Movie & Content Recommendations System

Varinder Singh (B20141)
Shivam Kumar (B20134)
Param Meena (B20118)
Nikhil Dhumale (B20116)

Mentor: Mrs. Arti Kashyap

Indian Institute of Technology, Mandi

Code: https://github.com/varinder269/Movie-Recommendation-Project

***Abstract*— The recommendation system we made using Spark a movie lens dataset of ~270 MB. The ALS collaborative recommendation engine by using Spark MLlib and give recommendations with user-based movie recommendations with a scalable matrix factorization technique.**

I. INTRODUCTION

The recommendation system is a widely used machine learning technique that has many applications in E-commerce (Amazon, Alibaba), video streaming (Netflix, Disney+), social networks (Facebook, Linkedin), and many other areas. Because of the large amount of data in those services, nowadays most industry-level recommendation systems are built in big data frameworks like Spark. Here we built a movie recommendation using spark.

II. PROCEDURE

A. 1st STEP

First load four of datasets, namely movie, rating, links, and tags, and conduct several data explorations on these data to get some basic information, such as the number of users, number of movies, number of rating perusers and per movies respectively, and distribution of movies on different genres.

B. 2nd STEP

After doing data preprocessing, we build an ALS model based on the rating data to predict the ratings, which is treated as the degree of preference for movies among different users. The parameters (master, rank, regParam) are tuned by grid search strategy via 5-fold cross-validation to obtain the model with the smallest RMSE on the validation set. This is the best prediction model.

C. 3rd Step

By the best model obtained from the above step, making predictions of ratings on movies in the test set and calculating the RMSE to evaluate the model performance are preparing for the next step.

D. 4th Step

In this step, I use the prediction results by the best model to recommend 5 movies for userID 575 and 232 respectively; and we also find 5 movies that are the most similar to a movie with movieID 471 and 463 by the approximate nearest neighbor search algorithm on the movie feature vector.

III. ALGORITHM

There are two types of algorithms to do the task discussed below.

A. Content-based Recommendation:

1. **Description: Utilizes product characteristics to recommend similar products to what a user previously liked.**
2. **Assumption: If person P1 and person P2 have the same opinion on product D1, then P1 is more likely to have the same opinion on product D2 with P2 than with a randomly chosen person Px.**
3. **Example: News/article recommendation**
4. **Advantages:**
   - **The model doesn't need any user data input, so easier to scale.**
   - **Capable of catching niche items with feature engineering.**
5. **Disadvantages:**
   - **Requires domain knowledge.**
   - **Limited ability to expand user's interests**

B. Collaborative filtering:

1. Description: Predicts the interest of a user by collecting preference information from many other users.
2. Assumption: If person P likes product D1 which has a collection of attributes, he/she is more likely to like product D2 which shares those attributes than product D3 which doesn't.
3. Example: movie recommend-ation, Amazon product recommendations.
4. Advantages:
   - No domain knowledge needed, highly transferrable model.
   - Capable of helping users discover new interests.
5. Disadvantages:
   - Cold-start problem: need to work with existing data, can't handle fresh users
   - Difficulty in expanding features for items.

C. Collaborative Filtering and spark ALS

we will use collaborative filtering as the recommendation algorithm. How collaborative filtering works is this: First, we consider the ratings of all users to all items as a matrix, and this matrix can be factorized into two separate matrices, one being a user matrix where rows represent users and columns are latent factors; the other being an item matrix where rows are latent factors and columns represent items. During this factorization process, the missing values in the rating matrix can be filled, which serve as predictions of user ratings of items, and then we can use them to give recommendations to users.

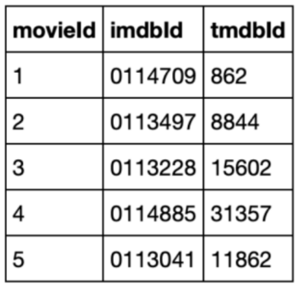
D. ALS (alternating Least Squares)

ALS is a mathematically optimized implementation of collaborative filtering that uses Alternating Least Squares (ALS) with Weighted-Lambda-Regularization (ALS-WR) to find optimal factor weights that minimize the least squares between predicted and actual ratings. Spark's MLlib's package has a built-in ALS function, and we will use it in this post.

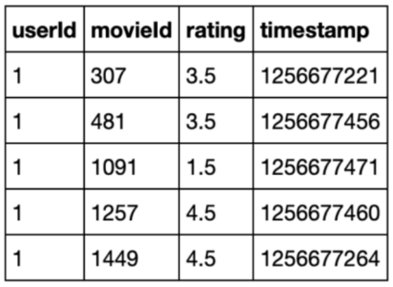
IV. DATASET

we have used the MovieLens Dataset. Which has approx. size of 265 mb.

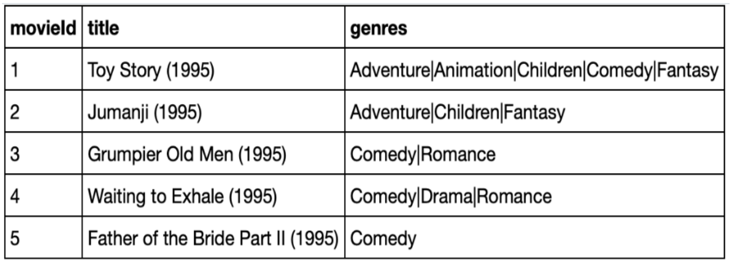
V. EXAMPLE



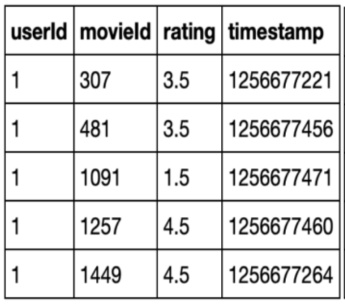
movieId	imdbId	tmdbId
1	0114709	862
2	0113497	8844
3	0113228	15602
4	0114885	31357
5	0113041	11862



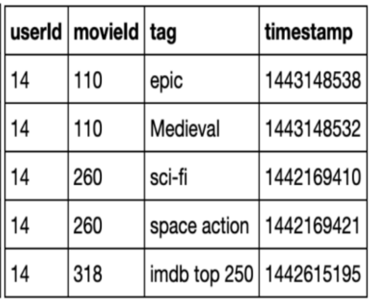
userId	movieId	rating	timestamp
1	307	3.5	1256677221
1	481	3.5	1256677456
1	1091	1.5	1256677471
1	1257	4.5	1256677460
1	1449	4.5	1256677264



movieId	title	genres
1	Toy Story (1995)	Adventure|Animation|Children|Comedy|Fantasy
2	Jumanji (1995)	Adventure|Children|Fantasy
3	Grumpier Old Men (1995)	Comedy|Romance
4	Waiting to Exhale (1995)	Comedy|Drama|Romance
5	Father of the Bride Part II (1995)	Comedy

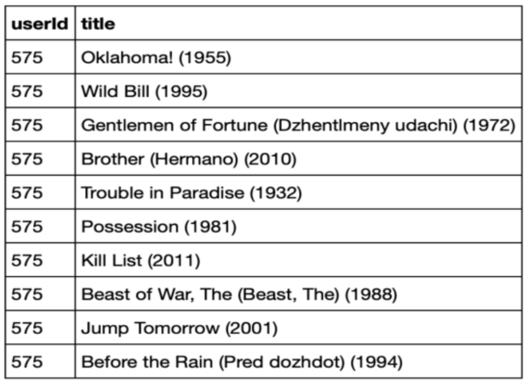


userId	movieId	rating	timestamp
1	307	3.5	1256677221
1	481	3.5	1256677456
1	1091	1.5	1256677471
1	1257	4.5	1256677460
1	1449	4.5	1256677264



userId	movieId	tag	timestamp
14	110	epic	1443148538
14	110	Medieval	1443148532
14	260	sci-fi	1442169410
14	260	space action	1442169421
14	318	imdb top 250	1442615195

For example, we need to find the recommendations for the userID = 575



userId	title
575	Oklahoma! (1955)
575	Wild Bill (1995)
575	Gentlemen of Fortune (Dzhentlmeny udachi) (1972)
575	Brother (Hermano) (2010)
575	Trouble in Paradise (1932)
575	Possession (1981)
575	Kill List (2011)
575	Beast of War, The (Beast, The) (1988)
575	Jump Tomorrow (2001)
575	Before the Rain (Pred dozhdot) (1994)

VII. ERROR Analysis

The error method to check the accuracy we used here is RMSE root mean square error method. To compute, calculate the residual for each data point, compute the norm of residual for each data point, compute the mean of residuals and take the square root of that mean.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} \|y(i) - \hat{y}(i)\|^2}{N}},$$

The error we got here is 1.41. which is not very much good. There are very good chances to improve the model and reduce the error. There are factors which effect the error directly or indirectly. Some factors are inversely proportional to the error an some are directly proportional to error. The RMSE of the best ALS model on the test data which is 1.41, indicating the model is with average performance in predicting the ratings for movies; ALS model can provide both recommendations of movies based on user's preferences and also similar movies to a specific movie, which shows its effectiveness as one of most critical techniques in recommendation system. More works can be considered to further improve the model performance, such as making use of information from other data sets such as genres of movies and tag information, building ALS model incorporating both explicit and implicit feedbacks, and try some other techniques such as KNN, Deep Learning, applying ensemble based on several methods, and so on.

VI. CONCLUSIONS

ALS model can provide both recommendations of movies based on user's preferences and similar movies to a specific movie, which shows its effectiveness as one of the most critical techniques in the recommendation system. More works can be considered to further improve the model performance, such as making use of information from other data sets such as genres of movies and tag information, and building the ALS model incorporating both explicit and implicit feedback. The map-reduce method on the other hand is faster and can be implemented on websites and servers for mass use and are hence scalable.

VIII. ACKNOWLEDGMENT

Professor Arti Kashyap and Teaching Assistant, Puneet Kumar were guiding us in this project. I want to thank them for their cooperation and investment in us.

REFERENCES

[1] M. Ahmed, M. T. Imtiaz, and R. Khan, "Movie recommendation system using clustering and pattern recognition network," 2018 IEEE 8th Annual Computing and Communication Workshop and Conference (CCWC), 2018, pp. 143-147, DOI: 10.1109/CCWC.2018.8301695.

[2] J. Zhang, Y. Wang, Z. Yuan, and Q. Jin, "Personalized real-time movie recommendation system: Practical prototype and evaluation," in Tsinghua Science and Technology, vol. 25, no. 2, pp. 180-191, April 2020, DOI: 10.26599/TST.2018.9010118.

[3] S. Halder, A. M. J. Sarkar, and Y. -K. Lee, "Movie Recommendation System Based on Movie Swarm," 2012 Second International Conference on Cloud and Green Computing, 2012, pp. 804-809, DOI: 10.1109/CGC.2012.121.

[4] Bhalse, N., & Thakur, R. (2021). Algorithm for movie recommendation system using collaborative filtering.