Documentation: Pratilipi Recommendation System

Overview

The Pratilipi Recommendation System is designed to predict the stories a user is likely to read in the future. The project leverages historical reading behavior and metadata to provide personalized recommendations. This system is structured into three main phases:

1. **Data Analysis**
2. **Train-Test Split and Model Selection**
3. **Building Recommendations**

Phase 1: Data Analysis

Objectives

1. Understand the structure and content of the datasets.
2. Validate the presence of necessary columns.
3. Merge datasets to create a unified dataset for further processing.

Steps

1. **Loading Datasets**:
   - The user_interaction.csv file contains user interaction data, including the percentage of a story read (read_percent).
   - The metadata.csv file contains metadata such as story categories (category_name) and reading time.
   - Both datasets are loaded using Pandas.
2. **Column Validation**:
   - Ensures that required columns are present in both datasets.
   - user_interaction.csv must include:
     - user_id: Unique identifier for users.
     - pratilipi_id: Unique identifier for stories.
     - read_percent: Percentage of a story read by the user.
     - updated_at: Timestamp of the interaction.

- metadata.csv must include:
  - author_id: Unique identifier for the author.
  - pratilipi_id: Unique identifier for stories.
  - category_name: Genre or type of the story.
  - reading_time: Estimated reading time.
  - updated_at and published_at: Timestamps for story updates and publishing.

3. **Merging Data**:
   - The datasets are merged on the pratilipi_id column to enrich the user interaction data with metadata, creating a comprehensive dataset for analysis.
4. **Insights from Data**:
   - Initial exploration includes checking the distribution of categories and user interactions to identify patterns that can inform model building.

Phase 2: Train-Test Split and Model Selection

Objectives

1. Prepare data for machine learning.
2. Split data into training and testing sets.
3. Select and train an appropriate machine learning model.

Steps

1. **Preprocessing**:
   - User IDs and story IDs are encoded as integers to serve as features for machine learning.
   - The target variable is read_percent, representing the percentage of a story read by a user.
2. **Train-Test Split**:
   - The dataset is split into 75% training data and 25% testing data using train_test_split from scikit-learn.

3. **Chosen Model: Linear Regression**:
    - **Why Linear Regression?**
        - Computationally efficient and interpretable.
        - Suitable for predicting continuous target variables such as read_percent.
        - Requires minimal computational resources compared to models like Random Forest or Neural Networks.
        - Avoids overfitting for linear relationships between features and the target variable.
    - **Training Process**:
        - Linear Regression is trained on the user_id and pratilipi_id features to predict the read_percent.
4. **Model Evaluation**:
    - The model is evaluated using Root Mean Square Error (RMSE) to measure prediction accuracy.
    - Example: An RMSE of ~21 indicates the average prediction error in percentage points.

Phase 3: Building Recommendations

Objectives

1. Provide personalized recommendations for existing users.
2. Handle first-time user scenarios with generic recommendations.

Steps

1. **Recommendations for Existing Users**:
    - For each user in the test set:
        - Predict read_percent for unseen stories.
        - Rank the stories by predicted read_percent.
        - Retrieve the top N stories along with their categories.

- Example Output:
- Top 5 Recommendations for Users with Categories:
- User 148640: [(235324, 'Romance'), (234110, 'Comedy'), (229172, 'Drama')]

User 20029: [(12107, 'Thriller'), (9556, 'Mystery')]

2. **Recommendations for First-Time Users**:
   - Users are prompted to enter their preferred category (e.g., Romance, Comedy).
   - If no stories match the selected category, random recommendations are provided.
   - Example Output:
   - First-time User Recommendation
   - Enter your preferred category (e.g., Romance, Comedy, etc.): Romance
   - Recommended Stories:
   - Story ID: 1377786216968011, Category: Romance

Conclusion

This system efficiently combines user interaction data and metadata to deliver personalized story recommendations. The use of Linear Regression ensures a balance between performance and interpretability, making the system suitable for moderately sized datasets. The addition of first-time user handling enhances the system's usability, making it versatile and user-friendly.