Movie Recommender System

Introduction

A recommendation system or recommendation engine is a model used for information filtering where it tries to predict the preferences of a user and provide suggestions based on these preferences. These systems have become increasingly popular nowadays and are widely used today in areas such as movies, music, books, videos, clothing, restaurants, food, places, and other utilities. These systems collect information about a user's preferences and behavior, then use this information to improve their suggestions in the future.

Movies can be easily differentiated through their genres comedy, thriller, animation, action, etc. Other ways to distinguish between movies can be either by releasing year, language, director, etc. Watching movies online, there are a number of movies to search for in our most liked movies. Movie Recommendation Systems help us to search our preferred movies among all of these different types of movies and hence reduce the trouble of spending a lot of time searching for our favorable movies. So, it requires that the movie recommendation system should be very reliable and should provide us with the recommendation of movies that are exactly the same or match our preferences.

A large number of companies are making use of recommendation systems to increase user interaction and enrich a user's shopping experience. Recommendation systems have several benefits, the most important being customer satisfaction and revenue. The movie Recommendation system is a very powerful and important system. But, due to the problems associated with a pure collaborative approach, movie recommendation systems also suffer from poor recommendation quality and scalability issues.

The ALS algorithm is one of the models of matrix factorization related to CF which is considered as the values in the item list of the user matrix. As there is a need to perform analysis on the ALS algorithm by selecting different parameters which can eventually help in building an efficient movie recommender engine.

Problem Statement

Some refer to the current period as the "era of abundance." As a result, thousands of options can be available for any product. Consider the following instances: social networking, online shopping, streaming videos, and more. Recommender systems assist in personalizing a platform and finding content. People always need more time with the myriad tasks they need to accomplish in the limited 24 hours. Therefore, recommendation systems are essential as they help them make the right choices without spending their cognitive resources.

From a business perspective, user engagement is higher the more relevant products they discover on the platform. Increasing platform revenue is a common outcome of this. Various sources claim that as much as 35–40% of the revenue of tech behemoths comes from just recommendations. To suggest the most well-liked items is the quickest and most straightforward way to accomplish this. However, we require specialized recommender systems to improve the user experience through personalized recommendations. The main focus of our recommendation system is to filter and predict only those movies that a user would prefer, given some data about the user**.**

Literature Review

In recent years, recommender systems have become widespread and are utilized in various fields. For example, some general applications incorporate music, books, movies, research papers, social labels, and items in general. Similarly, the journal recommendation system has also drawn considerable attention from the research community, which creates and disseminates books, patents, and research articles. Recommender system algorithms are widely used in e-commerce to offer more personalized and precise recommendations to online users. The recommender system based on Hadoop provides a solution to the information overload issue in e-commerce by combining the advantage of computational ability and scalability of MapReduce and hybrid recommendation algorithms. The algorithms either focus on the users, finding the nearest neighbors of a target user and making recommendations to the target user with his neighbors' purchases or preferences or focus on the products, recommending items similar to the items already purchased by the users. Some algorithms provide personalized recommendations to target users, while others make general recommendations. The commonly used algorithms are Collaborative Filtering, Content-based Filtering, and User Clustering Models.

DATASETS USED

We'll use the free MovieLens dataset from GroupLens for our own system. This dataset has 100K data points from different movies and users. This dataset (ml-25m) describes a 5-star rating and free-text tagging activity from MovieLens, a movie recommendation service. It contains

- 25,000,095 user ratings from 162,541 users (between 1995 and 2019)
- 62423 rated movies
- Table for linking MovieLens identifiers with IMDb and TMDb identifiers

Users were selected at random for inclusion. All selected users had rated at least 20 movies. No demographic information is included. An id represents each user, and no other information is provided. The data are contained in the files genome-scores.csv, genome-tags.csv, links.csv, movies.csv, ratings.csv, and tags.csv. This and other GroupLens data sets are publicly available for download at http://grouplens.org/datasets/.

userId	movieId	rating	timestamp
1	296	5.0	1147880044
1	306	3.5	1147868817
1	307	5.0	1147868828
1	665	5.0	1147878820
1	899	3.5	1147868510
1	1088	4.0	1147868495
1	1175	3.5	1147868826
1	1217	3.5	1147878326
1	1237	5.0	1147868839
1	1250	4.0	1147868414
1	1260	3.5	1147877857
1	1653	4.0	1147868097
1	2011	2.5	1147868079
1	2012	2.5	1147868068
1	2068	2.5	1147869044
1	2161	3.5	1147868609
1	2351	4.5	1147877957
1	2573	4.0	1147878923
1	2632	5.0	1147878248
1	2692	5.0	1147869100

userId	movieId	tag	timestamp
3	260	classic	1439472355
3	260	sci-fi	1439472256
4	1732	dark comedy	1573943598
4	1732	great dialogue	1573943604
4	7569	so bad it's good	1573943455
4	44665	unreliable narrators	1573943619
4	115569	tense	1573943077
4	115713	artificial intell...	1573942979
4	115713	philosophical	1573943033
4	115713	tense	1573943042
4	148426	so bad it's good	1573942965
4	164909	cliche	1573943721
4	164909	musical	1573943714
4	168250	horror	1573945163
4	168250	unpredictable	1573945171
19	2160	Oscar (Best Suppo...	1446909853
19	7099	adventure	1445286141
19	7099	anime	1445286127
19	7099	ecology	1445286153
19	7099	fantasy	1445286144

movieId	imdbId	tmdbId
1	114709	862
2	113497	8844
3	113228	15602
4	114885	31357
5	113041	11862
6	113277	949
7	114319	11860
8	112302	45325
9	114576	9091
10	113189	710
11	112346	9087
12	112896	12110
13	112453	21032
14	113987	10858
15	112760	1408
16	112641	524
17	114388	4584
18	113101	5
19	112281	9273
20	113845	11517

<table>
<tr><th>movieId</th><th>title</th><th>genres</th></tr>
<tr><td>1</td><td>Toy Story (1995)</td><td>Adventure|Animati...</td></tr>
<tr><td>2</td><td>Jumanji (1995)</td><td>Adventure|Childre...</td></tr>
<tr><td>3</td><td>Grumpier Old Men ...</td><td>Comedy|Romance</td></tr>
<tr><td>4</td><td>Waiting to Exhale...</td><td>Comedy|Drama|Romance</td></tr>
<tr><td>5</td><td>Father of the Bri...</td><td>Comedy</td></tr>
<tr><td>6</td><td>Heat (1995)</td><td>Action|Crime|Thri...</td></tr>
<tr><td>7</td><td>Sabrina (1995)</td><td>Comedy|Romance</td></tr>
<tr><td>8</td><td>Tom and Huck (1995)</td><td>Adventure|Children</td></tr>
<tr><td>9</td><td>Sudden Death (1995)</td><td>Action</td></tr>
<tr><td>10</td><td>GoldenEye (1995)</td><td>Action|Adventure|...</td></tr>
<tr><td>11</td><td>American Presiden...</td><td>Comedy|Drama|Romance</td></tr>
<tr><td>12</td><td>Dracula: Dead and...</td><td>Comedy|Horror</td></tr>
<tr><td>13</td><td>Balto (1995)</td><td>Adventure|Animati...</td></tr>
<tr><td>14</td><td>Nixon (1995)</td><td>Drama</td></tr>
<tr><td>15</td><td>Cutthroat Island ...</td><td>Action|Adventure|...</td></tr>
<tr><td>16</td><td>Casino (1995)</td><td>Crime|Drama</td></tr>
<tr><td>17</td><td>Sense and Sensibi...</td><td>Drama|Romance</td></tr>
<tr><td>18</td><td>Four Rooms (1995)</td><td>Comedy</td></tr>
<tr><td>19</td><td>Ace Ventura: When...</td><td>Comedy</td></tr>
<tr><td>20</td><td>Money Train (1995)</td><td>Action|Comedy|Cri...</td></tr>
</table>

Techniques

Popularity-based recommendation Filtering

Popularity-based recommendation systems work with the trend by using the items which are in trend right now. With respect to our project, a movie featured in the recommendation chart if the movie scores higher than at least 90% of the movies in the dataset. The score for each movie was calculated using IMDB's weighted average formula shown below.

weighted rating (WR) = (v ÷ (v+m)) × R + (m ÷ (v+m)) × C

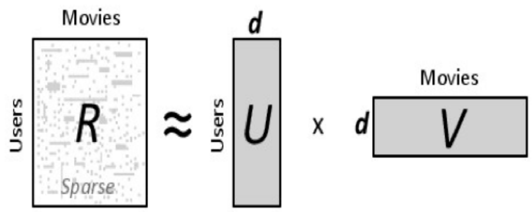
Where:

- R = average for the movie (mean) = (Rating)
- v = number of votes for the movie = (votes)
- m = minimum votes required to be listed in the Top 250 (currently 25,000)
- C = the mean vote across the whole report

Collaborative Filtering

Collaborative filtering technique allows filtering out items that a user might like by leveraging the ratings of similar users. The underlying assumption in recommendation using collaborative filtering is that, if user A and user B share a similar response (movie rating in our case) to a movie, then they are likely to share a similar response to any movie X, compared to any random user.

- Employed the model-based system of performing collaborative filtering on the MovieLens dataset.
- Implemented Alternating Least Square(ALS) with Spark. ALS is a matrix factorization technique to perform collaborative filtering. The objective function of ALS uses L1 regularization and optimizes the loss functions using Gradient Descent.
- The dataset contained movie_id and user_ratings in the format of a user-rating matrix shown as factors as given below:



Here, d would be the number of features we learned from each user and movie association. With ALS, we intend to minimize the error in the matrix calculation shown below:

$$\hat{R} = U \times V$$

And the error is given by the below equation:

$$RSS = \sum (R - U \times V)^2$$

We train the ALS model by tuning the below hyper-parameters:

- Rank: Indicating the number of latent factors generated in the matrix factorization
- regParam: The L1-regularization parameter used in ALS algorithm
- maxIter: The maximum number of iterations the algorithm is run

After tuning the parameters and implementing ALS with Cross-validation an optimal RMSE value of **0.8037 for 30 latent factors at the regParam value of 0.05 in 10 iterations.**

Results

Popularity-Based Model

We used group by function to select the movies which have the highest number of ratings. And then took the average ratings for each movie. Out of the shortlisted set of movies, we pulled the top 20 movies which would be recommended to new users based on the Avg. movie rating.

movieId	title	average
111	Taxi Driver (1976)	4.083479216009336
47	Seven (a.k.a. Se7...	4.0791663372598626
28	Persuasion (1995)	4.03
232	Eat Drink Man Wom...	4.027935943060498
17	Sense and Sensibi...	3.948806325713417
1	Toy Story (1995)	3.893707794587238
6	Heat (1995)	3.854908898649748
223	Clerks (1994)	3.847214137214137
41	Richard III (1995)	3.843413033286451
16	Casino (1995)	3.8237068028689416
154	Beauty of the Day...	3.786777431637245
101	Bottle Rocket (1996)	3.7673501577287065
96	In the Bleak Midw...	3.7004504504504503
235	Ed Wood (1994)	3.664388396868181
53	Lamerica (1994)	3.6496350364963503
40	Cry, the Beloved ...	3.6267252195734003
26	Othello (1995)	3.606708513142409
34	Babe (1995)	3.6031917599186163
230	Dolores Claiborne...	3.588200326943082
211	Browning Version,...	3.584600760456274

only showing top 20 rows

Collaborative Model

We used the ALS model to predict the movies which have the highest number of ratings given by a user, with **hyperparameters maxIter set to 4, rank set to 30, and regParam as 0.1**. The image below highlights the recommendations of 5 different movies based on a collaborative filtering approach.

movieId	userId	rating	title	genres
177209	5	6.3395796	Acı Aşk (2009)	Drama
194434	5	6.009604	Adrenaline (1990)	(no genres listed)
151989	5	5.8460293	The Thorn (1971)	Comedy
202231	5	5.768565	Foster (2018)	Documentary
200930	5	5.7514424	C'est quoi la vie...	Drama

User-Based

```
+-------+--------------------+
|movieId|     recommendations|
+-------+--------------------+
|      1| [{89631, 5.296187}]|
|      3|[{96471, 4.7021756}]|
|      5|[{131545, 4.711781}]|
|      6|[{156252, 5.16390...|
|      9|[{87426, 4.8842115}]|
+-------+--------------------+
```

Item-Base

Possible Future Work

There are plenty of ways to expand on the work done in this project. Firstly, the content-based method can be expanded to include more criteria to help categorize the movies. The most obvious idea is to add features to suggest movies with common actors, directors, or writers. In addition, movies released within the same time period could also receive a boost in the likelihood of a recommendation. Similarly, the movie's total gross could be used to identify a user's taste in terms of whether he/she prefers large-release blockbusters or smaller indie films. However, the above ideas may lead to overfitting, given that a user's taste can be highly varied, and we only have a guarantee that 20 movies (less than 0.2%) have been reviewed by the user. In addition, we could try to develop hybrid methods that try to combine the advantages of both content-based methods and collaborative filtering into one recommendation system.

References

1. Verma, J. P., Patel, B., & Patel, A. (2015). Big data analysis: Recommendation system with Hadoop framework. In 2015 IEEE International Conference on Computational Intelligence & Communication Technology (CICT). IEEE.
2. Katarya, R., & Verma, O. P. (2016). A collaborative recommender system enhanced with particle swarm optimization technique. Multimedia Tools and Applications, 75(15), 9225–9239.
3. https://docs.databricks.com/_static/notebooks/cs100x-2015-introduction-to-big-data/module-5–machine-learning-lab.html.
4. Wei, J., et al. (2016). Collaborative filtering and deep learning based hybrid recommendation for cold start problem. In 2016 IEEE 14th International Conference on

Dependable, Autonomic and Secure Computing, 14th International Conference on Pervasive Intelligence and Computing, 2nd International Conference on Big Data Intelligence and Computing and Cyber Science and Technology Congress (DASC/PiCom/DataCom/CyberSciTech). IEEE.

5. Kupisz, B., & Unold, O. (2015). Collaborative filtering recommendation algorithm based on Hadoop and Spark. In 2015 IEEE International Conference on Industrial Technology (ICIT). IEEE.
6. Zeng, X., et al. (2016). Parallelization of latent group model for group recommendation algorithm. In IEEE International Conference on Data Science in Cyberspace (DSC). IEEE.
7. Ponnam, L. T., et al. (2016). Movie recommender system using item based collaborative filtering technique. In the International Conference on Emerging Trends in Engineering, Technology, and Science (ICETETS). IEEE.
8. Halder, S., Sarkar, A. M. J., & Lee, Y.-K. (2012). Movie recommendation system based on movie swarm. In 2012 Second International Conference on Cloud and Green Computing (CGC). IEEE.
9. Dev, A. V., & Mohan, A. (2016). Recommendation system for big data applications based on set similarity of user preferences. In the International Conference on Next Generation Intelligent Systems (ICNGIS). IEEE.
10. Chen, Y.-C., et al. (2016). User behavior analysis and commodity recommendation for point-earning apps. In 2016 Conference on Technologies and Applications of Artificial Intelligence (TAAI). IEEE. 294 M. F. Aljunid and D. H. Manjaiah
11. Zhou, Y. H., Wilkinson, D., & Schreiber, R. (2008). Large scale parallel collaborative filtering for the Netflix prize. In Proceedings of 4th International Conference on Algorithmic Aspects in Information and Management (pp. 337–348). Shanghai: Springer.
12. https://spark.apache.org/docs/latest/. Accessed March 10, 2017.
13. https://grouplens.org/datasets/movielens/. Accessed May 15, 2017.
14. Delgado, J. A. (2000, February). Agent-based information filtering and recommender systems on the internet (Ph.D. thesis). Nagoya Institute of Technology.
15. Mooney, R. J., & Roy, L. (1999). Content-based book recommendation using learning for text categorization. In Proceedings of the Workshop on Recommender Systems: Algorithms and Evaluation (SIGIR '99). Berkeley, CA, USA.