

evaluation

August 12, 2025

0.1 Evaluation

0.1.1 Results

Revisiting the business objectives: 1. To provide a more streamlined experience for anime consumers looking to choose the next series to pick up based on their own preferences and preferences of users who liked similar anime.

2. To enable recommendations to be made to potential anime converts who may be new to the medium and wish to watch something with a particular storyline as their gateway anime.

We see that the models association rule mining models and the clustering models would provide a solution for objective 1 and the topic modelling model would provide a solution for objective 2.

Of the 2 association rule mining models, the one at support 0.2 has higher metrics for precision, recall and coverage. It is worth noting that the coverage metric is very low. This indicates that the models primarily recommend the most popular anime.

For the clustering models we see that the precision and recall metrics fall as the number of clusters increase. There is a significant drop at cluster 5. Cluster 2 has the highest values for precision and recall, but the cluster quality scores such as inertia, silhouette and davies-bouldin show that the most appropriate number of clusters is 4. Visual inspection of the clusters also indicate that 4 is the optimal number.

The predictive power of this model is considerably lower than the association rule mining, but it makes up for it in the way it provides more coverage of the entire anime catalogue. Therefore a combination of these 2 models will be used. First recommendations will be made based on association rules and if there are no recommendations, the clustering model will be used as a fallback.

AS for the topic model, the one that made use of stop words based on the frequency of the words in the reviews produced better topics and the results show in the plots of the similarity distribution as well as the manual checks.

0.1.2 Review

The data was collected from 3 sources mainly and then supplemented by calls to the unofficial API. The dataset all being from the same base source of myanimelist limited the amount of data that was gathered. For one the ratings from the earlier dataset were completely disregarded because of the possibility of overlap in user ratings. The seemed the most expedient course to take. A more thorough approach could have been taken which was to find the average of the scores of ratings that had the same user and anime id. This however could have a different problem. An assumption

would have to have been made that the user ids are in fact of the same users and that the ids weren't updated between datasets.

The focus of this data mining task was on the user ratings dataset. The use of clustering to do content based filtering on the anime is another route that was not explored. The vast majority of the data in those other anime fields were left untouched without gaining any insights from them. They were cleaned and prepared but ultimately a place for features such as genre and episode count could not be found.

The title column was not given much attention in terms of cleaning. None of the models used it, so some work may still need to be done there. Additionally, many of the anime that were sequels did not have a synopsis that was vastly different to the original. These could have been identified and their ratings put towards the original. The assumption being here that the user would have little trouble finding the sequel for a show he enjoyed. If they were removed from the dataset, the association rules would make more varied recommendations.

Also, all ratings were treated the same in this exercise. It would have been useful to incorporate the actual ratings in the models built so as to supply more fine-tuned recommendations which would add more value to the user.

For the topic modelling, further trials with adding to the stopwords could have a positive effect on the topics generated. The model using stopwords found frequently in the reviews stopped at 50, but increasing that number could have improved the results. Using tf-idf for finding stopwords would also have been an interesting approach.

For the association rule mining, the transactions were created using the title of the anime as opposed to the anime id. This choice was made purely because it is easier to just read the title of the anime rather than having to do a look up for the title based on the anime id.

0.1.3 Next Steps

There are many ways in which the models could be improved as outlined above. Experimenting with different kinds of clustering techniques, filtering for higher ratings in the user ratings when selecting transactions as well as improving on the stopwords used in the topic model are all paths not taken. Following along these paths and iterating further should lead to better models. As it stands, the current models do provide benefits and will therefore be deployed.

0.2 Deployment

1. Build the models and store them in a format that allows for easy retrieval.
2. Store models in AWS S3 buckets
3. Create microservice using flask to load the models on start up and accept requests through APIs to provide recommendations.
4. Deploy microservice to AWS to make it available publicly.
5. Create MySQL database for the application server which will store all anime and user information for the application.
6. The database for the application will also be hosted on AWS
7. Create application server using spring boot to handle the business logic for the website.
8. Deploy application server to AWS to facilitate communication with the recommendations microservice

9. Create user portal using NextJS.
10. Deploy user portal using Vercel to make it available publicly.

0.2.1 Monitoring

Provide sufficient logging throughout the application to facilitate the troubleshooting of any issues that may arise post-deployment.

AWS allows for the export of application logs to aid in troubleshooting. It also provides a general health status for the environment in which the application is running.

The database can be queried so as to get a clear state on the information being stored.