March (2023), **xx**:xx 1–7

scientific article

Comparison of the four algorithms for a recommender system.

chaimaa Bouabd[1]

[1] FSTG, Marrakech, 40000, Marrakech Safi, morocco. E-mail: chaimabouabd2000@gmail.com.

Keywords: , Collaborative Filtring, Matrix Factorization, Recommendation System, Model-based Factorization, Hadoop, Pyspark

Abstract
The aim of this article is to create a machine learning algorithm using MovieLens data to optimize the predictive power of the model for movie recommendations. Movie recommendation systems suggest movies to users based on their past preferences and similar movie ratings.

Contents

1. Introduction

Nowadays, with the vast amount of digital content available, it becomes increasingly challenging to find something that matches our interests. Movie recommendation systems come to the rescue by suggesting movies based on past user preferences and similar movie ratings. The project aims to develop a machine learning algorithm using MovieLens data to recommend films while optimizing the predictive power of the model. The primary goal is to search for content that could interest the user, and this system is widely used by companies to advertise movies, books, music, restaurants, and other applications. Specifically, movie recommendation systems predict the likelihood that users who watched a particular film may be interested in watching another movie based on their choices and the choices of other users.

The data used in these systems are the ratings left by users to evaluate their taste for a specific film. The importance of movie recommendation systems can be highlighted by large companies such as Netflix, which heavily rely on these systems to enhance user experience and satisfaction.

The remainder of this review is organized as follows. In Section 2 we present an overview of RS and review the frequently used CF approaches, techniques, evaluation metrics, and technical challenges on existing methods. Next, in Section 3 We will introduce the techniques and delve deeper into some proposed algorithms.in Section 4.we will discuss the results of each Model . Finally, we outline conclusions.

2. Related Works

A recommendation system is a type of information filtering system that predicts and suggests items of interest to users. These systems are widely used in various industries, including e-commerce, social media, and entertainment, to provide personalized recommendations to users based on their preferences and behavior. There are three main approaches to building recommendation systems: content-based filtering, collaborative filtering, and hybrid approaches that combine both content-based and collaborative filtering techniques.

2.1. The different approaches

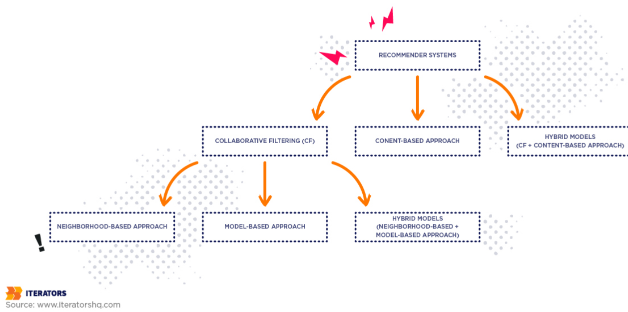

Figure 1. *Recommendation Systems Overview.*

2.1.1. Content-Based Filtering

Content-based filtering recommends items based on the similarity of the content of the items and the user's preferences. The system first creates a profile of the user's interests based on their interactions with the system. Then, it recommends items that have similar characteristics to the items the user has previously interacted with.

Advantages

- Works well for recommending niche or unique items that are not popular among other users.
- Doesn't require large amounts of user data to work effectively.
- Provides an explanation for why the recommendation was made, as it is based on the content of the items.

Disadvantages

- Limited to recommending items with similar content to the user's preferences.
- Can be difficult to identify relevant features of the content to use in the recommendation process.

- Unable to recommend items that the user has not yet interacted with.

2.1.2. Collaborative Filtering

Collaborative filtering recommends items based on the behavior and preferences of similar users. The system first creates a user-item matrix that records the interactions between users and items. Then, it identifies similar users based on their interactions with items and recommends items that those similar users have interacted with, but the user has not. "This approach helps us to understand the intention of the user or client from the first interaction with the system, without needing to know too much about the user beforehand."

Advantages

- Can recommend items that the user has not yet interacted with.
- Doesn't require knowledge of item content, only user-item interactions.
- Can be used to identify groups of users with similar preferences.

Disadvantages

- Limited to recommending popular items, as those are more likely to have interactions with other users.
- Requires a large amount of user data to work effectively.
- Can be affected by the cold-start problem, where new users or items have no interaction data, making it difficult to recommend anything to them.

2.1.3. Hybrid Approaches

Hybrid approaches combine both content-based and collaborative filtering techniques to provide more accurate recommendations. These systems use the strengths of both approaches to overcome the limitations of each.

Advantages

- Can provide more accurate and diverse recommendations than either content-based or collaborative filtering alone.
- Can handle the cold-start problem more effectively by using content-based filtering for new items or users.
- Provides a more flexible and adaptable recommendation system.

Disadvantages

- Requires more computational resources to implement effectively.
- Can be more complex to design and optimize.

2.2. Schema of different approaches in RS

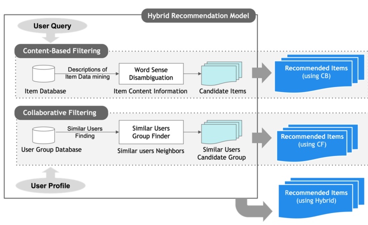

Figure 2. *RS Schema.*

2.3. Dataset

In this Artical , we will be using the movielens dataset (ml-100k). This dataset (ml-latest-small) describes 5-star rating and free-text tagging activity from MovieLens, a movie recommendation service.

Up next, we will explore and discuss more about Algorithms of models for our proposed solution in greater detail.

3. Proposed approach

In this section, we will suggest four types of models for movie recommendation systems and discuss the results of the loss function for each one.

3.1. Model 1 - The Overall Mean Model

The Overall Mean Model is a simple approach to recommend movies to users based on the average rating of all movies in the dataset.

The principle of this model is straightforward - it assumes that users will like movies that have a higher average rating from the general population. Therefore, for a new user, the model will recommend the movies with the highest average rating overall.

However, the major limitation of this model is that it does not take into account the individual preferences of users, and hence it may recommend movies that are not relevant or interesting to a particular user. Additionally, it assumes that all movies are rated equally, and does not consider factors such as genre, release date, or popularity.

Despite its simplicity, the Overall Mean Model can be used as a baseline for evaluating the performance of more complex recommendation algorithms.

The initial model can be stated as follows:

$$Y_{ui} = \mu + \epsilon_{ui}$$

3.2. Model 2: Multivariate film effect model

The Multivariate film effect model is a type of movie recommendation system that takes into account multiple factors, such as the genre of the movie, the cast and crew, and the ratings and reviews of other users, to generate personalized movie recommendations for individual users.

The basic principle of this model is to analyze the features and characteristics of each movie and to model the effect of these features on the user's preferences. For instance, if a user prefers action movies with a particular actor or director, the model will analyze these factors to recommend similar movies that match the user's preferences.

To achieve this, the model uses a multivariate statistical approach that considers several variables simultaneously, such as the user's past movie ratings, the movie's genre, the cast and crew, the release year, and the popularity of the movie.

The model then uses machine learning algorithms to predict the likelihood of a user liking a particular movie based on the variables analyzed. The model continues to learn and improve its recommendations as more data is collected and analyzed.

Overall, the Multivariate film effect model is a powerful tool for generating personalized movie recommendations that match the unique preferences of each user.

$$Y_{ui} = \mu + b_i + \epsilon_{ui}$$

3.3. Model 3: Multivariate film and user effect model

The Multivariate film and user effect model is an advanced type of movie recommendation system that considers both movie features and user preferences simultaneously to generate personalized recommendations.

The principle behind this model is to model the effect of both movie features and user preferences on the likelihood of a user liking a particular movie. For example, a user's preference for romantic comedies may be influenced by factors such as the actors, the director, the soundtrack, or the storyline.

To achieve this, the model uses a combination of collaborative filtering and content-based filtering techniques. Collaborative filtering is used to identify similar users based on their past movie ratings and preferences, while content-based filtering is used to analyze the features and characteristics of each movie and its relationship to user preferences.

The model then uses machine learning algorithms to predict the likelihood of a user liking a particular movie based on the variables analyzed. By incorporating both movie features and user preferences, the Multivariate film and user effect model is able to generate highly personalized and relevant movie recommendations.

Overall, the Multivariate film and user effect model is a powerful tool for generating personalized movie recommendations that consider both the user's past behavior and the unique features of each movie.

This implies that further improvements to our model can be:

$$Y_{ui} = \mu + b_i + b_u + \epsilon_{ui}$$

3.4. Model 4: Multivariate movie effect model and regularized user

The Multivariate movie effect model and regularized user is similar in concept to the Multivariate film and user effect model, but the main difference is that the former incorporates regularization techniques to improve the model's generalization and prevent overfitting.

The regularization technique is used to add a penalty term to the cost function of the model to discourage extreme values of the model parameters, thereby preventing the model from overfitting the training data and improving its generalization performance.

The Multivariate movie effect model and regularized user also takes into account the preferences of other users with similar tastes to the target user, through a process known as "neighborhood-based collaborative filtering." This involves finding a set of similar users to the target user based on their past preferences, and using their ratings to predict the target user's preferences.

Overall, the Multivariate movie effect model and regularized user is a powerful tool for generating highly personalized movie recommendations that consider both the unique features of each movie and the individual preferences of each user, while also avoiding overfitting and improving the generalization performance of the model.

4. Results and discussion

To implement all the theories we mentioned previously, we are going to use the famous dataset Movielens 100K. It contains 100K ratings of 1982 movies. These data were created by 943 users . This dataset was Released in 4/1998. The data are contained in the files test.csv, base.csv, ... More details about the contents and use of all these files follows.

The following table represents the final result of the implementation:

Table 1. *Comparative table of results.*

	Model 1	Model 2	Model 3	Model 4 -D
RMSE	1.15	1,03	0,98	0,?
Execution time	5,42 s	9,70 s	16,59 s	3,17 min

Interpretation
RMSE

As we can see, the RMSE value of Model 4 is lower than the values found in the other models. This suggests that the algorithm used in Model 4 is more effective compared to the other models, making it a suitable choice for accurate predictions. On the contrary, the highest RMSE value was represented by the first model, with a value of 1.15, indicating that Model 1 is not a good choice for predictions. Additionally, we found Model 2 and Model 3 with RMSE values of 1.03 and 0.9, respectively. In this case, Model 3 outperforms Model 2 and can be considered as a better option.

Execution time

On the one hand, we can observe that Model 1 requires less execution time, which can be considered efficient as it saves more time. This makes it the best choice if we want to achieve maximum results with less time and effort.

On the other hand, both Model 2 and Model 3 take more time to execute compared to Model 1. The fourth Model is the slowest in terms of time factor, regardless of its effectiveness.

5. Conclusion

In conclusion, the main objective of this project was to develop a machine learning algorithm that can provide movie recommendations by optimizing the predictive power of the model. The analysis began with an exploratory data analysis (EDA) using visualization techniques such as heatmaps and histograms. The primary evaluation metric for the models was RMSE (Root Mean Square Error), and four models were considered in the project. The first model was a simple global mean model, while the second model included the film effect. The third model added the user effect, and the fourth model included regularization for samples of small size. Overall, this project demonstrates the importance of machine learning algorithms in predicting user preferences and recommends movies tailored to their interests.

6. References

[RUI CHEN 1,2, QINGYI HUA1, YAN-SHUO CHANG3, BO WANG1,4, LEI ZHANG5, AND XIANGJIE KONG (November 19, 2018)]bib2 Survey of Collaborative Filtering-Based Recommender Systems: From Traditional Methods to Hybrid Methods Based on Social Networks

[Jeffery chiang(May 16, 2021)]bib1 https://medium.com/analytics-vidhya/model-based-recommendation-system-with-matrix-factorization-als-model-and-the-math-behind-fdce8b2ffe6d

souce code

https://github.com/chiang9/Medium$_blog/blob/main/ALS_model/movielen$