  
BACS2003 ARTIFICIAL INTELLIGENCE

**202301 Session, Year 2022/23**

**Assignment Documentation**

| **Full Name: Kong Zhi Lin** | |
| --- | --- |
| **Student ID: 22WMR05663** | |
| **Programme: Bachelor of Computer Science (Honours) in Software Engineering** | |
| **Tutorial Class: G7** | |
| **Project Title: Anime Recommender System** | |
| **Module In-Charged: Content-Based Filtering + User-Based Collaborative Filtering** | |
| **Other team members’ data**   | **No** | **Student Name** | **Module In Charge** | | --- | --- | --- | | **1** | **Tan Eng Lip** | **User-Based Collaborative Filtering + Item-Based Collaborative Filtering** | | **2** | **Chiew Hong Kuang** | **Content-Based Filtering + Item-Based Collaborative Filtering** | | |
| **Lecturer: Goh Ching Pang** | **Tutor : Ho Chuk Fong** |

# **Introduction**

## Problem Background

Many individuals nowadays are more interested in spiritual prosperity than they are in material wealth in this age of modernity and technological advancement. In order to relax after a stressful period, individuals all over the world are now searching out and taking pleasure in a range of entertainments, including movies, sports, anime, and video games (Mamat et al., 2022).

In this project, we will discuss Japanese animation (known as anime) which is a hand-drawn computer animation with novel culture that has recently been more popular in this century and has even gained cult status throughout the world (Qureshi, 2020). The popularity of the anime is increasing and it has been pointed out that the lack of a good anime recommender system on the market has caused many people to feel difficulty in finding the right anime for them and, as a result, reduced awareness of this amazing culture (Nuurshadieq, 201). As usual we can find out that those anime recommender systems always recommend the top-rated anime at the specific periods to the user instead of recommending personalized recommendations to the users. Therefore, it caused the time consuming for the users to figure out their interested anime through the system by manually searching and filtering. Fortunately, we have proposed a solution for this issue by directing users to highly regarded anime that fits their viewing preferences.

## Objectives/Aims

In this project, our primary objective and goal is to improve the user experience by providing personalized anime recommendations that are tailored to their interests and needs based on the anime they have watched previously. We also provide accurate searches among thousands of search results due to the anime recommended system that we have proposed being capable of producing a relevant and appropriate list of anime for the users. This allows users to find what they are looking for quickly and easily. This eventually can increase the users’ productivity, such as discovering new anime they were not familiar with before and also increased user engagement in the anime streaming platform (G.Shriram, U.G.Prithika, S.Dhivya, 2022).

Additionally, with the help of this anime recommender system a list of anime is generated to recommend based on the ratings in our dataset, we believe it can promote the anime culture and give users who are unfamiliar with anime a positive impression. As a result, by making individualized, high-quality recommendations rather than relying solely on blind, effortless searching, customer satisfaction will rise and eventually be improved (Bushra Alhijawi, Arafat Awajan and Salam Fraihat, 2022).

## Motivation

The use of a successful anime recommender system can have several benefits for both anime streaming platforms or publishers, and anime fans. Firstly, it can increase revenue by promoting content that users are more likely to watch or buy (Trojnarski, 2023). A recommender system can analyze users' viewing habits and preferences, and recommend similar anime titles that they may enjoy. This can encourage users to explore different genres and discover new anime titles, which can lead to increased revenue from rentals or purchases. Additionally, by promoting newer or less popular content, the recommender system can help to increase the visibility of these titles and potentially drive more revenue for the platform or publisher.

Moreover, personalized anime content recommendations can also increase user retention and engagement with the streaming platform. By providing recommendations that align with the user's interests, they are more likely to continue watching anime content and exploring new titles. This can help to build user loyalty and increase the likelihood of repeat business. A recommender system that suggests anime titles based on viewing habits can help guide users towards shows that they may enjoy, making it easier for them to become fans of the medium (*What is a recommendation system?,* 2021).

Furthermore, the recommender system can also play a role in promoting diversity and inclusivity in the anime community. For individuals who are new to the anime medium, the vast number of titles and genres can be overwhelming and intimidating. The recommender system can provide personalized recommendations that cater to their interests and make it easier for them to navigate the vast catalog of anime content available. By promoting popular anime at the moment and expanding the anime fanbase, the recommender system can help to increase awareness and appreciation for the art form. This, in turn, can encourage the production of more high-quality anime titles in the future based on the preferences of the anime fans.

## Timeline/Milestone

| **Tasks** | **Start Time** | **End Time** | **Duration** |
| --- | --- | --- | --- |
| Research Problem Background | 06/03/2023 | 13/03/2023 | 8 Days |
| Research Objective | 14/03/2023 | 20/03/2023 | 7 Days |
| Research Motivation | 14/03/2023 | 20/03/2023 | 7Days |
| Research the background of the recommender system. | 21/03/2023 | 25/03/2023 | 5 Days |
| Identify the tools required throughout the development process. | 26/03/2023 | 27/03/2023 | 2 Days |
| Find the suitable dataset for the recommender system. | 28/03/2023 | 29/03/2023 | 2 Days |
| Figure out the suitable pre-processing method to be applied to the dataset. | 30/03/2023 | 03/04/2023 | 5 Days |
| Draw flowchart for the recommender system. | 04/04/2023 | 08/04/2023 | 5 Days |
| Propose test plan and hypothesis. | 09/04/2023 | 10/04/2023 | 2 Days |
| Development. | 11/04/2023 | 30/04/2023 | 20 Days |
| Debugging. | 11/04/2023 | 30/04/2023 | 20 Days |
| Evaluate the result. | 01/05/2023 | 04/05/2023 | 4 Days |
| Discussion and interpretation. | 01/05/2023 | 04/05/2023 | 4 Days |

# 

# **Research Background**

## Background of the applications

An advanced technology known as a recommender system makes use of artificial intelligence (AI) to offer users personalized recommendations based on their historical behavior, preferences, and patterns (Belhekar, 2021). The main goal of a recommender system is to suggest items or content that the user is most likely to be interested in, based on their previous interactions with the system. Recommender systems have been widely applied in different areas, including e-commerce, online advertising, music, movie, and item recommendations, among others.

There are different types of recommender systems, including content-based filtering, collaborative filtering, and hybrid recommender systems. Content-based filtering recommends items that have similar attributes or features to those that the user has interacted with before (Ibtesama, 2020). On the other hand, collaborative filtering suggests items based on the behavior and preferences of other users who have similar interests to the user (Vaddy, 2020). Hybrid recommender systems combine the features of both content-based and collaborative filtering approaches.

Additionally, in order to make recommendations, recommender systems use various techniques and algorithms such as data mining, machine learning, and natural language processing. To be effective, these systems require a large amount of data about user behaviors and preferences, as well as the items being recommended. In addition, recommender systems are constantly learning and improving based on user feedback, which helps to enhance the accuracy and relevance of the recommendations over time.

## Analysis of selected tool with any other relevant tools

| **Tools comparison** | **Remark** | **Jupyter Notebook**  **(Python 3)** | **Visual Studio Code** | **Google Docs** | **Microsoft Excel (2019)** |
| --- | --- | --- | --- | --- | --- |
| Type of license and open source license | State all types of license | Open source | Free for private and commercial use. | Free for Google users. | Microsoft Office License required |
| Year founded | When is this tool being introduced? | 2014 | 2015 | 2006 | 1985 (oldest version of Excel) |
| Founding company | Owner | Fernando Pérez and Brian Granger | Microsoft Corporation | Upstartle | Microsoft Corporation |
| License Pricing | Compare the prices if the license is used for development and business/commercialization | None | Free | Free | None  (pre-install in laptop, so free for us) |
| Supported features | What features that it offers? | - Enable user to run interactive python code  - Markdown language supported | - Enable users to manage the code with the support of various extensions.  - Enable users to execute code in various file types such as python, c, java, ipynb etc.  - Enable connection to github for version control and collaboration. | - Acts as a tool for collaborative online document editing  - Can be shared, opened, modified, downloaded by multiple users simultaneously.  - Grammarly-check  - Spelling-check  - Can retrieve the history in order to get all version | - Inserting a pivot table  - Sorting of tabulated data  - Adding formulas to the sheet to perform calculation  - Visualize the data |
| Common applications | In what areas this tool is usually used? | - Understanding the dataset loaded in  - Perform data science tasks | - Manage and execute the code for the development process. | - Allow users to collaborate editing the same documentation and also manage the version of the document. | - Perform data analysis (by add-ins in Excel)  - Calculate the budget of the project |
| Customer support | How the customer support is given, e.g. proprietary, online community, etc. | Open-source community | Microsoft support | Google | Microsoft support |
| Limitations | The drawbacks of the software | Buffer on kernel will affect running of the kernel and requires to restart all the kernel and run again. | Difficult to manage plugins when used for different programming environments. | Requires internet connection to collaborate with other people or access to the document. | It is difficult to detect fraud/corruption |

## Justify why the selected tool is suitable

Visual Studio Code served as the primary tool for developing our recommendation algorithm that predicts suitable results or recommendations for users. As a versatile code editor, it supports multiple programming languages and file formats, including the widely-used .ipynb format for Jupyter notebooks. One of its major advantages is its ability to integrate with various plugins that provide useful functionalities such as code completion and error detection. This feature saves a considerable amount of time by preventing unnecessary trial and error attempts. Additionally, Visual Studio Code allows the import of libraries such as pandas, string, re, time, etc., which are essential for developing advanced machine learning algorithms. This capability makes it easier to use pre-built functions rather than writing code from scratch.

Moreover, Visual Studio Code offers powerful collaboration features that allow multiple team members to work on the same project simultaneously. It supports Git version control, which allows users to commit, push, and pull the latest code into GitHub, where other team members can easily make changes to the source code without any conflicts. Additionally, GitHub has version control features that enable users to revert to previous versions of the code and trace back who made specific changes. This collaborative approach enhances productivity, teamwork, and efficiency in software development projects. Visual Studio Code provides a user-friendly and efficient environment for developers to create and collaborate on various programming projects. Its extensive features and plugins make it an ideal tool for developing sophisticated machine learning algorithms, such as our anime recommendation system.

Google docs is used for documenting the introduction of this project, research background. methodology , result as well as the discussion and conclusion. Google docs provides a place that lets me explain my development of this project instead of code presentation. Moreover, I can create and edit documents via online using Google Docs as it is the web-based word processing tool.

# **Methodology**

## Description of dataset

The source of the dataset comes from Kaggle which is a website that provides various open source dataset. Our dataset is the “Anime Recommendations Database” provided by a user called “COOPERUNION” (CooperUnion, 2016). There are two files in this dataset which are anime.csv and rating.csv. This data set contains information on user preference data from 73,516 users on 12,294 anime. Each user is able to add anime to their completed list and give it a rating and this data set is a compilation of those ratings. Inside the anime.csv had 7 columns of variables about the information of anime movies which are anime\_id, name, genre, type, episode, rating and members. There are 3 columns of variables of user rating which are user\_id, anime\_id and rating inside the rating.csv.

anime.csv

| Numbering Column | | Description |
| --- | --- | --- |
| anime\_id | | myanimelist.net's unique id identifying an anime. |

| Inputs/Features | Represent | Description |
| --- | --- | --- |
| name | Name | Full name of anime. |
| genre | Genre | Comma separated list of genres for this anime. |
| type | Type | Type of show such as movie, TV, OVA, etc |
| episodes | Episodes | The number of episodes in this show. (1 if movie). |
| rating | Rating | Average rating out of 10 for this anime. |
| members | Members | Number of community members that are in this anime's  "group". |

rating.csv

| Numbering Column | | Description |
| --- | --- | --- |
| user\_id | | Non identifiable randomly generated user id. |

| Inputs/Features | Represent | Description |
| --- | --- | --- |
| anime\_id | Anime\_id | The anime that this user has rated. |
| rating | Rating | Rating out of 10 this user has assigned (-1 if the user watched it but didn't assign a rating). |

newRating.csv

New defined dataset that is the same label column as the rating.csv file but the rating data is collected from the users to test on the accuracy of the anime recommended system.

| Numbering Column | | Description |
| --- | --- | --- |
| user\_id | | Non identifiable randomly generated user id. |

| Inputs/Features | Represent | Description |
| --- | --- | --- |
| anime\_id | Anime\_id | The anime that this user has rated. |
| rating | Rating | Rating out of 10 this user has assigned (-1 if the user watched it but didn't assign a rating). |

### **Data Preprocessing**

**Data Transformation**

There are some variables belonging to improper data types in the anime.csv such as the data type of name, genre, type and episodes belongs to object while the name, genre and type should belong to string type and the episodes belong to int64. Figure 3.1.1 below shows the original data type of the variables in the anime.csv while the Figure 3.1.2 shows the output after applying astype() method in pandas to convert the data type of the variables. Data transformation is necessary and important since the algorithm that is used for generating the recommended system is sensitive to the type and format of data.

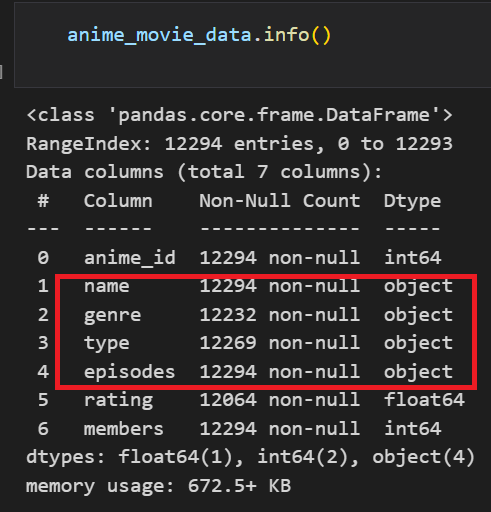


Figure 3.1.1 Before Data Transformation

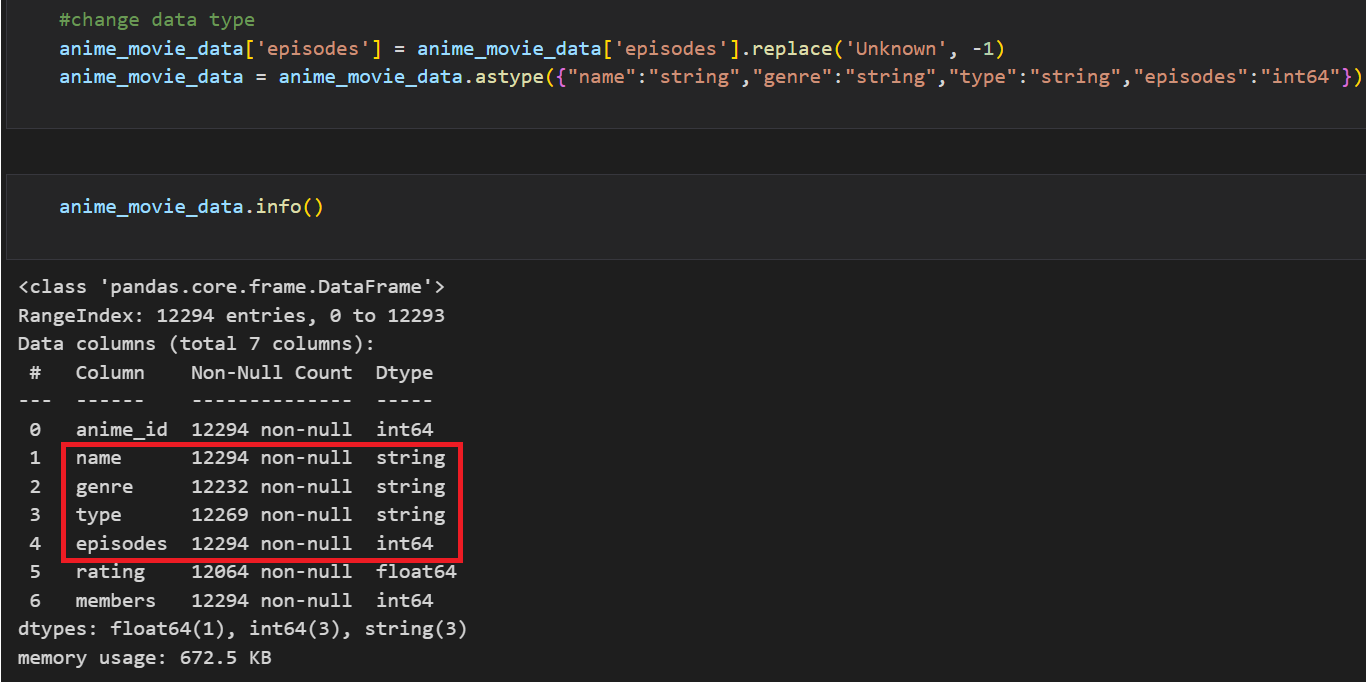


Figure 3.1.2 After Data Transformation by Using ‘astype()’

**Meaningful Variable Name**

1. rating

The column label of ‘rating’ appears at both the anime.csv and rating.csv but the meaning of ‘rating’ is different from each other. The ‘rating’ in the anime.csv means the average rating for each anime movie that is rated by the users while the ‘rating’ in the rating.csv means the user rating for each anime they had viewed. Therefore, there must be a modification on the variable name in order to reduce the confusion of the label. Figure 3.1.4 shows the original label of the anime.csv while Figure 3.1.5 shows the modified label of the anime.csv. Figure 3.1.6 shows the original label of the rating.csv and Figure 3.1.7 shows the modified label of the rating.csv.

‘rating’ in the anime.csv changes to ‘avg\_rating’

‘rating’ in the rating.csv changes to ‘user\_rating’



Figure 3.1.4 Original data frame of anime.csv with ‘rating’ label

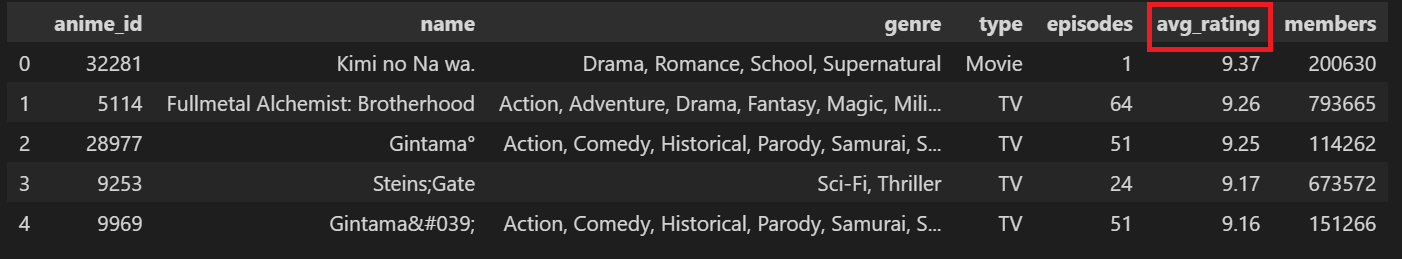


Figure 3.1.5 Modified data frame of anime.csv with ‘avg\_rating’ label

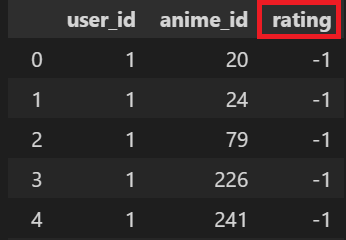


Figure 3.1.6 Original data frame of rating.csv with ‘rating’ label

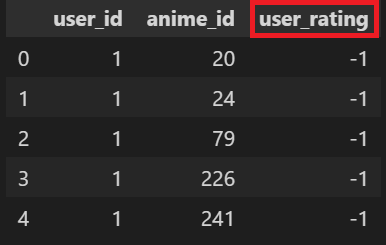
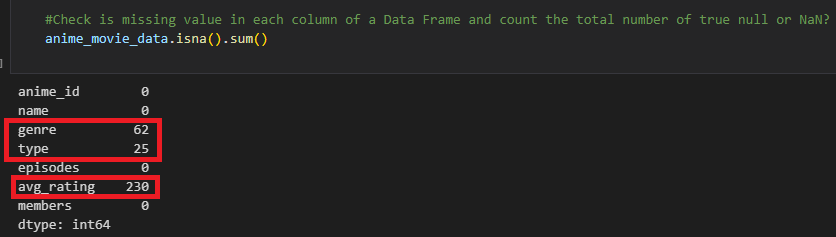


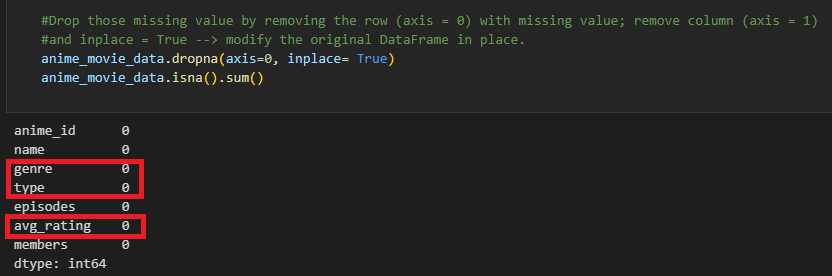
Figure 3.1.6 Original data frame of rating.csv with ‘rating’ label

**Removing Null Value**

Ensure there is no missing value of the data in the data frame by using the isna().sum() method in the pandas. dropna() method is used to remove all the missing values in the data frame.



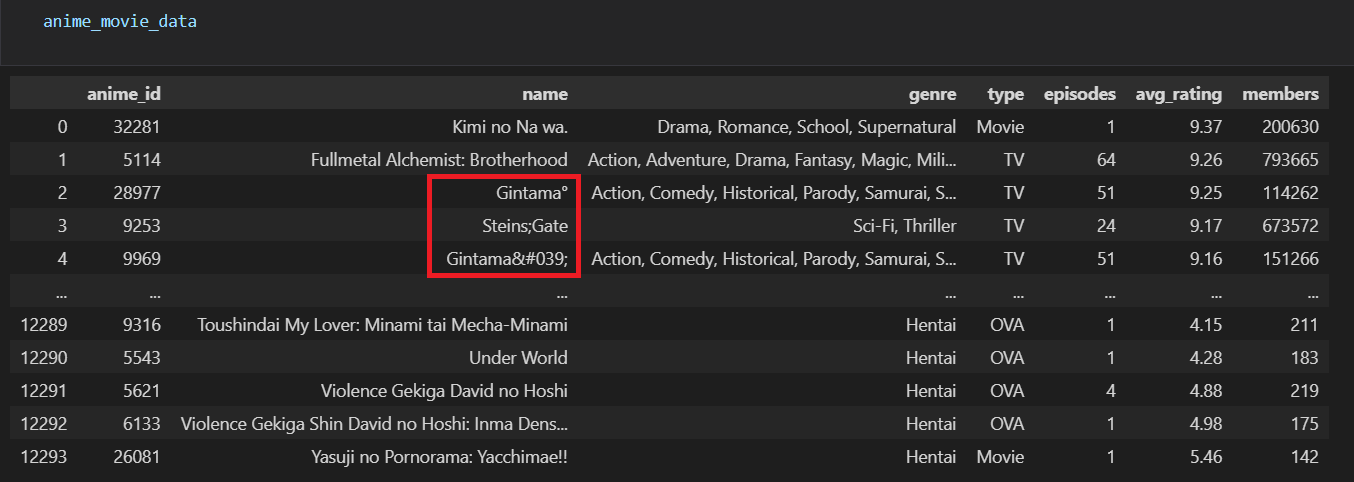
Before removing the missing value



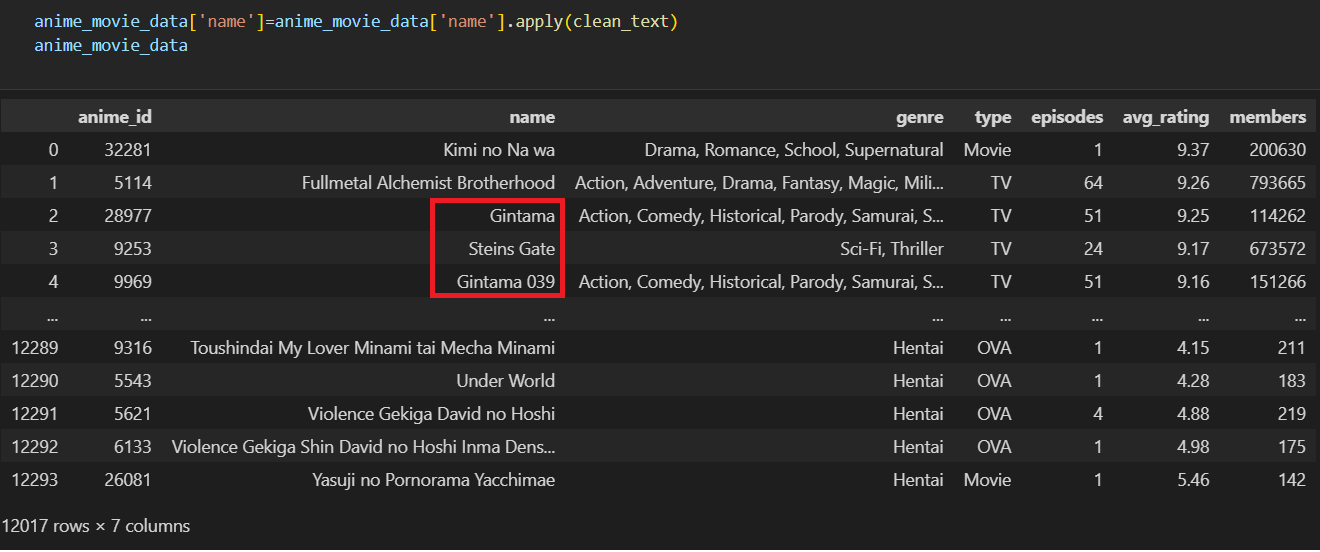
After removing the missing value

**Clean Text**

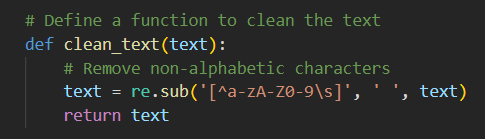
It is found that the anime movie contains special characters which will increase the difficulty of reading and identify the name. By using the clean\_text() function we can remove all those special characters in the name to make it readable and understandable.



Before using clean\_test() function



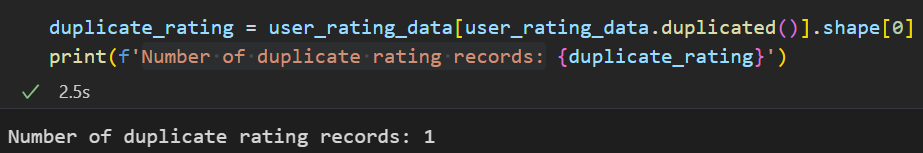
After using clean\_test() function



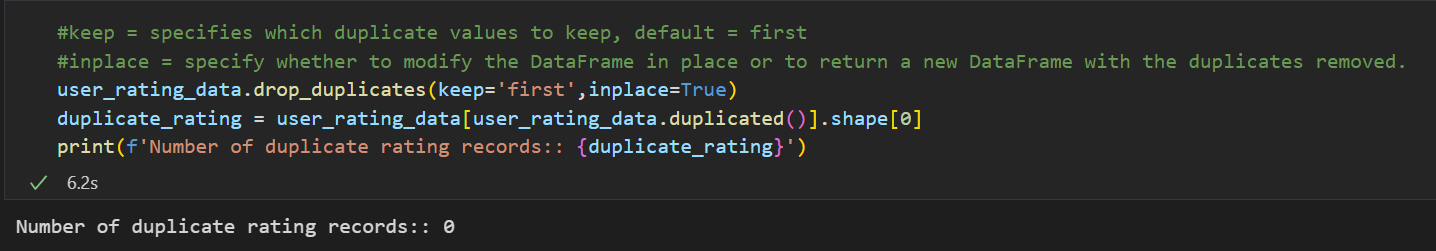
clean\_text function

**Remove Duplicate Records**

There should not be any duplicate records occurring in the anime.csv and rating.csv since it will make the recommender system become inaccurate and waste resources which will slow down the system’s performance. It is found that the rating.csv had 1 duplicate record by using duplicated() method to figure out and using the drop\_duplicates() method to remove the duplicate records.



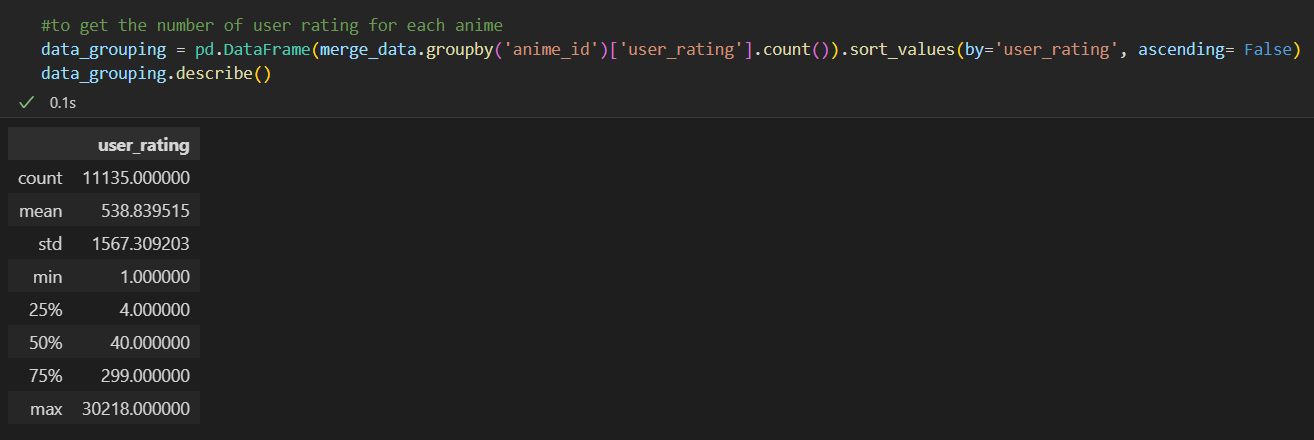
Before removing duplicate rating records



After removing duplicate rating records by using drop\_duplicates() function

**Data Trimming**

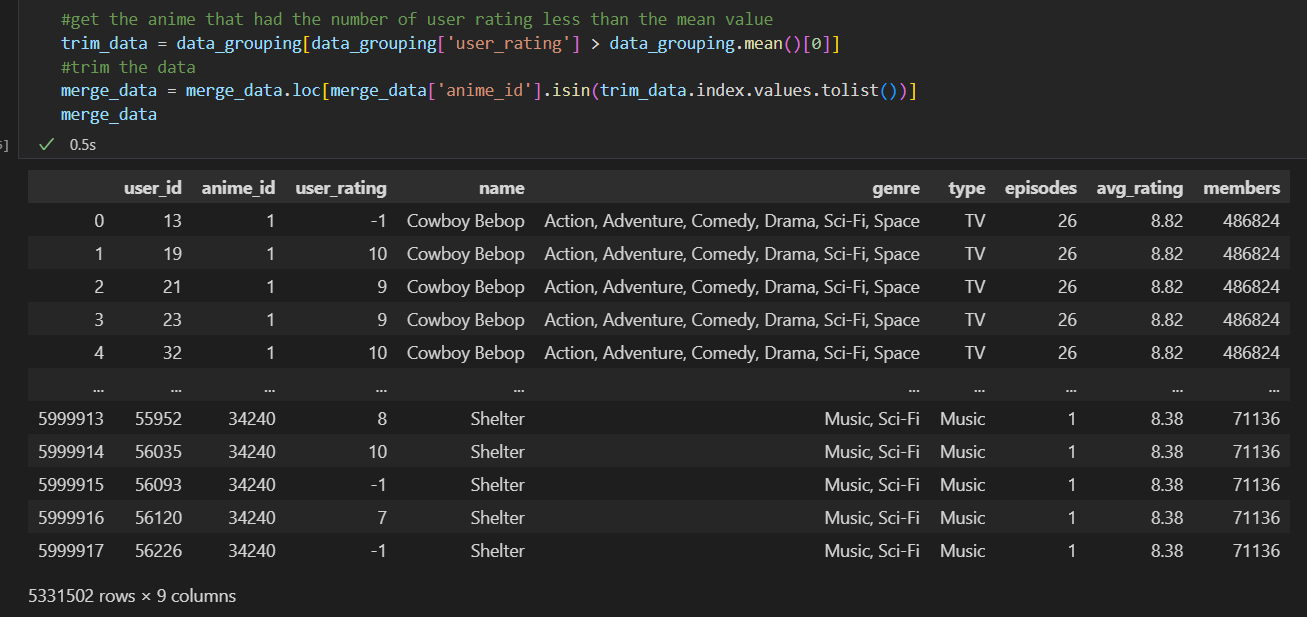
Data trimming is used to remove extreme ratings given by users that do not accurately represent their preferences in order to increase the accuracy of the recommended system. With the help of mean() after we group the anime\_id and the user\_rating by the groupby(), we can calculate the mean value among all these user rating values of the anime. Those values of user rating that are less than the mean value will be trimmed.



Group the anime\_id and user\_rating by using the groupby()

&

Use describe() to display the mean, standard deviation, minimum value and so on



Trim data for those user rating value is less than the mean

## Applications of the algorithm(s)

The algorithm that was used in this project is the combination of content-based filtering and the user-based collaborative filtering in order to provide anime recommendations to the users. The system will use the information about the movies such as genre, name, type and so on for the content-based filtering while the user-based filtering will be using the information about the users’ interactions with the movies such as the user ratings.

**Content-based filtering**

Content-based filtering will use the content features of movies including the name of the movie, genre of movie, type of movie and so on to recommend similar movies to the users. In this filtering, the tf-idf ( term frequency-inverse document frequency) matrix is constructed to represents each document in the collection as a vector of tf-idf values, the cosine similarity matrix is also constructed to represents the similarity between each pair of documents in the collection based on their tf-idf vectors and compute the cosine similarity score to each anime. The anime with the highest similarity score will be recommended to the users. Figure 3.2.1 shows the histogram of the frequency of cosine similarity scores from the 10 highest scores of anime.

Figure 3.2.2 shows the result of anime recommendation with the cosine similarity scores by using the content-based filtering.

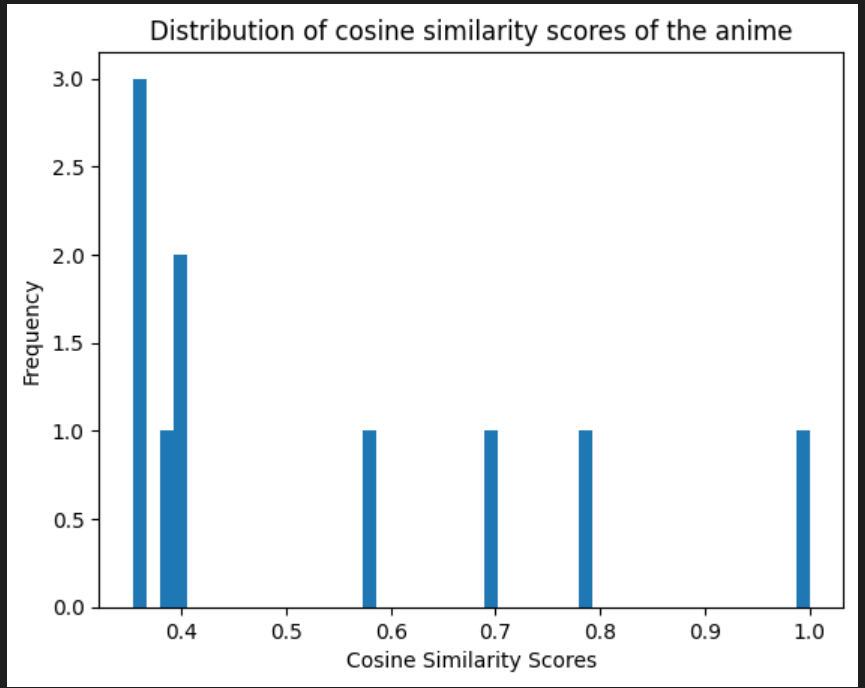


Figure 3.2.1 Histogram of frequency of the cosine similarity scores of anime

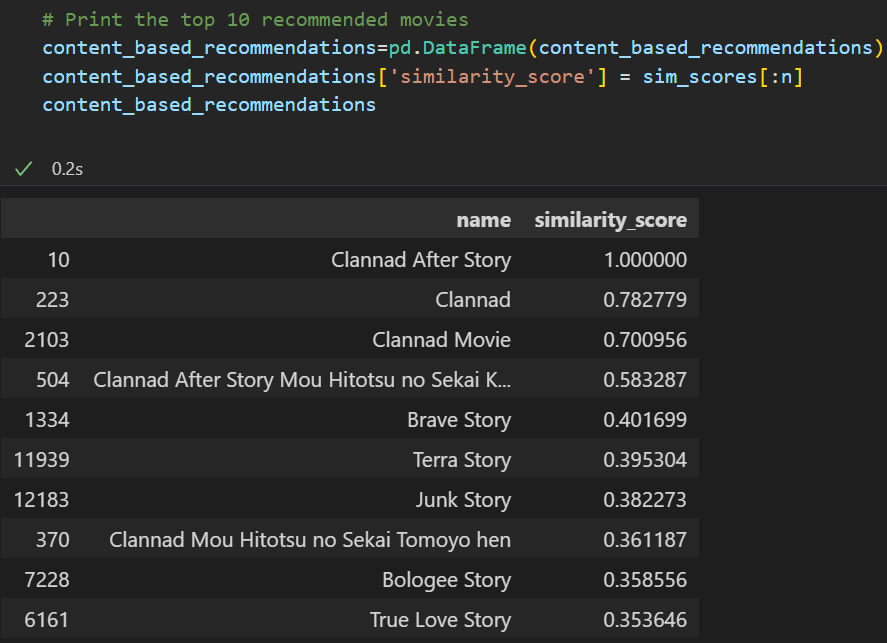


Figure 3.2.2 Result of content-based filtering

**User-based collaborative filtering**

User-based collaborative filtering makes use of the preferences of similar users to generate recommendations which means the system identifies users who have similar preferences, view histories to the current user, and then recommends the anime. In this filtering, Pearson Correlation Coefficient is used to determine the relationship between one user and anothers. The anime with the highest correlation between the selected users will be recommended to the users. Figure 3.2.3 shows the histogram of the frequency of correlation scores from the 10 highest scores of the users.

Figure 3.2.4 shows the result of anime recommendation with the correlation scores by using the user-based filtering.

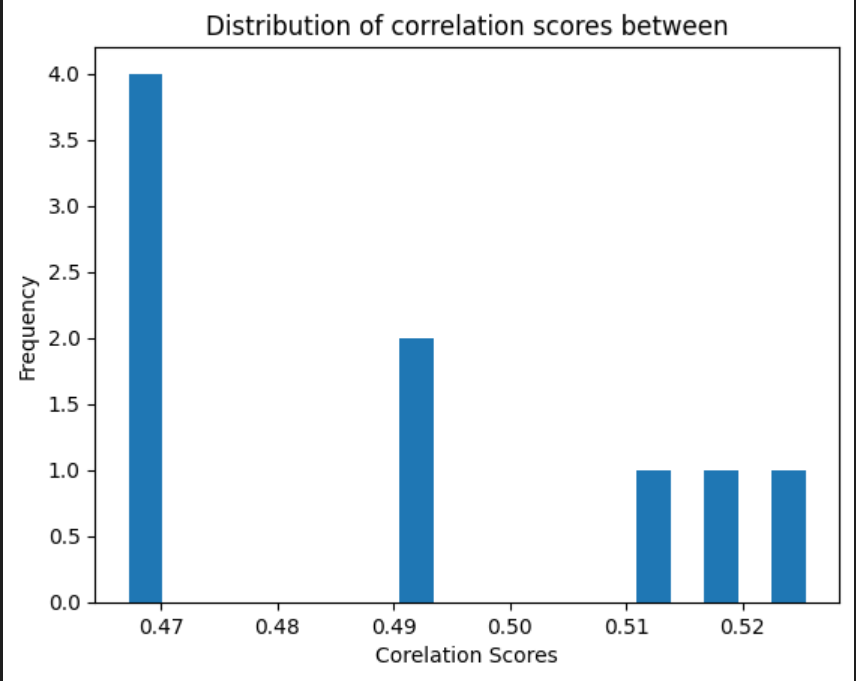


Figure 3.2.1 Histogram of frequency of the correlation scores of user

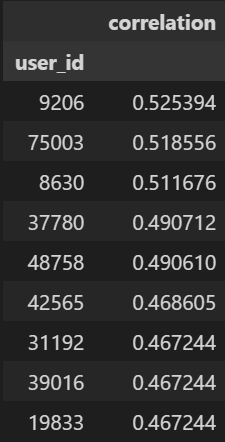
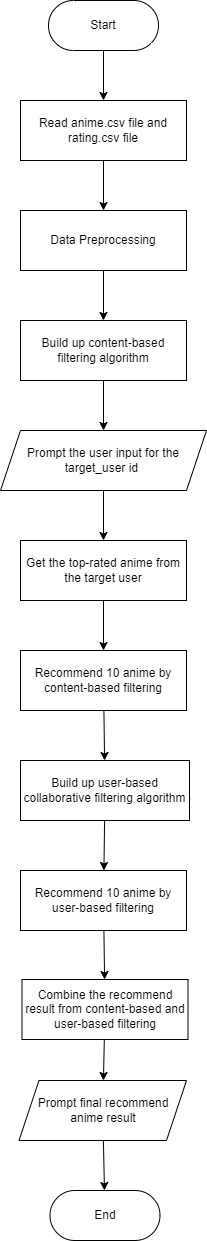


Figure 3.2.2 Result of user-based filtering

## System flowchart/activity diagram

**

## Proposed test plan/hypothesis

**Hypothesis**

Hypothesis that I had made for this hybrid anime recommender system project (combination of content-based filtering and the user-based filtering) is that the final recommended anime will have more than 80% accuracy.

**Test plan**

Since there does not have a discrete or systematic measurement to test on the accuracy of the recommended system, but it highly depends on the user who provides the data and the system recommend to the users as well as to get the satisfaction of the users. Therefore, in this case, a few people are going to help test the accuracy of this project by receiving their viewed anime and the rating about those anime. Then, we recommend them some anime via the 2 filtering algorithm and get the feedback of their satisfaction about the recommendation.

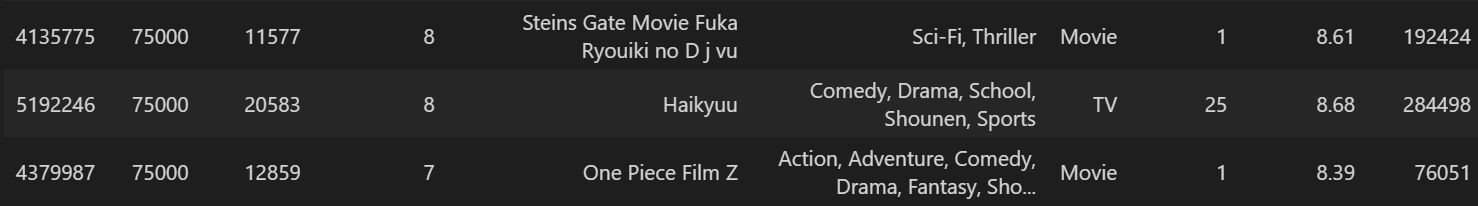
# **Result**

## Results

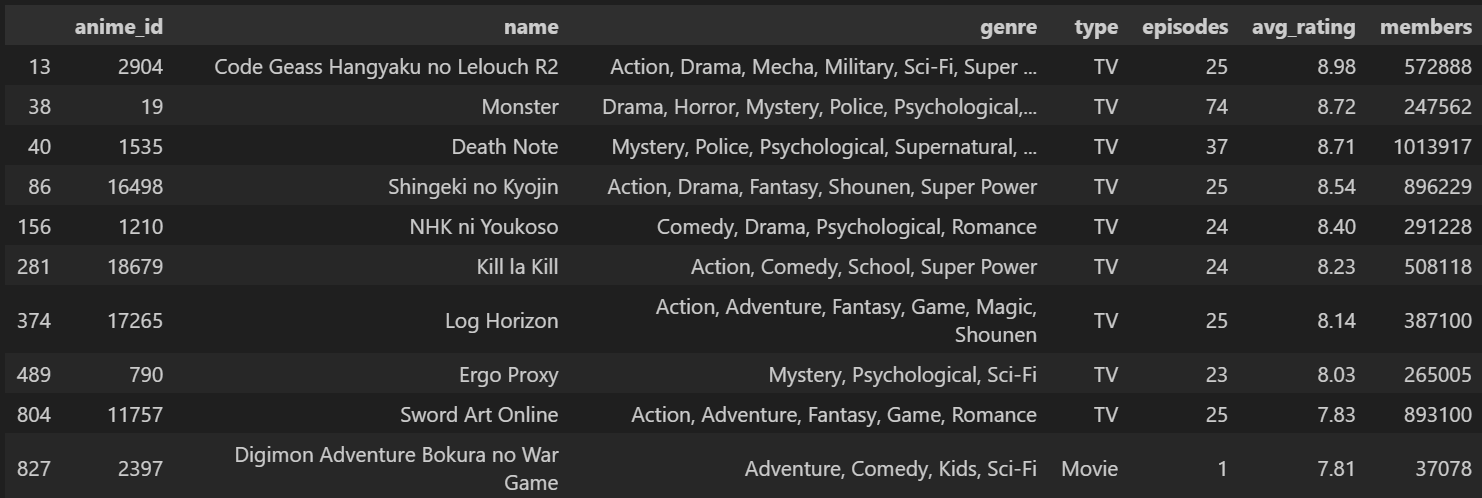
The user\_id starts from 75000 until 75005 is the users that used to test on the recommended system.

75000’s user rating for the anime he had viewed:





Recommended anime to 75000’s user:

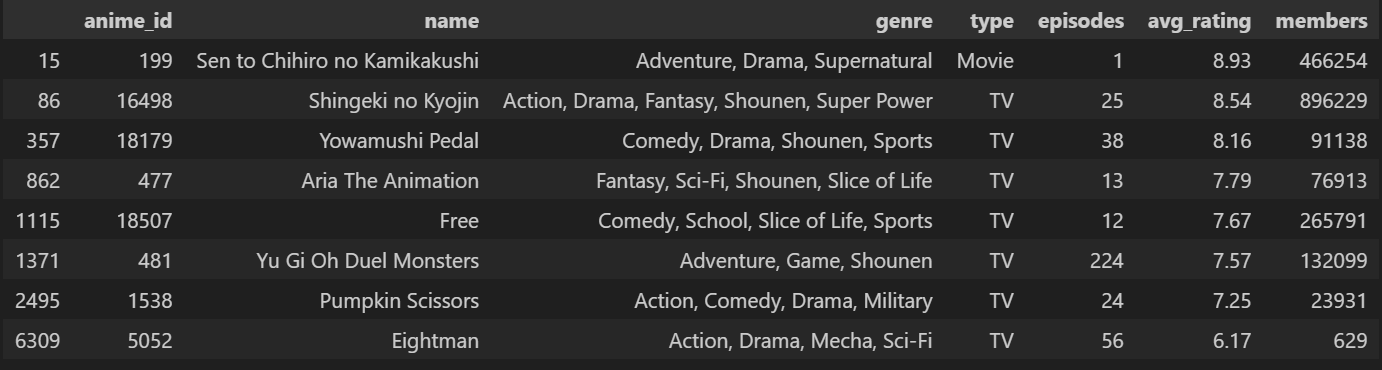


There are 8 out of 10 anime recommendations that users are interested in and satisfied with the recommended result. The 2 anime the users are not interested in are ‘Monster’ and ‘Ergo Proxy’. Percentage of accuracy of this recommended system is 80%.

75001’s user rating for the anime he had viewed:

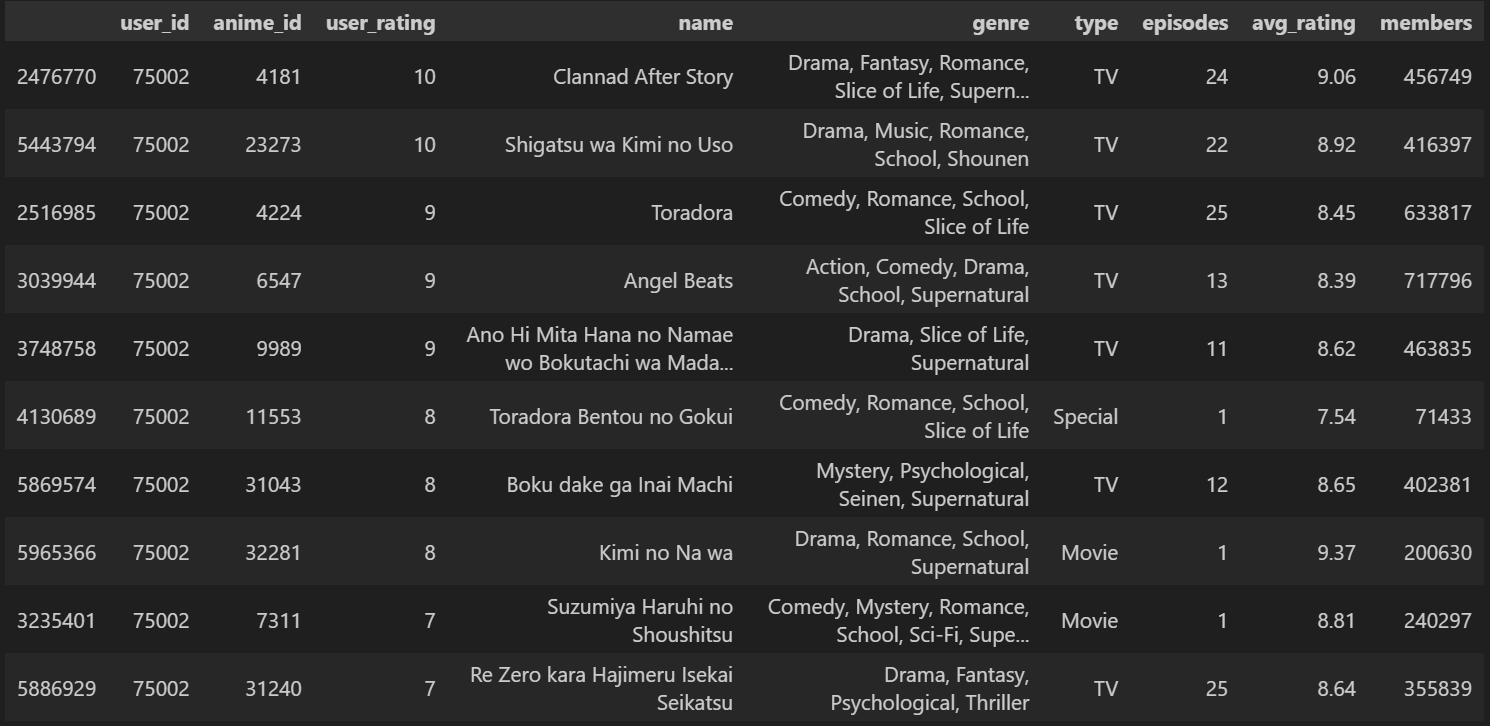


Recommended anime to 75001’s user:

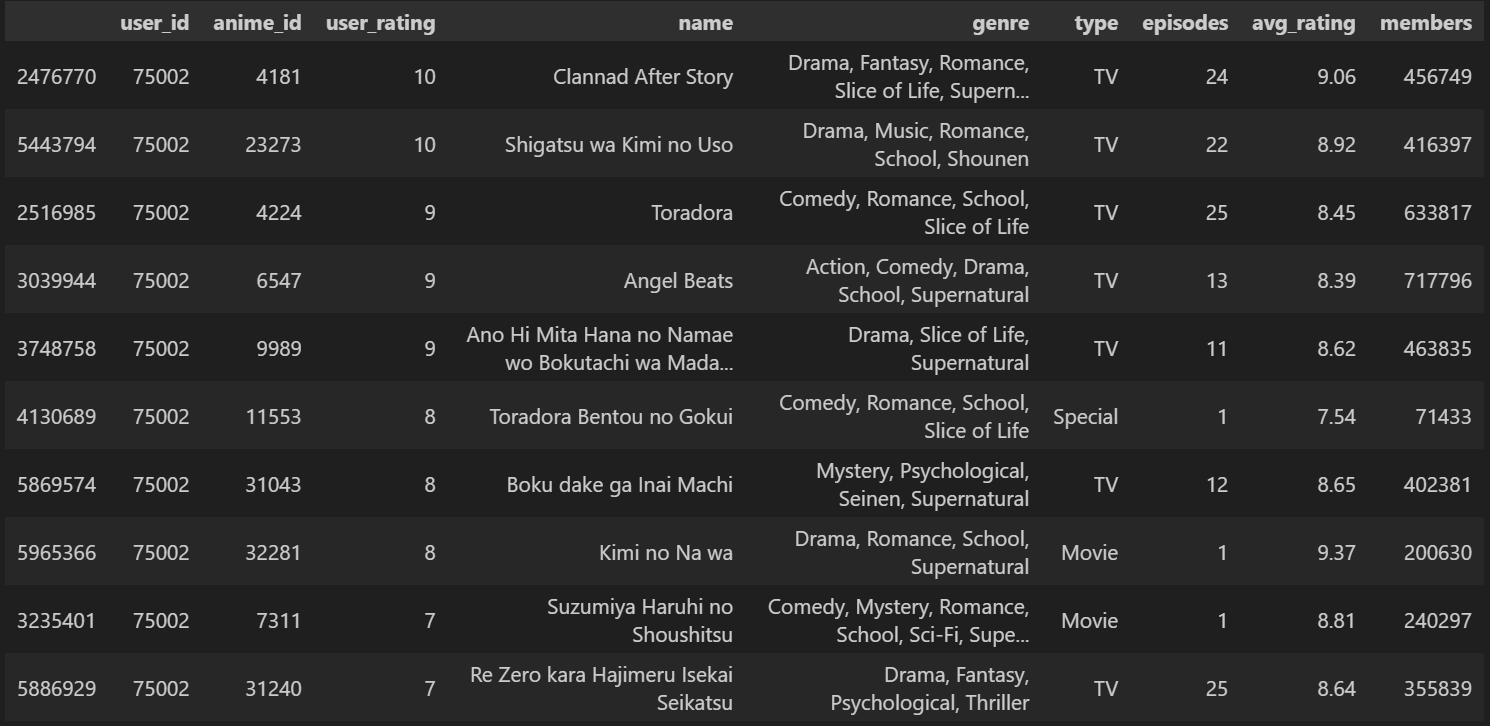


There are 5 out of 8 anime recommendations that users are interested in and satisfied with the recommended result. The 3 anime that the users are not interested in are ‘Yowamushi Pedal’, ‘Free’ and ‘Pumpkin Scissors’ . Percentage of accuracy of this recommended system is 62.5%.

75002’s user rating for the anime he had viewed:



Recommended anime to 75002’s user:

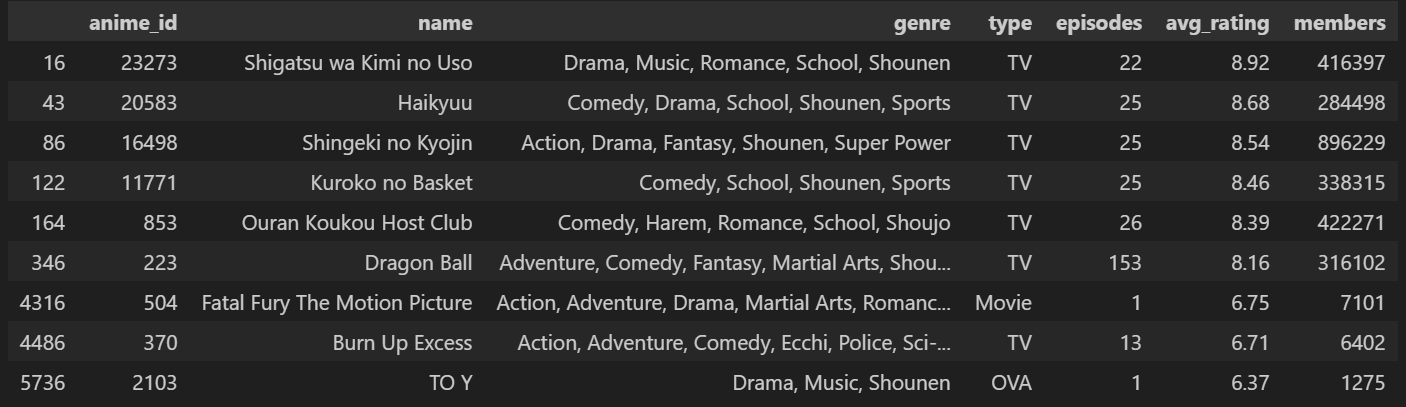


There are 10 out of 10 anime recommendations that users are interested in and satisfied with the recommended result. The user is interested in all of the anime that are recommended. Percentage of accuracy of this recommended system is 100%.

75003’s user rating for the anime he had viewed:



Recommended anime to 75003’s user:



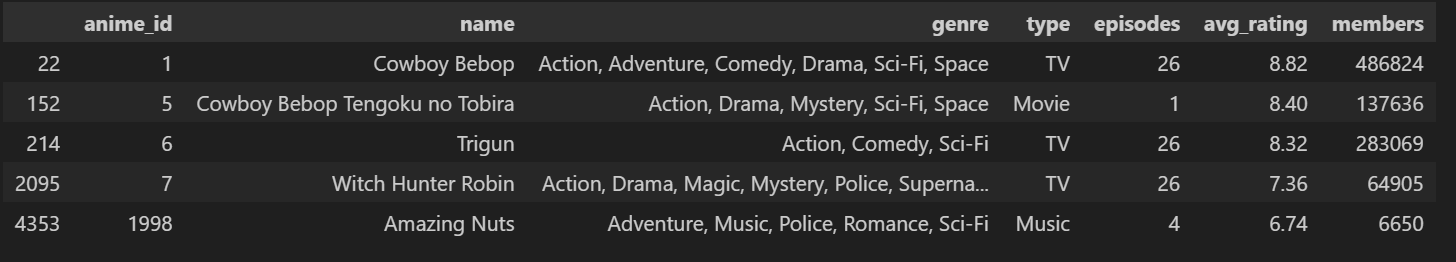
There are 8 out of 9 anime recommendations that users are interested in and satisfied with the recommended result. The anime that the users are not interested in are ‘To Y’. Percentage of accuracy of this recommended system is 88.8%.

75004’s user rating for the anime he had viewed:





Recommended anime to 75004’s user:

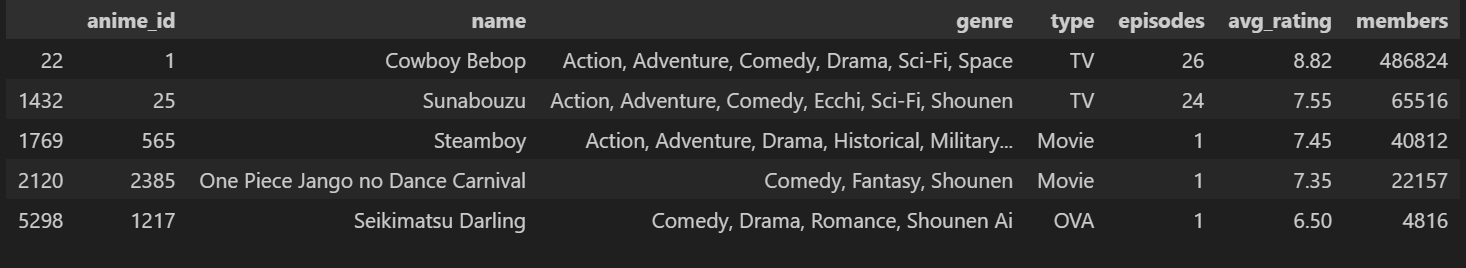


There are 4 out of 5 anime recommendations that users are interested in and satisfied with the recommended result. The anime that the users are not interested in are ‘Trigun’, ‘Free’ and ‘Pumpkin Scissors’ . Percentage of accuracy of this recommended system is 80%.

75005’s user rating for the anime he had viewed:



Recommended anime to 75005’s user:



There are 5 out of 5 anime recommendations that users are interested in and satisfied with the recommended result. The users are interested in all the anime that the system recommended. Percentage of accuracy of this recommended system is 100%.

## Discussion/Interpretation

The percentage of average accuracy of this anime recommended system with the combination of content-based filtering and user-based collaborative filtering is 85.22%. The formula to calculate the percentage of average accuracy is:

**Sum of the accuracy of recommendation for each user / number of users test the accuracy**

Calculation: (80% + 62.5% + 100% + 88.8% + 80% + 100%) / 6 = 85.22%

Therefore the hypothesis is accepted due to the percentage of average accuracy of this anime recommended system being over 80%.

# **Discussion and Conclusion**

## Achievements

This anime recommended system which contains 2 combinations of content-based filtering and the user-based filtering that aims to improve the user experience, promote anime to the public and increase the explosion of the anime type, genre and so on. With the proof of the recommended result above, we have achieved the goal or the objective of this project. We can be claimed to provide an accurate recommended anime result to the users in order to increase the satisfaction of the user. Since both of the algorithms covered a wide range of data that included the information about the anime and the information about the user interaction with the anime. The content-based filtering will be more focused on the anime’s information while the user-based filtering is focused on the user interactions with the anime and the combination of these 2 algorithms which makes the recommended system will be more accurate. This is because the system will go through the anime’s information to find similar anime where the top-rated anime of the target user and find the similar user top rated anime to generate the recommendation. This project can avoid the user blindly searching for the anime that they are interested in.

Moreover, this project has achieved to reduce the time of searching for anime that are interested by the users. According to the algorithms I had implemented, this system requires around 2 to 3 mins for generating the recommendation to the users. Therefore, the users can save their time on searching the anime as well as increase their user experience. The system will auto generate the recommendation based on their preference to the user instead of they finding the anime by memorizing the specific name of the anime or the members cast in the anime.

In a nutshell, this hybrid anime recommended system can be concluded as an accurate anime recommended system in this case. With the proof of the accuracy of this recommended system above and the achievements of this project, I can claim that this project is successfully implemented and performed like what is expected to meet the goal and the objective of this project.

## Limitations and Future Works

First and foremost, there is a significant limitation of this anime recommended system which is the cold-start problem. Cold-start problems meant by the system unable to provide an accurate and satisfactory recommendation when there is a new registered user occur in the system who does not have any information available about the user’s preferences while using the user-based collaborative filtering. Since this project is the hybrid anime recommender system, the content-based still can generate some recommendations to the user based on the anime’s information but it will not be personalized recommendations.

Moreover, the sparsity of the user-item matrix is also one of the limitations of this project. This is because we cannot ensure all the users will rate for every anime they have viewed until the number of anime that a user has rated is much smaller than the total number of anime in the system. As a result, it leads to a sparse matrix which increases the difficulty of finding similar users or anime.

Last but not least, this project consumes a huge amount of resources due to the combination of 2 algorithms increasing the use of computational power and memory resources. This is an obvious limitation in this project since the rating.csv will have a file size with 102mb which consists of a huge dataset.

In the future, we can implement a simple user interface that allows the user to interact with the recommendation system and provide their feedback on the recommendations that the system provided in order to improve the accuracy of the system. Additionally, we can implement recommendation systems which include multiple filtering algorithms in a recommended system such as adding the item-based collaborative filtering in the future.

## 

# 

# **Reference & Source**

1. Mamat, R. *et al.* (2022) *VTUBERS and anime culture: A case study of Japanese learners in two public universities in Malaysia*, *International journal of health sciences*. Available at: <https://doi.org/10.53730/ijhs.v6nS2.8231> (Accessed: March 6, 2023).
2. Nuurshadieq (201AD) *Leveraging side information to anime recommender system using deep ...* Available at: <https://ieeexplore.ieee.org/document/9315363/> (Accessed: March 6, 2023).
3. G.Shriram, U.G.Prithika and S.Dhivya (2022) *Anime recommendation system - ijcrt.org*. Available at: <https://ijcrt.org/papers/IJCRT2201084.pdf> (Accessed: March 9, 2023).
4. Qureshi, A. (2020) *Building a recommendation system for Anime*, *Medium*. Towards Data Science. Available at: <https://towardsdatascience.com/building-a-recommendation-system-for-anime-566f864acea8> (Accessed: March 9, 2023).
5. Bushra Alhijawi, Arafat Awajan and Salam Fraihat (2022) *Survey on the Objectives of Recommender Systems: Measures, Solutions, Evaluation Methodology, and New Perspectives*, *ACM Digital Library*. Available at: <https://doi.org/10.1145/3527449> (Accessed: March 15, 2023).
6. CooperUnion, C.U. (2016) *Anime recommendations database*, *Kaggle*. Available at: <https://www.kaggle.com/datasets/CooperUnion/anime-recommendations-database?select=anime.csv> (Accessed: March 29, 2023).
7. Trojnarski, M. (2023) *How to boost sales with a recommender system?*, *Neoteric*. Available at: <https://neoteric.eu/blog/how-to-boost-sales-with-a-recommender-system/> (Accessed: March 14, 2023).
8. *What is a recommendation system?* (2021) *NVIDIA Data Science Glossary*. Available at: <https://www.nvidia.com/en-us/glossary/data-science/recommendation-system/#:~:text=Recommender%20systems%20are%20highly%20useful,data%20gathered%20about%20their%20interactions>. (Accessed: March 14, 2023).
9. Vaddy, S. (2020) *Collaborative filtering movie recommendation*, *Medium*. Medium. Available at: <https://medium.com/@srinidhi14vaddy/collaborative-filtering-movie-recommendation-60461c7ef897> (Accessed: March 23, 2023).
10. Ibtesama, I.B.T.E.S.A.M.A.H.M.E.D. (2020) *Getting started with a movie recommendation system*, *Kaggle*. Kaggle. Available at: <https://www.kaggle.com/code/ibtesama/getting-started-with-a-movie-recommendation-system> (Accessed: March 23, 2023).
11. Belhekar, C. (2021) *Recommender System metrics - clearly explained*, *Medium*. Medium. Available at: <https://chaitanyabelhekar.medium.com/recommender-system-metrics-clearly-explained-1f2ba6690216> (Accessed: March 23, 2023).

## 

# 

# 

# 

# 

# 