ANIME RECOMMENDATIONS DATABASE

V Sai Akshith
CSE student
Of PES University
Akshith.vankayala@gmail.com

Akarsh siwach
CSE student
Of PES University
akarshaik10@gmail.com

Neha Beru
CSE student
Of PES University
nehaberu9@gmail.com

D. Lasya Priya
CSE student
Of PES University
lasyad2k01@gmail.com

I. Introduction and background:

Anime is a hand-drawn computer activity started in Japan which has drawn a lot of followers around the world.

Recommender systems are used to design recommend items to the user based on various factors. These systems predict the most likely item that the users are most likely to purchase and are interested in. Industries and Companies like Netflix (it recommends movies to their users), Flipkart, Amazon, etc. use recommender systems to help their users to choose the correct item or movies for them.

The recommender system deals with a large volume and variety of information by filtering the most important information based on the data provided by a user and other different factors like taking care of the user's interest. It also finds the best match between user and item and computes the similarities between users and items for recommending.
Recommendation system can be build using multiple approaches and models like Content based approach, and collaborative approach etc.

The recommendation system helps users to predict items that might be useful for the user. In terms of anime, anime versions are increasingly appearing with an interesting story and series quality. This will make it difficult and create confusion for the users to choose which movie or series to watch first.

Through this anime dataset, we plan to build and analyse different kinds of recommendation models and check their accuracy and choose the best model.

Previous work:

A brief review of only the most relevant predecessor work:

This data set contains information on user preference data from 73,516 users on 12,294 anime. There are 2 datasets i.e., anime.csv and rating.csv. The anime.csv has 7 columns

- anime_id unique id identifying an anime.
- name - full name of anime.
- genre - comma separated list of genres for this anime.
- type - movie, TV, OVA, etc.
- episodes - how many episodes in this show. (1 if movie).
- rating - average rating out of 10 for this anime.
- members - number of community members that are in this anime's "group".

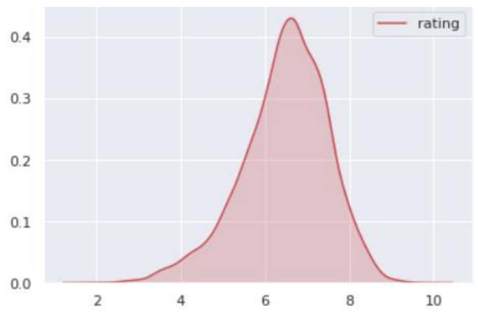
The rating.csv contains 3 columns

- user_id - non identifiable randomly generated user id.
- anime_id - the anime that this user has rated.
- rating - rating out of 10 this user has assigned (-1 if the user watched it but didn't assign a rating).

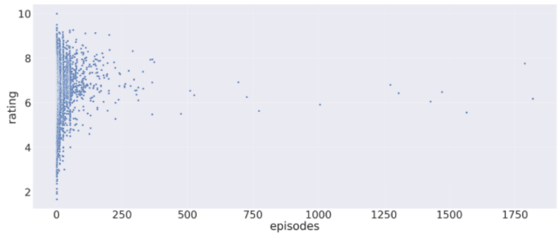
Pre-processing Data

We have done EDA and Visualization and gained some useful insights. Some examples:

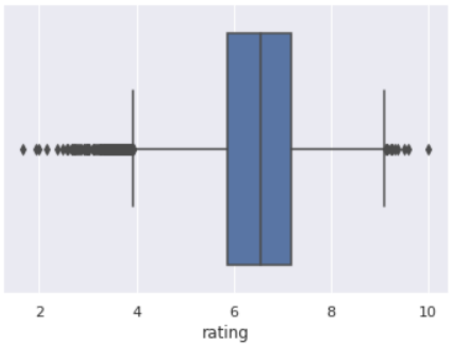
Rating Distribution: Plot shows that most common raing is between 6 and 8.



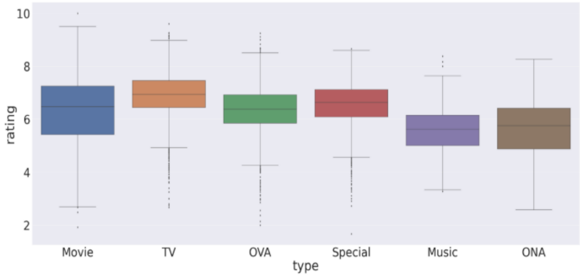
Scatter plot: That is plotted between rating and episodes shows that a large number of animes with lower number of episodes have greater variation in rating.



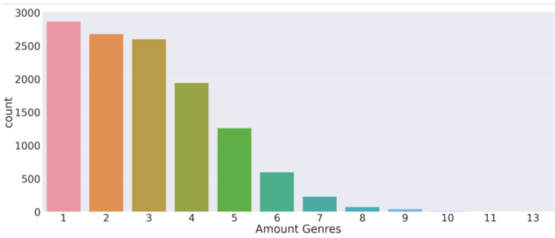
Boxplot: On rating column shows outliers based on ratings received per anime



Boxplot: On the type of anime shows that types ona and movie have the most regular behaviour.



Bar graph: between count and amount Genres shows that there is high density until 7 after witch number of genres decrease drastically.



Data Cleaning:

Cleaning the anime and ratings datasets.

Exploratory Data Analysis:

Exploring the data and trying to extract insights and patterns before building the recommendation system.

Literature survey

We have investigated multiple recommendation models from different research papers on the on anime dataset to find the best way to build a model.

These various models helped us understand the difference between each other and also understand what components we need to mainly consider.

Below we have mentioned a few models which we think are the most relevant along with their authors and a summary about the model: -

1. Collaborative Recommendation System in Users of Anime Films A S Girsang1* , B Al Faruq1* , H R Herlianto1* and S Simbolon1*
   Summary:
   A better anime recommendation system based only on user watch history. A simple recommendation system can measure the similarity between performances, users and help predict whether users will enjoy certain anime. However, due to limitations calculating very large data sets, 100k data sets selected. Using a much more powerful machine, if a 1 million data set could be experimenting with. This paper shows a very simple but efficient recommendation system.
2. AniReco: Japanese anime recommendation system
   Hayaki Kawata, Masahumi Muta, Soh Masuko, Jun'ichi Hoshino
   Summary:
   "AniReco" has been proposed which reflects users' potential preference. Consequently, recommendation presenting method using a network diagram has obtained a high evaluation in system usability clarifying that the system has an effect to reduce burden for animation works exploration and leads to new discovery of works as well.

What limitations have you identified that you seek to address in your work?

One of the biggest disadvantages of a content-based filtering algorithm is that it may only promote items of the same category. It will never propose things that the user has previously seen and enjoyed. As a result, if a user has only viewed or loved action movies in the past, the algorithm will only propose action movies to them. It's a pretty specific approach of constructing an engine.

The cold start problem is a common problem induced by data sparsity. Because collaborative filtering techniques promote items based on a user's prior preferences, new users will need to evaluate a large enough number of items for the system to effectively capture their preferences and provide credible recommendations.

With a problem presentation that it has become difficult in recent years to find animation works that correspond to individual preferences among vastly increased works as well as to recognise expanding related contents in a cross-sectional manner, an animation work recommendation system has been developed.

There are various challenges in recommender systems that may be revealed to improve their efficiency and performance, such as addressing data sparsity.

What assumptions have you made regarding the data/issue area/scope of the problem you're attempting to solve?

Recommender Systems are programmes that attempt to forecast a user's preferences and suggest items that are likely to pique their interest. They assist users in making decisions by introducing them to fresh and relevant stuff. As previously stated, we examine the work of three different types of recommenders.

We'll test a variety of recommendation systems, including collaborative, content-based, hybrid, and knowledge-based, to determine which model is the most accurate.

We'll divide the data into training and test sets first, so the recommender algorithms can learn the data before attempting to predict relearned outcomes.

Proposed solution-

An overview of the various components of your solution. We have built recommendation system that could act both as a means of exploring new anime genres and titles.

Models used:

Constraint based model

Since the dataset is very huge and there are many genres one can choose from. The choices can be limited to by putting a few constraints in the model i.e., rules. We take in the users' preferences on certain attributes like 'rating' and then reduce the data frame into a smaller piece. After this we find the IMDB rating for every single anime in the reduced set and sort it in the ascending order. This gives us all the anime which are currently popular and follow the user specified constraints.

Collaborative filtering method using SVD

Here we use SVD provided by the surprise module and we cross validate on a subset of the data frame i.e., the user id, name, user rating. Through each run of the epoch the RMSE is reduced. The model is now fit on the data frame and gives out predictions based on the user id and predicts the rating the user would give the anime. The data frame is sorted based on the estimated value and the top few anime's are shown as recommendations.

Hybrid recommendation system

This system uses SVD like the above recommendation system but, in case of this system the input is not just the username but instead it also takes an anime name similar to which you want your recommendations. This uses A similarity data frame which denotes the similarity between each and every anime. The similarity data frame is formed using the description of the anime (genres, episodes, type of anime). This gives out a data frame of anime names their ids and the estimated rating the user will give them. The data frame is then sorted based on the estimated rating the user would give and the top few animes are shown (The rating given to the recommended animes cannot cross the rating of the anime which has been used as the base for the similarity)

Collaborative filtering

Collaborative Filtering collects and analyses past user behaviour, and predicts what users like based on their similarity to other users. The accuracy can be improved with larger dataset on users/ items. With Collaborative Filtering we may discover the possible interests of users. Gives recommendations based on the preferences of other similar users.

User-based recommendation

Gives recommendations based on the similar users to the current user. Recommendations are made by finding the average rating of all the shows rated by these similar users and chooses the n-top rated shows among all these shows.

Similar users are found by filtering out all redundant data and then forming a matrix containing cosine similarity values between current user and every other user in the dataset.

Advantages of this model:
No prior domain knowledge is required as the embeddings are automatically learned

Disadvantages of this model:

It cannot handle new items very well. This is because of what is called a cold-start problem. It is also hard to include side features for queries/itms in this model.

Content-based recommendation system

The Systems that recommend an item to a user is based upon the description of the item and a various users having different interests. This model gives recommendations based on the preferences of other similar users. The basic idea is by using genre and type columns to find the similarity between anime and using that we recommend it to the user.

We use cosine similarity to find the top recommended anime based on the given anime. The model suggests other anime based of the anime name passed inside the main function. This model recommends items which are similar to the ones that a user has liked in the past. A content-based recommender works with data that the user provides

Drawbacks of this model:

This model can only make recommendations based on existing interests of the user. Basically, it means the model is limited ability on users existing interests.

Item-based recommendation

Gives recommendations based on the similarity between items and the rating history of the user.

Conclusion-

Developing a hybrid recommendation system that combines content-based filtering and collaborative filtering could potentially benefit from both the representation of content and the commonalities across users, overcoming the restrictions.

Each model devised here has a separate use, for example the content-based filtering gives best anime to watch given only one anime, the constraint-based model gives a starting point for a person, user – user model gives best anime to watch based one 2 users similarity.

Therefore, there are no good or bad recommendations, everything depends on the users taste and choice and every model is useful in a different situation

Contributions of the team members:
V Sai Akshith – collaborative filtering with SVD, Hybrid recommender system, Report
Lasya Priya – User-User similarity-based collaborative recommender system.
Neha beru - Content based recommender system, Report, Research.
Akarsh Siwach – Visualization, Popularity based recommender System.

References

https://michael-fuchs-python.netlify.app/2020/10/01/recommendation-systems-knowledged-based-recommender/#build-the-knowledged-based-recommender

https://towardsdatascience.com/3-approaches-to-build-a-recommendation-system-ce6a7a404576

https://surprise.readthedocs.io/en/stable/matrix_factorization.html

https://analyticsindiamag.com/hands-on-guide-to-recommendation-system-using-collaborative-filtering/

https://medium.com/kunalrdeshmukh/collaborative-filtering-in-recommendation-systems-2fa49be8f518

Collaborative Filtering Recommender Systems J. Ben Schafer, Dan Frankowski, Jon Herlocker , Shilad Sen

Knowledge-based recommender systems Robin Burke Department of Information and Computer Science University of California, Irvine