SCHOOL OF COMPUTER SCIENCES

UNIVERSITI SAINS MALAYSIA

CDS513: Predictive Business Analytics

Semester 2, 2020/2021

ASSIGNMENT 1:

C2_TOPIC 8: MOVIE RECOMMENDER SYSTEMS BASED ON PERSONALITY DATA AND MOVIE GENRES

LEE YONG MENG

(P-COM0012/20)

Date of Submission

26 May 2021

Table of Contents

1. Introduction

Recommender system is a subfield of information filtering system widely used in many applications in the business world. Some big online service providers implementing recommender systems in their services are Amazon online stores and Netflix for recommending items and movies to their customers, respectively. Other examples of recommender systems also include news article recommendations and friend suggestions in the social media. In simplest terms, recommender systems work by first predicting the user preferences towards items or contents in the system. Then, the predicted preferences are sorted and those items or contents with the highest predicted preferences are generated as lists of recommended items to their users.

For many decades, recommender system has been actively studied by many researchers. There are many approaches of implementing a recommender system, including collaborative filtering and content-based approaches. Collaborative filtering approaches utilizes the past feedback, whether implicitly or explicitly given, to generate a list of recommended items to the target users. There is a strong assumption that the target users will share the same interest on unseen items with other users with similar tastes who have seen those items in the past. Whereas for content-based approach, the recommendation is based on the attributes of the items. The system generates recommendations by matching users' interest with the attributes of the items, for example, the genres of different movies. There are also researches that attempts to incorporate user personality traits into the recommender systems to improve the personalized experience provided by the system. In recent years, recommender systems are also implemented using more advanced techniques such as machine learning and deep learning.

In this assignment, the main objective is to explore different approaches, including but not limited to collaborative filtering and content-based approaches, to implement movie recommender systems and then to examine and evaluate the performance of each recommender systems in recommending movies to different users in the system. Another objective of this assignment is to utilize the information of user personality traits and movie genre to construct movie recommender systems which can generate more robust movie recommendations to users in the system. The implementation of these recommender systems is performed on RapidMiner Studio version 9.9 (the latest version to-date), a data mining software for performing many data science tasks including machine learning and predictive analytics.

1.1. Problem Background

In the age of the Internet and smartphones, the growth of digital data around the globe has gained lots of attention from many people. Some factors contributing to such phenomenon include the advances of the communication technology and the improvement in data processing capabilities (Gnanasundaram & Shrivastava, 2012). This phenomenon has changed human lifestyle in many ways, including the way humans spend their leisure time. From the user viewpoint, many online business and services are growing and being made available to the public with just a few swipes and taps on the screen of their smartphones. Undoubtedly, this has brought a lot of convenience to human and makes human life more efficient by saving cost and time. However, there might be times when users are not able to make a quick decision when there are too many choices being offered by an online service. This phenomenon is referred to as information overload (Konstan & Riedl, 2012).

Information overload is one key problem when a user browses through the Internet to search for something that they might be interested. When there is an overwhelming amount of information reaching the users, users might be stressed out for not being able to process all that information quickly. In the context of an online movie streaming service, when users are flooded with a torrent of information about different movies, they might end up not making any movie selections at all due to the fear of making wrong choices of movies to watch. Therefore, a solution that provides personalized experience comes into play to handle such issue.

The most ideal solution is to implement a system to the online movie streaming service which can understand and accurately predicts users' preferences and tastes across different movies. With this information, the system can narrow down the choice of movies available in the system by generating a list of recommendation to the users. This system is known as recommender system (or sometimes recommendation systems), which enables users to speed up their decision making while improving the quality of their decision (Isinkaye, Folajimi, & Ojokoh, 2015). Other benefits of implementing a recommender system in an online movie streaming service also include the improved revenues from the business point of view.

1.2. Dataset and Description

There are two types of dataset used in this assignment: the primary dataset and the secondary dataset. The primary dataset used to accomplish the problem statement as stated in the previous section is the Personality 2018 dataset[1]. This dataset is initially collected by GroupLens, a research lab at the University of Minnesota to study about the correlation between user personality trait and their preferences towards a list of movies. This dataset consists of two CSV files, which are the "personality-data.csv" and "ratings.csv". These files store two sets of structured data, namely the personality and ratings data, respectively. The personality data consists of 1835 records, each corresponding to the user's personality trait and a list of 12 movies with their predicted ratings. On the other hand, the ratings data consists of more than 1 million ratings of more than 35 thousand movies given by the same users in the personality data from 1997 to 2019. Tables 1.1 and 1.2 summarize the description of the personality and ratings data respectively.

Table 1.1: Description of the personality data.

[1] Personality 2018 dataset: https://grouplens.org/datasets/personality-2018/

Attribute name	Attribute type	Attribute description
userid	Categorical data	User identifier consisting of 32 random characters in hexadecimal (base of 16).
openness	Numerical data	7-Point Likert scale indicating the degree of different aspects (OCEAN: openness, conscientiousness, extraversion, agreeableness, neuroticism) in the Big 5 personality trait (emotional_stability indicates the reverse of the “neuroticism” aspect).
agreeableness	Numerical data	
emotional_stability	Numerical data	
conscientiousness	Numerical data	
extraversion	Numerical data	
assigned_metric	Categorical data	Metric used to generate list of 12 movies to collect user feedback, possible values include: “all”, “popularity”, “diversity”, and “serendipity”.
assigned_condition	Categorical data	Degree of “assigned_metric” in generating list of 12 movies, possible values include: “default” (only when “assigned_metric” is “all”), “low”, “medium”, and “high”.
movie_*n*	Categorical data	ID of the *n*-th generated movie, where *n* = {1, 2, 3, …, 12}
predicted_rating_*n*	Numerical data	Predicted user rating of “movie_*n*”, where *n* = {1, 2, 3, …, 12}
is_personalized	Numerical data	5-Point Likert scale feedback from the user indicating the degree of personalization of the generated list of movies.
enjoy_watching	Numerical data	5-Point Likert scale feedback from the user indicating their preferences towards the generated list of movies.

Table 1.2: Description of the personality data.

Attribute name	Attribute type	Attribute description
useri	Categorical data	User identifier consisting of 32 random characters in hexadecimal (base of 16).
movie_id	Categorical data	Movie identifier ranging from 1 to 198117.
rating	Numerical data	5-Point Likert scale movie rating given by user.

Attribute name	Attribute type	Attribute description
tstamp	Date and time	Recorded date and time when user rates a movie, example: “2001-09-10 17:19:56”.

From the data description shown in Tables 1.1 and 1.2, the only information we know about the movie in the personality and ratings data is the unique movie IDs. This information is not sufficient for performing analysis and implementation of recommender systems which requires item attributes, for example, content-based recommender systems and hybrid recommender systems. Therefore, a secondary dataset with movie information is integrated with the primary dataset for more comprehensive analysis.

The secondary dataset used is the MovieLens 25M dataset[2]. This dataset consists of several CSV files. For this assignment, only the CSV file named “movies.csv” is used to supply the movie attributes and to perform analysis on the result. The movies data consists of 62,423 records of different movies. Table 1.3 summarizes the description of the movies data.

Table 1.3: Description of the movies data.

Attribute name	Attribute type	Attribute description
movieId	Categorical data	Movie identifier ranging from 1 to 209171
title	Categorical data	String containing title and release year of the movie.
genres	Categorical data	Movie genres, a movie might have zero or multiple genres, different genres are separated by the pipe “|” character.

The original attributes of these three sets of data are not ready to be used as the input to implement different types of recommender systems. Furthermore, some attributes are not relevant for the analysis. The data preprocessing steps are briefly described in the next section.

1.3. Data Preprocessing

To better adapt the three sets of data used in this assignment, some steps are performed to preprocess these data using Python version 3.8.3 on into the format suitable for implementing recommender systems in RapidMiner. As a result, the number of records in the ratings data has reduced from 1,028,751 to 120,601 after data preprocessing steps. The preprocessed ratings data consists of only unique ratings since 2016, by 925 users

[2] MovieLens 25M dataset: https://grouplens.org/datasets/movielens/

who have rated at least 20 movies, and 3,702 movies receiving at least 10 ratings. The ratings data is then split into training and test set with a proportion of 80:20.

Personality data and movies data are transformed into data with only two columns. In other words, the first column of the transformed data is the user or item identifier (namely: user ID and movie ID, respectively), whereas the second column is the attribute ID, which is required by RapidMiner to implement content-based recommender systems. After prerocessing, there are 9,416 and 4,625 records in the personality and movies data respectively. The data preprocessing steps is illustrated in Figure 1.1.

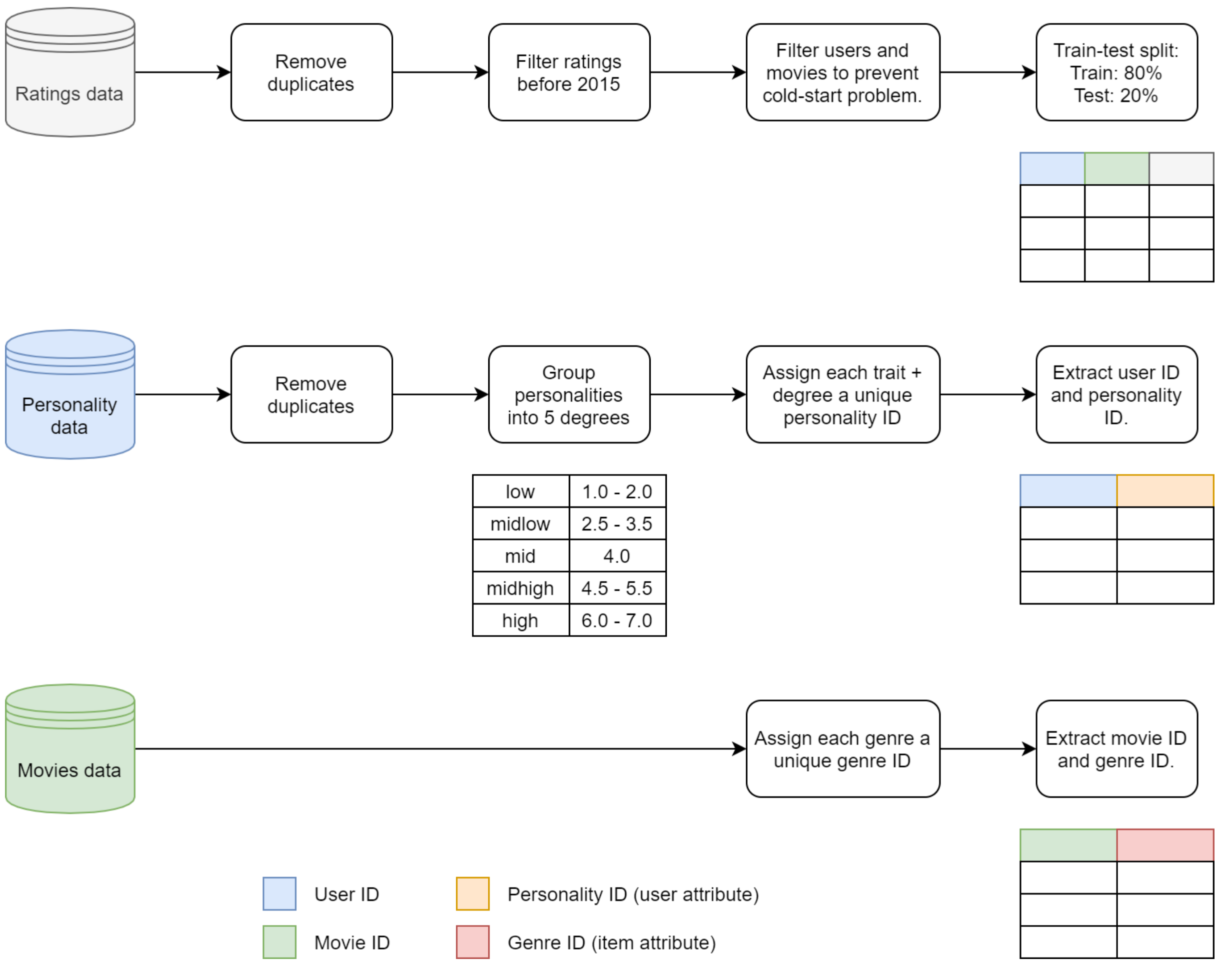

Figure 1.1: Data preprocessing steps to transform the data into the format suitable for implementing recommender systems.

2. Recommender Systems

Recommender systems (RS) are mainly used to generate a list of recommended items according to the ranking of target user’s predicted ratings or preferences towards items available in the system. In this assignment, the 3 main types of RS implemented to address the problem statement include collaborative filtering (CF) RS, content-based RS and hybrid RS. In simplest term, CF RS is one common type of RS used in the business domain. CF RS utilizes user feedbacks on the items in the system to recommend items to the target users. In this context, the user feedbacks are the movie ratings provided by the users in the system.

Content-based approach, on the other hand, is an alternative approach implemented in RS to recommend items to the target users. Unlike CF RS, content-based RS does not rely on user feedbacks to generate the list of recommended items. Instead, content-based RS uses the additional information of the user or item to find similarities between users or items liking certain items. This information is utilized by content-based RS to recommend items to the target users. In Hybrid RS, different approaches are combined to implement a more robust RS. Despite its increasing complexity, it is believed that hybrid RS can generate a more robust recommendations to the target user as it generates the recommendation list by utilizing information from different aspects.

In this chapter, the discussion mainly focuses on the implementation of different RS using RapidMiner. The discussion starts with the implementation of CF RS, followed by content-based and hybrid RS.

2.1. Collaborative Filtering Recommender System

Collaborative filtering (CF) RS is a common type of recommender system used in the business domain. The goal of a CF RS is to generate a list of recommended items ranked by the predicted ratings or preferences of items available in the system by the target user. This is done by utilizing the historical feedback data on these items by other users in the system. There are two main types of user feedback, namely implicit feedback and explicit feedback. In this assignment, the items are the movies with explicit feedback, the user ratings.

A CF RS requires the input of user ratings on a list of items (in this case, list of movies) in the form of a utility matrix. This utility matrix has m rows and n columns, where m is the number of users and n is the number of items in the dataset. In this assignment, the ratings data is mainly used for CF RS. The 3 different variations of CF RS implemented in this assignment are: User-to-User CF RS, Item-to-Item CF RS, and Context-Aware RS (CARS).

User-to-User Collaborative Filtering Recommender System

To implement User-to-User CF RS for movie recommendations in RapidMiner, the "User k-NN" operator is used in the RapidMiner main process. This operator receives input from the training set to generate a list of *k* most similar users to each user based on movie ratings given by the users in the system. The output of this operator, a model (or "Mod") is then passed to the "Apply Model" operator as one of the inputs (together with the test set) to generate list of movie recommendations most relevant to the target users based on the ratings given to each movie by the *k* most similar users to these target users.

There are two subprocesses used for this implementation, which are renamed as "Prepare Train" and "Prepare Test" respectively. These two subprocesses perform the same steps to assign "user identification" and "item identification" to the "user_id" and "movie_id" columns respectively. A "Multiply" operator is used in the "Prepare Test" subprocess to create two copies of the test set to be used for "Apply Model" and "Performance" operators in the main process. Figures 2.1, 2.2, and 2.3 illustrate the RapidMiner main process and subprocesses involved in the implementation of User-to-User CF RS.

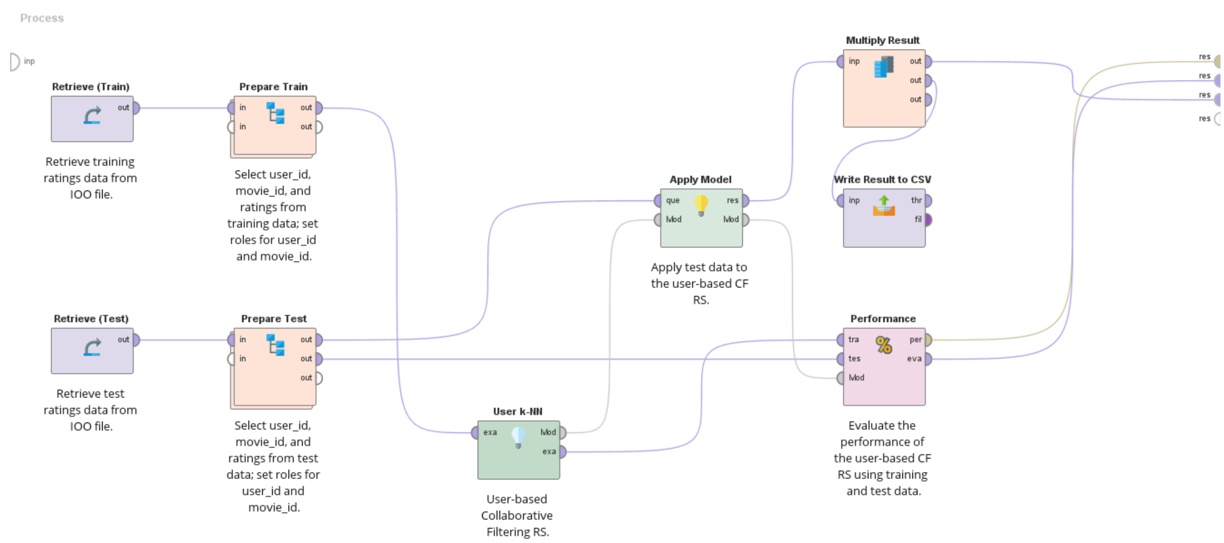

Figure 2.1: User-to-user Collaborative Filtering Recommender System.

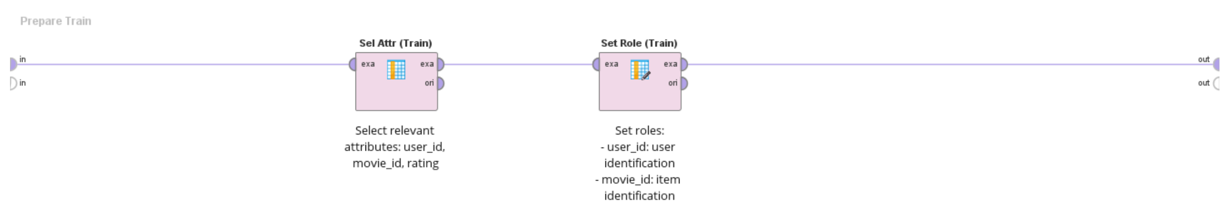

Figure 2.2: Process to prepare training data.

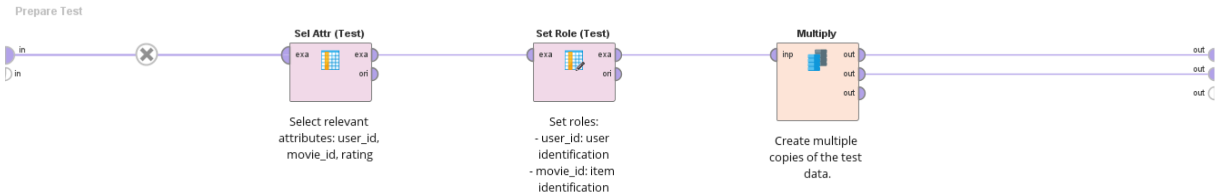

Figure 2.3: Process to prepare test data.

For the parameters of the "User k-NN" operator (see Figure 2.4), the values of *k* are repeated with 5 different values, which are 20, 40, 60, 80, and 100, respectively. There are two sets of performance results generated, for user k-NN and weighted user k-NN (by unchecking and checking the "weighted Knn" option), respectively.

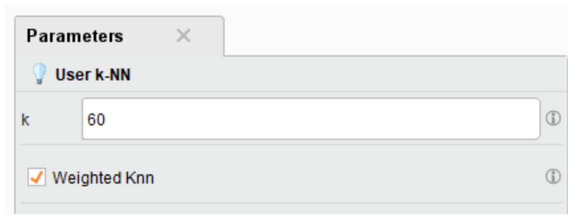

Figure 2.4: User k-NN: Parameters.

Item-to-Item Collaborative Filtering Recommender System

The implementation of Item-to-Item CF RS for movie recommendations in RapidMiner is almost like the User-to-User CF RS. The only difference is, the "Item k-NN" operator is used. This operator receives input from the training set to generate a list of *k* most similar movies to each target movie based on their ratings given by other users in the system. The output of this operator, a model (or "Mod") is also passed to the "Apply Model" operator as one of the inputs (together with the test set) to generate list of movie recommendations most relevant to the target users. The implementation of the subprocesses, namely the "Prepare Train" and "Prepare Test" subprocesses, are the same as previously implemented in the User-to-User CF RS. Figure 2.5 illustrates the RapidMiner main process involved in the implementation of Item-to-Item CF RS.

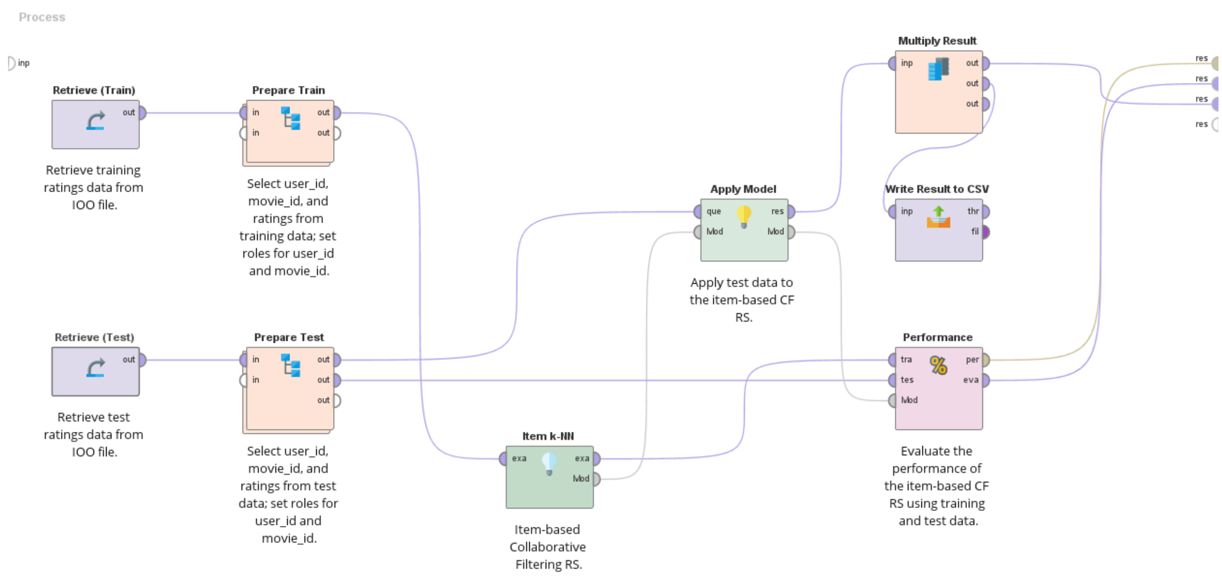

Figure 2.5: Item-to-Item Collaborative Filtering Recommender System.

For the parameters of the “Item k-NN” operator (see Figure 2.4), the values of *k* are also repeated with 5 different values, which are 20, 40, 60, 80, and 100, respectively. Again, there are two sets of performance results generated, for item k-NN and weighted item k-NN (by unchecking and checking the “weighted Knn” option), respectively.

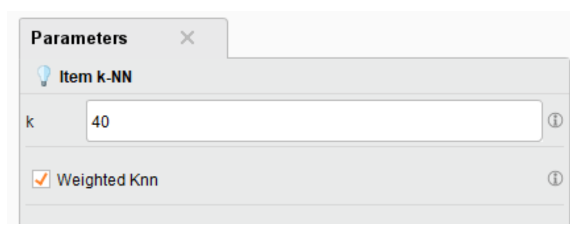

Figure 2.6: Item k-NN: Parameters.

Context-Aware Recommender System

In this assignment, an assumption is made when implementing CARS. A user will rate the movie differently during the weekdays and weekends. Therefore, the training and test sets used for building the CARS are filtered by keeping only the ratings given by users in the systems during weekdays or weekends depending on the settings. To implement this, a “Set Macro” operator is added to the main process (see Figures 2.7 and 2.8 for the main processes of User-to-User and Item-to-Item CARS respectively), followed by the “Select Subprocess” operator in the “Prepare Train” and “Prepare Test” subprocesses (see Figure 2.9) to filter different sets of examples (using “Filter Examples” operator, see Figure 2.10) from the training and test sets respectively.

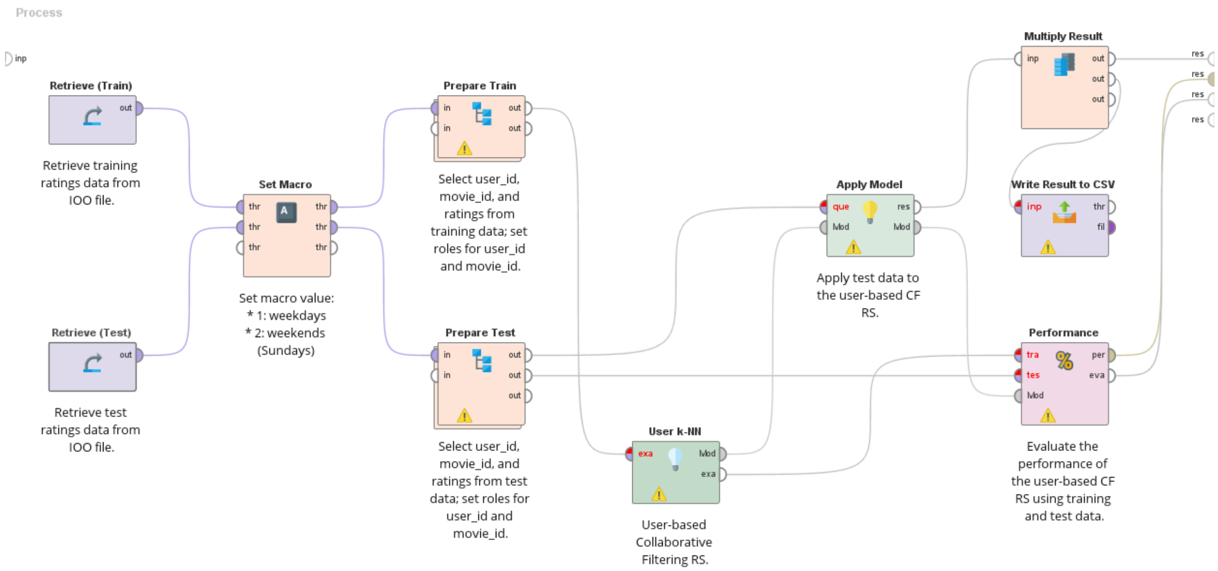

Figure 2.7: Context-Aware Recommender System (CARS) – User k-NN.

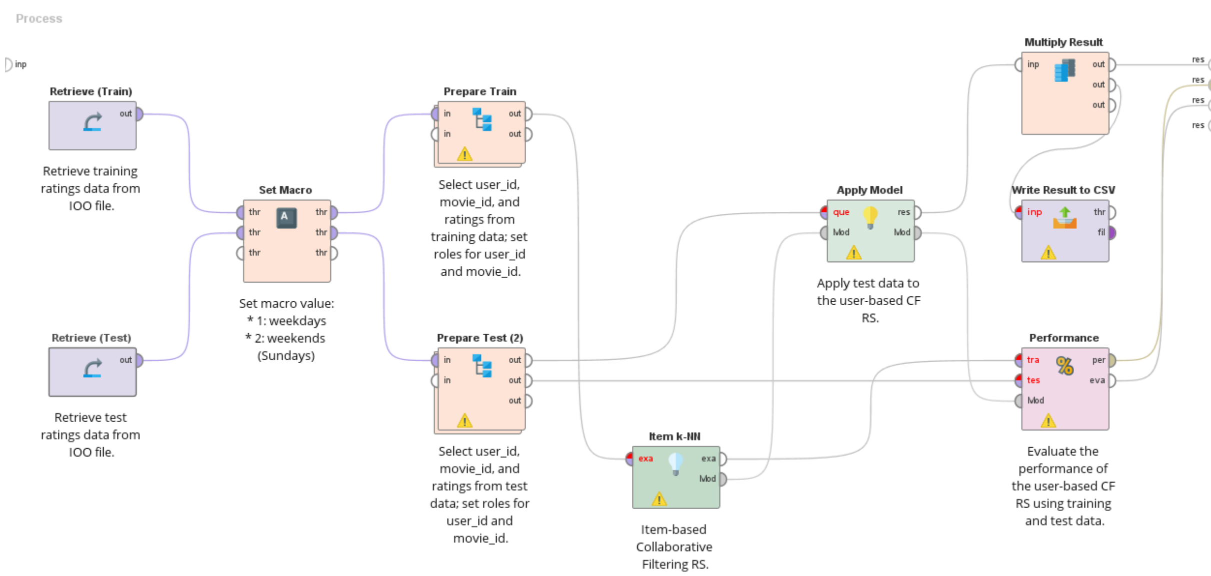

Figure 2.8: Context-Aware Recommender System (CARS) – Item k-NN.

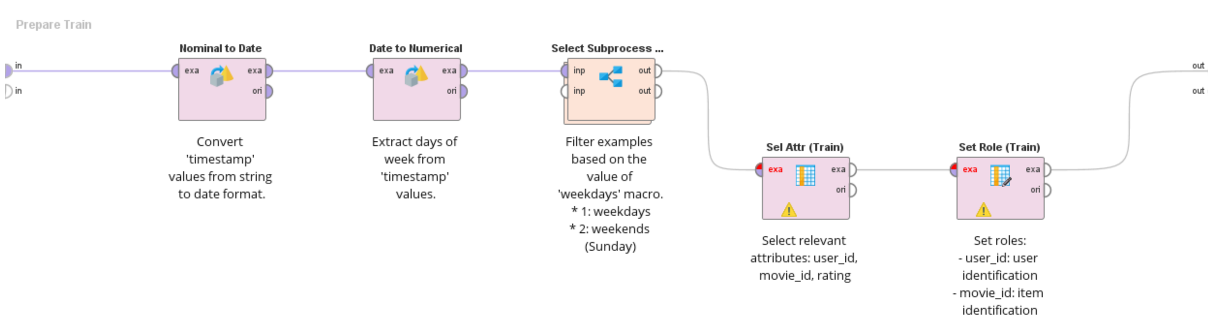

Figure 2.9: CARS: Process to prepare training data.

Figure 2.10: CARS: Filter Example operators under different subprocesses.

For the parameters of the “Set Macro” operator (see Figure 2.11), the “macro” value is set to “weekdays”, which is similar to the concept of defining a new variable named “weekdays” of Boolean type. The two possible values are 1 and 2, which indicates the instruction to keep records from “weekdays” and “weekends” only.

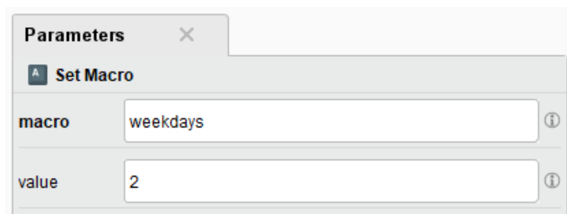

Figure 2.11: CARS: Set Macro parameters.

The “weekdays” macro is later used in the “Prepare Train” and “Prepare Test” subprocesses to select the subprocesses during the filtering process. The “Select Subprocess Train/Test” operator in Figure 2.9 takes in one parameter value, which is “select which”. The possible values of this parameter are from 1 to *n*, where *n* indicates the number of subprocesses implemented in the “Select Subprocess” operator. In this case, two subprocesses are implemented, as illustrated in Figure 2.10. Therefore, value of the “select which” parameter is set to follow the value of the “weekdays” macro, as illustrated in Figure 2.12.

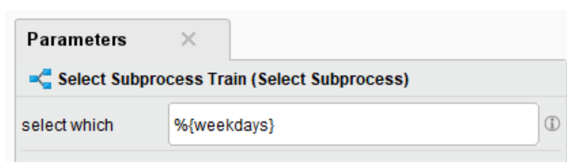

Figure 2.12: CARS: Select Subprocess parameters. The subprocess to run is set according to the value in the Set Macro operator.

Lastly, Figures 2.13 and 2.14 illustrates the settings of the “filters” parameter in the “Filter Examples” operator for subprocesses 1 and 2 respectively in Figure 2.10. The settings “timestamp ≠ 1” and “timestamp = 1” indicate the settings to keep records from weekdays and weekends, respectively. In the ratings data, the ratings are given only from Sundays to Fridays, in which, Sunday is the first day of the week. Therefore, another assumption is to treat Sundays from the ratings data (without Saturdays) as weekends. Therefore, “timestamp = 1” indicates the ratings from Sundays, in this case, the weekends.

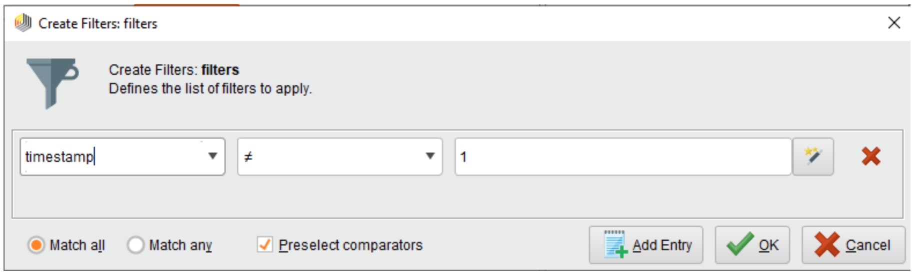

Figure 2.13: CARS: Filter by keeping examples with timestamps indicating weekdays only.

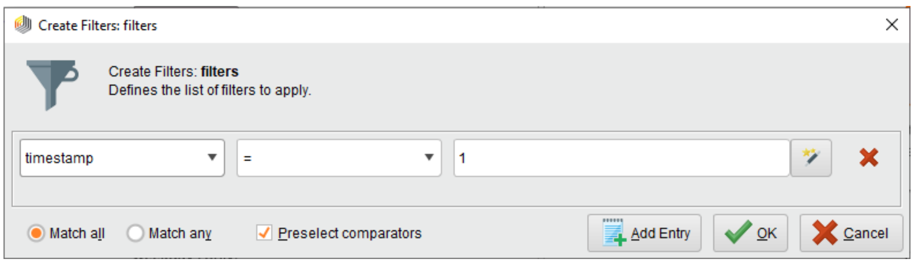

Figure 2.14: CARS: Filter by keeping examples with timestamps indicating weekends (Sundays) only.

2.2. Content-Based Recommender System

In RapidMiner, different types of content-based RS can be implemented using different operators implemented in the Recommender Extension. To construct attribute-based RS in RapidMiner, the 2 operators using the binomial item or user attributes to identify list of nearest neighbours during the item recommendation process are “Item Attribute k-NN” and “User Attribute k-NN” (Hofmann & Klinkenberg, 2017).

In this assignment, the user attributes are generated from the attributes indicating different aspects Big 5 Personality trait, namely: openness, conscientiousness, extraversion, agreeableness, and neuroticism, in the personality data. This attribute, together with the ratings data, are used to construct User Attribute-based RS in RapidMiner. On the other hand, item attributes, indicating different genres assigned to each movie, are adapted from the movies data, are used with the ratings data to construct Item Attribute-based RS in RapidMiner. The details of implementing different types of content-based RS are detailed in the following two subsections.

User Attribute-based Recommender System

To implement User Attribute-based RS for movie recommendation in RapidMiner, the “User Attribute k-NN” operator is used in the main process. This operator receives inputs from both the training set and the personality data – supplying the personality as user attributes. After that, the output of this operator, a model (or “Mod”) is passed to the “Apply Model” operator as one of the inputs (together with the test set) to generate list of movie recommendations most relevant to the target users by matching the user attributes to the list of movies in the test set. The implementation of the subprocesses, namely the "Prepare Train" and "Prepare Test" subprocesses, are the same as previously implemented in the CF RS from Section 2.1. Figure 2.15 illustrates the RapidMiner main process involved in the implementation of User Attribute-Based RS.

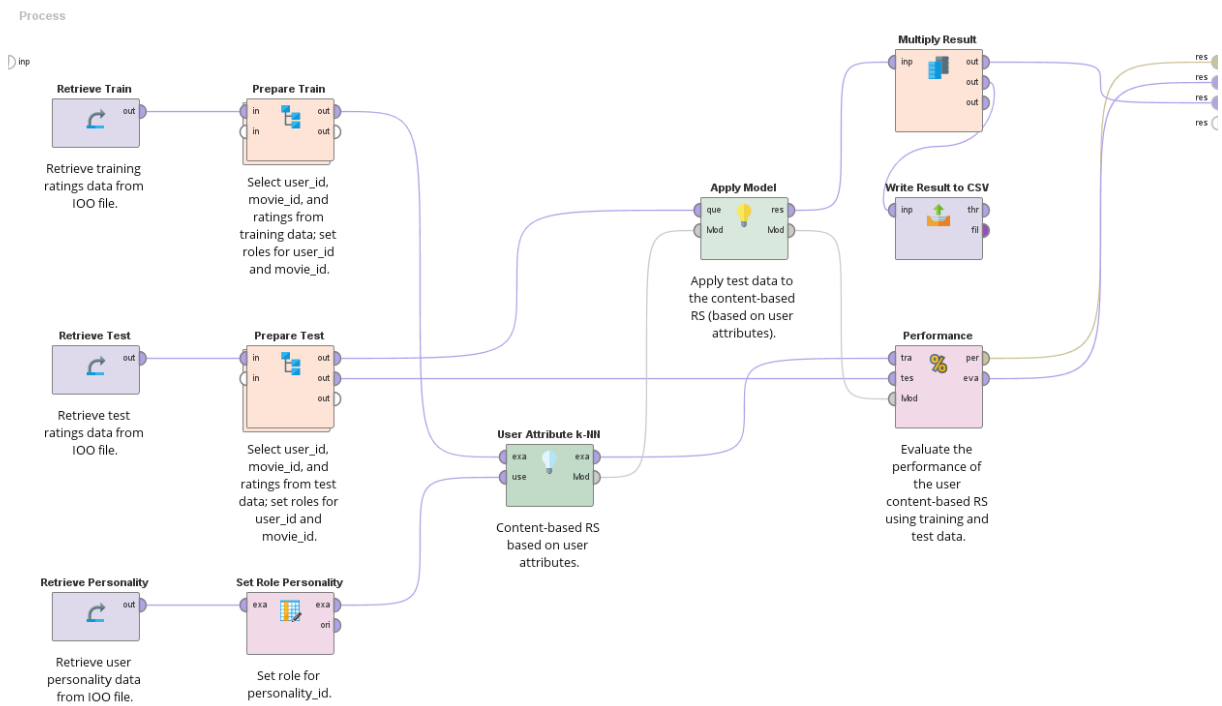

Figure 2.15: Content-Based Recommender System – User Attribute k-NN.

The only parameter of the "User Attribute k-NN" operator is *k* (see Figure 2.16), which is to set the value of *k* for the k-NN algorithm. The values of *k* are repeated with 5 different values, which are 20, 40, 60, 80, and 100, respectively.

Figure 2.16: User Attribute k-NN: Parameter.

Item Attribute-based Recommender System

The implementation of Item Attribute-based RS for movie recommendation in RapidMiner is similar to the User Attribute-based RS. In this case, the "Item Attribute k-NN" operator is used in the main process. This operator receives inputs from both the training set and the movie genre data as the item attributes. Again, the output of this operator, a model (or “Mod”) is passed to the “Apply Model” operator as one of the inputs (together with the test set). The list of movie recommendations to the target users is performed by matching the movie genres in the test set to the user preferences towards different movies in the past. The implementation of the subprocesses, namely the “Prepare Train” and “Prepare Test” subprocesses, are the same as previously implemented in the User Attribute-based RS. The main process in the implementation of Item Attribute-based RS is illustrated in Figure 2.17.

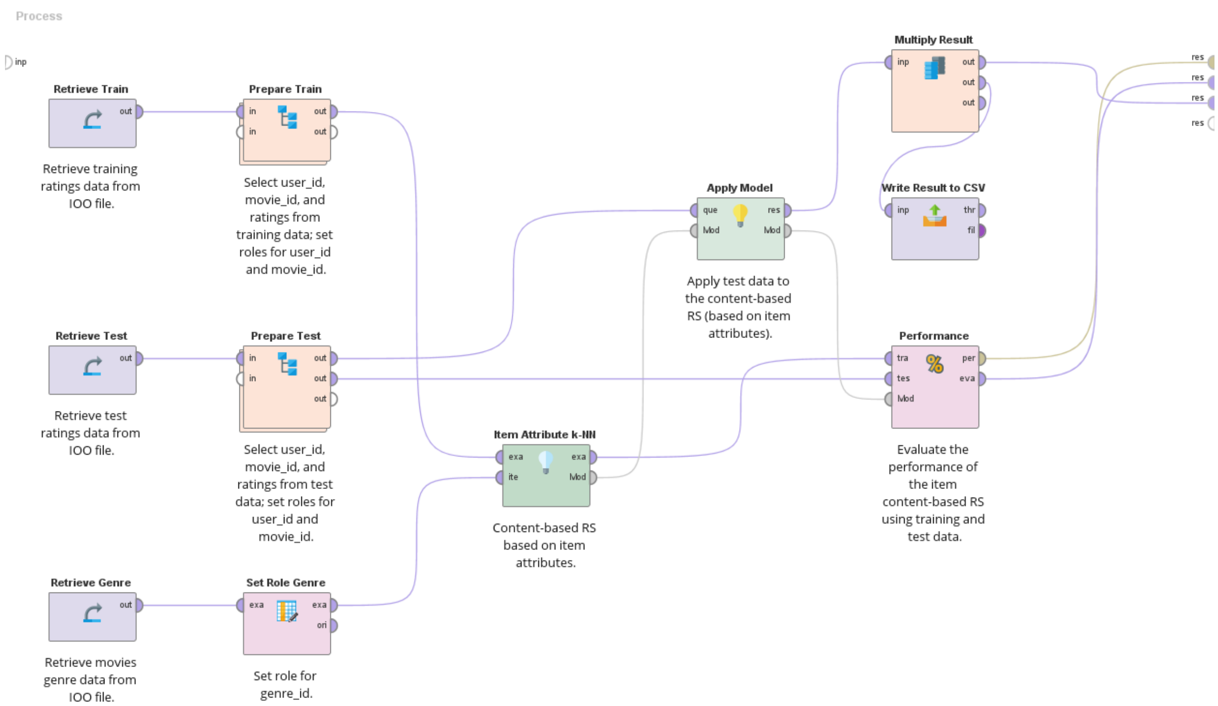

Figure 2.17: Content-Based Recommender System – Item Attribute k-NN.

Similar to the “User Attribute k-NN”, the parameter of the “Item Attribute k-NN” operator is also *k* (see Figure 2.18). The values of *k* are, again, repeated with 5 different values, which are 20, 40, 60, 80, and 100, respectively.

Figure 2.18: Item Attribute k-NN: Parameter.

2.3. Hybrid Recommender System

In RapidMiner, different types of RS can be combined using the “Model Combiner” operator to construct a hybrid RS. There are 2 types of hybrid RS implemented for this assignment for comparison, namely:

- Hybrid RS 1: Combining Item-to-Item CF and Item Attribute-based RS.
- Hybrid RS 2: Combining User-to-User CF and User Attribute-based RS.

Hybrid Recommender System 1: Combining Item-to-Item Collaborative Filtering and Item Attribute-based Recommender System

Hybrid RS 1 is implemented by combining the Weighted Item-to-Item CF and Item Attribute-based RS implemented in Sections 2.1 and 2.2 respectively. The main process is similar to those processes in the previously implemented RS in RapidMiner, as illustrated in Figure 2.19. The only difference is the introduction of a "Hybrid RS Model" processes.

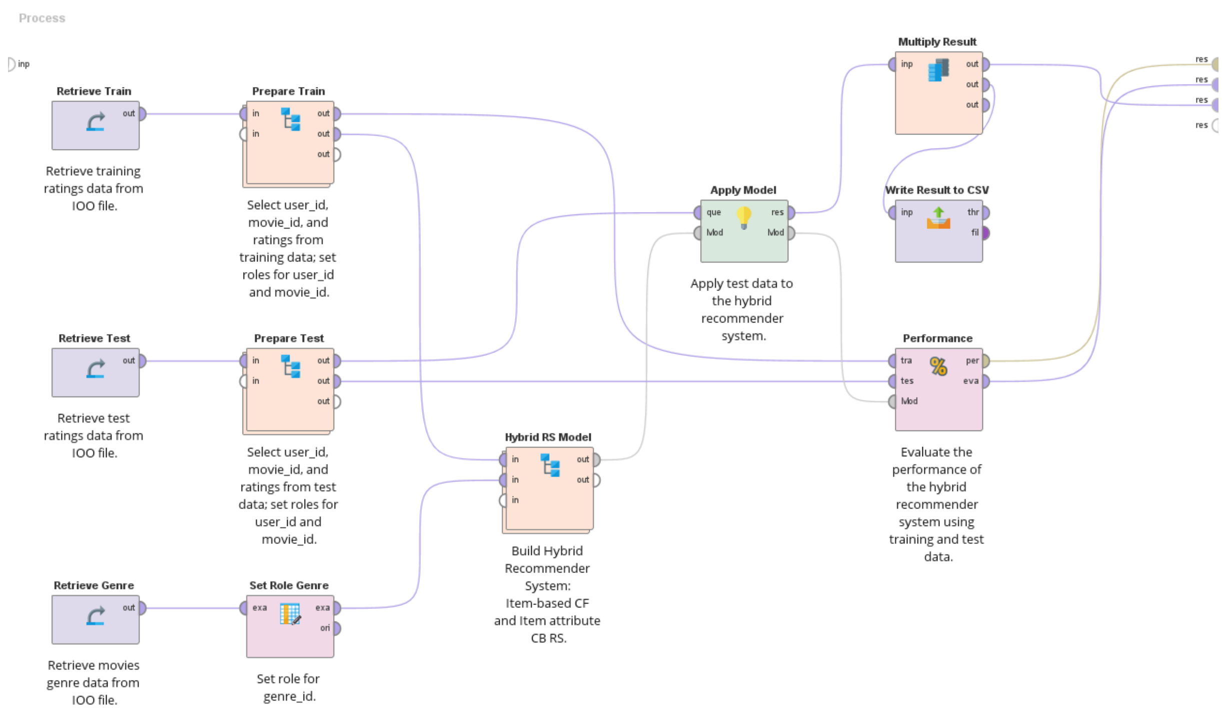

Figure 2.19: Hybrid Recommender System 1: Combining Item-to-Item Collaborative Filtering and Item Attribute-Based Recommender Systems.

The "Hybrid RS System" subprocess takes in the training set and movie genres data as input, passing these data into two distinct operators, namely "Item k-NN" and "Item Attributes k-NN" respectively, to create two different models. The two models are the Item-to-Item CF and Item Attribute-based RS, respectively. The "hybrid" component of the Hybrid RS comes into play when the two models are combined together using the "Model Combiner" operator, as illustrated in Figure 2.20. In this case, the parameter values of the "Model Combiner" operator are all set to default.

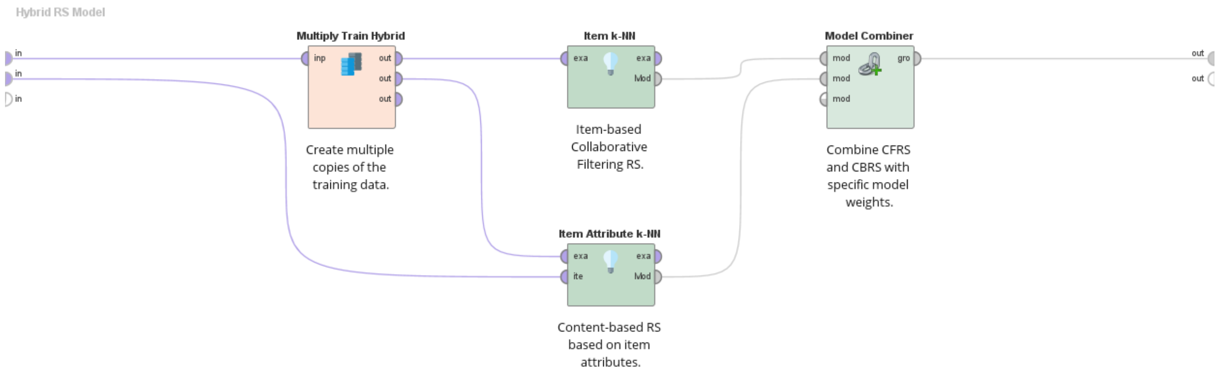

Figure 2.20: Hybrid Recommender System 1: Hybrid RS Model

Hybrid Recommender System 2: Combining User-to-User Collaborative Filtering and User Attribute-based Recommender System

Hybrid RS 2 is implemented in a similar manner as the Hybrid RS 1, even though two different RS, namely User-to-User CF and User Attribute-based RS are combined in this case. In this case, personality data is used to provide user attributes for the User Attribute-based RS. The main process and the “Hybrid RS Model” subprocess are illustrated in Figures 2.21 and 2.22 respectively.

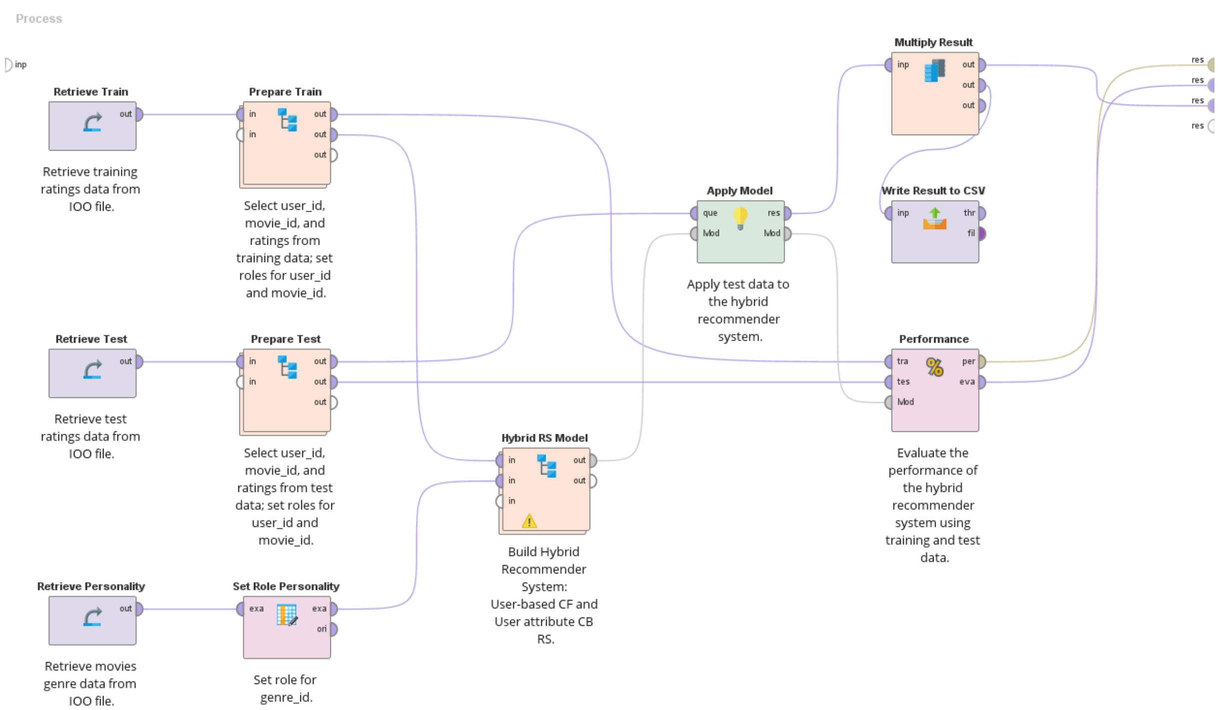

Figure 2.21: Hybrid Recommender System 2: Combining User-to-User Collaborative Filtering and User Attribute-based Recommender System.

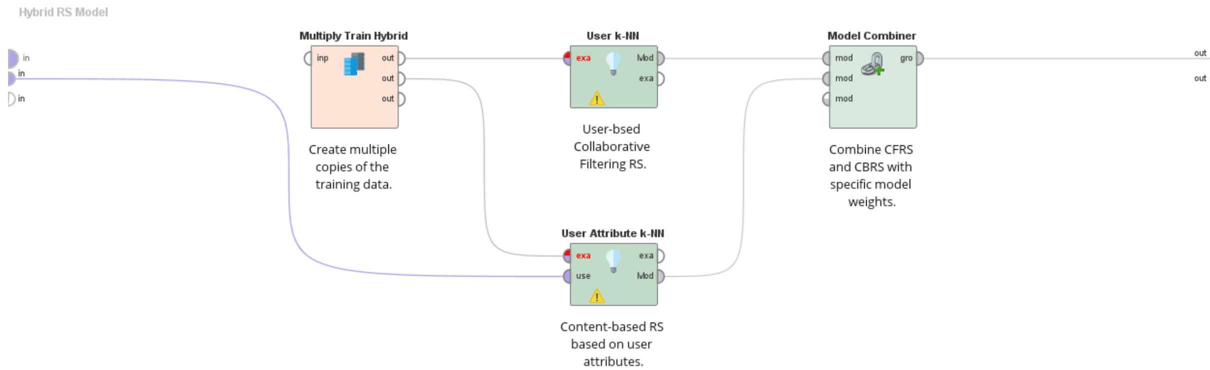

Figure 2.22: Hybrid Recommender System 2: Hybrid RS Model.

3. Evaluating the Performance of Recommender Systems

This chapter starts with the brief introduction of different evaluation metrics to measure the performance of RS, including the choice of metrics used to evaluate different RS implemented in this assignment with justifications. After that, the evaluation results of these RS are tabulated, discussed with some analysis is performed on the results.

3.1. Evaluation Metrics: AUC, Precision and Mean Average Precision

There are two main categories of evaluation metrics for RS, namely: the prediction accuracy metrics and the decision support metrics (Taifi, 2020). The prediction accuracy metrics can be used to evaluate RS that predicts the movie ratings by target users. Some examples include mean absolute error (MAE) and root mean squared error (RMSE). However, the RS implemented in this assignment is used to perform movie recommendations, which are more relevant to be evaluated using the decision support metrics, such as Precision, MAP (Mean Average Precision) and AUC (Area Under the Curve).

The “Performance” operator in the Recommender Extension of RapidMiner can calculate the performance of RS using the following metrics: AUC (Area Under the Curve), Prec@N (Precision at N), NDCG (Non-Discounted Cumulative Gain), and MAP (Mean Average Precision) (Hofmann & Klinkenberg, 2017). In this assignment, only the AUC, Prec@N, MAP values of different RS are calculated and analysed.

3.2. Evaluation Results

In this section, 3 different metrics are used to evaluate different types of RS implemented in Chapter 2, namely the AUC, Prec@N (where N stands for the number of recommended items by RS, in this case, N = 5, 10 and 15), and MAP. The evaluation results are in Tables 3.1 – 3.14 in the subsequent discussions. In these tables, different shades of colours on the blue and red colourmap (see Figure 3.1) are populated to each cell according to their corresponding values. The blue and red cells indicate low and high values respectively in each table.

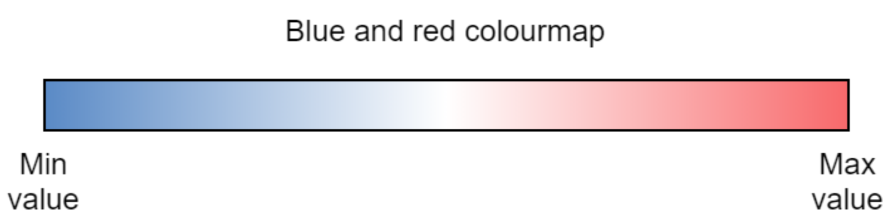

Figure 3.1: Blue and red colourmap.

In general, the AUC values increase with increasing values of *k* from 20 – 100 for all types of CF RS. Based on the evaluation result, CF RS with weighted item or user k-NN have higher AUC values across all values of *k*. User-to-User CF RS generally performs better with lower values of *k*, but Item-to-Item CF RS has better performance with higher values of *k*. Table 3.1 summarizes the AUC values for different CF RS for different values of *k*.

Table 3.1: AUC values for different CF RS by k.

k	User k-NN	Weighted user k-NN	Item k-NN	Weighted Item k-NN
20	0.7215	0.7252	0.7201	0.7231
40	0.7569	0.7603	0.7547	0.7580
60	0.7699	0.7732	0.7708	0.7744
80	0.7748	0.7782	0.7803	0.7836
100	**0.7779**	**0.7813**	**0.7841**	**0.7881**

Unlike AUC, MAP values of different CF RS do not increase monotonically with increasing values of *k* from 20 – 100. In 3 out of 4 cases, the CF RS reach their respective highest MAP when *k* = 60, whereas for Weighted Item-to-Item CF RS, it shows the highest MAP value when *k* = 80. Weighted Item-to-Item CF RS has the highest MAP values across all values of *k*. This might suggest that item-to-item CF approach is more scalable than user-to-user CF approach because the number of movies receiving ratings in the system is greater than the number of users in the system. Table 3.2 summarizes the MAP values for different CF RS for different values of *k*.

Table 3.2: MAP values for different CF RS by k.

k	User k-NN	Weighted user k-NN	Item k-NN	Weighted Item k-NN
20	0.0895	0.0914	0.0777	0.1007
40	0.0948	0.0959	0.0818	0.1011
60	**0.0955**	**0.0964**	**0.0832**	0.1014
80	0.0950	0.0961	0.0828	**0.1018**
100	0.0935	0.0948	0.0823	0.0998

To compare Prec@N for different values of N (N = 5, 10, 15), different CF RS are selected with the k values yielding the highest number of top Prec@N values in each RS. The result shows that Weighted Item-to-Item CF RS yields the highest Prec@N values. This matches the observation from Table 3.2, which shows that weighted Item-to-Item CF RS has the highest MAP values. The Prec@N of these RS are summarized in Table 3.3.

Table 3.3: Prec@N values for different CF RS.

Model	Prec@5	Prec@10	Prec@15
User k-NN (k=60)	0.2381	0.1891	0.1573

Model	Prec@5	Prec@10	Prec@15
Weighted User k-NN (k=60)	0.2396	0.1897	0.1584
Item k-NN (k=80)	0.1827	0.1627	0.1471
Weighted Item k-NN (k=40)	**0.2536**	**0.1991**	**0.1718**

For CARS, the AUC values also increase with increasing values of *k* from 20 – 100. In general, the recommendations made during weekdays (from Mondays to Fridays) show higher AUC values. This indicates that the RS can better predict the items liked by users in the system from the training set using only ratings from weekdays. One reason for this is because there are more ratings made during weekdays as compared to weekends. The AUC values of the CARS implemented using weighted item k-NN is the worst, which is below 0.50 for *k* = 20, 40, and 60. Table 3.4 summarizes the AUC values for different CARS for different values of *k*.

Table 3.4: AUC values for different CARS by k.

k	User-Weekdays	User-Weekends	Item-Weekdays	Item-Weekends
20	0.7062	0.5461	0.6856	0.4298
40	0.7435	0.6265	0.7233	0.4648
60	0.7558	0.6683	0.7423	0.4886
80	0.7635	0.6878	0.7520	0.5039
100	**0.7680**	**0.6963**	**0.7604**	**0.5212**

The same pattern can be observed in the table showing the MAP values for different CARS, in which, the MAP values of the recommendations made during weekdays. On top of that, the MAP values also do not necessarily increase monotonically with increasing values of *k*. Table 3.5 shows the MAP values for of different CARS for different values of *k*.

Table 3.5: MAP values for different CARS by k.

k	User-Weekdays	User-Weekends	Item-Weekdays	Item-Weekends
20	0.0748	0.0239	0.0785	**0.0091**
40	0.0778	0.0315	0.0792	0.0080
60	**0.0791**	0.0342	**0.0809**	0.0072
80	0.0784	0.0351	0.0806	0.0066
100	0.0781	**0.0374**	0.0807	0.0069

Similar to CF RS, the comparison of the Prec@N for different values of N (N = 5, 10, 15) is also made between different CARS. These CARS are selected with the k values yielding the highest number of top Prec@N values in each RS. From the comparison, the CARS implemented to generate list of recommended movies during the weekdays yields better performance. The CARS implemented using Weighted User k-NN yields the highest Prec@5 values. On the other hand, the CARS implemented using Weighted Item k-NN yields the highest Prec@10 and Prec@15 values. The selected value of *k* for both these implementations are *k* = 60. This observation matches with the previous discussions on the AUC and MAP for different CARS. The Prec@N of these RS are summarized in Table 3.6.

Table 3.6: Prec@N values for different CARS.

Model	Prec@5	Prec@10	Prec@15
Weighted User k-NN: weekdays (k=60)	**0.1712**	0.1371	0.1193
Weighted User k-NN: weekends (k=100)	0.0371	0.0307	0.0277
Weighted Item k-NN: weekdays (k=60)	0.1705	**0.1428**	**0.1242**
Weighted Item k-NN: weekends (k=20)	0.0079	0.0061	0.0055

Content-Based Recommender Systems

For content-based RS, the AUC values also increase with increasing values of *k* from 20 – 100. It is observed that User Attribute-based RS generates better list of recommended movies compared to Item Attribute-based RS. This might be due to different approaches used to generate list of attribute IDs for the movie genres and user personality traits respectively. During data preprocessing steps in Section 1.3, each of the 5 user personality traits are subdivided into 5 degrees. Therefore, there are fixed number of user attributes assigned to each user. Unlike user attributes, each movie is assigned different number of attribute IDs generated using different genres. This makes the computation of the user similarities is more consistent. Table 3.7: AUC values for different content-based recommender systems by k. shows the AUC values for User and Item Attribute-based RS across different values of *k* respectively.

Table 3.7: AUC values for different content-based recommender systems by k.

k	User Attribute-based	Item Attribute-based
20	0.6213	0.5409
40	0.6639	0.5759
60	0.6853	0.5933
80	0.6982	0.6068
100	**0.7067**	**0.6157**

The comparison is then made between User and Item Attribute-based RS across different values of *k* respectively. Similar to previous comparisons, the MAP values do not necessarily increase monotonically with increasing values of *k* from 20 – 100. It is observed that the MAP values for User Attribute-based RS is at least 3 times of those values in Item Attribute-based RS. This indicates that User Attribute-based RS is 3 times more likely to recommend relevant movies to target users than Item Attribute-based RS. Table 3.8: MAP values for different content-based RS by k. shows the MAP values for User and Item Attribute-based RS across different values of *k* respectively.

Table 3.8: MAP values for different content-based RS by k.

k	User Attribute-based	Item Attribute-based
20	0.0504	0.0165
40	0.0593	**0.0177**
60	0.0639	0.0176
80	0.0661	0.0173
100	**0.0679**	0.0171

Prec@N for different values of N (N = 5, 10, 15) is also compared between User and Item Attribute-based RS. The selected values of *k* for User and Item Attribute-based RS are 100 and 40 respectively. From the comparison result, User Attribute-based RS performs better for each N in the Prec@N values. This matches with the observation from Table 3.8. The Prec@N of these RS are summarized in Table 3.9.

Table 3.9: Prec@N values for different content-based RS.

Model	Prec@5	Prec@10	Prec@15
User Attribute-based (k=100)	**0.1818**	**0.1503**	**0.1281**
Item-Attribute based (k=40)	0.0238	0.0285	0.0262

Hybrid Recommender Systems

As mentioned in Section 2.3, the 2 types of Hybrid RS are implemented using combinations of RS with different approaches. Table 3.10 shows the configurations of the RS used for comparison in this section.

Table 3.10: Configurations of different RS used for comparison.

Model	Item + Item Attribute	User + User Attribute
Model 1	Item-to-Item CF RS (using weighted k-NN): *k* = 40	User-to-User CF RS (using weighted k-NN): *k* = 60
Model 2	Item Attribute-based RS: *k* = 60	User Attribute-based RS: *k* = 100
Hybrid	Model 1 + Model 2	Model 1 + Model 2

In general, the AUC values of the hybrid RS are higher than those RS implemented using only one approach. The comparison of the AUC values for different RS are shown in Table 3.11.

Table 3.11: AUC values for CF, content-based, and hybrid recommender systems.

Model	Item + Item Attribute	User + User Attribute
Model 1	0.7580	0.7732
Model 2	0.5759	0.7067
Hybrid	**0.7749**	**0.7766**

On the other hand, Hybrid RS 1: combining Item-to-Item CF and Item Attribute-Based RS yields a greater MAP values than the original RS, whereas Hybrid RS 2: combining User-to-User CF and User Attribute-Based RS yields a relatively lower MAP values than the User-to-User CF RS. The comparison of the MAP values for different RS are shown in Table 3.12.

Table 3.12: MAP values for CF, content-based, and the hybrid recommender systems.

Model	Item + Item Attribute	User + User Attribute
Model 1	0.1011	**0.0964**
Model 2	0.0177	0.0679
Hybrid	**0.1022**	0.0962

Finally, the comparison of the Prec@N (for N = 5, 10, 15) for Hybrid RS 1 and Hybrid RS 2 with different RS are summarized in Table 3.13 and Table 3.14 respectively. The result from these tables matches with the observation from Table 3.12, in which, Hybrid RS 1 yields the highest Prec@N across different values of N, whereas the Prec@N for Hybrid RS 2 is relatively lower than those for User-to-User CF RS for all values of N.

Table 3.13: Prec@N values for item-to-item CF, item attribute-based, and the hybrid RS.

Model	Prec@5	Prec@10	Prec@15
Model 1	0.2536	0.1991	0.1718
Model 2	0.0238	0.0285	0.0262
Hybrid	**0.2547**	**0.2012**	**0.1733**

Table 3.14: Prec@N values for user-to-user CF, user attribute-based, and the hybrid RS.

Model	Prec@5	Prec@10	Prec@15
Model 1	**0.2396**	**0.1897**	**0.1584**
Model 2	0.1818	0.1503	0.1281
Hybrid	0.2391	0.1890	0.1577

4. Analysis and Visualization

To analyse the performance of different recommender systems in the movie recommendation task, the lists of top 5 movies generated by the different recommender systems introduced and implemented in Chapter 2 are further inspected. For this purpose, two users with the following IDs are selected from the dataset for analysis, namely:

- User 1: `00fa91e202f5e48aa34c05d97867fa74`
- User 2: `022047320a00d607009323875a19face`

Table 4.1 shows list of different RS and their corresponding parameters used to generate the list of top 5 recommended movies. The movie ratings contributed by these two users to different movies are also listed in Appendix A and Appendix B respectively.

Table 4.1: List of RS and their corresponding parameters to generate list of top 5 recommended movies.

No	Recommender system	Parameters
1	Weighted User-to-User CF RS	User k-NN: k=60; Weighted k-NN
2	Weighted Item-to-Item CF RS	Item k-NN: k=40; Weighted k-NN
3	Weighted User-to-User CARS	User k-NN: k=60; Weighted k-NN; Weekdays = 1 (Mon-Fri)
4	Weighted User-to-User CARS	User k-NN: k=60; Weighted k-NN; Weekdays = 2 (Sun)
5	Weighted Item-to-Item CARS	Item k-NN: k=40; Weighted k-NN; Weekdays = 1 (Mon-Fri)
6	Weighted Item-to-Item CARS	Item k-NN: k=40; Weighted k-NN; Weeidays = 2 (Sun)
7	User Attribute-Based RS	User Attribute k-NN: k=60
8	Item Attribute-Based RS	Item Attribute k-NN: k=100
9	Hybrid RS 1	No 2 + No 8
10	Hybrid RS 2	No 1 + No 7

4.1. User 1: `00fa91e202f5e48aa34c05d97867fa74`

Table 4.2 lists the movies most frequently recommended to User 1 by different recommender systems implemented in Chapter 2.

Table 4.2: List of movies in the top 5 recommendation list generated to User 1 by at least 2 movie recommender systems.

Movie ID	Movie Name	How many RS recommend this movie?
122918	Guardians of the Galaxy 2 (2017)	6
168252	Logan (2017)	5
122912	Avengers: Infinity War - Part I (2018)	4
128360	The Hateful Eight (2015)	4
122886	Star Wars: Episode VII - The Force Awakens (2015)	3
122904	Deadpool (2016)	2
122924	X-Men: Apocalypse (2016)	2
139385	The Revenant (2015)	2
164179	Arrival (2016)	2
166528	Rogue One: A Star Wars Story (2016)	2
171763	Baby Driver (2017)	2

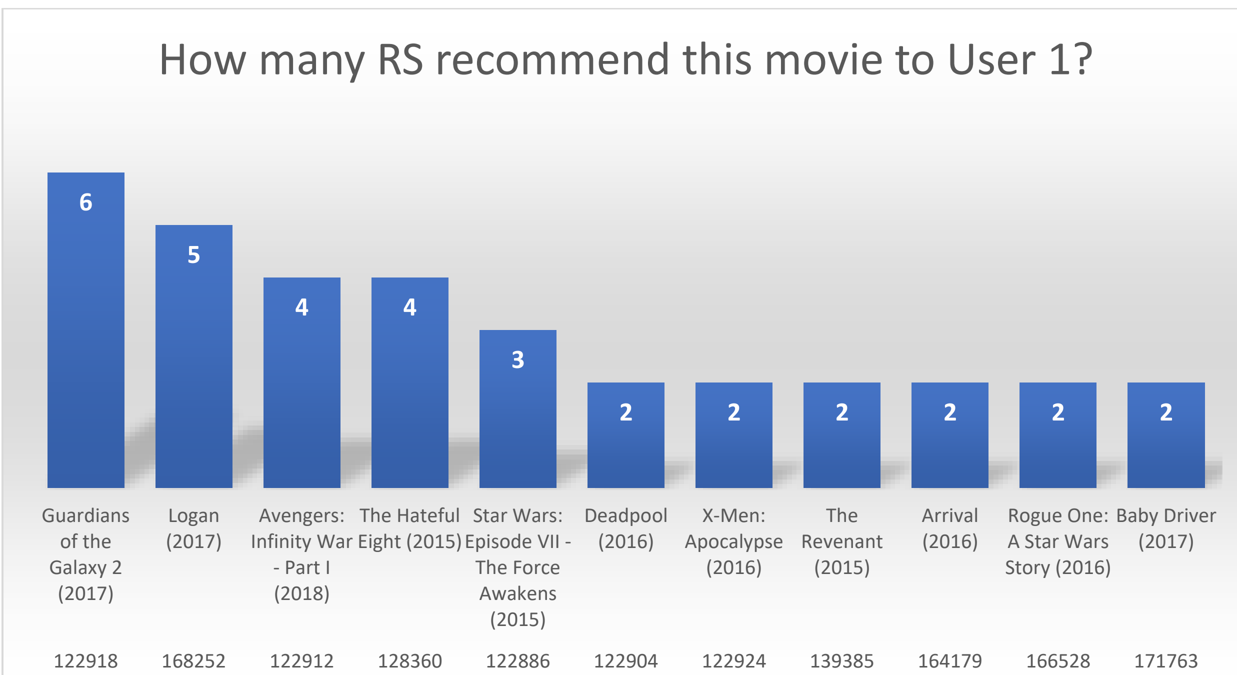

Figure 4.1: List of movies in the top 5 recommendation list generated to User 1 by at least 2 movie recommender systems.

From the movie list in Table 4.2, it is observed that most of the movies recommended to User 1 by these recommender systems are superhero movies, including the movies of the Marvel superheroes and Star Wars series. Among these movies, Guardians of the

Galaxy 2 (2017) and Logan (2017) are the two movies most frequently recommended to User 1. In the test set, both these movies are rated as 3.5 by User 1, which is above the average rating (approximately 3.36) given by User 1 to 155 different movies in the training set. This indicates that these movie recommender systems are able to provide personalized movie recommendations that are relevant to a user based on the past history of the user interactions, and also the movie genre or user personality for user attribute-based and item attribute-based recommender systems respectively.

Another movie is the Avengers: Infinity War – Part 1 (2018), which is never rated by User 1 in both the training and test sets. However, this movie is included as one of the recommended movies by both the user-to-user collaborative filtering and hybrid recommender systems. This shows that collaborative filtering approach in a recommender system is able to provide new recommendation to a user based on the feedback provided by the other users in the system. The list of top 5 movies recommended to User 1 by different recommender systems are displayed in Appendix C.

4.2. User 2: `022047320a00d607009323875a19face`

Table 4.3 lists the movies most frequently recommended to User 2 by different recommender systems implemented in Chapter 2.

Table 4.3: List of movies in the top 5 recommendation list generated to User 2 by at least 2 movie recommender systems.

Movie ID	Movie Name	How many RS recommend this movie?
122904	Deadpool (2016)	8
164179	Arrival (2016)	7
139644	Sicario (2015)	4
166528	Rogue One: A Star Wars Story (2016)	4
176371	Blade Runner 2049 (2017)	4
122886	Star Wars: Episode VII - The Force Awakens (2015)	3
128360	The Hateful Eight (2015)	3
168250	Get Out (2017)	3
180031	The Shape of Water (2017)	2

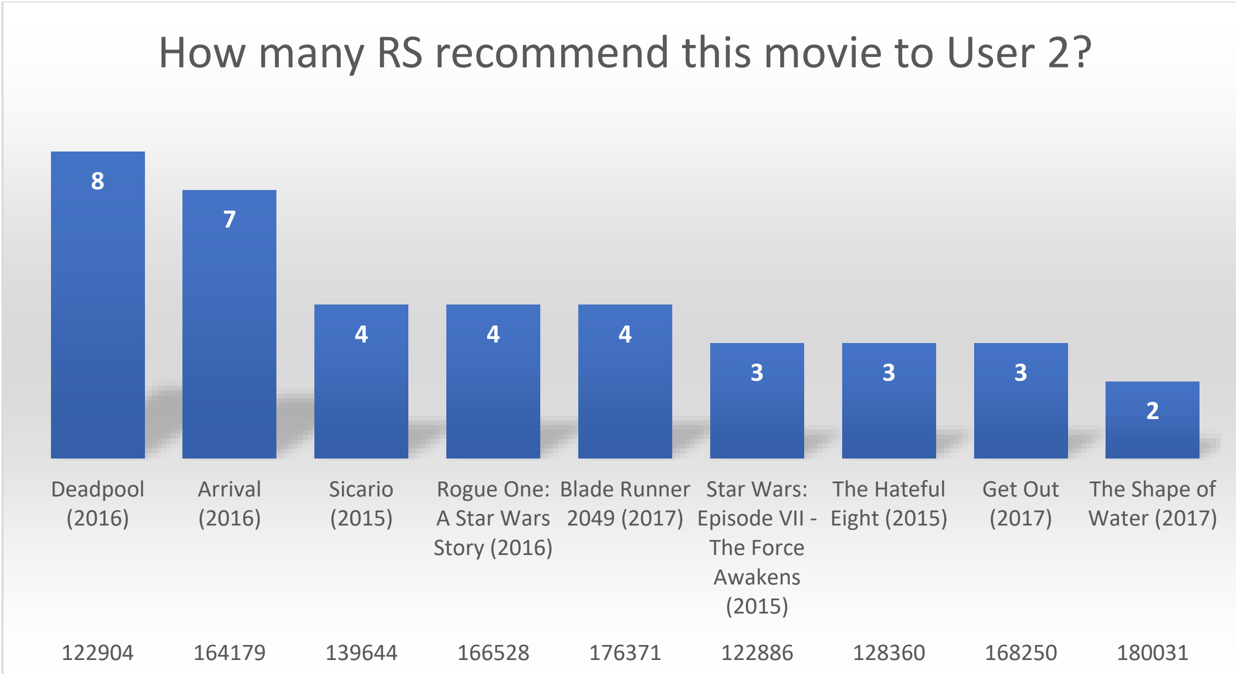

Figure 4.2: List of movies in the top 5 recommendation list generated to User 2 by at least 2 movie recommender systems.

From the movie list in Table 4.3, Deadpool (2016) receives the highest numbers of recommendations by 8 different movie recommender systems. It is among the top 2 recommended movies in these 8 recommender systems. In the test set, this movie is rated as 3.5 by User 2, which is higher than the average ratings (approximately 3.35) given by this user to 143 different movies in the training set. Therefore, it is a relevant recommendation to User 2.

The second most recommended movies to User 2 is Arrival (2016), even though this movie has never received rating from User 2 in both training and test sets. This case is similar to the recommendation of Avengers: Infinity War – Part 1 (2018) to User 1 in Section 4.1, in which, the recommendation is generated by different movie recommender systems based on different data, such as user and item similarities and past feedback provided by the target user and other users to the movies in the system.

The next movie on the list is Sicario (2015). The rating given by User 2 to this movie is only 2.0 in the test set, but this movie is recommended to User 2 by 4 different recommender systems implemented, which incorporates memory-based approaches. This indicates a strong assumption in the memory-based recommender systems, in which, user's taste and the similarity between users and items in the system can help to predict user's preference towards items in the recommender systems, while generating list of item recommendations to the target users. This assumption might not be true in some cases, but the memory-based recommender systems can perform generally well in most cases. The list of top 5 movies recommended to User 1 by different recommender systems are displayed in Appendix D.

5. Discussion and Conclusion

In this assignment, different movie RS are implemented, and then the performance of each RS is evaluated using metrics such as AUC, Prec@N and MAP. In general, weighted k-NN in CF RS yields better performance in recommending movies to the target users because it considers the similarities between users or movies when predicting the target users' preferences towards the unseen movies in the dataset. There is a strong assumption for the CARS implemented in this assignment, in which, users have different movie preferences between weekdays and weekends. Even though the main goal of implementing CARS is to generate a more relevant list of recommendation to users, the CARS do not perform better than the simple CF RS, especially during weekends. This is due to the reduced number of training data used to construct the RS. For content-based RS, user attribute-based RS outperforms the item attribute-based RS in generating list of recommended movies to target users in the system. Finally, hybrid RS that is constructed by combining item-to-item CF and item attribute-based RS yields better performance than the RS implemented using item-to-item CF or item attribute-based approaches in recommending list of movies to the target users.

An analysis is also performed to study the list of recommended movies by 10 different RS to 2 users in the system. This analysis shows that most of the RS can generate reliable list of movies to the target users from different information, such as the user or item similarities, and user and item attributes such as user personality traits and movie genres. This analysis also verifies the measurements of different metrics used to evaluate different RS because each RS has different sets of recommendations and their evaluation results are also different.

In this assignment, the RS is constructed using only merely 12% of the original ratings data. Besides, the train-test split performed to generate the training and test data is done in such a way that it maintains the proportion of each users in both training and test data. A better approach is to split the data by timestamp so that a small the data after certain timestamp. Nevertheless, this simple assumption is sufficient to generate fairly good, if not really personalized, list of recommendations to target users.

References

Gnanasundaram, S., & Shrivastava, A. (2012). *Information storage and management: storing, managing, and protecting digital information in classic, virtualized, and cloud environments.* John Wiley & Sons.

Hofmann, M., & Klinkenberg, R. (2017). *RapidMiner: data mining use cases and business analytics applications.* Chapman & Hall/CRC.

Isinkaye, F. O., Folajimi, Y. O., & Ojokoh, B. A. (2015). Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal, 16*, 261–273.

Konstan, J. A., & Riedl, J. (2012). Recommender systems: from algorithms to user experience. *User modeling and user-adapted interaction, 22*, 101–123.

Taifi, M. (6, 2020). MRR vs MAP vs NDCG: Rank-Aware Evaluation Metrics And When To Use Them. *MRR vs MAP vs NDCG: Rank-Aware Evaluation Metrics And When To Use Them*. The Startup. Retrieved from https://medium.com/swlh/rank-aware-recsys-evaluation-metrics-5191bba16832

Appendix A. Movie Ratings by User 1

The training and test sets sorted by ratings for User 1 from Section 4.1 are shown in Table A-1: and Table A-2 respectively.

Table A-1: Movie ratings by User 1 in the training set.

Movie ID	Movie Name	Rating
6863	School of Rock (2003)	5
1028	Mary Poppins (1964)	4.5
2018	Bambi (1942)	4.5
162414	Moonlight	4.5
2918	Ferris Bueller's Day Off (1986)	4.5
181315	Phantom Thread (2017)	4.5
168250	Get Out (2017)	4.5
1967	Labyrinth (1986)	4.5
48774	Children of Men (2006)	4.5
97194	Thing: Terror Takes Shape, The (1998)	4.5
3068	Verdict, The (1982)	4.5
5632	Bloody Sunday (2002)	4
122916	Thor: Ragnarok (2017)	4
93838	The Raid: Redemption (2011)	4
71579	Education, An (2009)	4
122926	Untitled Spider-Man Reboot (2017)	4
195159	Spider-Man: Into the Spider-Verse (2018)	4
98491	Paperman (2012)	4
133771	The Lobster (2015)	4
152081	Zootopia (2016)	4
177615	Lady Bird (2017)	4
3535	American Psycho (2000)	4
160848	The Red Turtle (2016)	4
4816	Zoolander (2001)	4
6593	Freaky Friday (2003)	4
166528	Rogue One: A Star Wars Story (2016)	4

Movie ID	Movie Name	Rating
162738	American Honey (2016)	4
77307	Dogtooth (Kynodontas) (2009)	4
102684	Only God Forgives (2013)	4
1198	Raiders of the Lost Ark (Indiana Jones and the Raiders of the Lost Ark) (1981)	4
166024	Whiplash (2013)	4
181671	The Breadwinner (2017)	4
72226	Fantastic Mr. Fox (2009)	4
148626	Big Short, The (2015)	4
174053	Photographer (2015)	4
899	Singin' in the Rain (1952)	4
121231	It Follows (2014)	4
143859	Hail, Caesar! (2016)	4
82173	Tiny Furniture (2010)	4
142488	Spotlight (2015)	4
56145	Mist, The (2007)	4
167564	The Little Hours (2017)	4
174727	Good Time (2017)	4
6552	Dirty Pretty Things (2002)	4
134130	The Martian (2015)	4
159817	Planet Earth (2006)	4
162376	The Rack Pack (2016)	3.5
112582	Life Itself (2014)	3.5
62511	Synecdoche, New York (2008)	3.5
109848	Under the Skin (2013)	3.5
6873	Intolerable Cruelty (2003)	3.5
104283	Wind Rises, The (Kaze tachinu) (2013)	3.5
122920	Captain America: Civil War (2016)	3.5
85774	Senna (2010)	3.5
2797	Big (1988)	3.5

Movie ID	Movie Name	Rating
94931	Take This Waltz (2011)	3.5
122904	Deadpool (2016)	3.5
115713	Ex Machina (2015)	3.5
139644	Sicario (2015)	3.5
137595	Magic Mike XXL (2015)	3.5
55814	Diving Bell and the Butterfly, The (Scaphandre et le papillon, Le) (2007)	3.5
2124	Addams Family, The (1991)	3.5
162082	Train to Busan (2016)	3.5
143355	Wonder Woman (2017)	3.5
104272	Blackfish (2013)	3.5
531	Secret Garden, The (1993)	3.5
72395	Precious (2009)	3.5
179819	Star Wars: The Last Jedi (2017)	3.5
160718	Piper (2016)	3.5
71379	Paranormal Activity (2009)	3.5
177651	The Florida Project (2017)	3.5
143385	Bridge of Spies (2015)	3.5
103801	Drinking Buddies (2013)	3.5
70728	Bronson (2009)	3.5
175655	Girls Trip (2017)	3.5
1227	Once Upon a Time in America (1984)	3.5
61323	Burn After Reading (2008)	3.5
122906	Black Panther (2017)	3.5
6188	Old School (2003)	3.5
173145	War for the Planet of the Apes (2017)	3.5
1372	Star Trek VI: The Undiscovered Country (1991)	3.5
1061	Sleepers (1996)	3.5
1292	Being There (1979)	3.5
138702	Feast (2014)	3.5

Movie ID	Movie Name	Rating
96588	Pitch Perfect (2012)	3
122892	Avengers: Age of Ultron (2015)	3
158528	The Shallows (2016)	3
40815	Harry Potter and the Goblet of Fire (2005)	3
134393	Trainwreck (2015)	3
72167	Boondock Saints II: All Saints Day, The (2009)	3
146656	Creed (2015)	3
69784	BrÃ¼no (Bruno) (2009)	3
5957	Two Weeks Notice (2002)	3
27773	Old Boy (2003)	3
48780	Prestige, The (2006)	3
155968	The Night Manager (2016)	3
6936	Elf (2003)	3
3034	Robin Hood (1973)	3
162600	Miss Peregrine's Home for Peculiar Children (2016)	3
5504	Spy Kids 2: The Island of Lost Dreams (2002)	3
104241	Kick-Ass 2 (2013)	3
122900	Ant-Man (2015)	3
71211	Informant!, The (2009)	3
190089	Hannah Gadsby: Nanette (2018)	3
3536	Keeping the Faith (2000)	3
88140	Captain America: The First Avenger (2011)	3
54001	Harry Potter and the Order of the Phoenix (2007)	3
106782	Wolf of Wall Street, The (2013)	3
3452	Romeo Must Die (2000)	3
155820	Keanu (2016)	3
180031	The Shape of Water (2017)	3
170697	Gifted (2017)	3
3826	Hollow Man (2000)	3
140267	The Witch (2015)	3

Movie ID	Movie Name	Rating
183869	Hereditary (2018)	3
176371	Blade Runner 2049 (2017)	3
163134	Your Name. (2016)	3
54286	Bourne Ultimatum, The (2007)	3
4896	Harry Potter and the Sorcerer's Stone (a.k.a. Harry Potter and the Philosopher's Stone) (2001)	3
7045	Witches, The (1990)	3
157276	Our Kind of Traitor (2016)	3
160080	Ghostbusters (2016)	3
158830	The BFG (2016)	3
141544	Turbo Kid (2015)	3
122922	Doctor Strange (2016)	3
27706	Lemony Snicket's A Series of Unfortunate Events (2004)	3
74948	Harry Brown (2009)	3
54372	Tell No One (Ne le dis Ã personne) (2006)	3
178061	I, Tonya (2017)	3
6254	Awful Truth, The (1937)	3
55363	Assassination of Jesse James by the Coward Robert Ford, The (2007)	2.5
73015	It's Complicated (2009)	2.5
7323	Good bye, Lenin! (2003)	2.5
81782	Unstoppable (2010)	2.5
148881	World of Tomorrow (2015)	2.5
136020	Spectre (2015)	2.5
6755	Bubba Ho-tep (2002)	2.5
82461	Tron: Legacy (2010)	2.5
180985	The Greatest Showman (2017)	2.5
160438	Jason Bourne (2016)	2.5
135569	Star Trek Beyond (2016)	2.5
164179	Arrival (2016)	2.5
72720	Single Man, A (2009)	2.5

Movie ID	Movie Name	Rating
136864	Batman v Superman: Dawn of Justice (2016)	2.5
1373	Star Trek V: The Final Frontier (1989)	2.5
58156	Semi-Pro (2008)	2.5
93272	Dr. Seuss' The Lorax (2012)	2.5
102445	Star Trek Into Darkness (2013)	2.5
159161	Ali Wong: Baby Cobra (2016)	2.5
68159	State of Play (2009)	2.5
5686	Russian Ark (Russkiy Kovcheg) (2002)	2
71033	Secret in Their Eyes, The (El secreto de sus ojos) (2009)	2
3702	Mad Max (1979)	2
168330	I Don't Feel at Home in This World Anymore (2017)	2
6934	Matrix Revolutions, The (2003)	2

Table A-2: Movie ratings by User 1 in the test set.

Movie ID	Movie Name	Rating
1210	Star Wars: Episode VI - Return of the Jedi (1983)	4.5
2571	Matrix, The (1999)	4.5
3910	Dancer in the Dark (2000)	4.5
104944	Short Term 12 (2013)	4.5
176419	Mother! (2017)	4.5
62155	Nick and Norah's Infinite Playlist (2008)	4
177689	The Killing of a Sacred Deer (2017)	4
165549	Manchester by the Sea (2016)	4
3741	Badlands (1973)	4
903	Vertigo (1958)	4
3310	Kid, The (1921)	4
86833	Bridesmaids (2011)	4
1704	Good Will Hunting (1997)	4
6290	House of 1000 Corpses (2003)	4
168326	The Big Sick (2017)	4

Movie ID	Movie Name	Rating
329	Star Trek: Generations (1994)	4
128360	The Hateful Eight (2015)	4
118246	Clouds of Sils Maria (2014)	3.5
166461	Moana (2016)	3.5
837	Matilda (1996)	3.5
122918	Guardians of the Galaxy 2 (2017)	3.5
116897	Wild Tales (2014)	3.5
167036	Sing (2016)	3.5
178335	Thelma (2017)	3.5
4306	Shrek (2001)	3.5
168252	Logan (2017)	3.5
6565	Seabiscuit (2003)	3
1375	Star Trek III: The Search for Spock (1984)	3
2105	Tron (1982)	3
100556	Act of Killing, The (2012)	3
166635	Passengers (2016)	3
122886	Star Wars: Episode VII - The Force Awakens (2015)	3
120466	Chappie (2015)	3
91535	Bourne Legacy, The (2012)	3
122902	Fantastic Four (2015)	2.5
93988	North & South (2004)	2.5
112421	Frank (2014)	2
66934	Dr. Horrible's Sing-Along Blog (2008)	2
51077	Ghost Rider (2007)	2

Appendix B. Movie Ratings by User 2

The training and test sets sorted by ratings for User 2 from Section 4.2 are shown in Table B-1 and Table B-2 respectively.

Table B-1: Movie ratings by User 2 in the training set.

Movie ID	Movie Name	Rating
142488	Spotlight (2015)	5
1348	Nosferatu (Nosferatu, eine Symphonie des Grauens) (1922)	5
1247	Graduate, The (1967)	5
1086	Dial M for Murder (1954)	4.5
923	Citizen Kane (1941)	4.5
134853	Inside Out (2015)	4.5
1256	Duck Soup (1933)	4.5
1693	Amistad (1997)	4.5
116411	Tangerines (2013)	4.5
122882	Mad Max: Fury Road (2015)	4
1307	When Harry Met Sally... (1989)	4
955	Bringing Up Baby (1938)	4
5013	Gosford Park (2001)	4
471	Hudsucker Proxy, The (1994)	4
134130	The Martian (2015)	4
71264	Cloudy with a Chance of Meatballs (2009)	4
139385	The Revenant (2015)	4
143355	Wonder Woman (2017)	4
1950	In the Heat of the Night (1967)	4
68157	Inglourious Basterds (2009)	4
8981	Closer (2004)	4
166643	Hidden Figures (2016)	4
54272	Simpsons Movie, The (2007)	4
34271	Hustle & Flow (2005)	4
92259	Intouchables (2011)	4
5218	Ice Age (2002)	4

Movie ID	Movie Name	Rating
163645	Hacksaw Ridge (2016)	4
49530	Blood Diamond (2006)	4
30812	Aviator, The (2004)	4
127152	Going Clear: Scientology and the Prison of Belief (2015)	4
168608	Mudbound (2017)	4
115713	Ex Machina (2015)	4
1120	People vs. Larry Flynt, The (1996)	4
2529	Planet of the Apes (1968)	4
105844	12 Years a Slave (2013)	4
105504	Captain Phillips (2013)	4
165549	Manchester by the Sea (2016)	4
90531	Shame (2011)	4
168252	Logan (2017)	4
133771	The Lobster (2015)	4
106100	Dallas Buyers Club (2013)	4
139525	Cartel Land (2015)	4
87234	Submarine (2010)	3.5
158966	Captain Fantastic (2016)	3.5
143385	Bridge of Spies (2015)	3.5
88163	Crazy, Stupid, Love. (2011)	3.5
6957	Bad Santa (2003)	3.5
102993	Way, Way Back, The (2013)	3.5
103984	Great Beauty, The (Grande Bellezza, La) (2013)	3.5
156605	Paterson	3.5
8533	Notebook, The (2004)	3.5
148626	Big Short, The (2015)	3.5
152081	Zootopia (2016)	3.5
111384	Blue Ruin (2013)	3.5
101895	42 (2013)	3.5
59369	Taken (2008)	3.5

Movie ID	Movie Name	Rating
182715	Annihilation (2018)	3.5
122906	Black Panther (2017)	3.5
180045	Molly's Game (2017)	3.5
235	Ed Wood (1994)	3.5
73017	Sherlock Holmes (2009)	3.5
1645	The Devil's Advocate (1997)	3.5
36529	Lord of War (2005)	3.5
71535	Zombieland (2009)	3.5
89118	Skin I Live In, The (La piel que habito) (2011)	3.5
805	Time to Kill, A (1996)	3.5
80549	Easy A (2010)	3.5
114342	Force Majeure (Turist) (2014)	3.5
122900	Ant-Man (2015)	3.5
158972	Toni Erdmann (2016)	3.5
7323	Good bye, Lenin! (2003)	3.5
110669	Hiding Cot (Piilopirtti) (1978)	3.5
145418	Trumbo (2015)	3.5
1049	Ghost and the Darkness, The (1996)	3.5
95088	Safety Not Guaranteed (2012)	3.5
180497	The Post (2017)	3.5
121231	It Follows (2014)	3.5
163	Desperado (1995)	3.5
6281	Phone Booth (2002)	3.5
177593	Three Billboards Outside Ebbing, Missouri (2017)	3.5
26236	White Sun of the Desert, The (Beloe solntse pustyni) (1970)	3
103253	Elysium (2013)	3
46972	Night at the Museum (2006)	3
1690	Alien: Resurrection (1997)	3
355	Flintstones, The (1994)	3
3617	Road Trip (2000)	3

Movie ID	Movie Name	Rating
122922	Doctor Strange (2016)	3
174909	Logan Lucky (2017)	3
68954	Up (2009)	3
145	Bad Boys (1995)	3
161580	Bad Moms (2016)	3
136018	Black Mass (2015)	3
91529	Dark Knight Rises, The (2012)	3
187593	Deadpool 2 (2018)	3
45186	Mission: Impossible III (2006)	3
3752	Me, Myself & Irene (2000)	3
112852	Guardians of the Galaxy (2014)	3
3744	Shaft (2000)	3
171763	Baby Driver (2017)	3
47997	Idiocracy (2006)	3
158238	The Nice Guys (2016)	3
724	Craft, The (1996)	3
90600	Headhunters (Hodejegerne) (2011)	3
3969	Pay It Forward (2000)	3
91542	Sherlock Holmes: A Game of Shadows (2011)	3
161127	The Infiltrator (2016)	3
1438	Dante’s Peak (1997)	3
176073	The Incredible Jessica James (2017)	3
7445	Man on Fire (2004)	3
99114	Django Unchained (2012)	3
34319	Island, The (2005)	3
86332	Thor (2011)	3
1644	I Know What You Did Last Summer (1997)	3
103341	World’s End, The (2013)	3
98361	Byzantium (2012)	3
178061	I, Tonya (2017)	3

Movie ID	Movie Name	Rating
183611	Game Night (2018)	3
84954	Adjustment Bureau, The (2011)	3
3301	Whole Nine Yards, The (2000)	3
3994	Unbreakable (2000)	3
1687	Jackal, The (1997)	3
44	Mortal Kombat (1995)	3
5574	Transporter, The (2002)	3
4148	Hannibal (2001)	2.5
31696	Constantine (2005)	2.5
44665	Lucky Number Slevin (2006)	2.5
122898	Justice League (2017)	2.5
106782	Wolf of Wall Street, The (2013)	2.5
162606	The Accountant (2016)	2.5
99112	Jack Reacher (2012)	2.5
3717	Gone in 60 Seconds (2000)	2.5
135861	Ted 2 (2015)	2.5
33679	Mr. & Mrs. Smith (2005)	2.5
148956	How to Be Single (2016)	2
673	Space Jam (1996)	2
1359	Jingle All the Way (1996)	2
4310	Pearl Harbor (2001)	2
208	Waterworld (1995)	2
4015	Dude, Where's My Car? (2000)	2
102903	Now You See Me (2013)	2
1556	Speed 2: Cruise Control (1997)	1.5
880	Island of Dr. Moreau, The (1996)	1.5
8641	Anchorman: The Legend of Ron Burgundy (2004)	1

Table B-2: Movie ratings by User 2 in the test set.

Movie ID	Movie Name	Rating
1207	To Kill a Mockingbird (1962)	5
1961	Rain Man (1988)	4.5
94959	Moonrise Kingdom (2012)	4
5060	M*A*S*H (a.k.a. MASH) (1970)	4
3504	Network (1976)	4
93838	The Raid: Redemption (2011)	4
122904	Deadpool (2016)	3.5
112556	Gone Girl (2014)	3.5
4018	What Women Want (2000)	3.5
1665	Bean (1997)	3.5
127116	Experimenter (2015)	3.5
118898	A Most Violent Year (2014)	3.5
100383	Side Effects (2013)	3.5
1676	Starship Troopers (1997)	3.5
181315	Phantom Thread (2017)	3.5
3798	What Lies Beneath (2000)	3.5
3755	Perfect Storm, The (2000)	3
3624	Shanghai Noon (2000)	3
34048	War of the Worlds (2005)	3
100163	Hansel & Gretel: Witch Hunters (2013)	3
1047	Long Kiss Goodnight, The (1996)	3
143271	Dheepan (2015)	3
160438	Jason Bourne (2016)	3
159415	Swiss Army Man (2016)	3
139644	Sicario (2015)	3
136020	Spectre (2015)	3
122886	Star Wars: Episode VII - The Force Awakens (2015)	3
119145	Kingsman: The Secret Service (2015)	3
173941	Atomic Blonde (2017)	3
179819	Star Wars: The Last Jedi (2017)	3

Movie ID	Movie Name	Rating
3753	Patriot, The (2000)	3
3536	Keeping the Faith (2000)	3
1597	Conspiracy Theory (1997)	3
88140	Captain America: The First Avenger (2011)	2.5
1588	George of the Jungle (1997)	2
1562	Batman & Robin (1997)	2

Appendix C. List of Recommended Movies to User 1

The following shows the lists of top 5 recommended movies to User 1 by different recommender systems implemented in Chapter 2.

Table C-1: List of top 5 movies recommended by Weighted User-to-User CF RS to User 1.

Rank	Movie ID	Movie Name
1	122918	Guardians of the Galaxy 2 (2017)
2	122912	Avengers: Infinity War - Part I (2018)
3	171763	Baby Driver (2017)
4	168252	Logan (2017)
5	122924	X-Men: Apocalypse (2016)

Table C-2: List of top 5 movies recommended by Weighted Item-to-Item CF RS to User 1.

Rank	Movie ID	Movie Name
1	122918	Guardians of the Galaxy 2 (2017)
2	122886	Star Wars: Episode VII - The Force Awakens (2015)
3	128360	The Hateful Eight (2015)
4	122912	Avengers: Infinity War - Part I (2018)
5	168252	Logan (2017)

Table C-3: List of top 5 movies recommended by CARS (with weighted user k-NN) to User 1 during weekdays.

Rank	Movie ID	Movie Name
1	122904	Deadpool (2016)
2	166528	Rogue One: A Star Wars Story (2016)
3	164179	Arrival (2016)
4	168252	Logan (2017)
5	122918	Guardians of the Galaxy 2 (2017)

Table C-4: List of top 5 movies recommended by CARS (with weighted user k-NN) to User 1 during weekends.

Rank	Movie ID	Movie Name
1	152081	Zootopia (2016)
2	164909	La La Land (2016)
3	152077	10 Cloverfield Lane (2016)
4	134130	The Martian (2015)
5	179819	Star Wars: The Last Jedi (2017)

Table C-5: List of top 5 movies recommended by CARS (with weighted item k-NN) to User 1 during weekdays.

Rank	Movie ID	Movie Name
1	164179	Arrival (2016)
2	122904	Deadpool (2016)
3	166528	Rogue One: A Star Wars Story (2016)
4	128360	The Hateful Eight (2015)
5	139385	The Revenant (2015)

Table C-6: List of top 5 movies recommended by CARS (with weighted item k-NN) to User 1 during weekends.

Rank	Movie ID	Movie Name
1	100556	Act of Killing, The (2012)
2	4902	Devil's Backbone, The (Espinazo del diablo, El) (2001)
3	4084	Beverly Hills Cop II (1987)
4	82852	Little Fockers (2010)
5	481	Kalifornia (1993)

Table C-7: List of top 5 movies recommended by Content-Based RS (with user attribute k-NN) to User 1.

Rank	Movie ID	Movie Name
1	122886	Star Wars: Episode VII - The Force Awakens (2015)
2	122918	Guardians of the Galaxy 2 (2017)
3	139385	The Revenant (2015)

Rank	Movie ID	Movie Name
4	128360	The Hateful Eight (2015)
5	135143	Fantastic Beasts and Where to Find Them (2016)

Table C-8: List of top 5 movies recommended by Content-Based RS (with item attribute k-NN) to User 1.

Rank	Movie ID	Movie Name
1	147378	Tut (2015)
2	60161	Bang Gang (A Modern Love Story) (2016)
3	66297	33 Scenes from Life (33 sceny z zycia) (2008)
4	108540	Rampage at Apache Wells (1965)
5	110873	Rocket, The (2013)

Table C-9: List of top 5 movies recommended by Hybrid RS 1: Combining Item Attribute-Based and Weighted Item-to-Item CF RS to User 1.

Rank	Movie ID	Movie Name
1	122918	Guardians of the Galaxy 2 (2017)
2	122886	Star Wars: Episode VII - The Force Awakens (2015)
3	128360	The Hateful Eight (2015)
4	122912	Avengers: Infinity War - Part I (2018)
5	168252	Logan (2017)

Table C-10: List of top 5 movies recommended by Hybrid RS 2: Combining User Attribute-Based and Weighted User-to-User CF RS to User 1.

Rank	Movie ID	Movie Name
1	122918	Guardians of the Galaxy 2 (2017)
2	122912	Avengers: Infinity War - Part I (2018)
3	171763	Baby Driver (2017)
4	168252	Logan (2017)
5	122924	X-Men: Apocalypse (2016)

Appendix D. List of Recommended Movies to User 2

The following shows the lists of top 5 recommended movies to User 2 by different recommender systems implemented in Chapter 2.

Table D-1: List of top 5 movies recommended by Weighted User-to-User CF RS to User 2.

Rank	Movie ID	Movie Name
1	122904	Deadpool (2016)
2	164179	Arrival (2016)
3	166528	Rogue One: A Star Wars Story (2016)
4	168250	Get Out (2017)
5	176371	Blade Runner 2049 (2017)

Table D-2: List of top 5 movies recommended by Weighted Item-to-Item CF RS to User 2.

Rank	Movie ID	Movie Name
1	122904	Deadpool (2016)
2	128360	The Hateful Eight (2015)
3	180031	The Shape of Water (2017)
4	164179	Arrival (2016)
5	139644	Sicario (2015)

Table D-3: List of top 5 movies recommended by CARS (with weighted user k-NN) to User 2 during weekdays.

Rank	Movie ID	Movie Name
1	122904	Deadpool (2016)
2	166528	Rogue One: A Star Wars Story (2016)
3	164179	Arrival (2016)
4	139644	Sicario (2015)
5	122886	Star Wars: Episode VII - The Force Awakens (2015)

Table D-4: List of top 5 movies recommended by CARS (with weighted user k-NN) to User 2 during weekends.

Rank	Movie ID	Movie Name
1	122904	Deadpool (2016)
2	176371	Blade Runner 2049 (2017)
3	122882	Mad Max: Fury Road (2015)
4	148626	Big Short, The (2015)
5	122886	Star Wars: Episode VII - The Force Awakens (2015)

Table D-5: List of top 5 movies recommended by CARS (with weighted item k-NN) to User 2 during weekdays.

Rank	Movie ID	Movie Name
1	139644	Sicario (2015)
2	122904	Deadpool (2016)
3	164179	Arrival (2016)
4	176371	Blade Runner 2049 (2017)
5	168250	Get Out (2017)

Table D-6: List of top 5 movies recommended by CARS (with weighted item k-NN) to User 2 during weekends.

Rank	Movie ID	Movie Name
1	181	Mighty Morphin Power Rangers: The Movie (1995)
2	3016	Creepshow (1982)
3	5944	Star Trek: Nemesis (2002)
4	55946	Lions For Lambs (2007)
5	151	Rob Roy (1995)

Table D-7: List of top 5 movies recommended by Content-Based RS (with user attribute k-NN) to User 2.

Rank	Movie ID	Movie Name
1	122904	Deadpool (2016)
2	122886	Star Wars: Episode VII - The Force Awakens (2015)
3	164179	Arrival (2016)

Rank	Movie ID	Movie Name
4	166528	Rogue One: A Star Wars Story (2016)
5	128360	The Hateful Eight (2015)

Table D-8: List of top 5 movies recommended by Content-Based RS (with item attribute k-NN) to User 2.

Rank	Movie ID	Movie Name
1	49961	Notes on a Scandal (2006)
2	62644	Wave, The (Welle, Die) (2008)
3	3897	Almost Famous (2000)
4	127202	Me and Earl and the Dying Girl (2015)
5	49961	Bloody Sunday (2002)

Table D-9: List of top 5 movies recommended by Hybrid RS 1: Combining Item Attribute-Based and Weighted Item-to-Item CF RS to User 2.

Rank	Movie ID	Movie Name
1	122904	Deadpool (2016)
2	128360	The Hateful Eight (2015)
3	180031	The Shape of Water (2017)
4	164179	Arrival (2016)
5	139644	Sicario (2015)

Table D-10: List of top 5 movies recommended by Hybrid RS 2: Combining User Attribute-Based and Weighted User-to-User CF RS to User 2.

Rank	Movie ID	Movie Name
1	122904	Deadpool (2016)
2	164179	Arrival (2016)
3	166528	Rogue One: A Star Wars Story (2016)
4	168250	Get Out (2017)
5	176371	Blade Runner 2049 (2017)