**GITHUB LINK:** [**https://github.com/Sukran-03/Sukran-03**](https://github.com/Sukran-03/Sukran-03)

**PROJECT TITLE: PERSONALIZING E-LEARNING EXPERIENCES USING STUDENT ENGAGEMENT AND PERFORMANCE ANALYTICS**

**PHASE-2**

1. **PROBLEM STATEMENT:**

In traditional e-learning systems, all learners are often provided with the same content and learning path, regardless of their individual engagement levels, learning styles, or academic performance.

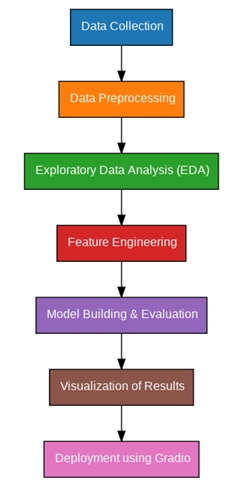
This one-size-fits-all approach fails to address the diverse needs of students, leading to reduced motivation, inconsistent performance, and high dropout rates. Although large amounts of student interaction and performance data are generated, they are rarely leveraged effectively to adapt the learning experience in real-time.

There is a pressing need for a system that can analyze student engagement and performance data to dynamically personalize content delivery, pacing, and instructional strategies, thereby improving learning outcomes and overall student satisfaction.

1. **PROJECT OBJECTIVES:**

* To collect and analyze student engagement data such as time spent on activities, interaction frequency, and participation levels.
* To monitor academic performance through assessments, quizzes, and progress tracking.
* To identify patterns and correlations between engagement levels and learning outcomes.
* To develop a personalized learning model using data analytics and machine learning techniques.
* To adapt content delivery, pacing, and difficulty based on individual learner profiles.
* To improve student motivation, comprehension, and retention through tailored learning paths.
* To evaluate the effectiveness of the personalized system in enhancing overall academic performance.
* To provide real-time feedback and recommendations to both students and instructors for continuous improvement
* To ensure scalability and flexibility of the system for integration with existing e-learning platforms and diverse educational environments.

1. **FLOWCHART OF THE PROJECT WORKFLOW:**

****

1. **DATA DESCRIPTION:**

* **Dataset Name:** E-Learning Data Set
* **Source:** kaggle
* **Type of Data:** Structured tabular data
* **Records and Features**: 2501 student records and 10 features (numeric + categorical)
* **Target Variable:** G3 (final grade, numeric)
* **Static or Dynamic:** Static dataset
* **Attributes Covered:** Demographics (age, address, parents’ education), academics (G1, G2, study time), and behavior (alcohol consumption, absences)
* Dataset Link: [E-Learning student dataset](https://www.kaggle.com/datasets/vikantkumar/e-learning-student-dataset/data)

1. **DATA PREPROCESSING:**

* Gather engagement and performance data from e-learning platforms (e.g., time spent, quiz scores, activity logs).
* Combine data from multiple sources into a unified format using consistent identifiers like student ID.
* Address missing data by removing, imputing, or flagging incomplete records.
* Remove duplicates, correct inconsistencies, and standardize formats (e.g., timestamps, names).
* Create new features such as engagement scores, activity frequency, or time-on-task metrics.
* Convert categorical variables (e.g., course type, learning style) into numeric format using label or one-hot encoding.
* Normalize or scale numerical features to ensure consistent data ranges for model training.
* Identify and treat unusual values that could skew model predictions using statistical or clustering techniques.

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

#### **1. Univariate Analysis**

#### **Engagement Time**: Plot histograms or box plots to understand the distribution of time spent by students on the platform.

* **Quiz Scores**: Analyze score distributions using histograms to identify common performance levels.
* **Login Frequency**: Examine how often students log in per week/month.
* **Dropout Flags**: Countplot to see the number of students who drop out or disengage early.

#### **2. Bivariate Analysis**

#### **Time Spent vs Quiz Scores**: Use scatter plots or line plots to see if more engagement time results in better performance.

* **Forum Participation vs Grades**: Analyze the correlation between discussion activity and academic outcomes.
* **Login Frequency vs Completion Rate**: Use bar plots or box plots to study trends between login patterns and course completion.
* **Course Type vs Performance**: Compare average grades across different types of course

## **3. Multivariate Analysis**

#### **Time Spent, Quiz Scores, and Course Difficulty**: Use 3D plots or heatmaps to see how these interact.

* **Engagement Score vs Performance across Student Groups**: Use grouped bar charts or faceted scatter plots to compare patterns by gender, age group, or prior academic level.
* **Clustering of Student Behavior**: Apply PCA or t-SNE for dimensionality reduction and visualize student clusters based on multiple features like time spent, forum activity, and scores.

#### **4. Key Insights**

#### Students with higher engagement (time spent, participation) tend to perform better academically.

* Quiz performance and login frequency are moderately correlated — consistent engagement supports better learning.
* Some students spend a lot of time but still perform poorly, indicating possible learning difficulties or content misalignment.

1. **FEATURE ENGINEERING:**

* Created engagement metrics like total time spent, login frequency, and forum activity levels.
* Derived performance features such as average quiz scores, improvement rates, and completion percentages.
* Engineered behavioral features like session patterns, revisit counts, and preferred study times.
* Developed composite indicators such as engagement score and performance index.
* Encoded categorical data (e.g., course type, learning style) into numerical formats for model training.

1. **MODEL BUILDING:**

* **Model Selection**
* Chose suitable machine learning models such as Decision Trees, Random Forest, or XGBoost for prediction tasks (e.g., performance prediction, dropout risk).
* **Training and Validation**
* Split the preprocessed data into training, validation, and test sets (e.g., 70/15/15) to avoid overfitting and ensure model generalization.
* **Personalization Approach**
* Implemented recommendation models (e.g., collaborative filtering or content-based filtering) to personalize content based on engagement and performance.
* **Hyperparameter Tuning**
* Used techniques like Grid Search or Random Search to optimize model parameters for better accuracy and performance.
* **Evaluation Metrics**
* Assessed model performance using metrics such as accuracy, precision, recall, F1-score (for classification) or RMSE/MAE (for regression).

1. **VISUALIZATION OF RESULTS & MODEL INSIGHTS:**

* **Engagement Analytics Dashboard**
* Heatmaps showing which content/pages students spend most time on.
* Time-series plots for tracking individual or group engagement over time (e.g., login frequency, video watch duration).
* Drop-off points visualized in courses (where students stop interacting).
* **Performance Analytics**
* Box plots comparing scores before and after personalization.
* Bar charts of average quiz scores across different learner clusters (e.g., high engagement vs. low engagement).
* Confusion matrices (if using classification models) to show prediction accuracy for learner success/failure.
* **Personalization Insights**
* Feature importance plots from models (e.g., from decision trees or SHAP values) to show which engagement factors most influence performance.
* Learner clustering visualizations using t-SNE or PCA to show how students are grouped based on behavior and outcomes.
* Intervention mapping, showing which personalized strategies (e.g., reminders, content difficulty adjustment) were triggered and their effects.
* **Case Studies or Sample Learner Journeys**
* Individual learner timelines showing how content was personalized and how performance improved.
* Before/after dashboards highlighting improvements from adaptive learning paths.
* **Model Performance Metrics**
* Accuracy, precision, recall, F1 score plots for predictive models.
* ROC curves or learning curves to show model behavior during training.

1. **TOOLS AND TECHNOLOGIES USED:**

**Programming Language:** Python

**Notebook Environment:** Google Colab

**Key Libraries:**

* pandas, numpy for data handling
* matplotlib, seaborn, plotly for visualizations
* scikit-learn for preprocessing and modeling
* Gradio for interface deployment

1. **TEAM MEMBERS AND CONTRIBUTIONS:**
2. **Alexander -** **Data cleaning,** **EDA**
3. **Aravind - Feature engineering**
4. **Sukran - Model development**
5. **Mohamed Riyaskhan-** **Documentation and reporting**