Université Claude Bernard Lyon 1

Viet Anh NGUYEN - Elliot FAUGIER

Data Mining

Anime - User Analysis

1 Overview of dataset

1.1 Source of dataset

The Anime-User dataset we utilize is sourced from the Kaggle dataset. Additionally, we incorporate an additional dataset containing information about each country (longitude, latitude, name). The dataset is focused on Anime and the Otaku community, offering comprehensive data for demographic analysis and trends within this group. It is designed to serve as a representative sample of the internet Otaku community, providing data on users (gender, location, birth date), anime details (genres, producers), and anime lists. All this information is gathered from the MyAnimeList website.

Within this dataset, we have **three main tables** :

Size of each table :

— **Anime** : 6,668 different anime entries (contains information about individual anime).

— **User** : 108,711 different users (contains information about users who watch anime).

— **Anime-User** : 35,802,010 records (represents anime lists of all users). Since this table is extensive, we will later, in the subsection on the Filtered version of the dataset, base our selection on the distribution of the number of times each anime has been watched. We will then consider only the 200 most popular anime. Therefore, our new version will consist of only 7,468,938 records, focusing on the most popular 200 anime.

The detailed method for filtering the Anime-User table, along with the characteristics of the dataset, will be covered in the corresponding Appendix.

2 Anime graph (based on users)

2.1 Constructing the Graph

For this graph, I used a filtered version of the Anime-User dataset, which contains 7,468,938 records. I constructed an unweighted undirected graph with nodes representing anime, and an edge exists if the number of common users falls within a predefined range. This will be explained in detail in the corresponding Appendix section.

2.2 Communities detection

We would also like to conduct an analysis on community detection. In this section, we utilize the Louvain algorithm with a resolution of 1.0 to identify three distinct communities.

2.2.1 Communities characteristics

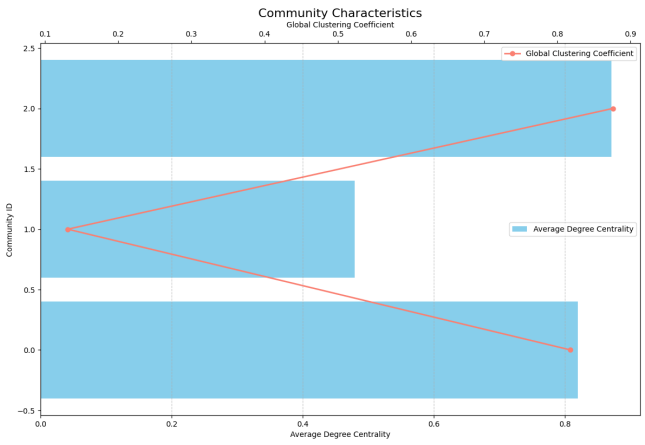


FIGURE 1 – Characteristics of each community (Average degree centrality & Global clustering coefficient)

As a result of the Louvain algorithm, we obtain three communities (Figure 1) :

1. **Community 0** : Includes 65 nodes, 1705 edges (Average degree centrality = 0.8197, Global clustering coefficient = 0.8178). The nodes in Community 0 are closely linked, forming a strong and well-knit network. The high clustering coefficient suggests that the nodes tend to form closely connected clusters within the community.

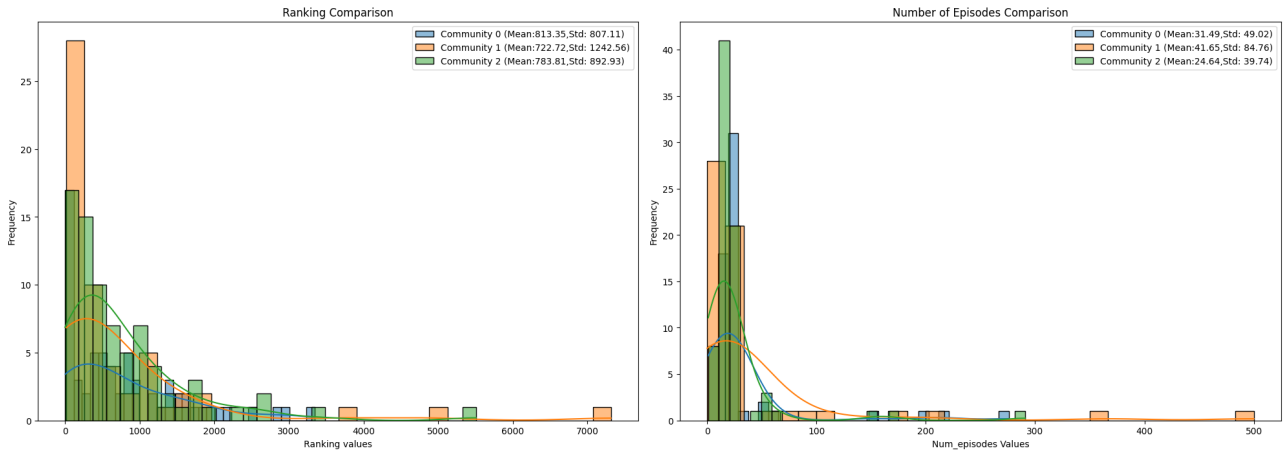
2. **Community 1** : Includes 57 nodes, 765 edges (Average degree centrality = 0.4793, Global clustering coefficient = 0.1311). Nodes in Community 1 are not as closely tied together, and the overall structure is more dispersed. The lower clustering coefficient indicates that nodes are less likely to form tight groups within this community.
3. **Community 2** : Includes 78 nodes, 2616 edges (Average degree centrality = 0.8711, Global clustering coefficient = 0.8758). Nodes in this community have numerous connections, forming a well-organized and closely-knit structure. The exceptionally high clustering coefficient suggests strong interconnectedness and clustering within the community (more than Community 0)

2.2.2 Additional Information - Joining with Anime table

Analyzing the graph alone provides information about interconnectedness and the density of structure within each community. However, I realized that by combining this information with the Anime table, we could gain more insights about each community of anime.

During the analysis of the communities, I observed that they share a focus on types like [Action, Comedy], but Community 0 emphasizes Romance more. Community 1 is associated with Drama, and Community 2 is linked with Supernatural. Additionally, across all communities, TV shows dominate over other types (such as 'Movie', 'OVA', 'Light Novel', 'Visual Novel',...). The popularity is roughly the same at around 120, although my approach to constructing the graph could highly potentially influence this measure.

Furthermore, I noticed differences in ranking and the number of episodes between communities.



(a) Ranking comparison between communities

(b) Number of episodes comparison between communities

FIGURE 2 – Differences between communities

Concerning the ranking 2a, Community 1, associated with Drama, has the highest mean ranking (722). However, the quality is not uniform, as we observe both a significant number of highly-rated movies and a few very low-quality movies (around 7000 in ranking). Community 0 (related to Romance) and Community 2 (related to Supernatural) have a lower mean ranking but are quite concentrated within the range of 0 to 3500.

Regarding the number of episodes 2b, as our data is dominated by TV shows, we observe that most anime have around 10 to 50 episodes per season. Distinctively, Community 1 (related to Drama) includes some anime with more than 500 episodes, which is rare. The supernatural theme (related to Community 2) has the lowest median of 25 episodes, while the Romance theme (Community 0) has 32 episodes, and the Drama theme (Community 1) has 42 episodes.

2.3 Centralities Exploitation

Since most of our users watch popular anime, we expect to retrieve them because they will form hubs in our graph. We can identify these hubs using betweenness centrality (see Figure 10) or pagerank centrality (see Figure 11).

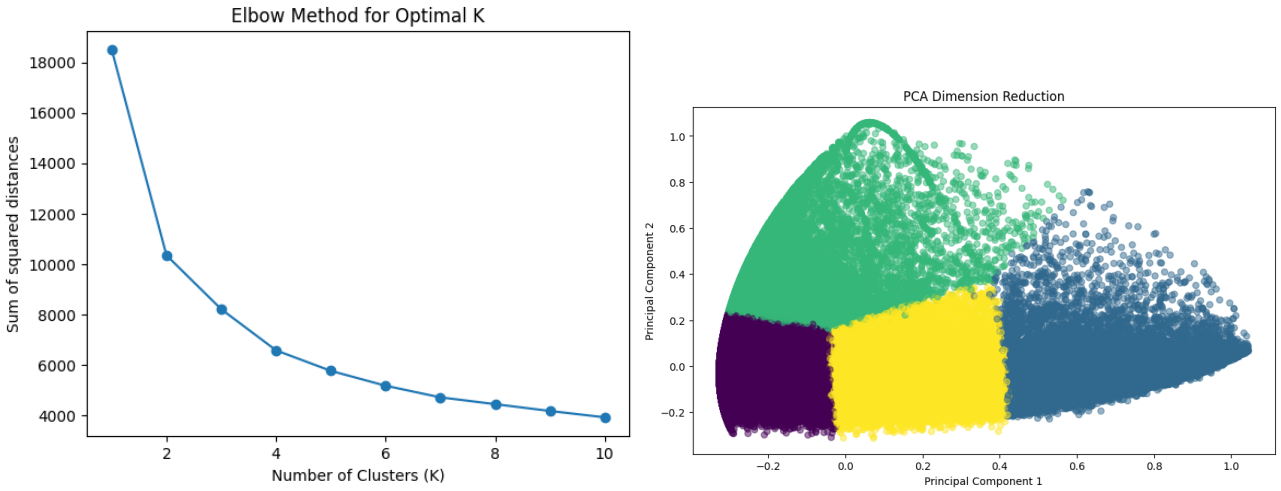
This way, we can easily identify highly reputed anime such as *Bleach*, *One Piece*, *Dragon Ball*, or *Great Teacher Onizuka*, as well as famous films like *My Neighbor Totoro* and *Akira*.

3 User clusterings

3.1 Pre-processing data

For this problem, I utilize the User table, which consists of 108,711 different users. I aim to understand different groups of users based on their patterns of watching/reading anime. Therefore, I will base my clustering on the following properties :

- `UserWatching` : Number of animes that the users are currently watching/reading.
- `UserCompleted` : Number of animes that the users have completed watching/reading.
- `UserOnHold` : Number of animes that the users have put on hold.
- `UserDropped` : Number of animes that the users have dropped.
- `UserPlantoWatch` : Number of animes that users plan to watch in the future.
- `UserDaysSpentWatching` : Number of days that the users spend watching/reading (per year).



(a) Elbow method to choose number of clustering (I choose 4)

(b) Dimension reduction for visualization (Kmeans clustering results)

FIGURE 3 – Kmeans clustering

All of the data is numeric and is normalized (because I use K-means methods in the later section, which is sensitive to the scales of data). To choose the number of clustering, I use the elbow method (as in the following Figure 3a). Furthermore, I carry out the dimension reduction with PCA method to verify the quality of clustering (Figure 3b).

3.2 Characteristics of Clusters

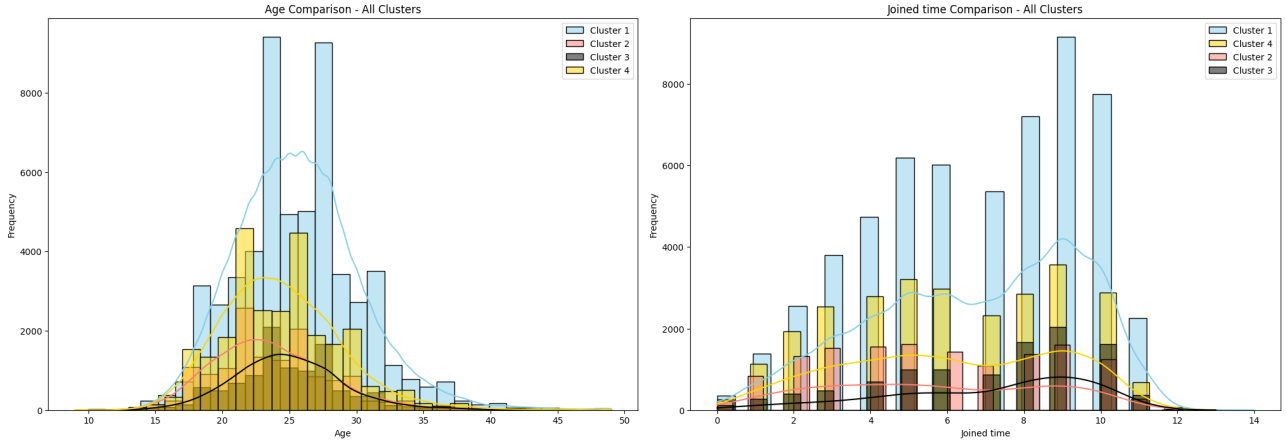
To further analyze the results of PCA dimension reduction, I would like to examine the characteristics of each centroid (as shown in Figure in the corresponding Appendix section) :

- **Cluster 1** : Users belonging to this group show great commitment to watching/reading anime. They have finished a large number of anime and spend a good amount of time (per year) on their hobbies. However, currently, they are not watching many anime.
- **Cluster 2** : These users have a lot of anime to watch in the future, but due to their limited time budget, they have not read many anime.
- **Cluster 3** : This group has the most large amount of time to watch anime (most passionate). They have read a huge number of anime and are currently reading a lot of anime as well.
- **Cluster 4** : The last group has a good amount of time for watching. They have completed a lot of anime and have a considerable number of anime planned to watch in the future.

3.3 Additional information

To further analyze the characteristics of clusters, I would like to investigate whether other factors, such as age or the duration of time the user has been on the platform, are related to the patterns observed in each cluster or not. Since the data is collected at the early of 2018, I calculate the age only until 2018 (to preserve the integrity of analysis).

Regarding age 4a, Cluster 3 has an average age of around 25 years old. These users have been working for a few years and spend their free time watching anime. That's why this group has a lot of time spent on anime



(a) Age comparison

(b) Amount of time from creating account on the platform

FIGURE 4 – Additional factors affecting the patterns belonging to each cluster

Antecedents	**Consequents**	**Confidence**	**Lift**
(Clannad, Toradora!)	(Clannad : After Story)	0.808640	1.843010
(Clannad : After Story)	(Clannad, Toradora!)	0.798531	1.843010

TABLE 1 – Association Rules Analysis (Selected Columns)

reading/watching. Furthermore, Cluster 2 and 4 have an average age of around 22, indicating that they are still studying. This could be the reason why they don't have much time to watch anime due to exams, TD, TP, and other academic commitments.

Concerning the time of joining the platform 4b, Cluster 1 has two main groups (5 and 9 years). We observe roughly the same characteristics for other clusters (2, 4). In Cluster 3, nearly every user has been watching anime for 9 years. This further confirms our hypothesis that this is the most passionate group.

4 Anime/Genre Recommendation - Frequent Patterns

I use the Frequent Patterns technique to further explore the problem of Anime/Genre Recommendation.

4.1 Anime Recommendation

4.1.1 Pre-processing the data

For Anime recommendation, I use the filtered version of the Anime-User table. So, I have 200 animes along with 108,296 users (a matrix of 108,296 x 200). Each user is represented as a transaction (a row), and the values of a row should be 0/1 (indicating whether they have watched the anime or not). Each anime corresponds to a column.

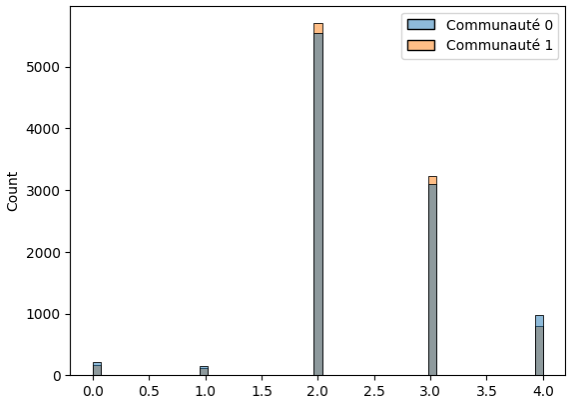
4.1.2 Analysis based on Apriori algorithm's result

I perform the Apriori algorithm ($min\_support = 0.35, maxlen = 4$), and then I rank the results using the **LIFT** metrics. From the results, I get some interesting information regarding anime recommendation :

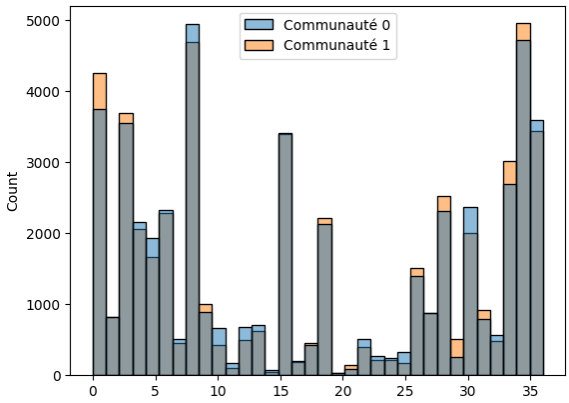
1. **Always recommend sequences of anime** (After reading/watching an anime, there's a high chance people will look for sequences of it) (Table 1) :

Antecedents	**Consequents**	**Confidence**	**Lift**
(Death Note, Code Geass : Hangyaku no Lelouch R2)	(Code Geass : Hangyaku no Lelouch, Fullmetal Alchemist : Brotherhood)	0.786678	1.724925
(Code Geass : Hangyaku no Lelouch, Fullmetal Alchemist : Brotherhood)	(Death Note, Code Geass : Hangyaku no Lelouch R2)	0.779510	1.724925

TABLE 2 – Association Rules Analysis (Selected Columns)

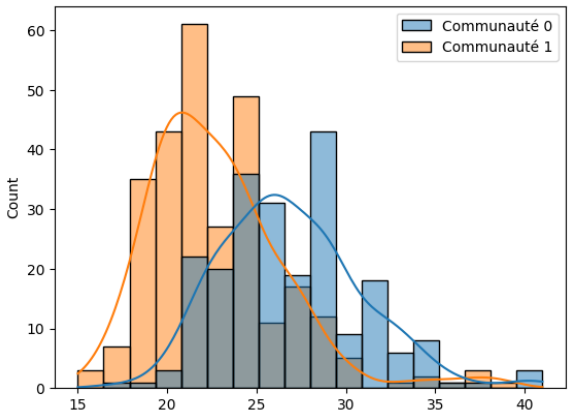


(a) Rating of anime watched (0 : "PG - Children", 1 : "G - All Ages", 2 : "PG-13 - Teens 13 or older", 3 : "R - 17+ (violence & profanity)", 4 : "R+ - Mild Nudity")

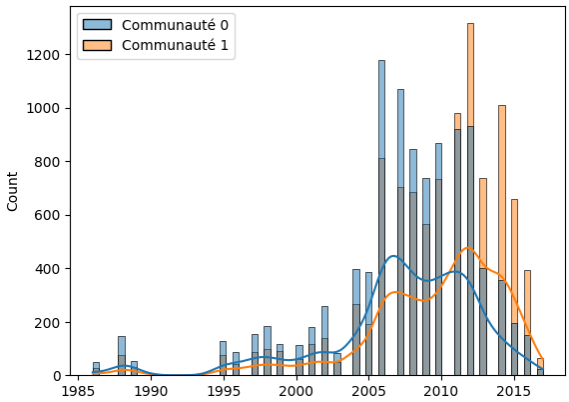


(b) Genre of anime watched

FIGURE 5 – Comparison of data distributions in the two communities



(a) Age of users



(b) Airing year of anime watched

FIGURE 6 – Comparison of data distributions in the two communities

2. **Similarity in genres, and highly similarity in context of anime** (Both Death Node and Code Geass concern a mind game) (Table 2).

5 User graph (based on animes)

5.1 Communities detection

We used the community detection algorithm of Gephi based on Modularity (default parameters), and we identified two communities that we want to characterize (see Figure 13). To achieve this, we compared some data distributions of our two communities.

The two clusters seem to watch the same types of anime when we examine the rating and genre distribution of anime watchers within the communities (see Figure 5a and Figure 5b).

A factor that could explain the difference between our communities could be the generation of our community users. Community 0 appears to be older than Community 1. Additionally, the younger community is the one that watches more recent anime (see Figure 6a and Figure 6b).

6 Genre Exploration

6.1 Processing

We employ the same anime and user processing as used in the construction of the user graph to select them. We then join this data with the anime dataset to retrieve genres and titles. Finally, we obtain a matrix with a

row for each user, and its columns represent the genres. Each column is filled with a 1 if the user has watched the genre, and 0 otherwise.

6.2 Frequent Pattern and Association Rules

We use the Apriori algorithm to extract the most frequent patterns, which are *(Action)*, *(Comedy)*, and **(Supernatural)**. Additionally, we identify larger item sets such as *(Comedy, School)* (like Great Teacher Onizuka, Assassination Classroom) or *(Shounen, Action)* (One Piece, Naruto) that are common and popular combinations in the anime panorama.

Another way to utilize our association rules could be to recommend the consequent of a rule to a given user if we identify the antecedent in their data.

antecedents	consequents	lift
(Comedy, SliceofLife)	(School)	2.77
(Adventure)	(Fantasy)	2.47
(Comedy, Action)	(Adventure)	2.44
(Comedy, Romance)	(School)	2.12
(Shounen, Adventure)	(Action)	1.9

(a) Association Rules and Lift

Latent Variable	Explanation
d1	Action
d2	Romance/comedy/Shōjo
d3	Shōnen/Fantasy/SF/Fantastic
d4	Poetic
d5	Drama/psychological

(b) Latent Variables

TABLE 3 – Corresponding results for Section 6.2 and 7.2

With this support, we can evaluate some association rules (as in Table 3a)

Some rules are predictable because certain genres are always highly related (*(Adventure) -> (Fantasy)*). Other rules can be interesting for people not familiar with anime. For example, the rule *(Comedy, Romance) -> (School)* is typical ; for an Occidental cartoon, such a combination would be an exception.

7 Anime Recommendation

Since we can obtain the rating of a user for a given anime they have watched in our dataset, we can use matrix factorization to characterize anime and observe which anime characteristics a given user likes.

7.1 Pre-processing

We use the same anime and user processing as used in the construction of the user graph to select them.

7.2 Matrix Factorization

We employed non-negative matrix factorization to identify 5 latent variables that characterize each anime. Once completed, we attempted to assign a noun to each latent variable (see Figure 3b). This was achieved by searching for commonalities among animes with high values for a given latent variable (see Figure 4). We utilized our personal knowledge and Nautiljon.com as a secondary database for anime we don't know.

	One Punch Man	Kami nomi zo Shiru Sekai	Bleach	Spirited Away	Cowboy Bebop
d1	**22.8**	0	4.5	2.0	1.44
d2	0	**18.7**	0	0	0.3
d3	1.0	3.2	**15.6**	1.4	4.1
d4	0	0.9	2.3	**9.3**	0
d5	1.5	0.8	0.1	5.0	**5.9**

TABLE 4 – Some Anime Characterization through latent variables

Users are also characterized through a sort of rating for every latent variable, indicating whether they are sensitive to the characteristic or not. Based on these values, we can find an anime to recommend to a given user using the anime characteristic database (4).

A Division of Work

Github project's link $VietAnhNguyen20/M2\_DM\_Anime$
Our group consists of 2 members, and the tasks are divided as follows :

— Viet Anh NGUYEN is responsible for the following sections : Anime graph, User clustering, Anime Recommendation (Frequent Patterns).
— Elliot FAUGIER is in charge of the following sections : User graph, Anime Recommendation (Collaborative filtering), Gendre Recommendation (Frequent Patterns).
— Other tasks, including finding and pre-processing the data, ... , are collaborative efforts.

Before working on this project (AnimeDataset), we had dealt with approximately 10GB of music metadata. However, despite making intensive efforts to process and select valuable information based on distribution, our machines were unable to handle even the filtered version of the music dataset.

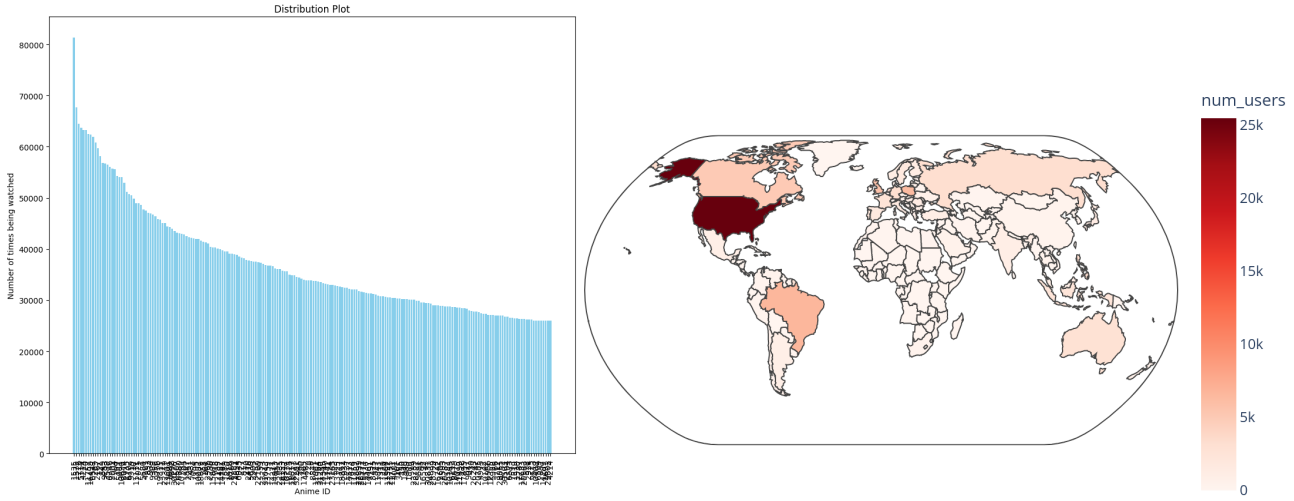
As a result, we decided to switch our dataset to the AnimeDataset, which is around 2GB in size. Through various pre-processing steps based on distribution, we managed to reduce the dataset to a few hundred megabytes of valuable data, making it suitable for analysis (under the conditions of our machines).

B Overview of dataset

B.1 Filtered version of dataset

The Anime table and the User table remain unchanged for further analysis using different techniques (Clusterings/ Collaborative filtering/ Frequent patterns). However, as we construct a graph based on the Anime-User dataset, which contains 35,802,010 records, the resulting graph becomes too large for feasible calculations. Therefore, we attempt to filter the dataset.

We aim to select the 200 most popular anime and their corresponding records. To achieve this, we perform a groupby operation on the AnimeID to calculate the number of times each anime has been watched. Subsequently, we draw the distribution for each anime, following the Power-Law distribution. As a result, we only consider the 200 most popular anime and extract all the records associated with these selections (7,468,938 records).



(a) 200 most-watched anime chosen (instead of all 6,668 anime)

(b) User distribution based on countries

FIGURE 7 – Characteristic of dataset

B.2 Characteristics of dataset

As we explore some characteristics of this dataset, we have identified certain biases. Despite the fact that anime/manga are predominantly created in Japanese, the platform (MyAnimeList) from which the data is collected (spanning from 2006 to 2018) was originally founded in Canada and later shifted its main services to the United States. Consequently, a majority of the users primarily came from the United States (followed by Canada) and other English-speaking countries (Figure 7b.

C Anime graph (based on users)

C.1 Constructing a graph

I follow the following specific steps to create a graph for anime based on users (2 steps) :

1. **Weighted Undirected Graph** : Each anime corresponds to a node in the graph. The link will exist if two animes have common users watching them (the weight will be the number of users). The graph is undirected since we only care if two animes have a number of users watching them in common, and we don't care about the order of watching.

   However, we find that the edge weights actually follow a distribution resembling a Gaussian distribution. It means that a few animes are so popular that they're watched by everyone (so the edge between these two animes would have a really high value). Conversely, there are some animes with poor quality that don't have many people watching (so the edge's weight would be really low). So, we want to focus on not too popular animes and also not too bad animes. For this reason, I remove every edge with weights less than 5000 or larger than 35000. The remaining distribution is shown in the following Figure :

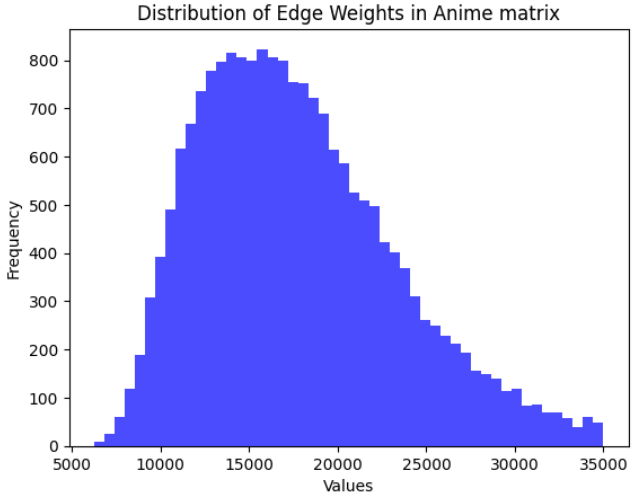


FIGURE 8 – Edge weights distribution (Keeping only values from 5000 to 35000)

2. **Non-Weighted Undirect Graph** : I would like to convert this graph to the non-weighted version. To do so, I calculate the mean and the standard deviation (std) of the remaining distribution. I only keep the edges with weights in the range of $[mean - 0.8 * std, mean + 0.8 * std]$ (which means [13318, 22151]), and then remove the weights (Edges are not weighted). For other edges, I eliminate them all. As a result, I have a non-weighted undirect graph with 200 nodes and 10,917 edges.

C.2 Visualization of Anime Communities

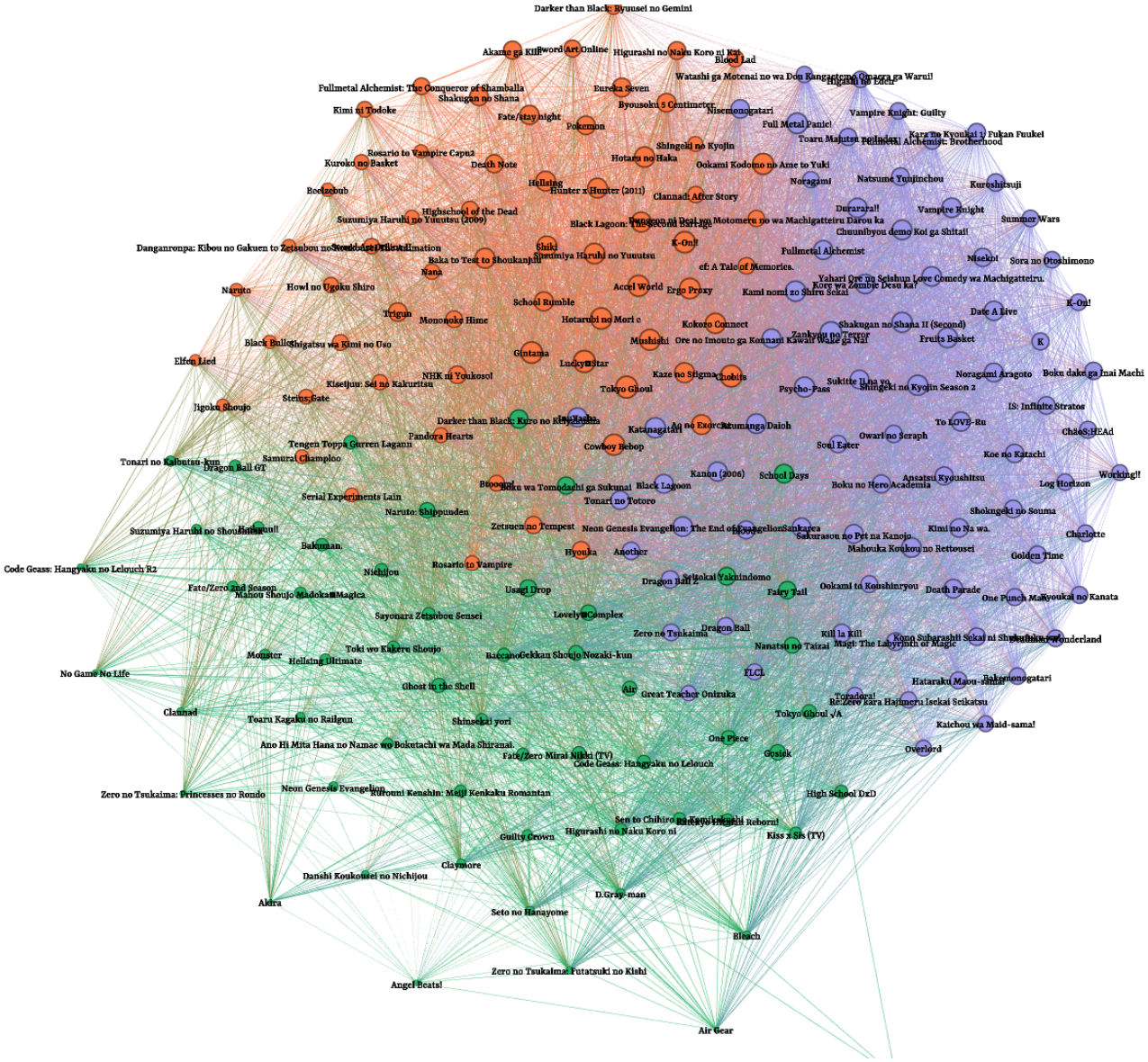


FIGURE 9 – Community plot (with Gephi)

C.3 Centralities

To employ centralities, we used the same anime graph as mentioned in the construction section, without deleting edges that have a weight greater than 35,000. This way, we preserve the importance of highly-watched animes.

C.3.1 Betweenness

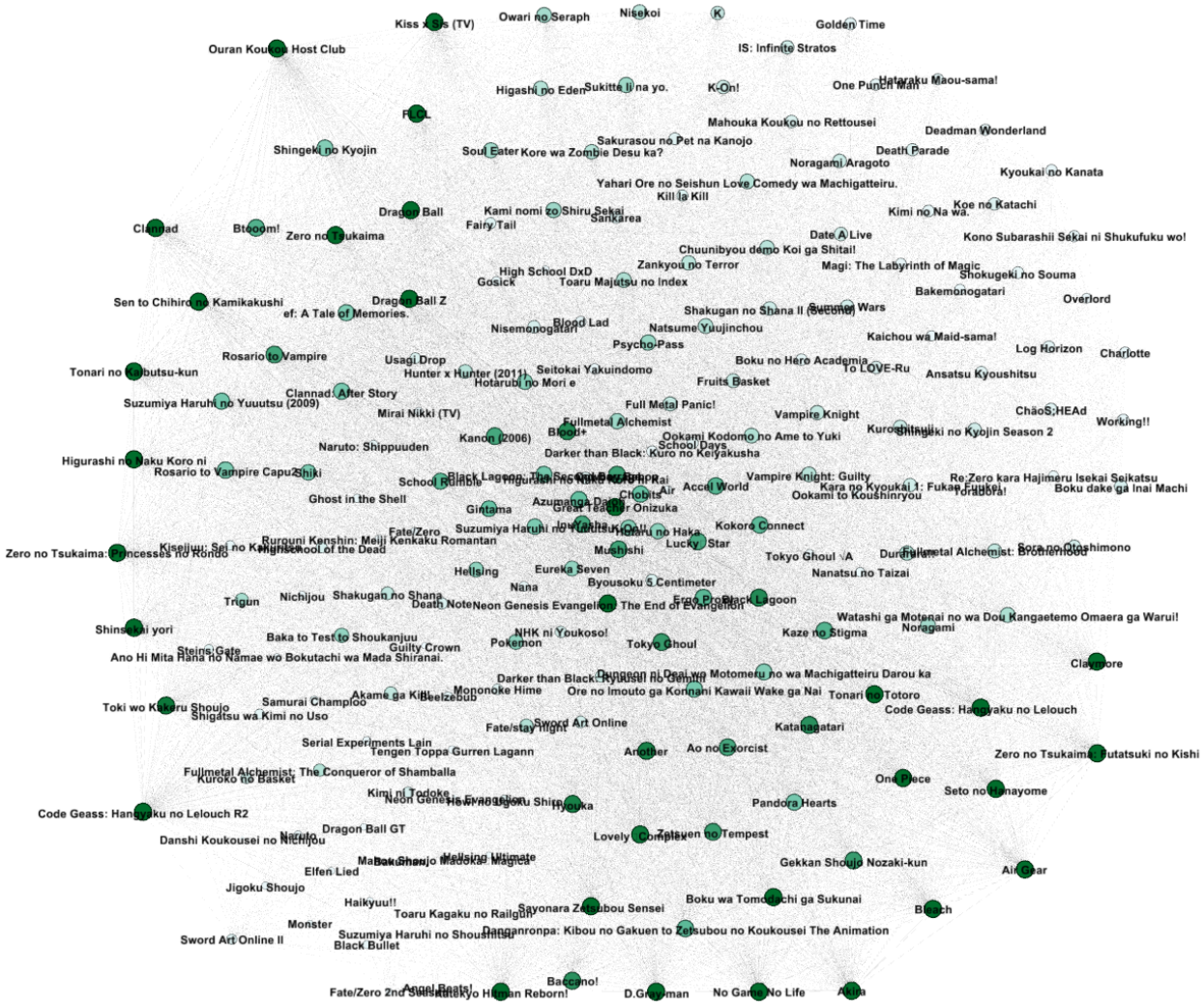


FIGURE 10 – Anime Graph colored depending on their betweenness

C.3.2 Pagerank

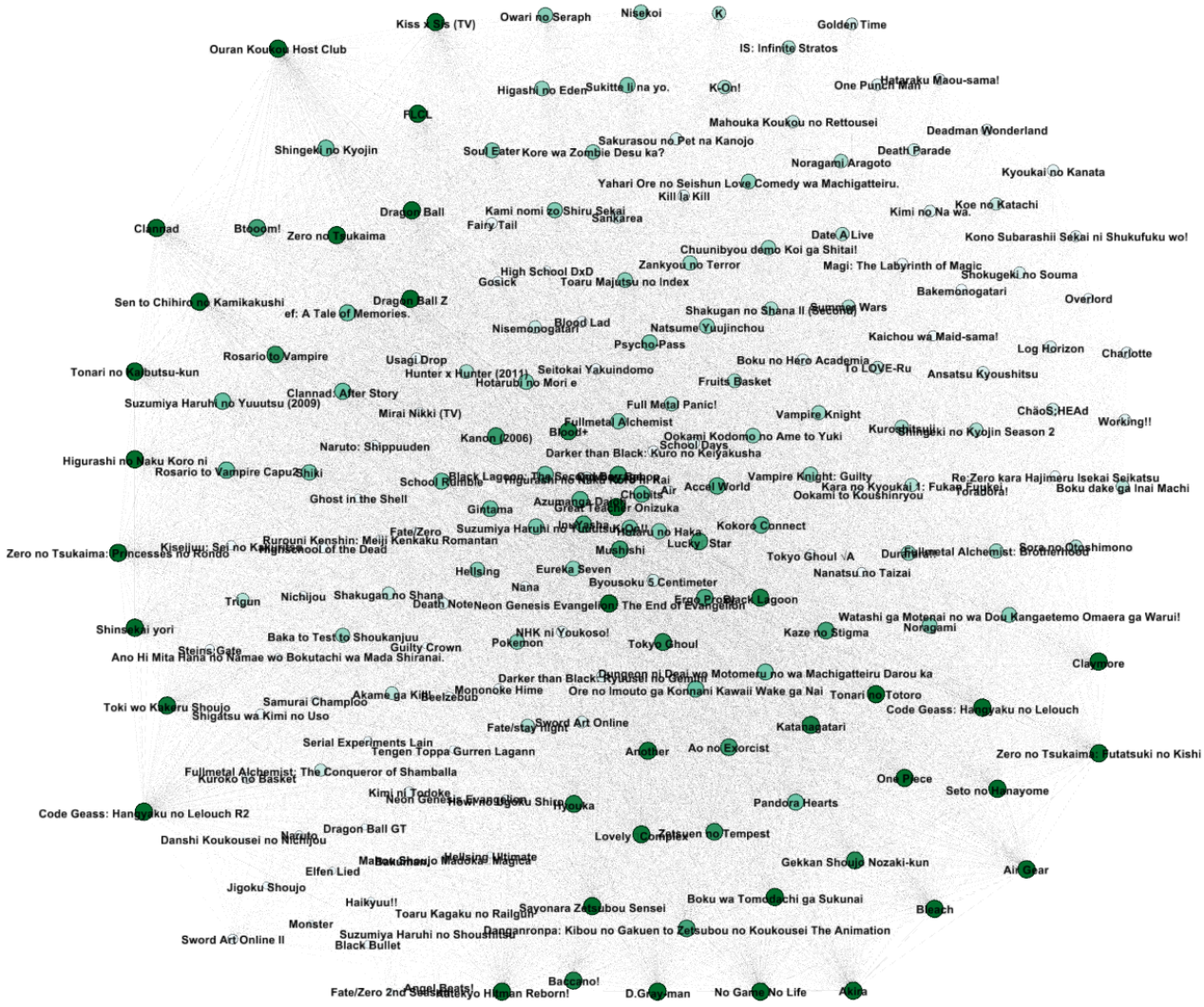


FIGURE 11 – Anime Graph colored depending on their pagerank

D User clustering

The corresponding characteristic for each user's clustering :

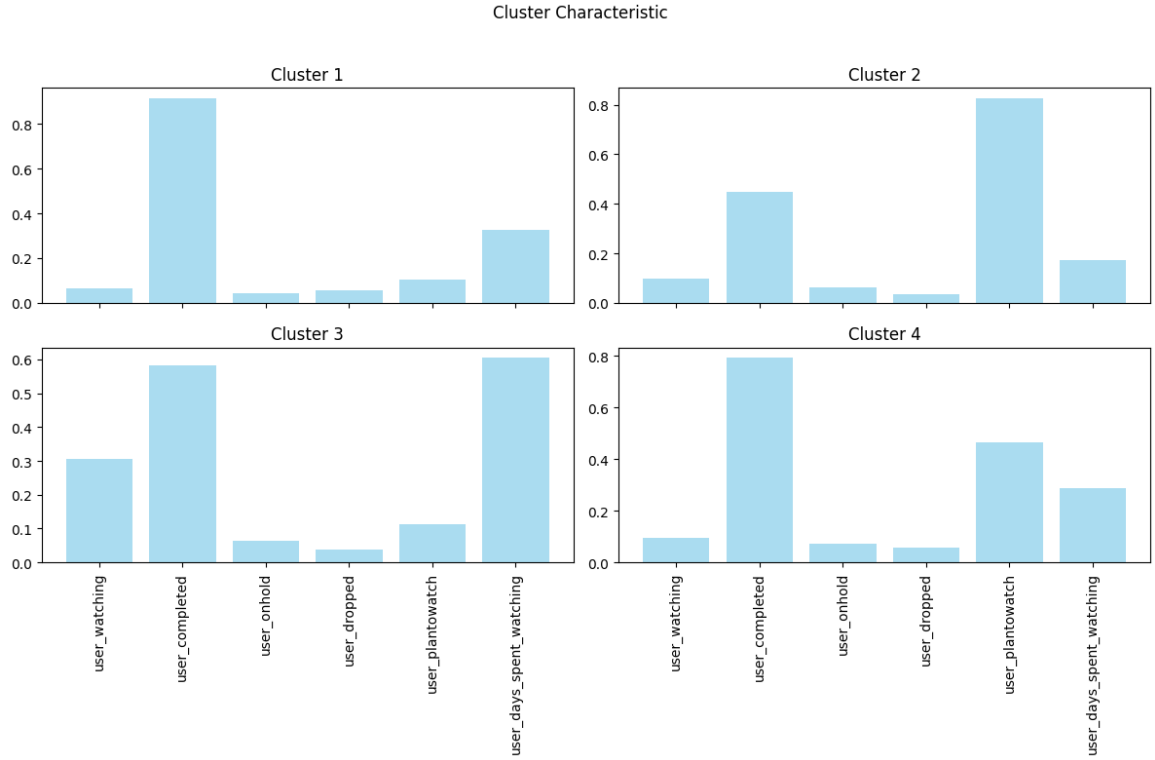


FIGURE 12 – Characteristics of each clustering (measured on centroid)

E User graph (based on animes)

E.1 Constructing the Graph

We have a few datasets containing every anime a user has watched. Based on this data, we want to construct a graph where users are nodes. We create a link between two users if they have watched the same anime. Couples of users can watch different animes ; to capture this, we increase the weight of the link accordingly. At this step, the adjacency matrix of our user graph is too large to fit into memory. To reduce the size of our graph, we filtered our data through two processes :

— Anime filtering : We selected the 200 most-watched animes.

— User filtering : We aim to sample users while retaining a representative subset, observing both well-informed watchers and casual watchers. To achieve this, we used the number of animes watched as a metric to represent the awareness of our users and sorted them based on this value. Finally, we selected 500 users around the mean of our metrics at regular intervals to sample a large variety of users. (Note : This sample may not be fully representative because we lost the notion of the proportion of informed users.)

Now, we can compute our adjacency matrix, but we encounter a graph density issue. To address this problem, we use a threshold to remove links that are too weak.

E.2 Community Detection

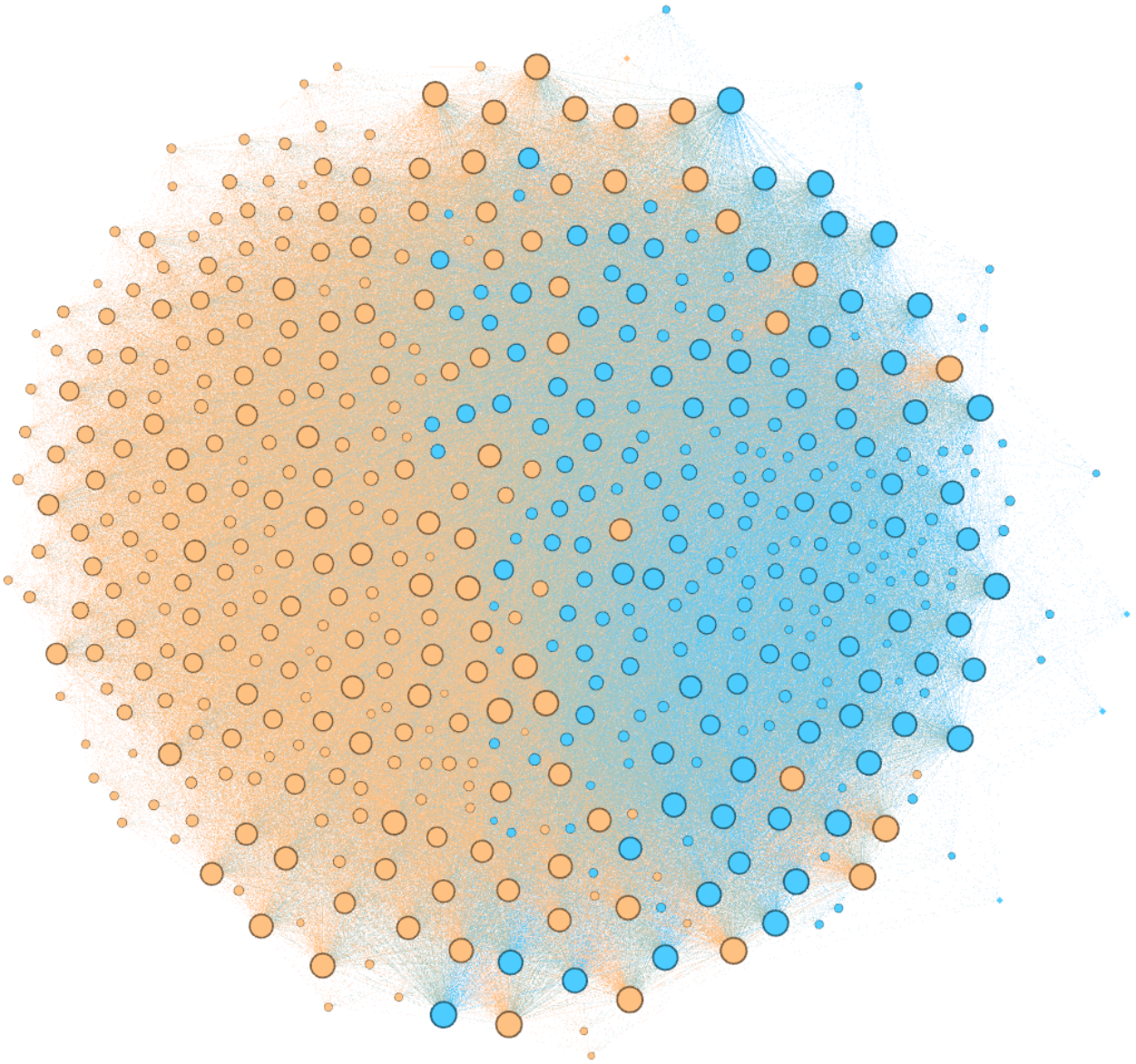


FIGURE 13 – User graph colored based on the community of users, with Community 0 represented in blue and Community 1 in beige