

CS EE World
<https://cseeworld.wixsite.com/home>
May 2023
18/34
C

Submitter Info:
Anonymous

Using evaluation metrics to analyze the efficiency of collaborative filtering technique

RQ: How does Netflix's Recommendation Algorithm use a collaborative filtering-based recommender system to provide efficient recommendations to users?

Registered Subject: Computer Science

Candidate Code: knw813

Session: May 2023

Word Count: 3912

Table of Contents

1. Introduction.....	3
2. Background Information.....	4
2.1 Evolution of Recommending Algorithms.....	4
2.2 Collaborative Filtering	5
2.3 Different Algorithms for Prediction.....	5
2.3.1 K – Nearest Neighbour (k-NN) Algorithm for Predictions	5
2.3.2 Cosine Similarity.....	6
2.4 Metrics used for evaluating recommender systems.....	7
2.4.1 Network Graphs	7
2.4.2 Dendrograms	7
2.4.3 Non-traditional metrics	8
2.4.4 Offline Experiments	8
3. Experiment: Simulating Netflix’s Recommendation Algorithm.....	9
3.1 Background Information.....	9
3.2 Hypothesis.....	12
3.3 Methodology for Prediction Using User-Based Collaborative Filtering.....	12
3.4 Experiment	12
3.4.1 Get user-item rating data.....	12
3.4.2 Create Correlation Similarity Tables	13
4. Experimental Results and Analysis.....	15
5. Evaluation of Hypothesis.....	19
6. Evaluation and Conclusion	19
Works Cited.....	21
Appendix.....	24

1. Introduction

The growth of the digital world is incessant, progressing at a pace that surpasses any other technological advancement in human history. It has been speculated that the pace of advancement in the digital realm is unparalleled. A great contribution to the success of the digital world is the ability to connect with everyone in their own individual methods. To be more specific, the ability to understand what an user likes, dislikes and wants, creates a personal bond with humans and technology that allows it to continue growing. Thus, it is with the help of recommender systems/ algorithms designed to suggest relevant items to users, that makes the digital world so powerful in its own way. Recommender systems is a general term for the one in many algorithms, each individually crafted in their own way to acquire certain information from the user, allowing technology to maintain the personal connection with the user. Many entertainment applications use recommender systems. Streaming service apps such as Netflix, Amazon Prime, Disney Plus all use recommender systems to suggest what the user can watch next, or what the user may like to watch. To quantify the significance of recommender systems, sources show that Amazon directly attributes an estimated 35% of sales to their recommending algorithms.

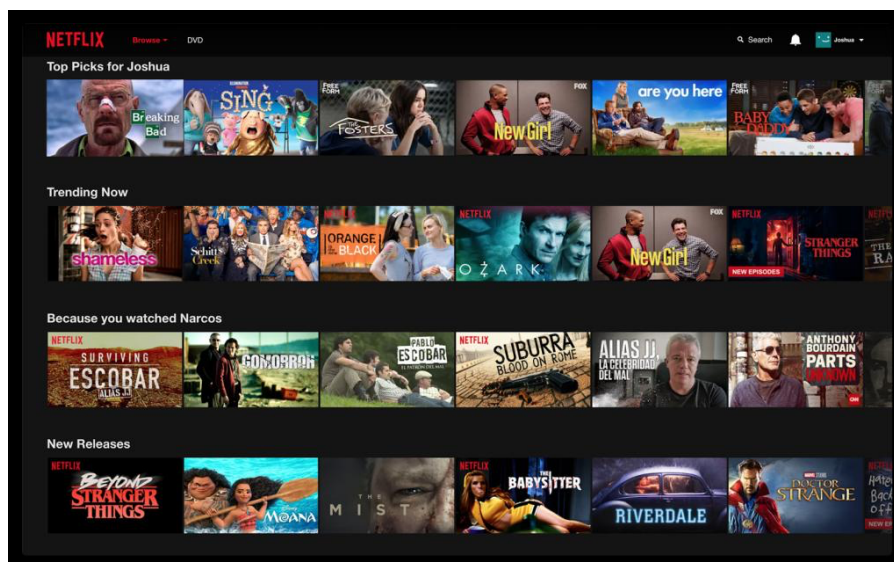


Figure 1. An example of the recommender system of Netflix. “Top Picks” and “Because you watched” all suggest an example of Netflix’s recommender.¹

¹. “Netflix Algorithm: Everything You Need to Know about the Recommendation System of the Most Popular Streaming Portal.” *Recostream*, recostream.com/blog/recommendation-system-netflix.

2. Background Information

2.1 Evolution of Recommending Algorithms

Recommending Algorithms were first introduced in the 1970s. Since then, a lot of trials and experiments have been performed for the development of recommender systems. It made its debut at a time when spam and pointless emails were becoming more prevalent. A collaborative filtering method was created by Xerox Palo Alto Research staff to ensure that users only received email from contacts or people they might like to hear from. Rest of the emails would go to the Spam Folder. This is exactly how our current mailing system works. To understand how a collaborative method was used, factors such as similar contacts, similar subscribed mails and similar locations all contribute to the algorithm deciding which emails should go to the user and which emails should be sent to the spam folder.² Since then, recommender systems have been evolving and the figure below represents the extent this psychological algorithm has grown to.

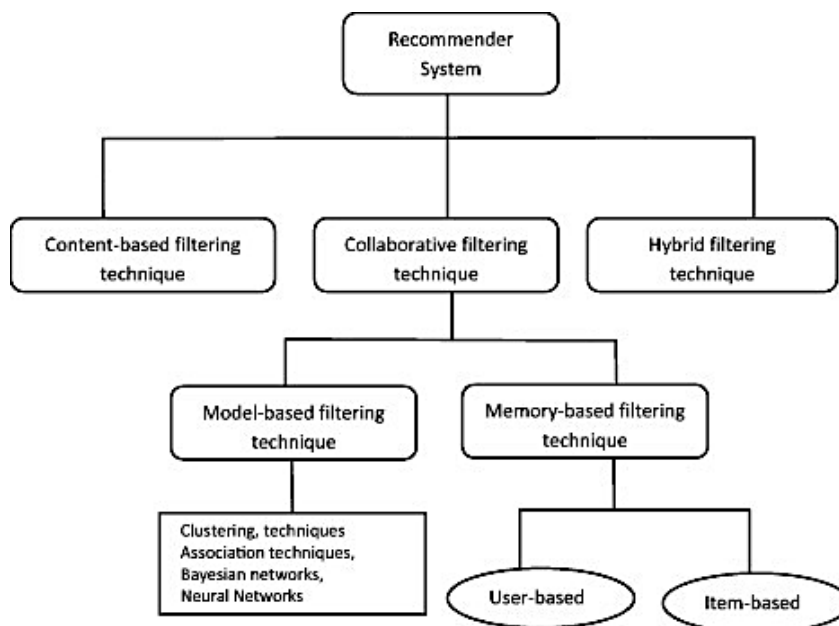


Figure 2. Chart representing different branches of recommender system

Recommender systems are branched out into three main techniques: Content-based filtering, Collaborative Filtering and Hybrid Filtering.

² Sharma, Richa, and Rahul Singh. "Evolution of Recommender Systems from Ancient Times to Modern Era: A Survey." *Indian Journal of Science and Technology*, vol. 9, no. 20, 2016, doi:10.17485/ijst/2016/v9i20/88005.

2.2 Collaborative Filtering

CBF is commonly referred to as "people-to-people correlation," and its fundamental concept is that users who share similar interests are likely to receive recommendations for comparable items. These similar interests can be measured from browsing history, watch history, ratings etc. However, a disadvantage of measuring from browsing histories is that sometimes, multiple users use a single device so the recommender system is prone to be confused in such situations. This can be easily understood with the example of Facebook's friend suggestions. Measuring the mutual friends, number of similar posts the user follows and likes, groups the users are in common, common places the users have been in are all used to suggest new potential "People that you may know". Thus, the name people to people correlation.

2.3 Different Algorithms for Prediction

2.3.1 K – Nearest Neighbour (k-NN) Algorithm for Predictions

The k-Nearest Neighbour (k-NN) algorithm is a machine learning technique utilized for classifying and predicting data points, whether categorical or continuous. The algorithm is straightforward and easy to comprehend, as it operates on the principle that objects or data points that are alike are typically situated in close proximity to one another. The k-NN algorithm operates by identifying the k most similar data points in the training dataset to a novel data point, where k is a user-defined parameter. These closest data points are called the k-nearest neighbours. The classification or prediction of the new data point is based on the class or value of the k-NN. For instance, the value of k is set to 3; in that case, the algorithm will locate the three data points in the training dataset that share the most similarity to the new data point. If two of the neighbours belong to one class and the third belongs to another class, the algorithm will classify the new data point as belonging to the class with the most neighbour's (in this case, the first class). In the case of regression tasks, the algorithm computes the average or median value of the k-nearest neighbour's and uses that as the predicted value for the new data point. One of the advantages of the k-NN algorithm is that it can work well with both linear and nonlinear data. However, the choice of the value for k and the distance metric used can play a major

role. Additionally, as the size of the training set grows, the computation required to find the k-nearest neighbour's becomes more expensive.³

2.3.2 Cosine Similarity

A common method in recommender systems to compare two things or two users based on their preferences or ratings is cosine similarity. It is a mathematical measure of the similarity between two vectors in a multi-dimensional space. In a recommender system, each item can be represented as a vector of its features, and the ratings given by the users can be represented as another vector in the same space. The formula for cosine similarity is given below. It ranges from -1 to 1.

When two vectors have a cosine similarity of 1, they are identical and point in the same direction, however when they have a cosine similarity of -1, they are exactly opposite. Two vectors are orthogonal and have no similarity when the cosine similarity is zero.

$$\cos \theta = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}} \quad (1)$$

Equation for cosine similarity

Since cosine similarity analyzes the distance between two different data points in a dimensional space, there are different ways to calculate the distance between them.

2.3.2.1 Euclidean Distance

The straight-line separation between two locations in a plane or three-dimensional space is known as Euclidean space. The Pythagorean theorem can be used to determine the Euclidean separation between two points (P1, Q1) and (P2, Q2). In this case, the two sides represent the differences between the x-coordinates and y-coordinates of the two points.

The formula for calculating the Euclidean distance between two points in two-dimensional space is:

$$Distance = \sqrt{(P1 - P2)^2 + (Q1 - Q2)^2}$$

In three-dimensional space, the formula becomes:

$$Distance = \sqrt{(P1 - P2)^2 + (Q1 - Q2)^2 + (Z1 - Z2)^2}$$

³ Lecture Notes on K-Nearest Neighbors. www.di.ens.fr/appstat/fall-2018/notes/cours_knn.pdf.

where (x_1, y_1, z_1) and (x_2, y_2, z_2) are the coordinates of two points⁴

In the context of a recommender system, we can use cosine similarity to compare the similarity that neighboring users have such as similar genres, cast, etc. We can find the items that have the highest cosine similarity with the user's rated items and recommend them to the user. For instance, if a user has shown interest in romantic comedies in the past, the recommender system might use cosine similarity to identify other romantic comedies that are similar in terms of their features, such as actors, directors, plot, and genre. The system might then recommend these similar movies to the user.⁵

Refer to the appendix for a recommending algorithm based on cosine similarity.

2.4 Metrics used for evaluating recommender systems

2.4.1 Network Graphs

In a recommending system, a network graph can be used to represent user-item interactions, where each node in the graph represents a user or an item and the lines between the users represent the interactions between them. For example, in a movie recommendation system, a network graph can represent the connections between users and movies or different users, where each node represents a user or a movie, and the lines would represent the user's interactions with the movie (e.g., rating, watching, etc.). Network graphs can also be used to represent other types of relationships and interactions, such as social connections, interests, and preferences. By visualizing the connections between different entities, network graphs can help to identify patterns and relationships that can inform the recommending system's algorithms and improve the accuracy of the recommendations.

2.4.2 Dendrograms

Dendrograms are a type of visualization tool used in recommending systems to represent the hierarchical clustering of items or users based on their similarities or preferences. They provide a visual representation of

⁴ "Euclidean Distance Formula - Derivation, Examples." *Cuemath*, www.cuemath.com/euclidean-distance-formula/.

⁵ Javed, Mahnoor. "Using Cosine Similarity to Build a Movie Recommendation System." *Medium*, Towards Data Science, 4 Nov. 2020, towardsdatascience.com/using-cosine-similarity-to-build-a-movie-recommendation-system-ae7f20842599.

the relationships and clusters between different items or users. In a recommending system, a dendrogram can be used to represent the similarity or preference scores between different items or users. Each leaf node in the dendrogram represents an item or user, and the height of the branches represents the similarity or preference score between them. The closer the leaf nodes are on the dendrogram, the more similar or preferred they are. Dendrograms can also be used to represent the hierarchical structure of items or users based on their attributes. For example, in a clothing recommendation system, a dendrogram can represent the hierarchical structure of clothing items based on their attributes such as style, color, and material. This can help users to explore and navigate the available items in a more intuitive way.⁶

2.4.3 Non-traditional metrics

Non-traditional metrics in a recommender system refer to factors like diversity, which assesses how varied the recommendations are, serendipity, which measures the likelihood of making accidental discoveries, and coverage, which is the percentage of items in the training data that the model is capable of recommending on a test set.

2.4.4 Offline Experiments

Interviews, surveys and other marketing related experiments are really proven to be helpful in measuring a recommending systems efficiency. Moreover, these are first-hand information gained from the physical presence of users which completely eliminates any source of malpractice or misunderstandings.⁷

All these metrics above are the most common when it comes to evaluating a recommender system. It needs to be understood that new ways for analyzing recommender systems keep changing. Metrics always help express a personality of a recommender system that needs to be maximized or minimized in order to keep the user interaction high and make the most use of the algorithm.

⁶ Bock, Tim. "What Is a Dendrogram?" *Displayr*, 13 Sept. 2022, www.displayr.com/what-is-dendrogram/.

⁷ "What Metrics Are Used for Evaluating Recommender Systems?" Quora, www.quora.com/What-metrics-are-used-for-evaluating-recommender-systems.

3. Experiment: Simulating Netflix's Recommendation Algorithm

3.1 Background Information

With more than 139 million paying subscribers across 190 countries, Netflix serves as the most popular streaming service in the world. It is also a very successful platform. Their incredibly intelligent recommendation algorithm, also known as the Netflix Recommendation Algorithm or NRE, is what makes them so successful. This software, which was created especially for the streaming service, has been crucial to Netflix's development.

Several algorithms make up the Netflix Recommendation Algorithm (NRE), in which they filter content depending on a user's profile. It examines more than 3,000 books utilizing 1,300 recommendation clusters, all of which are based on the tastes of the individual user. This approach is highly effective, as 80% of Netflix's viewer activity is due to personal recommendations, which prevents subscribers from canceling and saves Netflix billions of dollars annually. Netflix believes they only have 90 seconds to capture a viewer's attention, and by advertising content with a high chance of being viewed, they can ensure that their 80% success rate keeps customers happy. If Netflix didn't utilize the NRE, it fears that it would lose more than \$1 billion in revenue annually as a result of consumers cancelling their subscriptions. Personalization is crucial in the modern consumer world, as evident from its use by digital platforms like Spotify and Amazon.

Each row on the page is customized in three ways. Firstly, it has a personalized name such as "Trending Now" or "Award-Winning Dramas". Secondly, it displays specific titles that correspond to that particular row. Finally, each title is ranked within its respective row. The recommendation system uses various algorithmic methods, including reinforcement learning, casual modelling, matrix factorization, and bandits, to determine the order in which the rows are displayed. The rows that are most highly recommended are placed at the top of the page, while less strong recommendations are placed at the bottom.

Netflix gathers the following information from users:

1. Time duration of a viewer watching a video.
2. Viewing history.
3. How titles were rated by the user.
4. Other users who may have similar tastes.
5. Information about titles such as genre, actors and release year.
6. The time of day you watch.
7. When the user watches a scene more than once.
8. If the show was paused, rewound, or fast-forwarded.
9. If the viewer resumed watching after pausing.
10. The device you are watching on.
11. The number of searches and what is searched for.
12. Screen shots when the show was paused.
13. When the user left the show.⁸

From what was once a company that issued movies via snail mail, to becoming the largest online media streaming platform, Netflix has come a long way. A main contribution for that is their recommendation systems. Ever since they started giving importance to their recommendation systems, Netflix has received an abundance of customers annually. There is statistical evidence to support the claim that Netflix's recommendation algorithms are responsible for around 80% of the content that users choose to watch on the platform.

⁸ Invisibly. "The Full Guide on Netflix Recommendation Algorithm: How Does It Work?" *Invisibly*, 8 Dec. 2022, www.invisibly.com/learn-blog/netflix-recommendation-algorithm/.

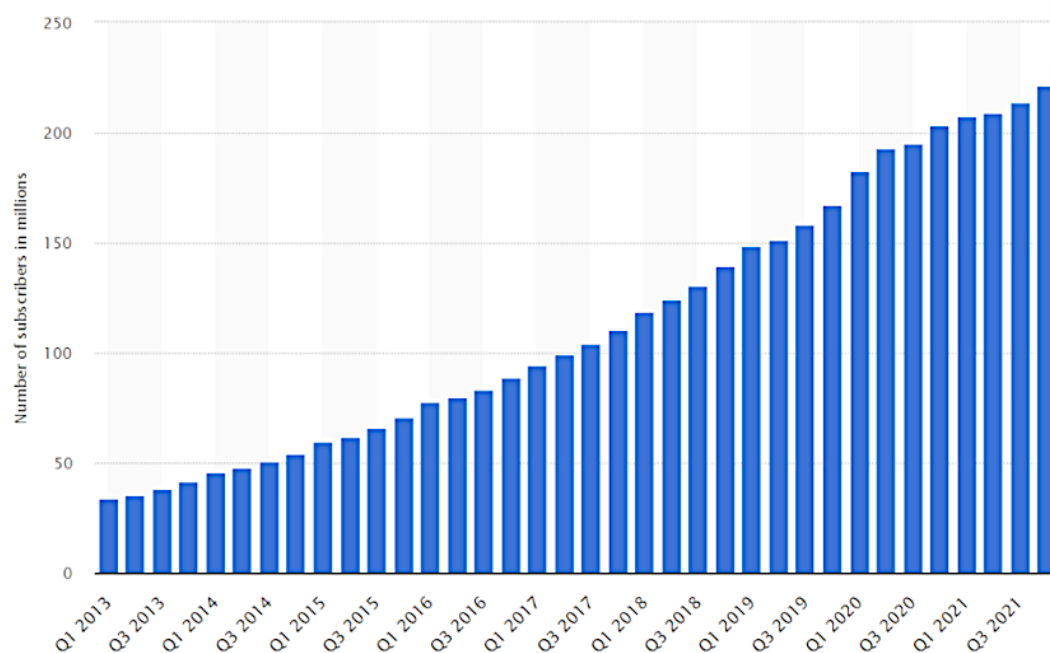


Figure 3. Statistical representation of number of subscribers for Netflix from 2013 quarterly.⁹

Netflix's recommendation algorithm is based on a collaborative filtering technique, but however there are much more complex machine learning algorithms imposed on this technique. Netflix provides predictions for a user based on the following: The ratings of a movie/series is stored in a database and when a new user rates a corresponding movie/series, it is compared with the database and a similarity computation is generated. Specifically, it tackles the similarities between the users and items to perform recommendations. To clearly understand Netflix's collaborative recommendation algorithm, an offline survey will be conducted of user's ratings of 5 different movies: *Avengers: Endgame*, *Home Alone*, *La La Land*, *The Conjuring*, *Harry Potter*. Some ratings for certain movies by certain users are not filled as they have not watched the movies, hence, the steps in prediction using k-NN method and matrix factorizations will be used in producing a prediction for the ratings that are unfilled in the survey.

⁹ "Netflix Recommendation System: How It Works." *RecoAI*, recoai.net/netflix-recommendation-system-how-it-works/#:~:text=Netflix%20began%20back%20in%201997,to%20improve%20their%20recommendations2.

3.2 Hypothesis

Using my knowledge in the research I've done in recommendation algorithms, my general hypothesis in how this algorithm works is by using a collaborative filtering system. Most companies use collaborative filtering method for their prediction algorithms more than content-based as content-based is very simple and only uses the content's characteristics while the latter considers the user's preferences as well. This explains why Netflix's recommending algorithms are very user centric. Hence, my final hypothesis would be that Netflix uses a k-NN algorithm with matrix factorization to provide recommendations to their users.

3.3 Methodology for Prediction Using User-Based Collaborative Filtering

1. Get User-Item Rating Data
2. Creating Cosine Similarity Matrixes for all users
3. Predicting ratings

3.4 Experiment

3.4.1 Get user-item rating data

Firstly, an offline survey was performed to 5 different users to compare the ratings of 5 different movies: Avenger: Endgame, Home Alone, La La Land, The Conjuring, Harry Potter. Each user was asked to rate each movie, each of different genre, out of 10. The evidence for conducting this experiment is attached in the appendix. The results of the survey are shown below.

	Avengers: EndGame	Home Alone	La La Land	The Conjuring	Harry Potter: Goblet of Fire
User 1	8	6	0	7	5
User 2	0	8	7	0	0
User 3	5	0	7	0	8

User 4	0	0	0	3	5
User 5	7	3	0	8	0

Table 1. Experimental data of five different users rating five different movies

Netflix requires access to users' historical data, which is typically organized in a matrix format known as the "user item matrix." Assuming that Netflix has M users and N items, the user item rating data needs to be stored in a matrix that has M rows representing each user and N columns representing each item. Each cell within the matrix contains a rating given by a user to a particular item. When we need to get a user item rating data, we get the [M*N] User item matrix as created above in Table 1.

3.4.2 Create Correlation Similarity Tables

In the second step we will be creating a user-to-user similarity matrix. To find the similarity between two users' interests, we need to have a similarity measure. For this any type of correlating method can be used and, therefore, the cosine similarity method will be used to predict ratings. The cosine similarity formula (1) is listed below:

$$\cos \theta = \frac{A \cdot B}{\|A\| \|B\|} = \frac{\sum_{i=1}^n A_i B_i}{\sqrt{\sum_{i=1}^n A_i^2} \sqrt{\sum_{i=1}^n B_i^2}}$$

Each user will be iterated through and the any ratings that the corresponding user hasn't rated, will be predicted.

	Avengers: Endgame	Home Alone	La La Land	The Conjuring	Harry Potter: Goblet of Fire
User 1	8	6	0	7	5
User 2	0	8	7	0	0
User 3	5	0	7	0	8
User 4	0	0	0	3	5
User 5	7	3	0	8	0

Table 1.

To make the use of cosine similarity much easier, the numerator and denominator is divided into two different calculations.

Using the SQRT, SUMPRODUCT and SUMSQ methods in Microsoft Excel, the calculations are done in a much more efficient manner.

Cosine Similarity for User 1			
Numerator	Denominator	Similarity	Ranking
48	140.2212537	0.342316152	4 (User 2)
80	154.9580588	0.516268728	3 (User 3)
46	76.91553809	0.598058613	2 (User 4)
130	145.6983185	0.892254635	1 (User 5)

Table 2. Cosine Similarity for User 1

The ranking is ordered from the greatest similarity to the least similarity.

The same is done for all users:

Cosine Similarity for User 2			
Numerator	Denominator	Similarity	Ranking
48	140.2212537	0.342316152	2 (User 1)
49	124.8759384	0.392389444	1 (User 3)
0	61.98386887	0	4 (User 5)
24	117.4137982	0.204405278	3 (User 4)

Table 3. Cosine Similarity for User 2

Cosine Similarity for User 3			
Numerator	Denominator	Similarity	Ranking
80	154.9580588	0.516268728	2 (User 1)
49	124.8759384	0.392389444	3 (User 2)
40	68.49817516	0.583957162	1 (User 4)
35	129.7536127	0.269742008	4 (User 5)

Table 4. Cosine Similarity for User 3

Cosine Similarity for User 4			
Numerator	Denominator	Similarity	Ranking
40	68.49817516	0.583957162	2 (User 3)

46	76.91553809	0.598058613	1 (User 1)
0	61.98386887	0	4 (User 2)
24	64.40496875	0.372642056	3 (User 5)

Table 5. Cosine Similarity for User 4

Cosine Similarity for User 5			
Numerator	Denominator	Similarity	Ranking
24	64.40496875	0.372642056	2 (User 4)
35	129.7536127	0.269742008	3 (User 3)
130	145.6983185	0.892254635	1 (User 1)
24	117.4137982	0.204405278	4 (User 2)

Table 6. Cosine Similarity for User 5

Using these cosine similarity tables we can give predict ratings for the movies that the user hasn't rated. This is done by calculating the averages of the ratings given by the two most similar users to the corresponding user that the movie is being predicted for. For instance, User 1 hasn't watched only one movie, *La La Land*. The prediction for this movie can be given by calculating the average rating of the two most similar users (user 5 and user 4). However, user 5 and user 4 have also not watched *La La Land* and, hence, the average rating of the three most similar users will be taken which is 2.33. This, therefore, is the prediction User 1 would give for *La La Land*. Similarly, the entire table is filled.¹⁰

4. Experimental Results and Analysis

	Avengers: Endgame	Home Alone	La La Land	The Conjuring	Harry Potter: Goblet of Fire
User 1	8	6	2.33	7	5
User 2	6.50	8	7	3.5	6.5
User 3	5	3	7	5	8
User 4	6.5	3	3.5	3	5

¹⁰ "Collaborative Filtering Recommender System with Excel." *YouTube*, YouTube, 24 June 2021, www.youtube.com/watch?v=efW4vPh9snc&t=1139s.

User 5	7	3	2.92	8	5
---------------	---	---	-------------	---	----------

Table 7. Predicted Ratings

Below is a diagram that visually represents what the user prefers and does not. Ratings higher than 5 have been considered as “liked” by the user which is denoted by an arrow navigating towards the corresponding movie. Any user that is not linked with a movie shows that the user has rated it below 5.

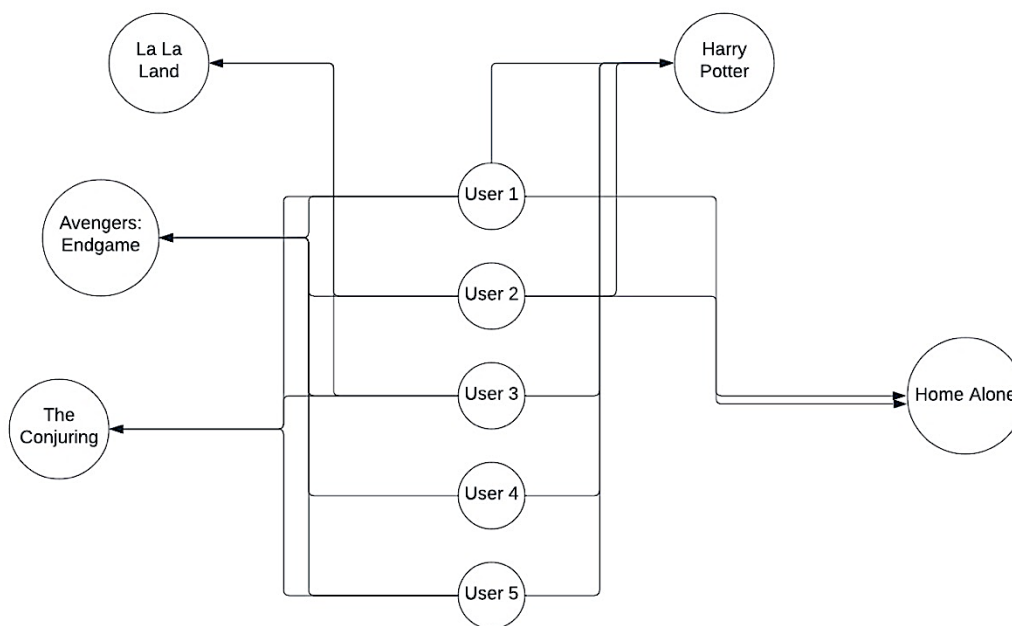


Figure 4. User-Item Node Structure

Using the visual representation, users have been clustered into a common movies that they have rated higher than 5, and common movies that they have rated below 5. This can be efficient in sorting users in groups and giving them similar recommendations. In addition to the diagrams, user-user similarity matrixes are created to further analyze the individual relation between each corresponding user.

	User 1	User 2	User 3	User 4	User 5
User 1	1	0.342316152	0.516268728	0.598058613	0.892254635
User 2	0.342316152	1	0.392389444	0	0.204405278
User 3	0.516268728	0.392389444	1	0.583957162	0.269742008
User 4	0.598058613	0	0.583957162	1	0.372642056
User 5	0.892254635	0.204405278	0.269742008	0.372642056	1

Table 8. User-User Similarity Matrix

These similarity matrixes are the basis of the many analyses the recommending systems perform to analyze the similarity between users.

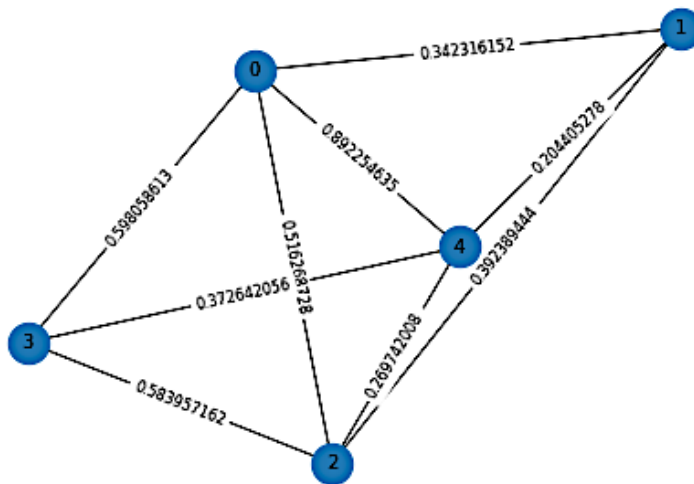


Figure 5. Network Graph

This is a network graph that compares each user with another user with a minimum similarity threshold. Only if this threshold measure (similarity value) is met by the users, the connection between that pair of users will be created. Depending on the recommending system's needs, the threshold measure can be altered. In this case, the threshold measure is set to 0 to analyze the similarity between every single user. The algorithm for generating this network graph can be found in the appendix.

A dendrogram below is also created to show the relationship between the distance between each user. It also helps the reader understand the hierarchical structure between different users, such as which users have a broad range of watch history and similarities with other users and which users have niche ratings. This can help with the analysis of which users are consistent in their ratings and which users aren't. Using a dendrogram below, algorithms can analyse the reliability of each user.

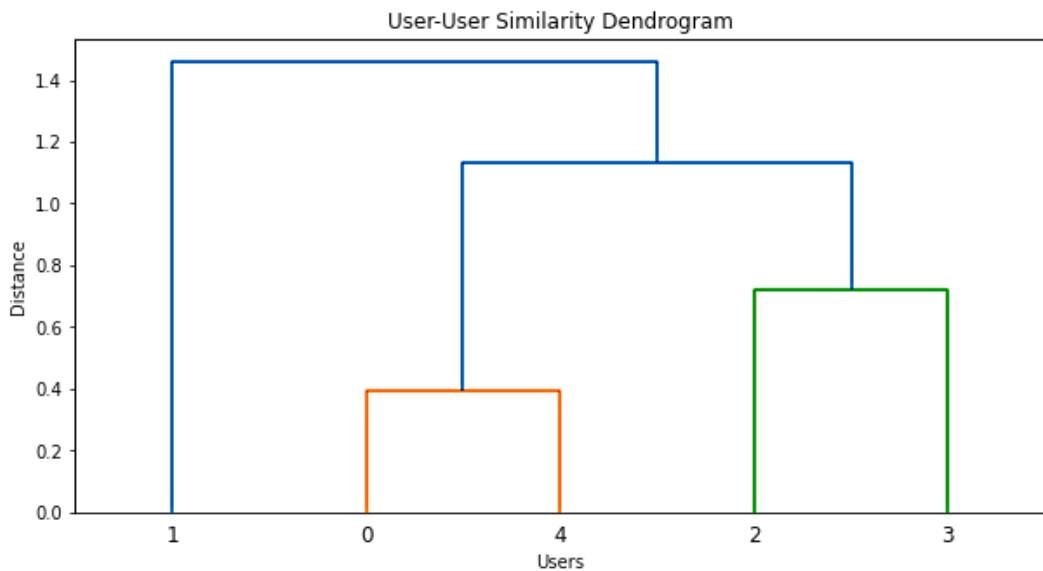


Figure 6. Dendrogram

Analyzing the graph, user 1 has the greatest distance measured with the other users, hence, it's known to be the most different or unique from the rest of its users. This is known as height analysis.

Netflix collects data about users' viewing habits, movie ratings, and other relevant information. This data is then used to build a profile of each user's preferences. Similarly, Netflix uses data about each movie's attributes such as genre, actors, director, and plot to build a profile of each movie. When a user visits Netflix, the system recommends movies based on the similarity score between the user's data and the data of available movies.

The recommendation engine calculates the cosine similarity between the user's profile and the profiles of movies that the user has not yet watched. The system then recommends the movies with the highest similarity score. For example, if a user has watched and rated several action movies, the system will recommend other action movies with similar attributes such as similar genre, actors, and plot. The recommendation engine will then use cosine similarity to compare the user's preferences with the features of the recommended action movies. The movies with the highest cosine similarity score will be recommended to the user.

5. Evaluation of Hypothesis

My hypothesis was partially correct, as my inference of using the k-NN algorithm (using users nearby/ratings to compare with) and the matrix factorization was correct. However, Netflix uses a much more complex system with multiple machine learning algorithms present in recommending the most preferable item for the user.

6. Evaluation and Conclusion

The experiment performed above was a simple experiment to demonstrate the immense complexity that Netflix's recommendation algorithm possesses. Through my experiment, I was able to predict the ratings of users' recommendation using the cosine similarity method. In Netflix's case, all the steps in the experiment are performed by multiple algorithms which use machine learning which makes the automation process very efficient. Moreover, in my experiment there is a possibility of human errors such as calculating mistakes or bias ratings but there is no space for either of these errors in digitized algorithms. A limitation that this experiment had is the amount of data collected. If a larger number of samples were collected, the data collected would have been more accurate. For instance, for certain movies the two most similar users have also not watched the corresponding movie, hence, it was difficult in some cases to predict the ratings with limited data. A question that is raised from this experiment and the evaluation methods is that how would the algorithm measure error without the user's actual rating. Netflix has a trained algorithm with millions of user's ratings, making it very reliable. However, the same cannot apply for an infant recommender system. How will initial trials and failures be measured? Are there any specific methods that a user follows to reduce the error in a prediction algorithm? Overall, this experiment shows basic processing of a recommender algorithm.

In terms of future scope, the dendrogram can be analyzed in more depth with multi-dimensional scaling. Multidimensional scaling (MDS) is a statistical technique that allows the visualization of the relationships between objects or clusters in a lower-dimensional space. By mapping the dendrogram onto a 2D or 3D space

using MDS, you can explore the relationships between clusters and identify patterns that may not be apparent in the dendrogram. In addition, the experiment would have had a possibility of providing more accurate results if there was to be a GUI developed which provides an option for the user to rate as that would seem more realistic for the user. If the user is being surveyed, he or she might be under pressure and will have an extra need to provide accurate results. Moreover, they can infer that their results will not be used commercially, hence, this puts an additional layer of pressure on them. This might result in accurate info. However, a GUI wouldn't have developed any sense of pressure on the user as they are used to it.

Recommender systems are a major contribution to today's digital world. Using my experiment, it is important to understand the significance of the impact it creates on users such as the manipulation of user's preferences. From scrolling through Netflix, Instagram, Tik Toks, they play a major role in keeping the user from spending more time in the corresponding app and it is important to know the science behind how devices lure users and maintain a healthy distance from it.

Works Cited

- Amatriain, Xavier, and Justin Basilico. "Recommender Systems in Industry: A Netflix Case Study." *SpringerLink*, Springer US, 1 Jan. 1970, link.springer.com/chapter/10.1007/978-1-4899-7637-6_11.
- Amy. "Recommendation System: User-Based Collaborative Filtering." *Grab N Go Info*, 4 Jan. 2023, grabngoinfo.com/recommendation-system-user-based-collaborative-filtering/.
- Author links open overlay panelF.O. Isinkaye a, et al. "Recommendation Systems: Principles, Methods and Evaluation." *Egyptian Informatics Journal*, Elsevier, 20 Aug. 2015, www.sciencedirect.com/science/article/pii/S1110866515000341.
- Baldha, Shivam. "Introduction to Collaborative Filtering." *Analytics Vidhya*, 22 Mar. 2022, www.analyticsvidhya.com/blog/2022/02/introduction-to-collaborative-filtering/.
- "Bayesian Networks." *University of Bergen*, www.uib.no/en/rg/ml/119695/bayesian-networks#:~:text=Bayesian%20networks%20are%20a%20widely,dom%20variables%20associated%20with%20nodes.
- Bock, Tim. "What Is a Dendrogram?" *Displayr*, 13 Sept. 2022, www.displayr.com/what-is-dendrogram/.
- "Collaborative Filtering Recommender System with Excel." *YouTube*, YouTube, 24 June 2021, www.youtube.com/watch?v=efW4vPh9snc&t=1139s.
- Computer Science Case Study: May I Recommend the Following?*
computersciencewiki.org/images/b/bc/2023_case_study.pdf.
- "EM Algorithm in Machine Learning - Javatpoint." *Www.javatpoint.com*, www.javatpoint.com/em-algorithm-in-machine-learning.
- "Euclidean Distance Formula - Derivation, Examples." *Cuemath*, www.cuemath.com/euclidean-distance-formula/.

“How Netflix's Recommendations System Works.” *Help Center*, help.netflix.com/en/node/100639.

“How Recommender Systems Work (Netflix/Amazon).” *YouTube*, YouTube, 28 Feb. 2020,

www.youtube.com/watch?v=n3RKsY2H-NE.

“IB Computer Science - Paper 3 - Case Study (2023) - May I Recommend the Following?” *YouTube*,

YouTube, 2 Sept. 2022, www.youtube.com/watch?v=6PZEVNuBL0g&t=2390s.

Invisibly. “The Full Guide on Netflix Recommendation Algorithm: How Does It Work?” *Invisibly*, 8 Dec.

2022, www.invisibly.com/learn-blog/netflix-recommendation-algorithm/.

Javed, Mahnoor. “Using Cosine Similarity to Build a Movie Recommendation System.” *Medium*, Towards

Data Science, 4 Nov. 2020, towardsdatascience.com/using-cosine-similarity-to-build-a-movie-recommendation-system-ae7f20842599.

Johnson, Daniel. “Reinforcement Learning: What Is, Algorithms, Types & Examples.” *Guru99*, 21 Jan.

2023, www.guru99.com/reinforcement-learning-tutorial.html.

Lecture Notes on K-Nearest Neighbors. www.di.ens.fr/appstat/fall-2018/notes/cours_knn.pdf.

Longo, Claire. “Evaluation Metrics for Recommender Systems.” *Medium*, Towards Data Science, 1 Dec.

2022, towardsdatascience.com/evaluation-metrics-for-recommender-systems-df56c6611093.

“Measuring Recommender System Accuracy.” *YouTube*, YouTube, 27 Aug. 2018,

www.youtube.com/watch?v=gTaN9bWOea0&list=PLiQ6aBl318Zk1Lshw6kKuZijq557N+bw+HG&index=1.

Meltzer, Rachel. “How Netflix Utilizes Data Science.” *Lighthouse Labs*, 7 July 2020,

www.lighthouselabs.ca/en/blog/how-netflix-uses-data-to-optimize-their-product.

“Netflix Algorithm: Everything You Need to Know about the Recommendation System of the Most Popular Streaming Portal.” *Recostream*, recostream.com/blog/recommendation-system-netflix.

“Netflix Recommendation System: How It Works.” *RecoAI*, recoai.net/netflix-recommendation-system-how-it-works/#:~:text=Netflix%20began%20ba ck%20in%201997,to%20improve%20their%20recommendations2.

Saluja, Chhavi. “Collaborative Filtering Based Recommendation Systems Exemplified..” *Medium*, Towards Data Science, 12 Mar. 2018, towardsdatascience.com/collaborative-filtering-based-recommendation-systems-exemplified-ecbffe1c20b1.

Saluja, Chhavi. “Collaborative Filtering Based Recommendation Systems Exemplified..” *Medium*, Towards Data Science, 12 Mar. 2018, towardsdatascience.com/collaborative-filtering-based-recommendation-systems-exemplified-ecbffe1c20b1#:~:text=Item%2DBased%20Collaborative%20Filtering&text=The%20rating%20for%20target%20item,between%20items%20i%20and%20j.

Sharma, Richa, and Rahul Singh. “Evolution of Recommender Systems from Ancient Times to Modern Era: A Survey.” *Indian Journal of Science and Technology*, vol. 9, no. 20, 2016, doi:10.17485/ijst/2016/v9i20/88005.

“What Metrics Are Used for Evaluating Recommender Systems?” *Quora*, www.quora.com/What-metrics-are-used-for-evaluating-recommender-systems.

Appendix

1. Recommender System using Cosine Similarity

```
# store the original dataset in 'df', and create the copy of df, df1 = df.copy().
def movie_recommender(user, num_neighbors, num_recommendation):

    number_neighbors = num_neighbors

    knn = NearestNeighbors(metric='cosine', algorithm='brute')
    knn.fit(df.values)
    distances, indices = knn.kneighbors(df.values, n_neighbors=number_neighbors)

    user_index = df.columns.tolist().index(user)

    for m,t in list(enumerate(df.index)):
        if df.iloc[m, user_index] == 0:
            sim_movies = indices[m].tolist()
            movie_distances = distances[m].tolist()

            if m in sim_movies:
                id_movie = sim_movies.index(m)
                sim_movies.remove(m)
                movie_distances.pop(id_movie)

            else:
                sim_movies = sim_movies[:n_neighbors-1]
                movie_distances = movie_distances[:n_neighbors-1]

            movie_similarity = [1-x for x in movie_distances]
            movie_similarity_copy = movie_similarity.copy()
            nominator = 0

            for s in range(0, len(movie_similarity)):
                if df.iloc[sim_movies[s], user_index] == 0:
                    if len(movie_similarity_copy) == (number_neighbors - 1):
                        movie_similarity_copy.pop(s)

                    else:
                        movie_similarity_copy.pop(s-(len(movie_similarity)-
len(movie_similarity_copy)))

                    else:
                        nominator = nominator + movie_similarity[s]*df.iloc[sim_movies[s],user_index]

            if len(movie_similarity_copy) > 0:
                if sum(movie_similarity_copy) > 0:
                    predicted_r = nominator/sum(movie_similarity_copy)

            else:
                predicted_r = 0

            else:
                predicted_r = 0

            df1.iloc[m,user_index] = predicted_r
            recommend_movies(user, num_recommendation)
```

2. Network Graph Algorithm


```

3. import numpy as np
4. import networkx as nx
5. import matplotlib.pyplot as plt
6.
7. # Generate a random similarity matrix
8. similarity_matrix =
   np.array([[1,0.342316152,0.516268728,0.598058613,0.892254635],
             [0.342316152,1,0.392389444,0,0.204405278],
             [0.516268728,0.392389444,1,0.583957162,0.269742008],
             [0.598058613,0,0.583957162,1,0.372642056],
             [0.892254635,0.204405278,0.269742008,0.372642056,1]])
9.
10.    # Set a threshold similarity score
11.    threshold = 0
12.
13.    # Create a graph object
14.    G = nx.Graph()
15.
16.    # Add nodes for each user
17.    for i in range(similarity_matrix.shape[0]):
18.        G.add_node(i)
19.
20.    # Add edges for pairs of users with similarity above the threshold
21.    for i in range(similarity_matrix.shape[0]):
22.        for j in range(i + 1, similarity_matrix.shape[0]):
23.            if similarity_matrix[i, j] > threshold:
24.                G.add_edge(i, j, weight=similarity_matrix[i, j])
25.
26.    # Draw the graph
27.    pos = nx.spring_layout(G)
28.    nx.draw(G, pos, with_labels=True, node_size=500, font_size=10)
29.    labels = nx.get_edge_attributes(G, "weight")
30.    nx.draw_networkx_edge_labels(G, pos, edge_labels=labels, font_size=8)
31.    plt.show()

```

3. Dendrogram Algorithm

```

import numpy as np
from scipy.cluster.hierarchy import dendrogram, linkage
import matplotlib.pyplot as plt

# Generate a random user-user similarity matrix
similarity_matrix = np.array([[1,0.342316152,0.516268728,0.598058613,0.892254635],
                              [0.342316152,1,0.392389444,0,0.204405278],
                              [0.516268728,0.392389444,1,0.583957162,0.269742008],
                              [0.598058613,0,0.583957162,1,0.372642056],
                              [0.892254635,0.204405278,0.269742008,0.372642056,1]])

# Convert similarity matrix into a distance matrix
distance_matrix = 1-similarity_matrix

```

```
# Apply hierarchical clustering using complete linkage
Z = linkage(distance_matrix, method='complete')

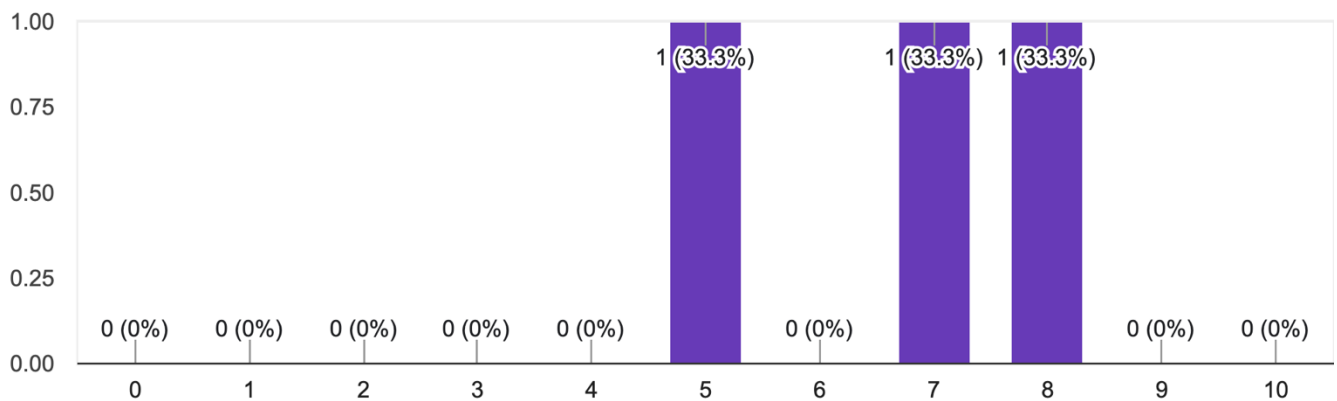
# Plot the dendrogram
plt.figure(figsize=(10, 5))
dendrogram(Z)
plt.title('User-User Similarity Dendrogram')
plt.xlabel('Users')
plt.ylabel('Distance')
plt.show()
```

4. Evidence for offline experiment

Link to google form: <https://forms.gle/ZEeiqaTxK8rFHE599>

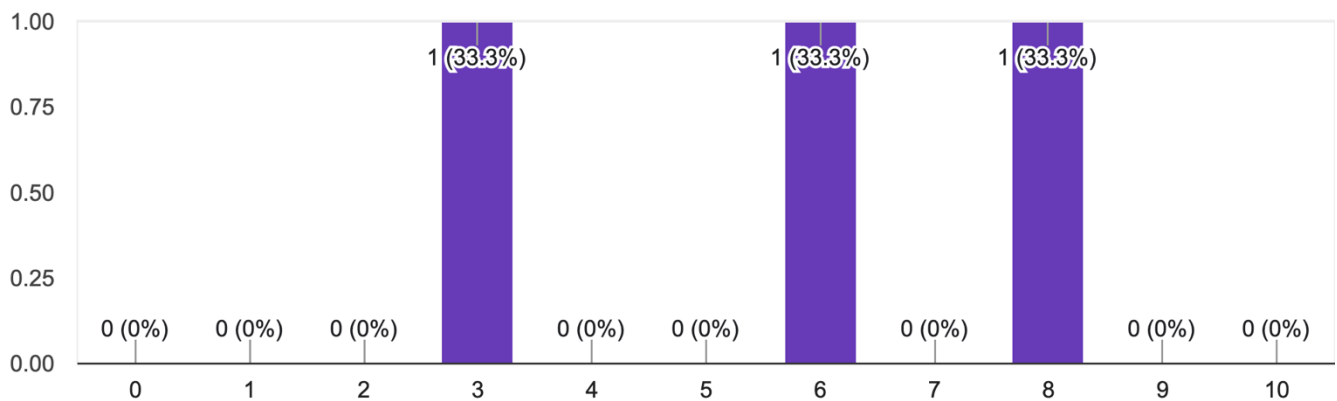
How much would you rate Avengers Endgame?

3 responses



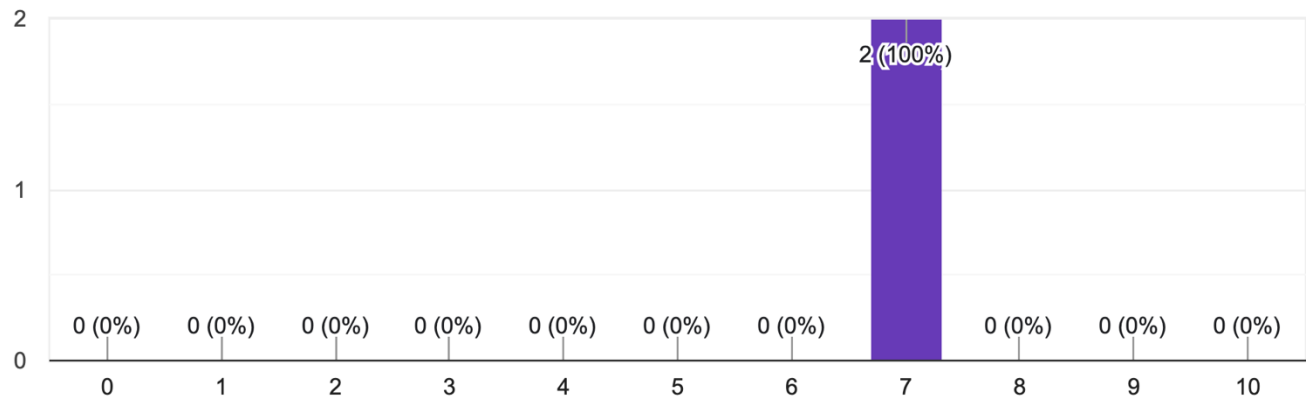
How much would you rate Home Alone?

3 responses



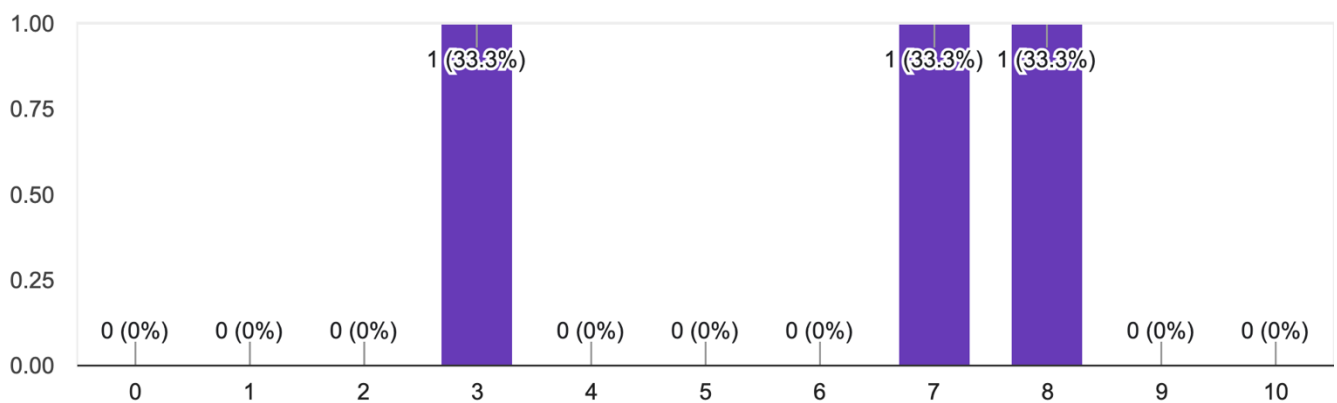
How much would you rate La La Land?

2 responses



How much would you rate The Conjuring?

3 responses



How much would you rate Harry Potter: Goblet of Fire?

3 responses

