Movie Recommendation System

# Introduction

Recommender systems are everywhere in our day to day lives. They basically aim to help the users in finding items that they would like to buy/consider based on amounts of data collected. Big Giants such as Amazon, Netflix, Facebook, EBay, etc. use these systems every day to help the users in shopping, also viewing the right content.

So, lets setback and think if there were no recommendation systems. Most of the new users would have faced lot of problems in finding the right content/items and it would have been a real difficult for the commercial websites to retain the users.

# How it works

Now that we know about the recommendation system, let’s answer the basic question – How they really work?

Well, parsing a huge amount of data to predict a user’s preference based on other users who are like him/her is what a recommendation system really do. There are number of approaches/models we can apply to a recommender system. Some of them are Collaborative Filtering, Content-based Filtering, Hybrid Filtering, etc.

For this project, I will try both Content-based and Collaborative Filtering to find out which model works better for a movie recommendation system. Again, Collaborative filtering are grouped into three categories:

* Item-based: Algorithm to predict an item based on similarity between items.
* User-based: Algorithm to predict an item based on similarity between users.
* Hybrid – Combining both the similarities of Users/Items to predict an item.

# Dataset

For my project, I have used the MovieLens dataset. It consists of over 1 million movies ratings by users on a 1-5 scale. The dataset is divided into 3 sections:

* Movies – It contains all the movies information.
* Users – It contains all the user’s information.
* Ratings – It contains the Ratings given by the users to different movies.

Let’s do a quick walkthrough on all the fields of these respective datasets:



Movies – The movies dataset has the unique movie id, name and the Genre

Users – Unique User ID, Sex(M/F), Age Group (The Age Group is classified as a single digit. E.g. 1(Under 18), 18(18-24 age), etc. Similarly The Occupation of the users(1 – Academic, 2 – Artist,etc) and the zip code.

Ratings – Consists of UserID/Movie ID/Rating (1-5)/Timestamp.

# Data Cleaning

Now that we have the required dataset, we must do some cleaning. For our Movie Recommendation project, we won’t be needing all the fields in the datasets. So, it’s better to consider only the important fields and get rid of others.

For example, we can get rid of Timestamp column from the Ratings dataset and the Zipcode from the users dataset. Zipcode might be needed in some cases but for this project I am not doing any calculation on Zipcode.

The most important fields overall are – **Movie ID, User ID, Genre, Rating, Sex, Age Group and Occupation.**

Let’s review on the cleaning of each dataset:

**Movies** – Sample Data:



If we see here, The Movie Name is the combination of Name and Year. So, first step was to remove the Year from the Name and creating a new column ‘Year’.

Most of the movies have more than one Genre listed. I separated the combined Genre of the movies to perform some calculations on the Genre. Though this process, increased the size of my dataset but I got very good insight on the Genre classification.

**Ratings –** Sample Data:



There was not much cleaning needed here. Only removed the Timestamp column.

**Users** – Sample Data:



Since the Occupation was having only the ID and no description, I created a new column ‘Profession’ which has the description of the Occupation ID’s.

Removed the Zip column.

After cleaning my dataset, I merged everything and below is the sample of my final dataset:



Note: I added some CSS customizations on the Pandas data table, because of which I am getting the table in a different format.

# Insights And Calculation

Now, that I have my final dataset for my Movie recommendation system it’s time to dive deep and work on my data story and asking the right questions.

Below is the list of questions I came with:

* Calculating the Number of Movies/Users(F,M)/Professions.
* Grouping the List of Genre’s and their respective count.
* Most watched Genre by Male/Female Users.
* What Genre is watched by Users of different Age Groups.
* Mean Ratings of Male/Female Users on Different Genre.
* List of Movies having the highest Mean Ratings.
* Mean Ratings based on Age-Group/ Profession.
* Mean Ratings of a single User.

After asking these questions, I got some good insight on the dataset.

Below is the list of my findings:

* Number of Male users are more than Female users.
* Around 12% of the users are College Student.
* Genre ‘Drama’ is most watched by Male/Female Users, whereas ‘Documentary’ is least watched.
* Mean ratings of Male users are slightly higher than female users.
* Mean ratings of Users who are over 56 is higher as compared to other age groups.
* By the profession, the mean ratings of retired users are more compared to other professions. This makes sense, assuming that most of the retired users must be in the age group of 56.

# Visualization

For this dataset, I was not able to plot the scatter type because it was not making any sense. The same goes for histogram as well.

Bar Graph seemed most appropriate for my dataset.

Below is the list of bar graphs I plotted:

* Different counts of Genre/Movies/Professions, etc.
* Watched Genre of Male Vs Female Users.
* Mean Ratings comparison of Male VS Female Users.
* Mean Ratings of Age Group/Profession.

# Recommendation system, Approaches

* Collaborative Filtering.
* Content- Based Filtering.
* Hybrid Recommender systems.

## Collaborative filtering

Collaborative filtering methods are based on collecting and analyzing a large amount of information on users’ behaviors, activities or preferences and predicting what users will like based on their similarity to other users.

Collaborative filtering assumes that people who agreed in the past will agree in the future, and that they will like similar kinds of items as they liked in the past.

Collaborative filtering systems have many forms, but many common systems can be reduced to two-three methodology, Item based, User-Based and Hybrid(Combination of both User/Item based.

*User-Based:*

1. Look for users who share the same rating patterns with the active user (the user whom the prediction is for).
2. Use the ratings from those like-minded users found in step 1 to calculate a prediction for the active user.

*Item-Based:*

1. Build an item-item matrix determining relationships between pairs of items
2. Infer the tastes of the current user by examining the matrix and matching that user's data

## Content based Filtering

Content-based filtering methods are based on a description of the item and a profile of the user’s preference. In a content-based recommender system, keywords are used to describe the items and a user profile is built to indicate the type of item this user likes. In other words, these algorithms try to recommend items that are like those that a user liked in the past (or is examining in the present).

## Hybrid Recommender systems

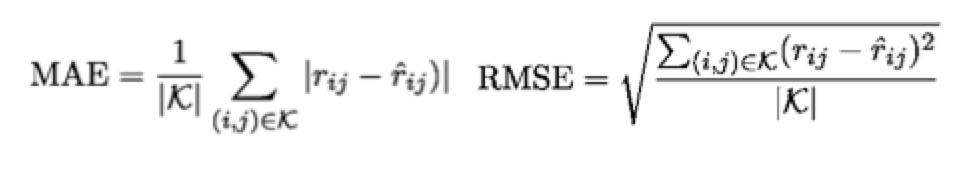
Combining all the known approaches such as Collaborative, Content-based, Demographic, Knowledge-based makes a hybrid model. E.g. Netflix is a good example of Hybrid models. It uses all the attributes and display the recommendations accordingly.

For the Movie Lens dataset, I have used Collaborative Filtering. So, we will be talking more about this.

We will focus on collaborative filtering models today which can be generally split into two classes: user- and item-based collaborative filtering. In either scenario, one builds a similarity matrix. For user-based collaborative filtering, the user-similarity matrix will consist of some distance metric that measures the similarity between any two pairs of users. Likewise, the item-similarity matrix will measure the similarity between any two pairs of items.

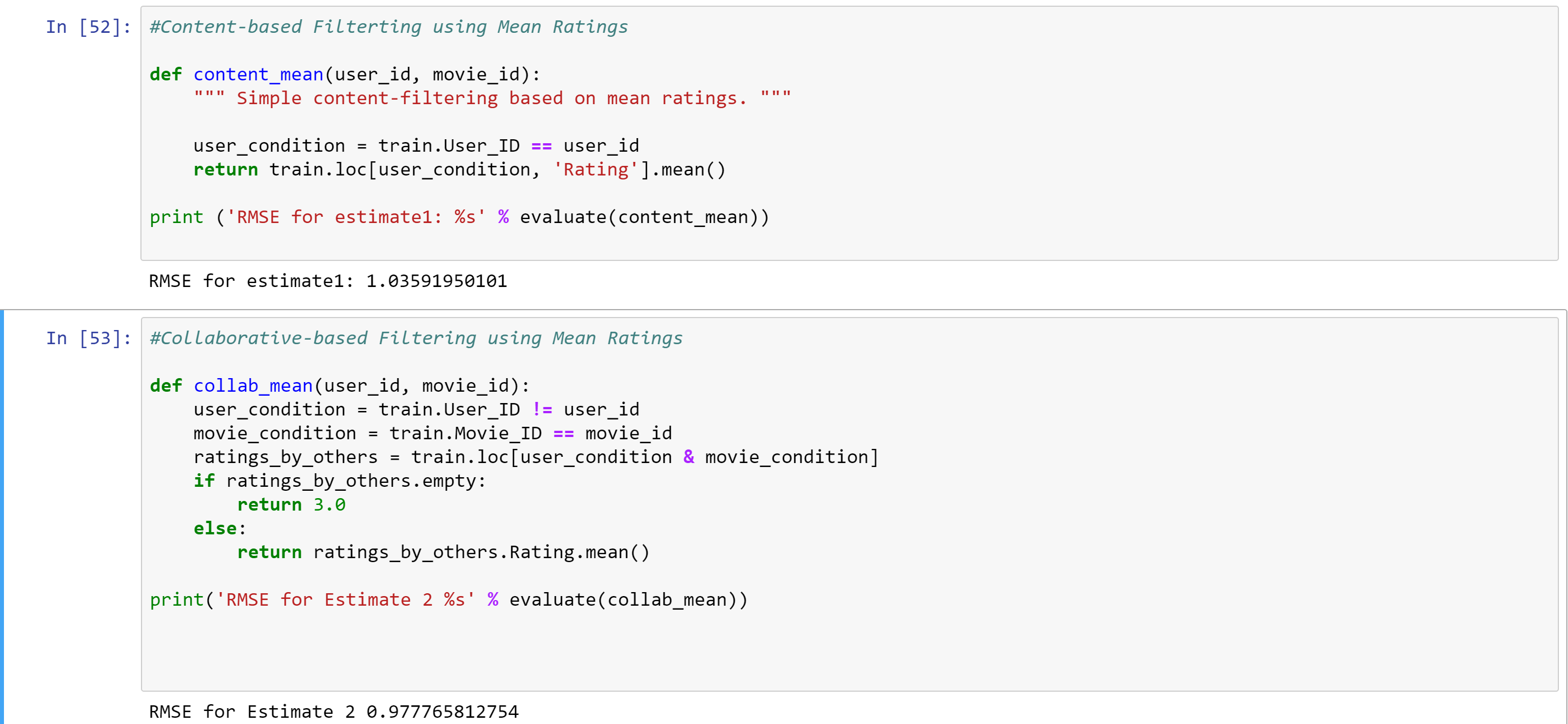
#### Validating Recommender Systems

The performance of the predictive task is typically measured by the deviation of the prediction from the true value. This is the basis for the Mean Average Error (MAE) or the squared version called Root Mean Square Error (RMAE)



In the formulas, K represents the set of all user-item pairings (i, j) for which we have a predicted rating rˆ\_ij and a known rating r\_ij, which was not used to learn the recommendation model. The basic idea behind these metrics is measuring the deviation between your predicted rated values and the real rated values over many users and items.

**Calculating the RMSE for both content and collaborative based filtering**

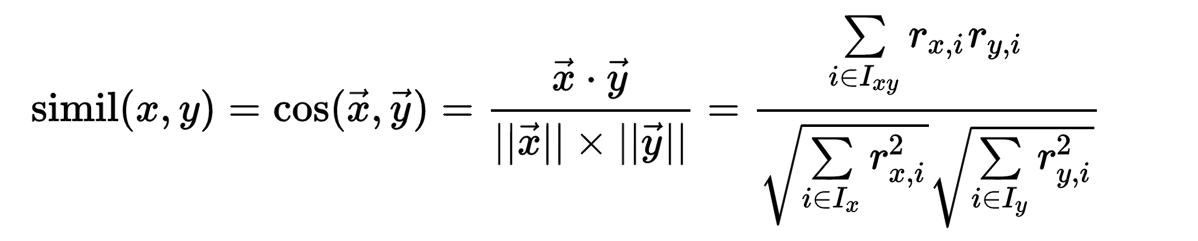


From the above observations, Collaborative-based filtering gave us better results on our training data.

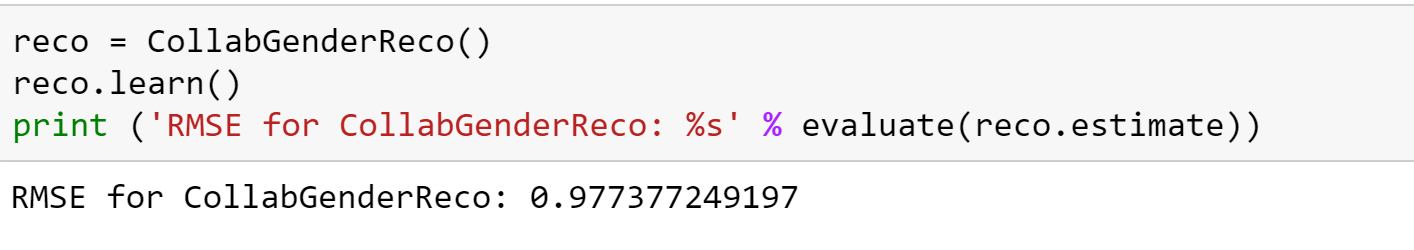
Cosine Similarity:

Cosine similarity is a measure of similarity between two non zero vectors of an inner product space that measures the cosine of the angle between them.

The cosine-based approach defines the cosine-similarity between two users x and y as:



After implying the implicit Sim function on the Gender label, I got some good RMSE value.



# Next Steps

Even though I have a good RMSE value, it can be improved by using other means. I can combine other models and check if it’s improving my score. Other similarity functions can be implemented to produce a good recommender.