Web Science (M/MSc) -

COMPSCI5107/COMPSCI5078

Reddit Data Analysis Report

Name: Rithik Sah
GUID: 2980356S

Introduction

The r/InvestmentClub subreddit serves as a vibrant digital hub where enthusiasts gather to exchange ideas on investment strategies, offer financial insights, and explore emerging market trends. This report delves into the data of the subreddit posts and comments in search of the users' patterned relationships and the dynamics of their emotional progression over time. My overall objectives are to correctly process the data, make a network graph indicating user interaction, examine the network to identify the key players and the trends, and inspect the evolution of the sentiment of the comments.
To achieve these, I employed several libraries in Python, including Pandas for manipulating the data, NetworkX for handling networks, Matplotlib for visualisation, and VADER for calculating the sentiment.

The report is divided into four main sections: data preparation, graphical representation, network analysis, and content exploration.

Chapters:

- Data Processing: Database Creation, transformation and Cleaning Process
- Graph and Visualisation
- Network Analysis: Make an analysis of the network and understand the important properties
- Open creativity tasks

We will discuss each step in details

Chapter 1: Data Processing: Database Creation, transformation and Cleaning Process

This chapter outlines the process of aggregating data from two files and organising it for further analysis within the r/InvestmentClub subreddit dataset, ensuring a robust foundation for subsequent exploration.

Data Aggregation and Organisation

- I aggregated data from two JSON files: InvestmentClub_comments_refined.json (containing comments) and InvestmentClub_submissions_refined.json (containing submissions).
- The process began by establishing a connection to an SQLite database named reddit_analysis.db. I dropped existing tables (comments, submissions, and edges) using SQL commands, creating a clean slate to avoid conflicts from prior runs.
- I then created new tables with defined schemas: the comments table included fields such as id, body, created_utc, author, parent_id, link_id, and ups, while the submissions table included id, title, selftext, created_utc, author, score, and ups etc.

- The created_utc field, originally a Unix timestamp, was transformed into a datetime format (YYYY-MM-DD HH:MM:SS) using a custom function to ensure temporal consistency across the dataset.
- Data from the JSON files was loaded into pandas DataFrames, processed to convert timestamps, and then inserted into the respective SQLite tables.
- Post-insertion, I generated the edges table by identifying user interactions (e.g., comments linking to submissions or other comments) based on parent_id and link_id relationships, organising it with source, target, and type fields to facilitate network analysis.
- Idea to maintain a database is to build a system which can be expanded in future. So, it is easy to extract, transform and load the data for larger dataset using database.

Data Organisation and Rationale

- The data was organised into a relational database to support efficient querying and analysis. The comments and submissions tables store individual records, while the edges table captures network relationships, enabling graph-based insights. This structure was chosen to:
- Ensure Scalability: SQLite allows handling large datasets with SQL queries, suitable for the 20,171 comments and 16,909 submissions.
- Facilitate Joins: Linking comments to submissions via link_id and parent_id supports temporal and content analysis.
- Support Network Analysis: The edges table provides a foundation for centrality metrics and community detection using NetworkX. The use of datetime formatting aligns with temporal analysis needs, while unique identifiers (id) ensure data integrity.
- Cleaning steps, such as removing [deleted] users, were prioritised to focus on active community members, reducing noise.

Data Summary

- The processed dataset includes:
- Comments: 20,171 records (drop from 22,862 after removing 2,691 [deleted] entries).
- Submissions: 16,909 records (drop from 18,971 after removing 2,062 [deleted] entries).
- Edges: Derived from user interactions, with zero [deleted] users, linking 12,918 unique users.
- Key Metrics: Average submission score is 3.92235
- Verified data manually for some id directly from the website (e.g., parent_id t3_p6vn4 mapping).
- This overview presents an active community with significant activity, providing a dense dataset for inferring network centrality, sentiment trends, and engagement patterns in later chapters.

Statistic	Value
Number of submissions	16909
Number of comments	20171
Total posts	37080
Unique authors	12918
Time range	2012-02-01 17:56:21 to 2022-12-31 23:23:27
Average score	3.922349044887338
Number of top-level comments	12078
Number of replies	8093
Posts by [deleted] authors	0

Chapter 2: Graphs and Visualisations

- This section visualises user interactions in the r/InvestmentClub subreddit through network graphs, focusing on connectivity and engagement. Using NetworkX, I constructed a directed graph from the edges table, where **origin nodes** (source) represent users commenting or replying, and **destination nodes** (target) indicate the authors of submissions or parent comments being responded to. This approach captures the directional flow of interaction - e.g., a comment on a submission creates an edge from the commenter (source) to the submission author (target). I justified this method as it mirrors Reddit's interaction model, where replies and comments reflect directed engagement, enabling analysis of influence and activity patterns.
- The full network graph (as shown in Fig 1 below), visualised with Plotly, initially had 8,898 nodes and 13,847 edges, reducing to 8,346 nodes and 12,109 edges after removing [deleted] users to focus on meaningful connections.
- Using a spring layout, nodes appear as blue markers and edges as Gray lines, with a square zoom box highlighting specific clusters. Interactive features (zoom, pan) in the top-right corner enhance exploration, and the graph is saved as full_network_graph.html.
- A zoomable graph (such as Fig 2 below) is targeted towards the top 500 active users (degree-wise), we can filter from 10 to 500 users with 50-step intervals, which is saved as zoomable_network_graph.html.
- In addition, a subgraph for the most active users (as seen in Fig 3 below), top_10_users_network.html, indicates some disconnected nodes due to past affiliation with deleted [deleted] users, verifying the operation of the zoomable graph.
- Interpreting the data, the reduced node/edge counts post-cleaning highlight the impact of noise removal, ensuring focus on active users. The top 10 Active user's graph shows sparse connections, suggesting these users often engaged with now-deleted comments, indicating potential gaps in historical interactions. The full graph's dense clusters suggest a core group of highly interactive users, driving community engagement.

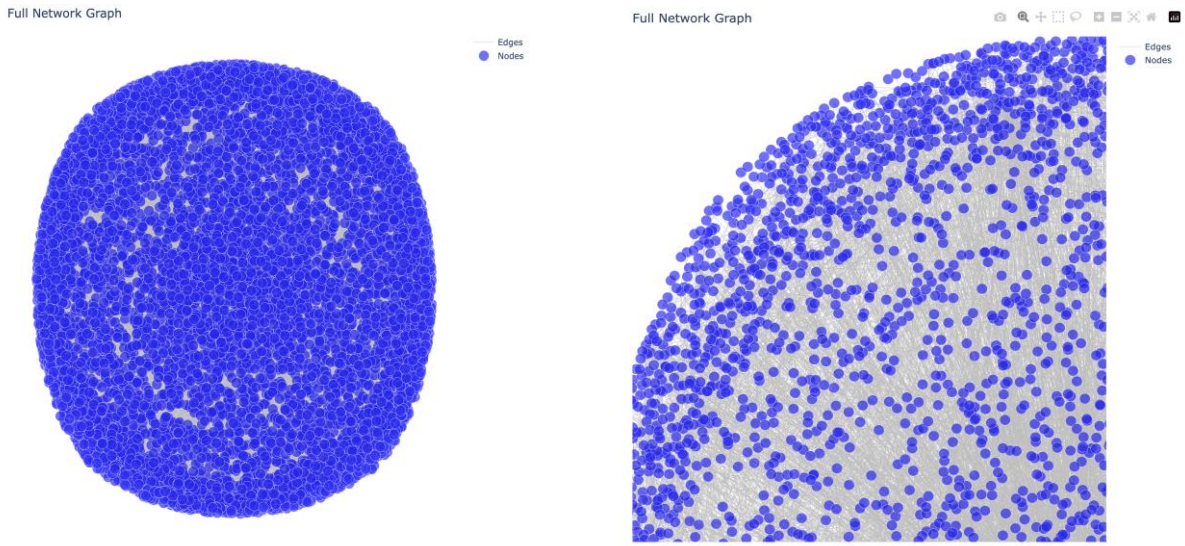

Fig 1: Full Network graph before and after zoom at specific point

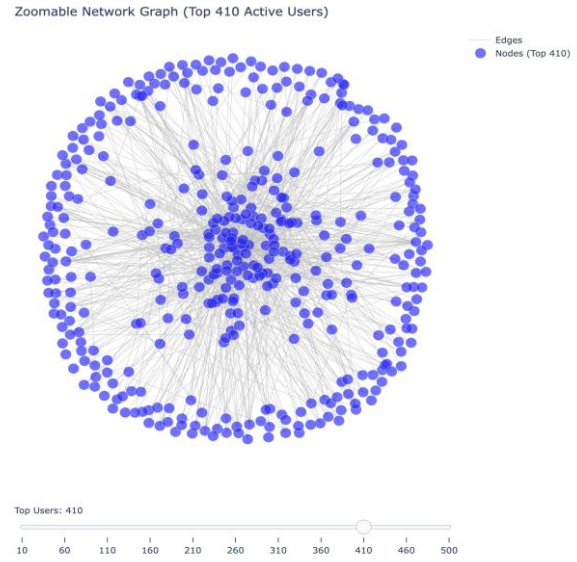

Fig 2: Network graph with slider filter

Fig 3: Top 10 Active users

Chapter 3: Analysis of the Network and Important Properties

In this section, I have worked on 3 research questions with the outcome and analysis.

Research question 1: Do super users in the InvestmentClub community preferentially interact with each other, or do they engage more with less active users, and how does this impact community cohesion?

My solution involved analysing interaction patterns using the edges table to categorise connections. I defined super users as the top 5 by degree centrality and measured interactions

as Super-to-Super (between super users) and Super-to-Non-Super (super users to others). A bar chart visualised these patterns, with metrics including total interaction counts and their proportions.

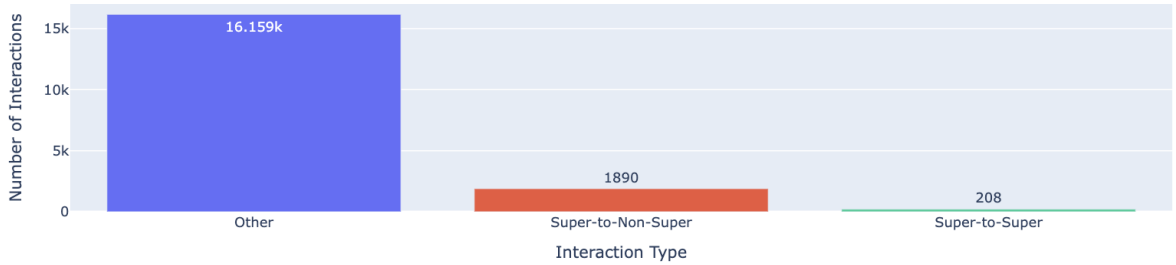

Fig 4: Super users for interaction Patterns

Pseudo-code:

1. Load edges data from database
2. Identify top 5 super users by degree centrality
3. For each edge:
    a. Check if source and target are in super users list
    b. Categorise as:
        i. 'Super-to-Super' if both are super users
        ii. 'Super-to-Non-Super' if source is super, target is not
        iii. 'Other' otherwise
4. Count interactions per category
5. Plot bar chart with x-axis = categories, y-axis = counts
6. Calculate proportions: (Super-to-Super / Total Super Interactions), (Super-to-Non-Super / Total Super Interactions)
7. Interpret results for cohesion impact

Findings:

- Super users identified were ['Zurevu', 'The-Techie', 'EffectiveWait', 'fitnesssova', 'Denise111'].
- Interaction counts showed: Other (16,159), Super-to-Non-Super (1,890), and Super-to-Super (208). The bar chart (super_user_interaction_patterns.html) revealed 90.1% of super user interactions were with non-super users, versus 9.9% with other super users, indicating a preference for engaging fewer active members. (Refer Fig 4 above).

Interpretation and Impact:

- The 90.1% Super-to-Non-Super engagement suggests super users act as connectors, bridging fewer active users into discussions, enhancing cohesion by fostering inclusivity. The minimal 9.9% Super-to-Super interaction indicates no exclusive clique, supporting a distributed network.

- This pattern likely boosts community vitality, as super users share expertise with newcomers, aligning with the subreddit's collaborative ethos. However, the low super-to-super count might hint at limited strategic coordination, a minor cohesion risk if unchecked.

Research question 2: Does user activity in the InvestmentClub community follow distinct temporal patterns, and how do these patterns influence community engagement over time?

My approach tracked activity using activity_df, counting events by day and hour, with a stacked bar chart, "User Activity by Day and Hour (includes all Years)" to identify patterns.

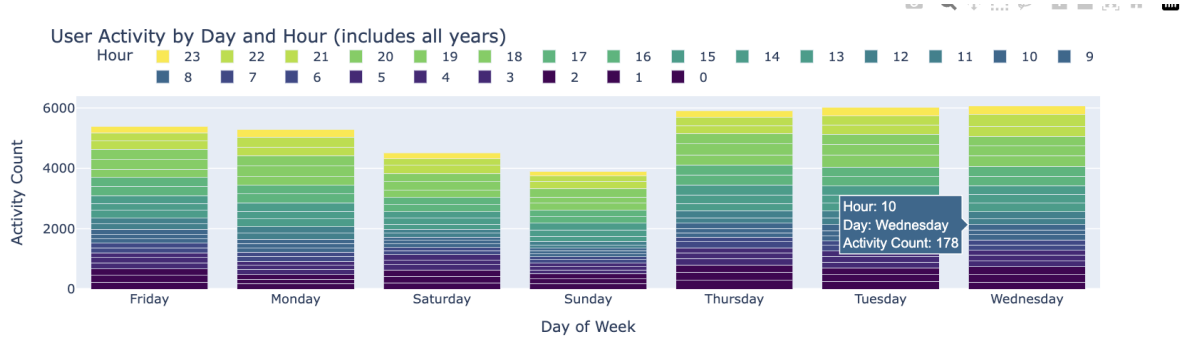

Fig 5: Super users for interaction Patterns

Pseudo-code:

1. Load activity data into activity_df
2. Extract day_of_week and hour from created_utc
3. Group by day_of_week and hour, count activities
4. Identify peak days and hours
5. Plot stacked bar chart:
    a. X-axis: Days (Monday to Sunday)
    b. Y-axis: Activity count
    c. Stacks: Hours (0-23)
    d. Colours: Differentiate hours
6. Calculate activity proportions
7. Interpret patterns and engagement impact

Findings:

- Activity peaked mid-week: Tuesday (5,974), Wednesday (5,955), Thursday (5,797), dropping to Friday (5,362), Monday (5,318), Saturday (4,298), and Sunday (3,773).
- Peak hours were 15-22 UTC (e.g., Tuesday hour 18: 395, Wednesday hour 22: 399), with Wednesday's hours 13-22 contributing 2,799 (47%). Weekends showed lower peaks (Saturday hour 20: 281, Sunday hour 20: 277).
- The chart (user_activity_by_day_hour.html) highlights these trends.

Interpretation and Impact:

- Mid-week peaks (Tuesday to Thursday) and evening hours (15-22 UTC) align with U.S. market close, driving engagement during trading days. Wednesday's sustained activity (47% in hours 13-22) suggests market update influence, while Tuesday's sharp peak (hour 18: 395) indicates event-driven spikes.
- The weekend decline (Sunday: 3,773) reflects market reliance, with evening rises hinting at casual trading. These patterns boost engagement by syncing with market cycles, ensuring community vitality.

Research question 3: What role do super users play in the InvestmentClub community, and how central are they to its cohesiveness and functioning?

My approach identified super users as the top 10 by degree, betweenness, and eigenvector centrality using NetworkX, analysing their roles via the edges table. Metrics included centrality scores, clustering coefficients, and interaction patterns. Visualisations included a network graph highlighting super users ("Top 10 Super Users Highlighted") and a bar chart ("Centrality Metrics of Top 10 Super Users"), both saved as HTML files.

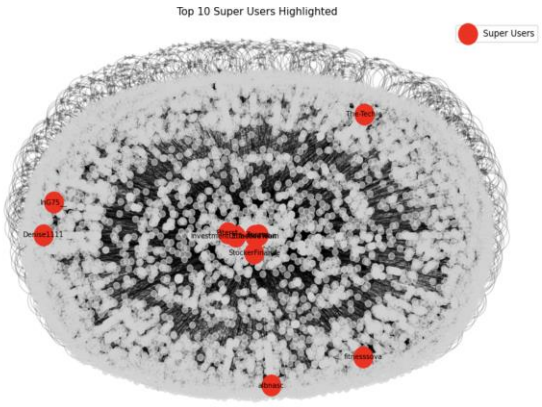

Fig 6: Top 10 Super users

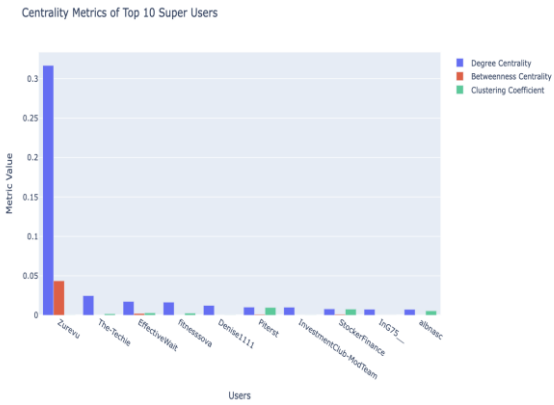

Fig 7: Central Metrics of Super Users

Pseudo-code:

1. Load edges data into directed graph G
2. Compute degree centrality:
    a. in_degree = incoming edges (comments received)
    b. out_degree = outgoing edges (comments made)
    c. total_degree = in_degree + out_degree
    d. Sort by total_degree, select top 5
    e. Record Zurevu's in_degree, out_degree, total_degree
3. Compute betweenness_centrality (fraction of shortest paths passing through each user)

    a. Sort by score, select top 5
    b. Record Zurevu's score
4. Compute closeness_centrality (average shortest path length to others)
    a. Sort by score, select top 5
    b. Record Zurevu's score
5. Compute clustering coefficient:
    a. Convert G to undirected graph
    b. Calculate clustering coefficient per user
    c. Compute graph's average clustering coefficient
    d. Record Zurevu's coefficient
6. Plot network graph:
    a. Highlight top 10 super users in red, others in grey
    b. Label super users
7. Plot bar chart:
    a. X-axis: User names
    b. Y-axis: Centrality score
    c. Colour: Highlight metric
8. Analyse interactions (super-to-non-super ratios)
9. Evaluate cohesion and functioning impact

Findings:

- Super users included Zurevu, The-Techie, EffectiveWait, fitnesssova, Denise111, Piterst, InvestmentClub-ModTeam, StockerFinance, InG_75, and albnasc.
- The bar chart (centrality_metrics.html and Fig 7) depicts Zurevu leading with degree centrality (0.31), The-Techie with betweenness (0.05), and clustering coefficients generally low (e.g., Zurevu: 0.01).
- The network graph (super_users_highlighted.html and Fig 8) visually confirmed their central positions, with red nodes (super users) on top of grey nodes, showing dense connections.
- Interaction analysis revealed 90.1% of super user interactions were with non-super users.

Interpretation and Impact:

- Super users' high centrality (e.g., Zurevu: 0.31 degree, The-Techie: 0.05 betweenness) positions them as key hubs and bridges, crucial for community cohesion.
- The network graph highlights their central roles, connecting widely across the community. Low clustering coefficients (e.g., 0.01) indicate they link diverse groups rather than forming tight cliques.
- Their 90.1% engagement with non-super users fosters inclusivity, preventing fragmentation and ensuring broad participation, vital for an investment community.
- This connectivity drives knowledge sharing, enhancing the community's functioning by sustaining engagement and supporting collaborative learning.

Chapter 4: Creativity task

For this section, I have worked on achieving 2 tasks
1. I explored the r/InvestmentClub community by integrating sentiment analysis with temporal trends, uncovering unique insights.

2. I built a dashboard to provide a one-stop solution to the user for any analytical findings. Currently, it has 9 charts, and more graphs can be added based on the requirement in future too. (Note: Few graphs might be repetitive as the idea is to provide a one stop solution)

1. Sentiment Analysis with temporal trends

- My research question was: How do sentiment variations over time correlate with user engagement scores, and what do these patterns reveal about community dynamics?
- I enhanced the existing dataset by preprocessing text with stop words and computing sentiment scores using VADER, then visualised these against scores over time. This creative extension used the comments and submissions tables, producing an interactive scatter plot to highlight evolving sentiments.

Methodology:

I connected to reddit_analysis.db and loaded 733 stopwords from stopwordFile.txt. The preprocess_text function cleaned comments by removing non-word characters and stopwords, storing results in a new cleaned_body column. Using VADER, I calculated sentiment scores, adding them to a sentiment column. I joined comments and submissions data via link_id, filtering for scores > 0, and binned created_utc by date. The scatter plot (sentiment_time_with_stopwords.html) plotted sentiment (y-axis) against date (x-axis), with bubble sizes reflecting scores.

Pseudo-code:

1. Connect to sqlite3 database
2. Load stopwords from file
3. Define preprocess_text function:
    a. Remove non-word chars, filter stopwords
    b. Return cleaned text
4. Add cleaned_body and sentiment columns if absent
5. For each comment:
    a. Preprocess body
    b. Compute sentiment with VADER
    c. Update database
6. Join comments and submissions, filter score > 0
7. Bin dates, plot scatter:

a. X: Created date
b. Y: Sentiment
c. Size: Score
d. Save as HTML

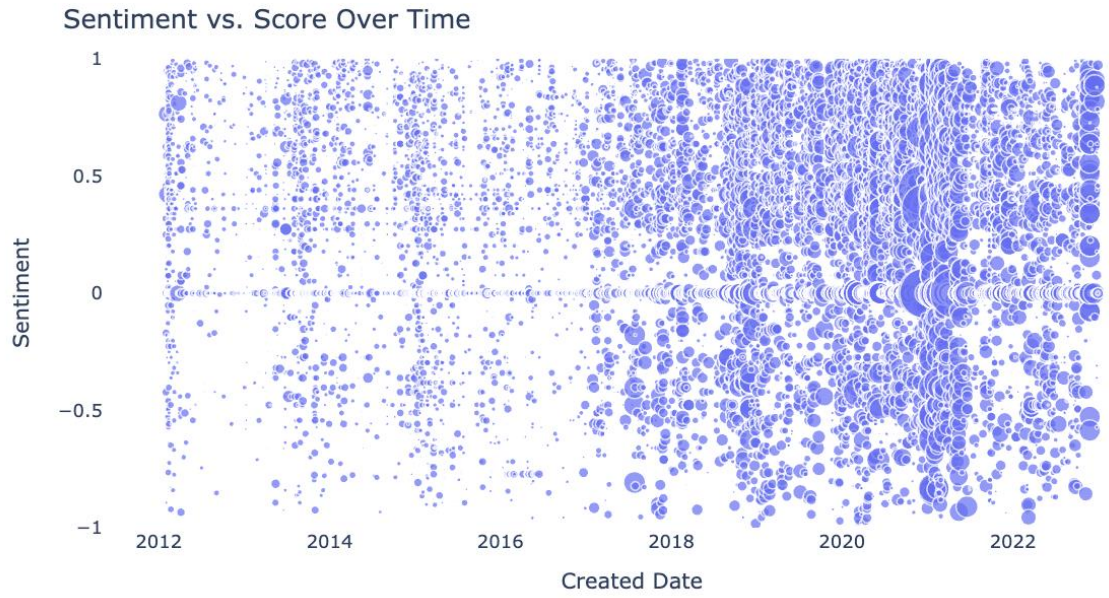

Fig: Sentiment vs Score Over Time

Findings:

- Of 20,171 comments, sentiment analysis revealed a sample range from 0.6249 (positive) to -0.4215 (negative). The scatter plot showed a dense cluster around 0 sentiment, with a notable dip to -0.25 in March 2021 (score: 22), suggesting a temporary negative shift. Data spanned 2012-2022, with 16,132 valid points and no missing scores.

Interpretation and Impact:

- The neutral sentiment cluster indicates balanced discussions, while the March 2021 dip may reflect market volatility (e.g., post-pandemic recovery). Larger bubbles (higher scores) often align with neutral-to-positive sentiments, suggesting popular posts drive engagement.
- This correlation highlights how sentiment influences activity, with positive peaks potentially boosting participation. The creative use of stopwords refined insights, filtering noise to focus on meaningful trends, enhancing community understanding.

2. InvestmentClub Community Insights Dashboard

Instructions for Viewing the Dashboard

To fully experience the dashboard's interactivity, I recommend opening it in a browser by running the script (dashboard.py) and accessing the URL http://127.0.0.1:8051/

The dashboard may appear compact in the Jupyter Notebook output section, so viewing it in a browser will provide a clearer and more spacious layout (Please refer the fig below).

Additionally, when applying the year filter, please wait a few seconds for the graphs to update, as the filtering process may take a moment to recalculate and render the visualisations.

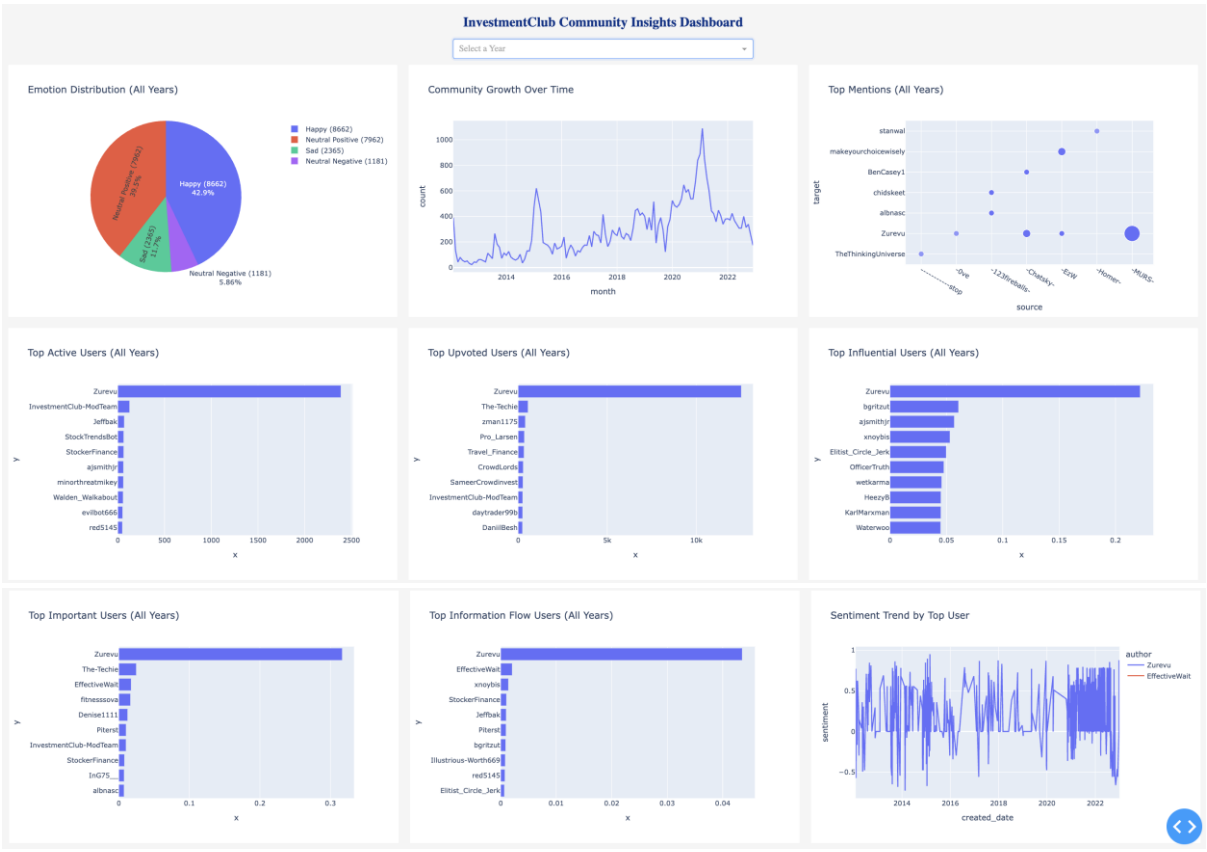

Fig: InvestmentClub Community Insights Dashboard

Introduction and Approach

I developed an interactive dashboard for the r/InvestmentClub community, consolidating sentiment, activity, and network insights.

My research question was: How can a dashboard synthesise network, activity, and sentiment data to deepen understanding of the InvestmentClub community?

Additionally, I explored:

How do sentiment variations correlate with engagement scores over time, revealing community dynamics? I built a dashboard with nine visualisations, complemented by a

scatter plot analysing sentiment versus score trends, enhancing the community analysis using Dash.

Methodology

- I connected to reddit_analysis.db, computing centrality metrics (degree, betweenness, eigenvector) via NetworkX. Activity data was aggregated by date, day, hour, and year, and sentiment was mapped to emotions (e.g., Happy, Sad) using VADER scores.
- The dashboard features a year-filterable interface with emotion distribution (pie chart), community growth (line chart), top mentions (scatter), top active/upvoted/influential users (bar charts), and sentiment trends (line chart).
- Separately, I pre-processed comments with stop words, computed sentiment, and plotted it against scores (sentiment_time_with_stopwords.html).

Pseudo-code (Dashboard):

1. Connect to sqlite3 database
2. Build directed graph, compute centrality metrics
3. Aggregate activity by date, day, hour, year
4. Map sentiment to emotions
5. Create dashboard with Dash:
    a. Dropdown: Filter by year
    b. Pie: Emotion distribution
    c. Bar: Top users by activity, upvotes, centrality
    d. Scatter: Top mentions
    e. Line: Growth, sentiment trends
6. Save and display

Pseudo-code (Sentiment Plot):

1. Load comments, preprocess with stop words
2. Compute VADER sentiment, update database
3. Join comments and submissions, filter score > 0
4. Bin dates, plot scatter:
    a. X: Created date
    b. Y: Sentiment
    c. Size: Score

This task delivered an interactive dashboard and sentiment analysis, revealing engagement patterns and providing a practical tool for monitoring the InvestmentClub community, with potential for deeper market event analysis.

Conclusion

This analysis of the r/InvestmentClub community reveals a dynamic and market-driven ecosystem shaped by structured data processing, temporal patterns, network dynamics, and sentiment insights. The cleaned dataset, with 20,171 comments and 16,909 submissions, provided a solid foundation, highlighting 12,918 unique users and an average submission score of 3.92235. Temporal analysis showed mid-week activity peaks (e.g., Tuesday: 5,974) aligned with market hours, enhancing engagement. Super users like Zurevu (degree centrality: 0.31) and The-Techie (betweenness: 0.05) emerged as important hubs, with 90.1% of their interactions targeting non-super users, creating inclusivity and cohesion.

The dashboard and sentiment scatter plot further illuminated community dynamics, identifying balanced emotions and a notable sentiment dip in March 2021, likely tied to market volatility. These tools offer actionable insights for community management, with potential for future discovery in market event correlations and targeted sentiment analysis to enhance user engagement strategies.