Movie Recommendation System

Table of Contents

Executive Summary

Implementing personalized recommendation systems has been viewed as a crucial differentiator in online streaming by most over-the-top (OTT) media service providers. This report demonstrates the process of developing a proof-of-concept recommendation engine for a startup online movie platform firm. The model uses individual film ratings gathered from 98 current Clemson University students and 280 anonymous users. Three candidate algorithms, *K Nearest Neighbors (KNN)*, *Matrix Factorization – Singular Value Decomposition (SVD)*, and a *memory-based approach*, have been chosen for the study. The matrix factorization model outperforms the other two approaches in the scenario at hand and is the model we chose. As further proof of concept, we also deal with the cold start item problem with historical data, and the cold start user problem using an item-based approach. The discussion is concluded by detailing how this system can benefit businesses in an entertainment and retail setting through boosted customer lifetime value and increased retention.

Introduction

The advent of over-the-top (OTT) services, such as Netflix, Hulu, and Prime Video, has given people access to nearly infinite digital media resources via the Internet at relatively low prices. However, this abundance of choices also complicates the user's process of finding the items that truly meet his or her interests. This paper discusses the development of a proof-of-concept recommendation system for a startup movie platform company. The problem at hand and the data in use will be detailed, followed by an in-depth performance assessment of the candidate algorithms and suggestions about how the recommendation engine can benefit the business in the long run.

Problem Formulation

We perform our analysis by building three recommendation models on the survey datasets both the current year's as well as historical data: 1. K Nearest Neighbors (KNN) 2. Matrix Factorization – Singular Value Decomposition (SVD) 3. Memory-based collaborative filtering that leverages all movies rated by a user. We also deal with cold start predictions for specific users, which occurs when we have to make predictions for new users with no historical data. Then, after collecting data on their habits, we investigate how the predictions change after we are able to personalize the predictions. We also conduct similar analysis for movies that had previously not been reviewed in the original survey.

We leverage the current students' rating data to predict ratings given by 4 users for 3 movies: Zero Dark Thirty, Imitation game, Dunkirk. We then predict ratings for the same 4 users for the following 3 new unrated movies: Winter's Bone, A Serious Man, and Son of Saul. After testing the respective models accuracy on our own team's movie reviews, the model chosen was matrix factorization using SVD, the details for which are contained in the model development section.

Data Characteristics

The dataset in hand consists of movie ratings for 50 movies. Ratings are taken in the form of a survey. This is composed of 98 current Clemson University students and 280 anonymous users that were collected at an earlier date. The rating ranges from 1, being the least, to 5, being the highest. If a user has not watched the movie this would be represented as a null value. Along with the existing users, 3 new users named Amy Russel, Shachi Govil, Camille Mack are included for cold start predictions.

Model Development, Estimation and Results

The model uses collaborative filtering, a technique used in many recommender systems which establishes connections between movies and users through their past ratings. This connection is then leveraged in order to make predictions about a user's review of a movie based on the reviews of other, similar users (Mehta & Rana, 2017). The way we chose to do this is through a principle called matrix factorization, which involves decomposing our original user-item matrix into separate matrices such that the product of these matrices returns the original data frame. In order to avoid biasing the results, all the movie ratings were normalized for a user's average rating. The model accuracy was assessed based on a set of 3 movies for each member in our group, the average error in our predictions was 0.32, and Xu (Sam) had the harshest rating. The full details can be found in Table 1.

As further evidence of the model's effectiveness, we predicted user ratings for three movies that were absent from the original survey. In order to make predictions, we chose to supplement the data with the historical dataset, which worked in our case because we had access to historical data. In instances when this data is not available, a simple approach would be to use a content-based approach where items are recommended based on factors like genre, language, region of origin, etc. For this, movie specific data would need to be collected and assigned, a potentially time-consuming effort. Another potential solution could be to assume each member will rate the new movie the average of all their other ratings, and that way it will show up in their recommendations so that data can be collected on the new movie. This approach is detailed in table 2 of the appendix.

Name	Son of Saul	Winter's Bone	A Serious Man
User 1	4.195768	4.213498	4.177095
User 2	3.779982	3.80015	3.75057
User 3	1.747865	1.759293	1.755276
User 4	3.926011	3.893553	3.944176

One drawback of a user-based approach is that it relies on the user's past information to make predictions, but in the case of a new user, this becomes difficult because we have no information about their habits. As a result, our strategy was to use item-based collaborative filtering which, instead of calculating predicted ratings based on similar users, bases its calculations on similar movies. Once data was collected on user behavior, we switched back to the original, user-based model because it had lower error than an item-based approach. Another, simpler version of this could be to recommend movies to users with the highest mean rating. Other potential methods of tackling this problem include calculating customer similarity based on demographic features like gender, age, and location, and recommend items to users based on what similar people engaged with.

Name	Movie	Cold Rating	Rating with Data	Change

Shachi Govil	Avatar	4.015152	4.232074	0.216922
	Wolf of Wall Street	4.116838	4.209444	0.092608
	Inception	4.490385	4.177216	0.313169
Amy Russell	Avatar	4.015152	3.171662	-0.84349
	Wolf of Wall Street	4.116838	3.2053	-0.911538
	Inception	4.490385	3.241959	-1.2485
Camille Mack	Avatar	4.015152	2.729097	-1.28606
	Wolf of Wall Street	4.116838	2.757855	-1.35898
	Inception	4.490385	2.798115	-1.69227

As we can see, item-based filtering gives a decent starting point, but it tends to overestimate a user's rating of popular movies which is why a user-based approach that takes into account ratings history is superior.

Recommendation and Managerial Implications

Recommender systems decrease the transaction costs to find and select items from hundreds or thousands of items in an online shopping environment. With increasing numbers of both users and items, it becomes extremely important to build a personalized experience for the users. We first used the memory-based approach i.e finding similar users based on cosine similarity but its disadvantage is that the performance reduces when the data is sparse (Business et al., 2021). In order to improve our recommendations, we then moved to model-based approaches such as KNN and Matrix factorization in which dimensionality reduction deals with the sparse data problem and scalability but it also comes with a disadvantage that it is difficult to interpret (Business et al., 2021). However, Based on the results, we found out that collaborative filtering using matrix factorization effectively predicts the ratings for a given user and movie.

Considering e-commerce businesses, recommender systems are proven to be very efficient in reducing the time required to find an item and as a result, boost customer engagement. The result is loyalty and increased satisfaction of users with these services, thereby reducing churn rates and increasing lifetime value of customers (Kordík, 2016). Recommendation systems are also very useful when it comes to gaining complex insights into product and customer bases. As a result of personalization, users tend to interact more with the items leading to higher consumption and hence, profits(Kordík, 2016). In driving sales, recommendation systems play a pivotal role in assorting the products. Considering information like purchase history and purchasing habits, retail markets can provide personalized product assortments for different users. Additional value functions which would refine an existing recommender system include: incorporating repeat purchases or viewings as well as Recency-Frequency-Monetary-Variety (RFMV) for a recommendation model. For product assortment in retail models, information on RFMV of customers would help in refining the recommendation systems. A study, by researchers at Carnegie Mellon University and The Wharton School found that the impact created by recommender systems is moderated by a lot of factors including product attributes or characteristics (Tepper School of Business).

Conclusion

We formulated a recommendation system using the survey data collected for movie ratings using the matrix factorization technique since it was most accurate and had the lowest error. We also made cold start predictions for three new users before and after obtaining movie ratings from the same three users. Cold start predictions in collaborative filtering are limited in their accuracy. One of the main disadvantages of collaborative filtering is that when we have a new item with no ratings, no matter how relevant the item would be to some users, it will not be recommended by the recommendation system. This is the cold start issue. In order to improve on the methods we learned in class, we can incorporate a hybrid model that combines both collaborative and content-based filtering.

References

Business, I. T. &, says:, S. P., says:, B., & *, N. (2021, July 15). *Collaborative filtering in Recommender Systems: Learn all you need to know* |. Iterators. Retrieved, from https://www.iteratorshq.com/blog/collaborative-filtering-in-recommender-systems/

Jeong, Yohan. “Item-Based Collaborative Filtering in Python.” *Medium*, Towards Data Science, 22 Apr. 2021, https://towardsdatascience.com/item-based-collaborative-filtering-in-python-91f747200fab.

Kordík, P. (2016, August 23). *The value of personalized recommendations for your business*. Medium. Retrieved, from https://medium.com/recombee-blog/the-value-of-personalized-recommendations-for-your-business-6b2e81ce0a4d

“Matrix Factorization for Movie Recommendations in Python.” *Nick Becker*, 10 Nov. 2016, https://beckernick.github.io/matrix-factorization-recommender/.

Milankovitch, M. (2017, July 5). *The cold start problem for Recommender Systems*. Medium. Retrieved, from https://medium.com/@markmilankovich/the-cold-start-problem-for-recommender-systems-89a76505a7

R. Mehta and K. Rana, "A review on matrix factorization techniques in recommender systems, *2017 2nd International Conference on Communication Systems, Computing and IT Applications (CSCITA)*, 2017, pp. 269-274, doi: 10.1109/CSCITA.2017.8066567.

University, C. M. (n.d.). *Effects of recommender systems in e-commerce vary by product attributes and review ratings - Tepper school of business - Carnegie Mellon University*. Effects of Recommender Systems in E-Commerce Vary by Product Attributes and Review Ratings - Tepper School of Business - Carnegie Mellon University. Retrieved, from https://www.cmu.edu/tepper/news/stories/2020/may/e-commerce-recommenders-research-lee.html

“What Is the K-Nearest Neighbors Algorithm?” *IBM*, IBM, https://www.ibm.com/topics/knn.

Appendix

Model Comparisons

The initial model used was item-based collaborative filtering which used cosine similarity, which is a rough estimate of how similar two users are to each other. The predicted rating was calculated as an average of all other movie ratings done by a particular user, weighted by the cosine similarity of that movie to other movies the user has rated. The equation is represented below:

$$R(m,u) = \frac{(\sum_k \quad S(m,k) * R(k,u))}{\sum_k \quad S(m,k)}$$

Where k represents the set of all movies rated by user u, S(m,k) represents the similarity between the movie m (the rating for which we are trying to predict) and all the movies similar to it (represented by k). R(k,u) represents the rating of all movies in the set k by user u, and R(m,u), which is what we are predicting, is the rating of a particular movie m by user u.

From this initial model, we switched to an algorithm called K nearest neighbor, which is a supervised machine learning algorithm that groups data based on a specified distance metric (IBM). We again used cosine similarity for our distance metric, and the same equation was used. However, the main difference is in what movies are in the set k. For our first algorithm, all the movie ratings of a user were taken into account, regardless of cosine similarity. In the case of K Nearest Neighbor, we were able to use the reviews of the 25 most similar movies to the movie whose rating we are predicting. This allows us to filter out irrelevant ratings and produce a more accurate rating prediction.

We then switched to a user-based approach, which is very similar to the item-based except it uses similarity between user similarity instead of items, but the predicted rating is still calculated in the same way.

The final iteration of our model switched directions completely and used a form of matrix factorization called Singular Value Decomposition (SVD) to decompose our large, user-item matrix into smaller matrices that can then be multiplied together to produce predictions for the missing values. This was the most accurate model, and as a result the one we chose.

Table 1

Testing our model on 3 movies for each user

Name	**Movie**	**True Rating**	**Predicted Rating**	**Error**
User 1	Zero Dark Thirty	5	4.663247	0.33673
	The Imitation Game	5	4.803538	0.19462
	Dunkirk	5	4.771874	0.228126
User 2	Zero Dark Thirty	5	4.160006	0.839994
	The Imitation Game	4	3.892395	0.107605
	Dunkirk	5	4.840539	0.159461
User 3	Zero Dark Thirty	Did not Watch	1.780871	N/A
	The Imitation Game	Did Not Watch	1.674751	N/A
	Dunkirk	2	2.071603	0.071603
User 4	Zero Dark Thirty	3	3.703362	0.703362

	The Imitation Game	4	4.221684	0.221684
	Dunkirk	4	4.093868	0.093868

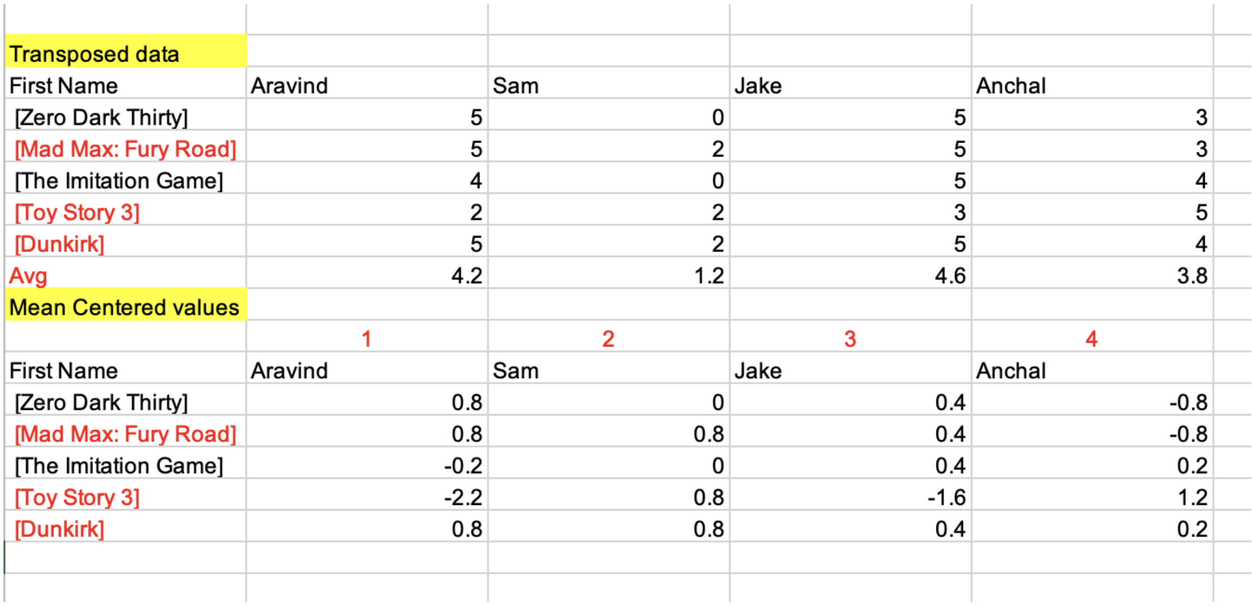
Table 2

Cold Start Prediction using user-based collaborative filtering where the baseline prediction is that movie's average rating.

Name	**Movie**	**Cold Rating**	**Rating with Data**	**Change**
Shachi Govil	Avatar	3.935887	4.232074	0.296189
	Wolf of Wall Street	4.059195	4.209444	-0.150249
	Inception	4.444184	4.177216	-0.266968
Amy Russell	Avatar	3.935887	3.171662	0.839994
	Wolf of Wall Street	4.059195	3.2053	-0.85385
	Inception	4.444184	3.241959	-1.202225
Camille Mack	Avatar	3.935887	2.729097	-1.20679
	Wolf of Wall Street	4.059195	2.757855	-1.30134
	Inception	4.444184	2.798115	-1.646069

Figure 1

The results of our first, memory-based approach (done in excel)



Transposed data				
First Name	Aravind	Sam	Jake	Anchal
[Zero Dark Thirty]	5	0	5	3
[Mad Max: Fury Road]	5	2	5	3
[The Imitation Game]	4	0	5	4
[Toy Story 3]	2	2	3	5
[Dunkirk]	5	2	5	4
Avg	4.2	1.2	4.6	3.8
Mean Centered values				
	1	2	3	4
First Name	Aravind	Sam	Jake	Anchal
[Zero Dark Thirty]	0.8	0	0.4	-0.8
[Mad Max: Fury Road]	0.8	0.8	0.4	-0.8
[The Imitation Game]	-0.2	0	0.4	0.2
[Toy Story 3]	-2.2	0.8	-1.6	1.2
[Dunkirk]	0.8	0.8	0.4	0.2

Table 3

These are the predicted and original ratings produced by the original memory-based approach.

Name	Movie	True Rating	Predicted Rating
User 1	Zero Dark Thirty	5	4.90446371
	Mad Max Fury Road	5	4.808927419
	The Imitation Game	5	4.023884073
	Toy Story 3	3	2.589403251
	Dunkirk	5	4.386712676

User 2	Zero Dark Thirty	5	4.751626305
	Mad Max Fury Road	5	4.697041496
	The Imitation Game	4	4.30432953
	Toy Story 3	2	2.833503519
	Dunkirk	5	4.249744721
User 3	Zero Dark Thirty	Did not watch	3.572690732
	Mad Max Fury Road	2	3.572690732
	The Imitation Game	Did not watch	4.140963901
	Toy Story 3	2	5.794812354
	Dunkirk	2	3.918842279
User 4	Zero Dark Thirty	3	3.658731591
	Mad Max Fury Road	3	3.746840675
	The Imitation Game	4	4.122038968
	Toy Story 3	5	5.989875343
	Dunkirk	4	3.746840675

Model Improvements

An issue with our original memory-based approach is that although there is only one user that rated both item1 and item2, the similarity between those items would be 1. In most cases, we can have millions of users and the similarity between two completely different movies could be very high simply because they have similar rank for the only user who ranked them both (Business et al., 2021).

We used Root Mean Square Error (RMSE) to determine model performance. The lower the RMSE, the better the performance. Matrix factorization is one of the collaborative filtering algorithms used in recommender systems which works by decomposing the user-item interaction matrix into the product of two lower dimensional rectangular matrices. This technique decomposes users and items in a lower dimensional latent space by introducing some k number of latent factors(Business et al., 2021). Collaborative filtering is a recommendation system that creates a prediction based on a user's previous behaviors and hence, it works best when there is a large amount of substantial history of user opinion available(Kordík, 2016).

Considering ecommerce businesses, recommender systems are proven to be very efficient in reducing the time required to find an item and as a result, increase the probability of discovering other items of interest. The result is loyalty and increased satisfaction of users with these services, thereby reducing churn rates and increasing lifetime value of customers (Kordík, 2016). It is observed that having a recommender increases user activity. There are different metrics to measure the performances of a recommender system such as click through rates.