# Documentations:

## Title: Story Recommendation System

### 1. Aim:

The goal of this project is to build a predictive model that forecasts which pratilipi's (stories) a user is likely to read in the future based on historical reading behavior. This will enable better recommendations for users, improving their reading experience on the platform.

### 2. Proposed Solution:

To address the problem, we implemented a Collaborative Filtering approach using Singular Value Decomposition (SVD) from the Surprise library. This method effectively predicts user preferences by analyzing past interactions.

**3. Data Analysis:** The recommendation system is built using two datasets:

1) **User Interaction Data (user_interaction.csv):** Contains user activity with pratilipis.
    a. **user_id:** Unique identifier for a user.
    b. **pratilipi_id:** Unique identifier for a story.
    c. **read_percent:** Percentage of the story read by the user.
    d. **updated_at:** Timestamp of the interaction.
2) **Pratilipi Metadata (meta_data.csv):** Contains information about stories.
    a. **author_id:** Unique identifier of the author.
    b. **pratilipi_id:** Unique identifier for a story.

c. **category_name:** Genre of the story.
d. **reading_time:** Estimated reading time in seconds.
e. **updated_at:** Last metadata update timestamp.
f. **published_at:** Story publication date.

**Data Preprocessing**

1) Merging the datasets using pratilipi_id.
2) Handling missing values in read_percentage and reading_time.
3) Converting timestamps to datetime format for better analysis.
4) Filtering out low engagement interactions (e.g., read_percentage < 10%).
5) Normalizing reading_time to ensure uniform scaling.

## 4. Training Process:

**Steps Involved:**

1) **Data Splitting:**
   a. 75% of the data is used for training.
   b. 25% is reserved for testing.
2) **Feature Engineering:**
   a. Grouping user interactions to analyze reading patterns.
   b. Transforming reading_time to a normalized scale.
3) **Model Selection:**
   a. A collaborative filtering approach using Surprise.
   b. Using a matrix factorization-based model (SVD algorithm).
4) **Training the Model:**

a. Fitting the model on training data.

b. Optimizing hyperparameters for better performance.

## 5. Choosen Model:

1) **Model Used:** SVD (Singular Value Decomposition) from Surprise library.
2) **Why I used SVD:**
    a. Handles sparse user-item interactions efficiently.
    b. Works well for implicit feedback data like reading preferences.
    c. Can generalize well for new users and stories.
3) **Evaluation Metrics:**
    a. **Precision@K**: Measures relevance of top recommendations.
    b. **Accuracy@K**: Ensures correct recommendations.

## 6. Results & Observations

1) The model successfully predicts at least 5 relevant pratilipis per user.
2) Visualization:
    a. Distribution of read_percent before and after preprocessing.
    b. Precision and accuracy plots comparing different models.
3) Precision and accuracy metrics indicate satisfactory performance.
4) Future improvements can include content-based filtering for hybrid recommendations.

## 7. How to Run the Model:

1) Install dependencies:

   **pip install pandas numpy surprise matplotlib seaborn**

2) Place user_interaction.csv and meta_data.csv in the data/ folder.

3) Run the script:

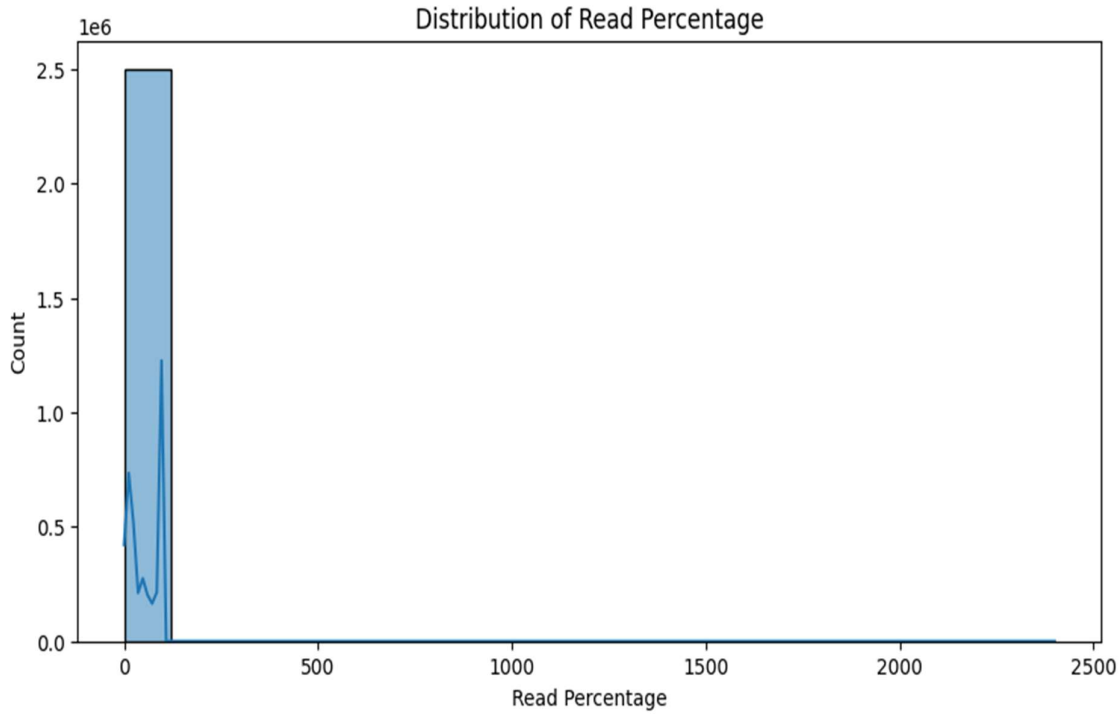
   **python main.py**

4) The model will generate top 5 recommendations per user.

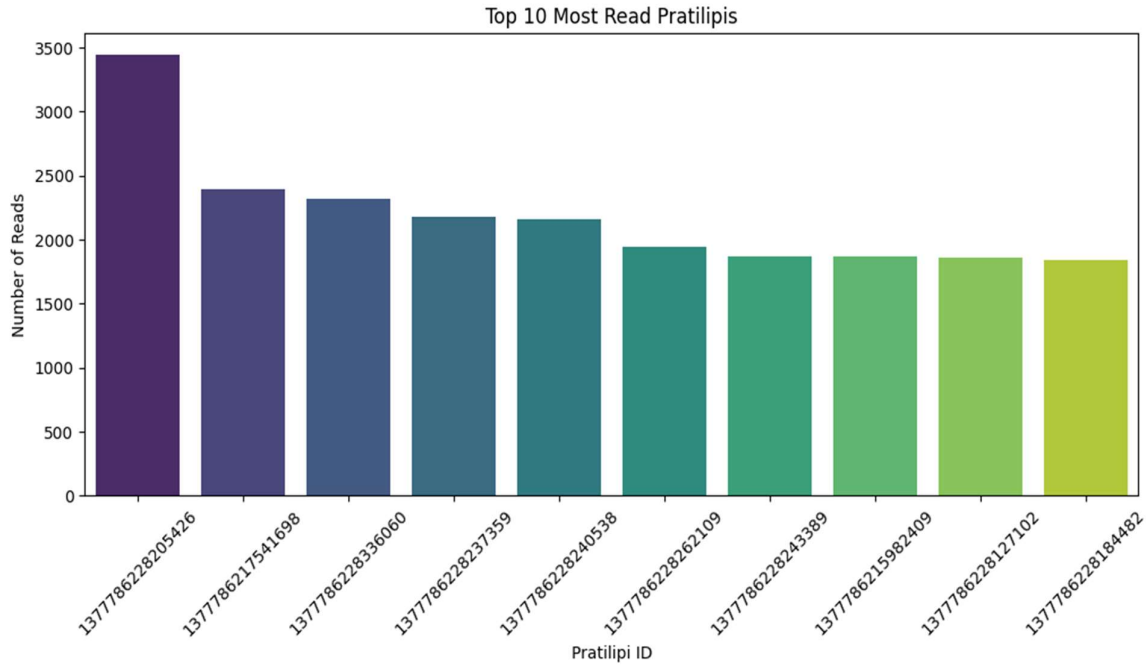
5) Evaluation metrics and visualizations will be displayed automatically.

## Visualizing Chart:

### 1) Distribution of Read Percent:



## 2) Top 10 Most Popular Stories




## 3) Category-wise Story Distribution