Amirali Najafizadeh
Parmida Azizi
Abel Mesfin
Date: Nov 26, 2023

Movie Recommendation System Report

Instructions on running the program (also available in README.txt):

The UI:

1. Download XAMPP
   > https://www.apachefriends.org/
2. Run mySQL on XAMPP
3. Click on "Admin"
4. Place the movie_rec folder in:
   > InstallationDirectoryOfXampp\htdocs
5. Run the following URL in your browser:
   > http://localhost/movie_rec/setup.php
6. Now you can see the website and use it at:
   > http://localhost/movie_rec

The Comparison:

1. Run “comparison.py”

Introduction

The Movie Recommendation System provides users with a dynamic and user-centric platform to rate movies and receive personalized recommendations. This report outlines the system's functionalities, user interactions, and the recommendation algorithm employed.

Functionality Overview

Rate the Movies Page (home.php)

New User Experience:

- New users are presented with a main page displaying all 250 movies.
- Users can rate movies on a scale of 1 to 5 by simply clicking on their desired rating.
- Ratings are updated in real-time in the database.

Existing User Experience:

- Existing users only see movies they haven't rated, ensuring a streamlined rating process.

Login as User Page (login.html)

- Users input an integer as their userID.
- Thirty pre-existing users with random movie ratings are available.
- New users can enter their own userID for a personalized experience.

See Recommendations Page (recs.php)

Activation of Recommendations:

- The "See Recommendations" button becomes active for users with at least 10 ratings.
- Clicking the button triggers the execution of user_based.py, the user-based recommender system.

Recommendation Generation:

- The top 20 recommendations based on user ratings are added to the database.
- The recommendations page displays the results from the database.

User Interaction

New User Button

- Allows users to switch to another userID.
- No strict authentication is implemented for ease of testing, enabling users to switch identities freely.

System Growth and Adaptation

- The recommendation system evolves with each user's activity.
- As new users rate movies, the system accumulates data, enhancing the accuracy of recommendations.
- Recommendations remain constant unless users provide additional ratings.

Recommendation Algorithm

Implemented Algorithms

User-Based Collaborative Filtering (user_based.py):

- **Building the System:**
    - Ratings are retrieved from the database using the read_ratings function, ensuring up-to-date data.

    - Cosine similarity is calculated using the cosine_similarity function, determining the similarity between the target user and other users.
- **Collaborative Filtering Process:**
    - **user_based_collaborative_filtering Function:**
        - This function is the heart of the recommendation process.
        - Predictions and ratings for unrated movies for the target user are generated based on the user's similarity to others.
        - The algorithm takes into account the ratings given by similar users, making the recommendations personalized.
    - **Dynamic System Growth:**
        - The system dynamically evolves with each user's activity, as the user_based_collaborative_filtering function adapts to new ratings.
        - Recommendations remain flexible, constantly improving with the influx of user data.

Matrix Factorization Collaborative Filtering (comparison.py):

Matrix Factorization Process:

- The matrix_factorization_collaborative_filtering function utilizes the scikit-learn and NumPy libraries to predict ratings for unrated movies.
- These predicted ratings are treated as if they were the actual ratings for comparison.

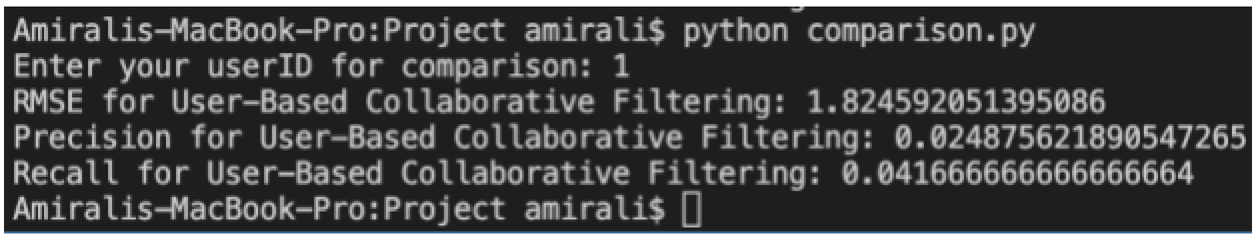
Comparison and Evaluation:

- In the evaluation phase, the ratings from Matrix Factorization CF are compared against the ratings generated by the User-Based CF system.
- By running comparison.py and entering a target user, the system calculates RMSE and recall/precision metrics.
- This comprehensive approach provides insights into how well the User-Based CF system performs against an alternative matrix factorization approach.

Comparison Metrics

RCME, Recall, and Precision:

- Metrics used to compare the performance of User-Based CF and Matrix Factorization CF.
- Evaluation conducted on a CSV dataset with userID, movieID, and rating columns.
- In order for the program to function properly, the dataset needs to be of the same format.
- A snapshot of a sample run is attached below.



```
Amiralis-MacBook-Pro:Project amirali$ python comparison.py
Enter your userID for comparison: 1
RMSE for User-Based Collaborative Filtering: 1.824592051395086
Precision for User-Based Collaborative Filtering: 0.024875621890547265
Recall for User-Based Collaborative Filtering: 0.041666666666666664
Amiralis-MacBook-Pro:Project amirali$ 
```

Sample run on comparison.py with target userID=1