Recommendation Systems

Background

Whenever you watch some movies on a website, you will find that the website can suggest so many other movies', among which you even don't know why it wants to recommend such kind of movies to you. Actually, all those suggestions that you have never seen come from recommendation systems. The system will first collect your information, including watching history, movie rating, etc, and then find similar audiences or similar movies. After that, it can provide suggestions based on other audience and movies to expand the range of ones you may be interested in. This report aims at digging out different kinds of recommendation systems and describe basic principles behind them. The data comes from Kaggle and covers anime information.

Popularity Recommendation System

The simplest recommendation can seize general people's favor and provide recommendations satisfying most of audiences' likes and dislikes. Generally, the higher the rating of one movie, the more popular it is. However, some movies may just have a small number of audiences to give ratings. Though they have higher ratings with 9-10, they cannot be regarded as popular ones. Therefore, my popularity recommendation system applies another more accurate measurement to get movies' scoring. This following function:

$$\textit{Weighted Rating (WR)} = v*R/(v+m) + m*C/(v+m)$$

takes the number of ratings for the movie (v), the minimum ratings required to be listed for the movie (m), the average rating of the movie (R), and the average rating of all movies (C) into consideration. By using WR, we can get popular movies with higher accuracy.

Content-Based Recommendation System

The content-based one will use different movies' description to find similar ones. In the respective notebook of this report, movies' genres are extracted from original dataset and work as the standard to measure similarity. For building content-based recommendation system, the most important and also most difficult part is how to transfer texts to some numbers or vectors that can be recognized by machine learning models. The notebook leverages the **TF-IDF** method to transform movies' texts and then assign vectors to different movies. TF-IDF stands for term frequency-inverse document frequency. The former measures how frequently a word occurs in the document and the latter shows how important a word is. After TF-IDF, **cosine similarity** will indicate how similar between one movie and others. When customers input the movie they want to watch, the system will also show other similar movies that they may want to enjoy.

Collaborative Filtering Recommendation System

This kind of system can be divided into two individual types, user-user, and item-item. The user-user means that by checking the similarity of various users based on their ratings or comments to a series of movies, the system can find several users that are allied to one target customer. And if you want to predict the ratings for some unseen movies of that customer, you can refer to his or her similar users' who have seen those movies and find movies with higher predictive ratings. And the item-item way applies the same way as user-user while uses the similarity of different items based on various users' ratings. The collaborative filtering recommendation system is actually the most general one in practice. What's more, the notebook makes use of **Surprise** package in Python to realize user-user recommendations.

Conclusion

Different kinds of recommendation systems will provide suggestions from totally different aspects, but everyone is useful and can provide valid information. For the first system, the top 5 movies appear as follows:

	anime_id	name	genre	type	episodes	rating	members
0	5114.0	Fullmetal Alchemist: Brotherhood	Action, Adventure, Drama, Fantasy, Magic, Mili...	TV	64	9.26	793665.0
1	9253.0	Steins;Gate	Sci-Fi, Thriller	TV	24	9.17	673572.0
2	4181.0	Clannad: After Story	Drama, Fantasy, Romance, Slice of Life, Supern...	TV	24	9.06	456749.0
3	2904.0	Code Geass: Hangyaku no Lelouch R2	Action, Drama, Mecha, Military, Sci-Fi, Super ...	TV	25	8.98	572888.0
4	199.0	Sen to Chihiro no Kamikakushi	Adventure, Drama, Supernatural	Movie	1	8.93	466254.0

For the latter two kinds of systems, they realize interactive recommendations. The content-based system can help users find more movies they will be interested in whenever they input one movies' name, while collaborative filtering system can provide a given number of suggestions for a specific user. The following picture just shows the top 5 suggested movies for user2:

	anime_id	name	genre	type	episodes	rating	members
0	11061	Hunter x Hunter (2011)	Action, Adventure, Shounen, Super Power	TV	148	9.13	425855.0
1	16498	Shingeki no Kyojin	Action, Drama, Fantasy, Shounen, Super Power	TV	25	8.54	896229.0
2	9253	Steins;Gate	Sci-Fi, Thriller	TV	24	9.17	673572.0
3	12355	Ookami Kodomo no Ame to Yuki	Fantasy, Slice of Life	Movie	1	8.84	226193.0
4	1535	Death Note	Mystery, Police, Psychological, Supernatural, ...	TV	37	8.71	1013917.0

Improvement

Seems that the best way to provide recommendations is to combine these three types of systems and give comprehensive suggestions. It needs me to spend much more time to find a proper method to combine them and get a better model.

(Kaggle: https://www.kaggle.com/zhenyufan/build-up-recommendation-system)