DSCI351 PROJECT

Anime Content-based Recommender System

Team 3: Harry Zhao, Yi Pan

Introduction

Since its inception, anime has always had a huge fan base due to its exciting stories and beautiful animation. The end of each anime is like the end of a part of one's life, and we can't forget it. When our favorite anime comes to an end, we cannot wait to explore other similar works and continue our anime journey. Our Anime Content-based Recommender System was created to solve the problem of anime recommendations.

Dataset

For our group's project, the dataset we chose is a public dataset on the topic of Anime that is open source on the kaggle platform. The link to this public dataset is https://www.kaggle.com/hernan4444/anime-recommendation-database-2020.

This is a public dataset about Japanese anime and the ratings of users who watched it. Each user can rate the anime they have watched, and this dataset is an aggregate of those ratings. The dataset is divided into five CSV files, but we only use three of them: anime.csv, anime_with_synopsis.csv, and rating.csv. anime.csv is a collection of data about all anime. anime_with_synopsis.csv is a collection of data that describes rating.csv is a collection of all data about each user's rating.

We merged these three files and extracted six of them as our features analysis data:

MAL_id: This id is a unique id in myanimelist.net, each id corresponds to a separate anime for identifying anime. Int data type.
English Name: The full name of the anime described in English. String data type.
Genre: the genre used to describe the anime. Each anime can have more than one genre, and if there are more than one genre, they are separated by commas. String data type.
Type: the type of the anime. For example, an anime can be a movie, a series of TV shows or an OVA and so on. String data type.
Producer: the anime is created by which anime company. String data type.
Synopsis: used to briefly introduce the story of each anime. String data type.

And we selected the rating data of specific users in rating.csv to judge the goodness of our recommendation results based on the rating of the recommended anime by the users.

Our System

We build a content-based recommendation system, and the structure of our system is rough, as shown in Figure 1. First, we extract the five features we need to use, namely English name, genres, type, producer, and synopsis, and then preprocess these features to make them more useful and reduce the impact of errors on the similarity calculation. Afterward, for the five processed features, we combine them to form a new feature called soup. For the newly generated features, we apply word2vec to downscale them since we have a total count of more than 40,000 to perform the similarity calculation better. We also used a counter vector to compare the excellence of both recommendation structures. After that, we used the generated word vector for similarity calculation to get the cosine similarity value between each anime, then combined with the user id to select a user's favorite anime as the target anime, find the corresponding id of the anime for recommendation, and come up with the most relevant ten animes, as well as each anime's corresponding genres and the rating value of the anime by the selected users.

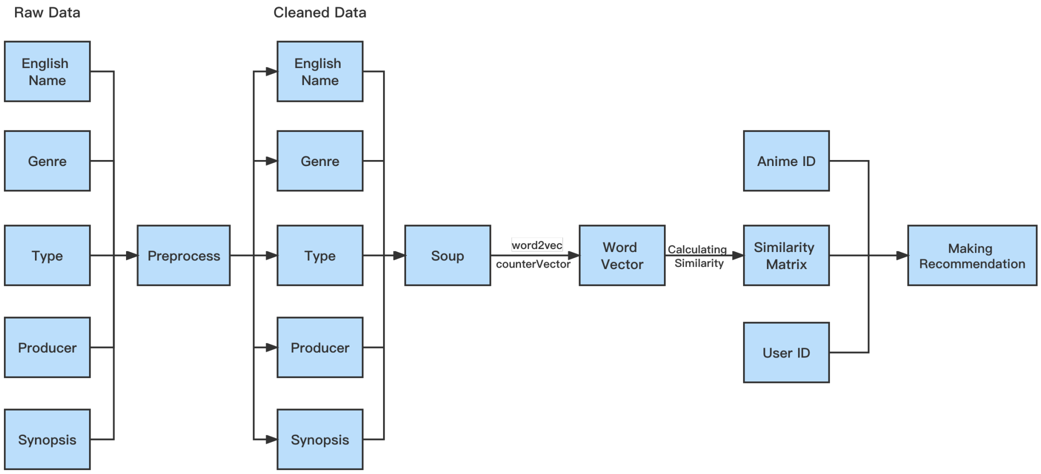

Figure 1 System Structure

Methods

Overall, our recommendation system uses a content-based recommendation system because it has the advantages of no cold start and sparsity problems, the ability to recommend new or less popular items, and high interpretability of results. In our content-based recommender system, we first preprocess the data using English-based stopwords to remove non-textual data such as spaces and punctuation marks, then normalize the feature data. In building the recommendation system to calculate the similarity, we use cosine similarity to calculate the similarity between each anime. In the similarity calculation, we chose two embeddings, CountVectorizer and word2vec to obtain different word vectors. CountVectorizer counts all the text words and calculates the similarity based on the frequency of each anime's text words in all text words. In contrast, word2vec will classify similar words by performing contextual analysis. For example, study and learn are essentially one kind of word, which will be considered the same in word2vec, while they are two different vectors in the CountVectorizer. When performing similarity calculation, the similarity calculated by the two methods will be different.

Results

For our recommendation system, we choose "*great teacher onizuka*" as the loser as an example, and recommend 10 animes that our selected user has rated based on this anime. Our output is the names of these ten animes, the genres, and the user's ratings for each anime. "*great teacher onizuka*" is an action comedy anime about a teacher, and most of our recommendations contain words like classroom or teacher, indicating similarities between them. Also, for the first three anime, we looked at their genres and found that they all contain at least one of school, action and comedy, which is related to the genre of "*great teacher onizuka*". The first anime recommended is a very famous one in the industry about the genre of teacher or school, and even anime lovers who are not interested in this genre will watch this famous anime, which means that the recommendation is very meaningful. We found that the ratings of the ten recommended anime were all higher than 6 except for one with a rating of 3, which means that it is the anime that the user likes, confirming our recommendation's effectiveness.

Evaluations

Evaluation	CountVectorizer	word2vec
A/B Test (20 Participant/5 Anime)	3.2	3.6
Spearman Rank Correlation	0.147	0.374
Diversity (Recommend Genre/Total Genre)	42.222%	46.667%

Table 1 Evaluation of Our System

We used three methods to evaluate our recommendation system. First, we interviewed 20 anime lovers for feedback on the results of our recommendations. For the two recommendation systems using CountVectorizer and word2vec, the results showed that of the ten anime recommended for their favorite anime, about 3-4 anime were considered relevant to their target anime, and they would consider watching it in the future. Comparing the genres of the target anime with the ten recommended animes, we found that the genres of the recommended animes were expanded from the original genres. Out of 45 genres, the genres recommended by word2vec accounted for 46.667% of the whole genres, and the CountVectorizer one accounted for about 42.222%. Finally, we use spearman to evaluate the ranking of the recommended movies and show that the spearman correlation of word2vec is 0.374, which is higher than that of the CountVectorizer at 0.147, so we can get better anime recommendations by using word2vec.

Limitations and Projections

The limitation of our project is that only one anime is recommended based on one user's favorite anime, which only focuses on the anime itself and ignores the user's tendency. Also, when combining all the features, the number of texts in the Synopsis of each anime is much larger than other features, such as genres, English name. If we assign reasonable weights, we can get better recommendation results. Therefore, in future work, we consider feature engineering to calculate the corresponding weights of each feature and then combine them. At the same time, we will increase the number of anime input for the recommendation, considering the first three or five animes that a user likes and recommending ten movies related to the user's preferences through multiple animes to improve our recommender system. In the future, we will further increase the use of User Profile, considering the content of the anime itself and taking into account the user's profile when making recommendations.

Conclusion

In this project, we learned and explored the building and improvement of content-based recommender systems, compared two different embedding approaches, and selected the one that performed better in real-world tests as the treatment of our model. Although our model still has some shortcomings and limitations, we have gained a lot from the process.