**Analysis of the Pumpkinmeter score at Ripe Pumpkins**

**OMSBA 5315, Data Translation Challenge, Seattle University**

**March 12, 2022**

**Christine Rich, Tamlyn Tamura, Ian Ceazar Basco, Glen Lewis, Andrew Nalundasan**

**Use Case Names**

1. Andrew\_User1
2. Christine\_User2

**Use Case URL**: https://github.com/ChristineRich/OMSBA-5315-DTC

**Introduction**

Ripe Pumpkin executives have noticed the success of recommendation engines in the streaming industry and hope to successfully implement its own metric to measure ranking and similarity, called the Pumpkinmeter score. To do this, they used a data set called MovieLens, collected by GroupLens, and tested the recommender algorithm using two volunteers (named Andrew\_User1 and Christine\_User2) to measure how well Ripe Pumpkin’s recommendation system accurately suggested similar content based on the movies they reported that they enjoyed.

Each volunteer’s top 10 movie picks generated 10 additional recommendations. Figure 1 in the Appendix illustrates each volunteer’s movie picks. Each volunteer’s use case was measured against two datasets—a collection of all the movies with 25 reviews or more, and a more precise collection of all the movies with 100 reviews or more.

**Dataset Used**

As part of the data reporting team, we used two datasets, which are *movies* and *ratings* datasets. GroupLens Research operates a movie recommender MovieLens, which is the source of the data. The dataset files are written as comma-separated values, CSV, files with a single header row and can be downloaded by visiting the [MovieLens website](http://grouplens.org/datasets/movielens/).

Information about the movie is contained in the file, *‘movies.csv*’.This data contains the *movieID*, *title* and *genres*. Certain movies belong to multiple genres, and they are formatted as *genre1, genre2, genre3*, etc. (Figure 2).

All the movie ratings are contained in the file, *`ratings.csv`*. Each line of this file represents one rating of one movie by one user, and has the following format *userId, movieId, rating,* and *timestamp*. Ratings are evaluated based on a 5-star scale, with half-star increments (0.5 stars - 5.0 stars). Lastly, we dropped the *timestamp* column because this information is not relevant to our recommender.

**Technical Details**

The Pumpkinmeter score implements content-based filtering by using parallel computing concepts. Unlike popular streaming services, such as Netflix and YouTube, which utilize collaborative filtering that recommends content based on clustering user data, the Pumpkinmeter score is based on content-based filtering that recommends movies based on clustering movie attributes, such as genre. Ripe Pumpkins intends to implement collaborative filtering in the future but will need to perform further research on users to sharpen the recommendation system with this enhanced method.

This algorithm implements parallel computing/parallel processing using Spark RDD. This method enables computations to be broken down and calculated simultaneously. Parallel computing has become an increasingly popular method when it comes to designing and developing AI-based models, such as recommenders, due to its ability to process large amounts of data quickly given its algorithmic functionality and using a computer’s RAM.

This algorithm also utilizes the Spark MLlib library to build the recommender model. This library provides a collaborative filtering implementation by using Alternating Least Squares (ALS). ALS is the model utilized to fit our data and discover similarities. The idea is basically to take a large matrix and factor it into some smaller representation of the original matrix.

**Debugging Details**

We encountered one main challenge during our coding process. We received the error referenced in Figure 3 during our initial run. The first debugging activity was to separate the movies RDD join statements. We joined the movie titles, created a new variable, then joined the movie ratings count. For reference, the screenshots in Figure 4 are the original code (top) and the revised code (bottom). We also discovered that when reading the complete ratings data, each token needed to have a data conversion. After implementing the conversion, our code ran successfully and returned movie recommendation outputs. To take a closer look at this code change, please refer to Figure 5 in the Appendix.

**Results**

Figures 6 and 7 in the Appendix show a summary of Andrew’s movie picks and the recommendations returned. The most noticeable attribute of Andrew’s list of recommendations was that there were many repeat franchise recommendations. Six different ‘Doctor Who’ movies were found on the list for both Scenario 1 and Scenario 2. Each list comes out to be 40% dominated by related content. Similarly, 20% of the Scenario 1 list were from the ‘Harry Potter’ franchise, and 27% for Scenario 2. The Pumpkinmeter recommendation system appears to recommend movies that have similar attributes due to the dominance of related content included in each scenario. The more accurate scenario for Andrew’s list was Scenario 2. An accuracy rate of 53% was found as he has previously seen eight out of the fifteen movies recommended, indicating that the model has succeeded in recommending movies that he would be likely to accept.

Christine was less familiar with the movies listed in her recommendations. Figures 8 and 9 show a summary of her top movie picks and her movie recommendations. Quite a few period pieces and mini-series were recommended even though none were included in her user ratings. In Scenario 1, Christine would watch two of the fifteen recommended movies, leading to an accuracy of 13%. And for Scenario 2, one of fifteen movies sounded interesting to Christine, making the recommendations 7% accurate. This may indicate that including more user ratings for Christine could lead to a higher accuracy in recommendations.

There were two perspectives with the interpretation of results. First, Andrew’s accuracy was calculated by assessing if he has already previously seen the recommended movies from his list. Christine on the other hand, researched the recommended titles and determined if she would take up the recommendation from Ripe Pumpkins or not. The term “accuracy” can mean different things to different people, and it will take a concerted effort from the Ripe Pumpkin’s data science and marketing teams to determine the perspective of accuracy that the model should be achieving. The team has determined that the perspective of “accuracy” from Christine’s perspective will lead to higher success due to the focus on maximizing continued engagement by watching recommended content.

**Insights and Recommendations**

As discussed in the Results, the algorithm has shortcomings when it comes to recommendation accuracy. Here, we measured accuracy as movies previously seen and movies one would consider watching. Predicting a user has watched 53% of the recommended content is decent performance, but not exceptional. Also, when a user would only watch 13% of the recommended content indicates that the Pumpkinmeter score is not telling the whole story.

The Pumpkinmeter score relies heavily on content-based filtering, so while the score was impacted by movie ratings, our research suggests that movie ratings would have been more useful if it was qualitatively measured rather than quantitatively. It was found that having a soft mix of ratings ranging from 1, 2, 3, 4, and 5 with increments of 0.5 creates a blurred distinction to determine what is truly likable amongst users. We believe that if the ratings quantified more distinctly such as like, dislike and neutral, the Pumpkinmeter score would better accurately classify its users and recommend more diverse content. 60% of Andrew’s recommendations were from the same series (*Doctor Who*, *Harry Potter*), which is something that Ripe Pumpkins should avoid.

Regarding the usage of qualitative metrics, we also believe that providing an explanation in addition to the quantitative ratings will be able to validate the data further by applying sentiment analysis over the movie rating explanations. By applying sentiment analysis, the Pumpkinmeter score will be able to make a similar recommendation more accurately by associating specific words of a movie review as being different tiers of positive, negative, or neutral sentiments towards the movie.

If Ripe Pumpkins determines that it wants to stick with content-based filtering over collaborative filtering for now, then more data needs to be analyzed to explain why users, like Andrew and Christine, preferred certain recommended content over others. The reality is that people's taste in movies are influenced from a wide range of reasons, not just genre. Some people may like movies for the cast, soundtrack, or time period a movie was made. Perhaps some people only like one or two movies in a specific genre but dislike the rest, simply because they enjoyed the plot, animation style, or dialogue. Perhaps someone who likes animated movies enjoyed *Into the Spiderverse* yet refuse to watch the other "Spider-Man" films.

Lastly, as Andrew’s results reflect, many of his recommended movies came from the same movie franchise (*Doctor Who*, *Harry Potter*). This is not very effective because, assuming Andrew had not seen *Doctor Who* or *Harry Potter*, then it would make the most sense to start at the beginning of the franchise, as opposed to the middle or last movie. By revising the code to only display the first movie in a movie franchise as a recommendation, the Pumpkinmeter score can use those extra “spots” for more diverse recommendations.

The Pumpkinmeter model is on the right path but needs time to further enhance its recommendation algorithm for Ripe Pumpkins to successfully compete amongst the major players in today’s market such as Netflix, Hulu, Peacock, and HBO. A partnership between the data science and marketing teams will help Ripe Pumpkins to refine the Pumpkinmeter model to maximize engagement with customers, enabling them to compete in this space.

**References**

Chatterjee, P., Bhattacharyya, S., & Ramnani, M. (2021, August 24). *Collaborative filtering*

*vs content-based filtering for Recommender Systems*. AIM. Retrieved March 9, 2022,

from [https://analyticsindiamag.com/collaborative-filtering-vs-content-based-filtering-](https://analyticsindiamag.com/collaborative-filtering-vs-content-based-filtering-for-recommender-systems/)

for-recommender-systems/

freeCodeCamp.org. (2018, November 14). *How to use spark clusters for parallel processing Big Data*. freeCodeCamp.org. Retrieved March 9, 2022, from <https://www.freecodecamp.org/news/how-to-use-spark-clusters-for-parallel-processing-big-data-86a22e7f8b50/>

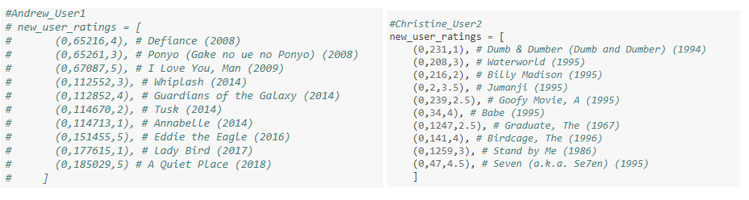
Google. (n.d.). *Content-based filtering | recommendation systems | google developers*. Google. Retrieved March 9, 2022, from <https://developers.google.com/machine-learning/recommendation/content-based/basics#:~:text=Content%2Dbased%20filtering%20uses%20item,previous%20actions%20or%20explicit%20feedback>.

Sunith Shetty (2018, September 6). *How to build a cold-start friendly content-based recommender using Apache Spark SQL*. Packt Hub. Retrieved March 9, 2022, from <https://hub.packtpub.com/how-to-build-a-cold-start-friendly-content-based-recommender-using-apache-spark-sql/>

*What is parallel computing? definition and faqs: OmniSci*. What is Parallel Computing? Definition and FAQs | OmniSci. (n.d.). Retrieved March 9, 2022, from <https://www.heavy.ai/technical-glossary/parallel-computing>

**Appendix**

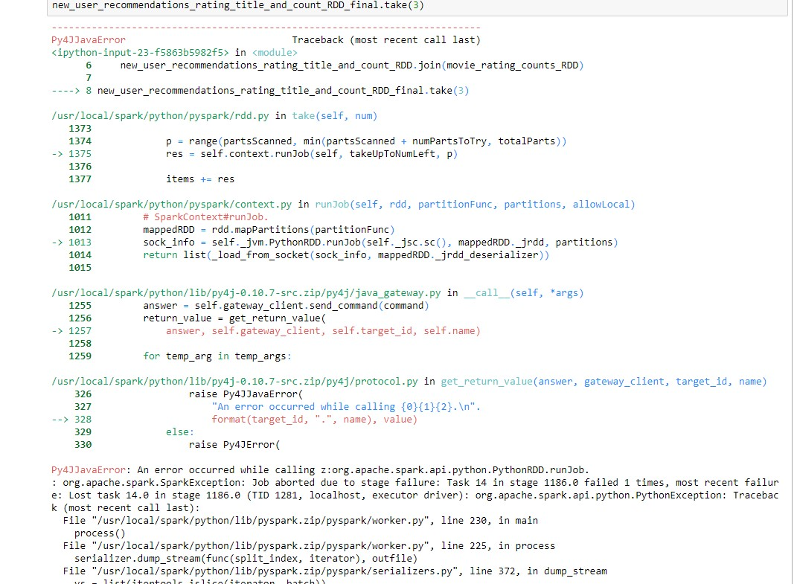
**Figure 1**



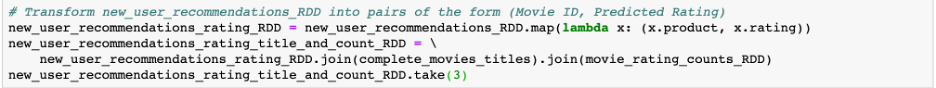
**Figure 2**

|  |  |
| --- | --- |
| *movies.csv* | *ratings.csv* |
| Dataset fields:   * *movieID* * *title* * *genres* | Dataset fields:   * *userId* * *movieId* * *rating* * *timestamp (d*ropped) |

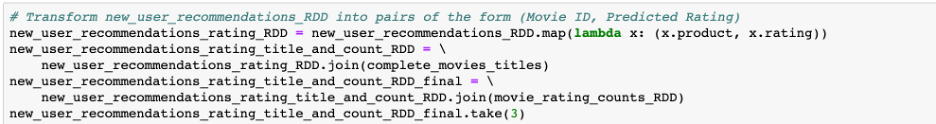
**Figure 3**



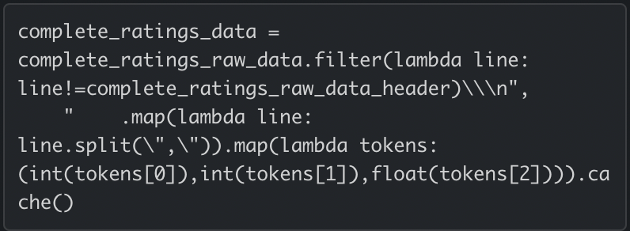
**Figure 4**







**Figure 5**



**Figure 6**

Andrew\_User1 movie recommendation

**Scenario 1**: Top 15 recommended movies with more than 25 reviews

|  |  |  |
| --- | --- | --- |
| Movie recommendation | Ratings | Number of reviews |
| Doctor Who: Voyage Of The Damned (2007) | 4.40 | 224 |
| Doctor Who: The Doctor | 4.34 | 179 |
| The Butterfly Circus (2009) | 4.32 | 28 |
| Cats (1998) | 4.32 | 25 |
| Doctor Who: Last Christmas (2014) | 4.30 | 228 |
| Harry Potter and the Deathly Hallows: Part 2 (2011) | 4.28 | 13262 |
| Doctor Who: A Christmas Carol (2010) | 4.27 | 271 |
| Doctor Who: The Husbands of River Song (2015) | 4.26 | 239 |
| Bridegroom (2013) | 4.26 | 36 |
| Avengers: Infinity War - Part I (2018) | 4.26 | 2668 |
| Harry Potter and the Deathly Hallows: Part 1 (2010) | 4.25 | 14055 |
| Doctor Who: The Time of the Doctor (2013) | 4.25 | 394 |
| Prayers for Bobby (2009) | 4.23 | 102 |
| Harry Potter and the Half-Blood Prince (2009) | 4.23 | 14115 |
| 13 reasons why | 4.22 | 129 |

**Figure 7**

Andrew\_User1 movie recommendation

**Scenario 2**: Top 15 recommended movies with more than 100 reviews

|  |  |  |
| --- | --- | --- |
| Movie recommendation | Ratings | Number of reviews |
| Doctor Who: Voyage Of The Damned (2007) | 4.40 | 224 |
| Doctor Who: The Doctor | 4.34 | 179 |
| Doctor Who: Last Christmas (2014) | 4.30 | 228 |
| Harry Potter and the Deathly Hallows: Part 2 (2011) | 4.28 | 13262 |
| Doctor Who: A Christmas Carol (2010) | 4.27 | 271 |
| Doctor Who: The Husbands of River Song (2015) | 4.26 | 239 |
| Avengers: Infinity War - Part I (2018) | 4.26 | 2668 |
| Harry Potter and the Deathly Hallows: Part 1 (2010) | 4.25 | 14055 |
| Doctor Who: The Time of the Doctor (2013 | 4.25 | 394 |
| Prayers for Bobby (2009) | 4.23 | 102 |
| Harry Potter and the Half-Blood Prince (2009) | 4.23 | 14115 |
| 13 reasons why | 4.22 | 129 |
| Love | 4.21 | 576 |
| Harry Potter and the Order of the Phoenix (2007) | 4.19 | 14349 |
| Sherlock - A Study in Pink (2010) | 4.19 | 213 |

**Figure 8**

Christine\_User2 movie recommendation

**Scenario 1:** Top 15 recommended movies with more than 25 reviews

|  |  |  |
| --- | --- | --- |
| Movie recommendation | Ratings | Number of reviews |
| Very Potter Sequel | 4.41 | 35 |
| Anne of Green Gables: The Sequel | 4.35 | 342 |
| Drishyam (2013) | 4.39 | 37 |
| Sense & Sensibility (2008) | 4.36 | 69 |
| North & South (2004) | 4.35 | 389 |
| Anne of Green Gables (1985) | 4.29 | 706 |
| Cranford (2007) | 4.28 | 35 |
| Pride and Prejudice (1995) | 4.27 | 2919 |
| Murder on the Orient Express (2010) | 4.23 | 29 |
| I Can Only Imagine (2018) | 4.08 | 30 |
| Winter in Prostokvashino (1984) | 4.06 | 67 |
| Boys (2014) | 4.23 | 96 |
| Little Dorrit (2008) | 4.22 | 55 |
| Vacations in Prostokvashino (1980) | 4.03 | 96 |
| Runaway Brain (1995) | 4.02 | 30 |

**Figure 9**

Christine\_User2 movie recommendation

**Scenario 2:** Top 15 recommended movies with more than 100 reviews

|  |  |  |
| --- | --- | --- |
| Movie recommendation | Ratings | Number of reviews |
| Anne of Green Gables: The Sequel | 4.35 | 342 |
| North & South (2004) | 4.35 | 389 |
| Anne of Green Gables (1985) | 4.29 | 706 |
| Pride and Prejudice (1995) | 4.27 | 2919 |
| Sound of Music | 4.10 | 17154 |
| Schindler's List (1993) | 4.11 | 71516 |
| Shawshank Redemption | 4.11 | 97999 |
| Wild China (2008) | 4.08 | 105 |
| Hidden Figures (2016) | 4.11 | 2647 |
| Emma (2009) | 4.08 | 385 |
| Civil War | 4.05 | 431 |
| It's a Wonderful Life (1946) | 3.89 | 17770 |
| Piper (2016) | 4.08 | 1253 |
| Jane Eyre (2006) | 3.89 | 327 |
| Persuasion (2007) | 4.06 | 349 |