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.

# Next Steps

I got some good insights on my dataset. Now it’s time to work on the problem.

First thing I need to do is split my Dataset into Training and Testing datasets. The main purpose of this is to use the Training Dataset to train my Collaborative filtering model for prediction. Then I will take this model and test it on the testing dataset to find out how well it is performing.

I will be using the RMSE(Root Mean Squared Error) concept to find out the mean error in my dataset.

I will try to implement both Content-based and Collaborative filtering on my dataset and calculate the RMSE value. Based on the RMSE Value, I will be implementing that model and train it for prediction.