project report

Name: Ganga Kaur

Student ID: C0727495

Email: [c0727495@mylambton.ca](mailto:c0727495@mylambton.ca)

Course: 1053 - Big Data Algorithms and Statistics

Contents

[**INTRODUCTION** 1](#_Toc521684423)

[**Types of recommendation models** 2](#_Toc521684424)

[**Content-based filtering** 2](#_Toc521684425)

[**Collaborative filtering** 2](#_Toc521684426)

[Application of Recommendation Engine 4](#_Toc521684427)

[**Amazon** 4](#_Toc521684428)

[**Netflix** 4](#_Toc521684429)

[**Spotify** 4](#_Toc521684430)

[**Best Buy** 4](#_Toc521684431)

[**Screenshot** 5](#_Toc521684432)

# **INTRODUCTION**

Recommendation engines are most likely experienced by almost everyone of us through the use of popular websites such as Amazon, Netflix, YouTube, Twitter, LinkedIn, and Facebook.

Recommendation engines try to present people with relevant content that they did not necessarily search for or that they might not even have heard of.

Typically, a recommendation engine tries to model the connections between users and some type of item. If users are shown movies related to a given movie, it could aid in discovering different movies and improving navigation on a site, again improving users' experience, engagement, and the relevance of that content to them. A recommendations engine can customize ads or sponsored content for a user based on their preferences

However, recommendation engines are not limited to movies, books, or products. The technique can be applied to just about any user-to-item relationship as well as user-to-user connections, such as those found on social networks, that make recommendations such as people you may know or who to follow.

In the immortal words of Steve Jobs - “**a lot of times, people don’t know what they want until you show it to them**.”

# **Types of recommendation models**

Recommender systems are widely studied, and there are many approaches used, but there are two that are probably most prevalent:

* Content-based filtering
* Collaborative filtering

## **Content-based filtering**

Content-based recommendation engine works with existing profiles of users and particular item. A user profile might be seen as a set of assigned keywords (terms, features) collected by algorithm from items found relevant (or interesting) by the user. An item profile is a set of assigned keywords (terms, features) of the item itself.

For a movie recommendation engine, a content-based approach would be to recommend movies that are of highest **similarity** based on its features, such as genres, actors, directors, year of production, etc. The assumption here is that users have preferences for a certain type of product, so we try to recommend a similar product to what the user has expressed liking for. Also, the goal here is to provide **alternatives** or **substitutes** to the item that was viewed.

Advantages

Content-based recommender systems don’t require a lot of user data. You just need item data and you’re able to start giving recommendations to users. Also, your recommendation engine does not depend on lots of user data, so it is possible to give recommendations to even your first customer as long as you have adequate data to build his user profile.

Disadvantages

Your item data needs to be well distributed. It won’t be effective to have a content-based recommender if 80% of your movies are action movies. Also, the recommendations you get will likely be direct substitutes, and not complements, of the item the user interacted with. Complements are more likely discovered through collaborative techniques, which will be discussed in a later section.

## **Collaborative filtering**

* **The User-Based Collaborative Filtering Approach**  
  The User-Based Collaborative Filtering approach groups users according to prior usage behavior or according to their preferences, and then recommends an item that a similar user in the same group viewed or liked. To put this in layman terms, if user 1 liked movie A, B and C, and if user 2 liked movie A and B, then movie C might make a good recommendation to user 2. The User-Based Collaborative Filtering approach mimics how word-of-mouth recommendations work in real life.

**Advantages**

User-based Collaborative Filtering gives recommendations that can be complements to the item the user was interacting with. This might be a stronger recommendation than what a content-based recommender can provide as users might not be looking for direct substitutes to a movie they had just viewed or previously watched.

Here we find look alike users based on similarity and recommend movies which first user’s look-alike has chosen in past. This algorithm is very effective but takes a lot of time and resources. It requires to compute every user pair information which takes time. Therefore, for big base platforms, this algorithm is hard to implement without a very strong parallelizable system.

**Disadvantages**

User-based Collaborative Filtering is a type of Memory-based Collaborative Filtering that uses all user data in the database to create recommendations. Comparing the pairwise correlation of every user in your dataset is not scalable. If there were millions of users, this computation would be very time consuming. Possible ways to get around this would be to implement some form of dimensionality reduction, such as Principal Component Analysis, or to use a model-based algorithm instead. Also, user-based collaborative filtering relies on past user choices to make future recommendations. The implications of this is that it assumes that a user’s taste and preference remains more or less constant over time, which might not be true and makes it difficult to pre-compute user similarities offline.

* **Item Based collaborative filtering Approach**

Item based collaborative filtering is a model-based algorithm for recommender engines. In item based collaborative filtering similarities between items are calculated from rating-matrix. And based upon these similarities, user’s preference for an item not rated by him is calculated.

It is quite similar to previous algorithm, but instead of finding user’s look-alike, we try finding movie’s look-alike. Once we have movie’s look-alike matrix, we can easily recommend alike movies to user who have rated any movie from the dataset. This algorithm is far less resource consuming than user-user collaborative filtering. Hence, for a new user, the algorithm takes far lesser time than user-user collaborate as we don’t need all similarity scores between users. And with fixed number of movies, movie-movie look alike matrix is fixed over time.

# Application of Recommendation Engine

## **Amazon**

Amazon.com uses recommendations as a targeted marketing tool throughout its website. When a customer clicks on the “your recommendations” the link leads to another page where recommendations may be filtered even further by subject area, product types, and ratings of previous products and purchases. The customer can even see why a particular product has been recommended.

“At Amazon.com, we use recommendation algorithms to personalize the online store for each customer. The store radically changes based on customer interests, showing programming titles to a software engineer and baby toys to a new mother,” explain Greg Linden, Brent Smith, and Jeremy York in their paper Amazon.com.

## **Netflix**

Netflix uses RS personalized diversity to generate Top Ten recommendations for user households, so that it can offer videos that each member of the household may be interested in. The company also focuses on awareness and promoting trust to help develop its personalized approach. Netflix implements these strategies by explaining why it makes video recommendation and encouraging members to give feedback, so no opportunities to personalize are missed.

## **Spotify**

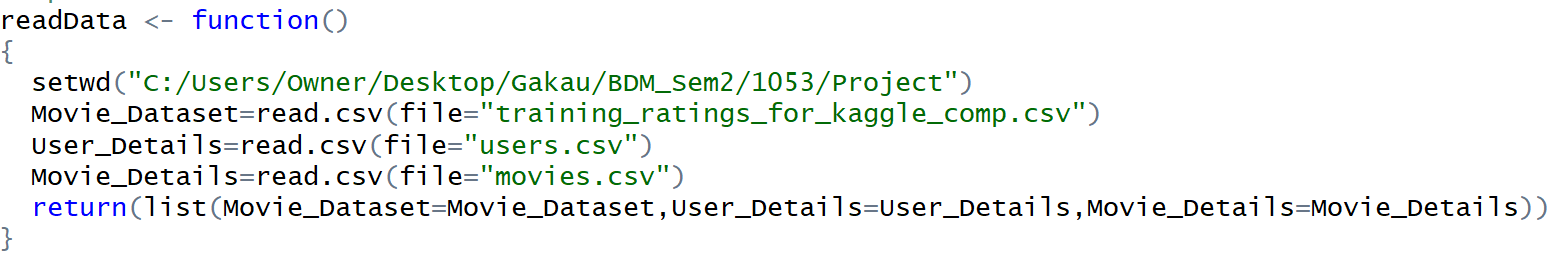
Possibly one of Spotify’s most innovative uses of AI and recommendation systems is their popular Discover Weekly playlist. Known as Release Radar, this algorithmically powered tool updates personal playlists on a weekly basis so that users won’t miss newly released music by artists they like.

## **Best Buy**

Best Buy has been using its recommendation system for eCommerce since 2015. The system works by predicting what a customer is interested in based on their individual browsing and purchase data.

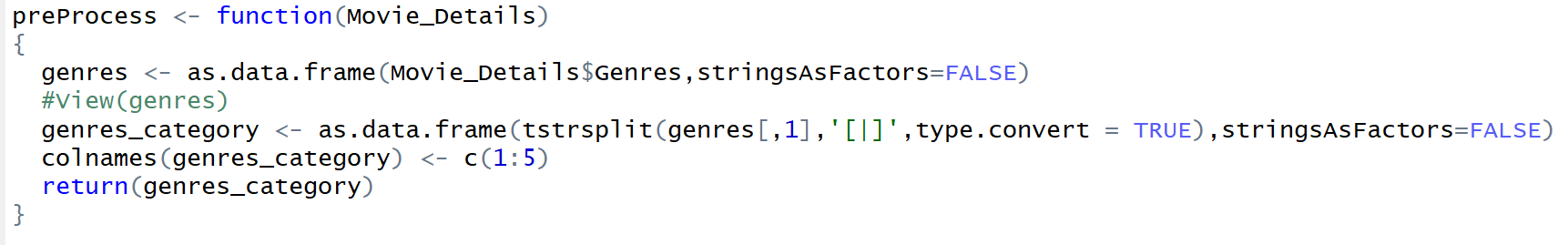
# **Screenshot**

1. Import Data



1. Data Processing

To obtain the movie features matrix, the pipe-separated genres available in the movies dataset had to be split. The data.table package has a tstrsplit() function that works well here to perform string splits.

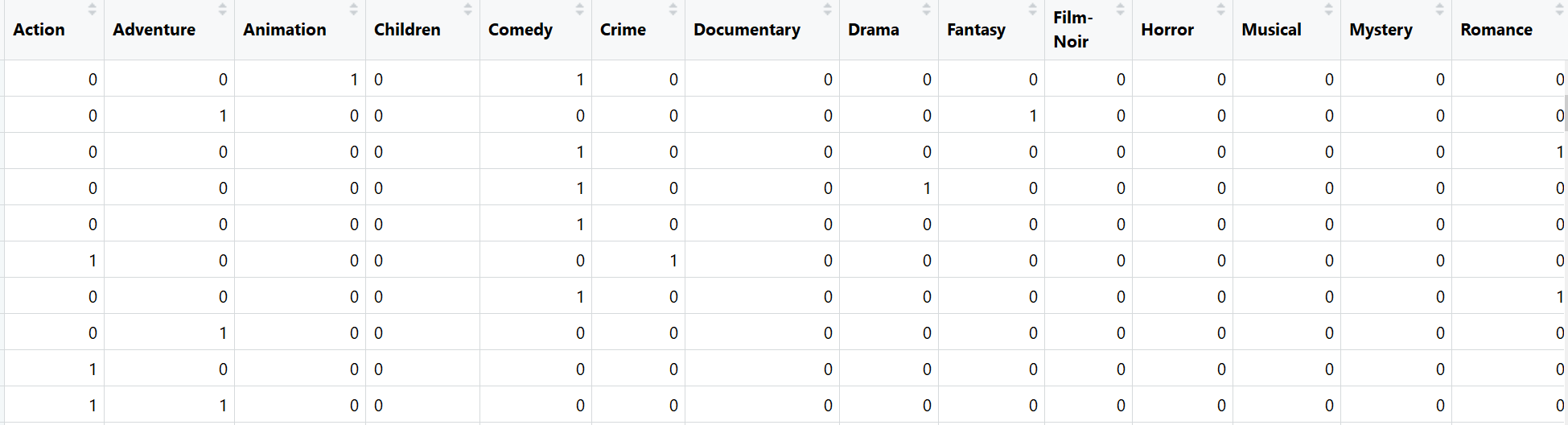




1. Genre Matrix

matrix with columns representing every unique genre, and indicate whether a genre was present or not in each movie.

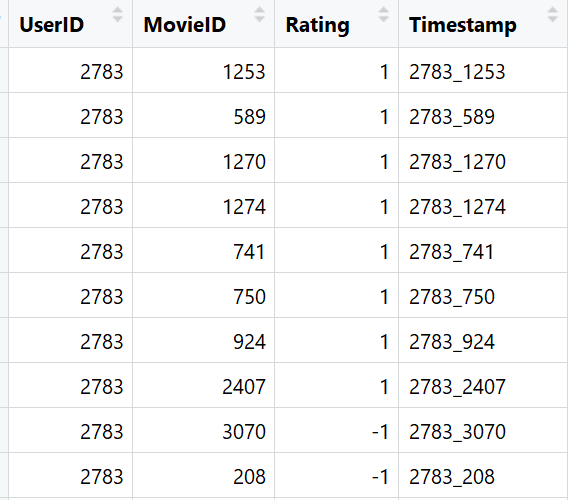




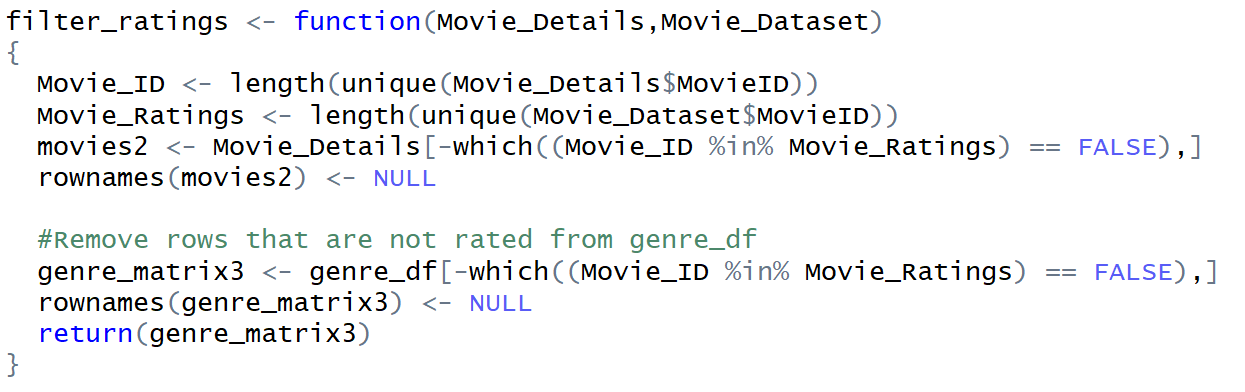
1. Binary Rating

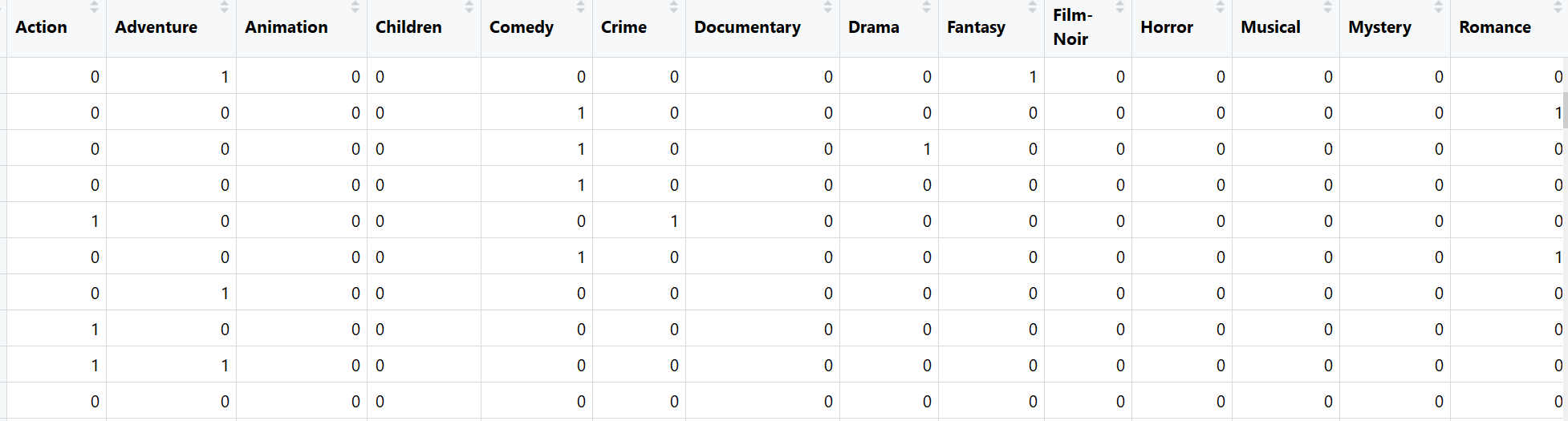
This can be easily done with the dcast() function in the reshape2 package. Ratings of 4 and 5 are mapped to 1, representing likes, and ratings of 3 and below are mapped to -1, representing dislikes.





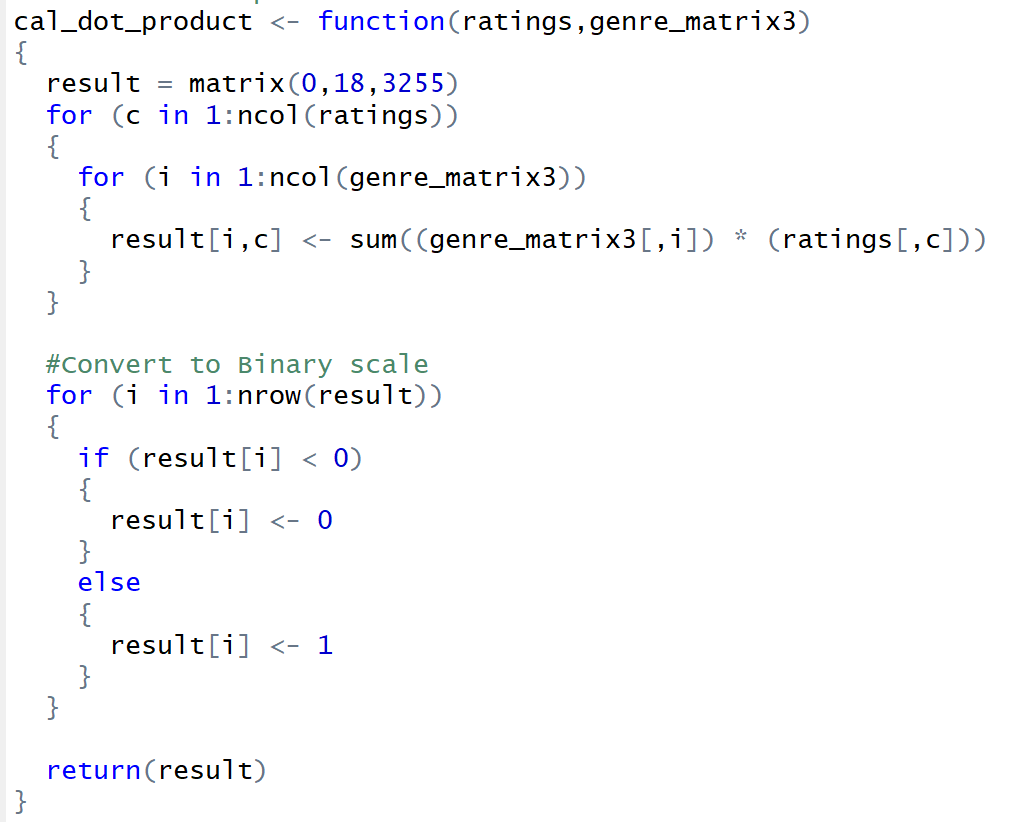
1. Genre Matrix (Binary Format)

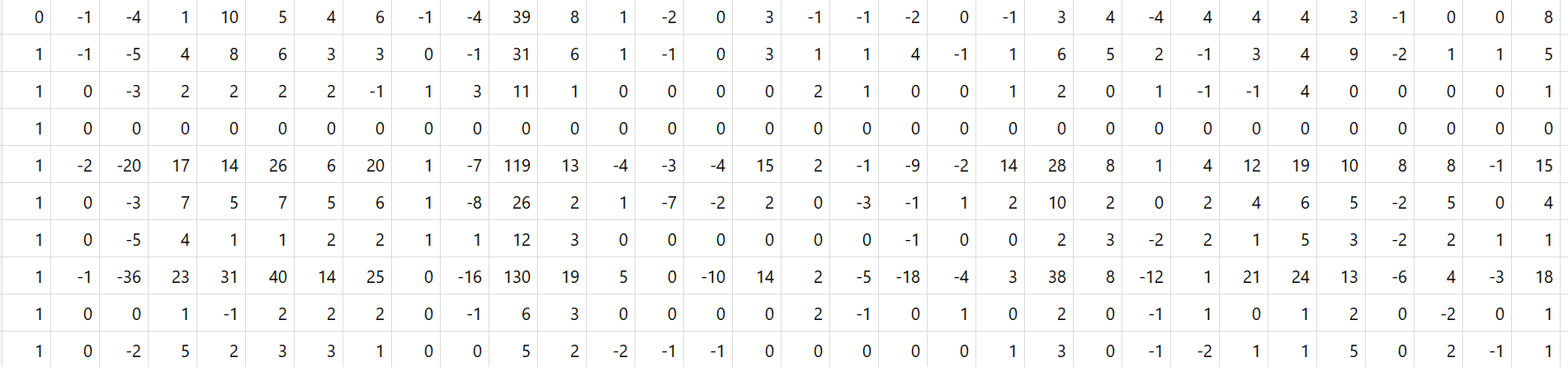




1. Dot Product

calculate the dot product of the genre matrix and the ratings matrix and obtain the user profiles.





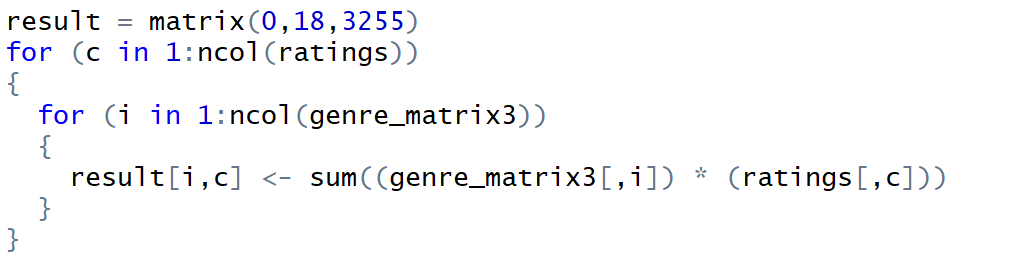
1. Movie Recommendation –

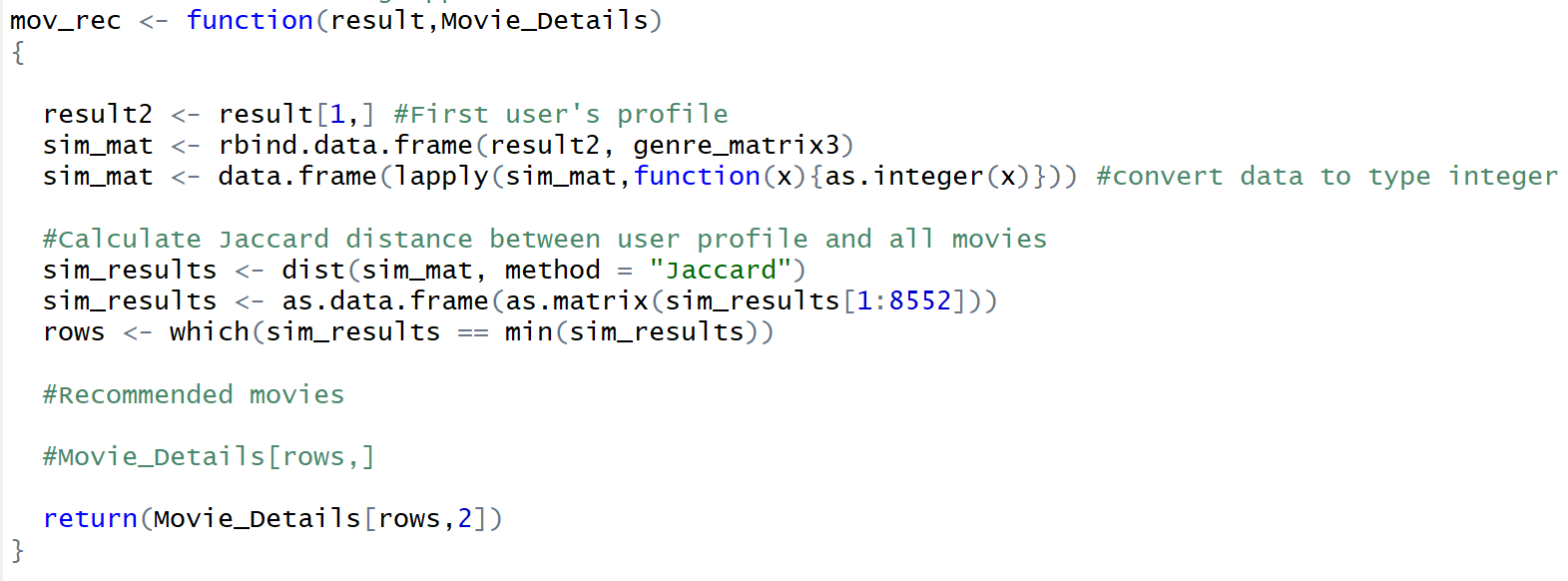
* Predict if a user likes an item based on the item descriptions (movie genres). This can be done by predicting user movie ratings.
* Assume that users like similar items, and retrieve movies that are closest in similarity to a user’s profile, which represents a user’s preference for an item’s feature.

1. Content Based Filtering

Using Jaccard Distance to measure the similarity between user profiles, and the movie genre matrix. Jaccard Distance suitable to analyze binary data.

Used the dist() function from the proxy library to calculate Jaccard Distance





Few recommendations for the first user



* User Based Content Filtering

The User-Based Collaborative Filtering approach groups users according to prior usage behavior or according to their preferences, and then recommends an item that a similar user in the same group viewed or liked.

Using User-Based Collaborative Filtering to generate a top-10 recommendation list for users using the [recommenderlab](http://cran.r-project.org/web/packages/recommenderlab/index.html) package available in R.

* Creation of the Recommender Model

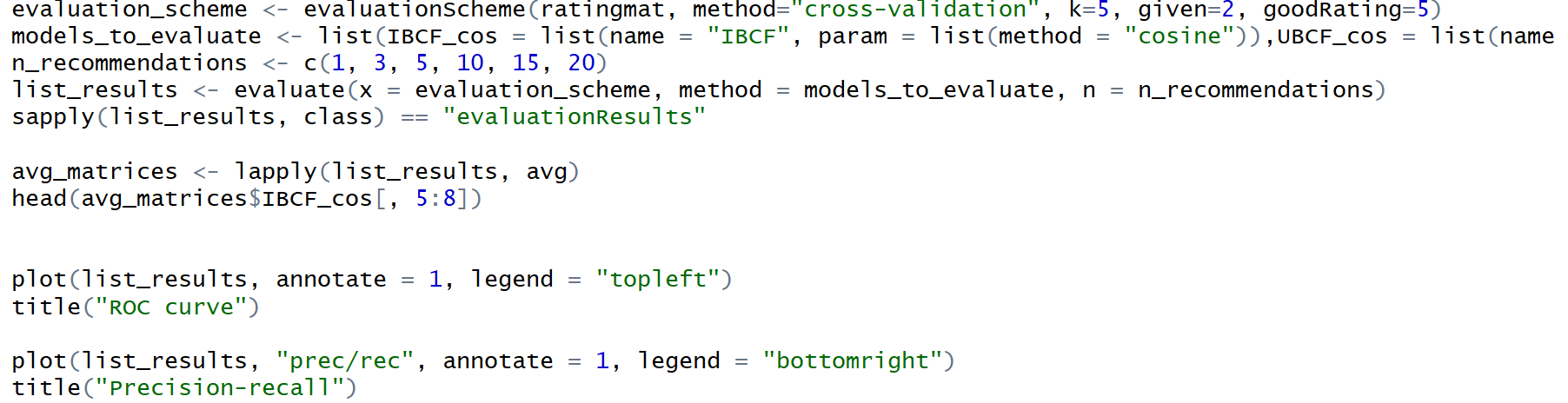
The User-based Collaborative Filtering recommender model was created with recommenderlab with the below parameters and the ratings matrix:

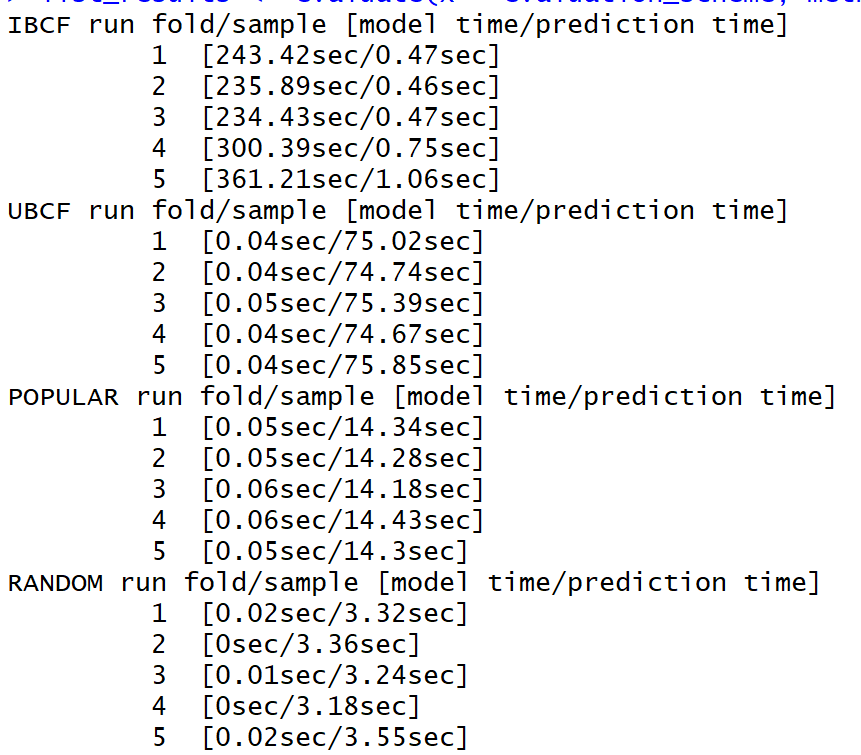
Method: UBCF  
Similarity Calculation Method: Cosine Similarity  
Nearest Neighbors: 30



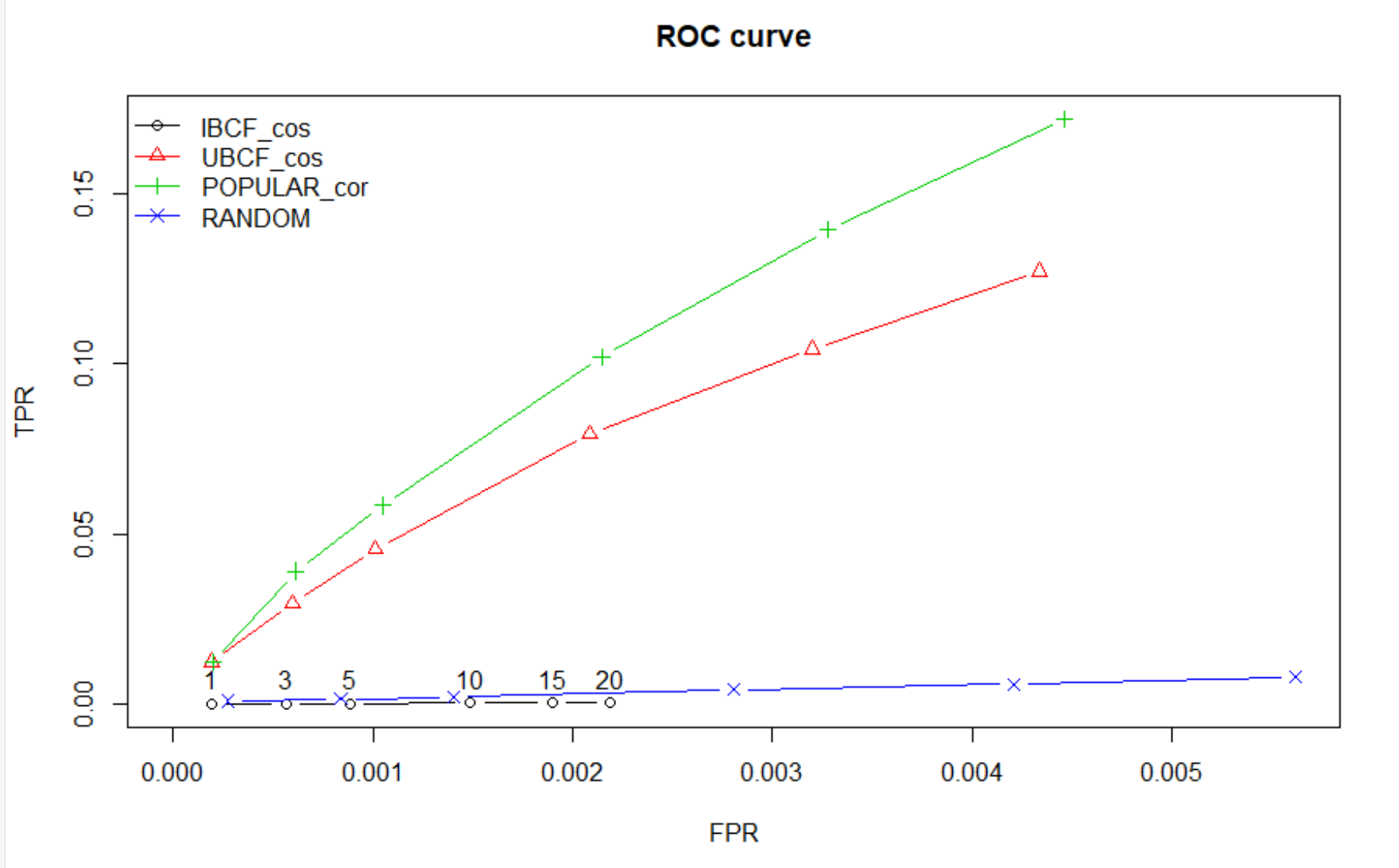


* Result Evaluation among different approaches to recommendation system





* ROC curve



* Precision-recall

