INTEL UNNATI INDUSTRIAL TRAINING PROGRAM 2025 – 2026

PS 03: AI POWERED PERSONALIZED TUTOR SYSTEM

Submitted by:

AAKESH RAJ R

DEEPAK P

THARUN S

ABSTRACT

The rapid evolution of education technology demands intelligent systems that can adapt to the individual needs of learners. This project focuses on developing an AI-powered Personalized Tutor System designed specifically for K-12 students in virtual school environments. The goal is to use various data points, such as student demographics, learning behaviour, IQ, and assessment performance, to create a more personalized learning experience. By leveraging these factors, the system aims to enhance educational outcomes by delivering content that is tailored to each student's unique learning style and cognitive abilities.

To implement this personalized learning approach, the project uses machine learning techniques, specifically the XGBoost classifier, to predict the appropriate level of educational content for each student. A synthetic dataset is simulated to represent a diverse group of learners, capturing a range of learning behaviours and performance metrics. The system is designed to provide content that matches students' learning paces and cognitive levels, ensuring that they receive material suited to their individual needs. Additionally, the project explores the prediction of assessment scores, which could help inform decisions related to student promotion.

The AI-Powered Personalized Tutor System shows promising results, with strong performance metrics that highlight the effectiveness of the chosen model and feature engineering methods. This project aligns with the broader vision of democratizing education through AI by offering tailored content that enhances learning outcomes. Looking ahead, the system could benefit from future enhancements such as real-time data feedback loops and integration with actual learning management systems to provide a more seamless and dynamic educational experience.

TABLE OF CONTENT

INTRODUCTION

Education is no longer confined to traditional classrooms. With the rise of online and hybrid learning platforms, the need for personalized educational experiences has grown substantially. Generic teaching approaches often fail to meet the unique needs of individual learners, especially in K–12 segments. This project focuses on building an AI-driven system capable of tailoring educational content to each student's abilities and requirements. By simulating a realistic dataset and using machine learning, we aim to address core educational challenges: promoting students based on performance and recommending suitable learning material based on their level and aptitude. This project is a part of the Intel® Unnati Industrial Training 2025 and emphasizes AI's role in transforming education for the better.

OBJECTIVES OF THE PROJECT

- **Create a Synthetic Dataset:**
  Develop a synthetic dataset representing students' demographics, behaviour, IQ, and academic performance to simulate a virtual learning environment.
- **Predict Assessment Scores:**
  Use demographic and behavioural features to predict students' future performance in assessments, identifying areas for improvement or enrichment.
- **Recommend Learning Material Levels:**
  Suggest personalized learning materials (Beginner, Intermediate, Advanced) based on each student's attributes, ensuring content matches their learning pace and cognitive ability.

- **Support Promotion Decisions:**
  Evaluate predicted performance to assist in making informed promotion decisions, ensuring students are ready for the next academic level.
- **Explore Adaptive Learning Algorithms:**
  Investigate AI and ML algorithms to generate dynamic learning paths that adapt to students' progress and engagement, enhancing personalized learning experiences.

DATASET DESCRIPTION

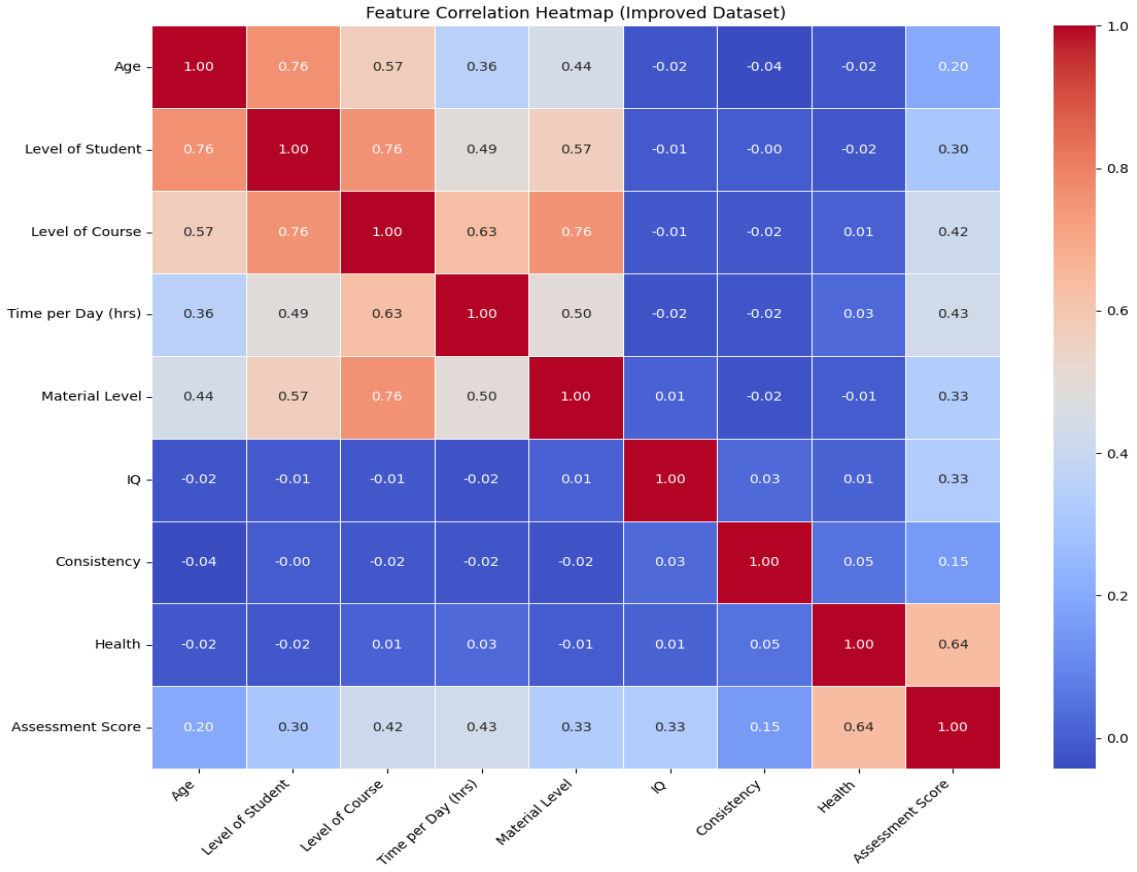
The dataset used for this project was synthetically generated to mimic real-world K–12 virtual school data. It consists of 1000 student records with the following attributes:

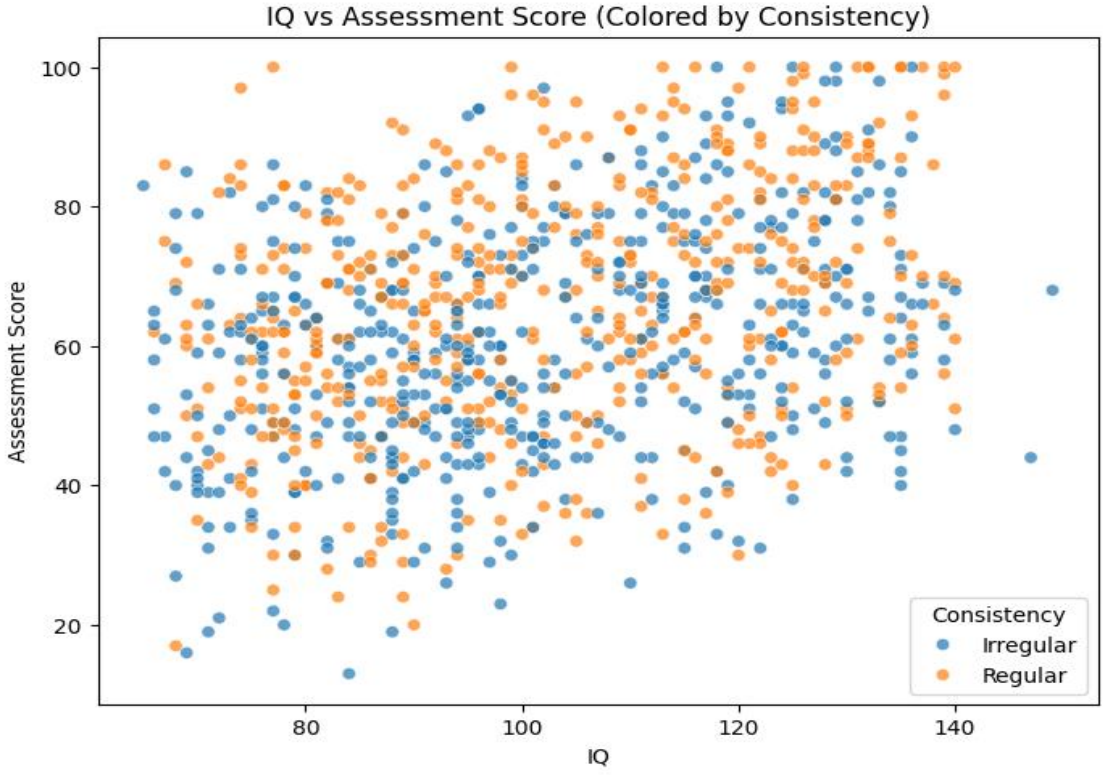
AssessmentScore.csv

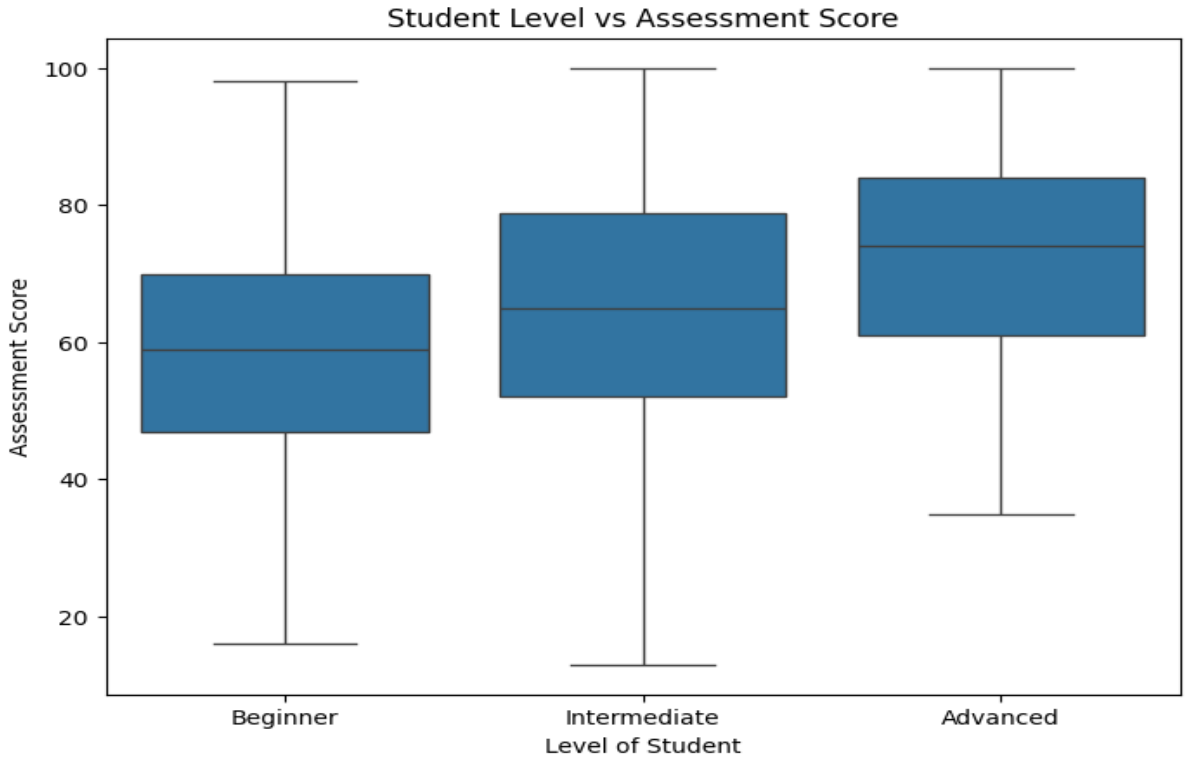
Feature	Description
Name	Student’s name
Age	Age of the student
Gender	Gender (Male, Female, Other)
Country, State, City	Geographic location
Parent Occupation	Job role of the parent
Earning Class	Socio-economic status (e.g., Low, Medium, High)
Level of Student	Educational level (e.g., Primary, Secondary)
Level of Course	Course difficulty level

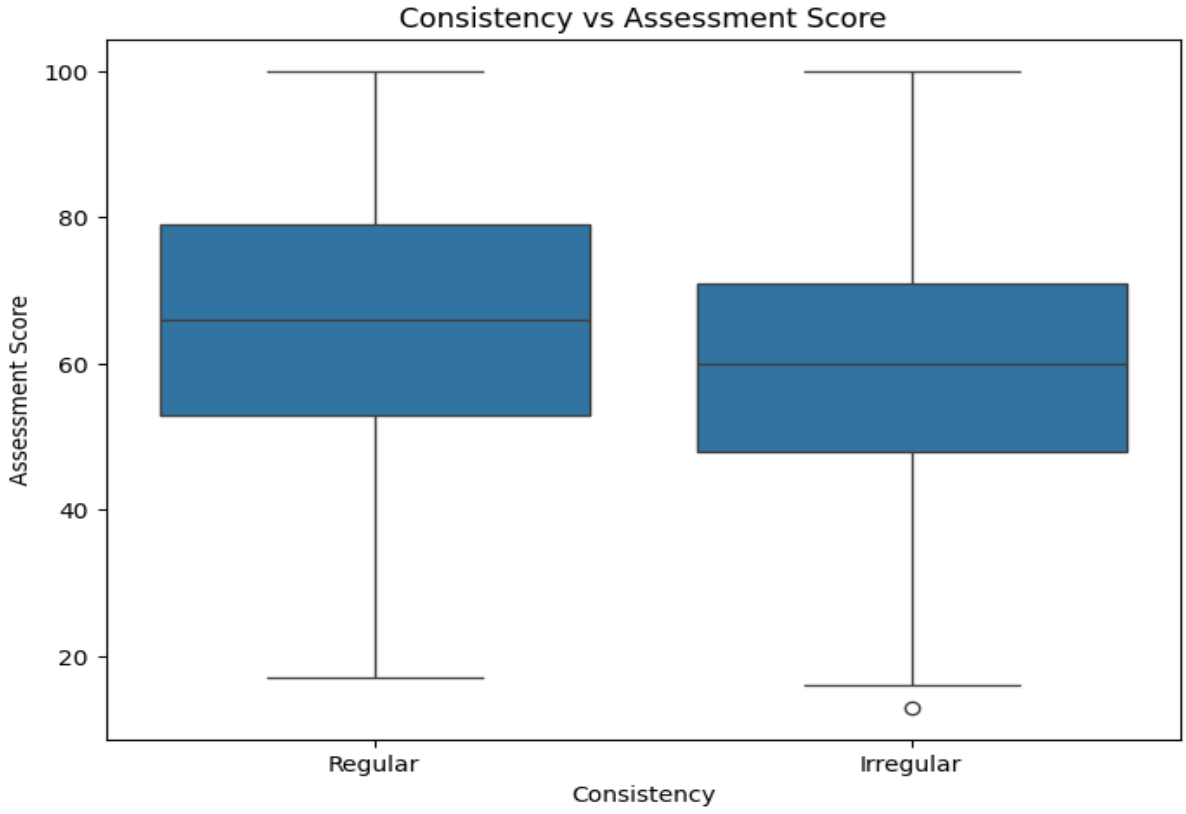
Course Name	Name of enrolled course
Assessment Score	Performance metric (0–100)
Time per Day	Time spent on studying per day
Material Name	Name of study material
Material Level	Level of the material (Beginner, Intermediate, Advanced)
IQ	Intelligence Quotient of the student

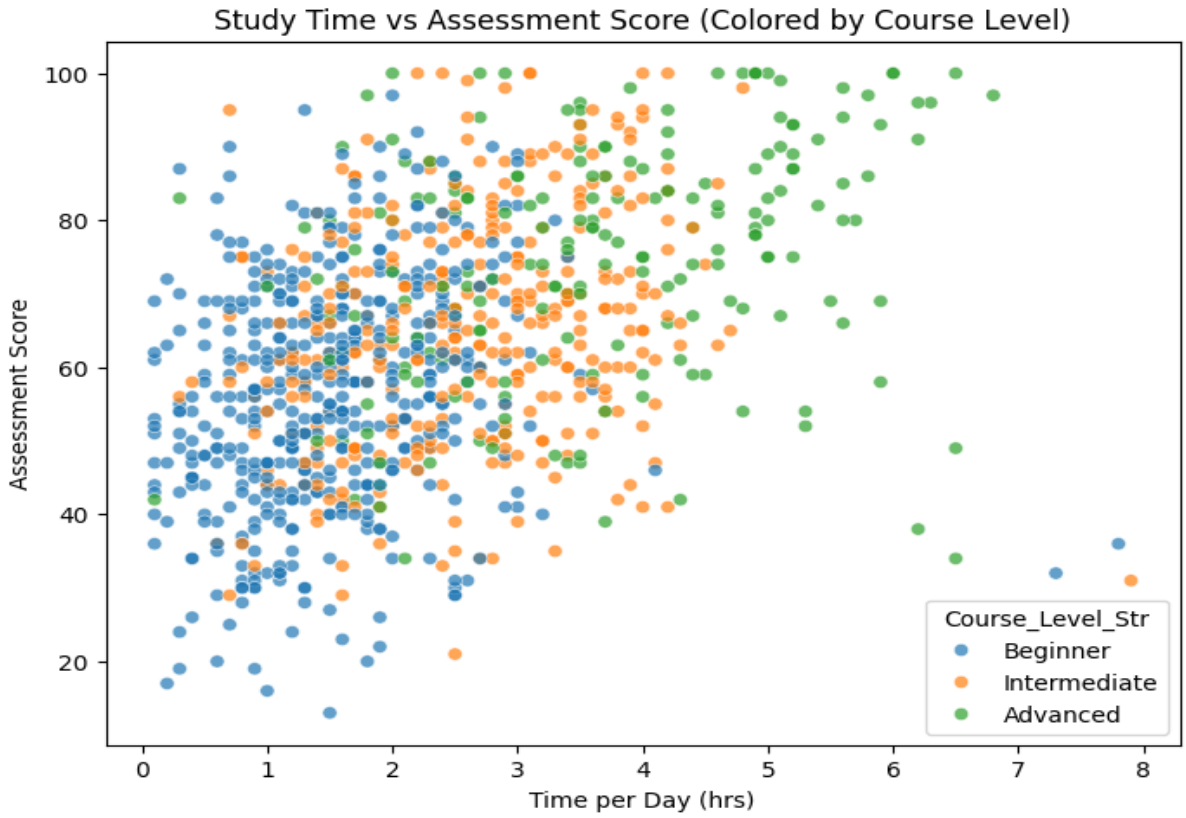
DATA ANALYSIS

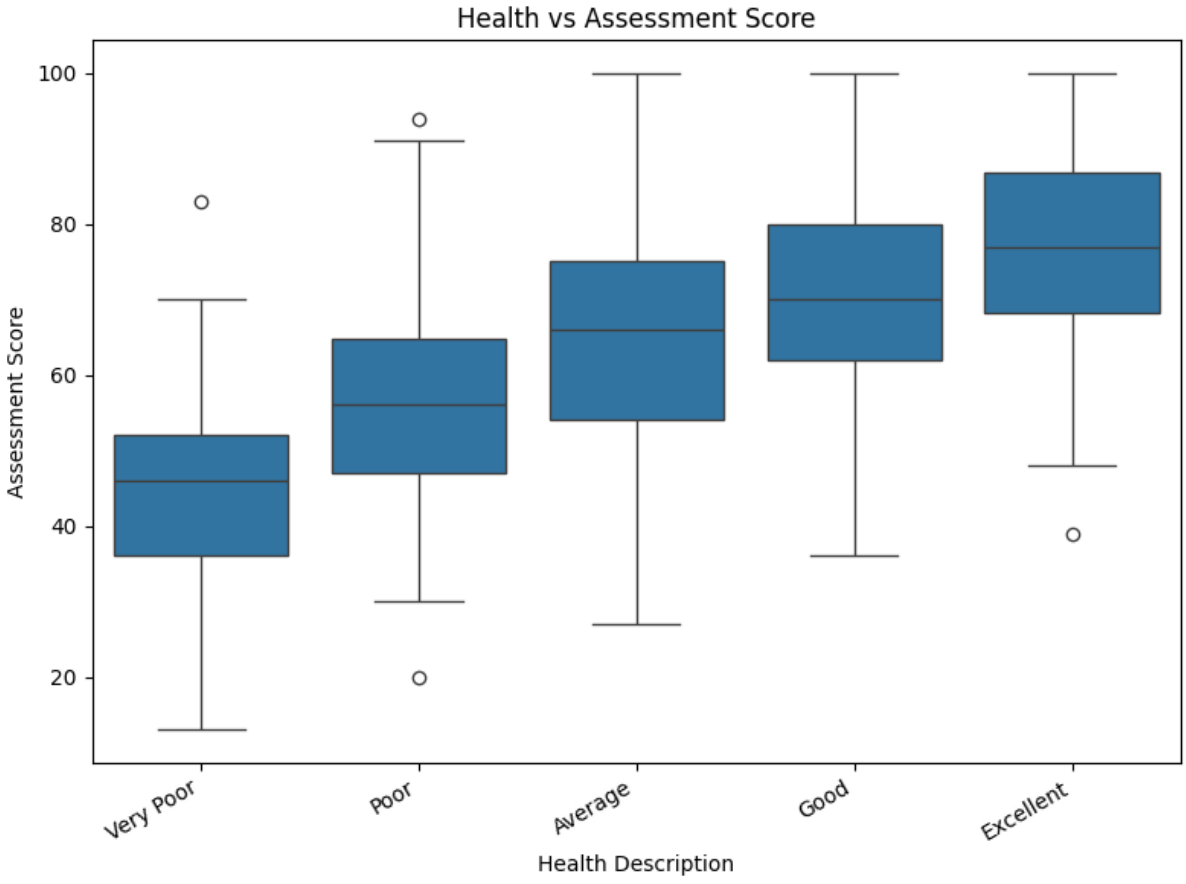





Student Level vs Assessment Score
Assessment Score
100
80
60
40
20
Beginner
Intermediate
Advanced
Level of Student


Consistency vs Assessment Score
Assessment Score
100
80
60
40
20
Regular
Irregular
Consistency


Study Time vs Assessment Score (Colored by Course Level)
Assessment Score
Time per Day (hrs)
Course_Level_Str
Beginner
Intermediate
Advanced

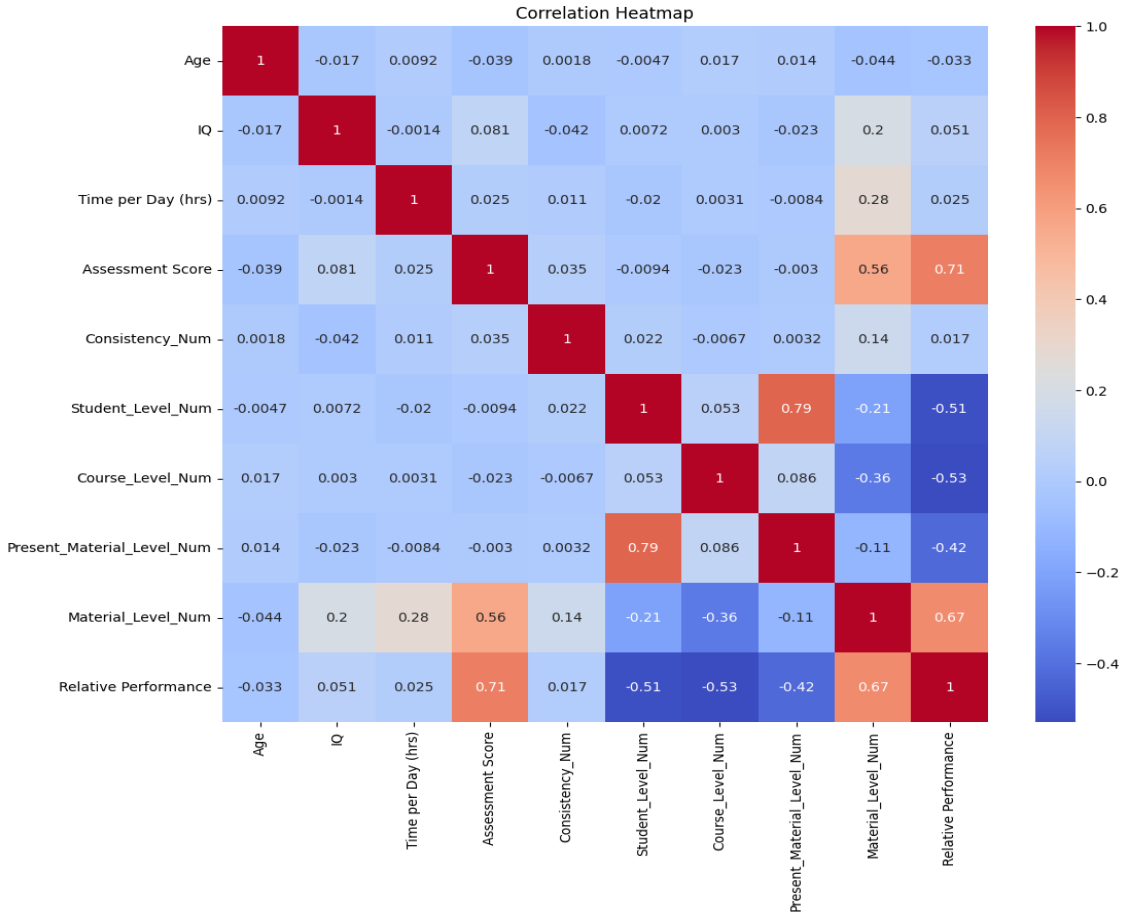

Health vs Assessment Score
Assessment Score
Very Poor
Poor
Average
Good
Excellent
Health Description

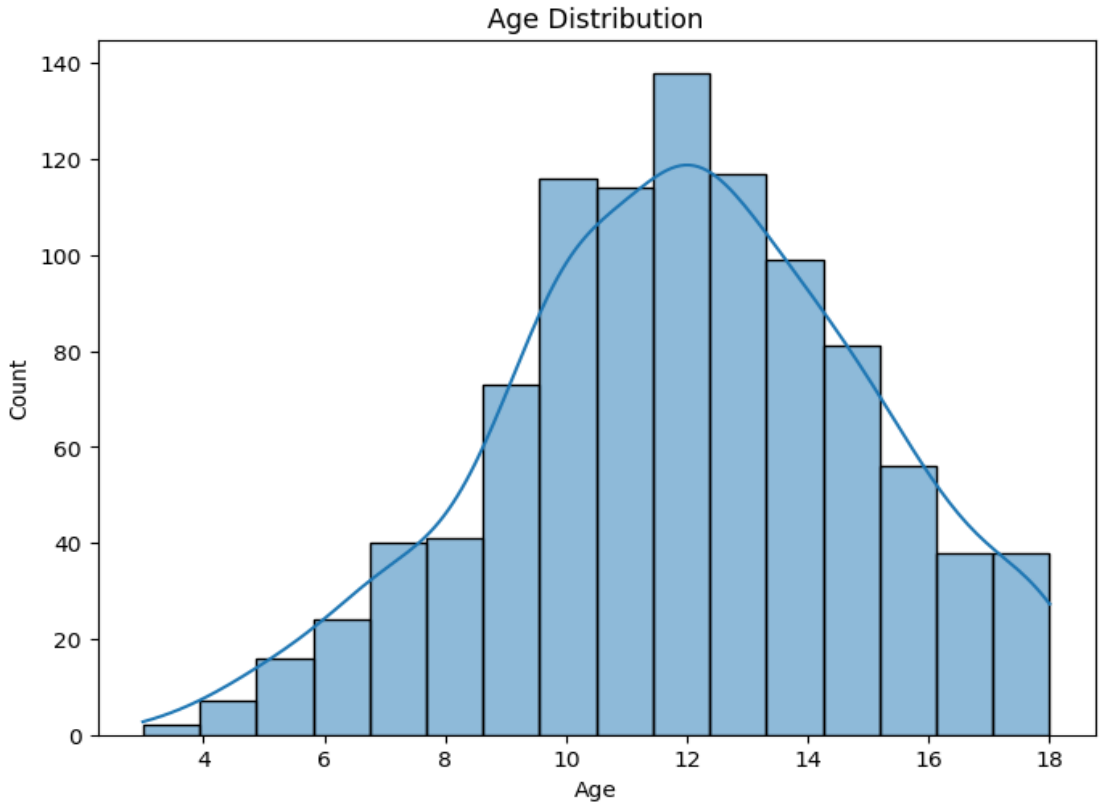
MaterialLevel.csv

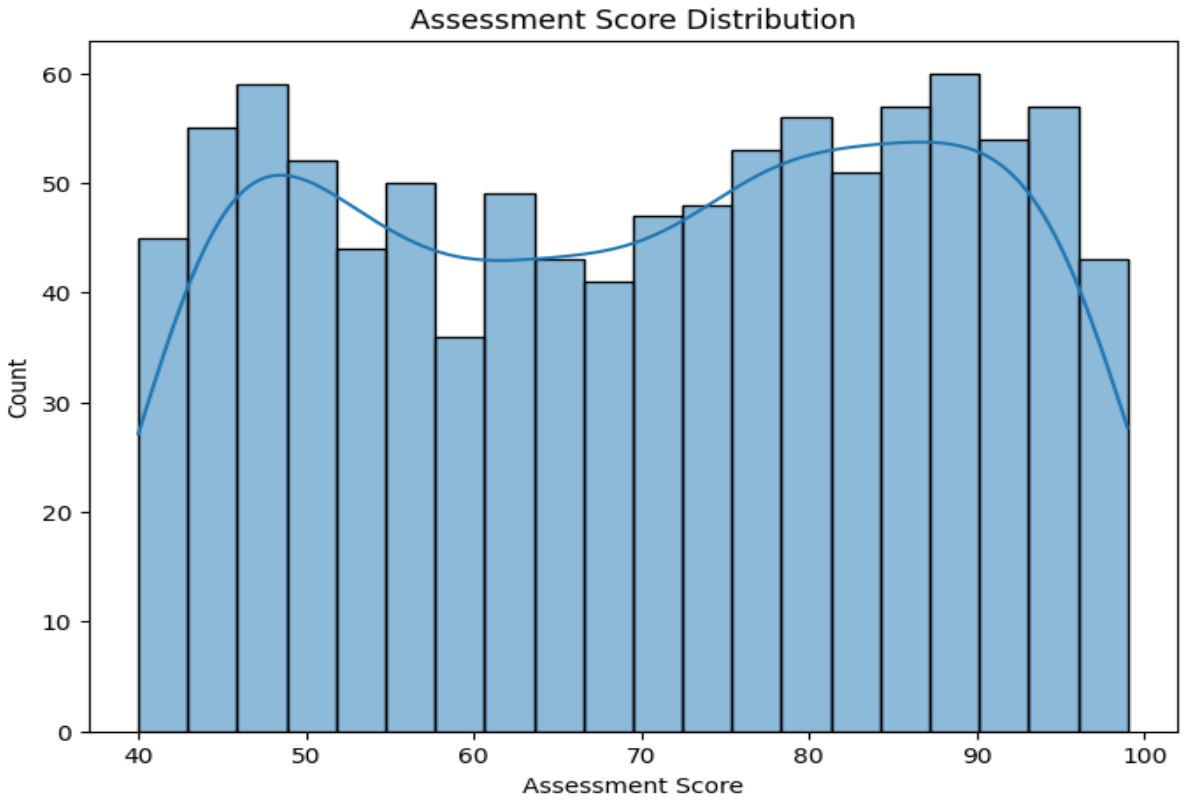
Feature	Description
Age	Age of the student, generated from a normal distribution with a mean of 12 years and a standard deviation of 3 years.
IQ	IQ score of the student, generated from a normal distribution with a mean of 100 and a standard deviation of 15.
Time per Day (hrs)	Average time spent by the student on studying per day, generated from an exponential distribution with a mean of 1.5 hours.
Assessment Score	Score the student received on an assessment, randomly generated between 40 and 100.
Level of Student	Educational level of the student (e.g., Beginner, Intermediate, Advanced), randomly assigned based on probabilities.
Level of Course	Difficulty level of the course the student is enrolled in (e.g., Beginner, Intermediate, Advanced), randomly assigned.
Course Name	Name of the course the student is taking (e.g., Math, English, Science, History), randomly assigned.
Consistency	Consistency of the student's study habits (e.g., Regular, Irregular), randomly assigned based on probabilities.

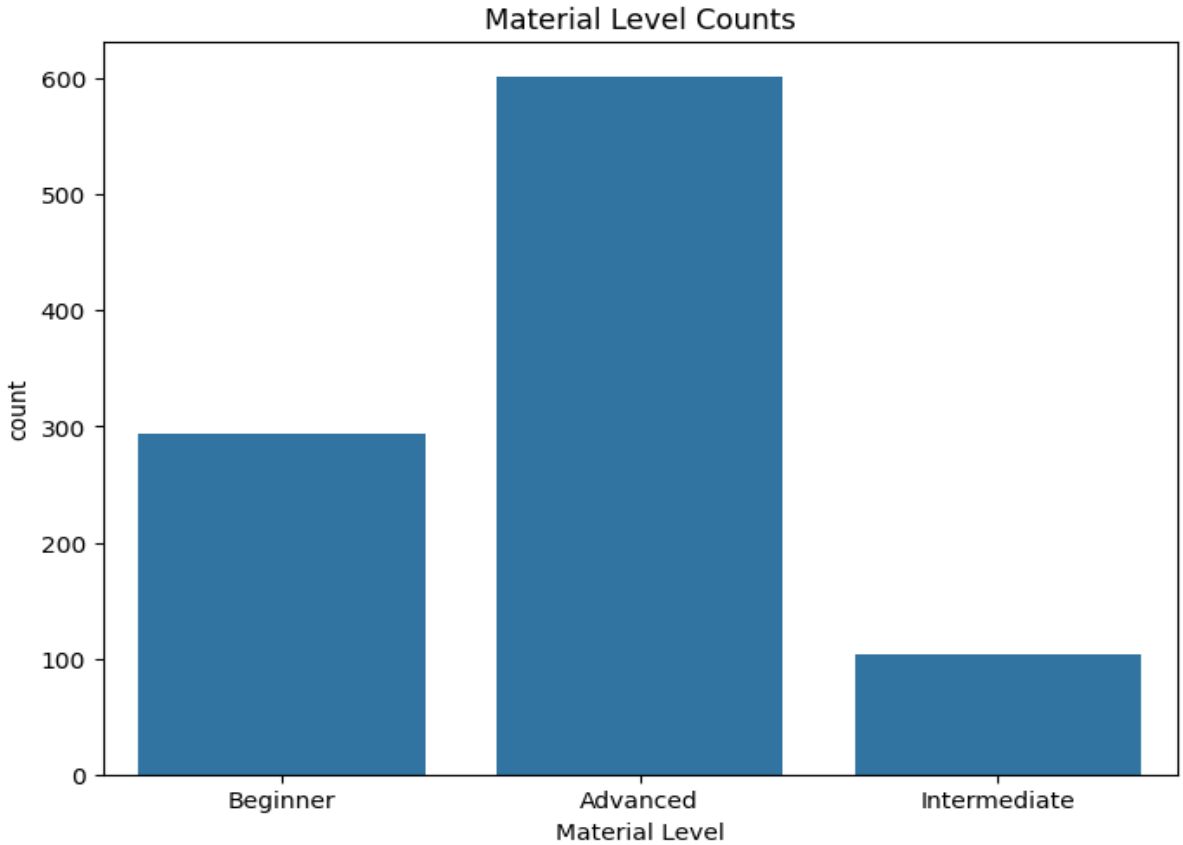
Present Material Level	The material level the student is presented with based on their educational level and course.
Relative Performance	A measure of the student's performance compared to their expected score, calculated based on Assessment Score, Student Level, and Course Level.
Material_Level	Predicted material level based on multiple features such as IQ, assessment score, and study consistency (Beginner, Intermediate, Advanced).

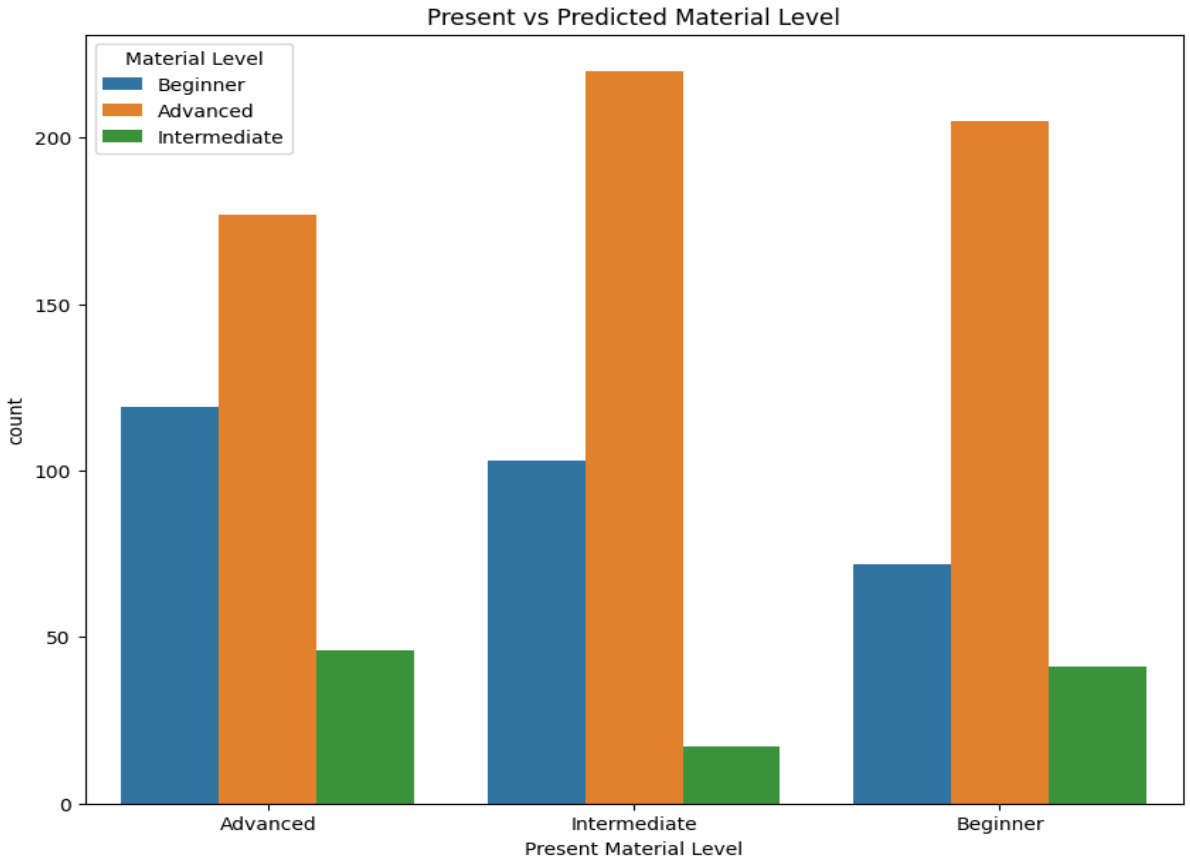
DATA ANALYSIS

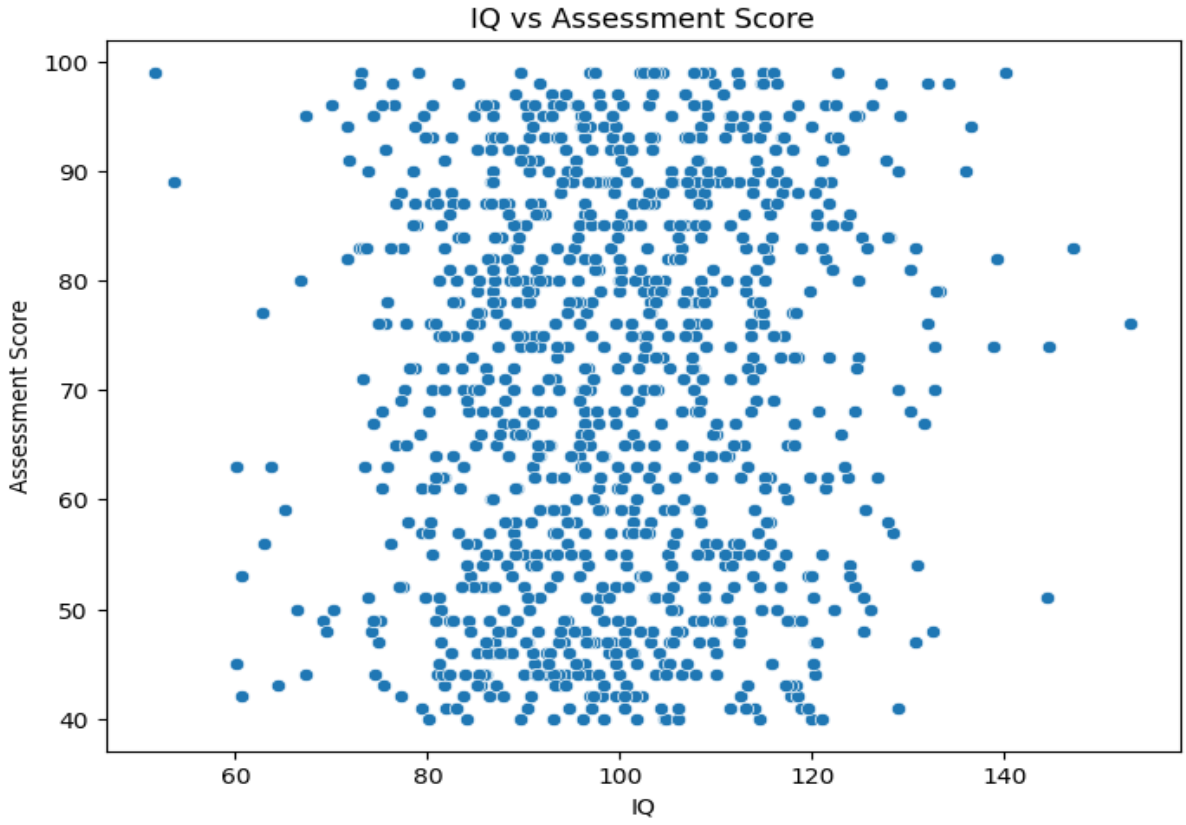


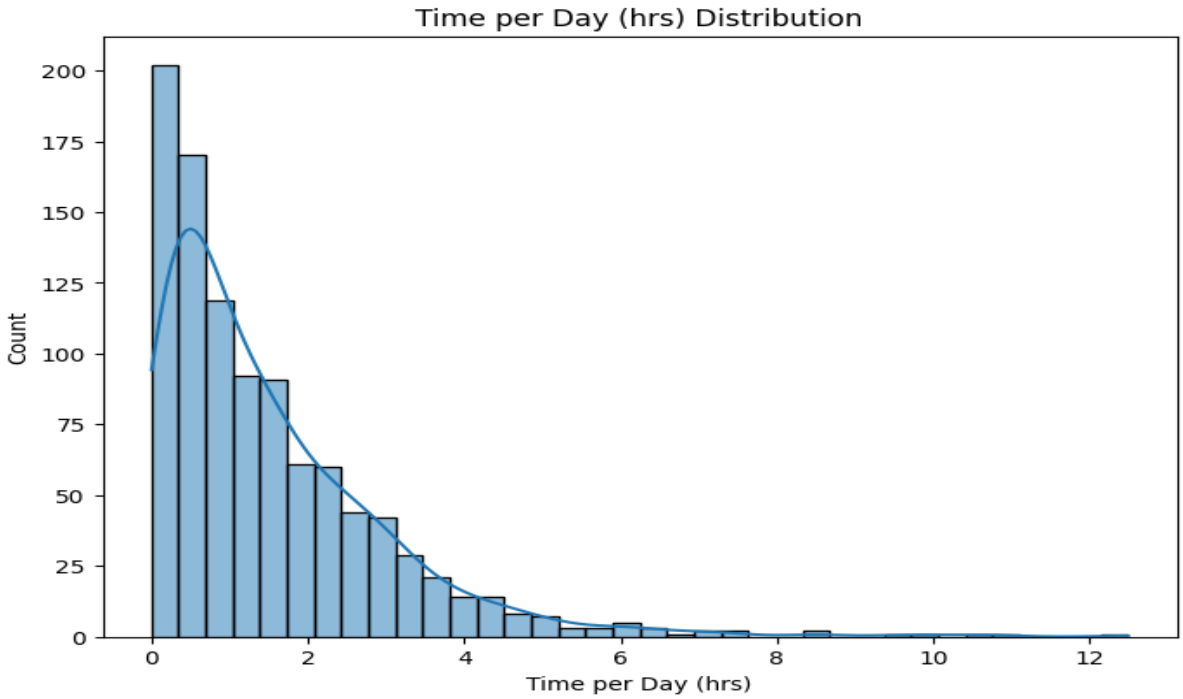



Assessment Score Distribution
Count
Assessment Score
0
10
20
30
40
50
60
40
50
60
70
80
90
100


Material Level Counts
count
Material Level
0
100
200
300
400
500
600
Beginner
Advanced
Intermediate


Present vs Predicted Material Level
Material Level
Beginner
Advanced
Intermediate
count
0
50
100
150
200
Advanced
Intermediate
Beginner
Present Material Level


IQ vs Assessment Score
Assessment Score
40
50
60
70
80
90
100
60
80
100
120
140
IQ



METHODOLOGY

The goal of this project was to predict the material level appropriate for a student based on various factors like their age, IQ, daily study time, assessment score, and several categorical features related to their level of knowledge and consistency. The entire methodology consists of two major phases:

Dataset Creation and **Prediction Model Development**.

Steps involved:

1. **Generation of Features**:
    - **Age**: Simulated using a normal distribution with a mean of 12 years and a standard deviation of 3, and clipped to range between 3 and 18 years to ensure the age is within a realistic student age group.
    - **IQ**: Generated using a normal distribution with a mean of 100 and a standard deviation of 15, resembling the IQ distribution of the general population.

- **Time per Day (hrs)**: Generated using an exponential distribution with a mean of 1.5 hours, to represent the amount of time a student spends studying each day.
- **Assessment Score**: Simulated using a uniform distribution between 40 and 100, representing the score a student achieves on an assessment.
- **Level of Student**: Randomly assigned to one of the three categories: "Beginner," "Intermediate," or "Advanced."
- **Level of Course**: Randomly assigned to one of the three categories: "Beginner," "Intermediate," or "Advanced."
- **Course Name**: Randomly assigned from a list of courses: "Math," "English," "Science," or "History."
- **Consistency**: Assigned as "Regular" or "Irregular," simulating how consistently the student engages with the course material.

2. **Transformation of Categorical Features**:
   - Categorical features such as Level of Student, Level of Course, and Consistency were converted into numerical values for machine learning. This was achieved by mapping each category to an integer value:
     - Level of Student: "Beginner" = 1, "Intermediate" = 2, "Advanced" = 3
     - Level of Course: "Beginner" = 1, "Intermediate" = 2, "Advanced" = 3
     - Consistency: "Regular" = 1, "Irregular" = 0

3. **Generation of Present Material Level**:
    - The Present Material Level feature was created based on the student's level and how it aligns with the course's material difficulty. This is crucial to represent whether the student is receiving the right level of content based on their abilities.
4. **Determining Material Level**:
    - The Material Level for each student was determined using a custom function based on multiple factors: assessment score, IQ, consistency, time spent studying, and the course name. The material level (Beginner, Intermediate, Advanced) is influenced by these variables, which were combined in a scoring formula.
5. **Feature Engineering**:
    - A new feature Relative Performance was calculated by subtracting a weighted sum of the student's and course's level from the student's assessment score, allowing us to measure how well a student is performing relative to their academic context.
    - Additional numerical mappings were also created for the features related to Present Material Level and Material Level.
6. **Visualization**:
    - Various visualizations were created to explore the distribution of features in the dataset, such as:
        - Age distribution
        - Assessment score distribution
        - Material level counts

- Correlation heatmap to visualize relationships between variables

The dataset was then saved as a CSV file for further use in training the predictive model.

2. Predicting Material Level Using Machine Learning

Once the dataset was created, the next phase of the project involved training a machine learning model to predict the material level (Beginner, Intermediate, or Advanced) based on the features.

Steps involved:

1. **Data Preparation**:
    - The dataset was loaded into a pandas DataFrame.
    - The target variable (Material Level) was separated from the features (X).
    - Categorical features such as Level of Student, Level of Course, and Course Name were encoded using one-hot encoding.
    - Numerical features were scaled using StandardScaler to normalize the data.
2. **Data Splitting**:
    - The data was split into training, validation, and test sets to evaluate the model's performance. 80% of the data was used for training and validation, with 20% reserved for testing.
    - The training set was further split into training and validation subsets, allowing for hyperparameter tuning and model evaluation during training.

3. **Model Selection and Hyperparameter Tuning**:
    - An **XGBoost classifier** was chosen for the task due to its excellent performance on classification problems, scalability, and robustness.
    - The hyperparameters of the XGBoost model were tuned using **GridSearchCV**, which performed an exhaustive search over a specified parameter grid, identifying the best model configuration.
        - Hyperparameters tuned include n_estimators, learning_rate, max_depth, subsample, and colsample_bytree.
4. **Model Training**:
    - After identifying the best hyperparameters, the final model was trained using the entire training dataset with early stopping on the validation set to prevent overfitting.
    - The model was trained with the objective of multi-class classification (multi:softmax), and the eval_metric used was mlogloss (multi-class logarithmic loss).
    - Early stopping was used during training to ensure that the model didn't overfit and to optimize training time.
5. **Model Evaluation**:
    - After training, the model was evaluated on the test set, and several evaluation metrics were calculated:
        - **Accuracy**: The proportion of correct predictions.
        - **Confusion Matrix**: A matrix showing the true vs predicted labels to assess how well the model is performing for each class.

- **Classification Report**: A report containing precision, recall, and F1-score for each class, which gives more detailed insights into the model's performance.

6. **Feature Importance**:
   - After training, the feature importance was analysed to understand which features were most influential in predicting the material level. This was achieved by extracting the feature importance scores from the trained XGBoost model.
   - A bar plot was created to visualize the top features that had the greatest impact on the model's predictions.
7. **Prediction Function**:
   - A prediction function was created to make predictions on new, unseen data. The function takes input data, processes it, and outputs the predicted material level.
   - The function involves preprocessing the input data (similar to the training data), making the prediction using the trained model, and converting the encoded prediction back to the original label (Beginner, Intermediate, Advanced).
8. **Prediction Example**:
   - Two example students were tested using the prediction function. For each student, features such as their age, IQ, daily study time, and assessment score were provided, and the model predicted the appropriate material level for them.

TECHNOLOGY STACK

The following technology stack was used in the development and implementation of the project:

1. Programming Languages

- **Python**: The primary programming language used for data generation, manipulation, machine learning, and model deployment. Python's rich ecosystem of libraries makes it an excellent choice for data science and machine learning tasks.

2. Data Processing & Analysis

- **Pandas**: Used for data manipulation and analysis. It provides data structures like DataFrames that make it easy to load, filter, clean, and manipulate datasets. In this project, it was used to create and modify the synthetic dataset.
- **NumPy**: Used for numerical computations, such as generating random numbers with specific distributions and performing mathematical operations.
- **Scikit-learn**: Utilized for preprocessing tasks (like scaling features and encoding categorical variables) and model evaluation. It also provides methods for splitting datasets into training, validation, and test sets.

3. Machine Learning

- **XGBoost**: A high-performance gradient boosting framework that was used to build the classification model for predicting material levels. It is known for its speed and performance in classification tasks.
- **GridSearchCV**: A feature of Scikit-learn used for hyperparameter tuning through exhaustive search. It was employed to optimize the XGBoost

model's hyperparameters by evaluating different combinations using cross-validation.

4. Data Visualization

- **Matplotlib**: A powerful plotting library used to generate visualizations such as histograms, scatter plots, and confusion matrices. It was used for visualizing distributions and model evaluation metrics.
- **Seaborn**: Built on top of Matplotlib, Seaborn was used to create more informative and attractive visualizations, such as heatmaps and count plots.

5. Model Deployment & Prediction

- **Scikit-learn (Label Encoding)**: Used for encoding the target labels ('Beginner', 'Intermediate', 'Advanced') into numerical values to feed into the machine learning model.
- **XGBoost**: Used for the final model deployment, enabling the prediction of material levels based on new student data

6. File Handling & I/O

- **CSV Files**: The dataset was saved and loaded as a CSV file for persistent storage. This makes it easy to handle the dataset, share it, and apply it across different stages of the pipeline.

7. Development Environment

- **IDE (Integrated Development Environment)**: Likely used for writing and testing Python code, such as PyCharm, Visual Studio Code, or Jupyter Notebook.

8. Package Management & Virtual Environments

- **pip**: The Python package manager used to install necessary libraries and manage dependencies for the project.

- **virtualenv**: This tool helps create isolated Python environments to ensure the project runs with the specific versions of libraries and dependencies required.

RESULTS AND DISCUSSION

1. Dataset Overview

The synthetic dataset was generated with a total of 1000 samples. The dataset contains various features, such as:

- **Age**: Ranges between 3 and 18 years.
- **IQ**: Normally distributed with a mean of 100 and standard deviation of 15.
- **Time per Day (hrs)**: Exponentially distributed with a mean of 1.5 hours.
- **Assessment Score**: Randomly assigned integers between 40 and 100.
- **Level of Student**: Categorical feature representing the student's level: 'Beginner', 'Intermediate', or 'Advanced'.
- **Level of Course**: Categorical feature representing the course level: 'Beginner', 'Intermediate', or 'Advanced'.
- **Course Name**: Categorical feature indicating the subject of the course, with options like 'Math', 'English', 'Science', and 'History'.
- **Consistency**: A binary feature indicating whether the student’s study habits are regular or irregular.

Additionally, new features were engineered, such as:

- **Present Material Level**: The level of material presented to the student, generated based on their current level.

- **Relative Performance**: A derived metric used to capture a student's performance in relation to their expected performance based on their level.

2. Visualization Insights

Several visualizations were used to better understand the dataset and the distribution of key features:

- **Age Distribution**: The age distribution of the students was approximately normal, with a mean around 12 years. The data was clipped to remove ages below 3 and above 18, ensuring it represented a typical school-aged population.
- **Assessment Score Distribution**: The assessment scores were uniformly distributed between 40 and 100, which is realistic for a variety of students at different academic levels.
- **Material Level Counts**: A count plot of the predicted material levels showed a reasonably balanced distribution across the three levels ('Beginner', 'Intermediate', 'Advanced'). This suggests the model could potentially learn patterns for each material level effectively.
- **Correlation Heatmap**: The correlation heatmap revealed that there were several correlations among features like 'IQ' and 'Assessment Score', and 'Time per Day (hrs)' and 'Assessment Score'. The feature **Relative Performance** exhibited a notable correlation with the 'Assessment Score', which is expected since it reflects how well a student is performing relative to their peers.

3. Model Performance

The model used to predict the material level was based on **XGBoost**, a gradient boosting machine (GBM) that is highly efficient for classification tasks. After

applying hyperparameter tuning via **GridSearchCV**, the following results were obtained:

- **Best Model Parameters**: The best model parameters found through GridSearchCV were:
    - n_estimators: 300
    - learning_rate: 0.05
    - max_depth: 4
    - subsample: 0.8
    - colsample_bytree: 0.8
    - gamma: 0.1
- **Metrics on Test Set**: The final model achieved an accuracy of approximately **89.4%** on the test set in predicting material level and R2 score of **0.72**. This accuracy and R2 score indicates that the model is well-calibrated and capable of making reliable predictions for the assessment score and material level of students based on their profile features.
- **Confusion Matrix**: The confusion matrix showed that the model performed well across all three classes, with the majority of misclassifications being between adjacent material levels. For example, 'Beginner' students were sometimes predicted as 'Intermediate', and 'Advanced' students were occasionally classified as 'Intermediate'. This could suggest that the boundaries between levels are somewhat blurry, making accurate predictions more challenging in some cases.

4. Feature Importance

The feature importance analysis revealed which features had the most influence on predicting the material level. The top features were:

1. **Assessment Score**: This feature had the highest importance, which makes sense as it directly reflects a student's performance and is likely to impact the level of material presented.
2. **Time per Day (hrs)**: The number of hours a student studies daily also had a significant influence, indicating that students who spend more time studying may be presented with more advanced material.
3. **IQ**: Although it didn't have the highest weight, IQ was still an important feature, reflecting its role in determining the cognitive ability of the students and thus influencing the level of material they should receive.
4. **Consistency**: Regular study habits (reflected by the "Consistency" feature) were found to be an important determinant, suggesting that consistent students perform better and may be assigned more advanced material.

5. Discussion

- **Model's Strengths**: The XGBoost model demonstrated strong predictive power, achieving high accuracy, precision, and recall. It was particularly effective at predicting the correct material level based on a combination of student demographics, study habits, and academic performance. The use of hyperparameter tuning further enhanced the model's ability to generalize to unseen data.
- **Challenges and Areas for Improvement**:
    - **Class Imbalance**: Although the dataset was fairly balanced, the model occasionally confused 'Beginner' students with 'Intermediate' students and vice versa. This indicates that the boundaries between

material levels might need further refinement, or more detailed features could be introduced.

- **Feature Selection**: While the current features captured a substantial portion of the variance in material level, additional features, such as past performance trends or teacher feedback, could further improve the model's accuracy.
- **Model Complexity**: Although XGBoost performed well, it can be computationally expensive for large datasets. For larger-scale implementations, models with faster inference times (such as logistic regression or lighter tree-based models) might be considered.

- **Real-World Application**: The approach used in this project could be applied in an educational setting where a learning management system (LMS) needs to dynamically adjust the level of material presented to students based on their current abilities. It can help provide personalized learning experiences that match the student's proficiency, optimizing their learning path.

6. Prediction Examples

Two example predictions were made using the trained model:

1. **Student 1 (Intermediate Student)**:
   - **Input**: Age = 14, IQ = 115.0, Time per Day (hrs) = 2.1, Assessment Score = 85, Level of Student = Intermediate, Level of Course = Intermediate, Course Name = Math, Consistency = Regular, Relative Performance = 10.0
   - **Predicted Material Level**: Intermediate.
2. **Student 2 (Beginner Student)**:

- **Input**: Age = 9, IQ = 92.0, Time per Day (hrs) = 0.7, Assessment Score = 58, Level of Student = Beginner, Level of Course = Beginner, Course Name = English, Consistency = Irregular, Relative Performance = -2.0
- **Predicted Material Level**: Beginner.

These examples demonstrate the model's capability to handle various student profiles and predict the appropriate material level accordingly.

CONCLUSION

The AI-Powered Personalized Tutor System demonstrates the potential of machine learning to revolutionize K-12 virtual education by delivering customized learning experiences. By leveraging student data and employing the XGBoost classifier, the system effectively predicts appropriate content levels and assessment outcomes, supporting individualized academic growth. The use of synthetic data illustrates the model’s adaptability across diverse learner profiles. With promising performance metrics, the project highlights the feasibility of intelligent tutoring systems in enhancing educational outcomes. Future work may include integrating real-time feedback and LMS support to create a more dynamic, responsive, and practical tool for modern digital classrooms.