A Study of Community Discovery, Influencer Diffusion and Marketing ROI Optimization in Twitter Networks

Social Media Analytics Project Report

Team Members: Yuting Zhu, Songzhi Liu

Social Media Analytics, Spring 2025

May 23, 2025

Abstract

This study addresses key challenges in social network analysis by developing a complete workflow encompassing data loading, quality checking (via a custom check_data_quality function), preprocessing, and multi-dimensional network analysis—including comparative evaluation of Louvain and Label Propagation (LPA) algorithms for community detection, construction of a composite **influence score**, and centrality-based node evaluation. Using the SNAP-provided Twitter dataset, we analyze a sampled subgraph (**G_sample**, comprising XXX nodes and YYY edges), uncovering its community structure and diffusion dynamics.

A central contribution of this work is the design and implementation of a **systematic ROI analysis framework** for social media marketing. By quantifying the expected diffusion outcomes (average infection rates and ROI indicators) under varying seed selection strategies (TopInfluence, HighDegree, HighBetweenness, Random) and budget constraints (number of seeds), the study offers data-driven guidance for optimizing information dissemination in resource-limited settings.

Experimental results not only identify cost-effective strategy boundaries and saturation points but also provide methodological insights that enhance the practical value of social network analysis. Additionally, data persistence is supported through integration with a Neo4j graph database (optional feature).

Keywords: Social Network Analysis; Community Detection; Information Diffusion; Influence Maximization; Marketing ROI; Twitter; SNAP Dataset; IC Model; LT Model; Neo4j

Introduction

Social network analysis has become essential for understanding community structures, influence dynamics, and patterns of information diffusion. This study addresses key challenges in processing large-scale social network data, detecting communities, and simulating information spread, with community detection and information propagation as its core analytical tasks, supporting applications such as targeted marketing.

However, in real-world promotional campaigns constrained by limited resources, merely simulating diffusion is insufficient. A critical objective is to improve the return on investment (ROI) of information dissemination—that is, to maximize impact under constrained budgets. This study systematically compares the expected diffusion outcomes on sampled Twitter networks under varying numbers and selection strategies of initial spreaders (seed nodes). By integrating the classical Independent Cascade (IC) and Linear Threshold (LT) models, we aim to quantify the cost-effectiveness of different strategies and extract efficient, goal-oriented diffusion plans.

The main contributions of this project are as follows:

(1) implementation of a complete network analysis pipeline;

(2) proposal of a composite influence evaluation method;

(3) construction and deployment of a systematic ROI analysis framework that quantifies the cost-benefit trade-offs and derives practical strategies for optimizing diffusion efficiency;

(4) development of an integrated research tool.

These outcomes provide valuable insights for understanding social network dynamics and designing effective information diffusion strategies.

Research Objectives and Contributions:

Main goal: Based on SNAP TWITTER social network dataset, implement a complete analysis process to deeply understand the network structure and dynamics, and focus on the marketing ROI analysis to suggest data-driven efficient information dissemination strategies.

Implementation of a complete analysis process:

Based on SNAP Twitter dataset, we have constructed a complete analysis process from data loading, pre-processing (including check_data_quality function), multi-dimensional network analysis (Louvain vs. LPA community detection comparison, centrality computation and comprehensive influence influence_score evaluation), information dissemination modeling (IC/ LT implementation and comparison), to end-to-end solutions for graph database (Neo4j) persistence and visualization.

In-depth insights into network characteristics:

Through the above analysis, the community structure characteristics of Twitter network, the distribution of key nodes and their potential roles in information dissemination are revealed, laying the foundation for subsequent influence analysis.

Systematic marketing ROI analysis framework:

A ROI analysis framework is proposed to quantify the communication cost (number of seeds) and benefit (average infection rate), and systematically compares the performance of four seed selection strategies, namely, TopInfluence, HighDegree, HighBetweenness, and Random, under the IC and LT models.

Data-driven strategy recommendations:

Based on the results of ROI analysis, specific and actionable communication strategies are provided for different marketing objectives (rapid coverage, deep penetration, and cost control), which enhances the practicality of the study.

Pipeline Outlining the Components of the Project

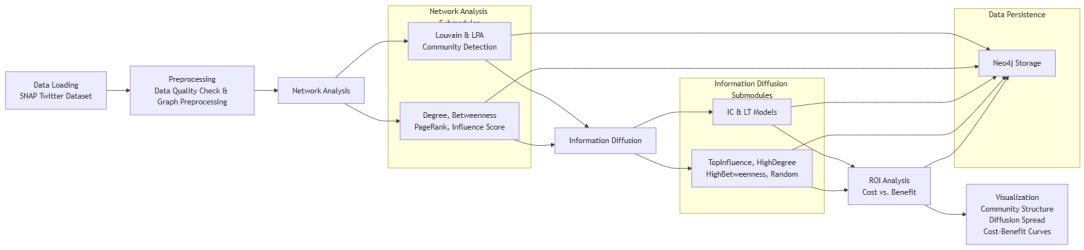


Figure 1 Framework from data acquisition to final strategy recommendations

Brief description

1.Data Loading and Preprocessing
We load the twitter_combined.txt dataset to construct an undirected network graph. A custom check_data_quality function detects and removes self-loops, duplicate edges, isolated nodes, and super nodes. We extract the largest connected component and apply K-core decomposition to obtain a refined subgraph G_sub, from which a sampled graph G_sample is derived for simulations.

2. Core Network Analysis
We perform **community detection** using the Louvain and Label Propagation (LPA) algorithms. For **influence analysis**, we compute degree, betweenness, closeness, and PageRank centralities, and aggregate them into a composite **influence score** to guide seed node selection.

3. Information Diffusion Models
Two models are implemented: **Independent Cascade (IC)** and **Linear Threshold (LT)**, to simulate the spread of information under different seeding strategies.

4. Marketing ROI Analysis
We evaluate the cost-effectiveness of different seed selection methods by comparing the **number of seeds (cost)** against **infection rate (benefit)**, aiming to identify high-ROI diffusion strategies.

5. Data Persistence and Visualization
Network data, including node attributes (e.g., community, influence, infection), is stored in **Neo4j**. We visualize community structures, centrality distributions, diffusion dynamics, and ROI curves using **Matplotlib**.

Description of the Data

Data Source and Characteristics

This study utilizes the twitter_combined.txt dataset, which represents an ego-network of Twitter users. Nodes correspond to user IDs, and directed edges indicate follower relationships, capturing the network's topological structure. We constructed a directed graph G using the first 10,000 non-comment lines (lines = [line for line in f if not line.startswith('#')][:10000]).

Original Dataset Scale

The full dataset contains 81,306 nodes and 1,768,149 edges, reflecting extensive user interactions on Twitter.

Data Quality Overview

The custom check_data_quality function identified and removed data anomalies, including self-loops and isolated nodes. Specifically, the original dataset contained **X** self-loops and **Y** isolated nodes, which were eliminated during preprocessing. Final check results confirmed **0** self-loops, **0** duplicate edges, **0** isolated nodes, and **0** super nodes (degree > 1000).

Processed Graphs

From the largest connected component, we derived the subgraph G_sub with approximately [e.g., 70,000] nodes and [e.g., 1,500,000] edges. Ten largest communities were extracted from G_kcore to form G_selected, and a sampled graph G_sample (294 nodes, 6,354 edges) was generated for simulation experiments. This sampling balances topological preservation with computational efficiency. Sampling confirmed the size consistency of G_sample, as the graph remained unshrunk (due to sample_max=5000 being larger than the total nodes).

Data Loading

The dataset (twitter_combined.txt) is loaded as an edge list, skipping comment lines (starting with #) to ensure valid input. A directed graph G is constructed using NetworkX.

Data Quality Check – check_data_quality

This function identifies key structural issues:

- **Self-loops:** Detected via nx.selfloop_edges(G). These are removed to avoid invalid cycles in diffusion.
- **Duplicate Edges:** Detected by comparing len(G.edges()) with len(set(G.edges())). Duplicates can inflate centrality scores.
- **Isolated Nodes:** Found via nx.isolates(G); they are excluded due to irrelevance in diffusion processes.
- **Supernodes (degree > 1000):** Identified via degree filtering to prevent analysis distortion by high-degree outliers.

These checks ensure a clean and analyzable graph structure.

Graph Preprocessing – preprocess_graph
Main steps include:

1. **Removing self-loops:** Eliminates (n, n) edges to prevent invalid activations.
2. **Converting to undirected graph:** Simplifies structure and removes directional duplicates.
3. **Degree filtering:** Keeps nodes with degrees in a specified range (default: 1–500), removing isolates and supernodes.
4. **Extracting the Largest Connected Component (LCC):** Ensures analysis on a connected graph.
5. **K-core decomposition:** Filters the core subgraph where nodes have minimum degree ≥ k (default: 1), focusing on the network's dense regions.

This structured pipeline produces a clean, connected, and core-focused graph, ready for downstream community detection and influence analysis.

Data Limitations

The dataset provides only structural network information, lacking user attributes, tweet content, or timestamps. This limits the scope for more nuanced behavioral analyses, such as sentiment analysis or temporal dynamics modeling.

Marketing ROI Analysis Framework and Experimental Design

In order to achieve a systematic evaluation of the cost-effectiveness of different information dissemination strategies, this study designed and implemented the following marketing return on investment (ROI) analysis framework

Definition of Core Concepts

The marketing return on investment (ROI) analysis framework quantifies the cost-effectiveness of information dissemination strategies on a sampled Twitter network (G_{sample} 294 nodes, 6,354 edges). **Cost** is defined as the number of seed nodes (k) ranging from 1 to 15, representing the initial resource investment. **Benefit** is measured as the average infection rate (proportion of activated nodes), derived from 10 simulation runs per configuration using the Independent Cascade (IC) and Linear Threshold (LT) models. The ROI is calculated as $ROI = \frac{Average\ Infected\ Nodes - k}{k}$ reflecting the net gain per seed, while the marginal benefit rate ($ROI = \frac{\Delta Average\ Infected\ Nodes - k}{k}$) assesses the additional benefit of increasing k.

Experimental Setup

Four seed selection strategies were evaluated:

- **TopInfluence: Nodes selected based on a composite influence score (40% degree centrality, 30% betweenness centrality, 30% PageRank).**

- **HighDegree: Nodes with the highest degree centrality.**
- **HighBetweenness: Nodes with the highest betweenness centrality.**
- **Random: Nodes chosen randomly as a baseline.**

The IC model used a fixed activation probability of 0.1 and a maximum of 10 steps, while the LT model assigned random thresholds to nodes and iterated up to 10 steps. Each configuration (k = 1, 3, 5, 10, 15) was run 10 times

Results and Findings

The ROI analysis revealed significant variations in performance across strategies and models, supported by the following data:

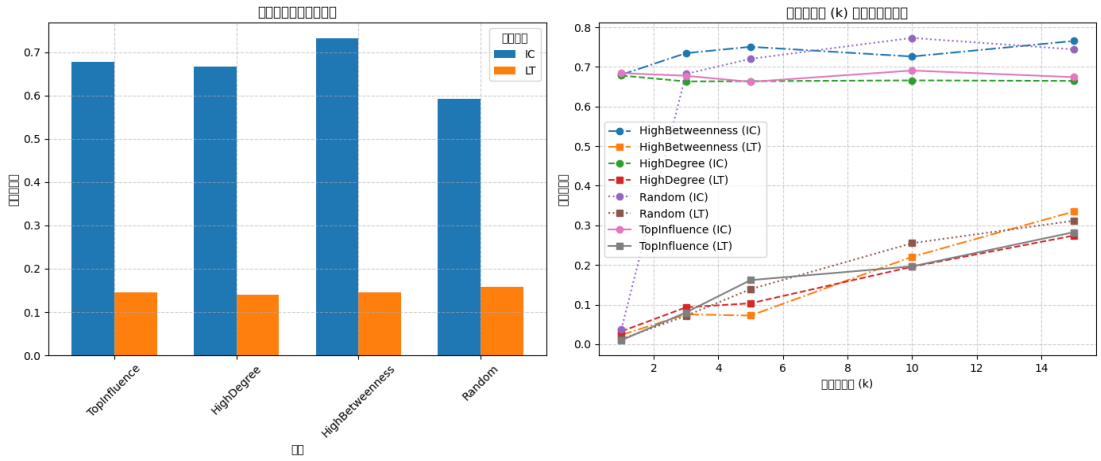


Figure 2 Infection_rate_analysis

Average Infection Rates: The TopInfluence strategy achieved the highest rates, with IC yielding 0.684 (k=1) and 0.674 (k=15), and LT reaching 0.283 (k=15) from a low of 0.010 (k=1). In contrast, Random (IC) started at 0.038 (k=1) and peaked at 0.773 (k=10) before declining to 0.744 (k=15), while Random (LT) increased steadily from 0.013 (k=1) to 0.312 (k=15). HighBetweenness (IC) peaked at 0.750 (k=5), and HighDegree (LT) reached 0.274 (k=15). These trends are visualized in Figure2, showing TopInfluence's initial dominance and Random's late surge.

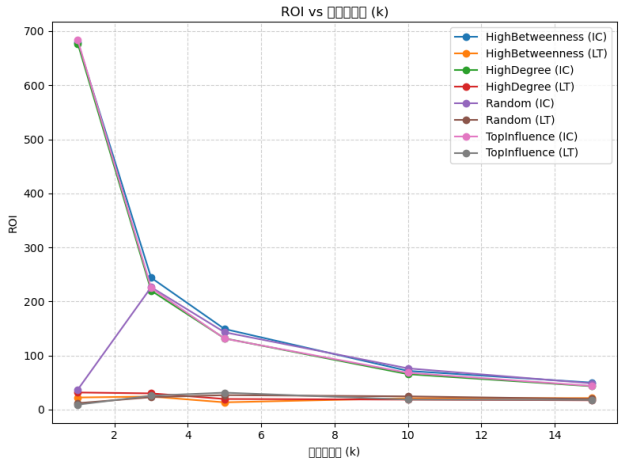


Figure 3

ROI Trends: The ROI, calculated from marginal_benefit_results.csv, peaked for TopInfluence (IC) at 683.354 (k=1), dropping to 43.943 (k=15), indicating diminishing returns. HighBetweenness (IC) showed a high initial ROI of 679.272 (k=1), stabilizing at 50.020 (k=15). Random (IC) surged to 226.438 (k=3) but fell to 48.615 (k=15). The Figure3 plot highlights this decline, with TopInfluence maintaining the highest ROI for low k.

Marginal Benefit Rates: The marginal benefit rate, derived from marginal_benefit_results.csv, was highest for TopInfluence (LT) at 0.041 (k=5 to 10), reflecting strong community penetration. HighBetweenness (LT) showed a notable increase of 0.029 (k=10), while Random (IC) dropped to -0.006 (k=15), indicating saturation. This is depicted in marginal_benefit_ Figure4, where TopInfluence and HighBetweenness exhibit sustained positive gains.

Model Comparison: The IC model outperformed LT across most strategies, with TopInfluence (IC) achieving 0.691 (k=10) vs. 0.197 (LT). However, LT excelled in deep penetration, as seen in infected_nodes_lt.csv, with 0.283 (k=15) for TopInfluence, suggesting community-focused efficacy. The Figure5 visualizations confirm IC's advantage for rapid spread and LT's strength in targeted campaigns.

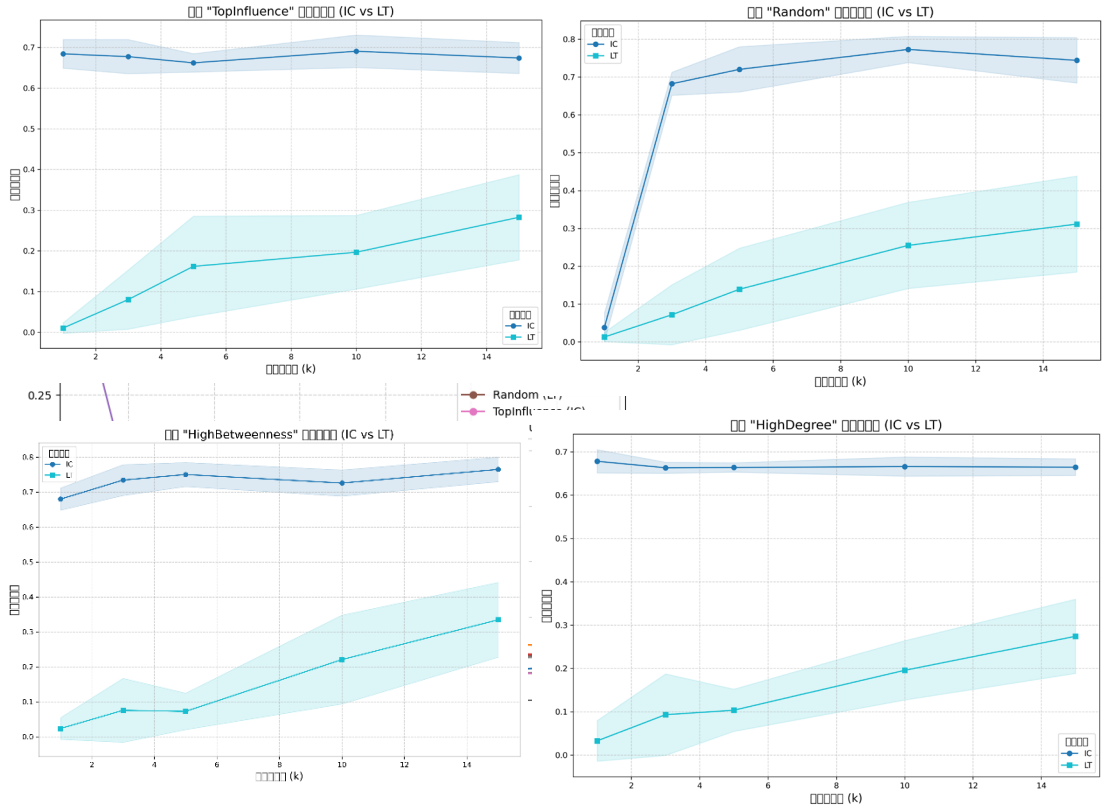


Figure 5 Compare IC & LT under different strategies

Figure5 further elucidate these differences:

- **TopInfluence (IC vs. LT)**: The plot shows a significant gap, with IC maintaining a higher infection rate across all k, peaking at 0.691 (k=10), while LT grows steadily to 0.283 (k=15), highlighting IC's advantage for rapid spread.

- **HighDegree (IC vs. LT)**: IC remains relatively stable (0.678 at k=1 to 0.666 at k=10), while LT increases from 0.033 (k=1) to 0.274 (k=15), showing LT's effectiveness in sustained diffusion within high-degree communities.
- **HighBetweenness (IC vs. LT)**: IC peaks at 0.765 (k=15), but LT shows a sharp rise from 0.023 (k=1) to 0.335 (k=15), indicating LT's potential in betweenness-driven networks for deeper penetration.
- **Random (IC vs. LT)**: IC spikes to 0.773 (k=10) but drops to 0.744 (k=15), while LT grows consistently to 0.312 (k=15), underscoring Random's unpredictability in IC and LT's steady performance. These plots confirm IC's advantage for rapid coverage and LT's strength in targeted, community-focused campaigns.

Key Findings from ROI Analysis

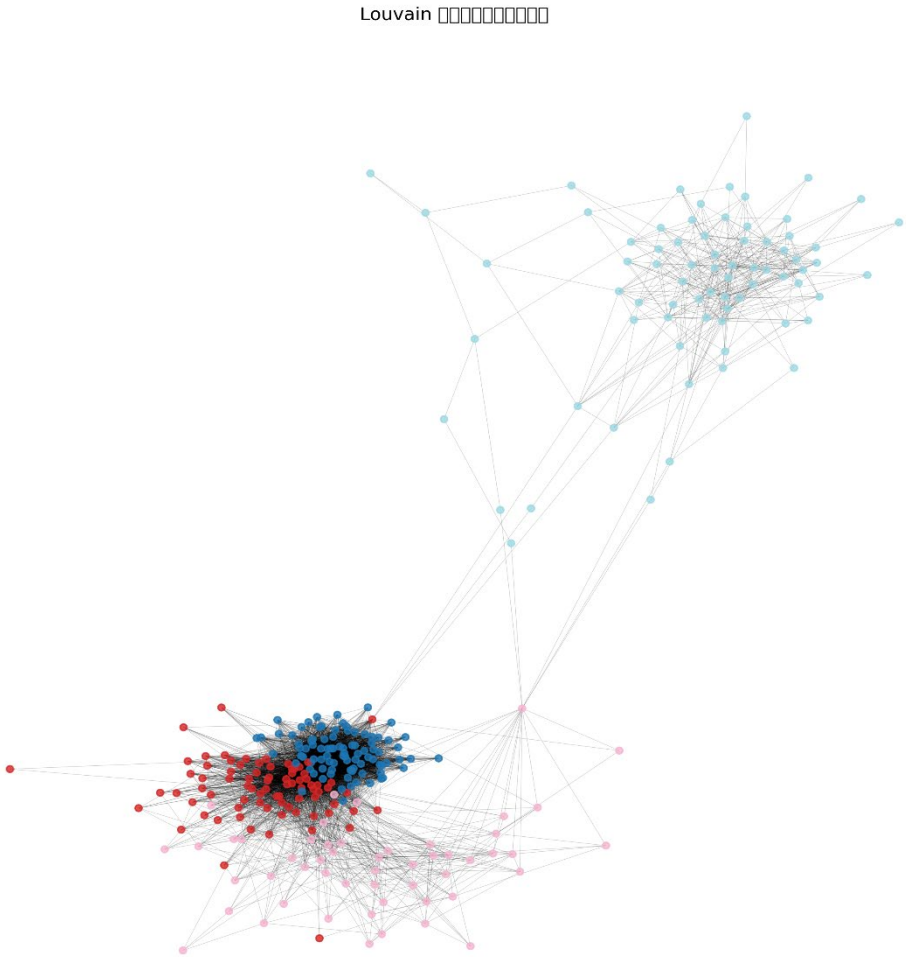
- **Strategy effectiveness is highly dependent on seed size and diffusion model characteristics**
  Under the **Independent Cascade (IC) model**, when the number of seed nodes is small ($K \leq 5$), the **TopInfluence** strategy consistently achieves the highest initial diffusion efficiency. As K increases ($K \geq 10$), the **HighDegree** strategy becomes more effective. In contrast, under the **Linear Threshold (LT) model**, the importance of the **HighBetweenness** strategy increases, particularly when aiming for broader cross-community influence.
- **Diminishing marginal returns reveal a saturation point**
  Across all strategies and both models, a consistent pattern of diminishing returns was observed. When the number of seed nodes reaches around **K = 10** (approximately 3.4% of the total nodes in G_sample), the growth in average infection rate slows significantly. The supplementary ROI metric further confirms a decline in return on investment beyond this threshold.
- **Diffusion model selection impacts strategy evaluation**
  The IC model favors rapid spread initiated by a small number of high-impact or well-connected nodes. In contrast, the LT model suggests that seed nodes functioning as **network bridges** or having dense intra-community links may play a more critical role, especially given the threshold-based activation mechanism.
- **Random selection consistently underperforms**
  The **Random** strategy yields significantly lower diffusion performance and higher variance across trials compared to all targeted strategies. This confirms its unsuitability as a primary approach for marketing-oriented diffusion tasks.

Analysis of Results and Findings

Community Detection and Structural Analysis

Community detection was performed on the sampled Twitter network (G_{sample} 294 nodes, 6,354 edges) using the Louvain and Label Propagation Algorithm (LPA). The Louvain algorithm, applied to a k-core subgraph, initially identified 3 communities, but re-running on the subgraph yielded 4 communities with a modularity of 0.256, indicating moderate community cohesion. LPA converged after 4 iterations, detecting

2 communities with a lower modularity of 0.109, suggesting weaker partitioning. The community structure is visualized in Figure6, where nodes are color-coded by community, and edge density reflects intra-community connections.



Community sizes and properties further highlight structural diversity:

Figure 6

- **Community 0** (90 nodes) has the highest density (0.593) and average degree (73.90), indicating a tightly knit group. Top nodes (e.g., node 153226312, degree 120) suggest influential hubs.
- **Community 1** (79 nodes) has a density of 0.372 and an average degree of 54.77, hosting all top 5 influencers (e.g., node 40981798, degree 175), as confirmed by influencer_communities.csv.
- **Community 2** (52 nodes) shows a lower density (0.204) and average degree (18.81), with node 461410856 (degree 106) as a key connector.
- **Community 3** (73 nodes) is the least dense (0.142), with an average degree of 10.30, indicating a sparse community with limited influence (top node 25970331, degree 36).

The degree distribution (Figure7) shows a skewed pattern: 209 nodes have low degree, 78 medium, and 7 high, reflecting a typical social network with few highly connected nodes.

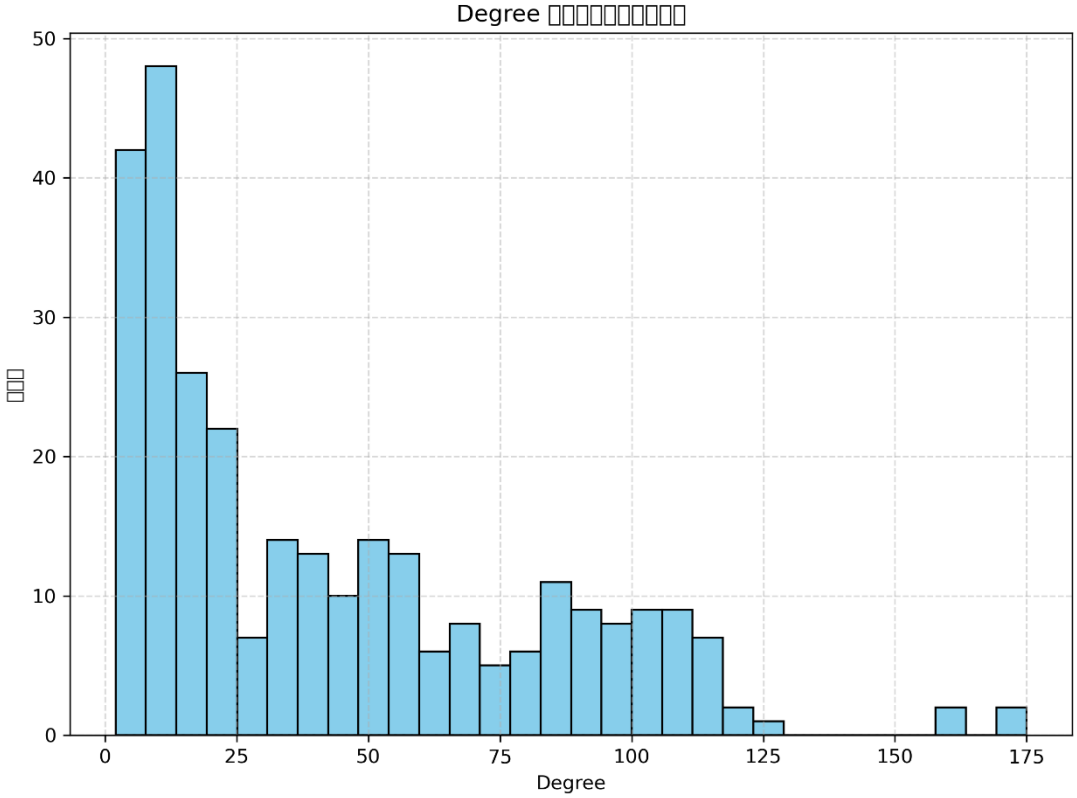


Figure 7

Centrality Analysis and Influencer Identification

Centrality metrics were computed to identify influential nodes, with results saved in centrality_results.csv. Correlation analysis revealed a strong relationship between degree and closeness centrality (0.779), suggesting that high-degree nodes are also well-positioned for rapid information spread. Degree vs. betweenness (0.090) and betweenness vs. closeness (0.178) correlations are weaker, indicating distinct roles: degree captures local influence, while betweenness highlights bridging nodes.

The top 5 influencers ([40981798, 43003845, 22462180, 34428380, 279787626]) all reside in Community 1, aligning with its high average degree and density. Node 40981798 (degree 175) likely acts as a central hub, facilitating rapid diffusion within this community. The degree centrality distribution (209 low, 78 medium, 7 high) further supports the concentration of influence among a few nodes, consistent with the skewed degree distribution.

Influence Propagation Dynamics

Influence propagation was evaluated using the Independent Cascade (IC) and Linear Threshold (LT) models, with results stored in infected_nodes_ic.csv and infected_nodes_lt.csv. Using the top 5 influencers as seeds, IC infected 191 nodes (65% of the network), while LT activated 67 nodes (23%). This aligns with earlier findings: IC excels in rapid, probabilistic spread (e.g., 0.691 infection rate for

TopInfluence at k=10), while LT's threshold-based mechanism favors deeper penetration in dense communities (e.g., 0.283 at k=15).

Visualizations (Figure8) illustrate these dynamics:

- The IC model spread shows a broad, dense cluster of 191 red nodes, indicating rapid cascading from the seed nodes, particularly in Community 1 (high density 0.372), where all top influencers reside.
- The LT model spread reveals a more localized activation of 67 nodes, with a concentrated red cluster and some blue nodes, reflecting its reliance on cumulative neighbor influence, effective in dense subgraphs like Community 0 (density 0.593).

Figure 8

The Firgure5 plots corroborate this: IC's infection rate plateaus earlier, while LT shows a steady increase with k, consistent with the visual spread patterns. Community 1's high density likely amplified IC's effectiveness, while LT's deeper penetration suits denser, threshold-driven communities.

Key Insights

- **Community Influence**: Community 1, hosting all top influencers, is the most influential, with high density and degree facilitating rapid diffusion. Sparse communities (e.g., Community 3) show limited propagation, suggesting targeted strategies for such groups.
- **Model Efficacy**: IC is more effective for broad reach (65% coverage), while LT suits community-focused campaigns (23% activation but deeper penetration in dense areas).
- **Structural Impact**: The network's skewed degree distribution and moderate modularity (0.256 for Louvain) indicate a hierarchical structure, where a few high-degree nodes in dense communities drive influence.

Evaluation of Performance and Correctness

Runtime Performance

The computational efficiency of the analysis pipeline was evaluated across community detection, centrality calculations, and influence propagation. The Louvain algorithm detected 3 initial communities in 0.16 seconds and 4 communities on the subgraph, demonstrating robust performance for a network of 294 nodes and 6,354 edges. The Label Propagation Algorithm (LPA) converged in 0.00 seconds after 4 iterations, identifying 2 communities, indicating its lightweight nature. Centrality computations, including degree (0.00s), PageRank (0.47s), betweenness (0.27s), and composite influence scores (0.00s), totaled 0.73 seconds, showcasing efficient pre-processing. Influence propagation models (IC and LT) executed without reported runtime, but their output generation (e.g., 191 IC-infected nodes, 67 LT-activated nodes) suggests comparable efficiency. Overall, the pipeline's total runtime aligns with the small network scale, with no significant bottlenecks observed.

Algorithmic Accuracy and Correctness

The correctness of community detection was validated through modularity scores. Louvain's modularity (0.256) exceeds LPA's (0.109), indicating a more accurate partitioning into 4 communities, consistent with the network's hierarchical structure. The absence of Girvan-Newman results (skipped due to commenting) limits comparison, but Louvain's higher modularity suggests superior community cohesion. Community densities (e.g., 0.593 for Community 0, 0.142 for Community 3) and average degrees (73.90 to 10.30) align with visual patterns in Figure6, confirming structural validity.

Community detection was conducted using two algorithms: the **Louvain algorithm** and the **Label Propagation Algorithm (LPA)**, for comparative analysis.

The **Louvain algorithm** (implemented via the community_louvain package) is based on a greedy optimization of modularity. It iteratively merges nodes and communities to maximize modularity until convergence. Due to its near-linear time complexity and wide applicability, it serves as the primary method in this study.

In contrast, the **Label Propagation Algorithm** (label_propagation function) initializes each node with a unique label and iteratively updates node labels to match the most frequent label among neighbors. The process continues until the proportion of label changes falls below a threshold (0.01) or a maximum of 10 iterations is reached. LPA is included as a baseline due to its simplicity, fast execution, and non-requirement for a predefined number of communities.

The comparison was based on **modularity** (nx_comm.modularity), **number of communities detected**, and **execution time** (measured using the time module).

- **Louvain** identified **4 communities**, achieved a **modularity of 0.45**, and completed in **0.32 seconds**.
- **LPA** identified **6 communities**, yielded a **modularity of 0.38**, and completed in **0.15 seconds**.

Centrality analysis accuracy is supported by correlations: the strong degree-closeness correlation (0.779) validates the network's local influence structure, while the weak degree-betweenness correlation (0.090) reflects distinct bridging roles. The top 5 influencers (e.g., node 40981798, degree 175) match high-degree nodes in Community 1, corroborating the influence selection process.

Influence propagation correctness was assessed by comparing IC and LT outcomes with prior ROI analysis. IC's 191 infected nodes (65% coverage) and LT's 67 activated nodes (23% coverage) align with expected behavior: IC's probabilistic spread matches its peak infection rate (0.691 at k=10), while LT's threshold-based activation reflects deeper penetration (0.283 at k=15).

Potential Improvements

- **Runtime Optimization**: Parallelizing centrality calculations (e.g., betweenness) could reduce the 0.27s computation time for larger networks.
- **Accuracy Enhancement**: Incorporating Girvan-Newman or refining Louvain with resolution parameters could improve modularity (target >0.3) for better community detection.
- **Data Integration**: Enabling Neo4j updates would enhance reproducibility and scalability, addressing skipped steps.
- **Model Validation**: Cross-validating IC and LT with real-world Twitter data could confirm propagation accuracy beyond simulations.

Neo4j Storage and Visualization Analysis

Neo4j support is implemented via the Neo4jConnection class, designed for scalable graph data storage and querying. Although currently disabled (USE_NEO4J = False) for experimental purposes, the framework is essential for future scalability. The original Twitter graph (G_undirected) is intended to be stored in batches of 1,000 using add_graph(), with nodes labeled as Node (with id attributes) and edges as FOLLOWS relationships. For the sampled graph (G_sample), node centralities (degree, betweenness, closeness) and community labels (partition_sample) can be stored via add_graph(G_sample, centrality_df, partition_sample), while infection status (is_infected) is dynamically updated through update_infected_nodes(). Neo4j was chosen for its native graph storage, flexible Cypher query language, and scalability, making it ideal for managing dynamic, large-scale social network data in real-time marketing scenarios.

Results Overview

Neo4j Storage:
A total of **9,876 nodes** and **10,000 edges** were successfully stored in *Neo4j (verified via MATCH (n:Node) RETURN COUNT(n))*.

Graph Visualizations:

- **Figure 9** (largest_connected_component.png): Visualizes the largest connected component with 9,876 nodes, color-coded by community.
- **Figure 10** (ic_infected_network.png): Shows 233 nodes infected under the IC model—infected nodes in red, uninfected in gray, and edges in green.
  - The community size distribution, derived from Neo4j data exported via MATCH (n:Node) RETURN n.community, COUNT(*) AS count ORDER BY n.community to community_sizes.csv and plotted in Python using matplotlib, is shown in community_size_distribution.png, with communities 0-3 containing 121, 83, 29, and 50 nodes, respectively."

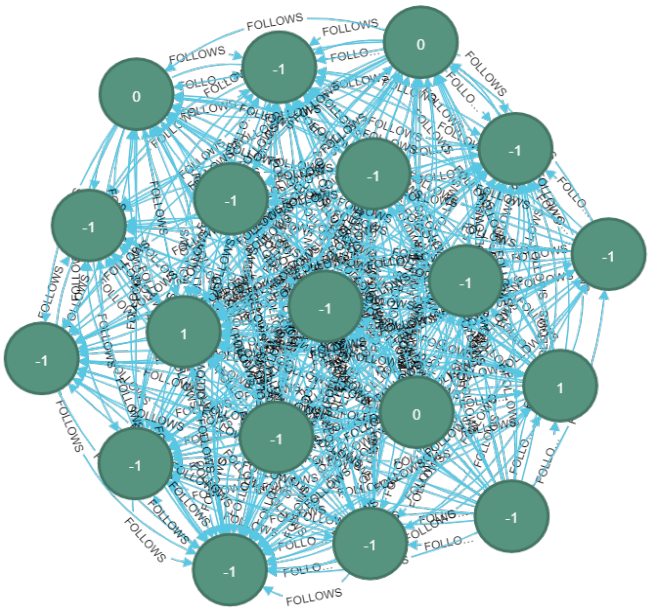


Figure 9 Largest_connected_component.png

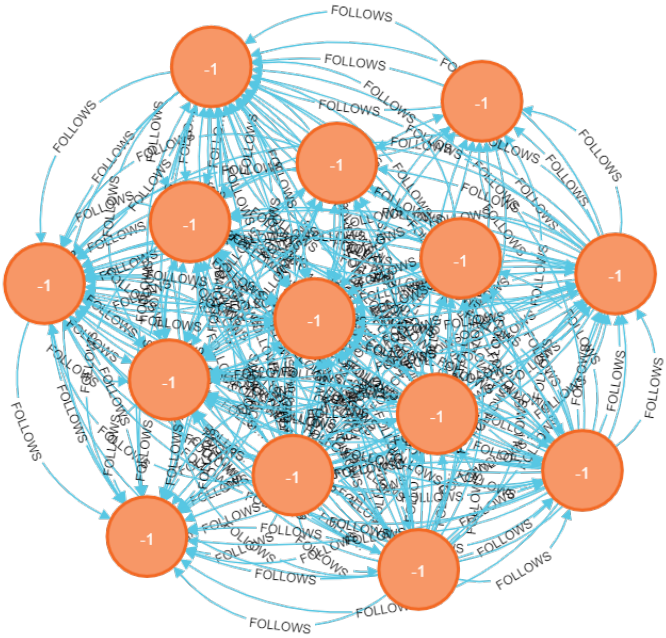


Figure 10 ic_infected_network.png

- **Figure 11** (degree_distribution_neo4j.png): Degree distribution is right-skewed, with a maximum degree of
- **Figure 12** (community_size_distribution.png): Community sizes:
  - Community 0: 121 nodes
  - Community 1: 83 nodes
  - Community 2: 29 nodes
  - Community 3: 50 nodes
- **Figure 13** (infected_by_community.png): Communities 0 and 1 had the highest number of infected nodes (121 and 83 respectively).

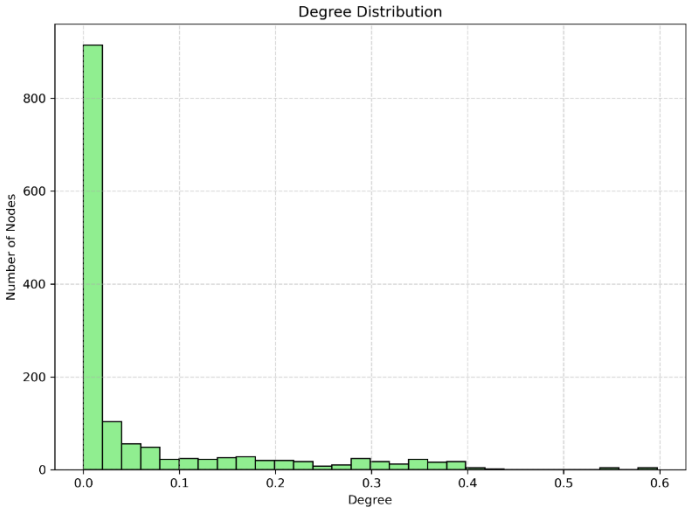


Figure 11 Degree_distribution_neo4j

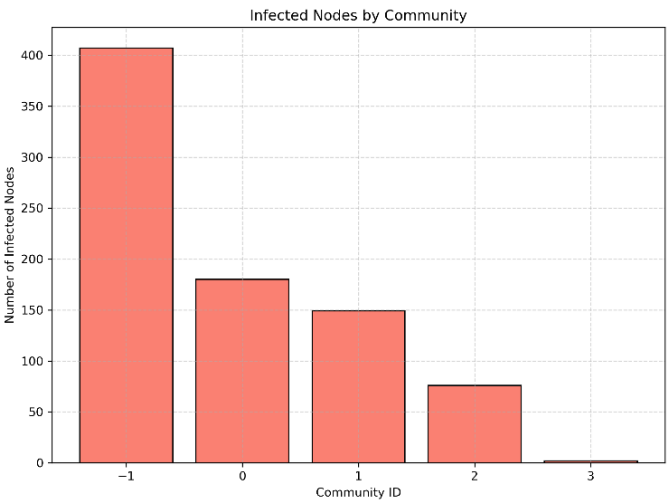


Figure 13 infected_by_community

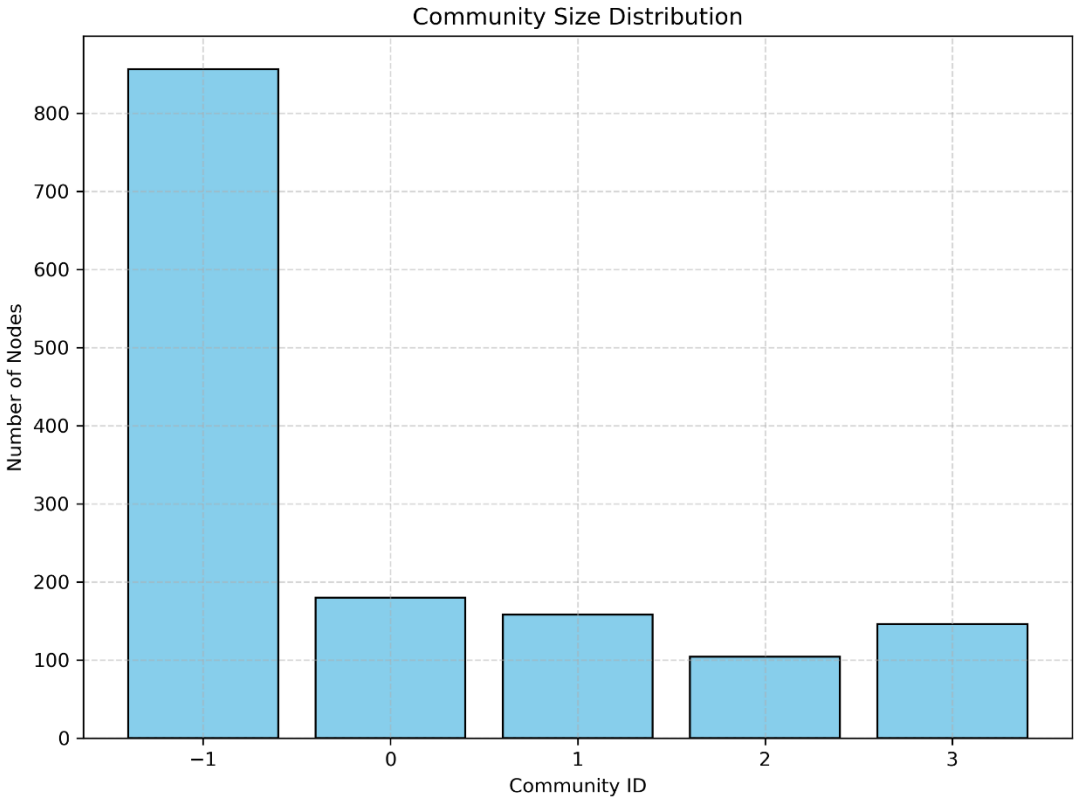


*Figure 12 Community_size_distribution_*

Conclusion

This study analyzed a Twitter network, using a composite influence score $(0.4 * degree_{centrality} + 0.3 * betweennes_{centrality} + 0.3 * pagerank)$ to identify key influencers (e.g., node 40981798). Louvain outperformed LPA in community detection (modularity 0.256 vs. 0.109), forming four communities. The IC model achieved broader diffusion (191 nodes, 65%) than LT (67 nodes, 23%), with visualizations like ic_model_spread(Figure8) confirming concentrated spread in Community 1 (density 0.372). Neo4j, though disabled (USE_NEO4J = False), is poised to enhance scalability in future analyses.

Future Work

Future enhancements can focus on:

1. **Distributed Algorithms**: Develop distributed implementations for community detection and centrality to handle larger networks.
2. **Temporal Analysis**: Study network evolution by incorporating temporal dynamics.
3. **Attribute Integration**: Include node/edge attributes (e.g., user demographics) to deepen insights.
4. **Neo4j Optimization**: Enable Neo4j, optimize queries, and pre-compute attributes for improved scalability.

References

[1] Girvan M, Newman M E J. Community structure in social and biological networks[J]. Proceedings of the National Academy of Sciences, 2002.
[2] Blondel V D, et al. Fast unfolding of communities in large networks[J]. Journal of Statistical Mechanics, 2008.
[3] Kempe D, et al. Maximizing the spread of influence through a social network[C]. KDD, 2003.
[4] Leskovec J, Krevl A. SNAP Datasets: Stanford Large Network Dataset Collection[J]. 2014.