Recommendation System Report

Submitted By: Aman Tiwari, Bhagyesh Gupta, Shailja Patil

1. Data Collection and Preprocessing

Data Scraping Process

To construct a movie-actor dataset, we used The Movie Database (TMDb) API. The scraping process involved:

- **Step 1: Searching for movies by title in the dataset (movies_set1.csv).**
- **Step 2: Extracting movie IDs for matching titles from the API.**
- **Step 3: Using movie IDs to fetch the top 5 cast members for each movie.**

Challenges Faced:

- **Rate-Limiting: The API limited requests per second, which we mitigated by introducing delays using the time library.**
- **Data Quality Issues: Some movies lacked detailed cast data, resulting in missing actor associations.**
- **Parallel Requests: To speed up the process, a ThreadPoolExecutor was used for concurrent API calls.**

Output: Movie-Actor Dataset

0	**120510**	**Value for Money (1955)**	**Comedy|Romance**
1	**212955**	**Face of Evil (1996)**	**Drama|Thriller**
2	**193912**	**Spring 1941 (2007)**	**Drama|Romance|War**
3	**163921**	**Wolf Creek (2016)**	**Crime|Horror|Thriller**
4	**126652**	**Raven the Little Rascal (2012)**	**Animation|Children**

---

2. User-Actor Rating Matrix Construction

Process

- **Loading Datasets: Combined ratings_set1.csv and movie_actors.csv to link user ratings to actors.**
- **Exploding Actor Lists: The actor list for each movie was expanded, mapping each actor to the corresponding user ratings.**
- **Calculating Ratings: Averaged user ratings for actors across the movies they starred in.**

Key Preprocessing Steps:

- **Filtered out actors with fewer than 5 ratings and users with fewer than 5 interactions to improve data quality.**

Output:

```
    userId           actor_name               average_rating
0       5        Aaron Izbicki                   4.0
1       5        Aaron Michael Lacey             4.0
2       5        Afemo Omilami                   4.0
3       5        Afram Bill Williams             4.0
4       5        Al Harrington                   4.0
```

---

3. Algorithm Selection

Selected Algorithm

We implemented Singular Value Decomposition (SVD) for collaborative filtering. The reasons for choosing SVD include:

- **Effectiveness: SVD is highly effective in reducing data sparsity and uncovering latent user-actor preferences.**
- **Efficiency: Computationally feasible for our dataset size.**
- **Popularity: Widely used for recommendation systems.**

Modifications for Optimization

- **Cross-Validation: Performed 3-fold cross-validation to fine-tune parameters and ensure generalization.**
  Cross-Validation Results (RMSE, MAE):

  {'test_rmse': array([0.4677603 , 0.46785738, 0.4674208]),

  'test_mae': array([0.29845942, 0.29874973, 0.29845284]),

  'fit_time': (255.3905746936798, 268.6272623538971, 262.6424765586853),

  'test_time': (112.79141664505005, 86.03103733062744, 75.63141942024231)}

- **Filtering Sparse Data: Removed low-activity users and actors to improve model robustness.**

```
Evaluating RMSE, MAE of algorithm SVD on 3 split(s).

                  Fold 1  Fold 2  Fold 3  Mean    Std
RMSE (testset)    0.4678  0.4679  0.4674  0.4677  0.0002
MAE (testset)     0.2985  0.2987  0.2985  0.2986  0.0001
Fit time          255.39  268.63  262.64  262.22  5.41
Test time         112.79  86.03   75.63   91.48   15.65
```

---

4. Evaluation and Analysis

Metrics Used

- **Root Mean Squared Error (RMSE): Measures the accuracy of predicted ratings.**
- **Mean Absolute Error (MAE): Provides the average magnitude of errors in predictions.**
- **Precision@k and Recall@k: Evaluate how well the system retrieves relevant actors for a user.**

Performance Results:

- **Cross-Validation Results:**
    - **RMSE: 0.4677603 , 0.46785738, 0.4674208**
    - **MAE: 0.29845942, 0.29874973, 0.29845284**

Precision@5, 0.9000

Recall@5, 0.3214

NDCG@5: 0.9306

Analysis

- **Strengths:**
    - **Accurate predictions for frequent users.**
    - **Effectively identifies top-rated actors.**
- **Limitations:**
    - **Struggles with cold-start problems (e.g., new actors or users).**

---

5. Recommendations and Visualizations

Sample Recommendations

Using the trained SVD model, we generated actor recommendations for user ID 1:

Top 10 recommended actors for user 1:

Actor ID: Clémentine Pons, Predicted Rating: 4.41

Actor ID: Andy Gillet, Predicted Rating: 4.38

Actor ID: Victoria Guerra, Predicted Rating: 4.35

Actor ID: Ricardo Pereira, Predicted Rating: 4.35

Actor ID: Jonathan Genet, Predicted Rating: 4.34

Actor ID: Sabine Azéma, Predicted Rating: 4.34

Actor ID: António Simão, Predicted Rating: 4.32

Actor ID: Benny Hill, Predicted Rating: 4.29

Actor ID: Claire Slemmer, Predicted Rating: 4.28

Actor ID: Chuck Brauchler, Predicted Rating: 4.28 Visualization

---

Conclusion

This report showcases a robust recommendation system using collaborative filtering. By leveraging TMDb data and SVD, we successfully predicted user preferences for actors. While the system performs well with frequent users and popular actors, further enhancements (e.g., hybrid models) could address cold-start challenges and improve accuracy for niche actors.