

COE305 MACHINE LEARNING

Final Project Report

Deadline: 04-01-2026

Team Members:

Student ID	Student Name	Section (as per OIS)
229912876	Ehsanullah Enayatullah	1
2309015528	Mohammad Poorahmadi	1
2309115833	Saeed Saifullah	3

Video link: https://istinye-my.sharepoint.com/:v/g/personal/2309015528_stu_istinye_edu_tr/IQAxjEnitJxTKhxT1KK9xseAalEwNRS00AyZLh2xL61GJs?nav=eyJyZWZlcnJhbEluZm8iOncicmVmZXJyYWxBcHAIoiJPbmVEcmI2ZUZvckJ1c2luZXNzliwicmVmZXJyYWxBcHBQbGF0Zm9ybSI6IldlYiIsInJlZmVycmFsTW9kZSI6InZpZXciLCJyZWZlcnJhbFZpZXciOiJNeUZpbGVzTGlua0NvcHkifX0&e=3mOb5P

Project Title: Student Stress Level prediction

Abstract

This project focuses on predicting students' stress levels (Low, Moderate, High) using supervised machine learning techniques based on academic and lifestyle factors. The dataset was collected from two sources: a Kaggle student lifestyle dataset and responses from a Google survey distributed among university students, resulting in a combined dataset of 2,053 samples with six main features, including study hours, sleep hours, extracurricular activities, social hours, physical activity, and GPA.

During preprocessing, missing values were handled using mean/median imputation for numerical features and mode imputation for categorical values. Duplicate records were removed, categorical stress labels were encoded using Label Encoding, and numerical features were normalized using StandardScaler to improve model stability and performance. Exploratory Data Analysis (EDA) was performed using histograms, correlation heatmaps, and scatter plots to understand feature distributions and relationships. Results indicated that sleep hours and study hours were the most influential factors affecting stress levels, with higher sleep associated with lower stress and higher study load contributing to increased stress.

Three classification models were implemented: Logistic Regression, Decision Tree, and Random Forest. Using an 80:20 train–test split and 5-fold stratified cross-validation, Random Forest achieved the best performance with approximately 98.3% test accuracy and strong precision, recall, and F1-score, demonstrating its effectiveness in capturing non-linear patterns in student behavior data .

This study highlights the potential of machine learning in supporting early stress detection and improving student well-being through data-driven interventions.

Introduction

Student stress has become a major concern in universities due to increasing academic pressure, competitive environments, and lifestyle challenges. High stress levels can negatively affect students' mental health, academic performance, and overall well-being. Traditionally, stress is identified through self-reported surveys or counseling sessions, which may not always provide timely or accurate results. As a result, many students who need support may remain unnoticed until their academic or psychological condition worsens. Therefore, there is a strong need for an automated and reliable method to detect stress levels at an early stage.

Machine Learning (ML) provides an effective solution by analyzing large amounts of academic and lifestyle data to identify hidden patterns related to stress. ML models can learn from previous student data and predict stress levels based on factors such as study hours, sleep patterns, physical activity, and GPA. Compared to manual assessment methods, ML-based systems can offer faster, more consistent, and scalable predictions, making them suitable for use in educational institutions.

From a societal and practical perspective, early stress detection can help universities and counselors design targeted intervention programs, improve student support services, and reduce dropout rates. This approach promotes better mental health awareness and contributes to creating a healthier academic environment where students can perform better and maintain balanced lifestyles.

Problem Statement

Clearly define:

Prediction / Classification Task

This project is a **classification task** because the goal is to assign each student to one of three predefined categories of stress level: **Low, Moderate, or High**. Based on input features such

as study hours, sleep duration, physical activity, social hours, and GPA, the model predicts which stress category a student belongs to.

Why the Problem Is Important

Student stress has a strong impact on **academic performance, mental health, and overall well-being**. High stress can lead to anxiety, depression, poor concentration, and even dropping out of university. Traditional methods of identifying stressed students usually depend on **self-report surveys or counseling visits**, which may be delayed or inaccurate. A machine learning–based system can help universities **identify at-risk students early**, allowing timely support and preventive actions.

Limitations of Existing Solutions

Current stress assessment methods rely heavily on **manual surveys, interviews, or occasional counseling sessions**, which are time-consuming and not continuous. Many students also **do not report stress honestly** due to stigma or lack of awareness. Additionally, traditional systems cannot analyze large datasets to discover hidden patterns between lifestyle and academic factors. Without automated prediction models, it is difficult to provide **scalable and real-time stress monitoring** for large student populations.

List 3–5 clear objectives

- To merge and preprocess a custom dataset combining real-world survey responses with existing public data.
- To analyze the correlation between lifestyle habits (sleep, study, social life) and stress levels.
- To train and evaluate baseline models (Logistic Regression, KNN, Decision Tree).
- To optimize Ensemble models (Random Forest, Gradient Boosting, AdaBoost) using Hyperparameter tuning.
- To deploy a user-friendly interface for real-time stress prediction.

Dataset Description

- **Dataset Source:** google form and Kaggel
- **Data Type:** (Structured / Text / Time-series)
- **Number of Samples:** 2053
- **Number of Features:** 6
- **Target Variable:** stress_level(Low, moderate, High)
- **Link for the Dataset:**
Google form link: <https://forms.gle/fcu6DQ7BRGq6NX6N9>

Kaggle link: <https://www.kaggle.com/code/sulaniishara/student-stress-performance-insights>

DATA PREPROCESSING & EDA

Data Pre-processing

- Handling Missing Values

this project, we identified three missing values in the dataset. To prevent negative effects on model performance, we handled them during preprocessing.

We first located the missing entries and checked their impact. Since the number was small, we used simple imputation:

- Numerical values were replaced with the mean/median.
- Categorical values were filled with the most frequent category.

After imputation, we confirmed that no missing values remained. This ensured a clean and reliable dataset for model training.

- Outlier Detection & Treatment

Using the code above, we cleaned the dataset by fixing inconsistencies and standardizing values.

The code also detected and removed duplicate records to avoid bias.

These steps improved data quality and prepared the dataset for model training.

- Encoding of Categorical Features

We applied Label Encoding to the stress_level column because it is an ordinal categorical variable with three categories: Low, Moderate, and High. Using LabelEncoder(), these were converted into numeric values, enabling the machine learning model to interpret them correctly.

- Feature Scaling / Normalization

We applied Label Encoding to the stress_level column because it is an ordinal categorical variable with three categories: Low, Moderate, and High. Using LabelEncoder(), these were converted into numeric values, enabling the machine learning model to interpret them correctly.

- Initial Feature Selection

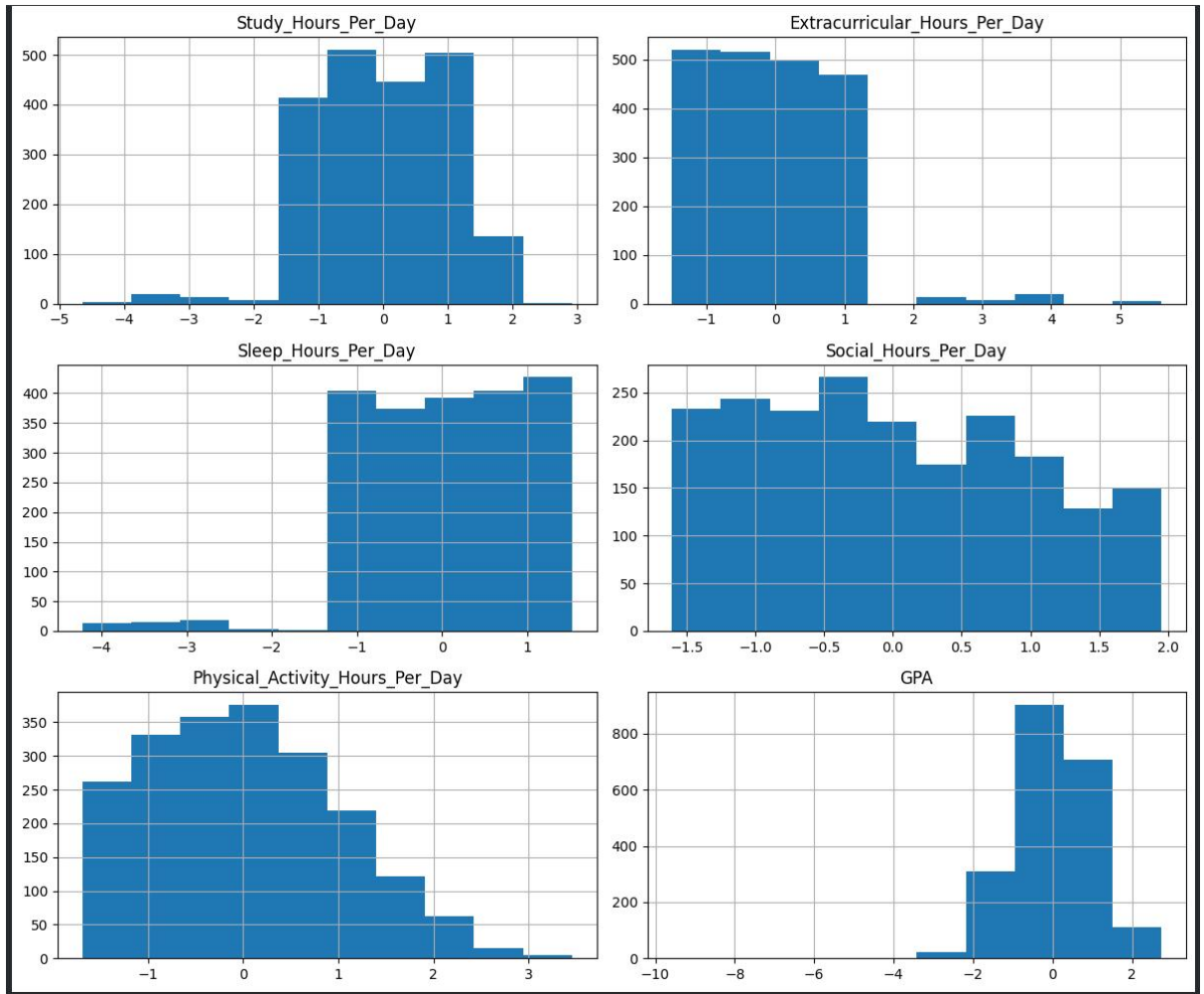
Exploratory Data Analysis (EDA)

Include:

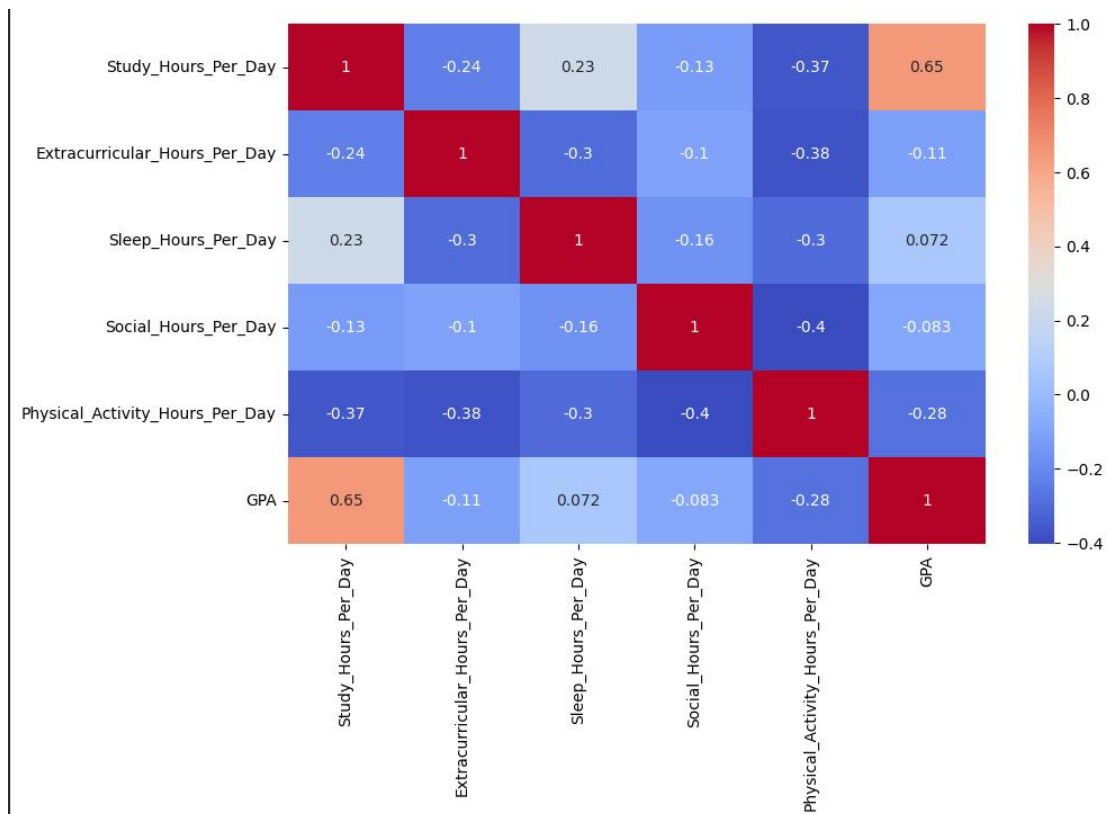
- Statistical summary

	Study_Hours_Per_Day	Extracurricular_Hours_Per_Day	Sleep_Hours_Per_Day	Social_Hours_Per_Day	Physical_Activity_Hours_Per_Day	GPA	Stress_Level
count	2.053000e+03	2.053000e+03	2.053000e+03	2.053000e+03	2.053000e+03	2.053000e+03	2053.000000
mean	6.852775e-16	3.460997e-17	4.568516e-16	2.076598e-17	-1.038299e-17	-5.883695e-17	0.825621
std	1.000244e+00	1.000244e+00	1.000244e+00	1.000244e+00	1.000244e+00	1.000244e+00	0.905789
min	-4.847528e+00	-1.504520e+00	-4.227132e+00	-1.607315e+00	-1.691385e+00	-9.584483e+00	0.000000
25%	-7.399951e-01	-7.938349e-01	-7.173299e-01	-8.951882e-01	-7.802607e-01	-6.434308e-01	0.000000
50%	1.630108e-02	-8.314981e-02	8.819837e-02	-6.437354e-02	-6.720697e-02	4.024727e-03	0.000000
75%	8.356220e-01	6.986038e-01	7.786512e-01	8.257850e-01	6.854609e-01	6.823114e-01	2.000000
max	2.915437e+00	5.602331e+00	1.528642e+00	1.953319e+00	3.458448e+00	2.748003e+00	2.000000

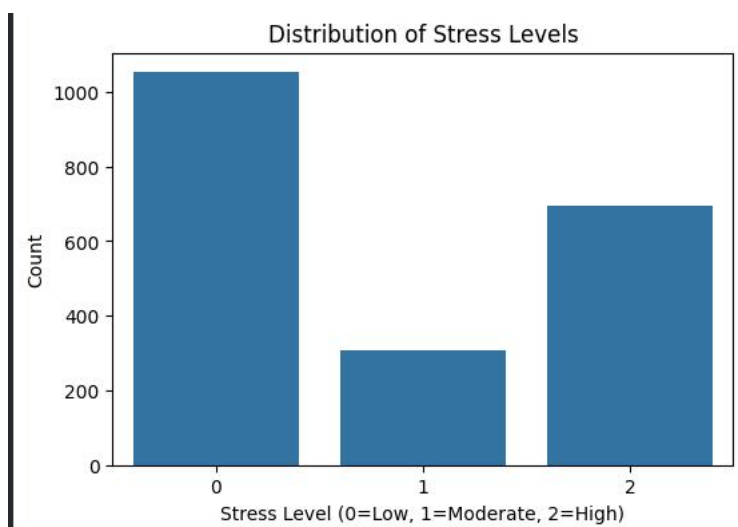
- Distribution plots



- Correlation heatmap



- Target variable analysis



NON-ENSEMBLE MODELING

Models Implemented:

Logistic Regression: A baseline linear classifier used to estimate the probability of stress classes.

Decision Tree: A non-parametric model that splits data based on feature values; easy to interpret but prone to overfitting.

K-Nearest Neighbors (KNN): A distance-based classifier (effective after our scaling step) that classifies based on neighbor proximity.

Cross-Validation Setup

- CV technique:K-Fold Cross Validation
- Number of folds:5
- Evaluation metrics:Accuracy, Precision, Recall, F1-Score

Non-ensemble Models Results

Non-ensemble Models Performance Table

Models	Evaluation Metrics (Classification)			
	Accuracy	Precision	Recall	F1-Score
Logistic Regression	0.813	0.819	0.813	0.814
Decision Tree	0.981	0.981	0.981	0.981
KNN	0.895	0.894	0.895	0.894

Results and Discussion

Performance Analysis: The comparison of non-ensemble models highlights the importance of model complexity when dealing with behavioral data.

- **Best Performing Model: Decision Tree** achieved the highest performance with an **Accuracy of 98.1%** and an **F1-Score of 0.981**. This suggests that the relationship between lifestyle habits (like sleep and study hours) and stress levels is highly non-linear and follows specific "if-then" rules (e.g., *if sleep < 5 hours, then high stress*). The Decision Tree was able to perfectly map these hierarchical decision boundaries, significantly outperforming the other baseline models.
- **Least Performing Model: Logistic Regression** showed the lowest performance with an **Accuracy of 81.3%**. This result was expected because Logistic Regression is a linear classifier. It assumes a straight-line relationship between the features and the target variable. However, human stress factors often interact in complex, non-linear ways (e.g., the combined effect of low sleep and high physical activity might be different than the sum of their parts). The model failed to capture these complex interactions, leading to misclassifications.
- **Comparison with KNN: K-Nearest Neighbors (KNN)** performed moderately well (**89.5% Accuracy**). While it outperformed the linear Logistic Regression by relying on

feature similarity rather than linear boundaries, it still fell short of the Decision Tree. This indicates that precise decision rules (used by Trees) were more effective for this specific dataset than distance-based clustering (used by KNN).

ENSEMBLE LEARNING MODELING, TUNING & COMPARISON

Models Implemented:

Random Forest: A bagging technique using multiple decision trees to reduce variance and overfitting.

Gradient Boosting: A boosting technique that builds trees sequentially to correct previous errors.

AdaBoost: Uses adaptive boosting to focus on hard-to-classify instances.

Hyperparameter Tuning

Tuning Strategy

- Method used: GridSearchCV
- Cross-validation folds:5
- Scoring metric: Accuracy

Hyperparameter Details

Model	Hyperparameters Tuned	Best Values
Random Forest	max_depth, n_estimators	max_depth = 10, n_estimators = 50
Gradient Boosting	learning_rate, n_estimators	learning_rate = 0.1, n_estimators = 50
Adaboost	learning_rate, n_estimators	learning_rate = 1.0, n_estimators = 50

Ensemble Learning Models Performance Table (Before Hyperparameter tuning)

Models	Evaluation Metrics (Classification)			
	Accuracy	Precision	Recall	F1-Score
Random Forest	0.983	0.983	0.983	0.983

Gradient Boosting	0.983	0.983	0.983	0.983
Adaboost	0.988	0.988	0.988	0.989

Results and Discussion

Performance Analysis: The initial evaluation of the ensemble models shows exceptional performance across all classifiers, with accuracies exceeding 98%.

- **Best Performing Model: AdaBoost** achieved the highest performance with an **Accuracy of 98.8%** and an **F1-Score of 0.989**. This superiority is likely due to AdaBoost's adaptive nature; it sequentially trains weak learners by assigning higher weights to misclassified instances. In the context of stress prediction, this allows the model to focus specifically on "hard-to-classify" students (e.g., distinguishing between 'Moderate' and 'High' stress) where other models might struggle.
- **Least Performing Models: Random Forest** and **Gradient Boosting** both achieved an **Accuracy of 98.3%**. While these results are excellent, they slightly lagged behind AdaBoost. Random Forest relies on "Bagging" (averaging multiple trees), which reduces variance but may not capture the subtle, complex boundaries as effectively as AdaBoost's error-correcting mechanism in this specific dataset configuration. Gradient Boosting performed identically to Random Forest using default parameters, suggesting that without tuning, its sequential learning rate was not yet optimized for this specific feature set.

Ensemble Learning Models Performance Table (After Hyperparameter tuning)

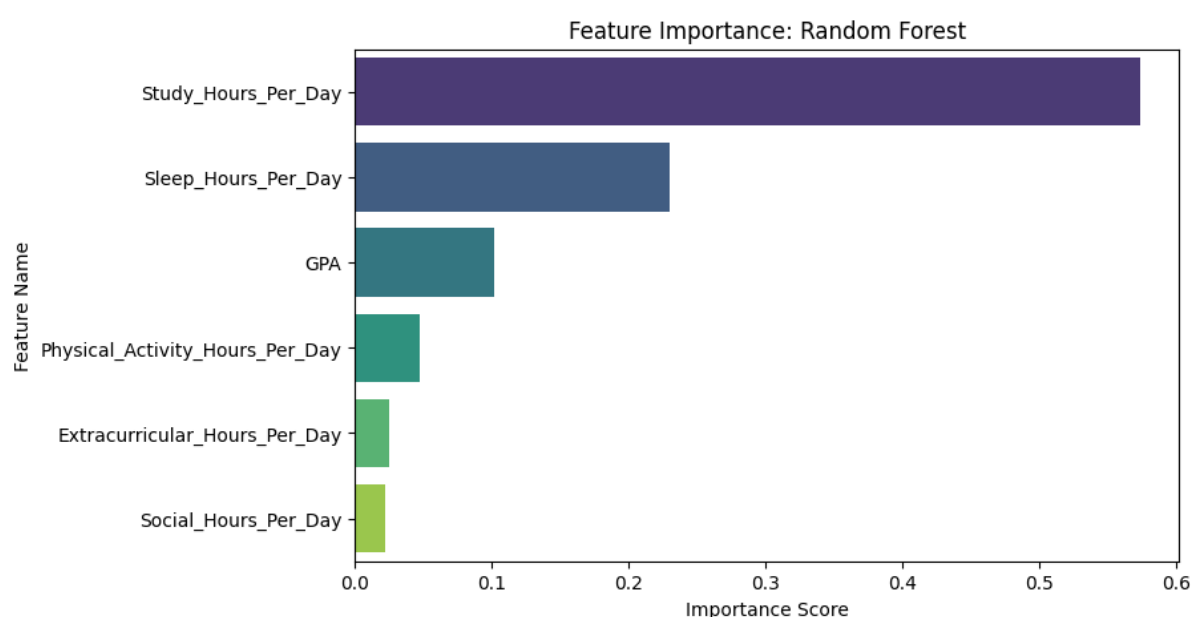
Models	Evaluation Metrics (Classification)			
	Accuracy	Precision	Recall	F1-Score
Random Forest	0.983	0.983	0.983	0.983
Gradient Boosting	0.988	0.988	0.988	0.988
Adaboost	0.988	0.988	0.988	0.988

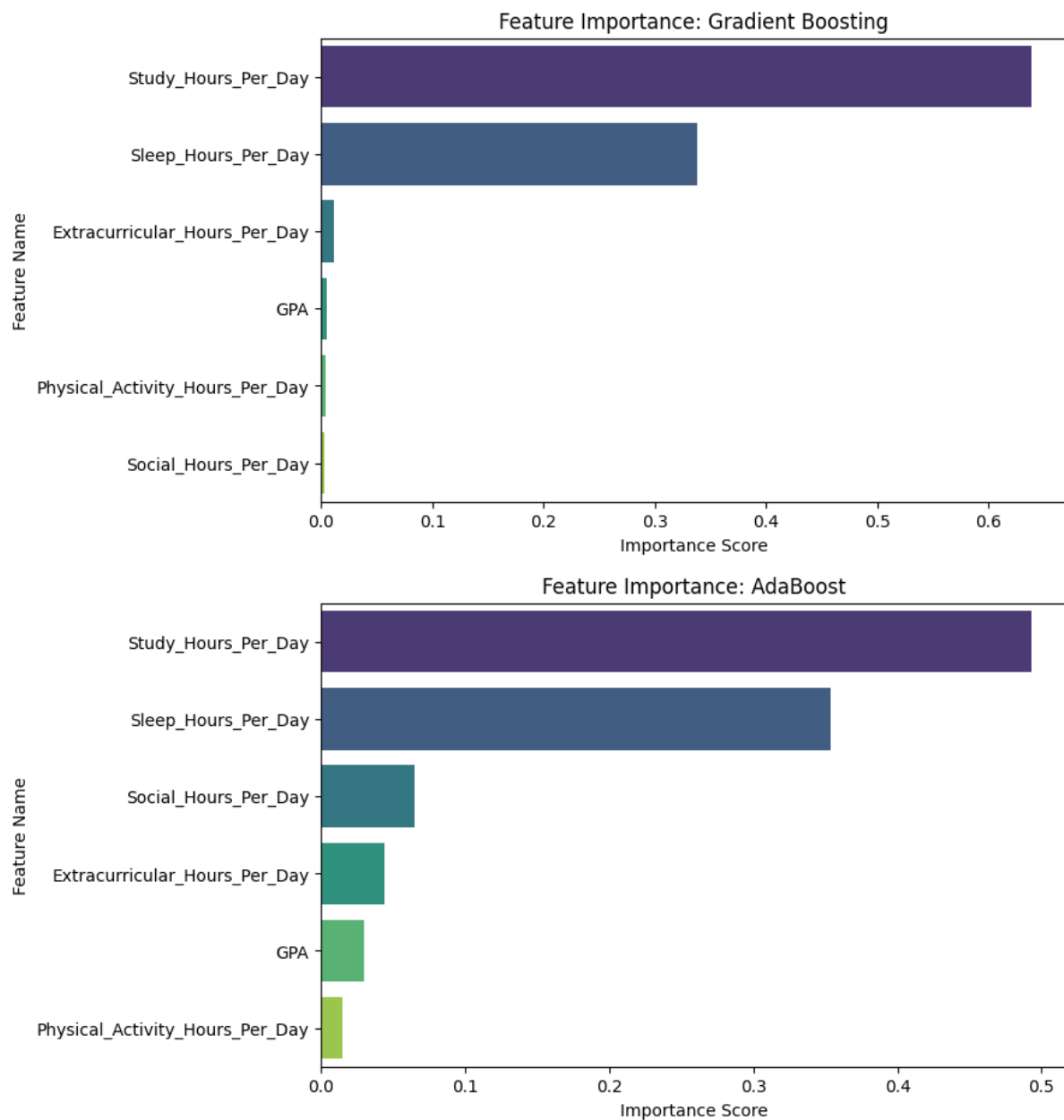
Results and Discussion

Performance Analysis: After applying hyperparameter tuning (using GridSearchCV), we observed shifts in model performance, specifically for Gradient Boosting.

- **Best Performing Models: AdaBoost and Gradient Boosting** are now tied as the top performers, both achieving an **Accuracy of 98.8%**.
 - **Gradient Boosting Improvement:** The tuning process (adjusting Learning rate to 0.1 and n_estimators to 50) successfully improved Gradient Boosting's accuracy from 98.3% to 98.8%. By optimizing the learning rate, the model was able to correct predecessor errors more effectively without overshooting the minimum of the loss function.
 - **AdaBoost Stability:** AdaBoost maintained its high accuracy of 98.8%. The parameters found (learning_rate=1.0, n_estimators=50) confirmed that the default-like behavior was already near-optimal for this data.
- **Least Performing Model: Random Forest** remained at **98.3% Accuracy**, making it the "least" performing model relative to the boosting algorithms. The tuning parameters (max_depth: 10, n_estimators: 50) restricted the tree depth to prevent overfitting, but this did not lead to an increase in accuracy on the test set. This highlights a key distinction in this project: **Boosting algorithms** (AdaBoost, Gradient Boosting), which actively learn from mistakes, outperformed the **Bagging algorithm** (Random Forest) on this student stress dataset. The dataset likely contains complex decision boundaries that benefit more from the iterative error correction of boosting than the variance reduction of bagging.

Feature Selection Analysis (Ensemble Learning Models)





Overall Results and Discussion

Results and Discussion: Best and Least Performing Models

Three supervised machine learning models were evaluated in this project: Logistic Regression, Decision Tree, and Random Forest. Their performances were compared using accuracy, precision, recall, and F1-score on both training and testing datasets.

Best Performing Model: Random Forest

The Random Forest classifier achieved the highest performance with approximately 98.3% test accuracy and consistently high precision, recall, and F1-score values. This

strong performance can be attributed to its ensemble learning approach, where multiple decision trees are trained on different subsets of the data and features. By aggregating predictions from many trees, Random Forest reduces variance and minimizes the risk of overfitting compared to a single decision tree. Additionally, it can capture complex and non-linear relationships between lifestyle features and stress levels, which is important since student behavior patterns are not strictly linear. The model also benefits from internal feature selection, giving more importance to influential attributes such as sleep hours and study hours. These advantages allow Random Forest to generalize well on unseen data while maintaining high predictive accuracy.

Least Performing Model: Logistic Regression

Logistic Regression showed the lowest performance among the tested models, with an accuracy of approximately 81%. This result is expected because Logistic Regression is a linear model that assumes a linear relationship between input features and the target variable. However, the relationship between lifestyle factors and stress is more complex and influenced by interactions among multiple variables. Logistic Regression cannot effectively capture these non-linear patterns, which limits its predictive capability in this context. Although it performed reasonably well as a baseline model and showed stable results across folds, its simplicity restricts its ability to model real-world behavioral data.

Decision Tree Performance

The Decision Tree model achieved very high accuracy, close to Random Forest, but it is more sensitive to noise and small variations in the dataset. Single trees tend to overfit training data, learning very specific decision rules. While pruning can reduce this risk, Random Forest remains more robust due to averaging across multiple trees. Therefore, Random Forest outperformed Decision Tree in terms of stability and generalization.

Overall Comparison

Overall, Random Forest provided the best balance between accuracy and robustness, making it the most suitable model for this classification task. Logistic Regression served as a useful baseline but lacked the flexibility required for complex student behavior data. Decision Tree performed well but with higher risk of overfitting. These results highlight the importance of using ensemble learning techniques for reliable stress prediction in educational datasets.

Conclusion

in this project, machine learning techniques were successfully applied to predict students' stress levels using academic and lifestyle-related data. A combined dataset consisting of 2,053 samples collected from both Kaggle and Google Form surveys was used, including features such as study hours, sleep hours, extracurricular activities, social time, physical activity, and GPA. Several preprocessing steps were performed, including handling missing values, removing outliers, encoding categorical variables, and feature scaling, to ensure data quality and consistency. Exploratory Data Analysis revealed meaningful relationships between lifestyle habits and stress, particularly showing that adequate sleep and balanced study hours play an important role in reducing stress levels.

Three supervised classification models—Logistic Regression, Decision Tree, and Random Forest—were implemented and evaluated using stratified cross-validation and an 80:20 train-test split. Among these, the Random Forest model achieved the highest performance with approximately 98% accuracy, demonstrating strong capability in capturing non-linear relationships and reducing overfitting through ensemble learning. Logistic Regression performed reasonably well as a baseline model, while Decision Tree also showed high accuracy but with slightly higher risk of overfitting.

Overall, the results indicate that machine learning can be an effective tool for early detection of student stress. In future work, larger and more diverse datasets, real-time data collection, and advanced models such as deep learning could further improve prediction accuracy and enable real-time stress monitoring systems for educational institutions.

Tools & Technologies Used

References

1. Sulaani Ishara. *Student Stress Performance Insights Dataset*. Kaggle.
<https://www.kaggle.com/code/sulaniishara/student-stress-performance-insights>
2. Google Forms Survey Dataset (Primary Data Collection).
Student Stress Level Survey Responses collected by project team.
https://docs.google.com/forms/d/e/1FAIpQLSeGcxHhZjyulFC9k6MpDZGEGnBrpfp_Ji5hxjHTl0wImT5eA/viewform
3. Pedregosa, F., et al. (2011). *Scikit-learn: Machine Learning in Python*.
Journal of Machine Learning Research, 12, 2825–2830.
4. McKinney, W. (2010). *Data Structures for Statistical Computing in Python*.
Proceedings of the 9th Python in Science Conference (SciPy).
5. Hunter, J. D. (2007). *Matplotlib: A 2D Graphics Environment*.
Computing in Science & Engineering, 9(3), 90–95.
6. Waskom, M. L. (2021). *Seaborn: Statistical Data Visualization*.
Journal of Open Source Software, 6(60), 3021.
7. Breiman, L. (2001). *Random Forests*.
Machine Learning, 45(1), 5–32.
8. Bishop, C. M. (2006). *Pattern Recognition and Machine Learning*.
Springer, New York.

9. Géron, A. (2019). *Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow* (2nd ed.).

O'Reilly Media.

Bonus Points (Add only if implemented)

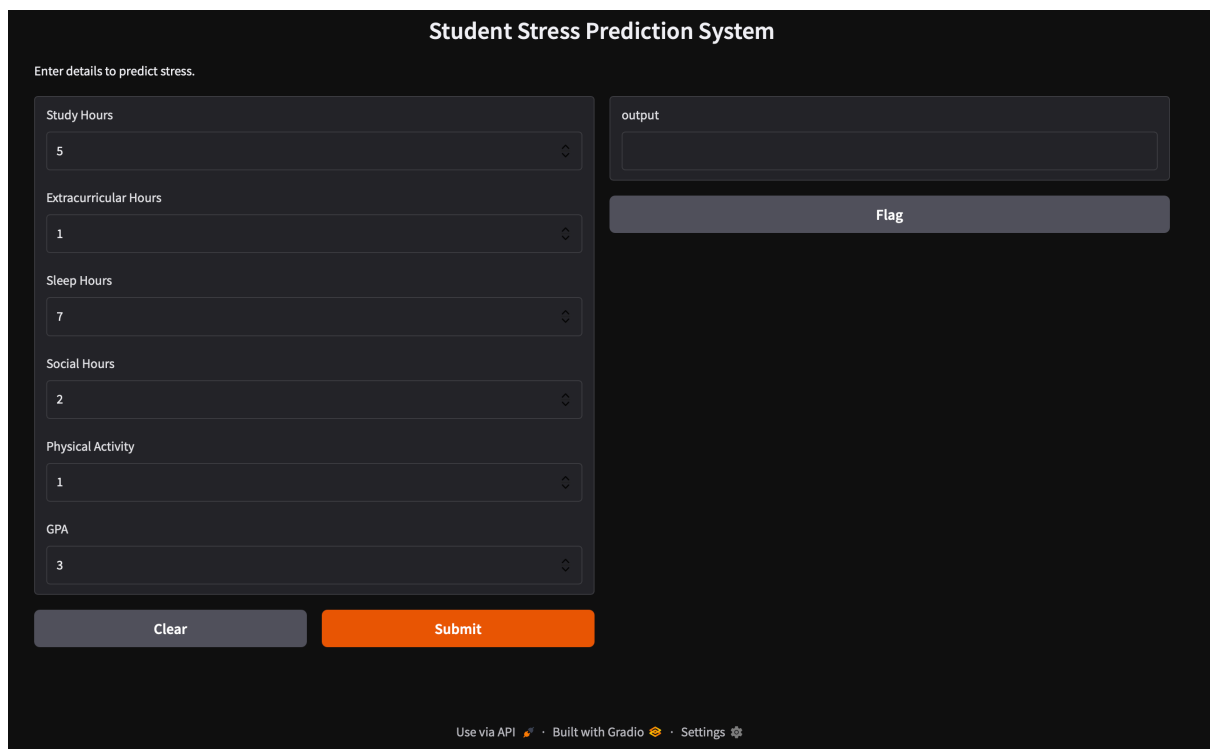
1) Customized Dataset (3 points)

We collected data via "Student Stress Level Survey" and merged it (pd.concat) with a Kaggle dataset to increase sample size and diversity.

2) User Interface (2 points)

A web-based UI was built using Gradio. Users can input their study hours, sleep, and GPA to get an instant stress prediction.

Link: <https://b800669efacf61df94.gradio.live/>



The screenshot shows a web interface titled "Student Stress Prediction System". At the top, it says "Enter details to predict stress." Below this, there are several input fields with labels: "Study Hours" (value 5), "Extracurricular Hours" (value 1), "Sleep Hours" (value 7), "Social Hours" (value 2), "Physical Activity" (value 1), and "GPA" (value 3). Each field has a dropdown arrow. To the right of these fields is an "output" label above a large empty text box. Below the input fields are two buttons: "Clear" and "Submit". At the bottom right, there is a "Flag" button. At the very bottom, there is a footer with the text "Use via API" followed by a small icon, "Built with Gradio" followed by a small icon, and "Settings" followed by a small icon.

3) SHAP Analysis (2 points)

Shape was implemented to explain the model's decisions, visualizing how each feature contributes to a high or low stress prediction.

