University of Glasgow

School of Computing Science

Reddit Data Analysis

Tao Ma

School of Computing Science
Sir Alwyn Williams Building
University of Glasgow
G12 8RZ

07/03/2025

Contents

Chapter 1 Abstract

1.1 Introduction

The main topic of this report is analysing user interactions within the *InvestmentClub* subreddit on Reddit. The study aims to discover engagement patterns and community sentiment by using a network analysis and sentiment measurement method.

1.2 Research questions

- How do user interaction and sentiment trends in the *InvestmentClub* subreddit evolve over time?
- What is the active users' preference form to engage in a discussion? What role do each active user play in this community?
- What are the daily activity patterns of users in the *InvestmentClub* subreddit?

Chapter 2 Data Preprocessing

To obtain an accurate dataset for analysis. It is necessary to use several preprocessing steps to ensure data quality and completeness, including loading and transforming Json, merging interaction, cleaning text, sentiment detection and aggregating user information.

2.1 Loading and transforming Json

The data preprocessing began with loading two JSON files containing **comments** and **submissions** from the *InvestmentClub* subreddit into two dataframe named comments and submissions. In these dataframe, timestamps which formed in Unix format are converted to a readable datetime format (YYYY-MM-DD HH:MM:SS) for analysis. Additionally, unnecessary attributes in dataframe are removed to retain only the most relevant information for analysis.

Before merging, attributes are modified to ensure consistency. **parent_id** is modified to remove the first three chars, "t1_" or "t3_", and duplicate records select only the first record.

2.2 Users Interaction

2.2.1 Merging interaction

The comments and submissions are concatenated as a new dataframe author_interact, which consist of all relevant information. In this structure, each record represents a user interaction, whether it is a comment or a submission. Currently, each record in author_interact contains attributes **author, id,** and

parent_id, but the parent author's is still missing. To map user interactions, author_interact merges itself by using **id** and **parent_id**. This allows each record to retrieve its parent author, establishing connections between users. Additionally, the attribute **created_utc** will be transformed to datetime type.

2.2.2 Clean text

To enhance data quality, the attributes **title** and **body** in author_interact are cleaned through several steps. Non-English text is translated into English for the next sentiment detection, while special characters and stopwords were removed. Additionally, the records with missing values in **the title** or **body** are filled with blanks.

2.2.3 Sentiment detection

String attributes **title** and **body** are added together to apply sentiment detection to evaluate the emotion score of discussions by using the package **nltk**. This process returns four new attributes – **negative, neutral, positive and compound – to** author_interact**.** For each new attribute:

- **Negative**: proportion of negative text, indicating discouragement.
- **Positive**: proportion of positive text reflecting optimism.
- **Neutral**: proportion of neutral text indicating factual, ranging from 0 very emotional to +1 objective.
- **Compound**: normalized sum of positive and negative, ranging from -1 most negative to +1 most positive.

2.2.4 Summary

This process produces a key dataframe, author_interact, which stores detailed information and connections for each comment and submission.

2.3 Aggregate user information

2.3.1 Replied and received

To quantify user engagement, it is necessary to measure output and input interactions. Output interactions refer to situations where users reply to others, while input interactions represent the number of replies a user receives. To aggregate these interactions, author_interact can be used:

- interact_out**:** group author_interact by **author**. For each user, aggregate the authors they interacted with into a list. Record the total number of replies in the attribute **interact_out** based on the size of the list, then remove all the NaN values in the list since submissions do not have a parent author. Making the nodes in the list unique to count the unique users they replied to and store in the attribute **interact_author_out**.

- interact_in: similarly with interact_out, group author_interact by **author_parent** to collect all users who replied to a given author. Store the result in the attribute **interact_in** and **interact_author_in**.

2.3.2 Count submissions and comments

In addition to interaction metrics, user contributions are also evaluated by counting the total number of **submissions** and **comments** made by each user. Simply count the number of records in comments and submissions grouped by their author and store them in the dataframe submissions_count and comments_count.

2.3.3 Aggregating user metrics

To provide a comprehensive overview of each user's activity, sentiment, and influence. The necessary dataframe author_interact, interact_out, interact_in, submissions_count and comments_count are prepared to construct the final dataframe author, where each row represents a user.

1. Merging interact_out and interact_in interactions
    a. The interact_out and interact_in are merged by outer join, ensuring all authors are included.
    b. Missing values are filled with zero for users who only reply or receive.
2. Count the total interactive users
    a. In author, empty values in interaction lists from interact_out and interact_in are replaced with empty lists.
    b. Both input and output interaction lists are combined to create a total interaction list for each user.
    c. For each author, store the number of unique users from this total interaction list to attribute **interact_author** in author.
3. Adding submissions_count and comments_count
    a. The number of submissions and comments for each user is added by merging with their respective author.
4. Aggregating engagement metrics
    a. Upvotes (ups), downvotes (downs), and overall scores (score) are summed for each author.
5. Aggregating sentiment
    a. The mean sentiment scores of **negative**, **neutral**, **positive**, and **compound** are added for each user based on their records.

2.3.4 Remove [deleted] users

To guarantee the accuracy of user interaction analysis, deleted users ("[deleted]") are retained for the initial processing stage. This enables a full historical count of all user interactions, even if some users were later removed from the subreddit. However, after the dataframe author construction is finished, which includes aggregating interaction counts, sentiment scores and engagement metrics, deleted users are removed since "[deleted]" is an invalid user and cannot function as a node in an interaction graph. Filtering is applied at the final stage. Specifically:

- In author, find the author names "[deleted]" and remove it.

- In author_interact, each record either attributes **author** or **author_parent** as "[deleted]" is excluded to prevent invalid edges in the user interaction graph.

2.3.5 Summary

All the data preprocessing is finished after the key dataframe author and author_interact are set up correctly.

Chapter 3 Graphs visualization

To visualise a graph for entire user interaction, a directed graph structure is recommended. In the directed graph, nodes represent users while edges refer to interactions between users. The author can be used to define the nodes and the author_interact link users based on records.

3.1 building graphs

3.1.1 Origin Nodes and Destination Nodes

By using author, each unique author in the dataframe is added as a node, with attributes of their interaction counts, submission and comment activity, sentiment scores, and engagement metrics.

Origin node represents the user who starts the conversation by replying to a post or commenting. In this interaction, these users make a comment to send a message to other users.

Destination nodes, in contrast, are users who receive messages from other users, meaning submissions or comments of these users are responded to by others. These users are the starting points of a discussion.

3.1.2 edges

By using author_interact, direction edges are established from the author of this record to the parent author. A record representing a submission which does not have a parent author or the direction edge has already been added will be skipped to avoid adding a duplicate direction edge. These can guarantee that each user is linked to the corresponding user they interact with.

3.1.3 Coloring

Each node is color-coded based on the number of users it has interacted with, using a manually setting from low interaction white to high interaction gold.

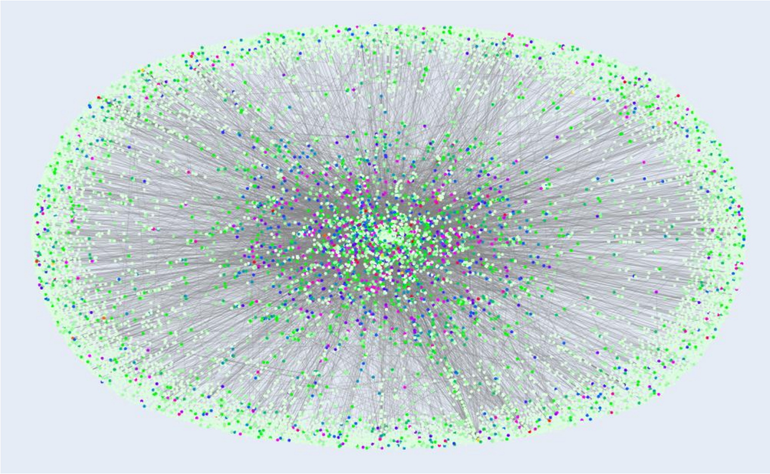


Figure 1 entire interaction graph

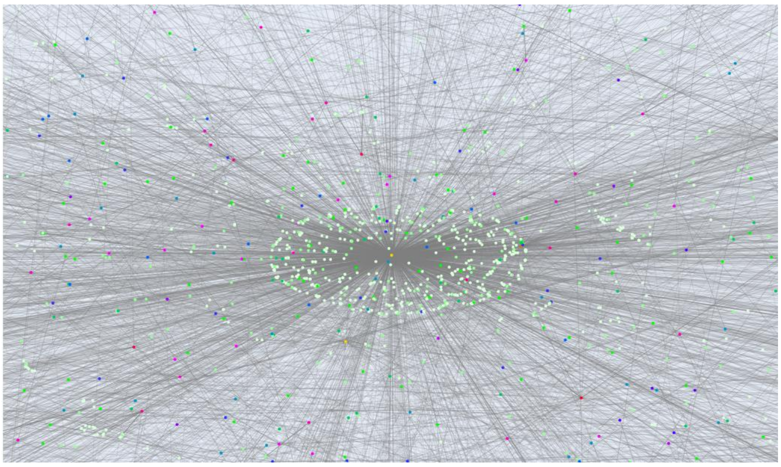
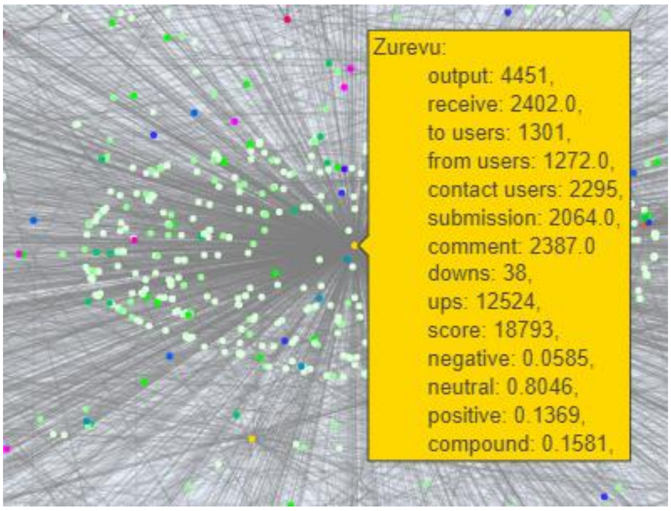


Figure 2 zoom to center



```
Zurevu:
        output: 4451,
        receive: 2402.0,
        to users: 1301,
        from users: 1272.0,
        contact users: 2295,
        submission: 2064.0,
        comment: 2387.0
        downs: 38,
        ups: 12524,
        score: 18793,
        negative: 0.0585,
        neutral: 0.8046,
        positive: 0.1369,
        compound: 0.1581,
```

Figure 3 node label

3.2 Interpretation and Observations

3.2.1 Overview

The visualisation of this Reddit comment interaction network shows a pattern of engagement, with a core group of highly active users forming the central hub of discussions. The figure 4 represents the entire dataset and demonstrates a highly connected core surrounded by a peripheral with fewer connections. This suggests that many participants engage in discussion at a low level. A subset of users plays a pivotal role to keep the network active.

3.2.2 Zoom in detail

When zoomed in to the centre, as in figure 2, a super user represented in gold can be observed. This user have a significantly higher number of interactions compared with other users. This most active superuser is located at the centre of this graph and surrounded by numerous low-activity users in white, who likely participate by engaging with these central superusers.

3.2.3 Label

By checking the most active user in the center of the graph as in figure 3, some useful information can be learned. For the most active user:

- Named: Zurevu,
- Output: the number of comments to other users and submissions is 4451.
- Receive: receive comments from other users has 2402 amount.
- To users: there are 1301 users have received message from this super user.
- From users: there are 1272 users sent message to this super user.
- Contact users: the total amount of users who has been interacted with is 2295.
- Submission: 2064 submissions created by this super user.
- Comment: 2387 comments created by this super user.
- Downs, ups, score: the total amount of upvotes, downvotes, overall scores across all interaction of this super user.
- Negative, neutral, positive and compound: scores of sentiment across all interaction of this super user.

Chapter 4 Network analysis

4.1 Evolution over time

4.1.1 Metrics and measures

The following key metrics should be considered to analyse how user activity and sentiment evolve over time:

- **Total_count**: Total amount of Interactions for all users each month

- **Super_count**: Total amount of Interactions for 10 most active users each month.
- **Total_compound:** mean compound of Interactions for all users each month.
- **Super_compound:** mean compound of Interactions for 10 most active users each month.
- **Total_neutral:** mean neutral of Interactions for all users each month.
- **Super_neutral:** mean neutral of Interactions for 10 most active users each month.

To track activity and sentiment trends over time, these metrics can be aggregated by grouping author_interact by year-month and using the sum or mean function to construct a dataframe for visualization. Two pseudo-code examples:

```
time_interact_total = author_interact.groupby(year-month).size(records)
```

```
time_compound_total = author_interact.groupby(year-month).mean(compound)
```

4.1.2 Interaction evolution

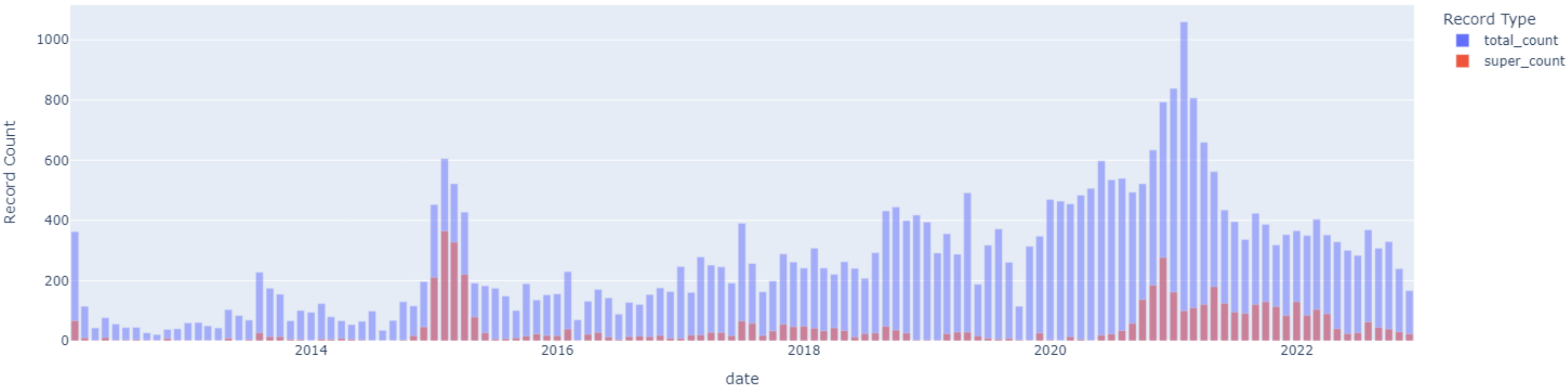


Figure 5 interaction over time

The number of blue total interactions and red active user interactions grouped by month have been visualized by this interaction evolution chart.

1. **Early Period (2013-2016):**

There was fluctuated activity in the subreddit during this period, but it was relatively low. Occasional peaks in interaction suggest that engagement had periodic explosions. During the first peak in 2015, super users were more active than at any other times.

2. **Increasing stage (2016-2020):**

After the first highest peak, total interactions increase stably, which indicates a growing community for investments, but super users were relatively less active in this period.

3. **Peak (2020-2021):**

In the peak month, there were more than 1000 interactions in the subreddit. Super users' contribution also started to be more active.

4. **Decline (2021-present):**

After the 2021 peak, interaction levels started to decline.

4.1.3 Sentiment evolution

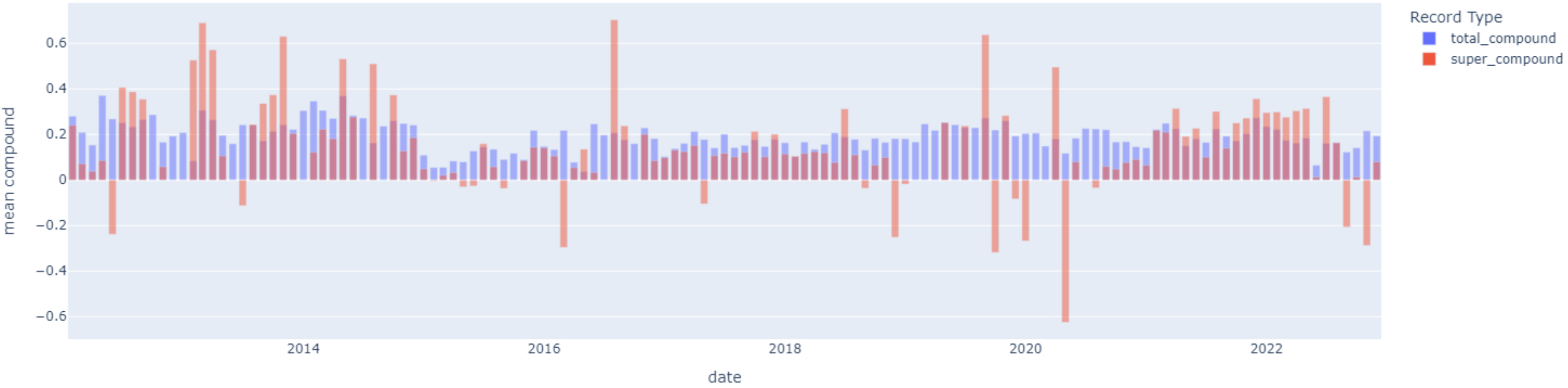


Figure 6 compound over time

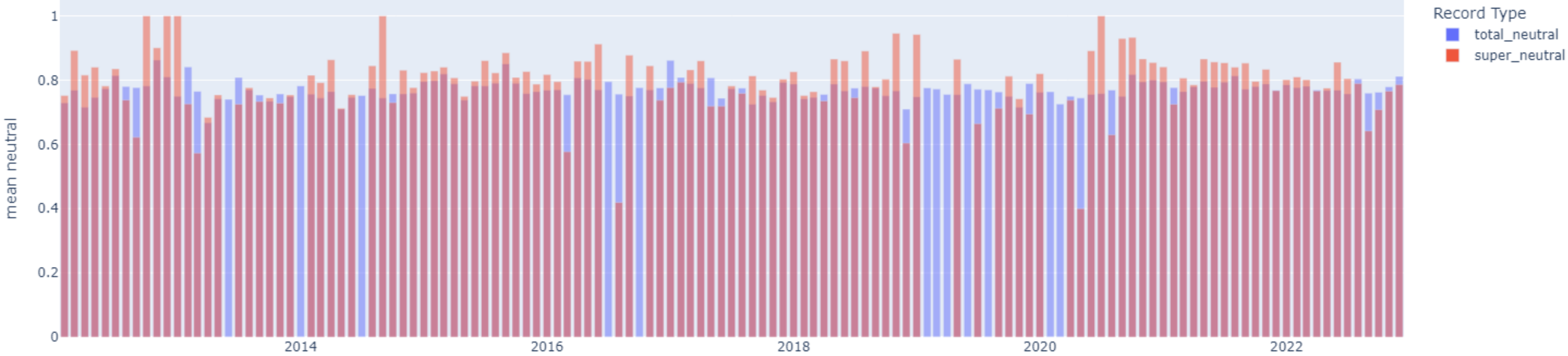


Figure 7 neutral over time

The overall environmental conditions in the community are generally optimistic since the compound level remains above 0 and around 0.2. Most of the time, the neutral sentiment score remains high, from 0.7 to 0.8, suggesting that most discussions are objective rather than emotional.

4.1.4 Conclusion

This community is generally objective and optimistic over time. Its scale gradually increased and it has two high peaks during this period.

4.2 Active user role

4.2.1 Metrics and measures

To understand active users' preference to participate in discussions, it is necessary to clarify some metrics:

- **Interact_out**: Total amount of replies for others from this user.
- **Interact_in**: Total amount of receives for this user from others.
- **Joker_super_user**: the user who comment a lot to other users but few people response.
- **Balance_super_user**: the user who comment a lot to others and receive a lot of replies from others
- **Fake_super_user**: the user who publishes a submission and receive huge amounts of comments, but the user himself does not attend any discussion.

Using author to list the 100 most active users, ranked by the number of users they have interacted with, is suitable for visualization:

```
author_super100 = author.nlargest(100, "interact_author")
```

By comparing the input and output amounts, the type of an active user can be determined.

```
joker_super_user = Interact_out >> 2*Interact_in
```

```
balance_super_user = Interact_out ≈ Interact_in
```

```
fake_super_user = 2*Interact_out << Interact_in
```

Additionally, since the most active user has much more interaction than other users, the y axis should be logarithmic:

```
compare_fig.update_layout(yaxis_type="log")
```

4.2.2 User interactions

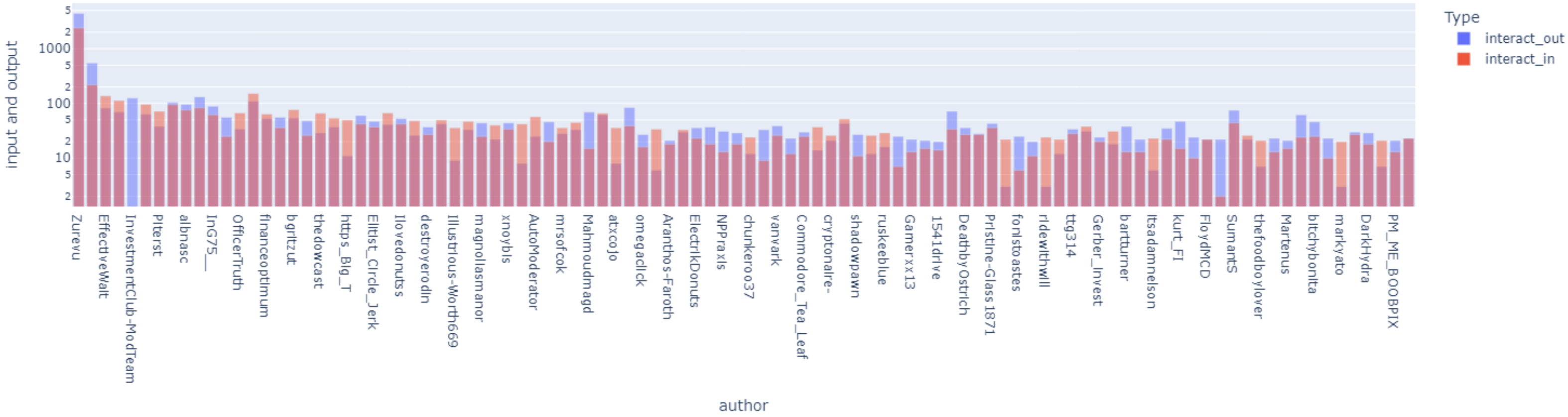


Figure 8 top 100 active users

According to figure 8, most of the active users are balanced, they both bring interesting topics to the community and keeping interact with other users. Two users whose bar color is almost blue indicate that they keep replying others, but the replies are rare. Three users with bars of half or more orange color suggest that they are relatively not active in replying others, but their comments or submissions bring considerable discussions. Additionally, since the y axis is logarithmic, the distance between input and output will be larger than the diagram display.

4.2.3 Conclusion

Most active users have a balance of input and output, few active users prefer single sided.

4.3 Daily pattern

4.3.1 Metrics and measures

To study user daily participation patterns, the metrics below should be defined:

- **Daily_total_count**: Total amount of Interactions for all users, summarized by hour within a day.
- **Daily_super_count**: Total amount of Interactions for 10 most active users, summarized by hour within a day.

Grouping author_interact by day-hour and count the number of records across the entire dataset can obtain the necessary data for chart:

day_total = author_interact.groupby(day-hour).size(records)

4.3.2 Daily pattern and conclusion

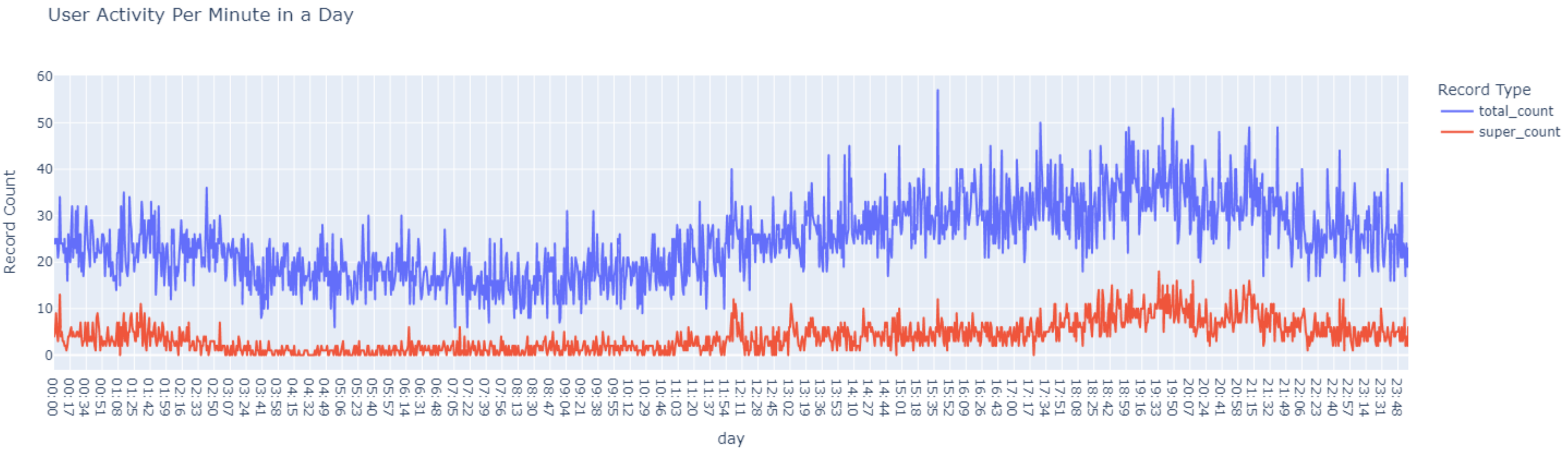


Figure 9 daily pattern

According to figure 9, users prefer to participate in discussions between 14:00 and 23:00. Active users' pattern are not as regular as normal users, which indicates that active users are more likely to engage in discussion at any time.

Chapter 5 Explore emotion distribution

To further study sentiment detail of each user, it is necessary to explore the dataframe author_interact and author deeper.

5.1 Research questions

- What is the sentiment of each interaction? Are interactions friendly and objective?
- What each user behaves in subreddit? Are they positive or negative?

5.2 Interactions distribution

5.2.1 Metrics and measures

To display the distribution of interaction, some metrics are required:

- **Compound**: an attribute has already been stored in author_interact, a scores computed by positive and negative in preprocessing chapter.
- **Netural**: an attribute has already been stored in author_interact and computed in preprocessing chapter.
- **interact_compound_total**: the number of records from all users in each compound score.
- **interact_compound_super**: the number of records from super users in each compound score.
- **interact_neutral_total**: the number of records from all users in each neutral score.
- **interact_neutral_super**: the number of records from super users in each neutral score.

Selecting the 3 most active users as super users is suitable. For comparsion, the color and size of interaction from normal users is blue and 1, while from super users it is red and 2:

```
author_super3 = author.nlargest(3, "interact_out")
```

```
color=author_interact.apply(red if in author_super3 else blue)
```

```
size=author_interact.apply(2 if in author_super3 else 1)
```

To count the number or records in different scores, grouping by two digits is sufficient. For example, in compound:

```
interact_compound_total = author_interact.groupby(compound).round(2).size()
```

```
interact_compound_super = interact_compound_total.isin(author_super3)
```

5.2.2 Interaction distribution

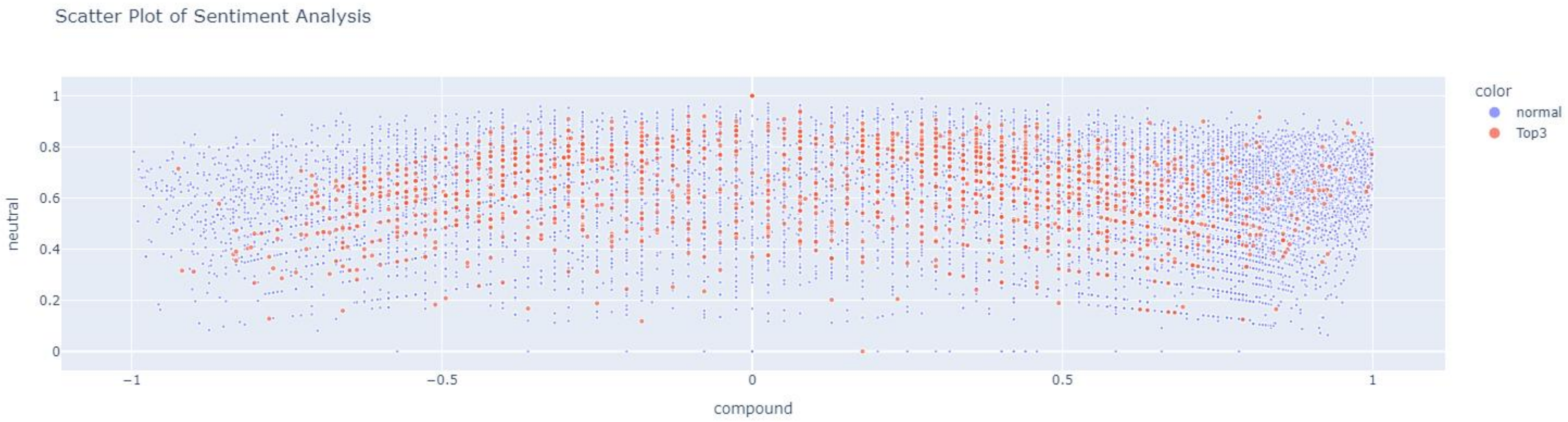


Figure 10 interaction emotion distribution

As each point is an interaction, this diagram shows that the density on the two sides and the top part is greater than the middle and bottom part, suggesting that the interactions are more likely to be discussed in an objective and positive manner.

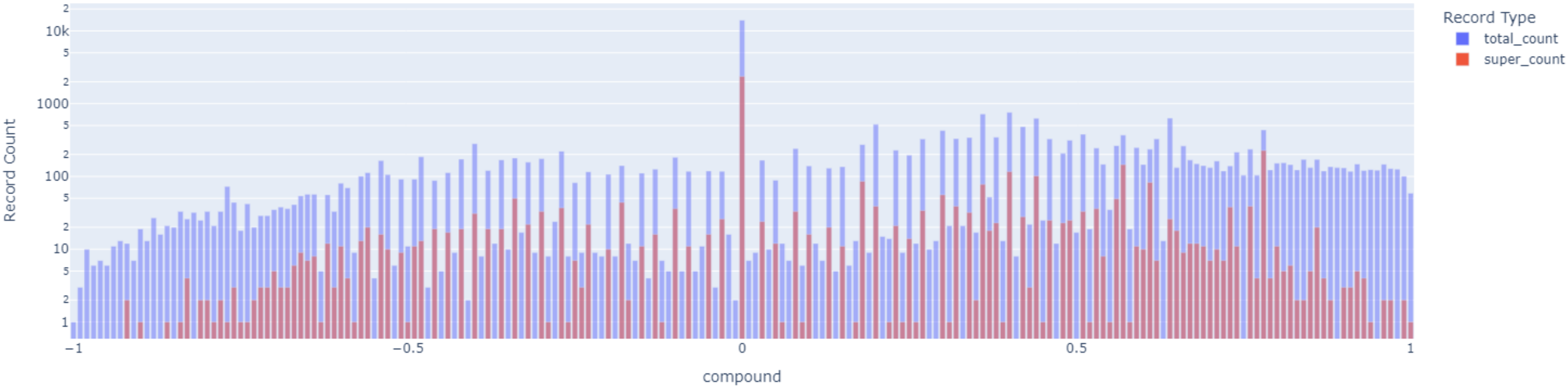


Figure 11 compound scores distribution

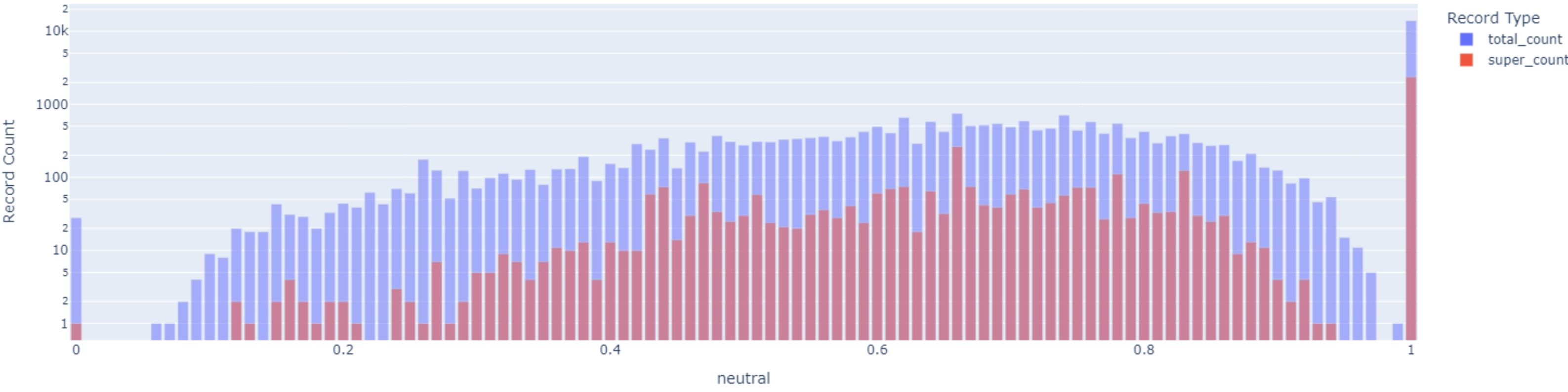


Figure 12 neutral scores distribution

These two charts provide a distribution of sentiment score, displaying the number of interactions in each sentiment scores. Interactions tend to have a high positive compound or a high negative compound instead of a low compound. However, the score of 0 has the greatest number of records.

The interaction also tends to be neutral, especially the scores 1 has the greatest number of records.

5.2.3 Conclusion

Interactions tend to be objective and positive, but interaction also more likely to be negative instead of middle emotion.

5.3 User distribution

5.3.1 Metrics and measures

Similar to the distribution of interaction, study the distribution of user needs metrics:

- **Compound**: an attribute has already been stored in author, indicate the mean compound of this user according to his all interactions.
- **Netural**: an attribute has already been stored in author, indicate the mean neutral of this user according to his all interactions.
- **user_compound_total**: the number of users in each compound score.
- **user_compound_super**: the number of super users in each compound score.
- **user_neutral_total**: the number of users in each neutral score.
- **user_neutral_super**: the number of super users in each neutral score.

Similar measure with interaction distribution, but instead of using author_interact, to study users, the data should be obtained from author:

```
author_super10 = author.nlargest(10, "interact_out")
```

```
color=author.apply(red if in author_super3 else blue)
```

```
size=author.apply(2 if in author_super3 else 1)
```

To count the number of users in different scores, grouping by two digits is sufficient. For example, in compound:

```
user_compound_total = author.groupby(compound).round(2).size()
```

```
user_compound_super = user_compound_total.isin(author_super10)
```

5.3.2 User distribution

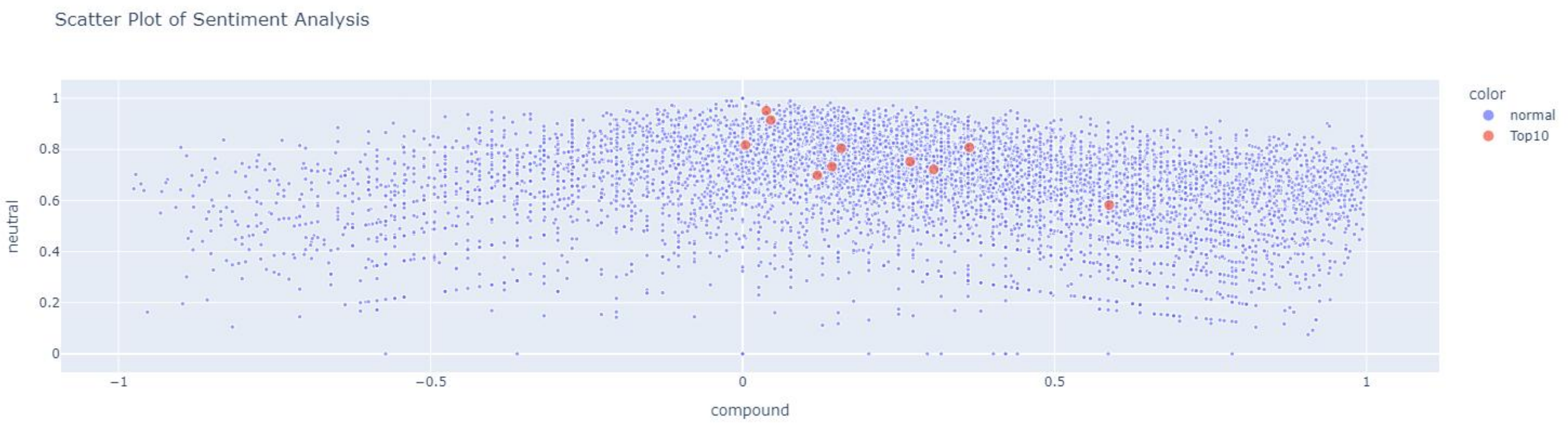


Figure 13 user sentiment distribution

Each point is a user, and there is an obvious difference of density in this diagram, the right side representing the positive sentiment and the top side representing the neutral manner with a higher density, which indicates that users tend to be positive and neutral. Additionally, the 10 most active users are all on the positive side.

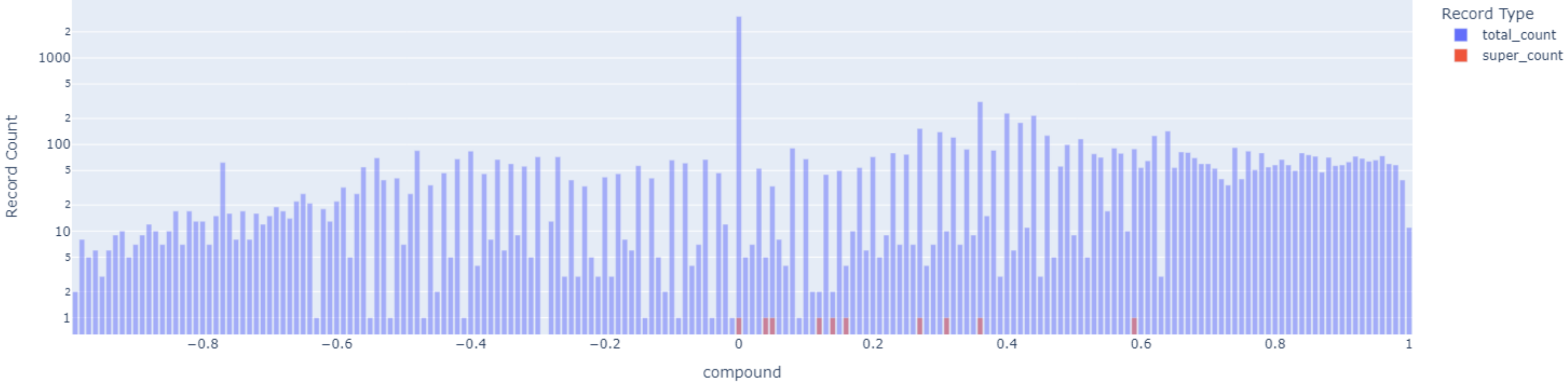


Figure 14 user compound distribution

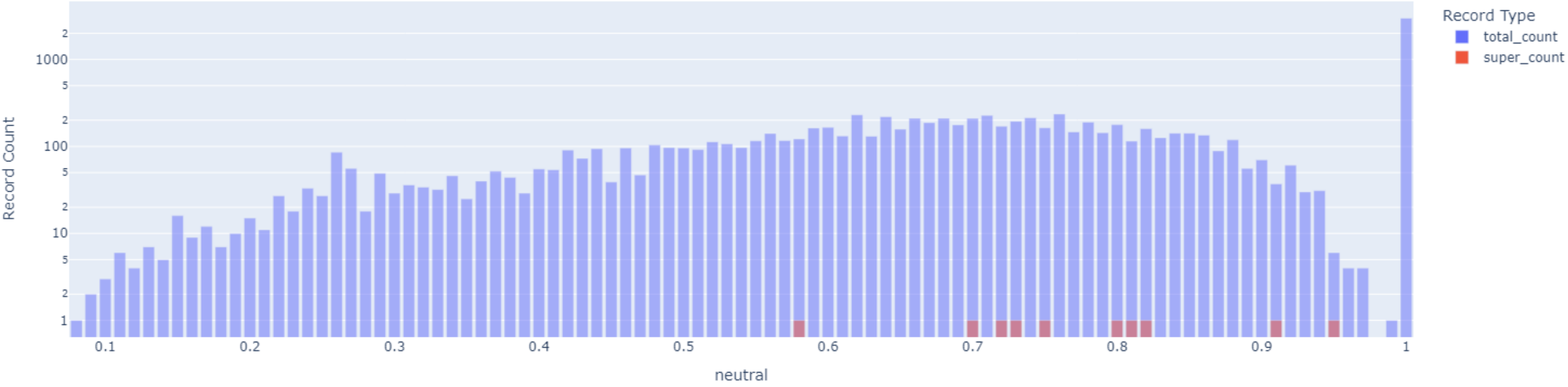


Figure 15 user neutral distribution

According to the diagram, the distribution of users is similar to the distribution of interaction, except for the super users, who have higher positive mean scores.

5.3.3 Conclusion

Users generally behave objectively and positively. Super users have a higher positive level.