National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

Introduction to Data Science

DS 401

Semester Project

Educational Data Mining for Student Performance Prediction

BSDS 1-A

SUBMITTED TO: Ma'am Rabia Irfan

SUBMITTED BY:

Student Name	CMS ID
Hareem Fatima	466141
Fizza Kashif	466184
Misbah Shaheen	461898
Sana Khan Khitran	464597

Submission Date : 9/05/2025

Table of Contents

Introduction

Problem Statement

Predicting student performance using academic data by identifying key factors like attendance, study time, past grades, etc., using regression, statistical analysis, and decision trees.

Background

In the modern educational environment, vast amounts of data are generated daily from student activities, assessments, and demographics. Educational Data Mining is an emerging discipline that uses data mining techniques to explore these datasets. By analyzing this data, educators can better understand student's learning behaviors and academic trends, ultimately improving teaching strategies and outcomes.

Motivation

High dropout rates, poor academic performance, and lack of personalized learning paths are ongoing challenges in education. By predicting student performance early, educators can offer support where needed. The motivation behind this study is to use data-driven insights to improve student retention, enhance learning experiences, and support data-informed decision-making in educational institutions.

Dataset & Preprocessing

Dataset Overview

- 2392 records with 15 features
- Mixed data types (numeric, categorical, binary)
- Target variables: GPA (continuous) and GradeClass (categorical)

Key Features

Category	Features
Demographic	Age, Gender, Ethnicity
Academic	StudyTimeWeekly, Absences, GPA, GradeClass
Behavioral	Tutoring, ParentalSupport, Extracurricular Activities

Data Cleaning Process

1. **Handling Missing values**

   There was no missing data in the dataset.

2. **Invalid Data Removal:**

   Applied validity checks:

   - GPA range constrained to [0, 4]
   - Minimum student age set to 15 years
   - Removed duplicate records

Data Type Standardization:

- **Numeric:** Age, StudyTimeWeekly, Absences, GPA
- **Categorical:** Gender, Ethnicity, ParentalEducation

Feature Engineering

Created three new features:

Feature	Formula
TotalExtracurricular	Sum of Sports/Music/Volunteering
StudyEfficiency	GPA ÷ (StudyTimeWeekly + 1)
AttendanceRate	1 - (Absences ÷ 30)

Feature Transformation

- **Binning**
  - StudyTimeWeekly = Categories (Low/Medium/High/Very High)
- **One-Hot Encoding**
  - Applied to: Gender, Ethnicity, ParentalEducation

Outlier Treatment

- Used IQR method (1.5×IQR bounds) for:
  - StudyTimeWeekly
  - Absences
  - GPA
- Capped extreme values at percentile thresholds

Data Partitioning & Scaling

1. **Train-Test Split**

- We used Stratified sampling to maintain target distribution .
- 80% training (1914 records) and 20% testing (480 records)

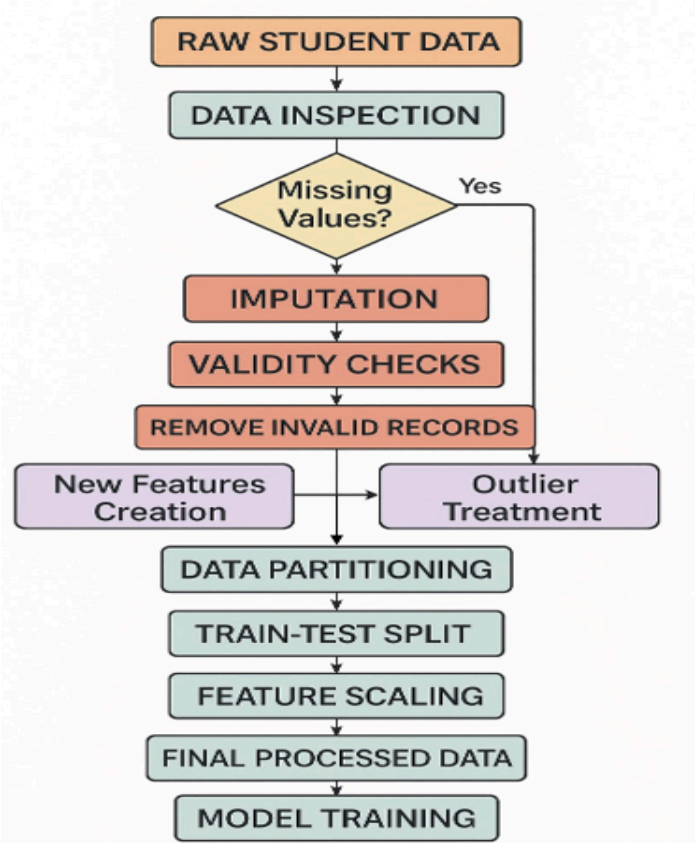
2. **Feature Scaling**

Applied StandardScaler (mean=0, variance=1) to all continuous numerical features

Output Datasets:

File	Contents	Records
Cleaned_Student_Performance_Data.csv	Fully processed Dataset	2392
X_train.csv	Scaled training features	1914
X_test.csv	Scaled test features	480
Y_train.csv	Training labels (GPA/GradeClass)	1914
Y_test.csv	Test labels (GPA/GradeClass)	480

Data Preprocessing Workflow:



Exploratory Data Analysis (EDA)

1. Introduction

Exploratory Data Analysis (EDA) is a critical step in any data-driven project, aimed at summarizing the main characteristics of a dataset and uncovering patterns, relationships, and anomalies. In this study, EDA is used to investigate the factors influencing student performance, such as demographic attributes, parental support, extracurricular activities, and study habits.

2. Dataset Overview

The dataset consists of various features representing students' academic and personal backgrounds, including GPA, gender, ethnicity, parental education, study time, and involvement in extracurricular activities. Initial exploration revealed a well-structured dataset with both numerical and categorical features.

3. Summary Statistics

Descriptive statistics provided a snapshot of central tendency and dispersion for the numerical features. The average GPA was found to be above 2.5, with some students performing at the extremes (near 0 or 4.0). Categorical data summaries revealed balanced representation across most features, such as gender and parental education levels, with slight skews in a few categories.

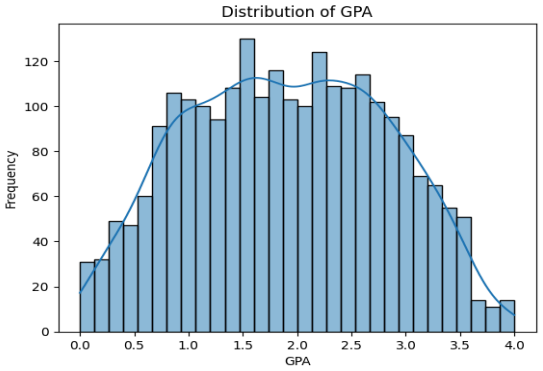
```
0s
# Summary statistics for numerical columns
print("Descriptive statistics:")
print(df.describe())
```

```
Descriptive statistics:
         StudentID  StudyTimeWeekly     Absences          GPA
count  2392.000000      2392.000000  2392.000000  2392.000000
mean   2196.500000         9.771992    14.541388     1.906186
std     690.655244         5.652774     8.467417     0.915156
min    1001.000000         0.001057     0.000000     0.000000
25%    1598.750000         5.043079     7.000000     1.174803
50%    2196.500000         9.705363    15.000000     1.893393
75%    2794.250000        14.408410    22.000000     2.622216
max    3392.000000        19.978094    29.000000     4.000000
```

4. Univariate Analysis

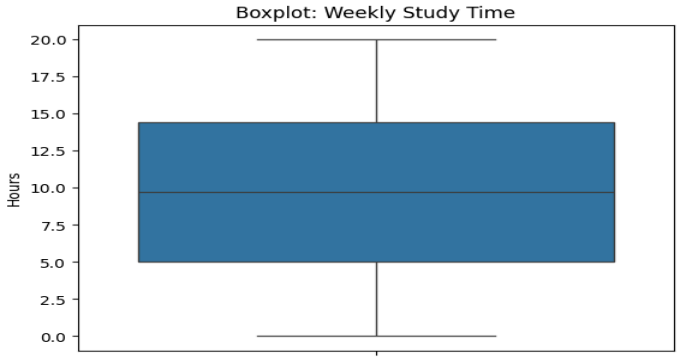
GPA Distribution

The GPA distribution was found to be slightly left-skewed, indicating that most students performed above average, with fewer low-performing individuals.



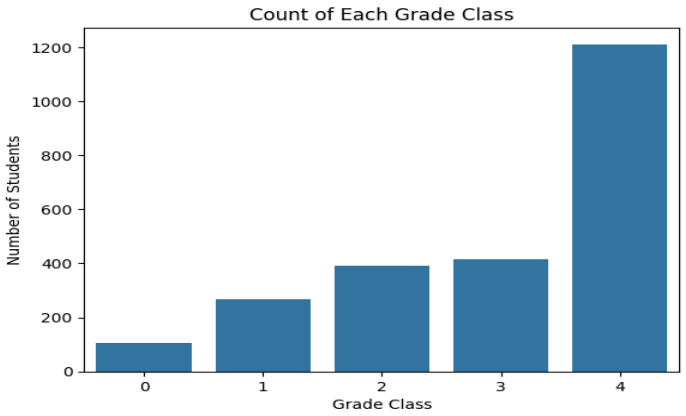
Study Time

Boxplots showed variability in weekly study hours, with a few students investing exceptionally high or low amounts of time. This variance highlighted the importance of analyzing study habits as a predictor of academic success.



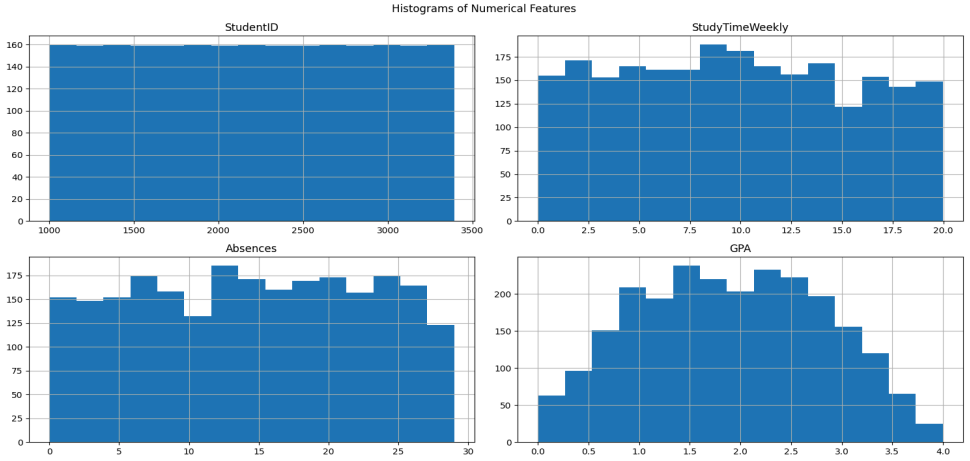
Grade Class Distribution

The dataset was imbalanced in terms of grade classification, with most students falling into average or high-performing categories, and fewer classified as failing. This imbalance is a consideration for future modeling stages.



5. Distribution of Features

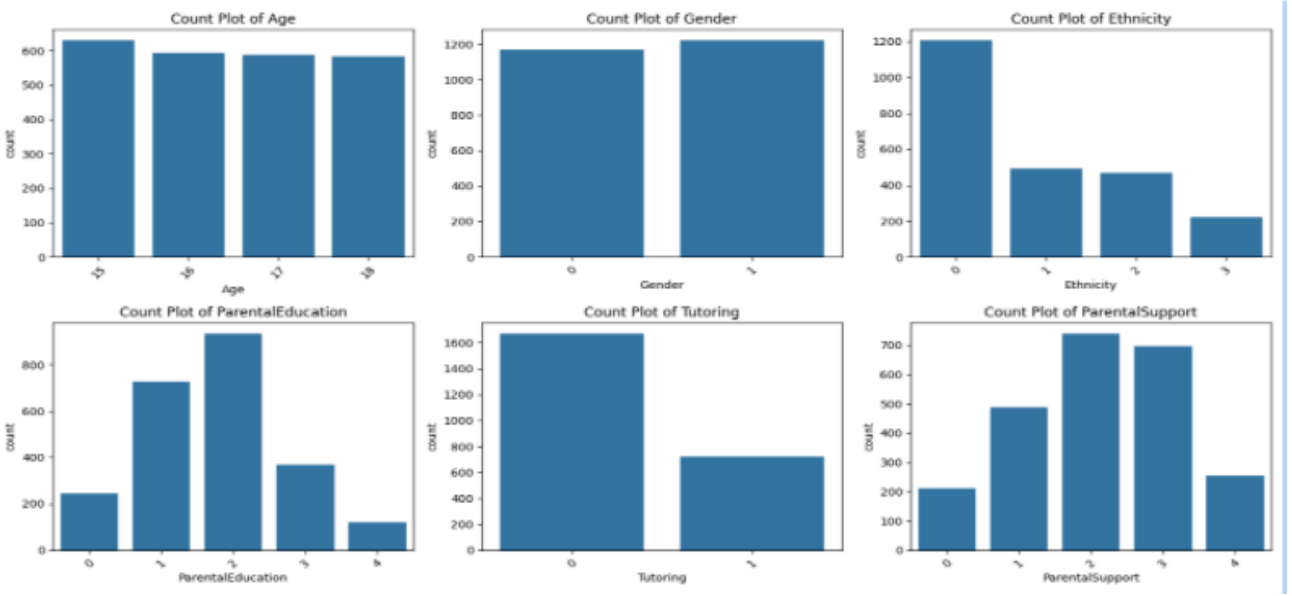
Histograms and boxplots were used to understand the spread and outliers in numerical data such as GPA, study time, and attendance. The visualizations confirmed that while most students exhibit typical behavior, a subset shows outlying characteristics, particularly in terms of low attendance and extremely high or low study times.

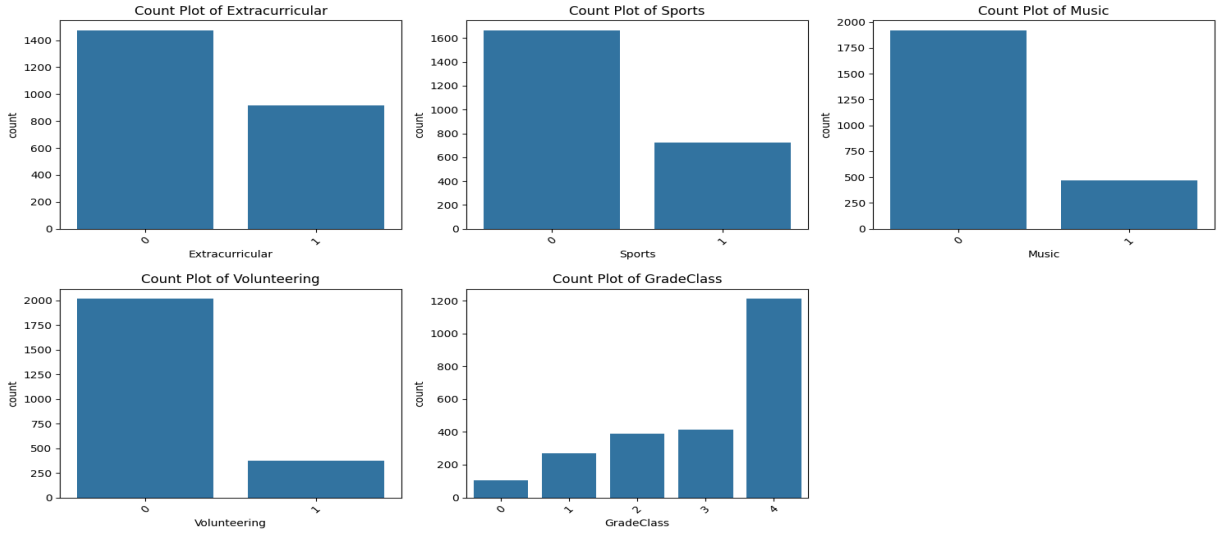


6. Categorical Feature Analysis

Count plots and pie charts were used to visualize the distribution of categorical variables. Key takeaways include:

- Gender and ethnicity were fairly balanced.
- A substantial portion of students participated in extracurricular activities such as sports, music, or volunteering.
- Parental education varied, with a notable concentration in secondary and post secondary levels.

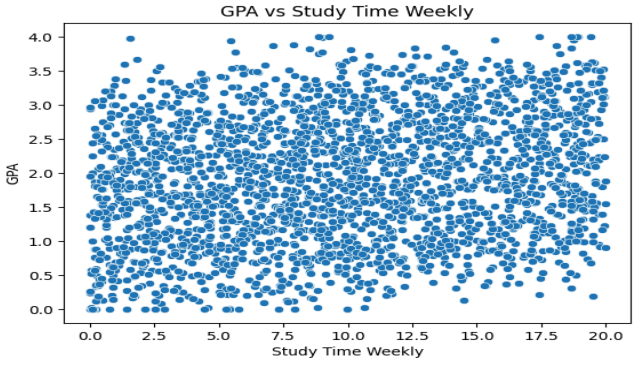




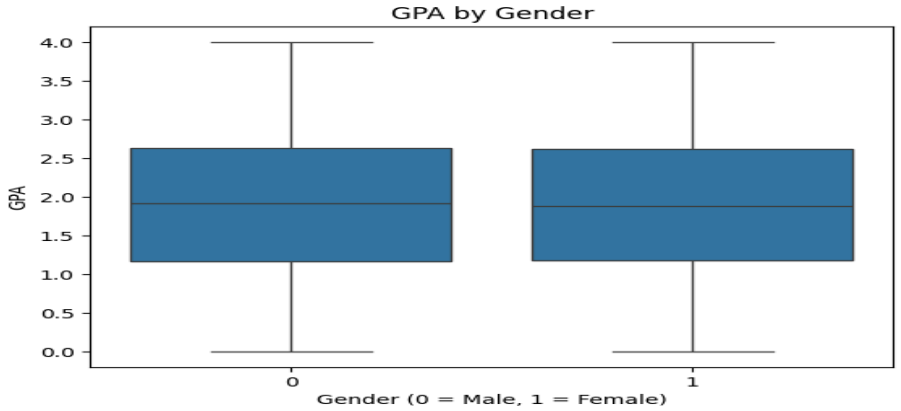
7. Bivariate Analysis

Relationships between GPA and other features were analyzed using scatter plots and box plots.

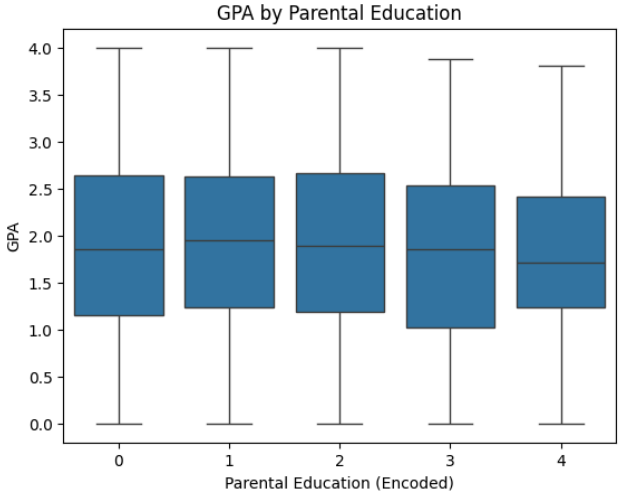
- GPA and Study Time: A positive trend was observed—students who spent more time studying tended to have higher GPAs, up to a certain threshold.



- GPA by Gender: A slight performance difference was detected between genders, with one group showing marginally higher average scores.



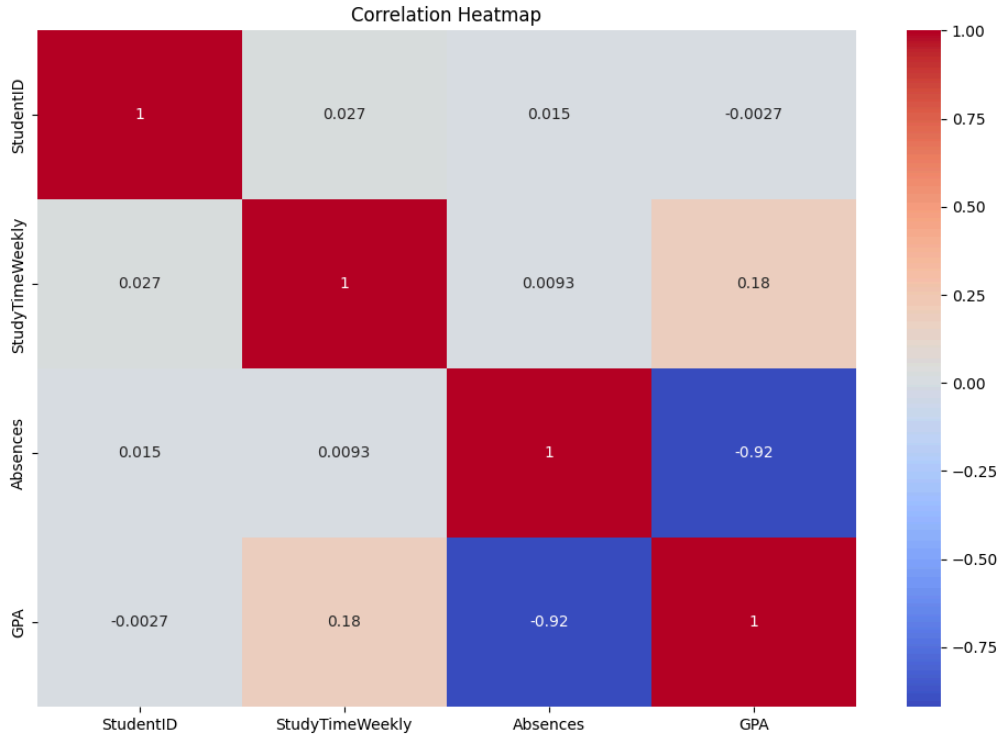
- GPA and Parental Education: Students with parents who had higher education levels tended to achieve higher GPA scores, highlighting the influence of a supportive educational environment at home.



8. Correlation Analysis

A correlation heatmap revealed meaningful relationships between features:

- GPA was positively correlated with weekly study time and parental support.
- Some variables, such as volunteering and music participation, had weak or negligible correlations with GPA, suggesting lower predictive power.



9. Covariance Analysis

Covariance analysis further explored the directional relationships between numerical variables. Features like GPA and study time exhibited positive covariance, indicating they increase together, though the magnitude of change varies.

10. Categorical Cross-tabulation

Cross-tabulations between categorical features (e.g., gender and grade class) revealed patterns in how different groups performed. For instance, certain grade classes had higher concentrations of one gender, prompting deeper exploration of group-specific academic trends.

11. Automated EDA Profiling

An automated EDA report generated using YData Profiling provided a comprehensive summary, including:

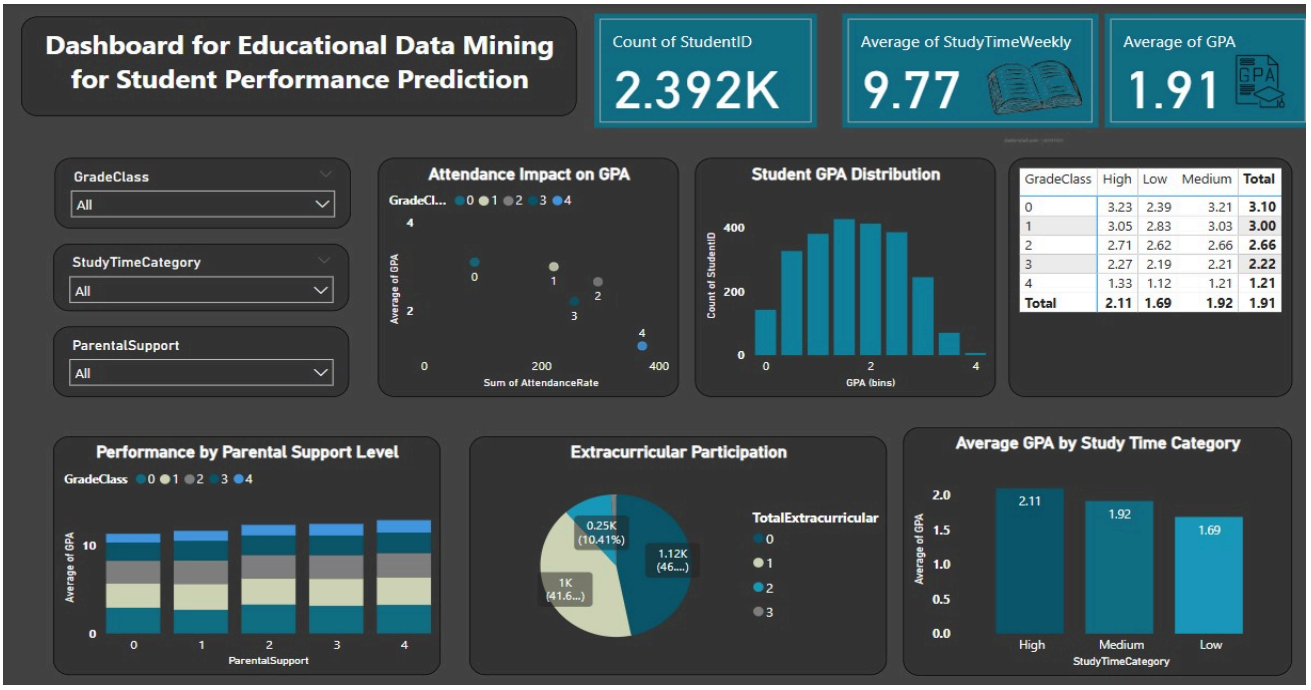
- Feature distributions
- Data quality alerts (e.g., imbalance, missing values)
- Correlation maps
- Interaction effects between variables

Insight	Description
GPA Distribution	The majority of students perform above average, with a few outliers on the lower end.
Study Time Impact	Increased study time generally leads to a better GPA.
Gender Differences	Small GPA variance between genders was observed.
Parental Education	Strong correlation with student GPA—higher parental education boosts performance.
Extracurriculars	Mixed influence; benefits may depend on the balance with academics.
Class Imbalance	GradeClass shows skewed distribution; modeling must account for this.
Data Quality	No missing data and few duplicates; overall data is clean and reliable.

The EDA process provided valuable insights into the academic and personal factors influencing student performance. Key relationships—such as the impact of study habits, parental background, and extracurricular involvement—were identified and quantified. These findings not only support feature engineering and model design but also offer practical implications for educational support systems. The dataset is now well-understood, clean, and ready for predictive modeling.

Dashboard:

We built a Power BI dashboard with predictive analytics to track student performance trends using attendance, study time, and parental support metrics.



1. **Dataset Overview**
   - 2,392 students analyzed
   - **Avg. weekly study time:** 9.77 hours
   - **Avg. GPA:** 1.91 (below average on a 0-4 scale)
2. **Performance Trends**
   - Most students cluster in low GPA ranges (400 in the lowest bin, only 2-4 in higher bins).
   - Higher study time correlates with better GPA (High: approx. 2.0, Medium: approx.1.5, Low: approx 1.0).
3. **Attendance Impact**
   - Data suggests attendance rates may influence GPA.
4. **Parental Support**
   - Higher parental support levels (0-7 scale) tend to link with better academic performance.
5. **Extracurricular Activities**
   - Very low participation (0.05%-1.12%), affecting student engagement.

This data could help educators identify at-risk students and develop targeted activities to improve study habits, attendance, and parental engagement.

Modeling Approach

Chosen models

This project involves both **regression** and **classification** tasks. The regression models are used to predict **continuous student GPA**, while classification models are designed to categorize students into **predefined grade classes**.

Regression Modeling Approach

To predict students' GPA several regression models were implemented using **scikit-learn**, along with a final ensemble approach to improve prediction performance. The process included preprocessing, individual model training, and ensemble learning.

Preprocessing

- **Categorical Encoding**:
  Categorical features were encoded using one-hot encoding. To avoid mismatched columns after encoding, both datasets were aligned.
- **Feature Scaling (for KNN)**:
  Since KNN is sensitive to feature magnitude, numeric features were standardized before training the KNN model. This was applied using a StandardScalar().

Trained Models

1. **Linear Regression**
   A baseline model used to evaluate linear relationships.
2. **Random Forest Regressor**
   A robust ensemble of 100 decision trees.
3. **Decision Tree Regressor**
   A single tree with maximum depth of 5 to avoid overfitting.
4. **k-Nearest Neighbors (KNN) Regressor**
   A distance-based model using k=3.
5. **Ensemble Model – Voting Regressor**
   To combine the strengths of all models, we built an ensemble using Voting Regressor. This averages the predictions from all base regressors. Each model is trained independently, and their outputs are combined by calculating the mean. This approach helps reduce variance and generally improves robustness.
6. **Ensemble Model -Stacking Regressor**
   The Stacking Regressor is a more advanced ensemble technique. It uses the outputs of several base regressors as input features to a meta-model (Linear Regression). This meta-model learns the optimal way to combine the base predictions to minimize overall error.

Classification Modeling Approach

In addition to predicting GPA via regression, classification models were developed to predict discrete class labels or grade categories. This was treated as a multiclass classification problem, where the target variable represented student performance classes.

Feature Scaling

Classification models such as XGBoost are sensitive to feature scale when numeric features dominate. Therefore, numeric features were standardized using Standard Scaler.

Trained Models

1. Random Forest Classifier

A robust ensemble method based on decision trees. Random Forest works well for classification tasks with mixed-type features and nonlinear relationships. The advantage of using Random Forest Classifier was that it handles both categorical and numerical data, is resistant to overfitting and is a good baseline for classification.

2. XGBoost Classifier

We used XGBoost Classifier, an advanced gradient boosting model optimized for speed and performance, especially effective for structured/tabular data. It was configured for multi-class classification using the softmax objective. The advantage of XGBoost classifier was that it has high performance for multiclass tasks, builtin regularization and handles missing values and interactions well.

These metrics helped in identifying not just overall accuracy, but how well the models performed across each class.

Performance Metrics

For Regression

Each model's predictions were compared against actual target values using the regression metrics: Mean Absolute Error (MAE), Mean Squared Error (MSE), Root Mean Squared Error (RMSE), R^2 Score

For Classification

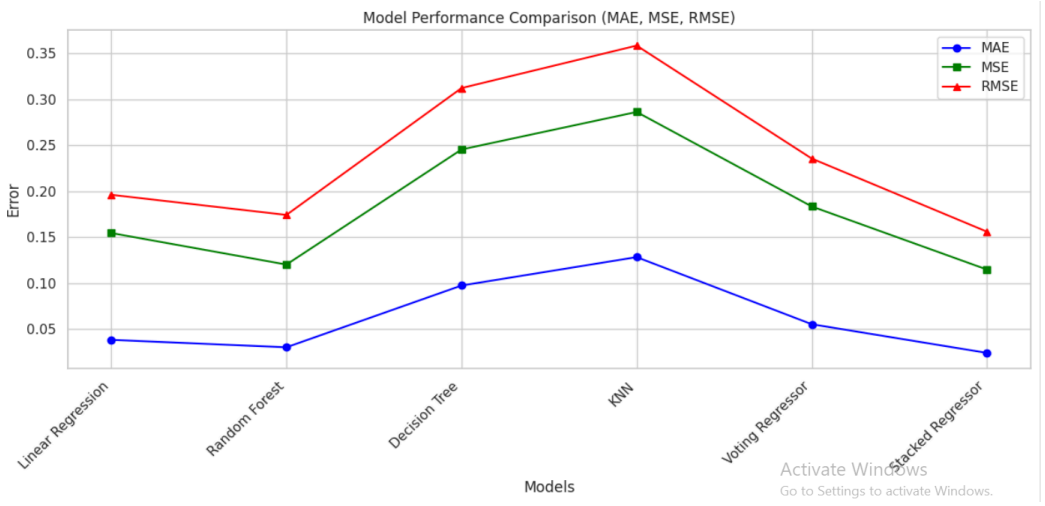
For both models, classification performance was assessed using: Accuracy, Precision, Recall, F1-score from the classification report.

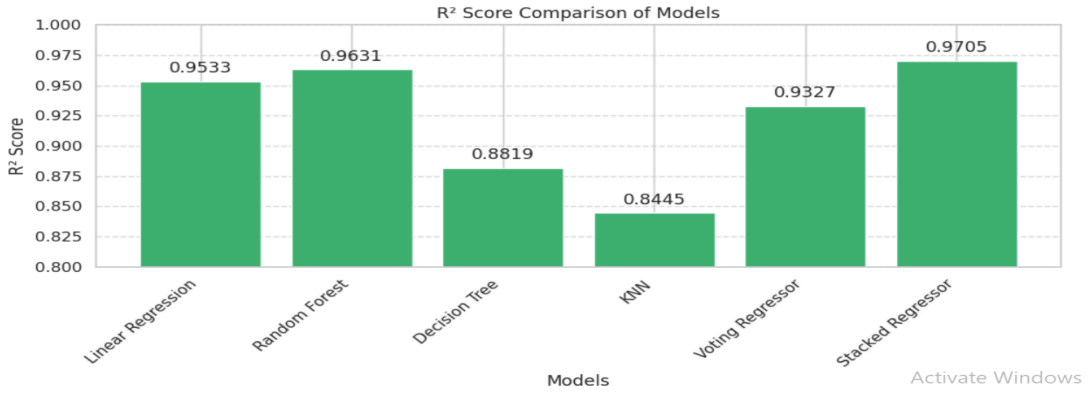
Results & Insights

Model evaluation:

For Regression:

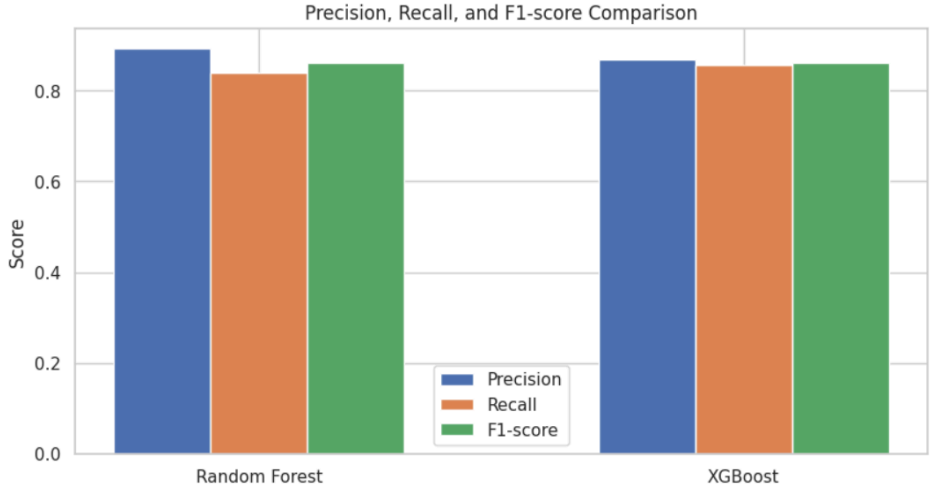
Model /Evaluation Metrics	MAE	MSE	RMSE	R^2 Score
Linear Regression	0.0385	0.1547	0.1963	0.9533
Random Forest Regression	0.0304	0.1204	0.1744	0.9631
Decision Tree Regressor	0.0976	0.2456	0.3124	0.8819
(KNN) Regressor	0.1285	0.2864	0.3586	0.84449
Ensemble Model – Voting Regressor	0.0555	0.1836	0.2357	0.9327
Ensemble Model -Stacking Regressor	0.0243	0.1150	0.1562	0.9705





For Classification

Model/ Evaluation Metrics	Accuracy	Precision	Recall	F1- Score
XGBoost Classifier	0.9178	0.8676	0.8564	0.8606



Validation

To assess model reliability and generalization, the dataset was split into training and testing sets using an 80/20 ratio. This ensured that the evaluation metrics reflected performance on unseen data. Models such as k-Nearest Neighbors and XGBoost, which are sensitive to feature scale, were trained using standardized input features. Categorical variables were handled using one-hot encoding, followed by column alignment to maintain consistency between training and testing sets. Even without cross validation, the models were able to generalize effectively.

Key takeaways

Among the regression models, the Stacking Regressor demonstrated the strongest overall performance, achieving the lowest mean absolute error (0.0243), mean squared error (0.1150), and root mean squared error (0.1562), alongside the highest R^2 score of 0.9705. This indicates that the

stacked ensemble was highly effective at capturing the variance in the target variable. In contrast, the k-Nearest Neighbor Regressor showed the weakest performance, likely due to its sensitivity to feature scaling and its reliance on local patterns in high-dimensional data. The ensemble approaches both Voting and Stacking consistently outperformed individual models, highlighting the advantage of combining multiple learners. In the classification task, the XGBoost Classifier outperformed the Random Forest Classifier, owing to its ability to model complex feature interactions and apply regularization to prevent overfitting. Preprocessing steps such as scaling and encoding played a critical role in model performance, especially for algorithms sensitive to input distributions. Overall, the findings suggest that thoughtful model selection, preprocessing, and ensemble methods significantly enhance predictive performance. Future work could explore hyperparameter optimization and cross-validation for further improvement.

Challenges & Future Work

Difficulties Faced

One of the main challenges encountered was managing preprocessing steps consistently across all models. Models like k-Nearest Neighbors and XGBoost were sensitive to the scale of features, requiring careful standardization. In addition, aligning one-hot encoded categorical variables between the training and test sets required attention to ensure that all expected columns were present and correctly ordered. This was especially critical when working with ensemble models that rely on multiple base learners.

Limitations

A notable limitation of this study was the use of a single 80/20 train-test split without incorporating cross-validation. This approach may not fully reflect how the model performs across different subsets of the data. Furthermore, the dataset size and potential imbalance in class distribution, particularly in the classification task, could have influenced the model's performance and the reliability of the evaluation metrics. In classification, metrics like recall and F1-score may have been affected by unbalanced class frequencies, as no techniques like SMOTE or class weighting were applied to address the issue.

Potential Improvements

To enhance this work, several improvements can be implemented. **First**, adopting k-fold cross-validation would offer a more stable and comprehensive assessment of model performance. **Second,** systematic hyperparameter tuning using methods such as RandomizedSearchCV would likely yield better-performing models, especially for ensembles like Random Forest, XGBoost, and Stacking Regressor. **Third,** applying class imbalance techniques—such as oversampling, undersampling, or adjusting class weights—could improve classification outcomes. **Lastly,** feature selection or dimensionality reduction could simplify models and improve interpretability, while using external validation datasets would better test the generalizability of the trained models.

Conclusion & Recommendations

This project explored a range of regression and classification models to predict student performance metrics such as GPA (regression) and grade classification (classification). The **regression analysis** involved six models including Linear Regression, Decision Tree Regressor, Random Forest Regressor, k-Nearest Neighbors, and two ensemble methods—Voting Regressor and Stacking Regressor. Among these, the Stacking Regressor demonstrated superior performance, achieving the highest R^2 score and the lowest error metrics (MAE, MSE, RMSE).

On the **classification side**, both Random Forest and XGBoost classifiers were implemented. XGBoost outperformed the Random Forest classifier in terms of accuracy and classification report metrics, demonstrating its ability to handle complex feature relationships and imbalanced data more effectively.

Preprocessing steps such as one-hot encoding, feature scaling, and proper alignment of datasets were critical to ensuring model stability and accuracy. Despite some limitations, the models produced reliable results with minimal overfitting, and ensemble methods consistently enhanced performance by combining the strengths of individual models.

Real-World Application

The outcomes of this project have meaningful implications for real-world educational settings. Accurate prediction of student grades or GPA can be valuable for academic advisors, institutions, and policymakers in identifying at-risk students early and implementing targeted interventions. Predictive models could be integrated into student information systems to provide automated, data-driven insights on academic performance.

Moreover, these models could be extended to recommend personalized learning plans, monitor academic progress, or guide resource allocation decisions. To further enhance real-world applicability, future implementations could incorporate real-time data pipelines, model retraining strategies, and ethical considerations around fairness and interpretability, ensuring the models remain relevant, equitable, and actionable in dynamic educational environments.

References

- Dataset
- Streamlit
- A Systematic Literature Review of Student' Performance Prediction Using Machine Learning Techniques
- Student Performance Prediction and Classification Using Machine Learning Algorithms