Content-Based Recommendation System Developed from the MovieLens 10M Dataset

Nicelle Sernadilla Macaspac

June 2023

Introduction

The Big Tech companies use recommendation systems to provide customized suggestions of products to users (Koren 2008, 426; Rocca 2019). These algorithms aim to enhance user experience and build customer loyalty (Koren 2008, 426). They are commonly developed using content-based methods founded on user and product information and collaborative filtering methods based on user-product interactions (Rocca 2019).

In 2006, the streaming service company Netflix offered USD 1,000,000 to any team that managed to come up with an algorithm that improved on their internal recommendation system, Cinematch, by 10% or more. The hybrid team of KorBell, Big Chaos and Pragmatic Chaos - Bell-Kor's Pragmatic Chaos - triumphed the challenge (Van Buskirk 2009). The team utilized advanced methods of collaborative filtering, restricted Boltzmann machine and gradient-boosted decision trees, yet Koren (2008, 434; 2009, 9) of team KorBell emphasized the significance of content information from the baseline predictors in capturing the main effects in the dataset and refining the prediction of the algorithm.

Following this lead, this project aims to develop a modest recommendation system that is based on the content information from the baseline predictors in a similar dataset of the movie recommender MovieLens and that predicts future movie ratings of users with a root mean squared error rate of less than 0.86490. In the succeeding sections, we examine the dataset and user preferences then subsequently build algorithms to develop the recommendation system.

This undertaking is part of the capstone in the Professional Certificate Program in Data Science of Harvard Online.

MovieLens 10M Dataset

The MovieLens 10M Dataset was composed of 10,000,054 movie ratings from users with 20 ratings or more (GroupLens, n.d.; Harper and Konstan 2015). The raw dataset was downloaded and subjected to wrangling using a code in the language R provided by Harvard Online, which standardized its formatting and partitioning into the edx set and the final holdout test set across all projects.

The edx set was used to examine user preferences and to train and test content-based algorithms to develop the recommendation system. It contained 9,000,055 movie ratings, each with 6 variables: userId, movieId, rating, timestamp, title and genres (fig. 1). UserIDs and movieIDs were unique to each of its 69,878 users and 10,677 movies, respectively. Ratings ranged from 0.5 to 5 stars. Timestamps were rendered in Unix time. Titles were captured manually according to how they appear in the movie database IMDb and included the year of release (GroupLens, n.d.).

On the other hand, the final holdout test set was reserved to evaluate the recommendation system. It contained 999,999 movie ratings, each with the aforementioned 6 variables. We circle back to this set in a later section.

userId	movieId	rating	timestamp	title	genres
1	122	5	838985046	Boomerang (1992)	Comedy|Romance
1	185	5	838983525	Net, The (1995)	Action|Crime|Thriller
1	292	5	838983421	Outbreak (1995)	Action|Drama|Sci-Fi|Thriller
1	316	5	838983392	Stargate (1994)	Action|Adventure|Sci-Fi
1	329	5	838983392	Star Trek: Generations (1994)	Action|Adventure|Drama|Sci-Fi

Figure 1: First rows of the edx set.

User Preference

The interactions of the variables of the edx set were visualized using the ggplot2 functions to examine user preferences.

A plot of user ratings against movieIds of a random sample of the data showed an aggregration of the ratings on a number of movieIds (fig. 2). This indicated the tendency of users to rate certain movies more than others and consequently imparted which movies are popular.

Figure 2: Scatterplot of user ratings vs movieIds of a random sample of the edx set.

Similarly, a plot of user ratings against genres of a random sample of the data exhibited accummulation of the ratings on genres such as Comedy and Drama (fig. 3), signifying the preference of users for certain genres more than others.

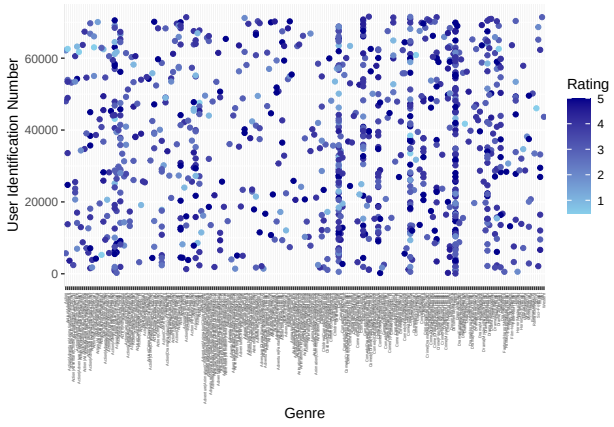

Figure 3: Scatterplot of user ratings vs genres of a random sample of the edx set.

Titles were not unique to each movie unlike movieIds and were therefore not good identifiers. However, they contained important information: the years of release. Hence, the years were extracted from the ends of

the titles utilizing the dplyr and stringr functions, and a plot of user ratings against the years of a random sample of the data was produced (fig. 4). This revealed the inclination of users to movies from the 1990s and 2000s more than those from other years.

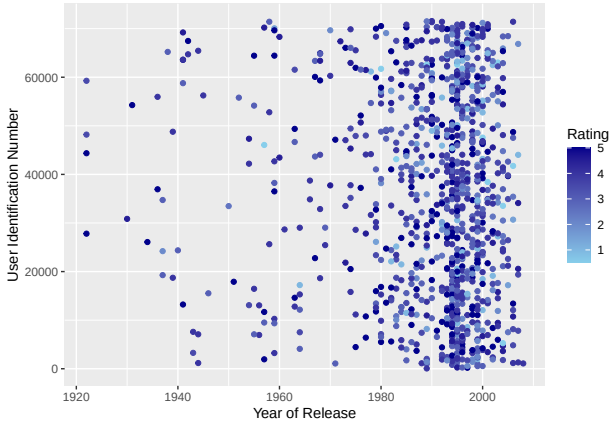

Figure 4: Scatterplot of user ratings per year of release of a random sample of the edx set.

A simple bar chart of average user ratings of a random sample of the data showed the tendencies for certain users to give a liberal average rating of 5. On the other hand, some users gave a conservative average rating of 1.

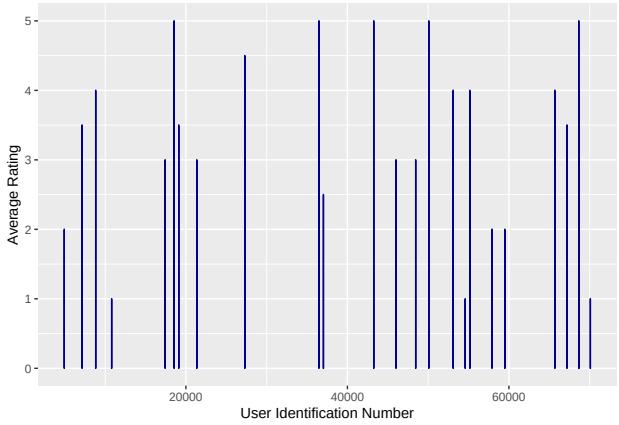

Figure 5: Bar chart of average user ratings of a random sample of the edx set.

Content-Based Algorithm

baseline predictors

$E = mc^2$

<TABLE 1> (Madhugiri 2022)

"In practice, we often report the root mean squared error (RMSE), because it is in the same units as the outcomes." (Irizarry 2022)

"if our recommender system is based on a model that outputs numeric values such as ratings predictions or matching probabilities, we can assess the quality of these outputs in a very classical manner using an error measurement metric such as, for example, mean square error (MSE)." (Rocca 2019)

"Explainability is another key point of the success of recommendation algorithms. Indeed, it has been proven that if users do not understand why they had been recommended a specific item, they tend to lose confidence in the recommender system." (Rocca 2019)

"In order to combat overfitting the sparse rating data, models are regularized so estimates are shrunk towards baseline defaults. Regularization is controlled by constants, which are denoted as: lambda1, lambda2, . . . " (Koren 2008, 427)

"The regularizing term lambda avoid overfitting by penalizing magnitudes of the parameters." (Koren 2009, 2)

Recommendation System

Conclusion

"In this paper, we suggested methods that lower the RMSE to 0.8870."

"However, they are entered manually, so errors and inconsistencies may exist." (GroupLens, n.d.)

<LIMIT: RECURRING TIME>

"in others our success is restricted by the randomness of the process, with movie recommendations for example." (Irizarry 2022)

Reference

GroupLens, n.d. "README.txt." Accessed May 29, 2023. https://grouplens.org/datasets/movielens/10m/.

Harper, F. Maxwell, and Joseph A. Konstan. 2015. "The MovieLens Datasets: History and Context." *ACM Transactions on Interactive Intelligent Systems* 5, no. 4: 1-19. https://doi.org/10.1145/2827872.

Irizarry, Rafael A. 2002. *Introduction to Data Science: Data Analysis and Prediction Algorithms with R.* http://rafalab.dfci.harvard.edu/dsbook/.

Koren, Yehuda. 2008. "Factorization Meets the Neighborhood: a Multifaceted Collaborative Filtering Model." https://people.engr.tamu.edu/huangrh/Spring16/papers_course/matrix_factorization.pdf.

Koren, Yehuda. 2009. "The BellKor Solution to the Netflix Grand Prize." https://www2.seas.gwu.edu/~simhaweb/champalg/cf/papers/KorenBellKor2009.pdf.

Madhugiri, Devashree. 2022. "Top 7 Packages for Making Beautiful Tables in R." https://towardsdatascience.com/top-7-packages-for-making-beautiful-tables-in-r-7683d054e541.

Rocca, Baptiste. 2019. "Introduction to Recommender Systems." https://towardsdatascience.com/introduction-to-recommender-systems-6c66cf15ada.

Van Buskirk, Eliot. 2009. "BellKor's Pragmatic Chaos Wins $1 Million Netflix Prize by Mere Minutes." https://www.wired.com/2009/09/bellkors-pragmatic-chaos-wins-1-million-netflix-prize/.