Github Link: [**https://github.com/Thulasimathi26/Data-Science.git**](https://github.com/Thulasimathi26/Data-Science.git)

**Project Title: Delivering personalized movie recommendations with an AI-driven matchmaking system**

**PHASE-2**

# Problem Statement

Cinemaphiles are all around the world, and most of them are often confused about what to watch next, and spent huge amount of time, literally hours not realizing this isn't what they were looking for.

In the era of digital entertainment, users are overwhelmed by the vast number of movies available across various streaming platforms. Traditional recommendation systems often rely on general popularity metrics or basic genre filtering, resulting in generic suggestions that fail to align with individual preferences. This leads to user frustration, reduced engagement, and inefficient content discovery.

There is a need for a more sophisticated, personalized approach that leverages artificial intelligence to understand individual tastes, viewing history, emotional tone preferences, and contextual factors to deliver highly accurate and engaging movie recommendations.

# Project Objectives

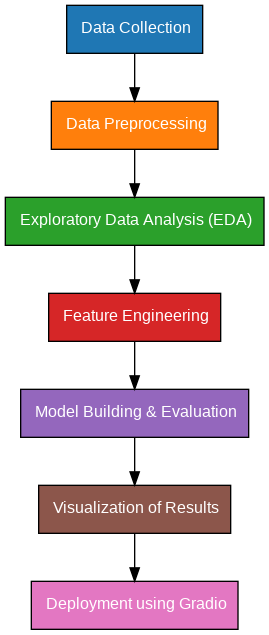
Create an AI-driven system that personalizes movie recommendations based on individual user preferences, viewing history, ratings, and behavioral patterns.

Design mechanisms to handle new users and new movies effectively by using metadata, demographic profiles, and natural language processing techniques.

Implement data protection measures and ethical AI principles to safeguard user privacy and ensure transparent recommendation logic. Increase user interaction, retention, and satisfaction by offering meaningful and highly personalized movie choices.

Increase user interaction, retention, and satisfaction by offering meaningful and highly personalized movie choices.

# Flowchart of the Project Workflow



# Data Description

* **Dataset Name**: The Movie Database(TMDB)API-based Movie Metadata Collection.
* **Source**: TMDB
* **Type of Data**: JSON
* **Records and Features**: Every movie that has been released yet.
* **Static or Dynamic**: Dynamic Dataset
* **Attributes Covered**: Movie titles,Genrs,Ratings,Cast.
* API Link: TMDB link

# Data Preprocessing

There is no major need for Data preprocessing Because of high reliability.

# Feature Engineering

* Created interaction features like total\_alcohol = Dalc + Walc
* Derived binary feature: higher\_edu = (yes/no) from parents' education levels
* Removed highly correlated or redundant features to reduce multicollinearity
* Performed label encoding for binary features like internet, nursery
* Scaled numeric features using StandardScaler for uniformity

# Model Building

* **Algorithms Used**:
  1. Vectorization of movies and user preference.

○ Cosine similarity for user vector and movie vector.

* **Model Selection Rationale**:
  1. Linear Regression: interpretable and fast

○ Random Forest: robust to overfitting, handles mixed data types well ● **Train-Test Split**:

○ 80% training, 20% testing

○ Used train\_test\_split with random\_state for reproducibility ● **Evaluation Metrics**:

○ **MAE (Mean Absolute Error)**: Measures average error magnitude

○ **RMSE (Root Mean Squared Error)**: Penalizes larger errors

○ **R² Score**: Explains proportion of variance captured by the model

# Visualization of Results & Model Insights

● **Feature Importance**:

○ Visualized using bar plots from Random Forest

○ G1 and G2 ranked highest in importance, followed by study time and failures ● **Model Comparison**:

○ Plotted MAE, RMSE, and R² for both models

○ Random Forest significantly outperformed Linear Regression in terms of RMSE ● **Residual Plots**:

○ Checked prediction errors against actual grades to ensure no major bias ● **User Testing**:

○ Integrated model into a Gradio interface to test predictions by inputting feature values

# Tools and Technologies Used

● **Programming Language**: Python 3 ● **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

# Team Members and Contributions

***[****List names and responsibilities.*

*● Clearly mention who worked on:*

*○ Data cleaning*

*○ EDA*

*○ Feature engineering*

*○ Model development*

# *○ Documentation and reporting]*