# Predicting Depression Risk Using Machine Learning: An XGBoost Approach

Kodithuwakku Kisara, Uthayasanker Thayasivam [1]

## Abstract

In this study, we use machine learning to predict the depression risk based on factors like lifestyle and demographic. We analyze a dataset with a mixture of categorical and numerical features, like age, work/ study hours, and family history of mental illness. The aim is to build a model which predict if someone is at risk of depression. Our approach include data cleaning, feature engineering, and using the XGBoost algorithm. The model achieved a good performance, with Accuracy score of 0.9381 and F1 score of 0.8252, showing that it has potential to help identify people at risk of depression.

## 1 Introduction

Depression is a common mental health problem that can have a big impact on people's lives. Early detection of those at risk is important so they can get the help they need. Machine learning offers a way to predict depression risk by analyzing data about a person's lifestyle and background. In this study, we use a dataset that includes information like age, work/study hours, occupation, work satisfaction, stress, and sleep patterns to predict whether someone is at risk of depression. Our goal is to build a model that can accurately make this prediction. We follow a step-by-step process, including exploring the data, cleaning it, creating new features, and training a machine learning model. The results show that our model works well, making it a useful tool for identifying people who might need support.

# 2 Methodology

## 2.1 Overview of the Dataset

The dataset contains a combination of categorical and numerical features that are intended to help identify how everyday factors could be linked to mental health risks, making it an invaluable resource for developing machine learning models aimed at predicting mental health outcomes [2].

The dataset consists of the following features.

**Numerical Features**:

- Age
- Work/Study Hours: Number of hours spent working or studying.
- CGPA: Cumulative Grade Point Average.

**Ordinal Features**:

- Academic Pressure
- Study Satisfaction
- Work Pressure
- Job Satisfaction
- Financial Stress

**Categorical Features**:

- Gender
- Working Professional or Student
- Dietary Habits
- Have you ever had suicidal thoughts?
- Family History of Mental Illness
- Sleep Duration (Later converted to an ordinal feature)

**Categorical Features (High Cardinality)**:

- Name
- City: City of residence
- Profession: Professional occupation
- Degree: Educational degree held by the individual

**Target Variable (Binary)**:

- Depression: Indicates whether the individual experiences depression

## 2.2 Data Preprocessing and Feature Engineering

To begin, the presence of missing values in the dataset was assessed with function df.isna().sum(). This step is important for understanding how much missing data there is, as it can affect the accuracy and effectiveness of later analysis and modeling work.

The dataset have missing values in some fields related to students and working professionals. These missing values was explored separately for each category to make sure they were handled properly based on their context.

Several feature engineering methods were considered to improve model performance, such as making composite features like "Pressure" and "Satisfaction" as features common to Students and Working professionals, and using imputation methods like mean/mode and KNN imputation. However, none of these methods result in big improvements in model accuracy.

For ordinal features and the "CGPA" column, constant imputation with -1 value was chosen. This simple method make sure that missing values are dealt with without disturbing the analysis or modeling process.

High cardinality categorical features, such as "Profession", "Degree", "Dietary Habits", and "City", created more challenges. For these, missing values were filled with "NA". Also, categories with less than 50 occurrences were grouped into an "Other" category to reduce noise and help model performance.

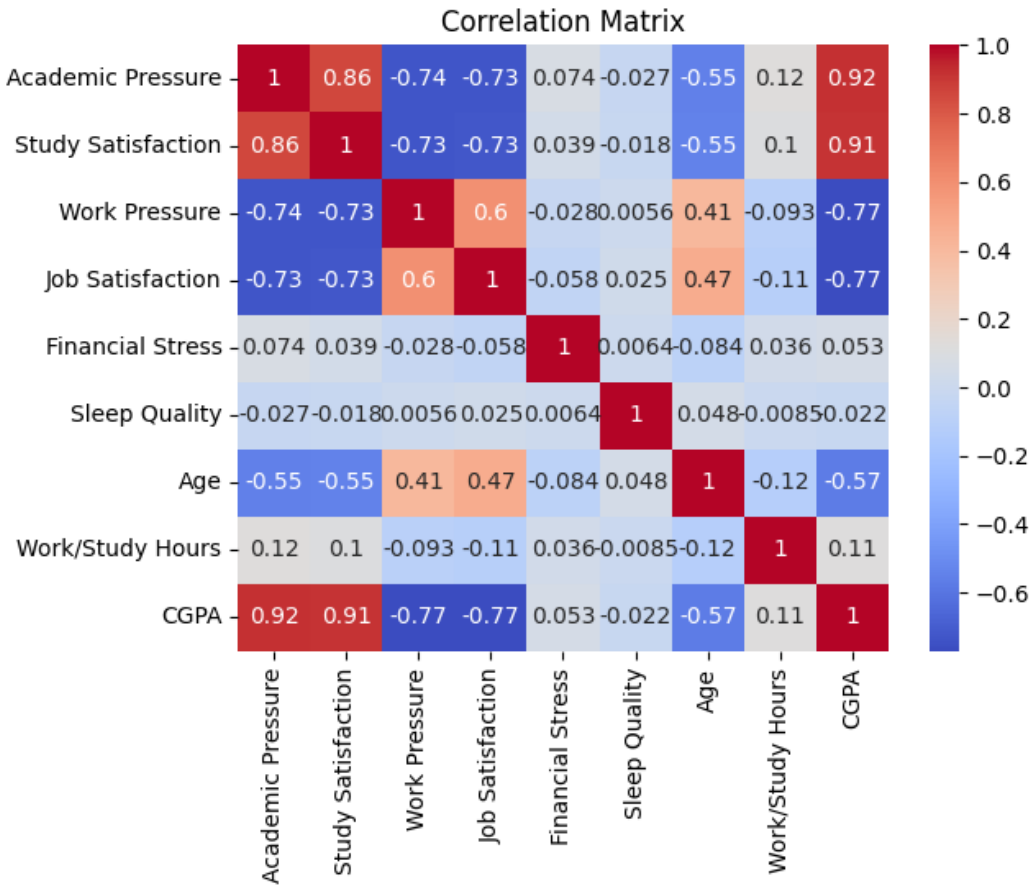
The "Sleep Duration" column was mapped to an engineered feature "Sleep Quality" scale by a predefined mapping. This conversion turned categorical sleep duration values into a numerical scale, making it easier for the model to interpret.

In the end, columns considered not relevant to the analysis, like "Name" and "Sleep Duration", were removed from the dataset. This simplification helped focus the analysis on the most important features, making the modeling process more efficient.

## 2.3 Exploratory Data Analysis

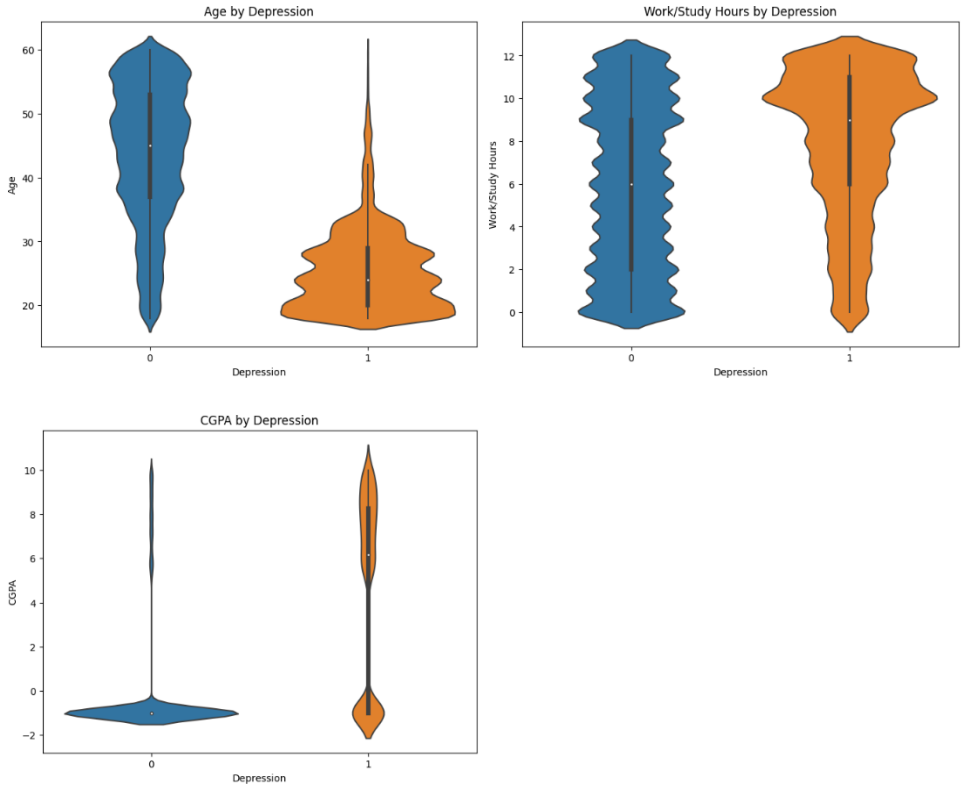
### Correlation Analysis

A correlation matrix is computed for numerical features to identify relationship between them. This help in understand how features interact with each other and can guide feature selection or engineering.

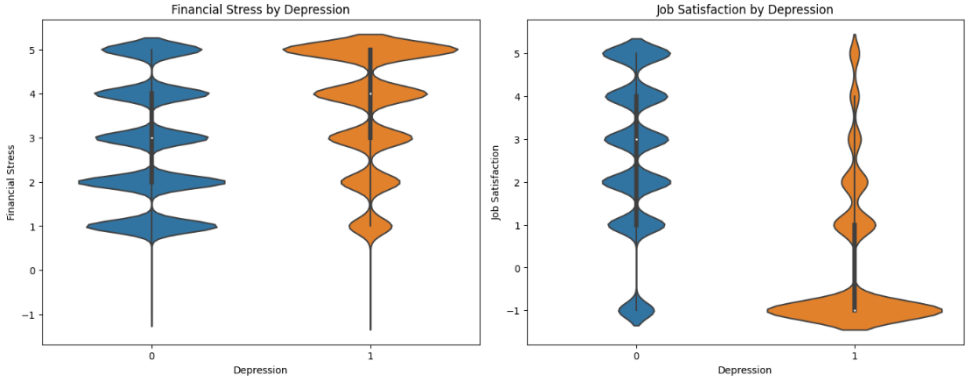


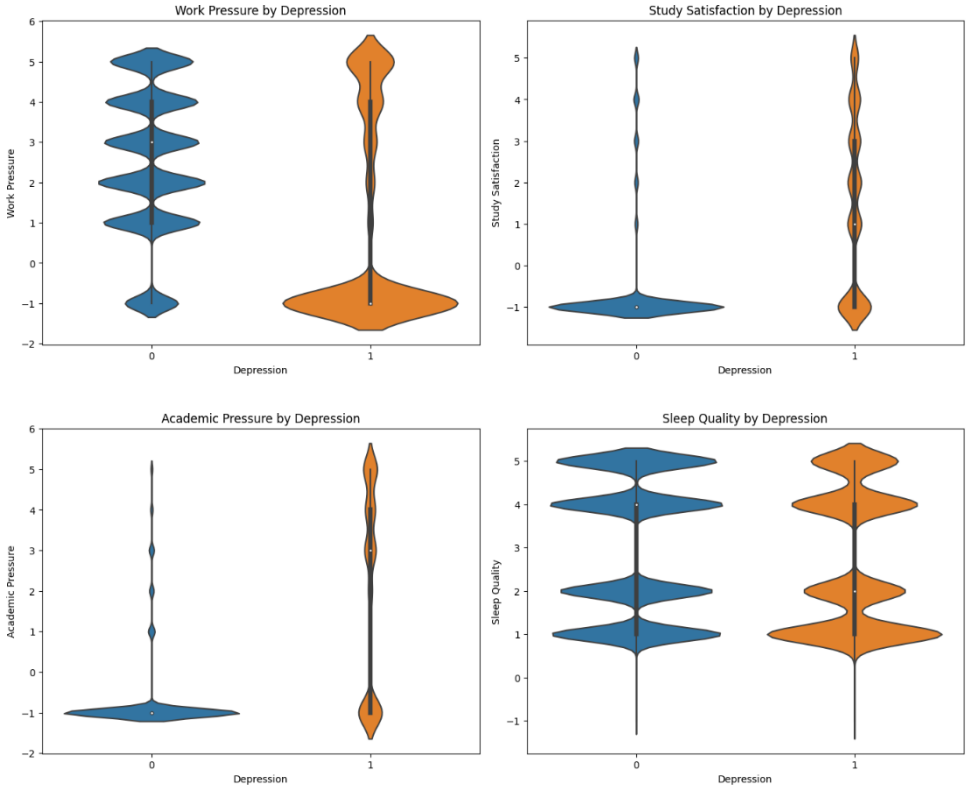
### Violin Plots for Numerical Features

Violin plots is used to visualize the distribution of numerical and ordinal features against the target variable (Depression). These plots provide insights into how the distributions differ between the two target classes, which can be useful for identifying patterns or anomalies.



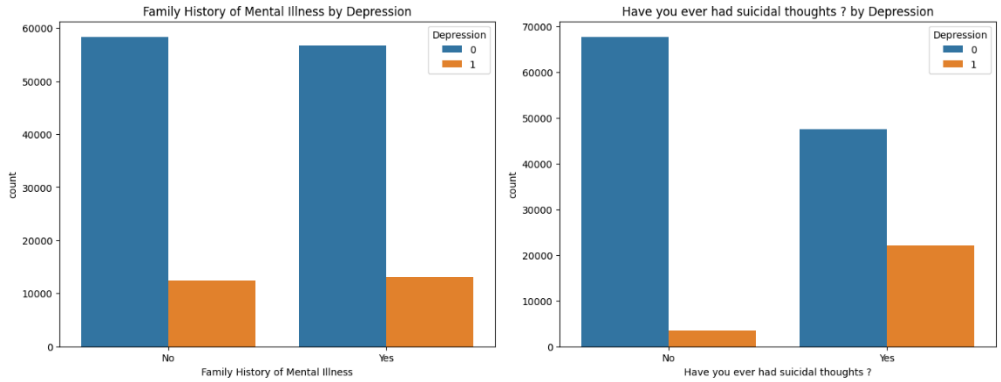

### Violin Plots for Ordinal Features

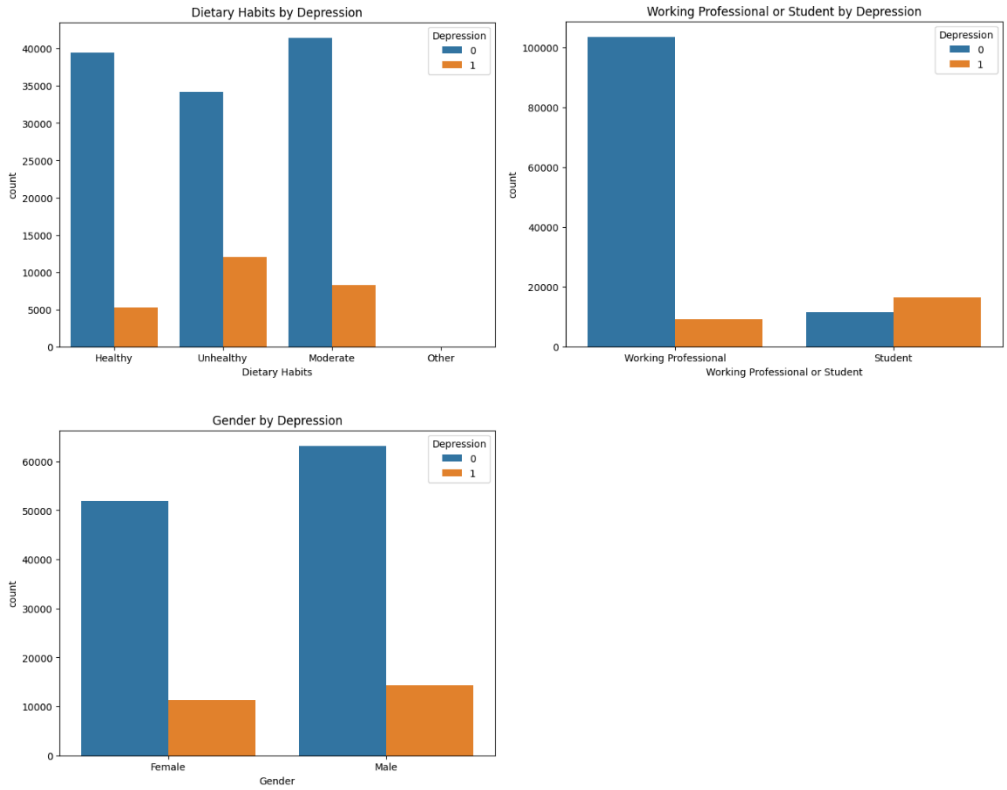




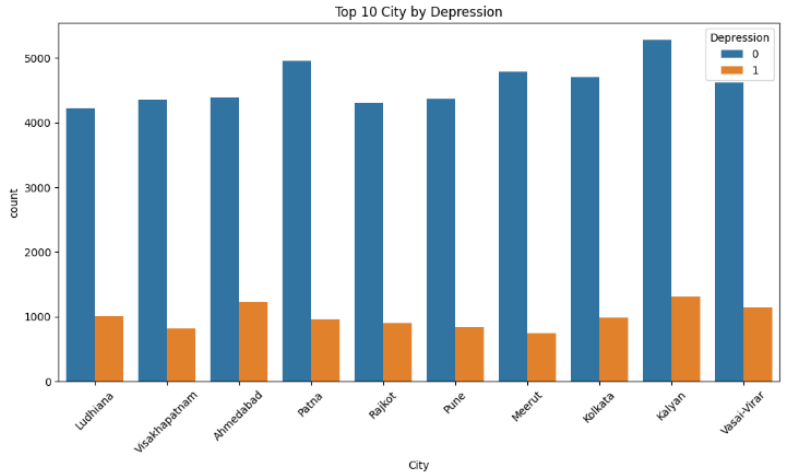

### Count Plots for Categorical Features

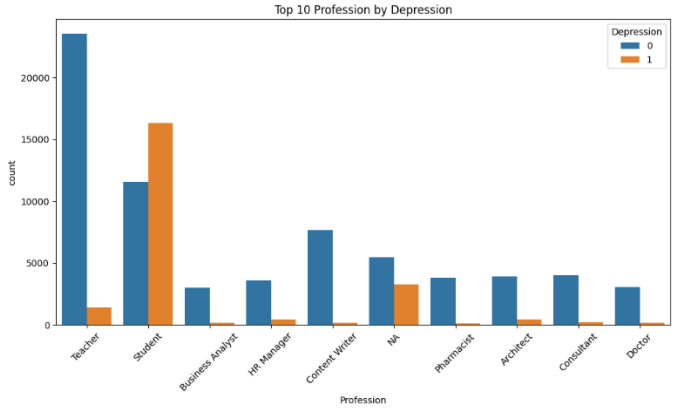
Count plots are used to visualize distribution of categorical features with respect to the target variable. This help in understanding frequency of each category and its association with the target.

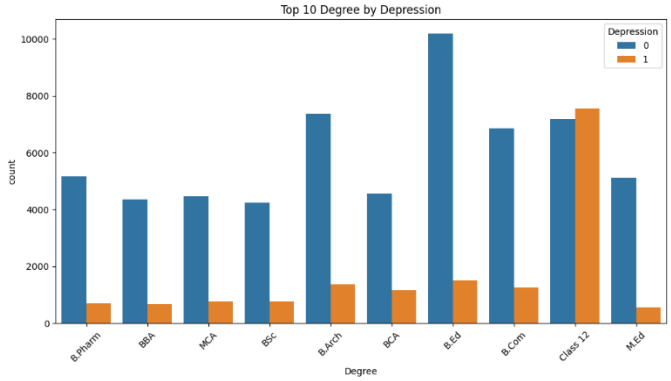





## Count Plots top-N for High Cardinality Features






## 2.4 Feature Encoding

To prepare data for modeling, feature encoding was applied to transform both categorical and numerical features into formats that are compatible with machine learning algorithms. This process involved many steps key.

First, numerical features like 'Age', 'Work/Study Hours', and 'CGPA' was standardized with StandardScaler. This ensure that all numerical features have mean of 0 and standard deviation of 1, which prevent features with larger magnitude from dominating the model.

Next, ordinal features, such as 'Academic Pressure' and 'Sleep Time', were encoded with OrdinalEncoder. This encoding keep the order of categories, which allow the model to interpret the relationship between these features correctly.

For low-cardinality categorical features, like 'Gender' and 'Dietary Habits', one-hot encoding was used. This method create binary columns for each category, which make sure the model can process the categorical data without assuming any ordinal relationship.

High-cardinality categorical features, like 'City' and 'Profession', were target encoded using TargetEncoder. This technique reduces dimensionality while keeping the relationship between the feature and target variable, improving model performance.

Finally, to maintain consistency between the training and test datasets, any missing columns in the test set were filled with zero, and columns were reordered to match those in training set. This step make sure that the model process data uniformly during evaluation.

## 2.5 Model Selection and Evaluation

The XGBoost algorithm was choosen for its ability to handle mixed data types and provide robust performance. The model was trained and evaluated through several steps key to ensure reliability and effectiveness.

To start, stratified K-fold cross-validation was applied, splitting the dataset into five folds. This method ensure that the proportion of target variable was maintained across all folds, which give a reliable estimate of model's performance and help to prevent overfitting.

Hyperparameter tuning was done to optimize model configuration. The following hyperparameters were chosen: n_estimators=1000 for the number of boosting rounds, learning_rate=0.01 for the step size shrinkage to avoid overfitting, max_depth=5 for the maximum depth of each tree, min_child_weight=3 for minimum sum of instance weights in a child, subsample=0.8 for fraction of samples used for training each tree, colsample_bytree=0.8 for fraction of features used, and gamma=0.1 for minimum loss reduction required to make a split.

The model was trained with early stopping, which monitor validation performance and stopped training if no improvement was seen for 50 rounds. This helped prevent overfitting and make sure model was optimized.

During cross-validation, the model got a mean AUC-ROC score of 0.9754, showing strong performance in distinguishing between individuals at risk of depression and those who are not. Finally, the model was trained on entire training dataset to maximize data usage, and this trained model was used for make predictions on the test dataset.

## 3 Results

The model's performance was evaluated on the test set using several key metrics. The AUC-ROC score was 0.9745, showing the model's ability to effectively distinguish between the two classes. The accuracy of model was 0.9381, meaning it correctly classified 93.81% of test instances. Also, the F1-score was 0.8252, which reflect the model's ability to balance precision and recall, especially in the context of imbalanced dataset.

To make prediction on the final test dataset, the model was retrained on entire training dataset. This approach made sure that model use all available data in the best way. The final predictions were then generated, with an accuracy score of 0.94005.

## 4 Conclusion

In conclusion, the feature encoding, model selection, and evaluation process showed the effectiveness of XGBoost algorithm in predicting depression risk. With an AUC-ROC score of 0.9745 and accuracy of 0.9381, the model performed strongly, highlighting the potential of machine learning to identify individuals at risk of depression. These results give a strong foundation for future research and applications in field of mental health.

## 5 References


[1] *Dr. Uthayasanker Thayasivam, PhD (U. Georgia), BSc Eng. (Hons) (Moratuwa).*

[2] "Predicting Depression: Machine Learning Challenge," [Online]. Available: https://www.kaggle.com/competitions/predicting-depression-machine-learning-challenge.