1. Overview

Data preparation and feature engineering form the backbone of our machine learning pipeline. We believe that no matter how advanced the model, its performance depends heavily on the quality of data fed into it. For our PLMS project, we spent considerable time cleaning and transforming the raw data to make it suitable for analysis. This included dealing with missing values, converting data into usable formats, and crafting new features that we believed would help the model learn student behavior more effectively. By focusing on this phase, we aimed to ensure our model could make accurate and meaningful predictions.

2 .Data Source and Preprocessing

To conduct this study, data was gathered through two primary means:

1. **Authentic Data Collection:**
   A Google Form was used to collect real-world responses from students. The form included questions related to internet access, average time spent on digital entertainment, preferred learning materials, and the usefulness of various study resources such as textbooks, YouTube channels, and online platforms.
2. **Synthetic Data Generation:**
   To enrich the dataset and simulate a larger population, synthetic data was generated using the scikit-learn library. The synthetic data closely mirrors real student behavior patterns and preferences, ensuring the dataset remains representative of typical learning scenarios while addressing limitations in the sample size of authentic responses.

Preprocessing Steps:

- Standardization: Inconsistent text formats (e.g., extra spaces in categorical values like "Yes " or "No") were cleaned using string stripping functions.
- Encoding: Ordinal categorical values such as "Internet access" (No, Sometimes, Yes) and "Average hours on digital entertainment" were mapped to numerical scales to support further analysis and modeling.
- Data Cleaning: Columns irrelevant to analysis such as Timestamp, Email Address, and Consent were removed.
- Handling Missing Values: Numeric columns with missing values were imputed using the median to maintain distribution integrity.
- Validation: A review was conducted to ensure data types matched their expected formats, and that all entries aligned logically (e.g., age values extracted numerically from text).

3. Data Cleaning

To ensure the quality and reliability of the analysis, several steps were taken to clean the raw student learning dataset:

3.1 Removing Irrelevant or Redundant Columns

The raw data contained fields that were either administrative (e.g., timestamp, email address) or related to consent and feedback questions, which were not necessary for analyzing learning behaviors. These columns were removed to reduce noise and streamline the dataset, focusing only on features directly influencing the study.

3.2 Standardizing Categorical Responses

Some of the text-based fields, like "Internet Access" and "Average Hours on Digital Entertainment," had inconsistencies such as extra spaces at the beginning or end. These inconsistencies can cause the same category to be treated as different entries. To avoid this issue and ensure that the categories were correctly grouped, the text responses were standardized by removing any unwanted spaces.

3.3 Encoding Categorical Variables Numerically

Since machine learning models and statistical methods typically require numerical inputs, categorical text responses were converted into numeric form:

- For **Internet Access**, responses were mapped to a scale (e.g., "No" → 1, "Sometimes" → 2, "Yes" → 3) to represent increasing levels of access.
- For **Average Hours Spent on Digital Entertainment**, the categories were mapped to approximate numerical values based on the midpoints of the ranges given (e.g., "Less than 1 hr" → 0.5, "2-4 hrs" → 3).

This step enabled easier quantitative analysis and modeling.

3.4 Extracting and Converting Age Values

The age data was originally stored in a format that included text. To make it usable for analysis, the numerical part of the age was extracted and converted into a float type. This transformation allowed age to be treated properly in descriptive statistics, visualizations, and modeling.

3.5 Identifying and Handling Missing Values

The dataset was checked for missing values across all columns. It was found that some numerical fields had missing entries.
To maintain data consistency without introducing bias, missing values were filled using the **median** of each column instead of the mean. The median is less sensitive to outliers and

provides a better central tendency measure when data may not be perfectly normally distributed.

3.6 Reviewing Data Distributions for Outliers

Summary statistics (like minimum, maximum, mean, and standard deviation) were reviewed to detect any unusual or extreme values. No extreme outliers were found in critical fields such as age or average hours spent on digital entertainment. As a result, no additional filtering or transformations for outlier handling were required.

3.7 Final Validation

After completing the cleaning steps:

- All categorical fields were properly standardized and encoded.
- All missing values were handled.
- The numerical columns were properly formatted for analysis.

This ensured that the final dataset was complete, consistent, and ready for exploratory data analysis and model development.

4 Exploratory Data Analysis (EDA)

After cleaning the dataset, an Exploratory Data Analysis (EDA) was conducted to understand the underlying patterns, relationships, and characteristics of the data. This phase helped uncover key insights and informed future modeling decisions.

4.1 Understanding the Distribution of Age

The students' age distribution was explored by creating a histogram.
The analysis showed that most students were clustered around the late teenage years (17–19 years old), which is consistent with the typical age range for high school students preparing for the Ethiopian University Entrance Exam (EUEE).
The distribution was relatively normal, with no extreme outliers, indicating a well-targeted sampling of the intended population.

Key Insight:

The majority of users fall into the key age group for intervention (EUEE preparation), confirming that the system’s personalization will target the appropriate audience.

4.2 Assessing Internet Access Among Students

Internet access is critical because the personalized learning system depends heavily on digital resources.

A bar chart was plotted to show the distribution of internet access levels among students.
It revealed that:

- A significant portion of students had **full access** to the internet.
- Some students reported **partial access** ("Sometimes"), indicating occasional connectivity challenges.
- A smaller but notable group **lacked regular internet access**, which would need to be considered for offline content delivery solutions.

Key Insight:

While many students have reliable internet access, designing content with low-data or offline options would help ensure inclusivity for underserved students.

4.3 Summary Statistics of Key Variables

Descriptive statistics (mean, median, min, max, and standard deviation) were calculated for all numeric fields, including:

- Age
- Average hours spent on digital entertainment
- Encoded Internet access levels

These summaries helped identify the central tendencies and variation in student behaviors and backgrounds.
No irregularities or extreme skewness were observed, suggesting the synthetic data appropriately modeled realistic student characteristics.

Key Insight:

The dataset covers a reasonable range of behaviors and backgrounds, which supports building a robust recommendation system.

4.4 Missing Values and Data Integrity Check

After cleaning, an additional check confirmed that no missing values remained.
This validation step ensured that the visualizations and insights drawn from the data were based on a complete and consistent dataset.

Conclusion of EDA

The exploratory analysis verified that:

- The dataset accurately represents the target student population.
- Internet access disparities need to be accounted for in system design.

- The synthetic data is suitable for modeling learning patterns and testing recommendation algorithms.

The insights gained from EDA will guide the development of personalized learning paths and help tailor interventions more effectively for Ethiopian high school students.

5. Feature Engineering

In this project, feature engineering was performed to enhance the predictive power of the dataset and to capture underlying relationships between variables. The following steps were taken:

5.1 Creation of New Features

Age Encoding:
The Age column contained numeric values but was inconsistent in format. To ensure numerical consistency and allow easier modeling, a new column Age_Encoded was created by converting Age to a numeric type.

```
df['Age_Encoded'] = pd.to_numeric(df['Age'], errors='coerce')
```

Interaction Feature: A new interaction feature, Entertainment_Age_Interaction, was created by multiplying the standardized number of hours spent on digital entertainment with the standardized age. This feature was designed to capture any interactive effects between entertainment consumption and age in influencing learning material ratings.

```
df['Entertainment_Age_Interaction'] = df['Avg hrs on Digital Entmnt (eg Gaming, Instagram, Youtube)'] * df['Age_Encoded']
```

➔5.2 Feature Selection

- After basic feature engineering, non-informative columns like Consent, Timestamp, and Email Address were dropped to eliminate noise from the dataset.
- Rating columns were selected for prediction (rating_columns) as the main target variables.

6. Data Transformation

Various data transformations were applied to prepare the dataset for machine learning algorithms:

6.1 Categorical Encoding

- Certain categorical variables were manually encoded into numerical values to enable machine learning models to interpret them:

Internet Access was encoded as:
internet_mapping = {'Yes': 1, 'No': 0}

```
df['Internet access_Encoded'] = df['Internet access'].map(internet_mapping)
```

Average Hours on Digital Entertainment responses were encoded based on predefined intervals:

```
avg_hours_mapping = {
  '<1 hour': 0.5,
  '1-2 hours': 1.5,
  '3-4 hours': 3.5,
  '5-6 hours': 5.5,
  '7-8 hours': 7.5,
  '>8 hours': 9
}
df['Avg_hrs_Digital_Entertainment_Encoded'] = df['Avg hrs on Digtal Entmnt (eg Gaming, Instagram, Youtube)'].map(avg_hours_mapping)
```

6.2 Scaling and Standardization

StandardScaler was used to standardize Age_Encoded and the newly created interaction feature. Standardization centers the data around the mean (0) and scales it to unit variance (1), improving convergence speed for some models and making features comparable.

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
```

```
df[['Age_Encoded', 'Entertainment_Age_Interaction']] = scaler.fit_transform(df[['Age_Encoded', 'Entertainment_Age_Interaction']])
```

6.3 Normalization

Min-Max Normalization was applied to all the rating columns to rescale them between 0 and 1. This ensures that each rating feature contributes equally during model training.

```
from sklearn.preprocessing import MinMaxScaler

scaler = MinMaxScaler()

df[rating_columns] = scaler.fit_transform(df[rating_columns])
```

6.4 Handling Outliers and Skewness

Outliers in numeric features were detected using IQR and Z-score methods. Although not removed, skewness in numeric features was corrected by applying log transformation (np.log1p()), which reduces the impact of extreme values and improves model stability.

```
df[numeric_cols] = df[numeric_cols].apply(lambda x: np.log1p(x))
```

Model selection

For this trial, we used Decision Tree (DT) as a baseline model. While it provided good precision with low MAE and RMSE, the R^2 scores revealed it struggles to capture complex relationships. To improve the recommendation system, the following machine learning models will be explored:

1. Random Forest: Captures nonlinear patterns and provides robust predictions.
2. Gradient Boosting Models (e.g., XGBoost, LightGBM): Excellent for high precision, capable of handling intricate relationships.
3. Support Vector Machines (SVMs): Effective for modeling nonlinear relationships.
4. Matrix Factorization (e.g., SVD): Tailored for collaborative filtering, capturing user-material interactions.

Future iterations will focus on these models to enhance performance and achieve better R^2 scores, making the system more robust and accurate.

Model Training

Decision Tree Model Documentation(v0.01)(Here is the link for this version notebook for future information

1. Model Overview

For this analysis, we used a Decision Tree Regressor (DT) as the baseline machine learning model to predict user ratings for various study materials. The decision tree is well-suited for capturing non-linear relationships and is interpretable, but requires further enhancements for precision and generalization.

2. Hyperparameters Used

- max_depth=5: The maximum depth of the tree was limited to 5 to avoid overfitting and ensure model simplicity.
- random_state=42: Ensured reproducibility by fixing the random seed for train-test splitting and tree building.

These hyperparameters were selected to balance model complexity and performance.

3. Preprocessing and Feature Engineering

- Features Used:
    - Age_Encoded: Scaled using StandardScaler to normalize the feature.
    - Internet access_Encoded: Encoded as a binary categorical variable.
    - Avg hrs on Digital Entmnt_Encoded: Encoded as a numerical feature.
    - Entertainment_Age_Interaction: An interaction term capturing the relationship between digital entertainment hours and age.
- Feature Normalization:
    - All numeric features were standardized using StandardScaler to ensure consistent scaling across the dataset.
    - Ratings were normalized using min-max scaling across all study materials.
- Outlier Handling:
    - Key features were transformed using scaling techniques to handle extreme values and ensure smooth distribution.

4. Training and Validation Process

- Data Splitting:
  The dataset was split into training and testing sets using an 80:20 ratio to evaluate the

model on unseen data:
X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.2, random_state=42)

- Cross-Validation:
  A simple train-test split was used for this trial. No k-fold cross-validation was implemented, but we will use it for future iterations to improve model robustness.
- Prediction:
  The Decision Tree was trained for each study material separately. Predictions were made on the test set for each material.

5. Evaluation Metrics

The model was evaluated using the following metrics:

- Mean Absolute Error (MAE): Measures average absolute error, reflecting overall prediction accuracy.
- Mean Squared Error (MSE): Measures squared differences, penalizing larger errors more heavily.
- Root Mean Squared Error (RMSE): Takes the square root of MSE for interpretability in the same unit as the target variable.
- R^2 Score: Evaluates how well the model explains the variance in ratings. Values close to 1 indicate better performance.

6. Observations

The results varied across study materials. For example:

- Best Performance: Royal Math, which had the lowest MAE and RMSE, and the highest R^2 score of 0.26.
- Weakest Performance: Materials like Ethio Matric and Exam Questions, which had lower R^2 values (0.18 and 0.03, respectively).
- MAE and RMSE: The model demonstrates good precision, with relatively low prediction errors across all materials.
- R^2 Scores: The model's explanation of variance is low (ranging from 0.03 to 0.26), indicating that the features currently used do not fully capture the variability in ratings.

7. Our future work

- Cross-Validation: Implement k-fold cross-validation to ensure consistent results across multiple train-test splits. (didn't do it because of time constraint)
- Feature Enrichment:
  - Introduce additional features that could explain variability in ratings, such as difficulty level or material type.
  - Engineer further interaction terms to capture hidden relationships.
- Advanced Models:

- Replace the Decision Tree with ensemble models like Random Forest or Gradient Boosting (XGBoost, LightGBM) for better predictive power.
- Explore collaborative filtering techniques (e.g., SVD) to enhance personalization.

Baseline DT model training result findings

Performance Overview:

Material	MAE	MSE	RMSE	R^2 Score
Royal Math	0.16	0.08	0.08	0.26
YouTube: Blackpen Redpen	0.26	0.12	0.12	0.10
YouTube: OCT	0.31	0.14	0.14	0.20
Ethio Matric	0.33	0.16	0.16	0.18
Exam Questions	0.20	0.09	0.09	0.03
TB	0.20	0.08	0.08	0.19
XBS	0.21	0.08	0.08	0.12
Telegram Channels (Notes)	0.28	0.12	0.12	0.24
Indian Websites (Byjus)	0.25	0.10	0.10	0.18
Key Exam Book	0.29	0.13	0.13	0.19
Top Exam Book	0.29	0.13	0.13	0.21
YouTube: Ethioeducation	0.32	0.15	0.15	0.08
YouTube: The Secret	0.32	0.17	0.17	0.21
Galaxy Math	0.22	0.09	0.09	0.10

Regression Model Evaluation:
Mean Absolute Error (MAE): 0.16
Mean Squared Error (MSE): 0.08
Root Mean Squared Error (RMSE): 0.08
R² Score: 0.26

Material: YouTube: Blackpen Redpen

Regression Model Evaluation:
Mean Absolute Error (MAE): 0.26
Mean Squared Error (MSE): 0.12
Root Mean Squared Error (RMSE): 0.12
R² Score: 0.10

Material: YouTube: OCT

Regression Model Evaluation:
Mean Absolute Error (MAE): 0.31
Mean Squared Error (MSE): 0.14
Root Mean Squared Error (RMSE): 0.14
R² Score: 0.20

Material: XBS

Regression Model Evaluation:
Mean Absolute Error (MAE): 0.21
Mean Squared Error (MSE): 0.08
Root Mean Squared Error (RMSE): 0.08
R² Score: 0.12

Material: Telegram Channels (Notes and Questions)

Regression Model Evaluation:
Mean Absolute Error (MAE): 0.28
Mean Squared Error (MSE): 0.12
Root Mean Squared Error (RMSE): 0.12
R² Score: 0.24

Material: Indian Websites (Byjus)

Regression Model Evaluation:
Mean Absolute Error (MAE): 0.25
Mean Squared Error (MSE): 0.10
Root Mean Squared Error (RMSE): 0.10

Regression Model Evaluation:
Mean Absolute Error (MAE): 0.33
Mean Squared Error (MSE): 0.16
Root Mean Squared Error (RMSE): 0.16
R² Score: 0.18

Material: Exam Questions

Regression Model Evaluation:
Mean Absolute Error (MAE): 0.20
Mean Squared Error (MSE): 0.09
Root Mean Squared Error (RMSE): 0.09
R² Score: 0.03

Material: TB

Regression Model Evaluation:
Mean Absolute Error (MAE): 0.20
Mean Squared Error (MSE): 0.08
Root Mean Squared Error (RMSE): 0.08
R² Score: 0.19

R² Score: 0.18

Material: Key Exam Book

Regression Model Evaluation:
Mean Absolute Error (MAE): 0.29
Mean Squared Error (MSE): 0.13
Root Mean Squared Error (RMSE): 0.13
R² Score: 0.19

Material: Top Exam Book

Regression Model Evaluation:
Mean Absolute Error (MAE): 0.29
Mean Squared Error (MSE): 0.13
Root Mean Squared Error (RMSE): 0.13
R² Score: 0.21

Material: YouTube: Ethioeducation

Regression Model Evaluation:
Mean Absolute Error (MAE): 0.32

Mean Squared Error (MSE): 0.15
Root Mean Squared Error (RMSE): 0.15
R² Score: 0.08

Material: YouTube: The Secret

Regression Model Evaluation:
Mean Absolute Error (MAE): 0.32
Mean Squared Error (MSE): 0.17
Root Mean Squared Error (RMSE): 0.17
R² Score: 0.21

Material: Galaxy Math

Regression Model Evaluation:
Mean Absolute Error (MAE): 0.22
Mean Squared Error (MSE): 0.09
Root Mean Squared Error (RMSE): 0.09
R² Score: 0.10