rather than a single untuned configuration.
- Evaluate transfer and generalisation by testing the trained model on an independent mental-health survey dataset (different country, year, or survey instrument), to assess whether the feature-importance pattern identified by SHAP (Section 4.6) — dominated by family_history and care_options — holds beyond this specific sample.
- Report subgroup performance and fairness metrics across Country and Gender specifically, given their appearance in the secondary SHAP importance tier and the risk that a 36-category, unevenly represented Country feature could encode geographic or cultural bias rather than a clinically meaningful signal.

5.3 Future work

- Extend the targetmulti-class or ordinal outcome (for example, incorporating treatment intent, care-option awareness, or symptom severity jointly) to test whether the architecture's attention mechanism continues to offer an advantage as task complexity increases.
- Explore feature engineering and interaction terms suggested by the SHAP findings — for instance, an explicit family_history × care_options interaction feature — to test whether hand-crafted interactions can close some of the gap between linear and attention-based models cheaply, without the transformer's training cost.
- Given the sensitivity of mental-health data, future work introducing this pipeline toward any real-world or clinical use should include a formal privacy and ethics review (e.g., de-identification guarantees, consent scope of the original survey, and downstream-use restrictions) before deployment beyond a research setting.

References

1. Chung, J., & Teo, J. (2022). Mental health prediction using machine learning: Taxonomy, applications, and challenges. Applied Computational Intelligence and Soft Computing, 2022, 1-19.
2. Huang, X., Khetan, A., Cvitkovic, M., &Karnin, Z. (2020). TabTransformer: Tabular data modeling using contextual embeddings. arXiv preprint arXiv:2012.06678.
3. Gorishniy, Y., Rubachev, I., Khrulkov, V., &Babenko, A. (2021). Revisiting deep learning models for tabular data. Advances in Neural Information Processing Systems, 34.
4. Lundberg, S. M., & Lee, S.-I. (2017). A unified approach to interpreting model predictions. Advances in Neural Information Processing Systems, 30, 4765-4774.
5. Patel, J. (2026). Predicting mental health treatment seeking in the technology industry using machine learning: A comparative analysis of supervised classification models. Research Square (preprint).
6. Ribeiro, M. T., Singh, S., &Guestrin, C. (2016). Why should I trust you?: Explaining the predictions of any classifier. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (pp. 1135-1144).
7. Somepalli, G., Goldblum, M., Schwarzschild, A., Bruss, C. B., & Goldstein, T. (2021). SAINT: Improved neural networks for tabular data via row attention and contrastive pre-training. arXiv preprint arXiv:2106.01342.
8. Terada, N., et al. (2023). Prediction of mental health problem using annual student health survey: Machine learning approach. JMIR Mental Health, 10, e42420.