**CHAPTER IV**

**METHODOLOGY**

# **4.1 Introduction**

This chapter outlines the methodology used in analyzing student depression using **machine learning (ML) techniques**. It describes the dataset, preprocessing steps, feature selection, model implementation, and evaluation methods. The goal is to develop an **effective prediction model** that identifies students at risk of depression.

# **4.2 Dataset Description**

The dataset used in this study contains various attributes related to students, including **demographic details, academic performance, psychological factors, and behavioral traits**. The key attributes in the dataset include:

* **Demographic Information**: Age, Gender, Family Background
* **Academic Performance**: Grades, Attendance, Study Hours
* **Psychological Factors**: Stress Levels, Anxiety, Sleep Patterns
* **Behavioral Aspects**: Social Media Usage, Exercise Habits, Eating Patterns
* **Depression Indicators**: Self-reported Depression Score (if available)

The dataset will be analyzed to identify patterns and correlations between these factors and depression symptoms.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Feature Name | Data Type | Min | Max | Description |
| Id | Integer | 2 | 140699 | Unique identifier for each student |
| Gender | Categorical | - | - | Student's gender (e.g., Male, Female, Other) |
| Age | Float | 18 | 59 | Student's age in years |
| City | Categorical | - | - | City where the student resides |
| Profession | Categorical | - | - | Student's field of study/work |
| Academic Pressure | Float | 0 | 5 | Level of academic stress (0 = None, 5 = High) |
| Work Pressure | Float | 0 | 5 | Level of work stress (0 = None, 5 = High) |
| CGPA | Float | 0 | 10 | Student's cumulative GPA (0-10 scale) |
| Study Satisfaction | Float | 0 | 5 | Satisfaction with studies (0 = Not at all, 5 = Very satisfied) |
| Job Satisfaction | Float | 0 | 4 | Satisfaction with job (if applicable) (0 = Not at all, 4 = Very satisfied) |
| Sleep Duration | Categorical | - | - | Average hours of sleep per day (e.g., Less than 4, 4-6, 6-8, More than 8) |
| Dietary Habits | Categorical | - | - | Eating habits (e.g., Healthy, Unhealthy, Mixed) |
| Degree | Categorical | - | - | Educational qualification (e.g., Bachelor's, Master's, PhD) |
| Have you ever had suicidal thoughts? | Categorical | - | - | Yes/No response to suicidal thoughts |
| Work/Study Hours | Float | 0 | 12 | Average hours spent studying/working per day |
| Financial Stress | Float | 1 | 5 | Level of financial stress (1 = None, 5 = High) |
| Family History of Mental Illness | Categorical | - | - | Whether there is a family history of mental health issues (Yes/No) |
| Depression | Integer | 0 | 1 | Depression label (0 = No, 1 = Yes, used as target variable for ML) |

# **4.3 Data Preprocessing**

Preprocessing is a crucial step in preparing the dataset for machine learning models. The following steps are applied:

1. **Handling Missing Values**: Missing data is imputed using mean, median, or mode, depending on the attribute type.
2. **Data Cleaning**: Removing outliers and inconsistencies in responses.
3. **Encoding Categorical Variables**: Converting categorical features (e.g., Gender, Sleep Quality) into numerical representations using **one-hot encoding** or **label encoding**.
4. **Feature Scaling**: Normalizing numerical data using **Min-Max Scaling** or **Standardization** to ensure uniformity in ML model input.
5. **Balancing the Dataset**: Using techniques like **SMOTE (Synthetic Minority Over-sampling Technique)** if the dataset is imbalanced (e.g., significantly more non-depressed than depressed students).

# **4.4 Feature Selection**

To improve model efficiency and accuracy, feature selection is performed using:

**1, Correlation Analysis:**

Correlation analysis is a **statistical technique** used to measure the **strength and direction of the relationship** between two variables. In the context of depression analysis, it helps identify the features that are **strongly associated** with depression levels. By examining the correlation coefficients, we can determine which **factors (features)**—such as sleep duration, academic stress, financial issues, or dietary habits—are significantly related to depression.

In this analysis, **Pearson’s correlation coefficient** is commonly used for **continuous variables**, while **Spearman’s or Kendall’s correlation** is applied for **ordinal or non-normally distributed variables**. The correlation coefficient ranges between **-1 and 1**:

* +1+1+1 → **Perfect positive correlation**: As one variable increases, the other increases.
* −1-1−1 → **Perfect negative correlation**: As one variable increases, the other decreases.
* 000 → **No correlation**: No linear relationship between the variables.

In depression-related datasets, features such as **sleep duration, financial stress, and academic pressure** are often found to have a **positive or negative correlation** with depression scores. For instance:

* **Positive correlation**: Higher financial stress or academic pressure is associated with higher depression scores.
* **Negative correlation**: Increased sleep duration or physical activity might be linked to lower depression levels.

By identifying highly correlated features, you can select the most relevant variables for **predictive modeling** (e.g., logistic regression) and remove redundant or less informative features. Highly correlated independent variables (multicollinearity) can distort model predictions, making correlation analysis essential for **feature selection and dimensionality reduction**.

**2, Principal Component Analysis (PCA):**

**Principal Component Analysis (PCA)** is a **dimensionality reduction technique** used to transform a large set of correlated variables into a **smaller set of uncorrelated variables** called **principal components (PCs)**. The goal is to **preserve as much of the original data’s variance** as possible while reducing the complexity of the dataset. This is particularly useful when dealing with **high-dimensional datasets**, such as depression-related datasets, where many features might contain redundant or correlated information.

**1, How PCA Works**

PCA performs the following steps:

1. **Standardization:** The data is first standardized by **scaling** the features to have a mean of **0** and a standard deviation of **1**. This ensures that features with larger magnitudes do not dominate the analysis.
2. **Covariance Matrix Calculation:** PCA computes the **covariance matrix** to measure how the variables are related to each other.
3. **Eigen Decomposition:** It calculates the **eigenvalues** and **eigenvectors** of the covariance matrix.
   * **Eigenvalues** → Indicate the amount of variance explained by each principal component.
   * **Eigenvectors** → Represent the directions (principal components) of maximum variance.
4. **Principal Components Formation:** The data is projected onto the new **orthogonal axes** (principal components).
5. **Variance Retention:** The **first few components** capture most of the variance in the data. By selecting the top kkk components, we reduce the dimensionality while retaining most of the essential information.

**2. Benefits of PCA in Depression Analysis**

In the context of **depression analysis**, PCA helps by:

* **Reducing noise and redundancy**: Depression datasets often contain correlated features (e.g., sleep quality, sleep duration, and fatigue levels). PCA merges these into fewer dimensions, reducing redundancy.
* **Improving model efficiency**: With fewer features, machine learning models (e.g., logistic regression, random forest) become **faster** and less prone to overfitting.
* **Visualizing complex data**: PCA enables **2D or 3D visualizations** of the data, making it easier to spot clusters or patterns related to depression.

**3. Interpretation of PCA Output**

* **Explained Variance Ratio:** This indicates the proportion of the total variance captured by each principal component.
  + For example, if **PC1** captures 60% of the variance and **PC2** captures 25%, together they explain **85%** of the variance.
* **Principal Component Scores:** The transformed dataset contains the scores (projections) of the original data on the new principal component axes.
* **Loading Scores:** These indicate how much each original feature contributes to a given principal component.

**4. PCA vs. Feature Selection**

While **feature selection** removes irrelevant features, PCA **combines features** into new dimensions. This makes PCA ideal when the goal is to **compress data** while retaining as much information as possible, even if individual features lose interpretability.

**5. Practical Applications of PCA in Depression Analysis**

* **Reducing survey data**: Depression surveys often contain numerous questions measuring similar constructs (e.g., anxiety, sleep patterns). PCA can reduce these into fewer dimensions.
* **Identifying underlying factors**: PCA reveals **latent factors** contributing to depression by combining correlated features into meaningful components.
* **Pre-processing step for modeling**: Applying PCA before running models (e.g., logistic regression, SVM) can reduce multicollinearity and enhance performance.

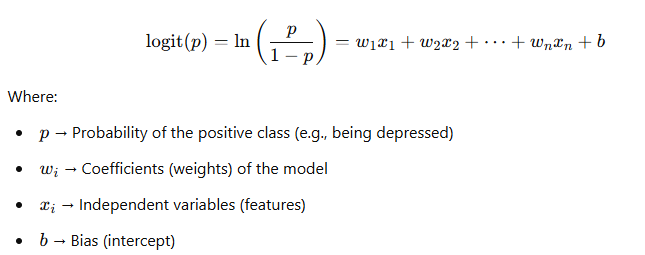
# **4.5 Machine Learning Models**

Various machine learning models are implemented to classify and predict student depression levels:

**1. Logistic Regression: A Simple Yet Effective Model for Binary Classification**

**Logistic regression** is a fundamental **statistical model** used for **binary classification** problems, where the target variable has two possible outcomes (e.g., depressed vs. not depressed). It predicts the **probability** of an instance belonging to a particular class by applying the **sigmoid (logistic) function** to a linear combination of the input features. The sigmoid function maps the output to a range between **0 and 1**, making it ideal for **probability estimation**.

Mathematically, logistic regression models the **log-odds** of the event occurring rather than directly modeling the outcome. The log-odds are calculated using the formula:



The model uses **Maximum Likelihood Estimation (MLE)** to find the coefficients that **maximize the likelihood** of observing the given data. The output is interpreted as the **probability of belonging to the positive class**. By applying a threshold (typically 0.5), the model classifies the observation into one of the two classes.

In depression analysis, logistic regression is effective for identifying **risk factors** associated with depression. For instance, it can predict whether a student is likely to experience depression based on features such as **academic stress, sleep duration, or financial pressure**. The model’s coefficients reveal the **impact of each feature** on the likelihood of depression, making it easy to interpret and apply in real-world scenarios.

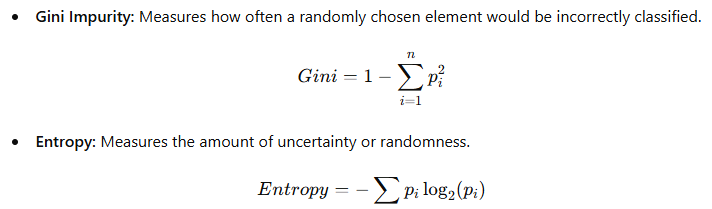
While logistic regression is simple and efficient, it assumes **linearity of the log-odds** and may struggle with **non-linear patterns**. Nevertheless, it serves as a strong baseline model for **binary classification** tasks in depression prediction and is often used as a benchmark before applying more complex algorithms.

**2. Decision Trees & Random Forests:**

**1. Decision Trees: Interpretable Classification Model**

**Decision trees** are a **supervised machine learning model** used for both **classification and regression** tasks. They use a **tree-like structure** where the data is recursively split based on feature values. Each internal node represents a **decision rule** based on a feature, branches represent possible outcomes, and leaf nodes contain the **predicted class label** or value.

The model uses **Gini impurity** or **Entropy** as the splitting criterion:



During training, the model selects the **feature and threshold** that results in the **largest information gain** or **lowest impurity**. The process continues until either:

* All samples in a node belong to the same class.
* A predefined **max depth** or **minimum samples per leaf** is reached.

**Advantages of Decision Trees:**

* Easy to interpret and visualize.
* No need for feature scaling or normalization.
* Handles both numerical and categorical features.

**Disadvantages of Decision Trees:**

* **Prone to overfitting**, especially on small or noisy datasets.
* **Unstable**: Small changes in the data can lead to a completely different tree structure.
* Less accurate compared to ensemble models.

**2. Random Forests: An Ensemble of Decision Trees**

**Random forests** are a **powerful ensemble model** that combines multiple **decision trees** to improve accuracy and reduce overfitting. The model creates a **forest of decision trees**, where each tree is trained on a **random subset of the data** and features. The final prediction is made by:

* **Classification:** Taking the **majority vote** of all decision trees.
* **Regression:** Averaging the predictions of all trees.

**Steps in Random Forests:**

1. **Bootstrap Sampling:** The model randomly selects a subset of the training data with replacement for each tree.
2. **Feature Randomness:** Each tree is trained on a random subset of features, preventing correlation between trees.
3. **Aggregation:** The predictions from all trees are combined through majority voting (classification) or averaging (regression).

**Advantages of Random Forests:**

* **Improved accuracy** compared to a single decision tree.
* **Reduces overfitting** by averaging multiple trees.
* Can handle **missing values** and noisy data effectively.
* Works well with large datasets.

**Disadvantages of Random Forests:**

* Less interpretable than a single decision tree.
* Requires more **computational resources**.
* May still overfit with very large ensembles.

**3. Key Differences: Decision Trees vs. Random Forests**

|  |  |  |
| --- | --- | --- |
| Aspect | Decision Trees | Random Forests |
| Model Type | Single tree structure | Ensemble of multiple decision trees |
| Accuracy | Less accurate, prone to overfitting | More accurate due to ensemble effect |
| Stability | Sensitive to small changes in data | Stable and robust |
| Interpretability | Easy to interpret | Less interpretable |
| Complexity | Simpler model, faster | More complex, requires more resources |
| Overfitting | Prone to overfitting | Reduces overfitting |

**4. Applications in Depression Analysis**

* **Decision Trees:** Useful for creating simple, interpretable models to predict whether a student is depressed or not based on factors like **sleep duration, academic stress, and financial pressure**.
* **Random Forests:** More suitable for complex datasets, as it reduces overfitting and improves prediction accuracy. It can identify **feature importance**, helping determine which factors contribute most to depression.

**3. Support Vector Machine (SVM):**

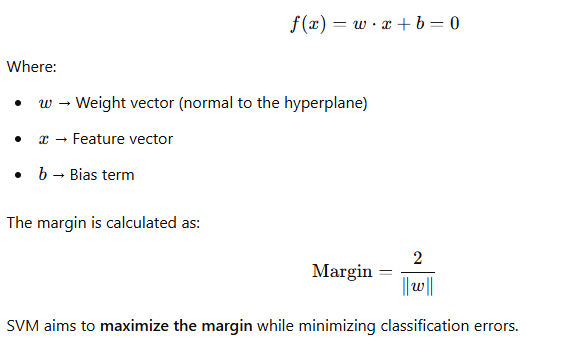
**Support Vector Machine (SVM)** is a **supervised machine learning algorithm** used for **classification and regression** tasks. It is particularly effective for **binary classification** problems and works well with both **linearly separable** and **non-linearly separable** data. SVM aims to find the **optimal hyperplane** that best separates the data into different classes with the **maximum margin**.

**1. How SVM Works**

In SVM, the goal is to create a **hyperplane** that separates the classes while maximizing the distance between the nearest points of each class. These nearest points are called **support vectors**.

* **For linearly separable data:** SVM finds the optimal hyperplane by maximizing the **margin**, which is the distance between the support vectors and the hyperplane.
* **For non-linearly separable data:** SVM uses a **kernel trick** to transform the data into a higher-dimensional space where it becomes linearly separable.

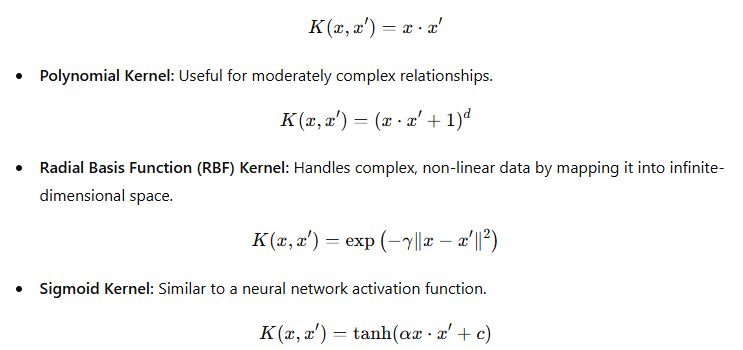
The **decision boundary** is defined by the equation:



**2. Kernel Trick for Non-Linearly Separable Data**

When the data is not linearly separable, SVM uses **kernel functions** to transform the input space into a higher-dimensional space where the classes become linearly separable. Common kernel functions include:

* **Linear Kernel:** Suitable for linearly separable data.



**3. Soft Margin and Regularization**

In real-world datasets, perfect separability is rare. SVM uses a **soft margin** to allow for some misclassifications. The **C parameter** controls the trade-off between **maximizing the margin** and **minimizing misclassifications**:

* **High C value:** Fewer misclassifications, but a smaller margin (more overfitting).
* **Low C value:** Larger margin but more misclassifications (better generalization).

**4. Advantages and Disadvantages**

**Advantages of SVM:**

* Effective for **high-dimensional data**.
* Works well with both **linear and non-linear** relationships using kernels.
* **Resistant to overfitting** when properly tuned.
* Suitable for **small to medium-sized datasets**.

**Disadvantages of SVM:**

* **Computationally expensive** for large datasets.
* Difficult to tune hyperparameters (**C** and **gamma**).
* Less interpretable compared to models like logistic regression or decision trees.

**5. Applications of SVM in Depression Analysis**

* **Depression Classification:** SVM can classify whether a student is likely to be depressed or not based on features like **academic stress, sleep duration, and financial stress**.
* **Feature Selection:** SVM can be used with **Recursive Feature Elimination (RFE)** to select the most important features contributing to depression.
* **Anomaly Detection:** SVM can detect outliers in depression-related datasets, helping identify **extreme cases**.

**4. K-Nearest Neighbors (KNN):**

K-Nearest Neighbors (KNN) is a popular supervised machine learning algorithm used for both classification and regression tasks. It is based on the principle that similar data points exist close to each other in feature space. KNN does not explicitly learn a model during training; instead, it memorizes the dataset and makes predictions based on the similarity between data points. This makes KNN a simple yet effective algorithm, especially for applications where pattern recognition and similarity measurement are important.

The working of KNN involves several steps. First, the user selects a value for K, which represents the number of nearest neighbors to consider. The choice of K significantly impacts the model’s performance. Once K is set, the algorithm calculates the distance between the test data point and all training data points. Common distance metrics used in KNN include Euclidean distance, Manhattan distance, and Minkowski distance. After computing distances, the algorithm selects the K closest neighbors.

In the case of classification, KNN assigns a class label to the test data point based on majority voting among its K nearest neighbors. The class that appears most frequently among the neighbors is chosen as the predicted class. For regression, the algorithm predicts the output by averaging the values of the K nearest neighbors. This approach ensures that the predictions are influenced by the most similar data points in the dataset.

Despite its simplicity, KNN has several advantages. It is easy to understand and implement, making it a good choice for initial machine learning models. It is also a non-parametric algorithm, meaning it does not assume any specific distribution of the data, making it versatile for different types of datasets. Furthermore, KNN works well with small datasets where computational cost is not a major concern.

However, KNN has some limitations. One of its main drawbacks is that it becomes computationally expensive for large datasets since it requires calculating distances for every new prediction. Additionally, the performance of KNN depends on the choice of K. A small K may result in a noisy model, while a large K may lead to oversmoothing, where important patterns are lost. The algorithm is also sensitive to irrelevant features and the curse of dimensionality, where the distance between data points becomes less meaningful as the number of features increases.

In the context of student depression analysis, KNN can be a useful tool for identifying students at risk based on survey responses, academic performance, and behavioral data. By analyzing past cases, the algorithm can classify students into different mental health risk groups, helping institutions provide necessary support. Additionally, KNN can be used to predict depression severity by comparing new cases with similar historical data. Its ability to work with labeled datasets makes it an effective approach for detecting patterns in student mental health assessments.

Each model's performance is compared to determine the most accurate predictor of student depression.

# **4.6 Model Training and Evaluation**

To ensure robust model performance, the dataset is divided into:

* **Training Set (80%)**: Used to train the ML models.
* **Testing Set (20%)**: Used to evaluate the model's generalizability.

The models are evaluated using the following metrics:

* **Accuracy**: Measures the overall correctness of the model.
* **Precision & Recall**: Evaluate model effectiveness in detecting depression.
* **F1-Score**: Balances precision and recall.
* **ROC-AUC Curve**: Assesses the model's ability to distinguish between depressed and non-depressed students.

# **4.7 Implementation Tools**

The following tools and libraries are used for implementation:

* **Python**: Primary programming language.
* **Pandas & NumPy**: For data handling and preprocessing.
* **Scikit-learn**: ML model implementation and evaluation.
* **Matplotlib & Seaborn**: Data visualization.

# **4.8 Ethical Considerations**

Since the study involves **mental health data**, ethical considerations include:

* Ensuring **data anonymity and privacy**.
* Using the data strictly for **academic and research purposes**.
* Preventing **bias in ML predictions** by using a diverse dataset.