Indian Institute of Technology, Dharwad

|| सा विद्या या विमुक्तये ||
ಭಾ.ತಂ.ಸಂ. ಧಾರವಾಡ
भा. प्रौ. सं. धारवाड
I.I.T. DHARWAD

CS209 : Artificial intelligence

And

CS214 : Artificial Intelligence Laboratory

Project: **Predicting Student Depression using Machine Learning**

Course Instructor:

Dr. Dileep A.D.

Mentor Name:

Mrs. Anupama P Bidargaddi

Submitted by:

1. Surya Prakash S	(CS23BT068)
2. P Sri Smaran	(CS23BT039)
3. Rahul Ratlavath	(CS23BT018)
4. Banoth Pavan Kumar	(CS23BT024)

Contents

1 Introduction

Artificial Intelligence (AI) represents a groundbreaking domain within computer science, dedicated to the development of systems capable of executing tasks that traditionally demand human intelligence. These tasks encompass a wide array of capabilities, including learning from experience, reasoning through complex situations, solving intricate problems, and interpreting natural language. The importance of AI stems from its transformative potential across numerous sectors. By enhancing decision-making processes, automating repetitive operations, and extracting profound insights from expansive datasets, AI is poised to revolutionize fields such as healthcare, transportation, and education, thereby reshaping the way we interact with technology and the world around us.

In this report, we harness the power of AI and Machine Learning to tackle a pressing and sensitive issue: the prediction of student depression. The core problem statement can be summarized as:

1. **Exploration of Factors Contributing to Student Depression:** We aim to conduct a comprehensive analysis to uncover the intricate interplay between an array of factors and the incidence of depression among students. Through exploratory data analysis, this objective seeks to identify significant patterns, correlations, and potential causative relationships that influence mental health outcomes in the student population.

2. **Construction of a Robust Machine Learning Model for Depression Prediction:** Our goal is to design and implement a sophisticated machine learning model capable of accurately predicting the depression status of students based on the provided dataset. This involves selecting appropriate algorithms, training the model, and rigorously evaluating its performance using a suite of metrics—including accuracy, precision, recall, and F1-score—to ensure its predictive reliability and practical utility.

3. **Determination of Key Predictors via Feature Importance Analysis:** By leveraging advanced feature importance analysis techniques, we intend to isolate and highlight the most impactful variables driving student depression. This step is crucial for deepening our understanding of the underlying dynamics and for providing actionable insights that can guide the development of targeted interventions and support mechanisms.

4. **Evaluation of Ethical Considerations and Real-World Implications:** Recognizing the sensitivity of applying predictive models within educational environments, an examination of the ethical and practical consequences of our work. This includes addressing critical issues such as ensuring informed consent from participants, safeguarding data privacy, and promoting the responsible deployment of AI-driven solutions to support student mental health initiatives.

2 Methodology

2.1 Dataset Description

The dataset consists of 27,902 rows and features a mix of numerical and categorical attributes. In order to thoroughly understand the data structure and prepare it for predictive modeling, we conducted an extensive Exploratory Data Analysis (EDA) and applied several preprocessing techniques.

The given dataset the following attributes such as *id, Gender, Age, City, Profession, Academic Pressure, Work Pressure, CGPA, Study Satisfaction, Job Satisfaction, Sleep Duration, Dietary Habits, Degree, Have you ever had suicidal thoughts?, Work/Study Hours, Finacial Stress, Family History of Mental Illness, Depression*

Here the last column `Depression` is our target variable for classifying where the given person is depressed or not.

2.2 Exploratory Data Analysis

Our initial analysis involved:

- **Visual Inspection:** Plotting box-plots, correlation matrices, and kernel density estimation (KDE) plots on the raw data to inspect distributions, detect outliers, and understand inter-feature relationships.
- **Data Distribution:** We visualized the underlying data patterns using the following graphs:

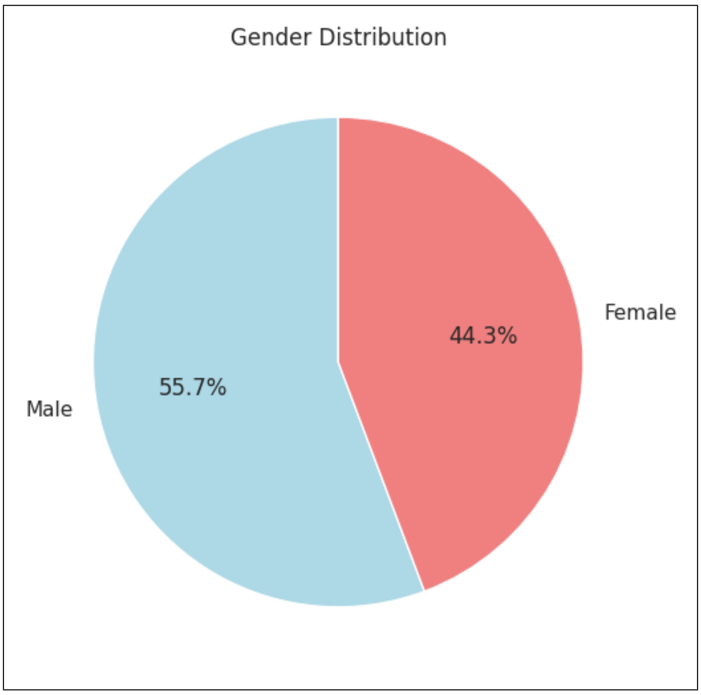


Figure 2.1: Gender Distribution

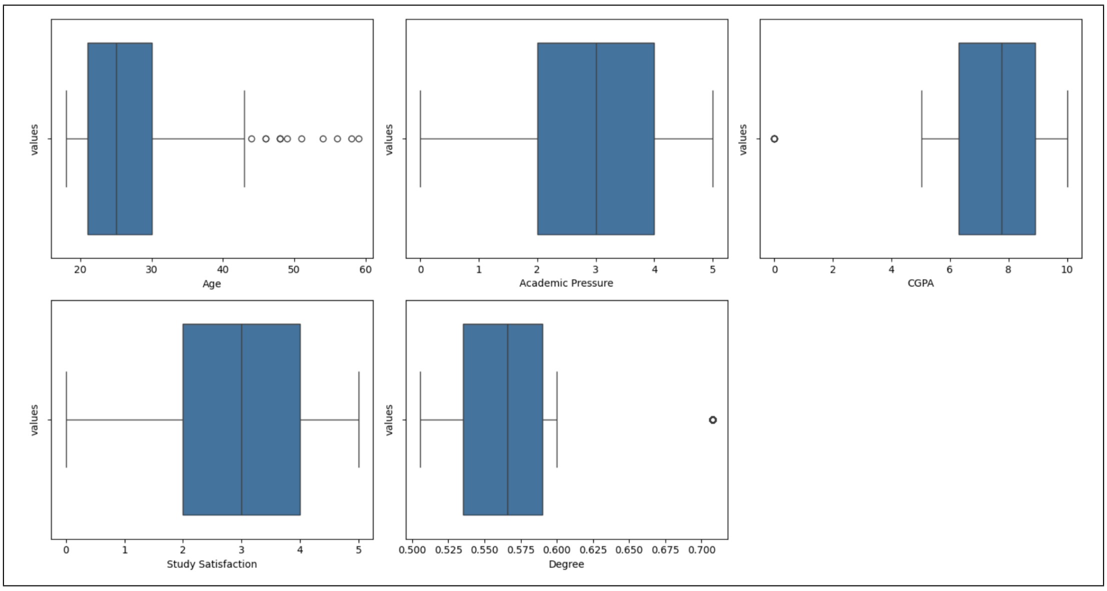


Figure 2.2: Box-plots for key numerical attributes (raw data)

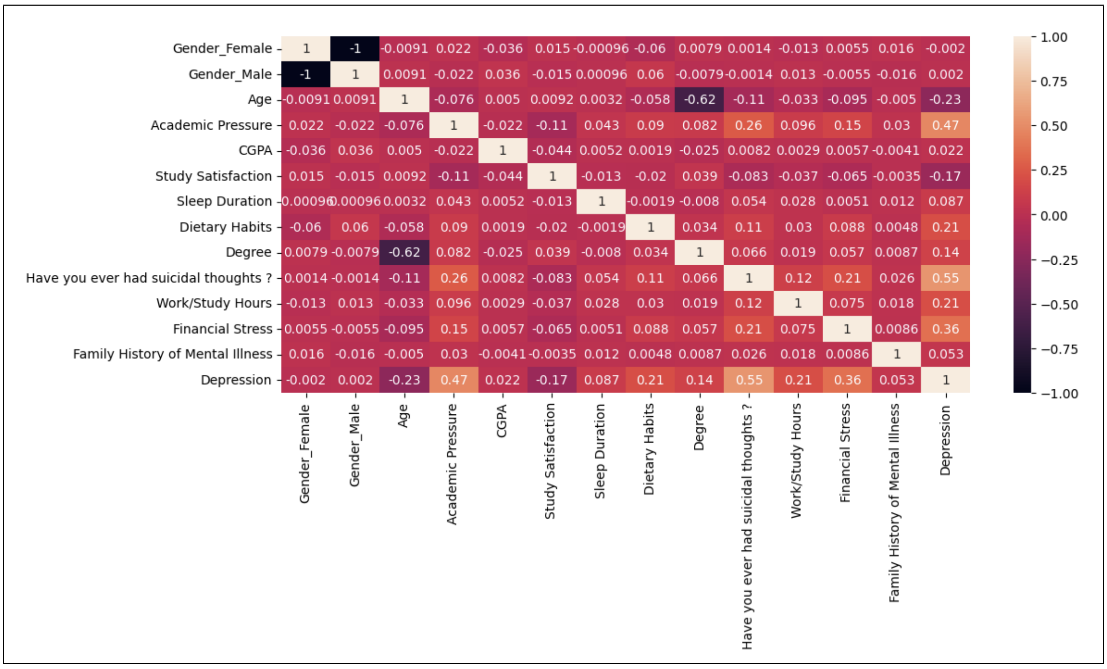


Figure 2.3: Correlation matrix illustrating feature relationships

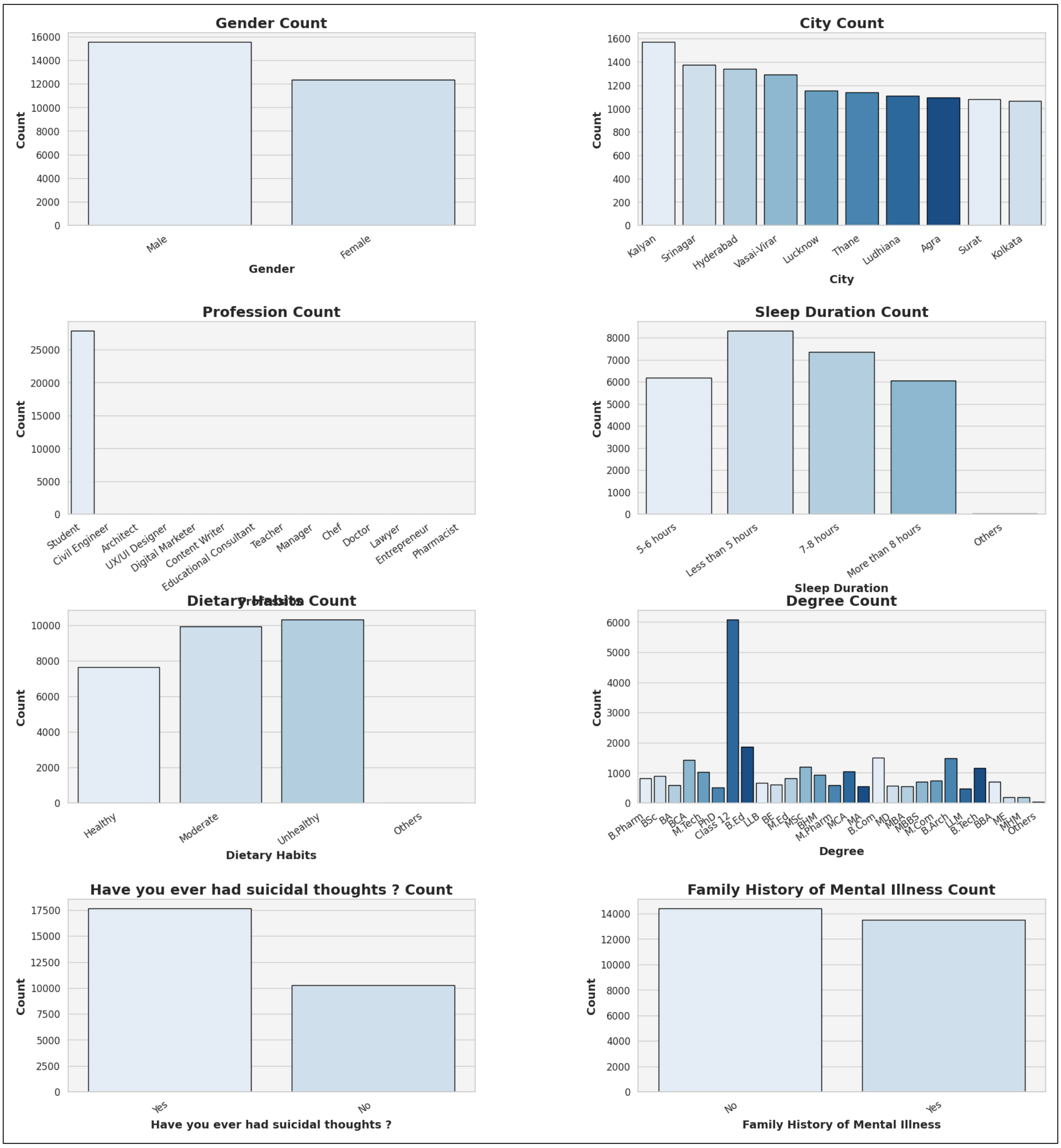


Figure 2.4: Categorical feature distribution

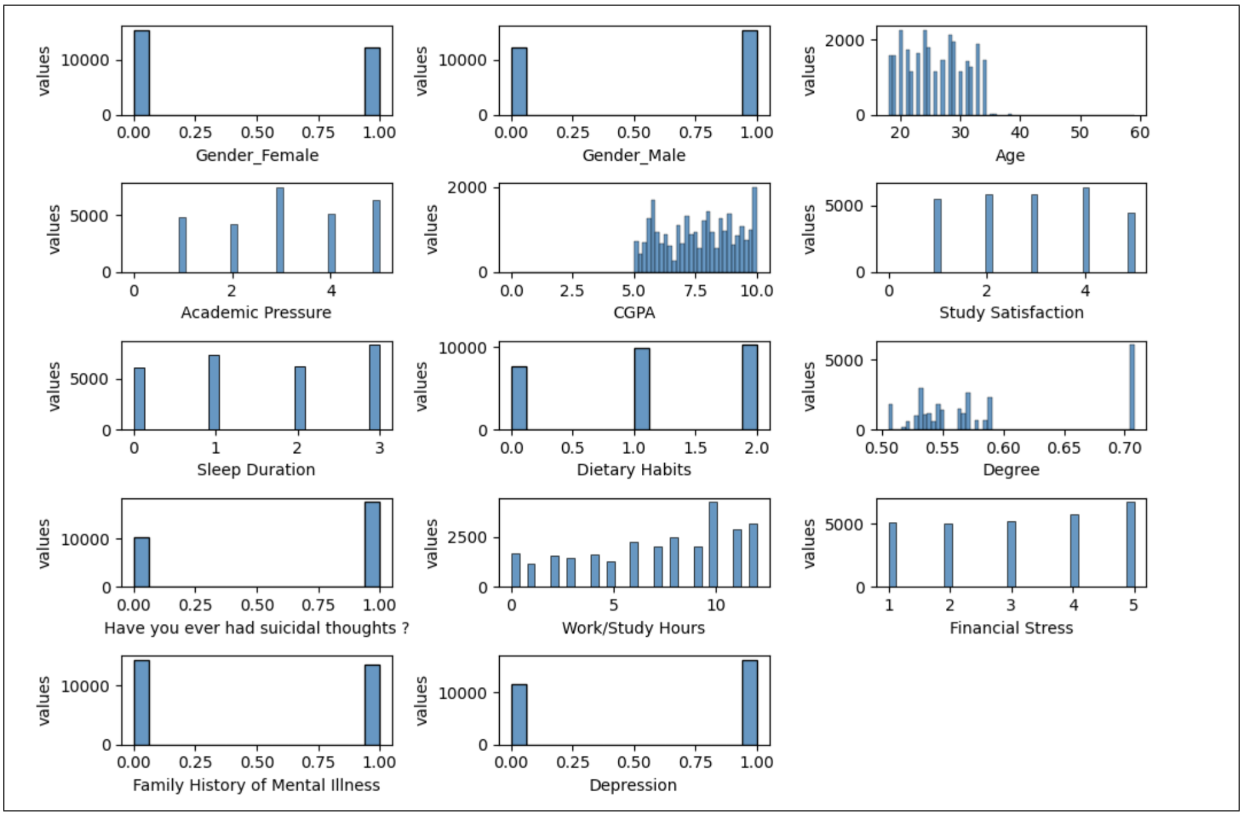


Figure 2.5: Kernel Density Estimation (KDE) plots for all attributes

2.3 Data Preprocessing

Data preprocessing was vital to ensure the reliability of our predictive models. The following steps were implemented:

2.3.1 Missing Values

We verified that there were no missing values in our dataset. Despite this, we established a mechanism to impute missing values by replacing them with the median of the respective attribute. The median method is robust to outliers and preserves the original data distribution.

2.3.2 Outliers

We observed around 10-15 outliers in each numerical column, so we didnt do anything about them as there was no measurable change in model performance. As for categorical, the `Profession` column had over 27000 rows with the value `student` so we ended up dropping all rows which weren't student.

2.3.3 Handling Redundant Features

Since the vast majority of the samples correspond to students, we removed the **Profession** attribute. In addition, we dropped:

- **Job Satisfaction:** Constant zero values for students.
- **Work Pressure:** Constant zero values for students.
- **ID:** Irrelevant for predictive analysis.

2.3.4 Categorical Encoding

Given the diversity of categorical attributes, we applied various encoding techniques:

- **One-Hot Encoding:** For the *Gender* attribute (male/female), we transformed it into binary columns (e.g., `isMale`, `isFemale`), ensuring models can process these as numerical data.
- **Ordinal Encoding:** For attributes with an inherent order, we mapped their categorical values to numerical values:
  - *Dietary Habits*: {"Healthy": 0, "Moderate": 1, "Unhealthy": 2, "Others": 3}
  - *Have you ever had suicidal thoughts?*: {"No": 0, "Yes": 1}
  - *Sleep Duration*: {"More than 8 hours": 0, "7-8 hours": 1, "5-6 hours": 2, "Less than 5 hours": 3, "Others": 4}

  Ordinal encoding is chosen here because these features carry an intrinsic rank or order which the models can benefit from while maintaining interpretability.
- **Target Encoding:** For features such as *City* and *Degree*, which are ordinal by nature, we applied target encoding. This method replaces categorical values with the average value of the target (depression indicator), capturing subtle relationships between the category levels and the outcome.

2.3.5 Feature Scaling with Z-Score Normalization

To normalize numerical features, we applied Z-score normalization using the formula:

$$z = \frac{x - \mu}{\sigma}$$

where x is a given value, μ is the mean of the feature, and σ is the standard deviation. This transformation standardizes the features to have a mean of zero and a standard deviation of one, which is essential for many machine learning algorithms.

2.4 Training Methodology

2.4.1 Train-Test Split

The preprocessed data was split into training (80%) and testing (20%) sets in a stratified manner, particularly preserving the distribution of the depression outcome. This approach maintains the statistical properties of the target variable in both subsets.

2.4.2 Cross-Validation

For a reliable performance estimation, we used 5-fold cross-validation via stratified k-fold splitting. In addition, a custom `Cleaner` class was implemented with `fit` and `transform` methods to ensure that the preprocessing steps are encapsulated within the pipeline, preventing any potential data leakage.

2.4.3 Bayesian Optimization for Hyperparameter Tuning

Instead of an exhaustive grid search, we adopted Bayesian optimization, which efficiently guides the hyperparameter search towards promising regions of the search space. By building a probabilistic model of the objective function, Bayesian optimization adaptively samples the hyperparameter space. Formally, we aim to maximize the cross-validation score $f(\theta)$ where θ represents a set of hyperparameters:

$$\theta^* = \arg\max_{\theta} f(\theta)$$

This method leverages prior evaluations to inform subsequent choices, significantly reducing computational cost while effectively optimizing model performance.

2.4.4 SHAP for Model Explainability

To improve interpretability of the model predictions, we employed SHAP (SHapley Additive exPlanations). SHAP values quantify the contribution of each feature to individual predictions, based on cooperative game theory. This method provides local explanations that help us understand how features influence the model's decision, ensuring transparency in predictive outcomes.

`Cooperative game theory` is a branch of game theory that studies how groups of players (or agents) can work together (cooperate) to achieve a common goal and how the total payoff can be fairly distributed among them. In the context of SHAP, each feature is considered a ”player” in a game, and the prediction is the ”payout.” SHAP calculates each feature's contribution by considering all possible combinations of features and how much each one adds to the prediction, ensuring a fair and consistent attribution.

2.5 Causal Analysis

2.5.1 PC Algorithm for Causal Discovery

The PC algorithm is a constraint-based method for inferring causal structures from observational data. It operates by testing conditional independencies among variables to construct a partially directed acyclic graph (PDAG). The algorithm assumes faithfulness, meaning that all independencies in the data are implied by the graph structure.

Given a set of variables $\mathbf{V}$, the PC algorithm iteratively tests whether variables X and Y are conditionally independent given a subset $\mathbf{S} \subseteq \mathbf{V} \setminus \{X, Y\}$, denoted as $X \perp Y \mid \mathbf{S}$. If such a set $\mathbf{S}$ exists, no direct edge is placed between X and Y; otherwise, an edge is added.

2.5.2 Causal Effect Estimation

Once the causal graph is established, the causal effect of a treatment T on an outcome Y is estimated using the backdoor criterion. For a set of variables $\mathbf{Z}$ that blocks all backdoor paths from T to Y, the causal effect is given by:

$$P(Y \mid do(T = t)) = \int P(Y \mid T = t, \mathbf{Z} = \mathbf{z}) P(\mathbf{z}) d\mathbf{z}$$

In practice, this is approximated using linear regression or propensity score matching, depending on the nature of the outcome variable. But we have used the linear regression method.

2.5.3 Refutation Tests

To assess the robustness of the estimated effects, refutation tests are conducted:

- *Random Common Cause*: Introduces a random variable as a confounder to check if the estimated effect changes significantly.
- *Placebo Treatment*: Replaces the treatment with a placebo (random variable) to ensure that the estimated effect is zero, validating the model's integrity.

High p-values (> 0.05) in these tests indicate that the estimates are robust.

2.5.4 Introduction to ATE with DoWhy

The Average Treatment Effect (ATE) is a key measure in causal inference, representing the average difference in outcomes if everyone in a population received a treatment versus if no one did. Mathematically, ATE is expressed as:

$$\text{ATE} = \mathbb{E}[Y(1)] - \mathbb{E}[Y(0)]$$

where $Y(1)$ denotes the potential outcome under treatment, and $Y(0)$ denotes the potential outcome without treatment.

2.5.5 Nonparametric Estimation

Nonparametric methods estimate ATE without assuming a specific functional form for the relationship between the treatment, covariates, and outcomes. This approach is particularly valuable when the true relationship is unknown or complex, allowing the data to guide the estimation process without imposing restrictive assumptions.

2.5.6 DoWhy Framework

DoWhy is a Python library tailored for causal inference, offering a range of tools to estimate causal effects, including nonparametric methods.

2.5.7 Nonparametric Estimation in DoWhy

In DoWhy, nonparametric ATE estimation can be performed using techniques like matching or weighting. For example, propensity score matching balances treated and untreated groups based on the propensity score, defined as $e(X) = P(T = 1|X)$, the probability of receiving treatment given covariates X. This helps ensure that the groups are comparable, enabling a reliable estimate of ATE.

The estimated ATE can be computed as:

$$\widehat{\text{ATE}} = \frac{1}{N} \sum_{i=1}^{N} \left(\hat{Y}_i(1) - \hat{Y}_i(0) \right)$$

where $\hat{Y}_i(1)$ and $\hat{Y}_i(0)$ are the estimated potential outcomes for individual i under treatment and control conditions, respectively, derived from methods like matching or weighting.

2.6 Causal Analysis Methodology

2.6.1 Initial Graph Discovery with the PC Algorithm

The PC algorithm was applied to the dataset containing variables such as Academic Pressure, CGPA, Sleep Duration, Suicidal Thoughts, Dietary Habits, Age, and Family History of Mental Illness, with Depression as the outcome. This generated an initial causal graph, which included both plausible and implausible edges (e.g., Depression → Age).

2.6.2 Pruning the Causal Graph

The initial graph was manually pruned to remove nonsensical edges and ensure logical consistency:

- Removed edges like Depression → Age and Depression → Family History.
- Reversed edges such as Family History → Depression to reflect plausible causality.

- Simplified the graph by retaining only theoretically justified edges.

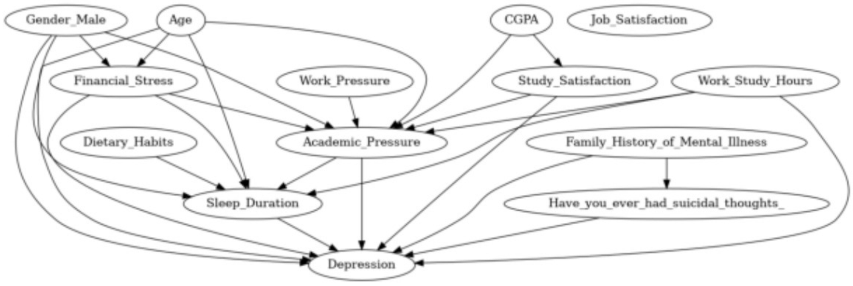


Figure 2.6: The manually pruned causal graph used for causal effect estimation.

The pruned graph served as the basis for causal effect estimation.

2.7 Model Development

2.7.1 Logistic Regression

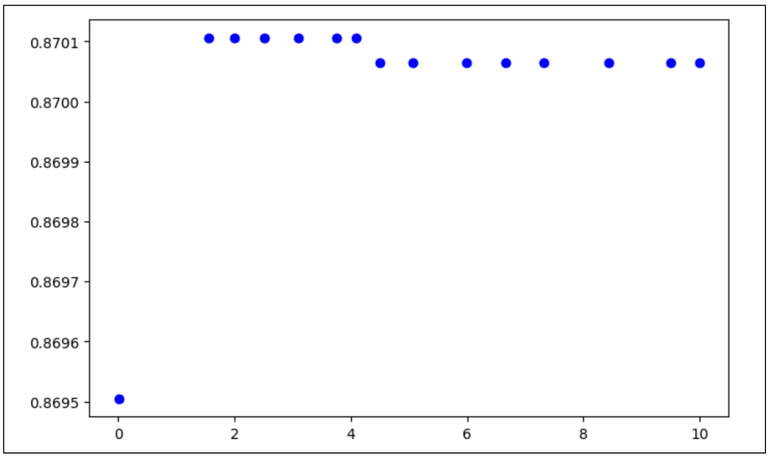
Logistic Regression is a linear model used for binary classification problems. It estimates the probability that a given input belongs to a particular class using the sigmoid function. Mathematical Formulation: The predicted probability that the output $y = 1$ given an input vector $\boldsymbol{x}$ is computed as:

$$P(y = 1 \mid \boldsymbol{x}) = \frac{1}{1 + e^{-(\boldsymbol{w}^\top \boldsymbol{x} + b)}} \tag{1}$$

Here, $\boldsymbol{w}$ is the weight vector, b is the bias term, and $\boldsymbol{w}^\top \boldsymbol{x}$ is the dot product between weights and features. **Decision Rule** The model assigns the class label based on the following rule (with t as threshold):

$$\hat{y} = \begin{cases} 0 & \text{if } P(y = 1 \mid \boldsymbol{x}) \leq t \\ 1 & \text{if } P(y = 1 \mid \boldsymbol{x}) > t \end{cases}$$

The below are the f1-scores for different c values while tuning using Bayesian optimization



Optimization Plot

C Value	F1 Score
3.7460	0.8701
9.5072	0.8701
7.3202	0.8701
5.9870	0.8701
1.5610	0.8701
0.0011	0.8695
4.0807	0.8701
2.5183	0.8701
8.4322	0.8701
5.0645	0.8701
9.9994	0.8701
1.9964	0.8701
3.1046	0.8701
6.6576	0.8701
4.5039	0.8701

Top C values and F1 scores

Figure 2.7: Logistic Regression optimization: visualization and results

2.7.2 Support Vector Machine (SVM)

Support Vector Machine (SVM) is a supervised learning algorithm used for classification. It finds the optimal hyperplane that maximizes the margin between two classes in a high-dimensional space. **Mathematical Formulation** For linearly separable data, SVM solves the following optimization problem:

$$\begin{aligned} \min_{\mathbf{w}, b} \quad & \frac{1}{2}\|\mathbf{w}\|^2 \\ \text{subject to} \quad & y_i \left(\mathbf{w}^\top \mathbf{x}_i + b \right) \geq 1, \quad \forall i = 1, \ldots, N \end{aligned}$$

Here:

- $\boldsymbol{w}$ is the weight vector perpendicular to the hyperplane.
- b is the bias.
- $y_i \in \{-1, +1\}$ are the true class labels.
- $\boldsymbol{x}_i$ are the feature vectors.

The objective is to maximize the margin, which is the distance between the hyperplane and the nearest data points from either class. **Kernel Trick** For non-linearly separable data, SVM uses kernel functions (e.g., RBF, polynomial) to project data into a higher-dimensional space where it becomes linearly separable.

2.7.3 Decision Trees

Decision Trees are tree-structured classifiers that recursively split the dataset based on feature values to maximize information gain. They are non-parametric and handle both classification and regression tasks. **Mathematical Formulation** One of the most common criteria for splitting is **Information Gain**, which is based on entropy. It measures the reduction in uncertainty about the target variable after splitting the dataset on a given feature.

$$IG(D,A) = Entropy(D) - \sum_{v \in Values(A)} \frac{|D_v|}{|D|} \cdot Entropy(D_v) \tag{2}$$

Where:

- D is the dataset,
- A is the attribute (feature),
- D_v is the subset of D where attribute A has value v,
- $Entropy(D)$ is given by:

  $$Entropy(D) = -\sum_{i=1}^{c} p_i \log_2 p_i$$

  with p_i being the proportion of class i in dataset D.

The tree grows recursively by selecting the feature that yields the highest information gain until a stopping criterion is met (e.g., maximum depth, minimum samples per leaf). We have experimented with different maxdepths ranging from 3-15 and minimum samples ranging from 2-20 using Bayesian Optimisation.

Table 2.1: Decision trees Parameters and F1 Scores

Max Depth	**Min Samples Split**	**F1 Score**
7.4945	19.1129	0.8572
11.7839	12.7759	0.8384
4.8722	4.8079	0.8556
3.6970	17.5912	0.8527
10.2134	14.7453	0.8490
6.9862	17.0609	0.8572
8.8446	2.0012	0.8540
11.4221	19.9816	0.8454
3.0144	2.0323	0.8550
3.0287	8.4337	0.8550
4.3891	12.6567	0.8527
8.0980	6.3170	0.8571
11.5586	5.1243	0.8334
5.9455	7.3614	0.8543
5.9385	2.0936	0.8543

2.7.4 Random Forest

Random Forest is an ensemble learning method that combines the predictions of multiple decision trees to improve accuracy and reduce overfitting. It operates using the technique known as **bagging**, where each tree is trained on a random subset of the data with replacement.

Mathematical Formulation: Given T decision trees $\{h_1, h_2, \ldots, h_T\}$, the final prediction $\hat{y}$ is made using a majority vote for classification:

$$\hat{y} = \text{majority vote of } \{h_1(x), h_2(x), \ldots, h_T(x)\} \tag{3}$$

Key aspects:

- Each tree is trained on a bootstrapped subset of the data.
- A random subset of features is considered at each split to ensure diversity.
- Aggregation of predictions increases stability and generalization.

N Estimators	**Max Depth**	**F1 Score**
287.6786	7.4945	0.8659
199.6646	11.7839	0.8670
88.9986	4.8722	0.8617
266.5440	3.6970	0.8603
227.0181	10.2134	0.8681
225.8352	9.7782	0.8681
170.8405	3.2097	0.8595
50.0808	14.5894	0.8624
213.4618	14.8356	0.8623
229.0850	3.0898	0.8596
225.5192	13.3620	0.8648
196.5468	9.0180	0.8679
194.1313	12.9960	0.8645
200.0551	6.6382	0.8659
193.5800	5.4257	0.8633

Table 2.2: Performance metrics for different configurations of n$_e stimatorsandmax_d epth$.

2.7.5 XGBoost

XGBoost (Extreme Gradient Boosting) is an efficient and scalable implementation of gradient boosted decision trees. It builds trees sequentially, where each tree tries to correct the errors of the previous one, and includes regularization to prevent overfitting. **Mathematical Formulation** The objective function combines a loss function and a regularization term:

$$Obj = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \tag{4}$$

Where:

- $l(y_i, \hat{y}_i)$ is a differentiable convex loss function (e.g., log loss for classification),
- $\Omega(f_k) = \gamma T + \frac{1}{2}\lambda\|w\|^2$ is the regularization term to penalize model complexity,
- f_k is the k^{th} decision tree,
- T is the number of leaves, and w are the leaf weights.

Key advantages of XGBoost include:

- Regularized objective function for better generalization,
- Column sub sampling,
- Fast and scalable parallel tree construction.

2.7.6 K-Nearest Neighbors (KNN)

Predicts based on nearest n samples:
Here $\hat{y}$ is the class to be predicted with the given data tuple

$$\hat{y} = \text{mode}(y_i \mid x_i \in \mathcal{N}_k(x))$$

We have experimented on the parameters of n ranging 1-29 and other parameters such as weights and metrics for getting the best model.

n neighbors	weight idx	metric idx	f1 score
28.5707	0.7320	0.3745	0.8637
5.5245	0.1560	0.5987	0.8522
26.1191	0.6011	0.0581	0.8643
1.5970	0.9699	0.7081	0.8125
7.1578	0.1818	0.8324	0.8584
19.4364	0.9636	0.9710	0.8642
13.9492	0.0442	0.1868	0.8613
22.5690	0.0611	0.0198	0.8627
17.1935	0.0511	0.0326	0.8641
10.9851	0.9944	0.0224	0.8583
15.8784	0.9767	0.9375	0.8622
29.9327	0.0070	0.8918	0.8663
24.4980	0.9809	0.9999	0.8656
26.2744	0.0006	0.9951	0.8649
29.9332	0.9535	0.9730	0.8652

Table 2.3: Bayesian Optimization Results for KNN

2.7.7 DBSCAN (Density-Based Spatial Clustering of Applications with Noise)

DBSCAN groups together points that are closely packed and marks points that lie alone in low-density regions as outliers.

Key Parameters:

- ε (eps): Maximum distance between two samples to be considered in the same neighborhood.
- MinPts: Minimum number of points required to form a dense region.

Core Concept:

- Points are classified as Core, Border, or Noise.
- Clusters are formed from connected core points.

No assumption of cluster shape or number of clusters.

eps	min samples	f1 score
1.1862	14.4086	0.7866
2.2228	10.1839	0.5854
0.5525	4.8719	0.5854
1.8432	11.4969	0.7801
1.2648	14.3890	0.7905
2.7816	12.0484	0.5854
2.0824	14.9926	0.5853
1.0192	11.0689	0.6972
2.4569	13.5836	0.5854
1.4288	3.2335	0.7800
2.2629	3.8349	0.5854
0.4556	3.0419	0.5918
2.0010	6.4656	0.7801

Table 2.4: DBSCAN Bayesian Optimization Results (filtered for f1 score ¿ 0)

2.7.8 K-Means Clustering

K-Means clustering is an unsupervised learning algorithm that partitions a dataset into K clusters by minimizing the variance within each cluster. Given a dataset $X = \{x_1, x_2, \ldots, x_N\}$, where each $x_i \in \mathbb{R}^d$, the algorithm seeks to divide the data into K disjoint subsets $C = \{C_1, C_2, \ldots, C_K\}$.

- Objective Function
  The goal of K-Means is to minimize the within-cluster sum of squares (WCSS), expressed by the objective function:
  $$J = \sum_{k=1}^{K} \sum_{x_i \in C_k} \|x_i - \mu_k\|^2$$
  Each centroid μ_k is computed as:
  $$\mu_k = \frac{1}{|C_k|} \sum_{x_i \in C_k} x_i$$
  where $|C_k|$ is the number of data points in cluster C_k.

2.7.9 Neural Networks

In this work, we employ a *Feedforward Neural Network (FFNN)* implemented using the `MLPClassifier` from `scikit-learn`. This model is a type of neural network composed of fully connected layers, where each neuron applies a weighted sum of its inputs followed by a nonlinear activation function such as `ReLU` or `tanh`.

The model is trained using *backpropagation* with stochastic gradient descent (SGD)-based optimizers, which adjust the weights to minimize the prediction error.

We experimented with several hyperparameters, including:

- *Hidden layer sizes* – different combinations of neurons across two hidden layers (e.g., (64, 32), (128, 64), etc.)
- *Learning rate* – controlling how fast the model adapts (e.g., 0.002, 0.01, 0.05)
- *Number of epochs (max_iter)* – defining the number of training iterations

First Layer	Second Layer	Learning Rate	F1 Score
64	32	0.002	0.8724
64	32	0.010	0.8740
64	32	0.050	0.8744
64	64	0.002	0.8756
64	64	0.010	0.8702
64	64	0.050	0.8715
64	128	0.002	0.8736
64	128	0.010	0.8706
64	128	0.050	0.8727
128	32	0.002	0.8737
128	32	0.010	0.8750
128	32	0.050	0.8715
128	64	0.002	0.8725
128	64	0.010	0.8758
128	64	0.050	0.8707
128	128	0.002	0.8766
128	128	0.010	0.8725
128	128	0.050	0.8751

Table 2.5: F1 scores of MLPClassifier for different ANN architectures and learning rates

2.7.10 Ensembling

Ensemble methods combine predictions from multiple machine learning models to produce a final output that is often more accurate and robust than any individual model. In this project, we tried to use model ensembling implemented using `VotingClassifier` to enhance the performance of classification for predicting student depression.

- **Hard Voting**: Takes the majority class predicted by each classifier.
- **Soft Voting**: Averages the predicted probabilities and chooses the class with the highest average probability.

We employed both voting strategies using models such as Logistic Regression, SVM, Decision Tree, Random Forest, XGBoost, KNN, and Neural Networks. **Mathematical Formulation**
Let $M_1, M_2, \ldots, M_n$ be n classifiers, and let $P_i(y = k|x)$ be the predicted probability that model M_i assigns to class k for input x.

- **Soft Voting:**

$$\hat{y} = \arg\max_k \left(\frac{1}{n} \sum_{i=1}^{n} P_i(y = k|x) \right)$$

- **Hard Voting:**

$$\hat{y} = \text{mode}\left(M_1(x), M_2(x), \ldots, M_n(x)\right)$$

Advantages of Ensembling

- Reduces variance (bagging), bias (boosting), or both.

- Improves generalization.
- Combines strengths of individual models.

Implementation Summary

We created an ensemble using the following base classifiers:

- Logistic Regression
- Decision Tree
- Random Forest
- XGBoost
- K-Nearest Neighbors
- FFNN with MLPClassifier
- TabNet

Predictions from these models were combined using both soft and hard voting. We performed 5-fold cross-validation and optimized hyperparameters using Bayesian Optimization for each model before ensembling.These are the shap values:

2.7.11 TabNet Classifier

TabNet is a deep learning architecture tailored specifically for tabular data, combining the interpretability of tree-based models with the representational capacity of neural networks. Unlike traditional feedforward networks, TabNet utilizes a novel feature selection mechanism through a sequential attention mechanism that allows the model to focus on the most relevant features at each decision step.

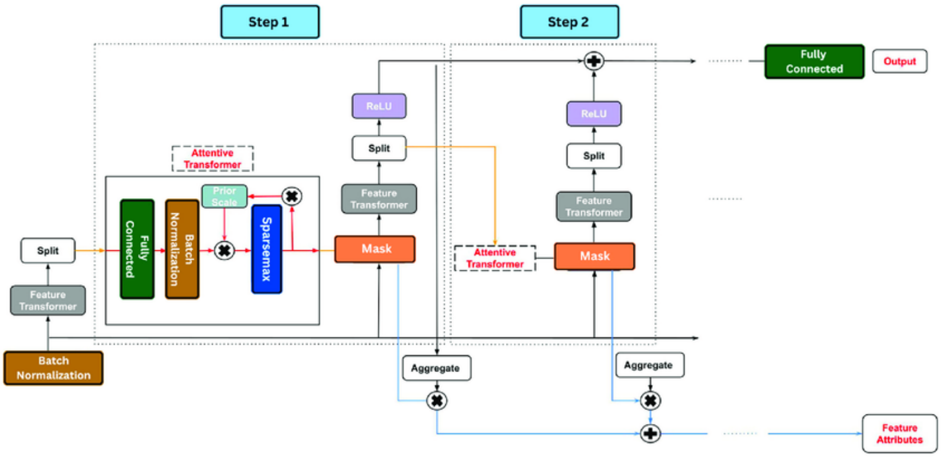


Figure 2.8: Overview of the TabNet Architecture

Key Hyperparameters

The performance of TabNet is significantly influenced by several hyperparameters:

- *n_d, n_a*: Dimensions of the decision and attention steps respectively. Larger values increase the model capacity.
- *n_steps*: Number of decision steps in the network. More steps allow deeper representation.

- *gamma*: Relaxation parameter controlling feature reusage between steps. Higher values encourage reuse of features.
- *lambda_sparsity*: Coefficient for sparsity loss which promotes feature selection.
- *learning_rate*: Determines the step size during gradient descent. Critical for convergence.
- *optimizer_params*: Parameters such as momentum or weight decay used by the optimizer.
- *mask_type*: Can be 'sparsemax' or 'entmax'. Controls the sparsity of attention.

In our implementation, we tuned these hyperparameters using cross-validation and achieved competitive results when compared to other models like XGBoost and ANN.

2.7.12 AutoML using FLAML

In this project, we utilize *FLAML (Fast Lightweight AutoML)*, a Python library designed for efficient and economical automated machine learning. FLAML performs hyperparameter tuning and model selection with a strong focus on time and resource constraints.
We configured FLAML to search across multiple tree-based model estimators, namely:

- *lgbm* – Light Gradient Boosting Machine
- *rf* – Random Forest
- *extra_tree* – Extremely Randomized Trees (ExtraTreesClassifier)

These estimators were automatically tuned by FLAML to optimize the F1-score using a time-aware search strategy.

2.7.13 LGBM as an Estimator

The *lgbm* estimator refers to the `LGBMClassifier` from the LightGBM library. It is a gradient boosting framework that uses tree-based learning algorithms. LGBM is known for its high performance, especially on large datasets, due to its efficiency and speed.
Within FLAML, LGBM was used as one of the candidate models during the AutoML run. FLAML automatically handled:

- *Hyperparameter tuning* – adjusting learning rate, number of leaves, max depth, and regularization parameters
- *Early stopping* – to avoid overfitting by terminating underperforming trials early
- *F1-score optimization* – models were evaluated based on their cross-validation F1-score

This integration allowed us to benefit from LGBM's predictive power while leveraging FLAML's efficient search to identify the best-performing configuration with minimal manual intervention.These are the shap values:

2.7.14 Stacking Ensemble with LightGBM as Meta-Learner

To further improve performance, we implemented a *stacking ensemble*, a powerful ensemble technique that combines multiple diverse models to leverage their individual strengths. In stacking, the predictions of several base models are used as features for a final meta-model which makes the ultimate prediction.

Our stacking framework consists of two levels:

- **Base-level models:**
  - Logistic Regression
  - Artificial Neural Network (ANN)
  - XGBoost Classifier
  - Random Forest Classifier
  - LightGBM Classifier
- **Meta-level model:**
  - LightGBM (LGBMClassifier)

Advantages

- *Improved generalization:* By combining different types of models, we reduce the risk of overfitting and improve robustness.
- *Leverages model diversity:* Each model captures different aspects of the data, and the meta-model learns how to best combine them.
- *LightGBM as meta-learner:* LightGBM, known for its efficiency and accuracy, performs well in aggregating predictions from heterogeneous models.

3 Experimental Results

3.1 Model Evaluation

3.1.1 Train Data Results

The mean cross validated F1 score of the models evaluated thus far are as follows:

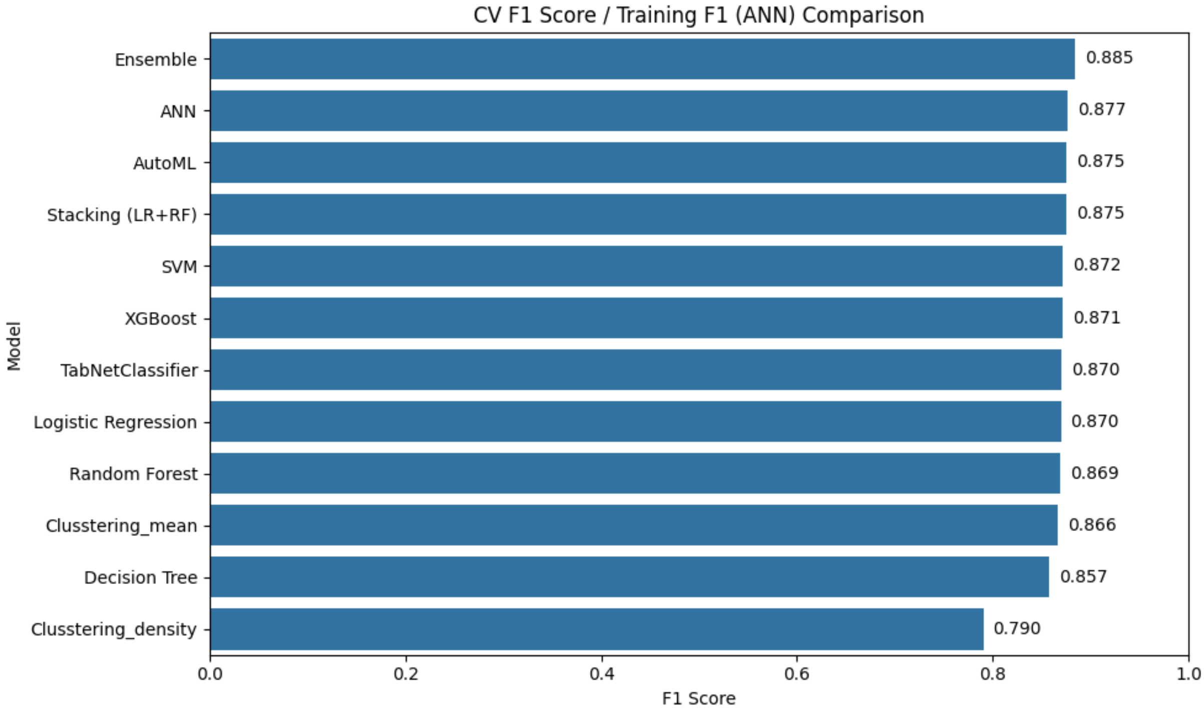


Figure 3.1: F1 score on training dataset.

3.1.2 Test Data Results

Post-training, the models were evaluated on the test set using a variety of metrics:

- *Accuracy, Precision, Recall, and F1-score:* Standard classification metrics to assess performance.
- *Confusion Matrix:* Visual representation of model predictions versus actual values.

Since we have considered F1 score to be our primary metric for evaluation and tuning of models, our best performing model was the *Stacking* based model discussed above.

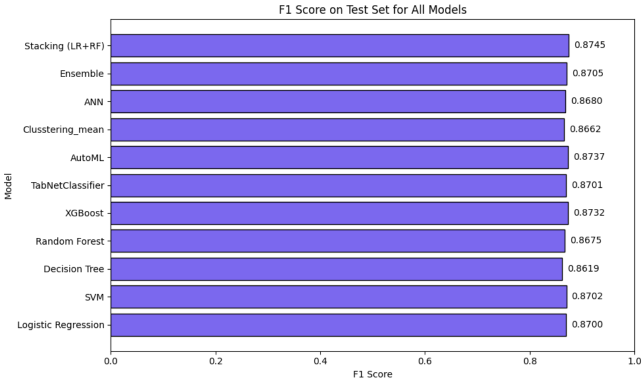


Figure 3.2: F1 Score

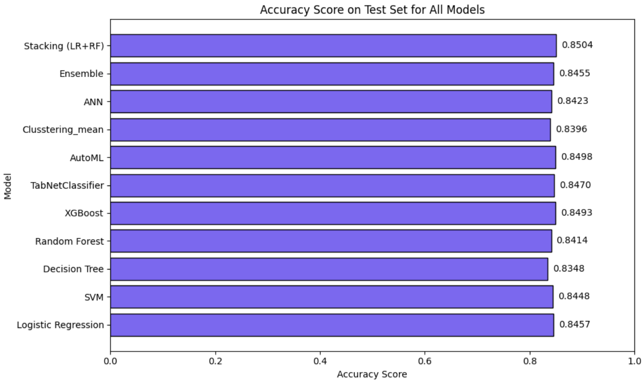


Figure 3.3: Accuracy Score

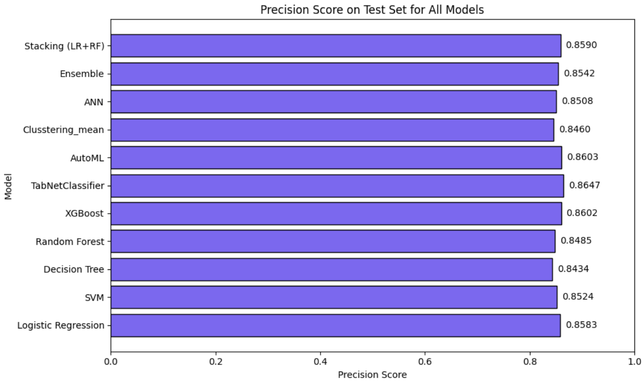


Figure 3.4: Precision Score

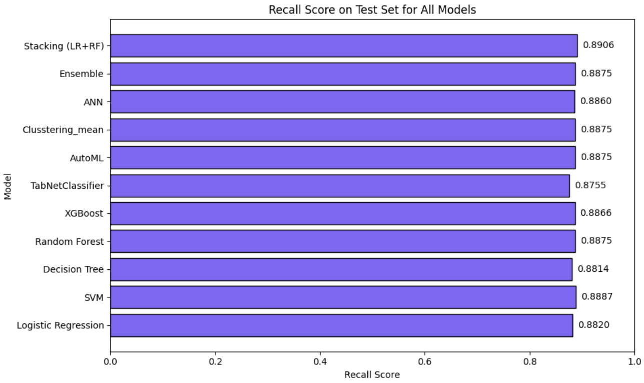


Figure 3.5: Recall Score

3.2 Causality Analysis Results

Using the pruned graph, the nonparametric average treatment effect (ATE) was estimated for each treatment via backdoor linear regression. The ATE represents the expected change in depression for a one-unit increase in the treatment variable. Conditional estimates were also computed to explore subgroup variations, and refutation tests were performed to validate the results. The estimated causal effects and refutation results are summarized in Table 3.1.

3.2.1 Analysis of Each Factor

- **Academic Pressure**
  - *Estimated Effect*: 0.1935

- *Interpretation*: A one-unit increase in academic pressure leads to a 0.1935-unit increase in depression.
- *Conditional Estimates*: The effect ranges from 0.1574 to 0.2028 across subgroups, indicating variability.
- *Refutation*: Robust (p-values: 0.98, 1.0).
- *Explanation*: Academic pressure contributes to depression by increasing stress and anxiety, common precursors to depressive symptoms.

- **CGPA**
  - *Estimated Effect*: 0.0105
  - *Interpretation*: A one-unit increase in CGPA results in a negligible 0.0105-unit increase in depression.
  - *Conditional Estimates*: Effects range from -0.0011 to 0.0237 but remain small.
  - *Refutation*: Robust (p-values: 0.92, 1.0).
  - *Explanation*: CGPA does not significantly contribute to depression, likely because its impact is mediated by other factors like academic pressure.

- **Sleep Duration**
  - *Estimated Effect*: -1.1e-16
  - *Interpretation*: No detectable causal effect on depression.
  - *Conditional Estimates*: Consistently 0.0 across subgroups
  - *Refutation*: Robust (p-values: 0.98).
  - *Explanation*: This lack of effect may reflect indirect influence or that it actually doesnt play that big of a role as we theorized.

- **Suicidal Thoughts**
  - *Estimated Effect*: 0.2666
  - *Interpretation*: Presence of suicidal thoughts increases depression by 0.2666 units.
  - *Conditional Estimates*: Not provided.
  - *Refutation*: Robust (p-values: 1.0).
  - *Explanation*: Suicidal thoughts both stem from and reinforce depression, creating a strong feedback loop.

- **Dietary Habits**
  - *Estimated Effect*: 0.1017
  - *Interpretation*: A one-unit improvement (A smaller value is a better dietary habit as we provided a smaller score for a better diet in ordinal encoding.) in dietary habits decreases depression by 0.1017 units.
  - *Conditional Estimates*: Effects range from 0.0673 to 0.1233.
  - *Refutation*: Robust (p-value: 0.97).
  - *Explanation*: This indicates the importance of good dietary choices in maintaining one's health.

- **Age**
  - *Estimated Effect*: -0.1102
  - *Interpretation*: Each additional year of age decreases depression by 0.1102 units.

- *Conditional Estimates*: Effects range from -0.1290 to -0.0911.
- *Refutation*: Robust (p-value: 0.98).
- *Explanation*: Age appears protective, possibly due to better coping strategies in older students.

- **Family History of Mental Illness**
  - *Estimated Effect*: 0.0263
  - *Interpretation*: Family history increases depression by 0.0263 units.
  - *Conditional Estimates*: Not provided.
  - *Refutation*: Robust (p-value: 0.86).
  - *Explanation*: Genetic and environmental factors slightly elevate risk, but current stressors play a larger role.

- **Study Satisfaction**
  - *Estimated Effect*: -0.0834
  - *Interpretation*: An increase of one unit in Study Satisfaction decreases depression by approximately 0.0834 units.
  - *Conditional Estimates*: Range from -0.1015 to -0.0453 across subgroups defined by suicidal thoughts, age, work pressure, work-study hours, gender, financial stress, dietary habits, and family history of mental illness.
  - *Refutation*: Robust (p-value: 0.92).
  - *Explanation*: Study Satisfaction has a small protective effect against depression, potentially reducing stress or enhancing a sense of achievement, with consistent effects across subgroups and robust refutation supporting its reliability.

- **Gender (Male)**
  - *Estimated Effect*: 0.0010
  - *Interpretation*: Being male increases depression by a negligible 0.0010 units compared to being female.
  - *Conditional Estimates*: Range from -0.0214 to 0.0188 across subgroups defined by suicidal thoughts, age, work pressure, work-study hours, study satisfaction, CGPA, dietary habits, and family history of mental illness.
  - *Refutation*: Robust (p-value: 0.96).
  - *Explanation*: Gender (being male) has no meaningful causal impact on depression, as the near-zero effect and robust refutation indicate minimal gender differences in depression risk in this context.

This analysis, based on a pruned graph from the PC algorithm and DoWhy's backdoor linear regression, identifies factors impacting depression among students. We evaluated individual effects of Academic Pressure, CGPA, Sleep Duration, Suicidal Thoughts, Dietary Habits, Age, Family History, Study Satisfaction, and Gender (Male), but did not analyze variable combinations, which could reveal interaction effects in future studies.

Treatment	Estimated Effect	Random Common Cause p-value
Academic Pressure	0.1935	0.98
CGPA	0.0105	0.92
Sleep Duration	0.0	0.98
Suicidal Thoughts	0.2666	1.0
Dietary Habits	0.1017	0.97
Age	-0.1102	0.98
Family History	0.0263	0.86
Sleep Duration	-1.1e-16	0.98
Gender-Male	0.0010	0.96

Table 3.1: Summary of Causal Effects on Depression

3.3 SHAP Analysis Results

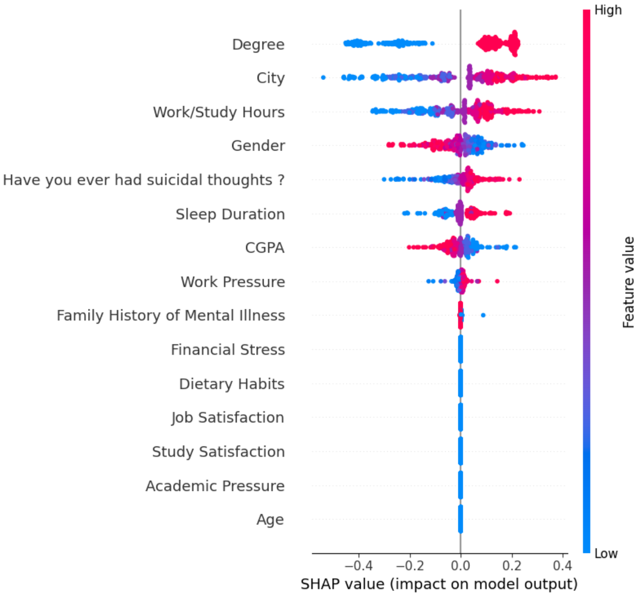


Figure 3.6: Logistic Regression shap values

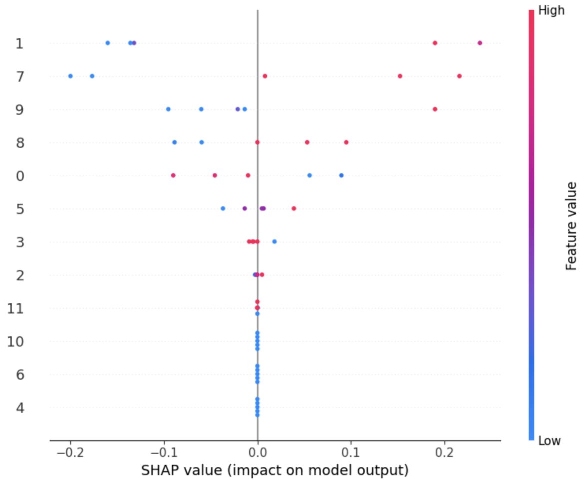


Figure 3.7: SVM shap values

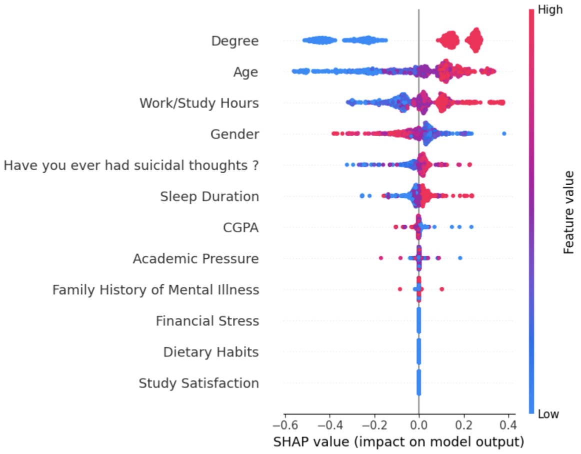


Figure 3.8: Decision Tree Shap

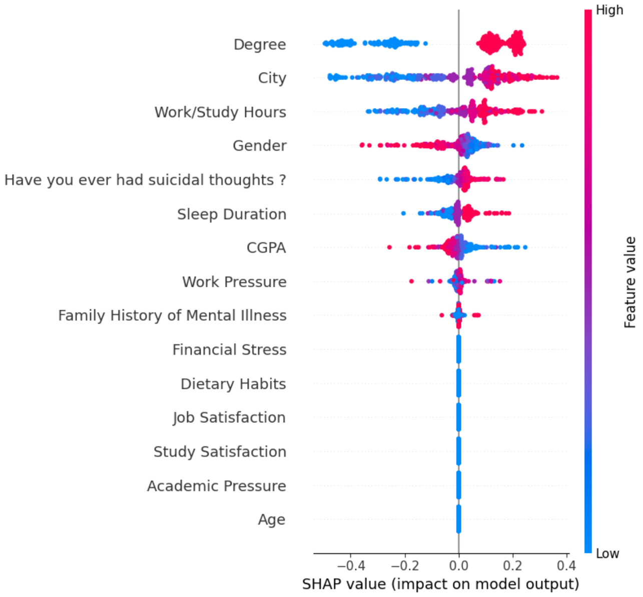


Figure 3.9: Random Forest Feature shap values

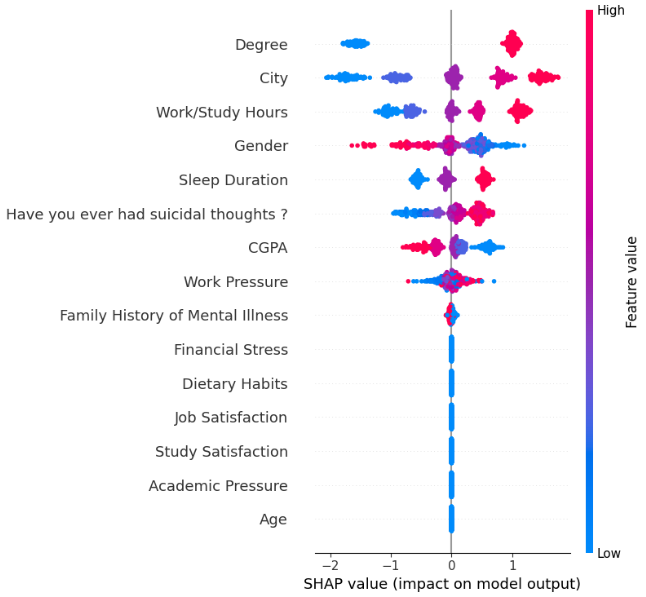


Figure 3.10: XGBoost SHAP Summary Plot

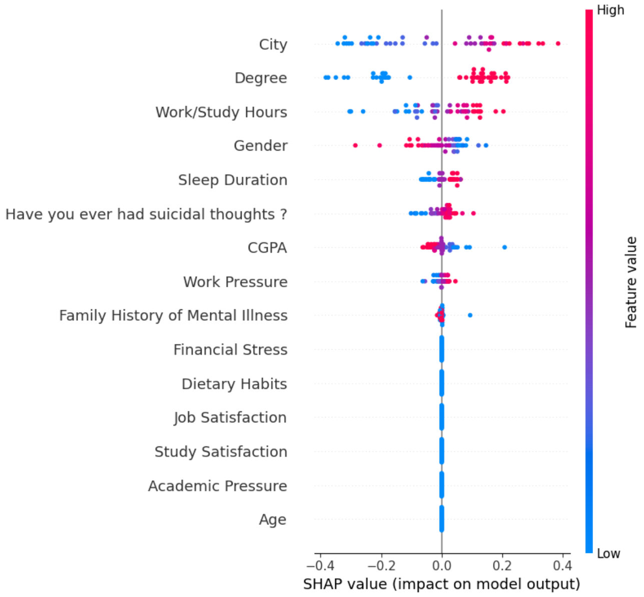


Figure 3.11: MLP shap values

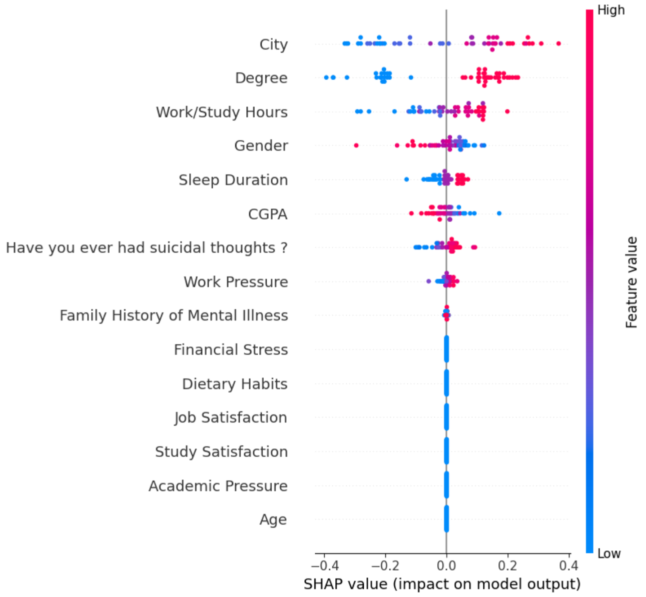


Figure 3.12: Ensembling shap values

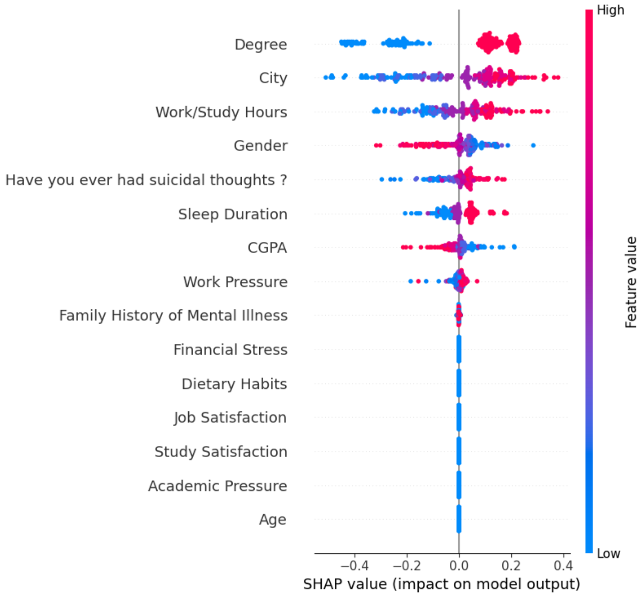


Figure 3.13: AutoML shap values

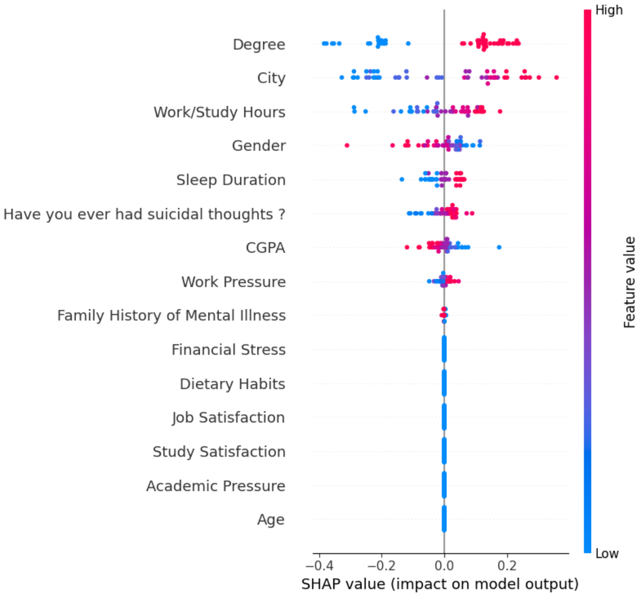


Figure 3.14: Stacking shap values

3.4 Comparing SHAP and ATE in Depression Risk Factors

Understanding the factors contributing to depression is vital for crafting effective interventions. We compare SHAP values from the best predictive model obtained (Stacking), with Average Treatment Effects (ATEs) from causality analysis we had done initially to differentiate between correlation and causation in depression risk factors. SHAP values show how features influence individual predictions, reflecting model learning, while ATEs estimate the actual causal impact of each feature on depression.

Predictive Model SHAP Values (F1 score: 0.874)

Age: 0.5685

Gender_Male: 0.0

City: 0.0830

Profession: 0.0

Academic_Pressure: -0.1031

Work_Pressure: 0.0357

CGPA: 0.0

Study_Satisfaction: 0.0

Job_Satisfaction: -0.1059

Sleep_Duration: 0.0

Dietary_Habits: 1.0974

Degree: 0.4134

Suicidal_Thoughts: -1.0933

Work_Study_Hours: 0.0

Financial_Stress: -0.0122

Interpretation: Positive SHAP values (e.g., *Dietary_Habits:* 1.0974) increase the model's prediction for depression, while negative values (e.g., *Suicidal_Thoughts:* -1.0933) decrease it. A value of 0 indicates no predictive contribution.

Causality Analysis ATEs

Age: -0.1102

Gender_Male: 0.0010

Academic_Pressure: 0.1935

CGPA: 0.0105

Study_Satisfaction: -0.0834

Sleep_Duration: 0.0

Dietary_Habits: 0.1017

Suicidal_Thoughts: 0.2666

Work_Study_Hours: 0.1023

Interpretation: Positive ATEs (e.g., *Suicidal_Thoughts:* 0.2666) indicate a causal increase in depression risk, while negative ATEs (e.g., *Age:* -0.1102) suggest a protective effect. A value of zero implies no causal impact.

Agreements and Disagreements Based on Magnitude of Effect

Agreements:

Sleep_Duration: SHAP = 0.0, ATE = 0.0 (no effect)

CGPA: SHAP = 0.0, ATE = 0.0105 (minimal effect)

Gender_Male: SHAP = 0.0, ATE = 0.0010 (negligible effect)

Disagreements:

Age: SHAP = 0.5685 (strong positive), ATE = -0.1102 (moderate negative)

Academic_Pressure: SHAP = -0.1031 (moderate negative), ATE = 0.1935 (strong positive)

Suicidal_Thoughts: SHAP = -1.0933 (strong negative), ATE = 0.2666 (strong positive)

Dietary_Habits: SHAP = 1.0974 (very strong positive), ATE = 0.1017 (moderate positive)

Study_Satisfaction: SHAP = 0.0 (no effect), ATE = -0.0834 (moderate negative)

Work_Study_Hours: SHAP = 0.0 (no effect), ATE = 0.1023 (moderate positive)

Discrepancies, especially in magnitude and direction (e.g., *Suicidal_Thoughts*), highlight differences between predictive and causal insights.

3.4.1 Why Feature Importance May Not Always Exhibit Real-World Causes and the Need for Causality Analysis

SHAP values reflect predictive patterns and feature importance, which are based on statistical correlations and associations within the dataset. However, these patterns do not necessarily equate to true causal relationships. For instance, the negative SHAP value for *Suicidal_Thoughts* contradicts its positive ATE, potentially due to confounding variables or encoding nuances. Relying solely on predictive models without a causal framework may lead to interventions based on misleading associations. Therefore, integrating causality analysis (like ATE estimation) is critical to identify true risk drivers and design effective, evidence-based interventions.

4 Conclusion

4.1 Key Findings on Depression Risk Factors

- Having **suicidal thoughts** emerged as the strongest predictor of depression, contributing the most to the overall risk.
- **Academic pressure** is another significant driver, indicating that the stress associated with academic demands may contribute meaningfully to depressive symptoms.
- **Work-study hours** show a moderate association with depression, suggesting that excessive workload or poor time balance may impact mental well-being.
- Protective factors include **age** with older individuals showing a reduced likelihood of depression, and study satisfaction, where higher satisfaction levels are linked with lower risk. These findings point to the importance of emotional maturity and perceived academic fulfillment in mental health outcomes.
- Some variables, such as **sleep duration, CGPA**, and **gender**, demonstrated minimal or no clear impact on depression in our model. This suggests that these factors are less predictive in this context or not well captured by the dataset.
-

4.2 How to Reduce Depression

Using the insights derived from our analysis of both SHAP and ATE, we suggest a few methods to reduce depression in school setting:

Academic Pressure: Implement strategies to reduce workload and enhance support systems and educate parents on their childern's mental wellbeing.

Suicidal Thoughts: Increase access to counseling and therapy services.

Work-Study Hours: Encourage students to engage in sports along with studies.

Study Satisfaction: Improve educational environments, teaching styles and student engagement through hands on learning.

Dietary Habits: Eventhough we are not able to absolutely show dietary habits play a role, having a good diet helps to maintain a healhy body which in turn leads to a healthy body.

4.3 Ethical Considerations and Real-World Implications

4.3.1 Role of Predictive Modeling in Designing Mental Health Support Initiatives

Predictive modeling, when combined with tools like SHAP, can play a significant role in shaping mental health support:

Early Identification: Predictive models enable educational institutions to identify students who are at a heightened risk of depression. By analyzing individual-level predictors, institutions can flag potential issues before they escalate.

Personalized Interventions: With insights from SHAP values, tailored mental health support programs can be developed. For instance, if a model highlights high dietary risks or academic pressure, schools can design specific support programs (e.g., nutritional counseling or academic counseling) targeted at those areas.

4.3.2 Informed Consent and Data Protection

The use of predictive modeling in mental health, especially within sensitive environments like educational institutions, must address several ethical issues:

Informed Consent:

– It is crucial that students (or their guardians) are fully informed about the nature of the data being collected.

– Clear communication regarding how the data will be used—whether for early intervention, academic research, or service improvement—ensures that consent is truly informed.

– Consent procedures should provide options for students to opt out of data collection without penalty.

Data Protection:

– The sensitive nature of mental health data necessitates robust safeguards to prevent unauthorized access or misuse.

– Institutions must implement strict data governance policies that cover storage, processing, and sharing of information.

– Anonymization and secure encryption practices should be standard to protect individual privacy.

Avoiding Misguided Interventions:

– Over-reliance on SHAP values from predictive models without corroborating causal insights may lead to interventions that target non-causal factors, potentially misallocating resources or inadvertently worsening outcomes.

– A balanced approach, integrating both predictive and causal analyses, minimizes the risk of ethical oversights while maximizing intervention efficacy.

Differential Privacy:

– Differential privacy ensures that insights drawn from data do not compromise the privacy of individuals, even in aggregate analyses.

– By adding mathematically calibrated noise to datasets or query results, this method limits the risk of re-identifying individuals from statistical outputs.

Machine Unlearning:

- Machine unlearning allows for the selective removal of a user's data from trained models, ensuring that withdrawn consent is respected even post-processing.
- This approach is vital in upholding the "right to be forgotten" and enhancing trust in AI systems by providing control over personal data.
- Integrating unlearning mechanisms supports dynamic consent models where students can revoke participation without long-term data retention consequences.

4.3.3 Real-World Implications

The integration of predictive modeling and causal analysis has wide-ranging implications for the real-world application of mental health support:

Improving Student Outcomes: By precisely targeting the actual causal drivers of depression, educational institutions can implement more effective support systems that lead to improved mental health and academic performance.

Policy and Program Development: Ethical, data-driven insights can inform policy-level decisions and promote the design of comprehensive mental health programs tailored to the unique needs of student populations.

Trust and Transparency: Maintaining high ethical standards in informed consent and data protection fosters trust among students, parents, and other stakeholders, which is essential for the sustainable adoption of these technologies.

Scalability and Adaptability: With validated models that accurately capture both predictive correlations and causal relationships, similar approaches can be scaled to different educational settings or adapted to address other mental health challenges.

4.4 Future Improvements

- Apply differentially private algorithms to create predictive models which ensures privacy of participants, thereby reducing hesitancy in people when it comes to sharing personal info..
- Obtain a much diverse dataset to train on and derive insights from.
- Perform indepth multivariate analysis(only single variable has been done) on data to improve explainability and give a deeper insight on which factors affect depression, allowing actionable insights to be derived.
- Apply advanced NN - based algorithms like SAINT, AutoNet, GANDALF, TabTransformer which promises better accuracies compared to traditional tree based models. Also different combinations of ensembling can be experimented .