**Data Preparation/Feature Engineering**

1. **Overview**

The data preparation and feature engineering phase is critical to building effective machine learning models that drive personalized, accessible, and inclusive learning experiences. Leveraging open datasets such as the Student Performance Prediction Dataset, Students’ Adaptability Level in Online Education, and xAPI-Edu-Data, the goal is to extract meaningful patterns related to academic achievement, adaptability to online environments, and behavioral engagement. These insights inform the development of adaptive learning paths, performance prediction models, and targeted support mechanisms. Spectrum-EDU is designed to bridge educational gaps faced by children in under-resourced, geographically isolated, conflict-affected, or disability-challenged environments. By integrating personalization, gamification, accessibility features, multilingual (Amharic and English) content, and cultural adaptation, the platform aligns with Sustainable Development Goals SDG 4 (Quality Education) and SDG 10 (Reduced Inequalities).

1. **Data Collection**

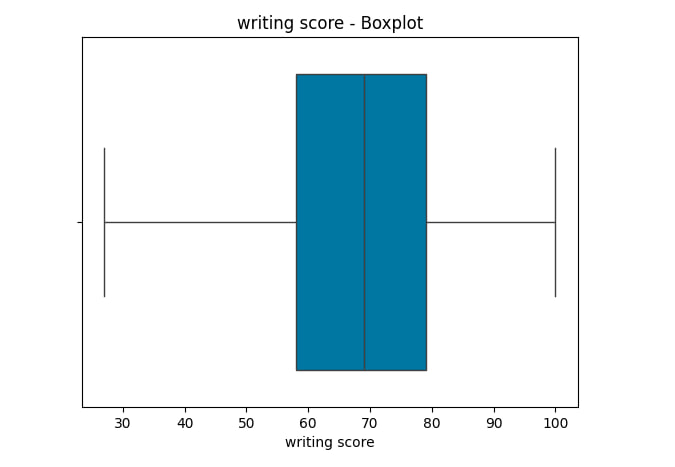
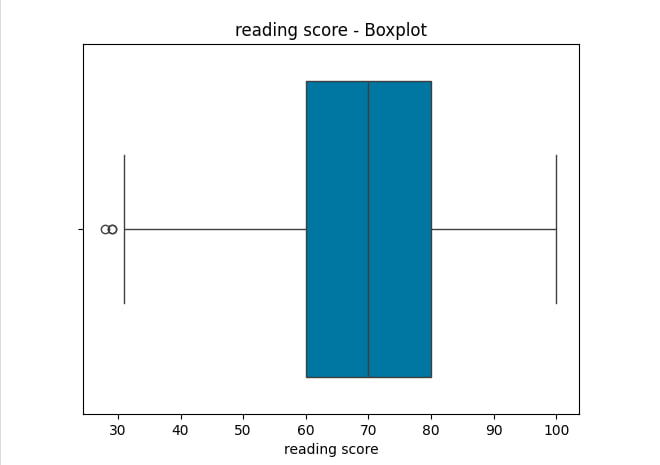
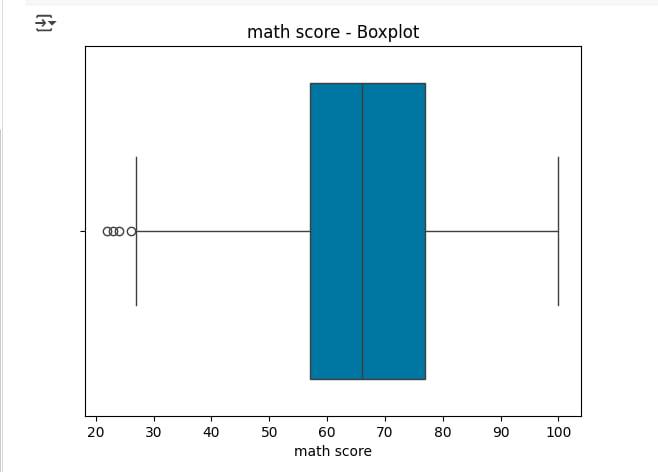
The Spectrum-EDU project utilizes publicly available datasets from reputable educational and research sources to support model development and evaluation. Currently, we have worked with the **Kaggle Student Performance Dataset**, which includes detailed demographic and academic performance data and serves as a strong foundation for modeling learning support needs and personalization strategies. We are also planning to integrate two additional datasets in the future: the **Students’ Adaptability Level in Online Education** dataset, which captures psychological and situational factors affecting students' ability to adapt to online learning; and the **xAPI-Edu-Data** dataset, which offers behavioral interaction logs essential for analyzing engagement and emotional focus. For the dataset already in use, we performed preprocessing steps such as standardizing column names, removing irrelevant or redundant fields, handling missing values through imputation, and ensuring consistent formatting for numerical and categorical features. Future datasets will undergo similar cleaning and normalization processes to align with the platform's schema and support accurate downstream analysis and machine learning modeling.

1. **Data Cleaning**

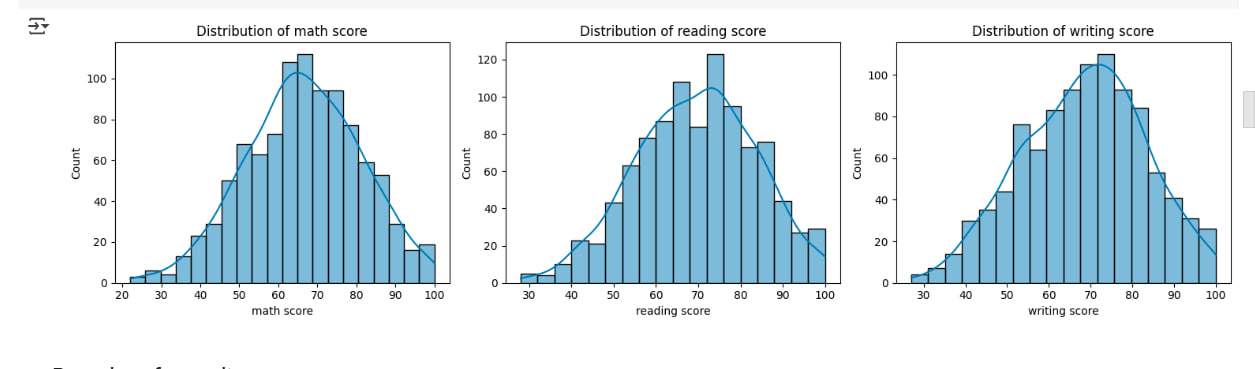
To clean the raw student performance dataset, the following main steps were taken: loading the dataset, checking for missing values, handling duplicates, detecting and removing outliers, encoding categorical variables, and scaling numeric features. The dataset was first read from a CSV file and found to have no missing or duplicate entries, ensuring completeness and integrity. Boxplots were used to visualize outliers, which were then removed using the z-score method, reducing the data from 1000 to 993 rows. Categorical columns like gender, race/ethnicity, parental education, lunch type, and test preparation course were encoded using one-hot encoding while dropping the first category to avoid multicollinearity. Finally, the numeric features—math, reading, and writing scores—were standardized using StandardScaler to ensure all features were on a comparable scale. This comprehensive preprocessing ensured the dataset was clean and well-prepared for analysis or modeling.

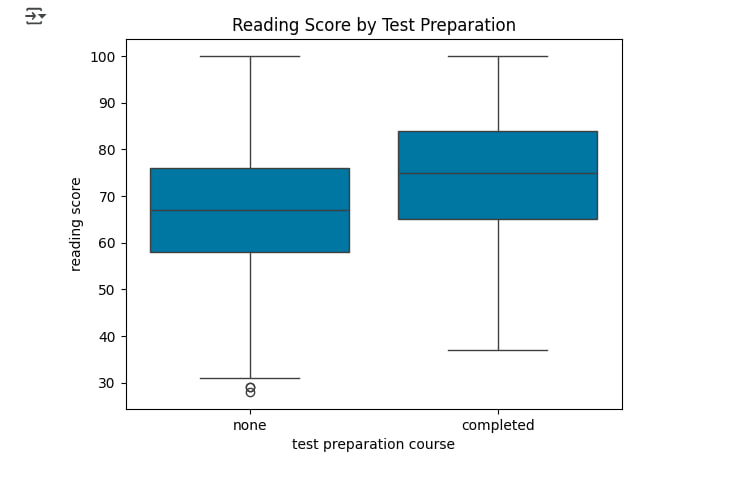
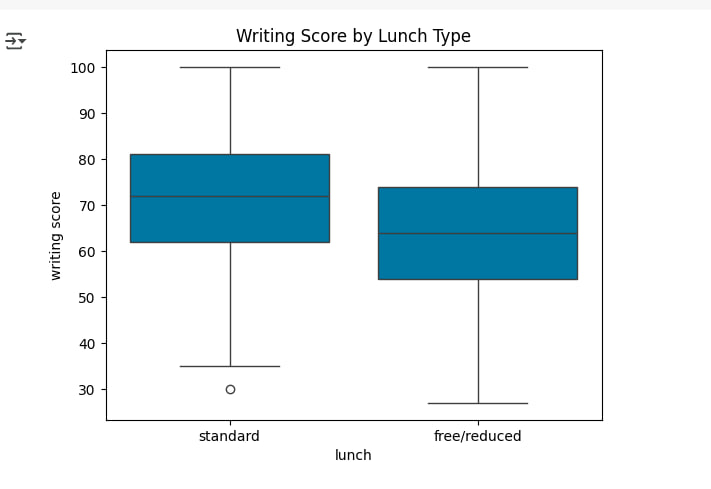
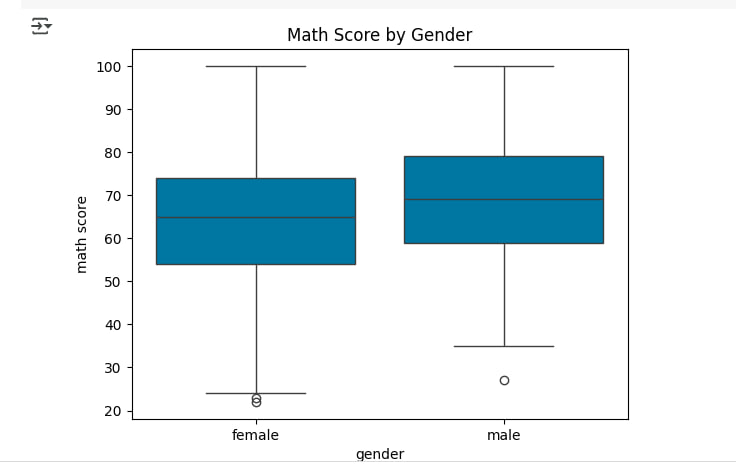
1. **Exploratory Data Analysis (EDA)**

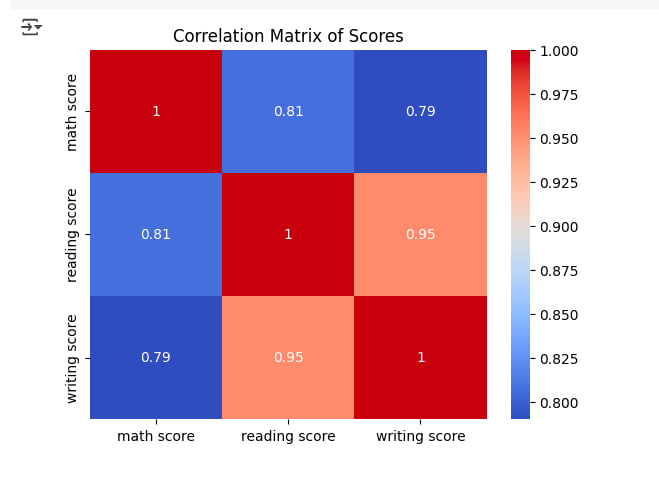
* **Box Plots – Outlier Detection**: Box plots were generated for the three numeric features: **math score**, **reading score**, and **writing score**. These visualizations helped identify outliers in the dataset. A few extreme low scores were found, which were later removed using the Z-score method to ensure clean, normally distributed data for better model performance.



* **Histograms – Score Distribution:** Histograms were plotted to examine the distribution of scores across the three subjects. The plots revealed a slight right-skew in all three distributions, indicating that most students scored relatively high, though a few students performed poorly.



* **Count Plots – Categorical Feature Distribution**: The **Count Plots** for **Gender**, **Lunch**, and **Test Preparation Course** reveal key trends. The **Gender vs. Score** plot shows female students outscoring males in **reading** and **writing**, while males performed slightly better in **math**. The **Lunch vs. Score** plot indicates that students with **standard lunch** scored higher than those on **free/reduced lunch**. The **Test Preparation Course vs. Score** plot highlights that students who completed the **test prep course** scored better, especially in **math** and **writing**. These plots emphasize the influence of gender, nutrition, and preparation on academic performance.
* **Correlation Heatmap** – Relationship Between Scores: A correlation heatmap was generated to visualize relationships between the numeric features. Strong positive correlations were observed between **reading and writing scores**, while math scores had a slightly lower correlation with the other two subjects. This suggests that students strong in reading are likely to also excel in writing.



1. **Feature Engineering**

 **One-Hot Encoding**: Categorical variables, such as gender, race/ethnicity, and test preparation course, were one-hot encoded. This converts each category into binary columns, allowing the model to interpret the data numerically and capture relationships with the target variables.

 **Standardization of Numerical Features**: Numerical features like math, reading, and writing scores were standardized using StandardScaler. This transforms the features to have a mean of 0 and a standard deviation of 1, ensuring all features contribute equally to the model and preventing dominance by features with larger ranges.

 **Interaction Features**: Interaction features were created by combining variables such as math scores and test preparation. For instance, combining test preparation completion with math scores captures complex relationships that improve the model's ability to predict the target variable more accurately.

 **Binning / Discretization**: Continuous numerical features, like math scores, were binned into categories such as "Low," "Medium," and "High". This transformation helps simplify the model by grouping similar values, which enhances the model's ability to recognize patterns effectively.

1. **Data Transformation**

**Encoding(One-Hot Encoding)**

drop\_first = True avoids multicollinearity by removing one dummy column from each category.Applied to variables like: gender, race/ethnicity, parental level of education, lunch, and test preparation course.

df\_encoded = pd.get\_dummies(data,drop\_first=True)

print(df\_encoded.head())

**Scaling(Standardization (Scaling Numerical Features))**

You standardized numerical features (math score, reading score, writing score) using StandardScaler. This transforms features to have **mean = 0** and **standard deviation = 1**, ensuring uniform contribution to the model.

from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

X\_train\_scaled = scaler.fit\_transform(X\_train)

X\_test\_scaled = scaler.transform(X\_test)

**Model Exploration**

**1. Model Selection**

XGBoost was chosen for this project due to its high accuracy, efficiency, and ability to handle both numerical and categorical data, making it ideal for predicting student performance from structured datasets like the one used here. Its built-in regularization helps prevent overfitting, and it performs well even with limited data. While it requires some parameter tuning and is less interpretable than simpler models, its strong performance and ability to reveal feature importance make it a powerful tool for supporting the personalized learning goals of SpectrumEdu.

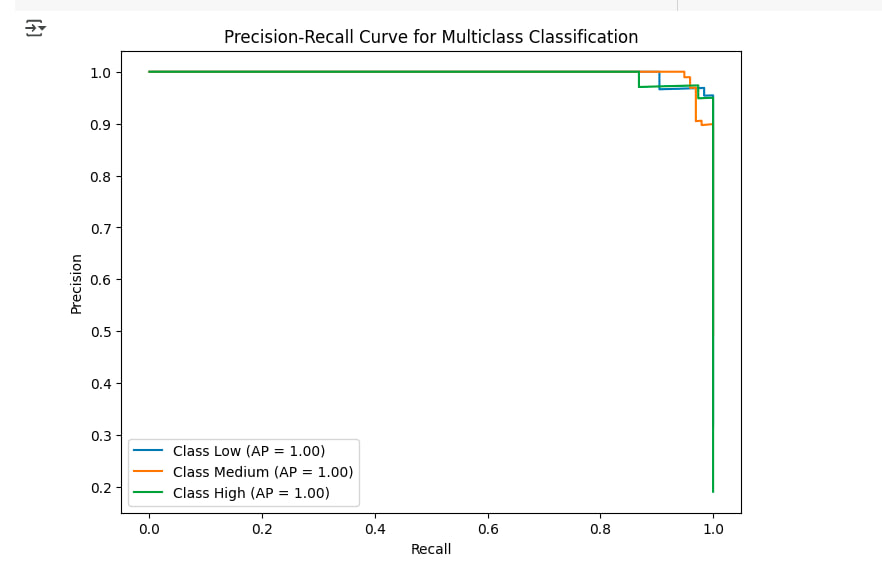
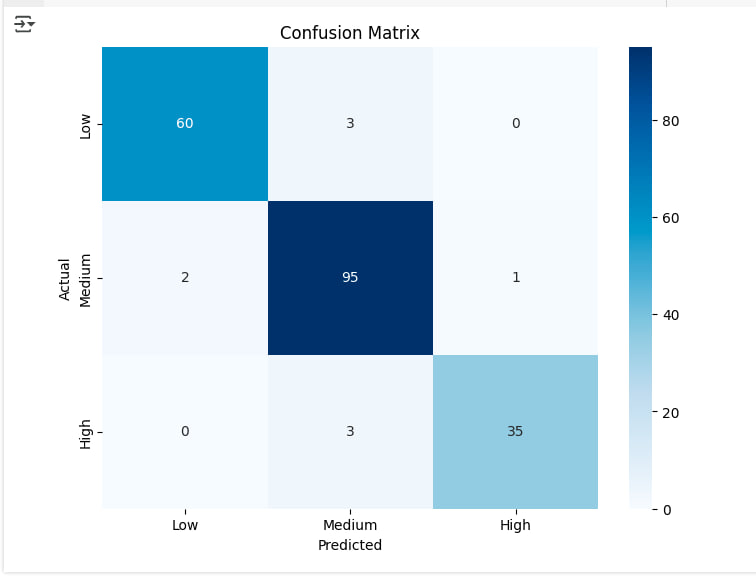
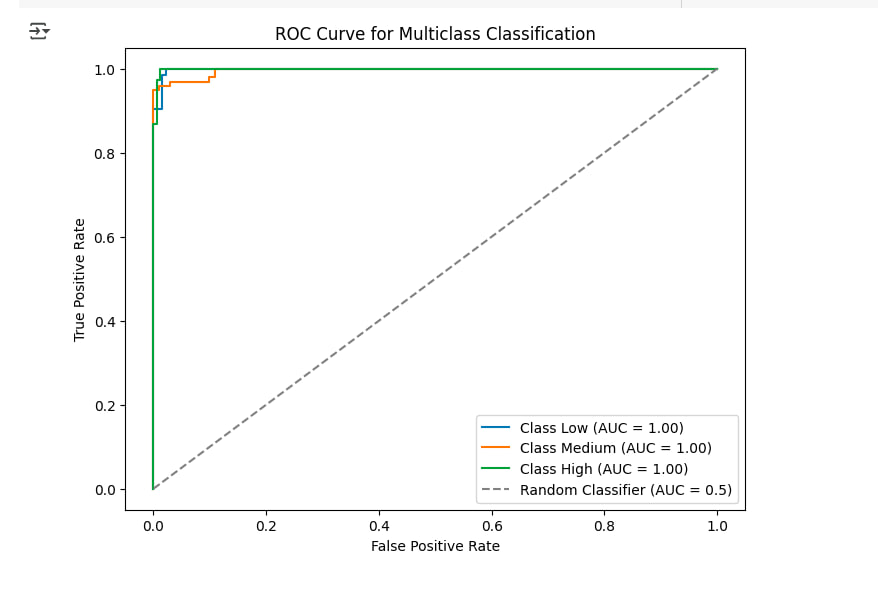
**2. Model Training**

The model was trained using an XGBClassifier with hyperparameters like n\_estimators=100, learning\_rate=0.1, and max\_depth=4, which were selected to optimize performance while preventing overfitting. To ensure robust performance, 5-fold cross-validation was applied, achieving a mean accuracy of 97.48%. The final model, after training on the full dataset, achieved an accuracy of 95.48% on the test set, with strong precision, recall, and F1-scores across all performance levels (Low, Medium, High). This approach enables effective classification of student performance for the SpectrumEdu project, ensuring personalized learning experiences.

**3. Model Evaluation**

The performance of the XGBoost classifier model was assessed using multiple evaluation metrics. The model achieved an impressive accuracy of 95.48%, with a detailed classification report indicating strong performance across all classes (Low, Medium, and High), particularly in precision and recall for "Low" and "High" categories. A confusion matrix was visualized, showing how well the model predicted each class, while ROC curves for each class demonstrated the model’s ability to distinguish between classes, with high AUC values. Additionally, precision-recall curves further illustrated the trade-offs between precision and recall. These metrics and visualizations provide a comprehensive view of the model's robustness, particularly in handling class imbalances and maintaining performance across different thresholds.

The images below shows the confusion matrix, ROC curves and precision-recall curves respectively

**4. Code Implementation**

Here is the link: [Link](https://colab.research.google.com/drive/1myuFeRcncgnQjGThW7olTL-6l441KvSD#scrollTo=Aef4KLkp-GYo)

The model is currently in progress. Some steps have been completed like Data preprocessing and feature engineering (including one-hot encoding and scaling), Model training using the XGBoost classifier,Model evaluation with metrics such as accuracy, precision, recall, and AUC-ROC.

In the Future we are planning to add hyper parameter tuning using grid search, exploration of additional features and model optimizations and further validation and model improvement.