Project Report

Dataset analysis and preprocessing steps.

The dataset consists of multiple CSV files containing movie metadata, ratings, keywords, and links.

The preprocessing steps included:

1. Loading the datasets using pandas.

2. Checking the shapes of the datasets to understand their structure.

3. Identifying and filling missing data.

4. Converting relevant columns to appropriate data types.

5. Extracting useful features.

6. Normalizing numerical columns.

7. Processing JSON columns to extract meaningful insights.

Explanation of the algorithms explored.

1.Popularity-Based Filtering:

Calculates mean rating for each movie, Then calculates number of ratings for each movie. Converts 'id' column in movies_metadata to integer. Merges mean ratings and rating counts with movies metadata. Sorts movies by rating count and mean rating and finally returns top N popular movies.

2.Content-Based Filtering:

Merges keywords and credits with movies_metadata. Combines relevant metadata into a single string. Creates a TF-IDF Vectorizer. Computes the cosine similarity matrix. Gets the index of the movie that matches the title. Gets the pairwise similarity scores of all movies with that movie. Sorts the movies based on the similarity scores. Gets the scores of the 10 most similar movies.

3.Collaborative Filtering:

User-based Collaborative Filtering :Creates a user-item matrix. Converts the user-item matrix to a sparse matrix. Computes the cosine similarity matrix. Gets the similarity scores for the given user. Gets the indices of the top n most similar users. Gets the ratings of the similar users. Sorts the ratings in descending order. Maps movieId to movie title. Replaces NaN titles with 'Unknown'.

Item-based Collaborative Filtering: Computes the cosine similarity matrix. Gets the similarity scores for the given movie. Gets the indices of the top N most similar movies. Sorts the ratings in descending order. Maps movieId to movie title. Replaces NaN titles with 'Unknown'.

4.Hybrid Systems:

Gets content-based recommendations. Gets user-based collaborative filtering recommendations. Combines the recommendations.

5.Advanced Techniques (Optional):

…

Justification for Chosen Algorithm

We chose to implement a hybrid recommendation system combining content-based and user-based filtering. This approach leverages the strengths of both methods to provide more accurate recommendations. Content-based filtering helps in recommending similar movies based on metadata, while user-based captures user preferences based on ratings.

Evaluate Your Recommender System

- Use appropriate metrics to evaluate the performance of your recommender system. -
Compare the performance of different algorithms if implementing more than one

Output:

```
  Precision at K=3: 0.8

  Recall at K=3: 0.4
```

Insights and Recommendations for Improving the System

1. Data Enrichment: Incorporate additional metadata such as user demographics, movie reviews ,and social media interactions to enhance the recommendation quality.
2. Algorithm Optimization: Experiment with different hyperparameters and advanced techniques like neural collaborative filtering to improve accuracy.
3. Real-Time Recommendations: Implement real-time recommendation updates based on user interactions to provide more dynamic and personalized suggestions.
4. User Feedback Loop: Collect user feedback on recommendations to continuously refine and improve the recommendation algorithms.