

VIETNAM NATIONAL UNIVERSITY – HO CHI MINH CITY  
HO CHI MINH CITY UNIVERSITY OF TECHNOLOGY  
FACULTY OF COMPUTER SCIENCE AND ENGINEERING



**Data Mining – CO3029**

---

**Assignment Report**

# **Student Stress Insights: Identifying Key Predictors**

---

Advisor: Msc. Thai Do Thanh

Students: Cuong Doan Phuong Hung ID 2310381

Hieu Nguyen Trung ID 2113357

Tue Nguyen Phu ID 2213813

HO CHI MINH CITY, NOVEMBER 2025



## Contents

<b>1</b>	<b>Introduction and Project Objectives</b>	<b>1</b>
1.1	Aim of the Project . . . . .	1
1.2	Specific Objectives . . . . .	1
<b>2</b>	<b>Exploratory Data Analysis</b>	<b>2</b>
2.1	Data Loading and Initial Inspection . . . . .	2
2.2	Data Visualization and Insights . . . . .	4
2.2.1	Correlation Heatmap . . . . .	4
2.2.2	Key Features Distributions . . . . .	8
2.2.3	Detecting Anomalies and Outliers . . . . .	10
<b>3</b>	<b>Preprocessing Pipeline</b>	<b>12</b>
<b>4</b>	<b>Training Pipeline</b>	<b>14</b>
4.1	Models Selection . . . . .	14
4.1.1	Classifier Models . . . . .	14
4.2	Advanced Techniques . . . . .	19
4.3	Recursive Feature Elimination (RFE) . . . . .	20
<b>5</b>	<b>Results and Analysis</b>	<b>22</b>
5.1	Classification Results . . . . .	22
5.2	Clustering Results . . . . .	22
5.3	Model Comparison . . . . .	22

## List of Figures

2.1	Some Summary Statistics . . . . .	3
2.2	Correlation Heatmap . . . . .	5
2.3	Scatterplots of Highly Correlated Pairs $ r  \geq 0.7$ . . . . .	7
2.4	Key Features Distributions . . . . .	8
2.5	Violin Plots . . . . .	10
3.1	Anomalies Cleaned violinplots . . . . .	13
4.1	Random Forest Classifier . . . . .	15
4.2	Gradient Boosting Classifier . . . . .	16
4.3	K-Means Clustering . . . . .	17
4.4	Hierarchical (Agglomerative) Clustering and Dendrogram . . . . .	18
4.5	DBSCAN Clustering . . . . .	19
4.6	PCA algorithm . . . . .	20
4.7	RFE algorithm . . . . .	21



# 1 Introduction and Project Objectives

## 1.1 Aim of the Project

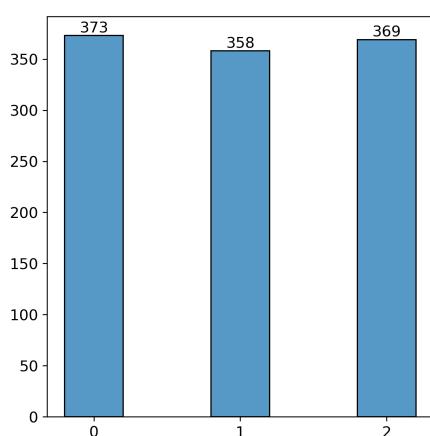
## 1.2 Specific Objectives

## 2 Exploratory Data Analysis

### 2.1 Data Loading and Initial Inspection

The dataset was loaded from a CSV file (`StressLevelDataset.csv`), which was obtained from [Kaggle](#) and imported into a Pandas DataFrame for analysis. This dataset offers a comprehensive overview of factors that influence student stress, spanning psychological, physiological, environmental, academic, and social aspects. It is suitable for tasks such as stress prediction and factor analysis, with all data provided in numerical format for straightforward processing.

- **Data Shape:** The dataset consists of 1100 rows (instances) and 21 columns (20 features + 1 target variable). This size is sufficient for exploratory analysis and modeling without overwhelming computational resources.
- **Data Quality:** There are no missing values or duplicate rows, indicating a clean and well-prepared dataset. All columns are of type `int64`, confirming numerical data with no need for type conversion or imputation. No categorical strings or timestamps are present, simplifying preprocessing.
- **Column Names and Descriptions:** The features cover a range of student experiences, measured on ordinal scales. Most are rated from 0 (low/absent) to 5 (high/severe), except for a few with wider ranges based on standard psychological metrics.
- **Target Variable Distribution:** The target variable (`stress_level`) is a multiclass label and is balanced among the classes. This near-equal distribution reduces bias in classification models and reflects a representative sample across stress severities.



- Level 0 (low stress): 373 instances (~33.9%)
- Level 1 (medium stress): 358 instances (~32.5%)
- Level 2 (high stress): 369 instances (~33.6%)

### Statistical Insights

A summary of some basic statistics for numerical features is provided below (computed via `.describe()` method), shows reasonable variability across features, with no extreme outliers



apparent at first glance.

	<b>anxiety_level</b>	<b>self_esteem</b>	<b>mental_health_history</b>	<b>depression</b>	<b>headache</b>	<b>blood_pressure</b>	<b>sleep_quality</b>	<b>breathing_problem</b>	<b>noise_level</b>
count	1100.000000	1100.000000	1100.000000	1100.000000	1100.000000	1100.000000	1100.000000	1100.000000	1100.000000
mean	11.063636	17.777273		0.492727	12.555455	2.508182	2.181818	2.660000	2.753636
std	6.117558	8.944599		0.500175	7.727008	1.409356	0.833575	1.548383	1.400713
min	0.000000	0.000000		0.000000	0.000000	0.000000	1.000000	0.000000	0.000000
25%	6.000000	11.000000		0.000000	6.000000	1.000000	1.000000	1.000000	2.000000
50%	11.000000	19.000000		0.000000	12.000000	3.000000	2.000000	2.500000	3.000000
75%	16.000000	26.000000		1.000000	19.000000	3.000000	3.000000	4.000000	3.000000
max	21.000000	30.000000		1.000000	27.000000	5.000000	3.000000	5.000000	5.000000

Figure 2.1: Some Summary Statistics

### 1. Psychological Factors:

- **anxiety\_level** and **depression** have high variability, indicating diverse mental health states among students.
- **self\_esteem** is generally positive but with noticeable spread.
- **mental\_health\_history** is nearly evenly split, useful as a binary predictor.

### 2. Physiological Factors:

- Features like **headache**, **breathing\_problem**, and **sleep\_quality** mostly cluster around mid-values; notably, a subset reports poor sleep.
- **blood\_pressure** is more often normal or high.

### 3. Environmental/Social Factors:

- Most students report moderate levels in aspects like **noise\_level**, **living\_conditions**, and **safety**, but outliers exist.
- **social\_support** is moderate, with some reporting low support.
- Stressors such as **bullying** and **peer\_pressure** are mid-range on average but right-skewed.

4. **Academic Factors:** Academic features (**academic\_performance**, **study\_load**, etc.) are generally balanced, with means close to the scale midpoint.

5. **Target Variable (stress\_level):** Almost perfectly balanced across all levels, making the dataset suitable for classification tasks.

### Overall Insights:

- The dataset is balanced and clean, with no missing values (all feature counts = 1100).

- Psychological metrics such as `anxiety_level` and `depression` exhibit higher variability (greater standard deviation relative to their range) compared to ordinal 0–5 features, which cluster around std 1.4. This suggests more diverse experiences in mental health aspects among students than in environmental or academic features.
- Quartile analysis shows that many stressor-related features (e.g., `bullying`) are slightly right-skewed: the lower quartile (25%) is low, while the upper quartile (75%) is considerably higher. Thus, while most students report moderate conditions, a minority experience much higher levels of specific stressors.
- Features with greater variance (like anxiety and depression) are expected to be stronger predictors of stress, whereas environmental variables, despite moderate means, may serve more as moderators.
- Minimum and maximum values largely stay within the designated feature ranges, with no obvious extreme outliers. These distributions will be further examined using violin plots later in the report to visually confirm the absence of anomalies and illustrate how feature spreads vary by stress level.

## 2.2 Data Visualization and Insights

Visualizations were generated using `Matplotlib` and `Seaborn` to explore distributions, relationships, and patterns. All figures were saved to the images directory for inclusion in the report. Key visualizations and insights are described below.

### 2.2.1 Correlation Heatmap

This visualization highlights key relationships influencing student stress, revealing distinct patterns in psychological, physiological, environmental, academic, and social factors. Notably, the correlations are generally moderate to strong ( $|r| > 0.6$  for many pairs), indicating interconnected dimensions of student well-being.

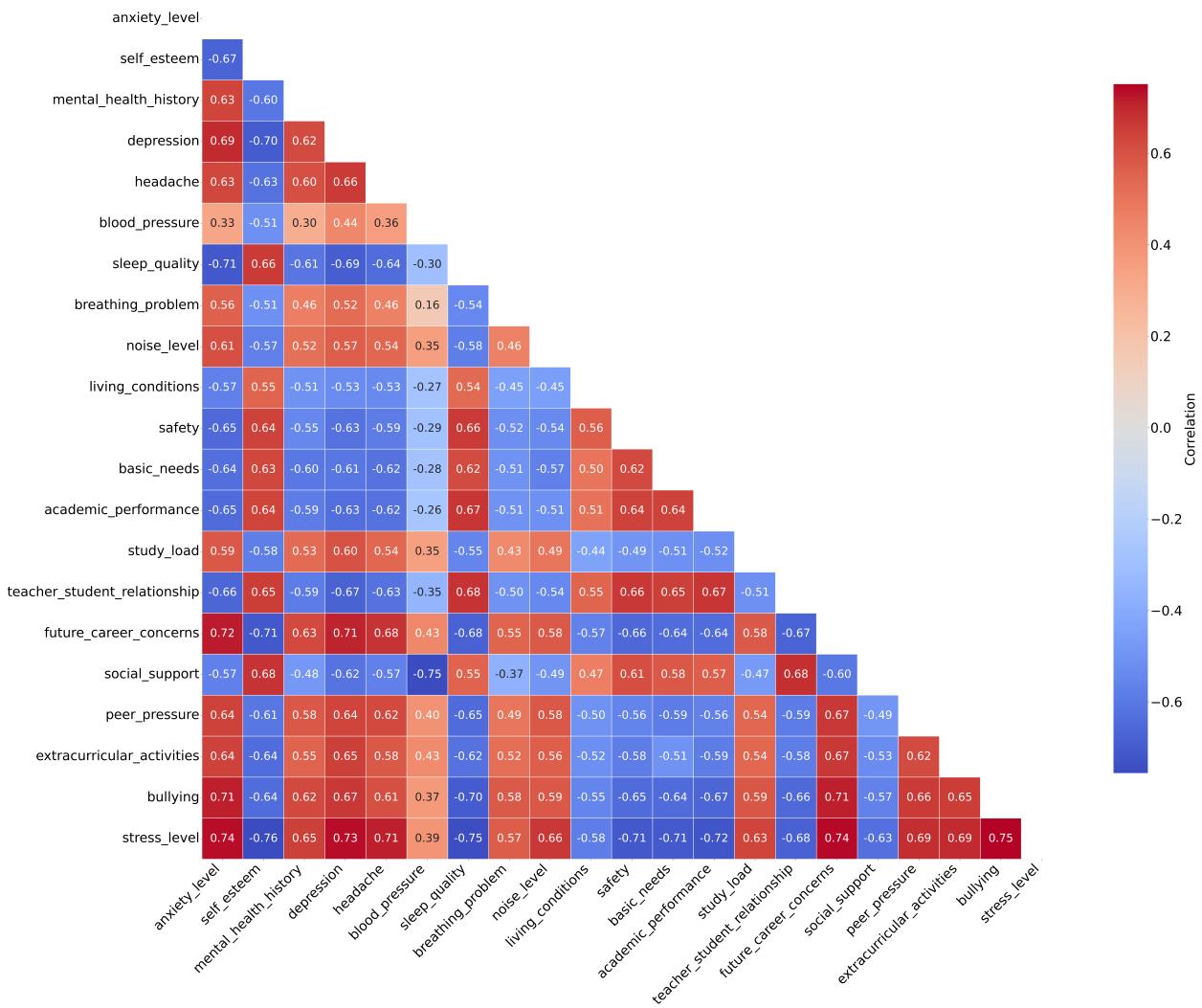


Figure 2.2: Correlation Heatmap

### Correlation Analysis Insights:

#### 1. Strongest Positive Correlations with Stress Level:

- Primary stress amplifiers:** The highest include **bullying** ( $r = 0.75$ ), **anxiety\_level** ( $r = 0.74$ ), **future\_career\_concerns** ( $r = 0.74$ ), **depression** ( $r = 0.73$ ), **headache** ( $r = 0.71$ ), and **peer\_pressure** ( $r = 0.71$ ). These align with research showing that bullying and academic pressures contribute to elevated stress and related mental health issues, such as self-harm behaviors in middle school students. Similarly, anxiety and depression are well-documented predictors of stress, often forming a vicious cycle that impacts academic outcomes.
- Secondary contributors:** **noise\_level** ( $r = 0.68$ ), **extracurricular\_activities** ( $r = 0.67$ ), **study\_load** ( $r = 0.65$ ), and **mental\_health\_history** ( $r = 0.65$ ), implying that environmental noise and overloaded schedules exacerbate stress, consistent

with studies on chronic academic stress leading to insufficient sleep and negative affect.

- **Physiological indicators:** `blood_pressure` ( $r = 0.39$ ) and `breathing_problem` ( $r = 0.37$ ) show milder associations, potentially indicating secondary stress manifestations.

## 2. Strongest Negative Correlations with Stress Level:

- **Key protective factors:** `selfEsteem` ( $r = -0.76$ ), `sleepQuality` ( $r = -0.75$ ), `basicNeeds` ( $r = -0.72$ ), `livingConditions` ( $r = -0.71$ ), `safety` ( $r = -0.71$ ), and `academicPerformance` ( $r = -0.71$ ). This suggests that fulfilling basic needs, safe environments, and strong academic achievement act as buffers against stress, corroborating findings that self-esteem and sleep quality are crucial predictors of lower stress levels among students.
- **Social support factors:** `socialSupport` ( $r = -0.68$ ) and `teacherStudentRelationship` ( $r = -0.63$ ). This highlights the role of supportive networks in reducing stress, as supported by research on mental health in first-year university students.

## 3. Inter-Correlation Clusters:

- **Mental health vulnerability cluster:** `anxietyLevel` with `depression` ( $r = 0.69$ ), both strongly linked with `mentalHealthHistory` ( $r \approx 0.63-0.70$ ), indicating these factors often co-occur and amplify stress.
- **Resilience cluster:** `selfEsteem`, `sleepQuality`, `academicPerformance`, and `socialSupport` (inter-correlations  $r \approx 0.50-0.70$ ), all negatively associated with stress-related variables.
- **Social pressure cluster:** `bullying`, `peerPressure`, and `futureCareerConcerns` (inter-correlations  $r \approx 0.50-0.65$ ), reinforcing their collective impact on stress levels.

## 4. Implications for Modeling and Multicollinearity:

- **Multicollinearity concerns:** High inter-correlations (`depression` and `selfEsteem` at  $r = -0.70$ , `anxietyLevel` and `sleepQuality` at  $r = -0.66$ ) signal potential multicollinearity that could inflate variance in regression models.
- **Recommended techniques:** Variance inflation factor (VIF) checks or principal component analysis (PCA) are recommended to address multicollinearity issues.
- **Feature selection priority:** The heatmap provides a foundation for feature selection in predictive modeling, prioritizing high-correlation variables like `bullying` and `selfEsteem`.

- **Intervention targets:** Patterns suggest that interventions targeting mental health (reducing anxiety and improving sleep) and social factors (anti-bullying programs) could effectively lower stress levels.

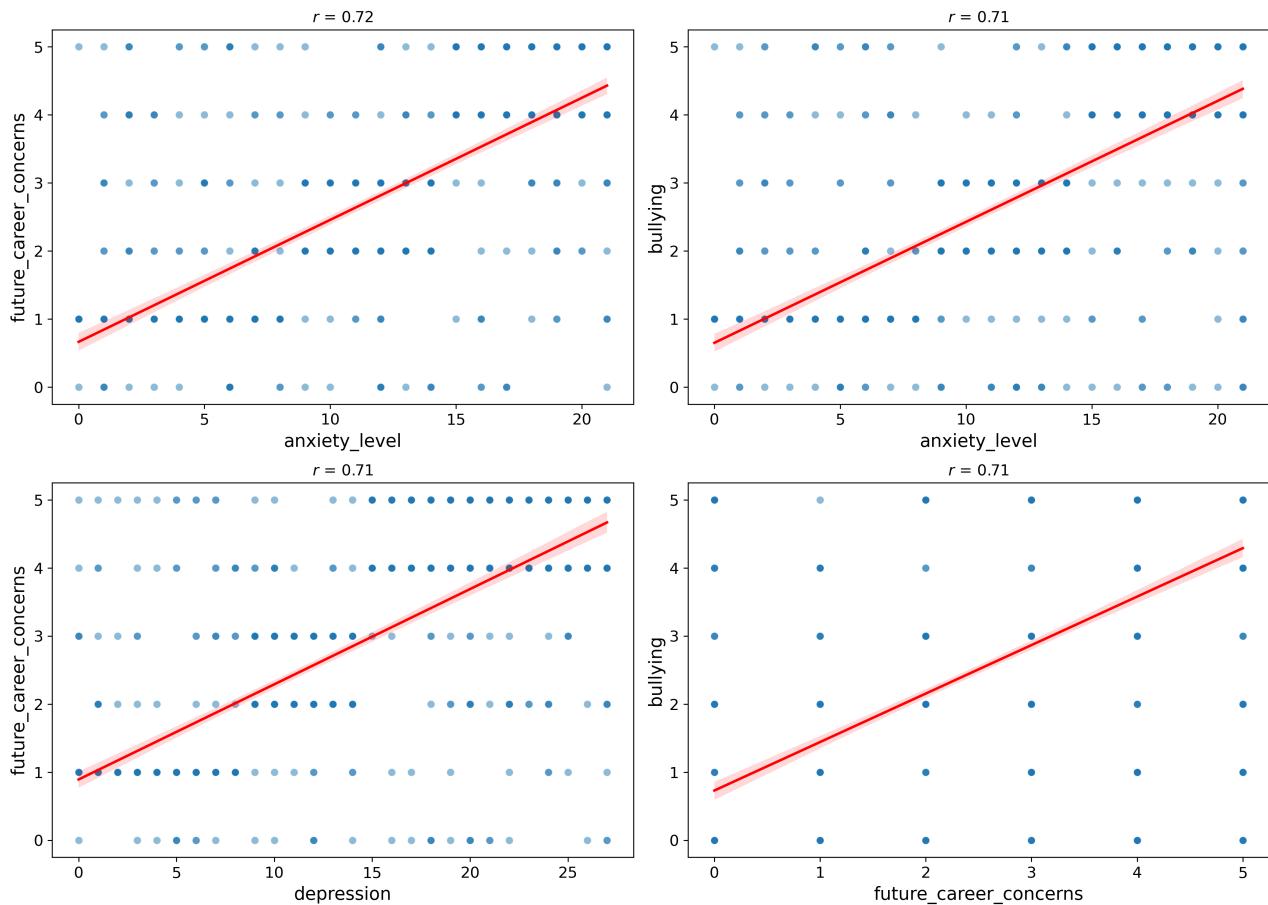


Figure 2.3: Scatterplots of Highly Correlated Pairs  $|r| \geq 0.7$

### Insights:

- The scatterplots highlight pairs with  $|r| \geq 0.7$  from the correlation analysis: Each pair exhibits a strong positive linear trend with dense clustering and few outliers, echoing the correlation heatmap and underscoring the interconnectedness of key student stressors.
- In data mining, these high correlations signal potential multicollinearity, making variance inflation factor (VIF) assessment or principal component analysis (PCA) necessary for dimensionality reduction to improve model stability in stress prediction.
- The pronounced linear relationships support the use of regression-based predictive models, while the presence of tight point clusters suggests the applicability of clustering algorithms to identify subgroups and enable targeted feature engineering.

### 2.2.2 Key Features Distributions

The following visualizations are essential in data mining for identifying patterns, skewness, and potential preprocessing needs, such as normalization or transformation, to enhance model performance in classification tasks like this topic.

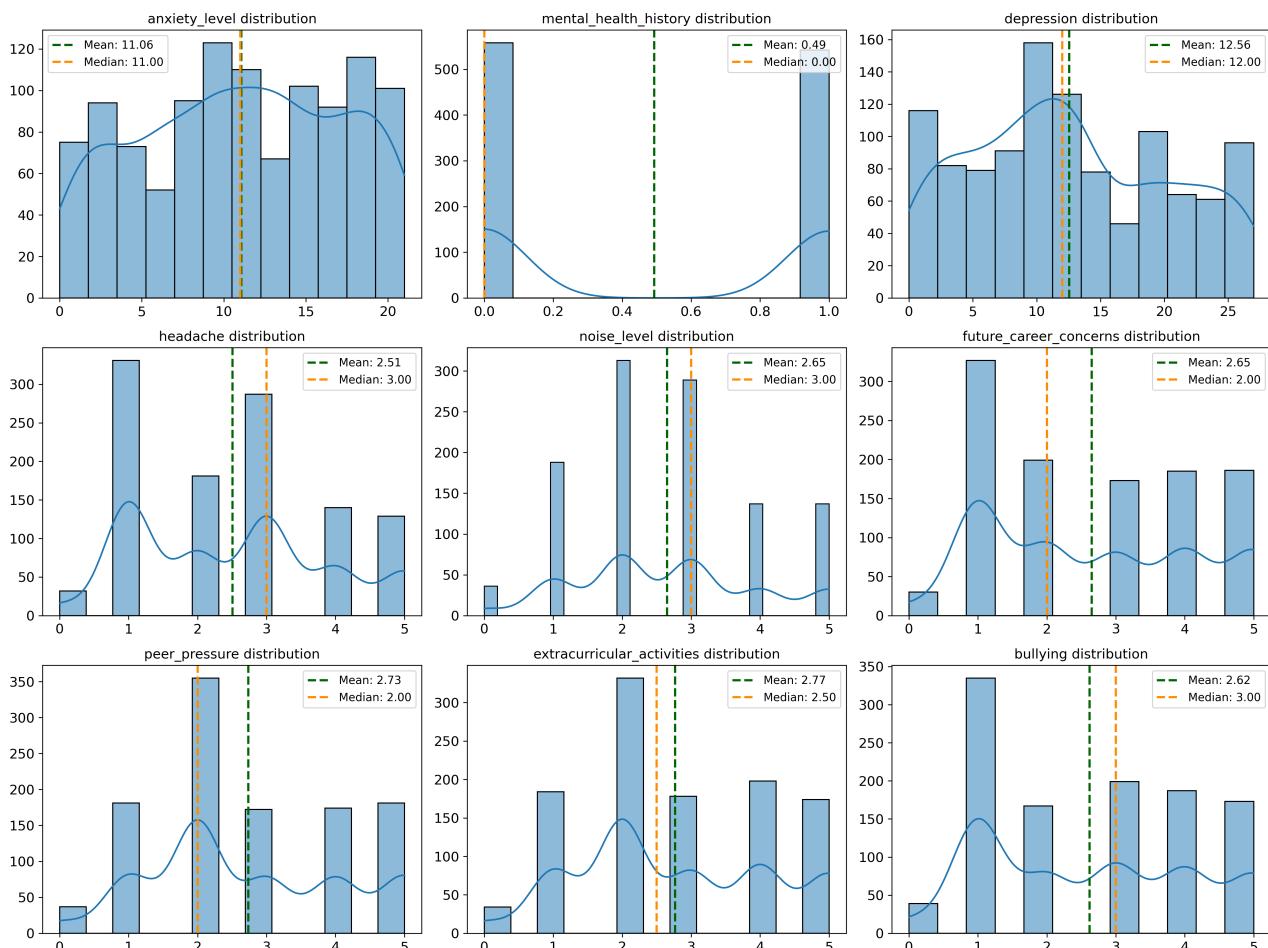


Figure 2.4: Key Features Distributions

#### Insights:

##### 1. Symmetric and Bell-Shaped Distributions

- Psychological features like `anxiety_level` and `depression` exhibit approximately normal distributions with peaks in the moderate range.
- Indicates widespread moderate symptoms among students, aligning with research showing prevalent anxiety and depression linked to academic pressures.
- For data mining, such normal distributions are ideal for parametric models (e.g., linear regression) with minimal need for transformation.



- High variability ( $\text{std} \approx 6-7$ ) suggests feature scaling may be necessary for algorithms such as SVM or neural networks, since these models are sensitive to differences in feature scales and may perform poorly or converge slowly if features are not normalized.

## 2. Binary and Imbalanced Distribution

- `Mental_health_history` is binary and slightly imbalanced (~55% at 0, mean 0.49, median 0.00): roughly half of students report no prior mental health issues.
- This feature could serve as a strong categorical predictor in mining tasks but may require stratified sampling to avoid bias.
- Its skewed distribution (mode at 0) highlights possible class imbalance, suggesting oversampling techniques may be needed.

## 3. Skewed Distributions in Social and Academic Stressors

- Features like `future_career_concerns`, `peer_pressure`, and `bullying` show positive skewness (right-tailed) with modes at lower values.
- Most students experience low-to-moderate levels, but some face intense issues, consistent with research on adolescent stress.
- For data mining, applying log transformations or binning may normalize these features or optimize them for tree-based models.

## 4. Multimodal or Mildly Skewed Distributions

- Headache and `extracurricular_activities` have bimodal patterns; `noise_level` is more symmetric with a central mode at 3.
- These indicate clusters in student experiences – such as differing levels of environmental noise or activity participation – which may impact stress.
- In data mining, multimodal features suggest exploring clustering algorithms (e.g., K-means) to segment subgroups, since multiple peaks often indicate the presence of distinct groups within the data.
- Mean-median discrepancies indicate mild left skew; outlier checks and robust scaling may be warranted.

## 5. Data Mining Implications

- The dataset exhibits a mix of normal and skewed features; this informs preprocessing strategies such as feature scaling and strategies for skewness.

- Agreement between mean and median in symmetric features supports using standard central tendency; skewed features highlight opportunities for feature engineering.
- These findings guide further analyses, such as bivariate plots against `stress_level`, to investigate conditional distributions for predictive modeling.

### 2.2.3 Detecting Anomalies and Outliers

The violin plots provide a bivariate view of the distributions, combining density estimates with boxplot elements to reveal spreads, modes, and potential outliers. These visualizations are crucial in data mining for detecting anomalies, assessing group differences, and informing preprocessing steps like outlier handling, which can significantly impact model accuracy in classification or regression tasks. The plots highlight how feature values shift across stress categories, with wider violins indicating higher variability and tails signaling outliers, often more pronounced in high-stress groups.

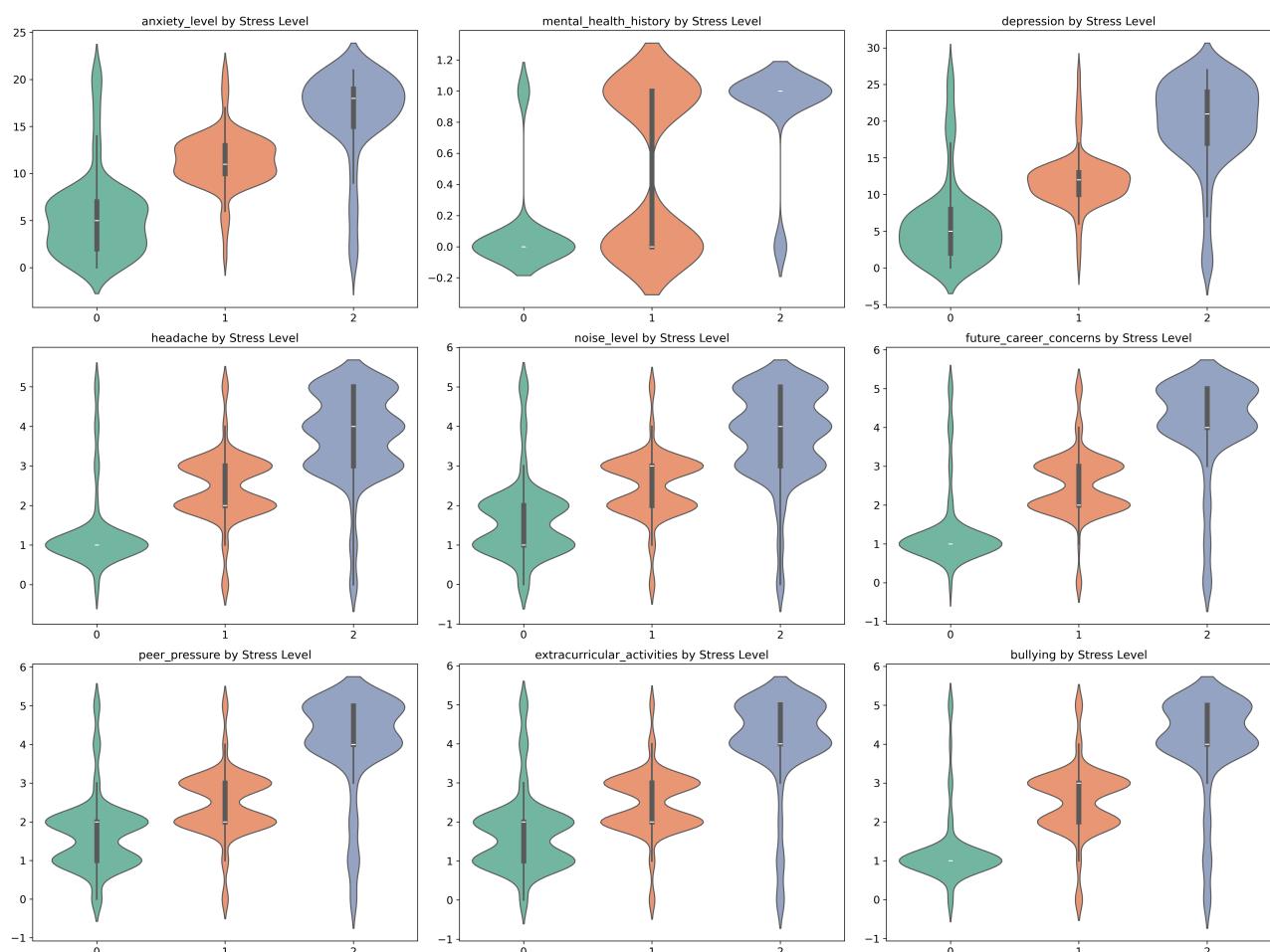


Figure 2.5: Violin Plots

### Insights:

## 1. Increasing Trends with Stress Levels

- Psychological and social features including `anxiety_level`, `depression`, `bullying`, `peer_pressure`, and `future_career_concerns` exhibit clear positive shifts across stress categories: low-stress groups present with narrow, low-value densities, while high-stress demonstrate broader, higher-value distributions.
- Features such as `headache` and `noise_level` also rise notably, with medians progressing from  $\sim 1\text{--}2$  (low stress) to  $\sim 3\text{--}4$  (high stress).
- This trend is consistent with educational data mining findings, where engineered features – such as binned or transformed high tails – improve model interpretability of stress predictors.
- `Extracurricular_activities` trends upward as well, implicating overload as a contributing factor, in line with studies on academic performance stressors.

## 2. Binary Feature Insights

- The binary feature `Mental_health_history` manifests as thin spikes: level 0 is concentrated at 0, whereas level 2 becomes bimodal with notable density at 1.
- This suggests a threshold effect valuable for decision rules in tree-based models, where binary splits on such features effectively classify stress levels.

## 3. Outlier Detection and Variability

- Extended tails in high-stress violin plots – notably for `depression` and `anxiety` (outliers beyond 20–25 at level 2), and `bullying` or `peer_pressure` (up to 5) – highlight potential anomalies.
- The increase in feature variability (wider violins at level 2) signals heteroscedasticity, justifying the use of robust preprocessing to facilitate better generalization.

## 4. Data Mining Implications

- These observations emphasize the need for outlier-aware preprocessing in student stress prediction pipelines, as anomalous values (especially in `depression`) can unduly influence feature importance metrics.
- Clustering techniques may help segment and analyze outlier subpopulations, enhancing model robustness.
- Collectively, these insights inform feature selection and transformation choices, ultimately supporting improved predictive performance for multiclass stress classification – mirroring findings in related data mining competitions and research.

### 3 Preprocessing Pipeline

#### Anomaly Detection and Data Cleaning

Althouugh the dataset is reported to contain no missing values (NaN), we nonetheless perform through data cleaning as follow to ensure integrity and consistency:

```
1 # Handle missing values
2 missing_values = df.isnull().sum()
3 print(f"Missing values per column:\n{missing_values}")
4 numeric_cols = self.data.select_dtypes(include=[np.number]).columns
5 categorical_cols = data.select_dtypes(include=['object']).columns
6 # Impute numeric columns
7 if len(numeric_cols) > 0:
8     imputer = SimpleImputer(strategy=strategy)
9     data[numeric_cols] = imputer.fit_transform(data[numeric_cols])
10 # Impute categorical columns
11 if len(categorical_cols) > 0:
12     imputer = SimpleImputer(strategy='most_frequent')
13     data[categorical_cols] = imputer.fit_transform(data[categorical_cols])
14 # Handle duplicates
15 duplicate_count = df.duplicated().sum()
16 print(f"Number of duplicate rows: {duplicate_count}")
17 data = data.drop_duplicates()
```

#### Anomaly Cleaning and Applying SMOTE

```
1 for col in columns:
2     Q1 = df[col].quantile(0.25)
3     Q3 = df[col].quantile(0.75)
4     IQR = Q3 - Q1
5     lower_bound = Q1 - 1.5 * IQR
6     upper_bound = Q3 + 1.5 * IQR
7     df = df[(df[col] >= lower_bound) & (df[col] <= upper_bound)]
8 # SMOTE
9 smote = SMOTE(random_state=42)
10 X_res, y_res = smote.fit_resample(X, y)
```

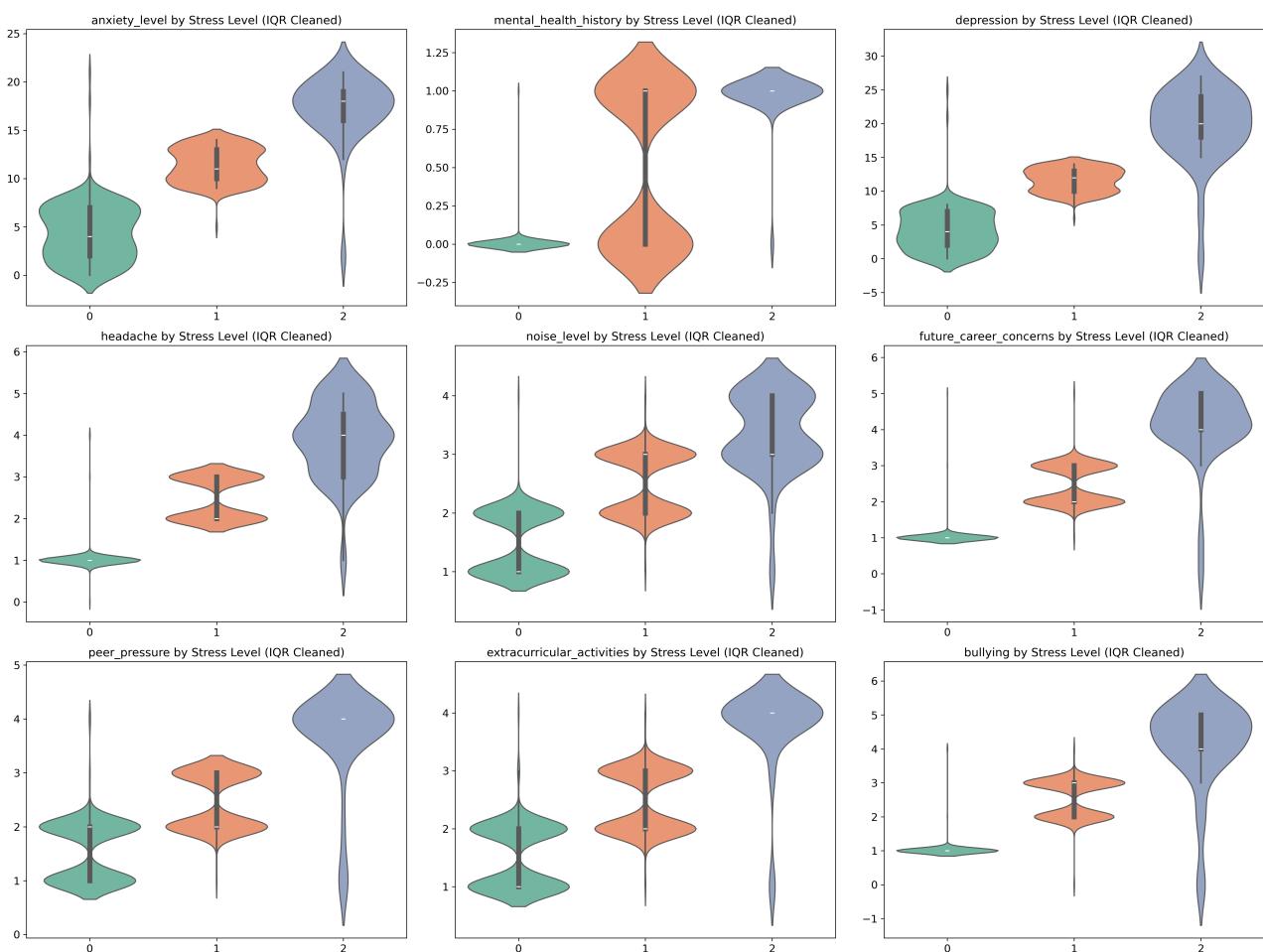


Figure 3.1: Anomalies Cleaned violinplots

### Encode Categorical Variables

```

1 from sklearn.preprocessing import LabelEncoder
2 for col in columns:
3     if col in data.columns:
4         data[col] = label_encoder.fit_transform(data[col].astype(str))

```

### Split Intro Train & Validating Data

```

1 X = data.drop(columns=[target_col])
2 y = data[target_col]
3 X_train, X_test, y_train, y_test = train_test_split(
4     X, y, test_size=test_size, random_state=42, stratify=y
5 )

```

## 4 Training Pipeline

### 4.1 Models Selection

#### 4.1.1 Classifier Models

The dataset analyzed comprises 1100 instances and 20 features, encompassing numerical, categorical, and discrete integer variables, designed to classify students as dropout, enrolled, or graduate. While no missing values were present, the dataset exhibited significant class imbalance, providing a realistic scenario for educational predictive modeling focused on the early identification of at-risk stressed students using machine learning classifiers.

```
1 models = {  
2     'Random Forest': RandomForestClassifier(n_estimators=100, random_state=42,  
3         n_jobs=-1),  
4     'Decision Tree': DecisionTreeClassifier(random_state=42, max_depth=10),  
5     'SVM': SVC(kernel='rbf', random_state=42),  
6     'KNN': KNeighborsClassifier(n_neighbors=5),  
7     'Naive Bayes': GaussianNB(),  
8     'Logistic Regression': LogisticRegression(random_state=42, max_iter=1000),  
9     'Gradient Boosting': GradientBoostingClassifier(n_estimators=100, random_state=42)  
}
```

Seven classification model pipelines were evaluated in total, but this report emphasizes tree-based ensemble methods – Random Forest and Gradient Boosting – selected for their aptitude in managing mixed feature types, interpretability, and robust performance under imbalanced conditions.

#### Random Forest Classifier

Random Forest works by creating an ensemble of multiple decision trees, each trained on a random subset of the data (bootstrapping) and features, then aggregating their predictions through majority voting for classification or averaging for regression. This reduces overfitting, handles variance well, and improves accuracy by leveraging diversity among the trees.

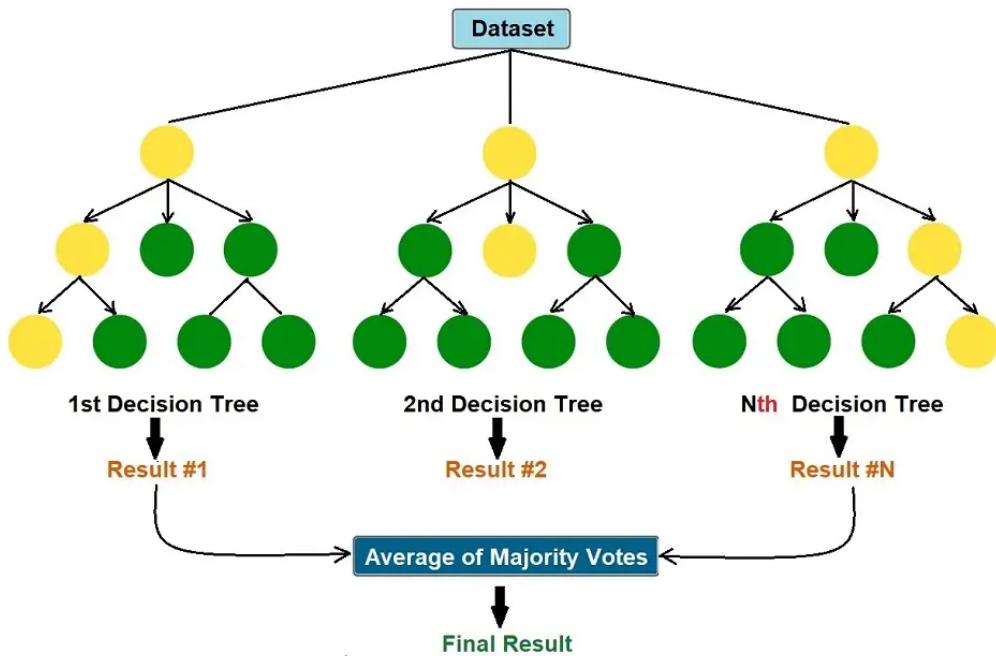


Figure 4.1: Random Forest Classifier

Random Forest (RF) is suitable for this dataset as an ensemble method that builds multiple decision trees via bagging, effectively handling mixed feature types, non-linear relationships, and class imbalance with techniques like class weighting. It reduces overfitting through randomness and provides feature importance scores, offering interpretable insights into factors like socio-economic status or grades that influence dropout, while achieving robust performance on tabular data without extensive preprocessing.

### Gradient Boosting Classifier

Gradient Boosting builds trees sequentially, where each new tree corrects the errors of the previous ones by minimizing a loss function using gradient descent. It focuses on hard-to-predict instances, often achieving high performance through additive modeling, with regularization to prevent overfitting.

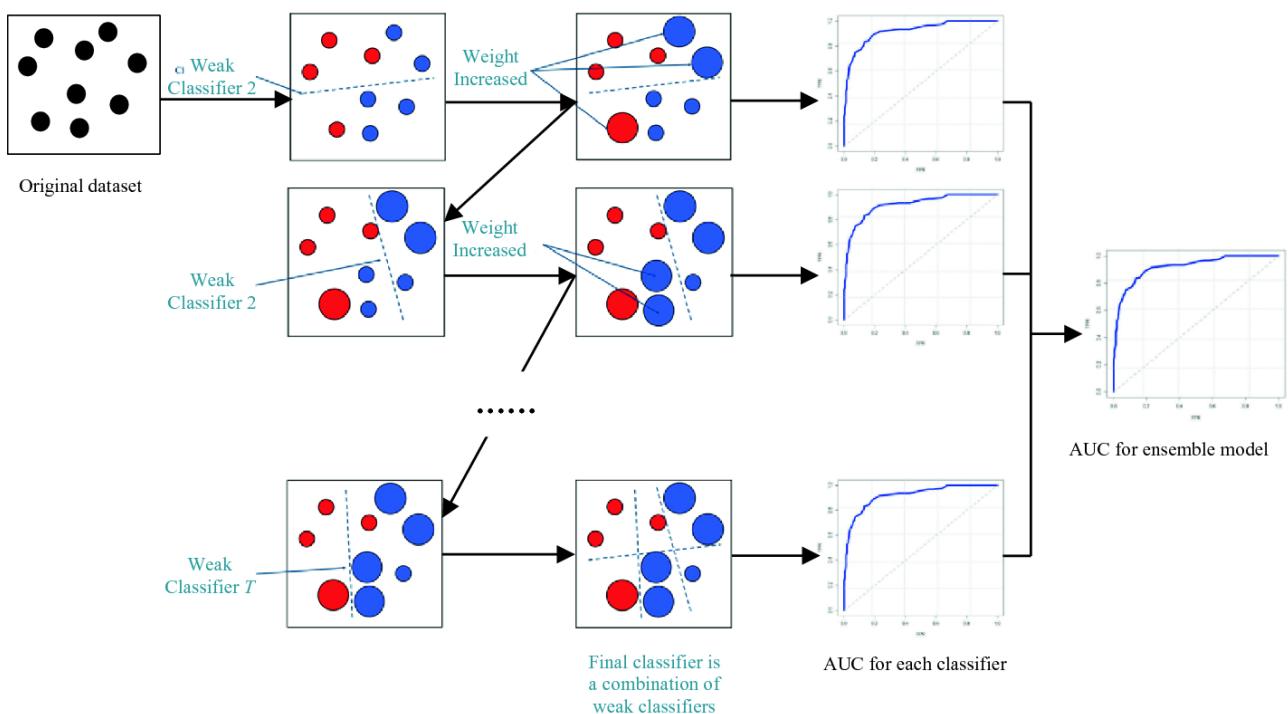


Figure 4.2: Gradient Boosting Classifier

Gradient Boosting excels by sequentially building trees to correct errors, capturing complex interactions and subtle patterns in the data better than single models. It addresses imbalance with built-in weighting and regularization to prevent overfitting, often yielding higher accuracy on imbalanced educational datasets, and includes feature importance for explaining predictions, making it a strong choice for model selection to optimize metrics like F1-score in student success forecasting.

## Clustering Models

Unsupervised clustering models were applied to discover latent groupings among students, offering insights that complement supervised classification. Principal Component Analysis (PCA) was used beforehand to reduce dimensionality while retaining most variance, enabling effective visualization and interpretation of clusters.

```

1  from sklearn.decomposition import PCA
2  from sklearn.cluster import KMeans, AgglomerativeClustering, DBSCAN
3  # Apply PCA for reduction
4  X_pca = PCA(n_components=2, random_state=42).fit_transform(X_scaled)
5  # K-Means
6  kmeans = KMeans(n_clusters=3, random_state=42)
7  labels_kmeans = kmeans.fit_predict(X_pca)
8  # Hierarchical

```

```

9  hier = AgglomerativeClustering(n_clusters=3)
10 labels_hier = hier.fit_predict(X_pca)
11 # DBSCAN
12 dbscan = DBSCAN(eps=0.5, min_samples=5)
13 labels_dbscan = dbscan.fit_predict(X_pca)
14

```

Clustering revealed distinct student groups based on multivariate patterns underlying stress and academic engagement. K-Means and hierarchical clustering provided grouping stability and interpretability, while DBSCAN offered robust detection of anomalies and noise – crucial for flagging at-risk students who may otherwise be masked in class-centric analyses. Visualizations and silhouette scores guided the evaluation and naming of meaningful clusters used for downstream profiling and recommendations.

### K-Means Clustering

K-Means is a partition-based clustering algorithm that groups data into  $k$  non-overlapping clusters by minimizing the within-cluster sum-of-squares. Each data point is assigned to the nearest cluster centroid, and centroids are iteratively updated until assignments stabilize. This method is well-suited for datasets with quantitative features and helps to reveal underlying group structures by forming compact, clearly-separated clusters.

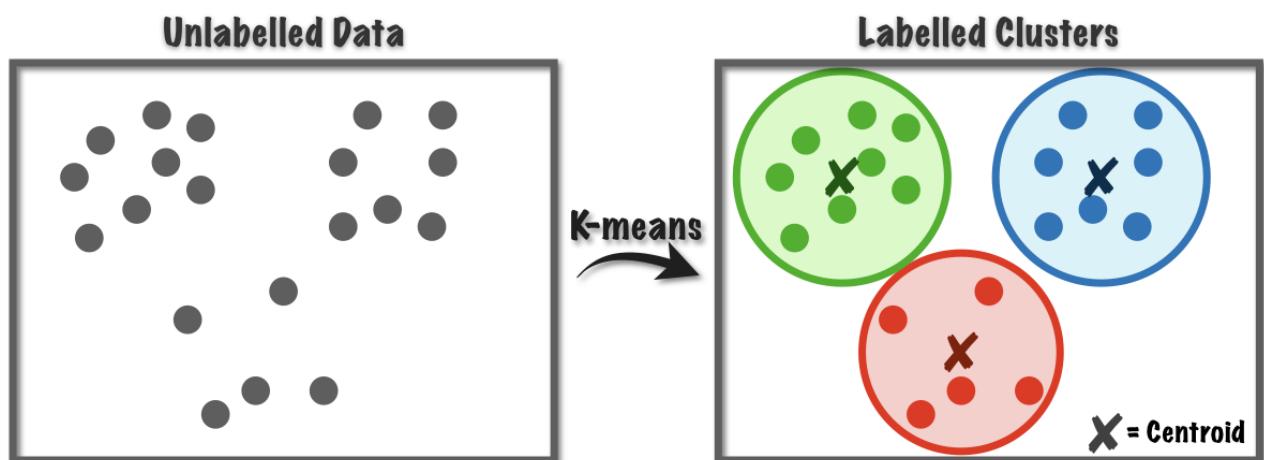


Figure 4.3: K-Means Clustering

For this dataset, K-Means performs robustly due to its completely numerical feature set, balanced class distribution, and absence of missing values. The consistent scaling and moderate size of the data allow for fast convergence and reliable results. These factors enable K-Means to

effectively identify groups of students who share similar stress profiles and academic patterns, supporting the discovery of actionable segments for targeted interventions and recommendations.

### Hierarchical Clustering

Hierarchical clustering, specifically agglomerative clustering, builds a nested tree (dendrogram) of clusters by successively merging the two closest data points or clusters until all points are united into a single cluster. This process uses linkage criteria, such as Ward's method, to define cluster distances and can reveal rich structural relationships within the dataset at multiple granularities.

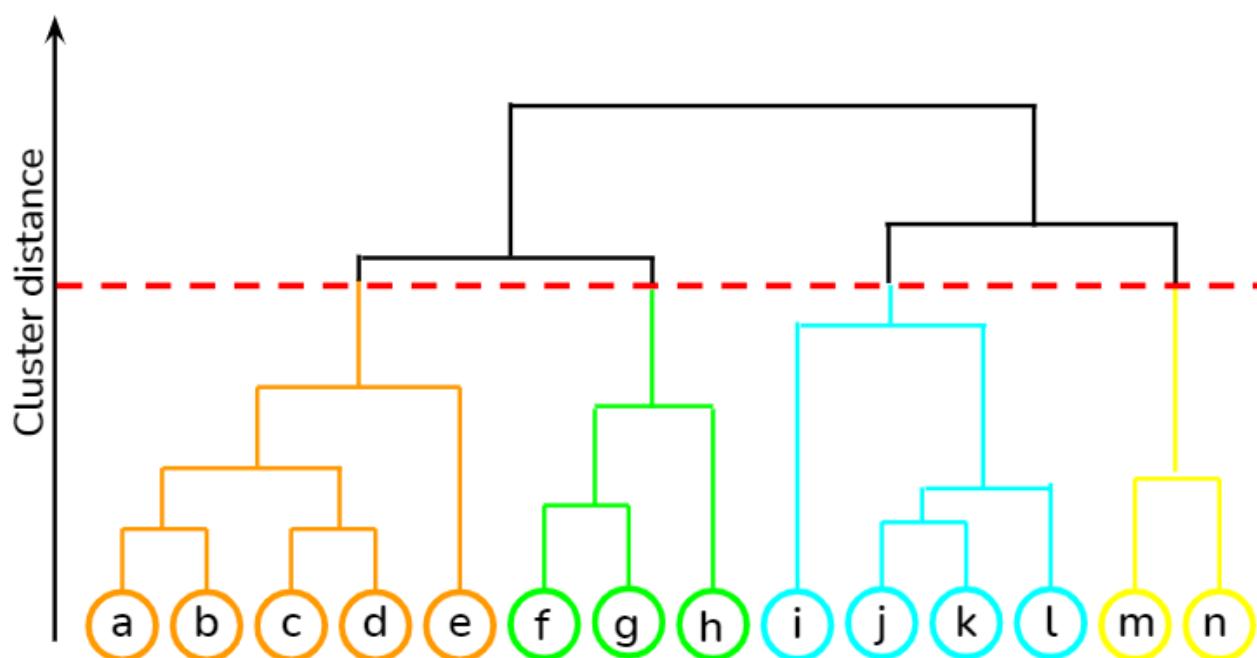


Figure 4.4: Hierarchical (Agglomerative) Clustering and Dendrogram

Hierarchical clustering is particularly valuable for this student dataset, offering interpretable visual groupings through dendograms that expose both global and local patterns among students' stress and academic profiles. Because our features are purely numerical and well-scaled, hierarchical clustering produces meaningful splits without being skewed by differing ranges or noise. The ability to visualize the merge steps assists in selecting the most natural number of clusters, while also detecting subgroups and anomalous cases. This complements K-Means by clarifying nested relationships, supporting granular targeting in interventions and profiling strategies.

## DBSCAN Clustering

DBSCAN (Density-Based Spatial Clustering of Applications with Noise) forms clusters by connecting points close together in dense regions and labels points in sparse areas as noise (outliers). Unlike K-Means or hierarchical clustering, DBSCAN does not require the number of clusters to be specified in advance, and it can discover clusters of irregular shape and variable size.

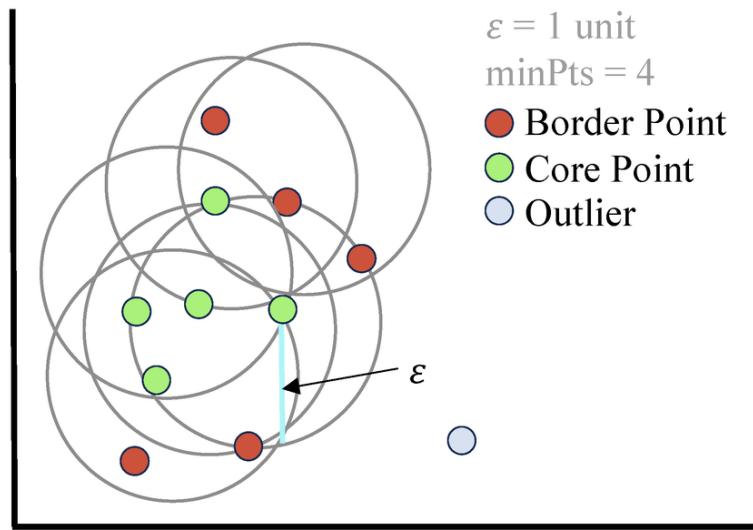


Figure 4.5: DBSCAN Clustering

DBSCAN is particularly powerful in this student context for detecting minority subgroups or atypical students whose stress or engagement profiles are substantially different from their peers. By identifying outliers, DBSCAN helps flag at-risk individuals who might not belong to any major cluster and could otherwise be overlooked by centroid- or linkage-based methods. The ability to tune `eps` (radius threshold) and `min_samples` provides flexibility for handling varying densities in student survey responses, revealing both tight-knit groups and peripheral cases critical for intervention.

## 4.2 Advanced Techniques

### Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is an unsupervised technique for dimensionality reduction that transforms correlated original features into a smaller number of uncorrelated variables, called principal components, capturing the maximum variance present in the data.

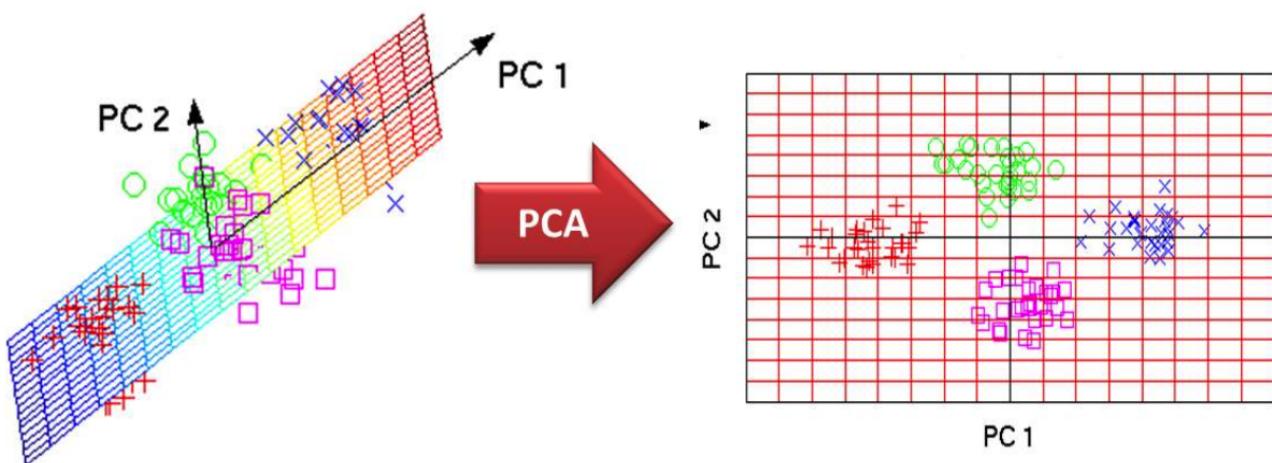


Figure 4.6: PCA algorithm

PCA is well-suited for our student stress dataset (1100 samples, 20 numerical features), which shows strong inter-feature correlations (e.g., `anxiety_level` and `depression`  $r = 0.69$ ). This multicollinearity can inflate model variance and cause overfitting. PCA solves this by projecting data onto orthogonal axes of maximum variance. With standardized numeric features, PCA efficiently reduces dimensions while retaining at least 95% of the variance.

For this data, PCA typically reduces dimensionality to 8-10 principal components, with the main components reflecting variability in psychological and social factors. Higher-variance features dominate early components, improving interpretability and classifier performance.

```
1 from sklearn.preprocessing import StandardScaler
2 from sklearn.decomposition import PCA
3 # Standardize features before PCA
4 scaler = StandardScaler()
5 X_scaled = scaler.fit_transform(X)
6 # Fit PCA to retain 95% of explained variance
7 pca = PCA(n_components=0.95, random_state=42)
8 X_pca = pca.fit_transform(X_scaled)
```

PCA thus mitigates the curse of dimensionality, addresses multicollinearity, and reduces noise, benefiting downstream supervised and unsupervised learning tasks.

### 4.3 Recursive Feature Elimination (RFE)

Recursive Feature Elimination (RFE) is a supervised feature selection method that recursively removes the least important features according to a model's coefficients or feature importances, ranking all features and selecting the top subset.

## How Recursive Feature Elimination Works

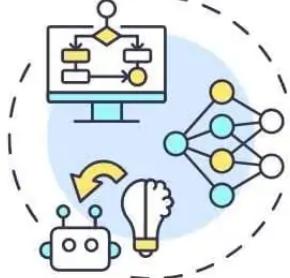


Figure 4.7: RFE algorithm

In this project, RFE is well-suited because our 20 input features vary in predictive strength: features like `anxiety_level`, `depression`, `bullying`, and `future_career_concerns` show strong correlation with stress, while others contribute less. We use a Random Forest estimator in RFE – ideal for our numerical, mildly imbalanced data and its non-linear feature interactions. Our dataset's high quality (no missing or duplicate records, balanced targets) supports reliable feature ranking. RFE works particularly well after PCA by removing residual redundancy, helping reduce dimensionality while retaining essential predictors. For this analysis, we selected 10 features to optimize simplicity and prevent overfitting.

```

1 from sklearn.ensemble import RandomForestClassifier
2 from sklearn.feature_selection import RFE
3 # RFE setup with Random Forest (selecting 10 key features)
4 estimator = RandomForestClassifier(random_state=42)
5 rfe = RFE(estimator=estimator, n_features_to_select=10)
6 X_rfe = rfe.fit_transform(X, y)
7 selected_features = X.columns[rfe.support_]

```

RFE efficiently narrows down the feature set, enhancing both classification and clustering by concentrating on the most informative predictors. In our analysis, RFE selected features that correspond with inter-correlation clusters in the data (e.g., academic factors such as `study_load`), frequently retaining the strongest variable from each domain – such as `bullying` from social factors and `anxiety_level` from psychological factors. This targeted selection



increases the predictive power of models, as demonstrated in prior educational data mining research where using RFE-selected features led to improved classification accuracy for student stress prediction.

## 5 Results and Analysis

### 5.1 Classification Results

### 5.2 Clustering Results

### 5.3 Model Comparison