📘 **Project Title:**

Personalized Course Recommendation System Using Matrix Factorization

🎯 **Objective**

To build a recommendation system that suggests relevant courses for new users based on inferred interests from past enrollment behaviors, without relying on explicit user ratings.

🗂️ **Dataset Overview**

- **Primary Source:** A curated dataset of online courses with user enrollment history.
- **Key Fields:**
  - title – Course name
  - enrollment_number – Number of users enrolled
  - rating – Course rating (optional enhancement)
  - Other fields such as subject, university, etc. (not heavily used in this build)

🧪 **Exploratory Data Analysis (EDA)**

✅ **Key Insights:**

- **Course popularity** follows a long-tail distribution—few courses have very high enrollments.
- **Rating distribution** showed a central tendency near 4.0, with a mild right skew.
- Checked for:
  - Null values and dropped or imputed as needed
  - Course title duplication
  - Distribution of enrollment_number using histograms

📊 **Visualizations:**

- **Histogram of Enrollment Numbers**
- **Bar chart of Top 10 Most Popular Courses**
- **Box plot of Course Ratings**
- **Heatmap (optional) of course-user matrix density**

🧠 **Modeling Approach**

1️⃣ **Collaborative Filtering with Matrix Factorization**

We used a **latent factor model** to map users and courses into a shared feature space:

- user_course_matrix created from enrollments
- **Matrix Decomposition** yielded:
  - enrollment_factors – representation of users in latent space
  - course_profiles – latent representation of each course

✅ This setup allowed us to score unseen courses using the dot product:

python

```
user_vector.dot(course_profiles.T)
```

2️⃣ **Enrollment Matching for Cold Starts**

New users without historical data are mapped by the **closest enrollment match**:

python

```
closest_enrollment = np.abs(numeric_index - enrollment_number).argmin()
```

This provides an **entry point for cold-start users** based on similar prior behaviors.

3️⃣ **Scoring & Ranking**

- Used **dot product** between user vector and transposed course embeddings
- Ranked top-N predicted scores using pd.Series(...).sort_values()

4️⃣ **Enhancements**

- **MinMax Normalization** for better interpretability of scores
- **Weighted scoring with course ratings**
- (Optional) Blended with **content-based similarity scores**

📈 **Key Visualizations Developed**

Purpose	Visualization
Data quality	missingno.matrix() to visualize missing values
Popularity trends	sns.histplot() of enrollments
Recommendation quality	Top-N predicted scores for sample users
Score interpretability	Bar plot of normalized predictions

Purpose	Visualization
Course embeddings (optional)	2D PCA projection of course_profiles

✅ **Outcomes**

- **Generated personalized recommendations** for new users purely based on enrollment proximity.
- **Handled common errors**: index mismatches, shape misalignments, missing labels.
- Created a **robust and modular pipeline** ready for scaling or deployment.
- **Successfully visualized** the data journey from raw values to intelligent suggestions.