DSC 680 -Project Milestone 1

Mehtre Vijaykumar

## **Topic – Movie Recommendation**

The goal of this project is to provide a recommendation system for video content providers to predict whether someone will enjoy a movie based on how much they liked or disliked other movies.

### Business Problem

Major companies like YouTube, Amazon, Netflix use recommendation systems in social and e-commerce sites use recommendation system for its users to suggest for an individual according to their requirement more precise and accurate item.

These online content and service providers have a huge amount of content so the problem which arises is which data is required for whom so the problem of providing apposite content frequently. This project represents the overview and approaches of techniques generated in a recommendation system.

Recommendation Systems are classified in mainly three categories as below:

1. Content-based systems
2. Collaborative Filtering system
3. Hybrid recommendation system.

Content based systems works based on the label or genre of an item. If a user watched a movie so it recommends similar movies based on director, a genre, and many more aspects.

The theory behind the collaborative filtering is that if user’s ‘A’ and ‘B’ have rated correspondingly in the past, then there will be an assumption that they will rate correspondingly in the future

Diagram

Description automatically generated

### datasets

I have used movie-lens dataset from: https://grouplens.org/datasets/movielens/latest/

**Data description**

The dataset contains 100k+ ratings and 3k+ tag applications across 9k+ movies. The data was captured for 600+ users between 1996 and 2018. This dataset was generated in September 2018.

### Methods

Recommendation system works on basically on two things - product details and user details. We must collect them from the system or from the database and make decisions based on ratings if similar items were found then it will generate recommendation system otherwise no recommendation system will be generated.

We will apply item based collaborative filtering. The reason behind this is because user taste may change with respect to time, but item doesn’t change it remains same. There are certain stages to make our recommendation system efficiently to respond.

• **Data Loading** – To load the data and display accordingly we must perform some operation like merging the two files in the dataset.

• **Data Slicing** – Here we are removing unnecessary column and data.

• **Data Cleaning** – In the real-world data if we make a table of ratings in the recommendation system, we find that most of the user are not rating the movies and are mostly inactive. The same cases are with movies either users don’t watch or it’s get too old. To make our computation more accurate we will remove such users and movies from our research.

Now to predict similarity we have two methods either we can use correlation method or cosine method.

**Approach:**

* The Simple Recommender offers generalized recommendations to every user based on movie popularity and genre.
* The basic idea behind this recommender is that movies that are more popular and more critically acclaimed will have a higher probability of being liked by the average audience.
* This model will not provide personalized recommendations based on the user.

**Implementation:**

* The implementation of this model is extremely trivial.
* All we have to do is sort our movies based on ratings and popularity and display the top movies of our list.
* As an added step, we can pass in a genre argument to get the top movies of a particular genre.

### Ethical Considerations

Users for all the reviews in the dataset were selected at random for inclusion. All selected users had rated at least 20 movies. No demographic information is included. Each user is represented by an id, and no other information is provided.

### Challenges

**New User**: A newly released movie cannot be recommended to the user until it gets some ratings. A new user or item added based problem is difficult to handle as it is impossible to obtain a similar user without knowing previous interest or preferences.

**Synonymy** arises when a single item is represented with two or more different names or listings of items having similar meanings, in such condition, the recommendation system can’t recognize whether the terms show various items or the same item.

### references

[1]Herlocker, J, Konstan, J., Terveen, L., and Riedl, J. Evaluating Collaborative Filtering Recommender Systems. ACM Transactions on Information Systems 22 (2004), ACM Press, 5-53.

[2] Koren, Yehuda. “Factorization meets the neighborhood: a multifaceted collaborative filtering model.” In Proceeding of the 14th ACM SIGKDD international conference on Knowledge discovery and data mining, 426–434. ACM, 2008.

[3] <https://www.mygreatlearning.com/blog/masterclass-on-movie-recommendation-system/>

[4] <https://docs.microsoft.com/en-us/dotnet/machine-learning/tutorials/movie-recommendation>

[5] MovieLens 2018 Introduction-to-Machine-Learning https://github.com/codeheroku/Introduction-toMachine Learning/tree/master/CollaborativeFiltering/dataset