Movie Recommender System Report

Introduction
This report outlines the development of a Movie Recommender System using the MovieLens 100K dataset, employing collaborative filtering within the RecTools framework. RecTools simplifies and structures the building of recommendation systems, offering tools for data processing, metrics calculation, and various model implementations. We aim to demonstrate the efficiency of collaborative filtering in predicting user preferences and the practicality of RecTools in streamlining the development process. This project covers data analysis, model implementation, and evaluation, highlighting our system's capability to enhance user experience in digital media.

Data Analysis

Data Exploration

- **User Demographics**: Analysis of user data highlights a diverse demographic spread, suggesting varied preferences and behaviors among users.
- **Movie Popularity**: The ratings data reveals that certain movies are significantly more popular, receiving a higher number of ratings, which could be pivotal in understanding user preferences.
- **Ratings Distribution**: There is a noticeable disparity in the number of ratings per movie and per user. Some movies and users have a disproportionately high number of ratings, indicating potential biases in user engagement or movie accessibility.
- **Genre Popularity**: There is a noticeable disparity in the popularity of genres. The most popular one is Drama, and Fantasy is the least one.

Data Preprocessing

The preprocessing of the movie ratings dataset involves a series of systematic steps to prepare the data for analysis and modeling. Key aspects of the process include:

File Pattern Creation
- **Purpose**: Generate a structured list of filenames for user groups (``u1`` to ``u5``, ``ua``, ``ub``) and data types (``base``, ``test``).
- **Outcome**: Efficient access to various components of the dataset.

User Data Handling
- **Loading**: Read user data from a CSV file, discarding the `zip_code` column.
- **Normalization**: Scale the `age` column for appropriate data representation.
- **Preparation**: Format user data to align with the `RecTools` framework.

Genre Data Processing
- **Reading**: Load genre information from a CSV file.
- **Extraction**: Create a list of unique genres for movie categorization.

Movie Information Processing
- **Loading**: Read movie details, omitting non-essential information.
- **Formatting**: Retain only movie IDs and genre data for analysis.

Interaction Data Processing
- **Loading and Conversion**: Import user interactions (ratings), convert timestamps to a readable format, and set interaction weights as floats.
- **File Handling**: Determine the saving path (benchmark or interim data) based on file patterns.

Data Saving
- **CSV Output**: Save processed interactions, user features, and item features as CSV files for each data file pattern.
- **Dataset Structuring**: Organize data for seamless integration with the `RecTools` dataset construction.

This structured preprocessing ensures the raw data is transformed into a format suitable for further analytical and modeling tasks.

Model Implementation

Data Preparation
The process starts with data preparation. I use structured CSV files for user-item interactions, user features, and item features. The data is segmented into distinct user groups and divided into training ('base') and testing ('test') sets, facilitating a thorough evaluation of the model's performance.

Models reviewed
I selected a variety of recommendation models for evaluation:

- **PureSVDModel**: This matrix factorization-based model is adept at uncovering latent factors from user-item interactions.
- **PopularModel**: Recommends items based on their popularity, providing a benchmark for comparison.
- **RandomModel**: Offers baseline performance by randomly recommending items.
- **LightFMWrapperModel**: Utilizes the LightFM library, which combines collaborative filtering and content-based methods, allowing it to incorporate both user and item features into the recommendation process.

LightFMWrapperModel Factor Search
A specific focus is given to the LightFMWrapperModel. I conduct a

detailed search for the optimal number of latent factors, testing the model across various settings and evaluating it using the same set of metrics. The best parameter $K=256$ was found. However, this detailed search doesn't help because PureSVDModel performs better in all cases.

K	NDCG	Recall	MAP	RMSE
10	0.221648	0.078979	0.043760	42.636597
20	0.231141	0.083301	0.046841	41.882701
64	0.259504	0.105807	0.058514	40.757604
128	0.284097	0.128887	0.070554	40.064808
192	0.277933	0.124825	0.069523	39.892247
256	0.272259	0.129972	0.070233	39.128541
320	0.280287	0.133455	0.072139	39.415287
384	0.274126	0.130782	0.070095	39.718312

Model selection
As a result of comparison of different metric for beforementioned models, PureSVD outperforms the other models across all metrics (MAP, Recall, NDCG, RMSE), indicating it is the most effective in both ranking relevant items and predicting ratings accurately.

Model	MAP	Recall	NDCG	RMSE
Popular	0.054351	0.113488	0.236856	356.213094
PureSVD	0.132432	0.210558	0.415729	2.188762
Random	0.001843	0.006507	0.017979	3.672863
ImplicitKNN	0.070233	0.129972	0.272259	39.128541

PureSVDModel advantages and disadvantages

Advantages

1. **Latent Factor Extraction**: Effective at uncovering underlying patterns in user-item interactions.
2. **Dimensionality Reduction**: Helps mitigate issues of sparsity and overfitting.
3. **Scalability**: Suitable for large-scale datasets.
4. **Interpretability**: Provides insights into influencing dimensions or features.
5. **Versatility**: Applicable to a wide range of recommendation system problems.

Disadvantages

1. **Cold Start Problem**: Struggles with new users or items with no interaction history.
2. **Dense Matrix Requirement**: Needs significant preprocessing for sparse datasets.
3. **Linearity**: May not capture complex nonlinear relationships as effectively as other models.
4. **Computational Intensity**: The computation of SVD can be resource-intensive.
5. **Limited Personalization**: Might not offer highly personalized

recommendations compared to other models.
6. **Sensitivity to Outliers**: Outliers in the data can affect the quality of the recommendations.

Model Training

The process involved 5-fold cross-validation, where the dataset was divided into five parts. Each part was used once as a test set while the rest served as the training set. This method ensures that every data point contributes to both training and testing, enhancing the models' reliability.

The models implemented include the PopularModel, PureSVDModel, RandomModel, and ImplicitKNNModel, each designed with distinct recommendation strategies. Additionally, we conducted experiments with the LightFM model, testing different numbers of factors to find the optimal setup.

Throughout this process, we adhered to the guidelines provided in the rectools library documentation, ensuring a standardized approach to fitting the data to each model. This section lays the groundwork for the subsequent evaluation of the models' performance.

Evaluation

We employed a range of metrics to assess various aspects of the model's performance. This comprehensive evaluation was conducted on two distinct user groups, 'ua' and 'ub', to ensure a robust analysis.

Metrics Used
We utilized the following metrics for our evaluation:

1. **MAP (Mean Average Precision): Measures the precision of recommendations, indicating how relevant the recommendations are.**
2. **Recall: Assesses the model's ability to retrieve relevant items.**
3. **NDCG (Normalized Discounted Cumulative Gain): Evaluates the ranking quality of the recommendations.**
4. **Serendipity: Measures the unexpectedness of the recommendations, a factor that can enhance user satisfaction.**
5. **MIUF (Mean Inverse User Frequency): Assesses the model's tendency to recommend less popular items, promoting diversity.**
6. **RMSE (Root Mean Square Error): Indicates the accuracy of the predicted ratings.**

Methodology
The evaluation was conducted using a pre-trained model (best_model), loaded from a saved state. For each user group, the model was fitted with the base dataset and then used to generate recommendations for the test dataset users. The generated recommendations were then compared against actual user interactions in the test set.

Results
The aggregated results across both user groups for PureSVDModel:

Metric	Value
MAP	0.145749
Recall	0.237222
NDCG	0.279178
Serendipity	0.005995
MIUF	1.976700
RMSE	2.354321