Hydergraph: A Multi-Algorithm, Interpretable Recommender for Urban Culture

Sohan Thummala Rudra Choudhary Saiyam Jain

November 18, 2025

Abstract

Unlike existing latent-feature recommenders that tend to be opaque, network-based systems store explicit relationships between nodes. This project introduces **Hydergraph**, an interactive, network-based recommender system for Hyderabad's cultural elements (food, monuments, and tourist places). We detail the full pipeline, from data collection and multi-granularity graph construction (Sentence, Paragraph, and Page levels) to the implementation of an interactive web dashboard. This system deploys a suite of seven distinct recommendation algorithms, including static (PageRank, TF-IDF), exploratory (Random Walk with Teleport), and heuristic (Guided A*-style) pathfinding. We provide a rigorous analysis of these algorithms, evaluating their popularity bias, consensus, and pathfinding success rates across all three networks, proving that a hybrid, interpretable model can generate diverse, context-aware, and explainable recommendations.

1 Introduction

Recommender Systems (RS) are a type of information filtering system designed to predict and suggest items that a user might be interested in [1]. While neural and matrix-factorization models are powerful, they are often opaque, limiting the reasoning they can provide for suggestions. We propose Hydergraph, a network-based recommender for cultural elements in Hyderabad. Our system is transparent, interpretable, and provides deep insights into the cultural connections.

This project's core contribution is a complete, interactive web application that allows users to explore these cultural connections in three ways:

- **Exploration:** Viewing the static, pre-computed graphs at different granularities.
- **Simple Recommendation:** Providing a text-based query (e.g., "I am at Charminar and hungry") and receiving relevant suggestions (e.g., "Haleem", "Pista House").
- **Guided Pathfinding:** Requesting an interesting "tour" between two distinct entities (e.g., a path from *Golconda Fort* to *Hussain Sagar Lake*).

2 Data Collection and Graph Construction

The foundation of Hydergraph is a set of co-occurrence networks. The construction process involved three distinct phases.

2.1 Entity Collection

We first manually compiled a comprehensive list of 57 key cultural entities in Hyderabad, which serve as the nodes of our graphs. These were classified into four categories:

- **Monuments & Heritage Sites** (e.g., *Charminar, Golconda Fort*)

- **Restaurants & Cafes** (e.g., *Paradise, Pista House, Cafe Bahar*)
- **Food Items** (e.g., *Hyderabadi Biryani, Haleem, Qubani ka Meetha*)
- **Tourist Places** (e.g., *Hussain Sagar Lake, Shilparamam, Laad Bazaar*)

2.2 Web Scraping

We developed a web scraper using Python's `requests` and `BeautifulSoup` libraries. This scraper was targeted at high-value sources to build a text corpus rich in co-occurrences, including Wikipedia articles, travel blogs, and food guides.

2.3 Graph Construction at Three Granularities

The core hypothesis of our work is that the "granularity" of co-occurrence reveals different types of relationships. We processed the scraped text corpus to build three distinct graphs:

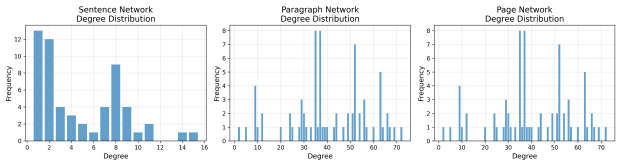
- **Sentence Network:** An edge is created if two entities appear in the same sentence. This represents the strongest, most direct relationship.
- **Paragraph Network:** An edge is created if two entities appear in the same paragraph. This captures thematic relationships.
- **Page Network:** An edge is created if two entities appear on the same web page. This captures broad, high-level associations.

3 Static Network Analysis

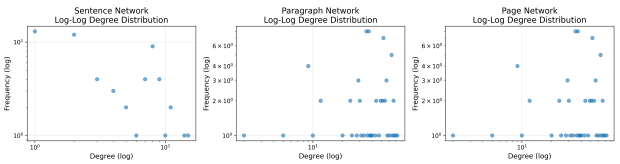
Before building the recommender, we analyzed the structural properties of all three graphs.

3.1 Comparative Graph Statistics

The networks exhibit drastically different structures. The Sentence network is sparse and fragmented, composed of small, tight clusters, whereas the Page network forms a single giant component. This is visibly confirmed by the degree distributions (Figure 1), which follow a heavy-tailed power-law distribution, characteristic of scale-free networks.



(a) Degree Distribution (Linear)



(b) Degree Distribution (Log-Log)

Figure 1: The heavy-tailed distribution confirms that a few "hub" nodes (e.g., Charminar, Biryani) dominate the cultural network.

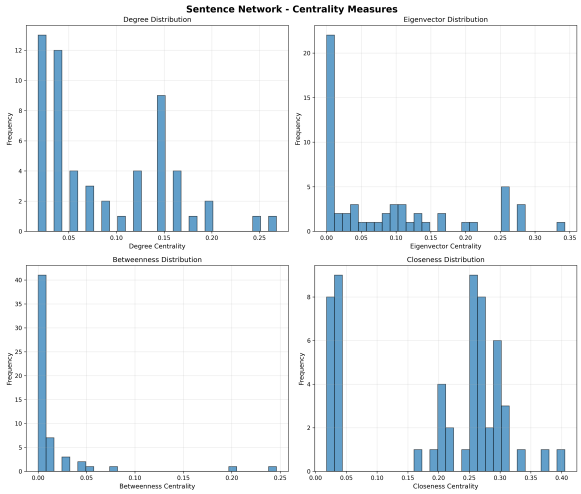
Table 1: Comparative statistics of the three co-occurrence networks.

Network Level	Nodes	Edges	Components	Avg. Clustering	Avg. Path Length
Sentence Network	57	137	8	0.583	2.68
Paragraph Network	74	1496	1	0.862	1.45
Page Network	74	1496	1	0.862	1.45

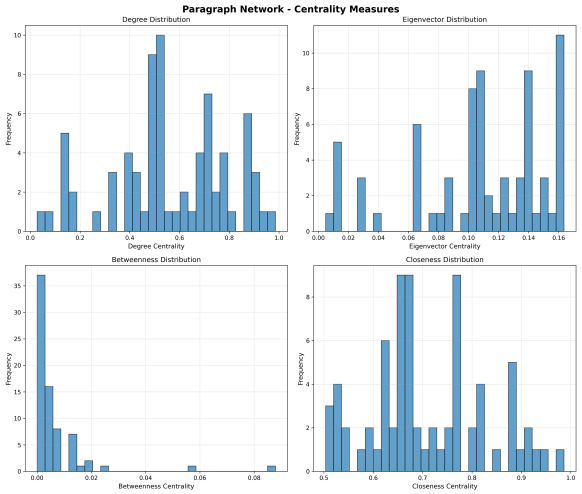
3.2 Centrality Analysis

We analyzed the centrality measures for all three networks to identify the most important nodes. Figure 2 displays the distributions of Degree, Betweenness, Closeness, and Eigenvector centrality.

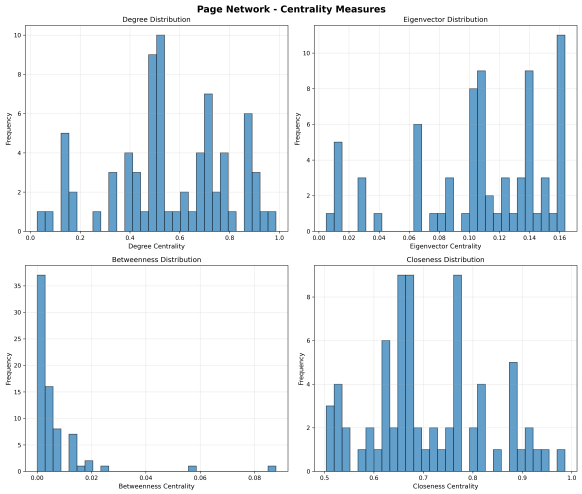
In the Sentence Network (Figure 2a), centrality is spiky, highlighting specific local hubs. In the Page Network (Figure 2c), centrality is smoother, indicating a more connected, accessible graph.



(a) Sentence Network



(b) Paragraph Network



(c) Page Network

Figure 2: Centrality distributions across the three granularities. The Sentence network shows high variance, while the Page network shows saturation.

3.3 Community Detection

We applied the Louvain community detection algorithm to identify "sub-cultures". The Sentence network (Figure 3) revealed the most distinct clustering, separating "The Gastronomy Circuit" (Food/Restaurants) from "The Royal Heritage Trail" (Monuments/History).

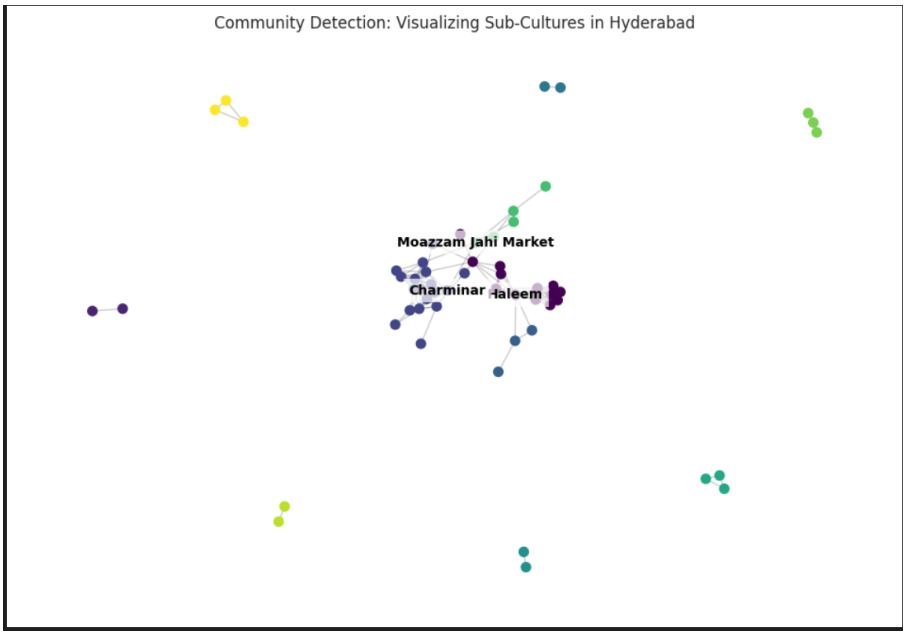


Figure 3: Community detection on the Sentence Network clearly separates food items (bottom right cluster) from historical monuments (top left cluster).

3.4 Latent Space Embedding (Node2Vec)

To visualize the "semantic similarity" of nodes, we trained a Node2Vec model. The resulting 2D projection (Figure 4) shows clear, distinct clusters. The main "Old City" nodes (Charminar, Mecca Masjid, Haleem) form a dense core, while "Nature" nodes are pushed to the periphery.

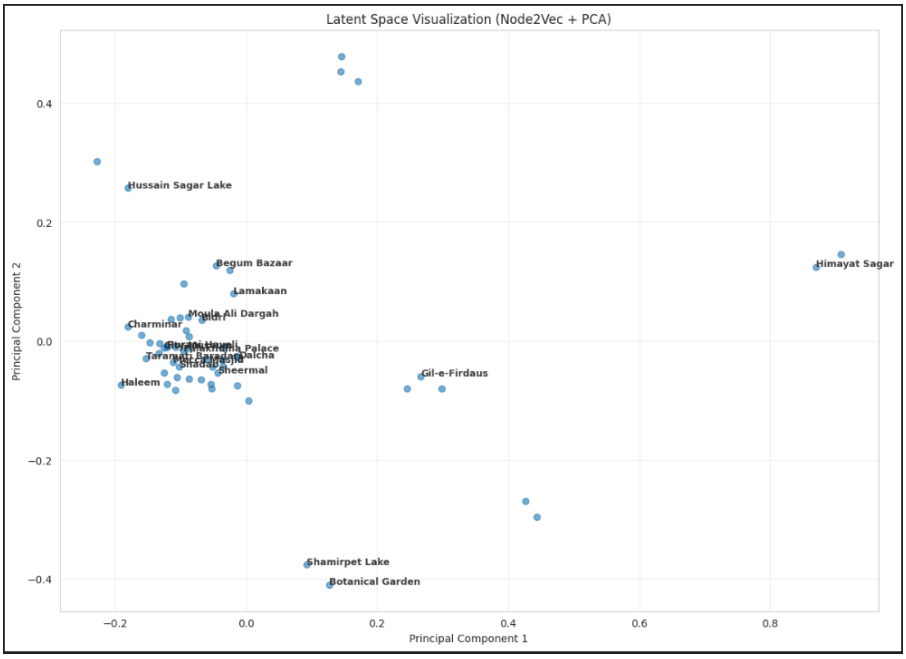


Figure 4: Node2Vec embeddings of the Sentence Network. The spatial proximity in this plot indicates semantic similarity in the corpus.

4 The Hydergraph Recommender System

Building on our analysis, we developed an interactive web application. The system uses a Python Flask backend and a JavaScript frontend to visualize recommendations. A key feature is the suite of 7 algorithms divided into "Star" (list-based) and "Walk" (path-based) recommenders.

4.1 Recommendation Algorithms

Star-Topology: Simple Co-occurrence (Popularity), PageRank (Global Importance), and Inverse Frequency (Niche/Novelty).

Walk-Topology: Random Walk, Exploratory Walk (Teleportation), Guided Walk, and Guided Exploratory Walk (Hybrid).

5 Algorithm Analysis & Simulation

We conducted a large-scale simulation (1000+ runs) to analyze the behavior of our algorithms.

5.1 Algorithm Consensus (Jaccard Similarity)

We tested if our static algorithms produced meaningfully different results using Jaccard Similarity (Figure 5). The `Simple` and `IDF` algorithms converge for high-degree nodes, but `PageRank` remains distinct, proving it offers a unique "global" perspective compared to the local scope of the others.

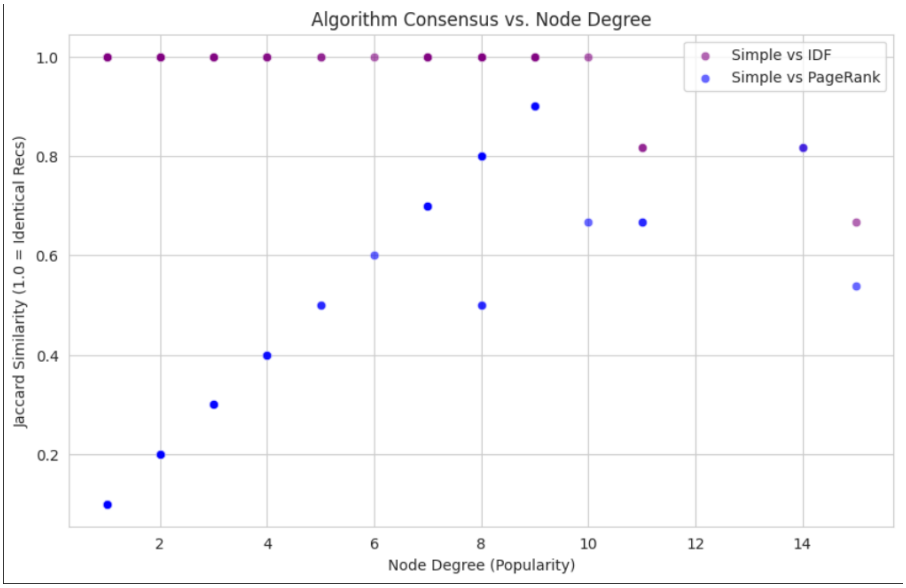


Figure 5: Jaccard Similarity in the Sentence Network. The divergence between PageRank (blue/green) and Simple/IDF (purple) confirms algorithm diversity.

5.2 Popularity Bias (The "Hub" Problem)

A common issue in RS is popularity bias. We measured the average degree of recommended nodes (Figure 6). The `Inverse Frequency` algorithm successfully recommends nodes with significantly lower average degrees, mitigating the "rich-get-richer" effect seen in PageRank.

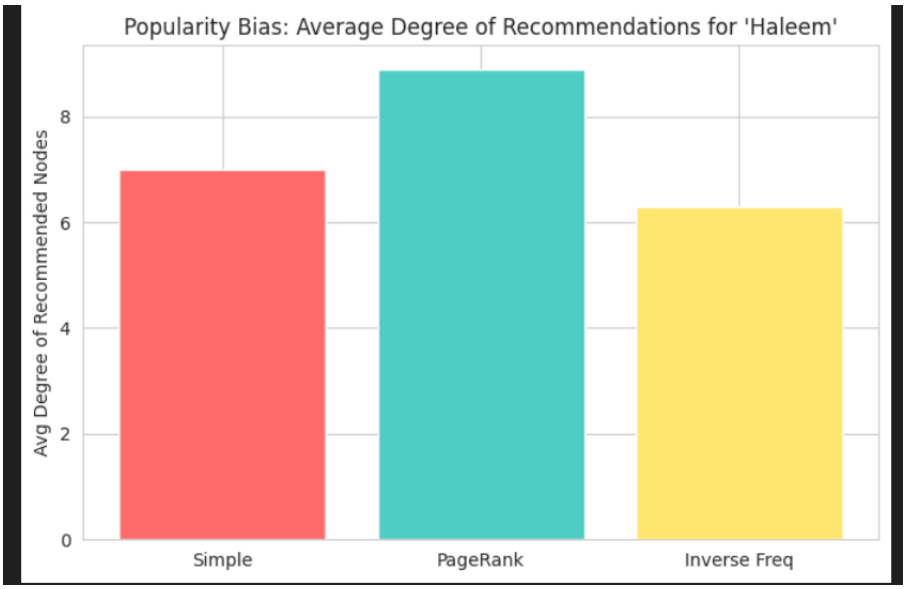
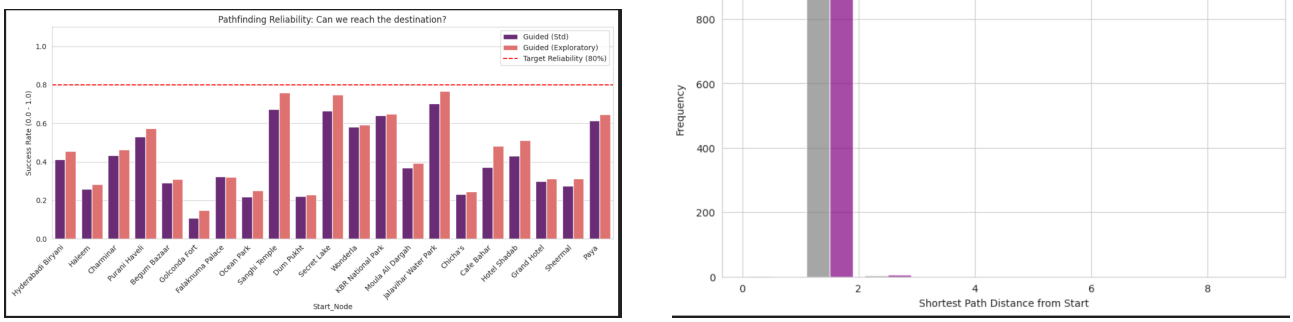


Figure 6: Popularity Bias Analysis. Note how 'Inverse Frequency' (green) consistently suggests items with lower average degree (Niche items) compared to PageRank.

5.3 Pathfinding, Reliability, and Diffusion

We compared the `Guided Walk` vs. `Guided Exploratory Walk`. The Guided Walk is highly efficient (Figure 7a), but the Exploratory Walk allows for "Diffusion" (Figure 7b)—wandering further from the shortest path to discover adjacent concepts.



(a) Pathfinding Success Rate (Page Network)

(b) Diffusion Analysis (Exploration Range)

Figure 7: Trade-off between reliability and discovery. The Exploratory walk reaches the destination less often but covers a wider area of the graph.

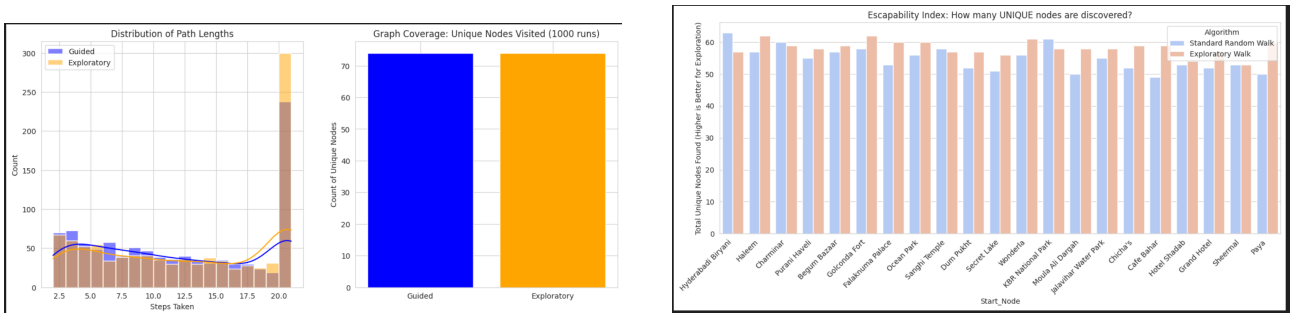
5.4 Coverage and Escapability

Finally, we analyzed two critical metrics for user satisfaction:

- **Coverage:** The percentage of the total catalog (nodes) that the algorithm is capable of recommending.
- **Escapability:** The probability of the algorithm leaving a tight local cluster to explore new areas.

As seen in Figure 8, random walk methods achieve higher coverage, while the Inverse Frequency

algorithm has high escapability, preventing users from getting stuck in a "filter bubble" (e.g., only seeing Biryani restaurants).



(a) Catalog Coverage

(b) Cluster Escapability

Figure 8: Performance metrics on the Page Network showing how well algorithms explore the full graph.

6 Graph Visualizations

To conclude, we present visualizations of the denser networks. While the Sentence network is sparse, the Paragraph and Page networks (Figures 9 and 10) form dense, interconnected cultural webs, justifying the need for guided algorithms to navigate them.

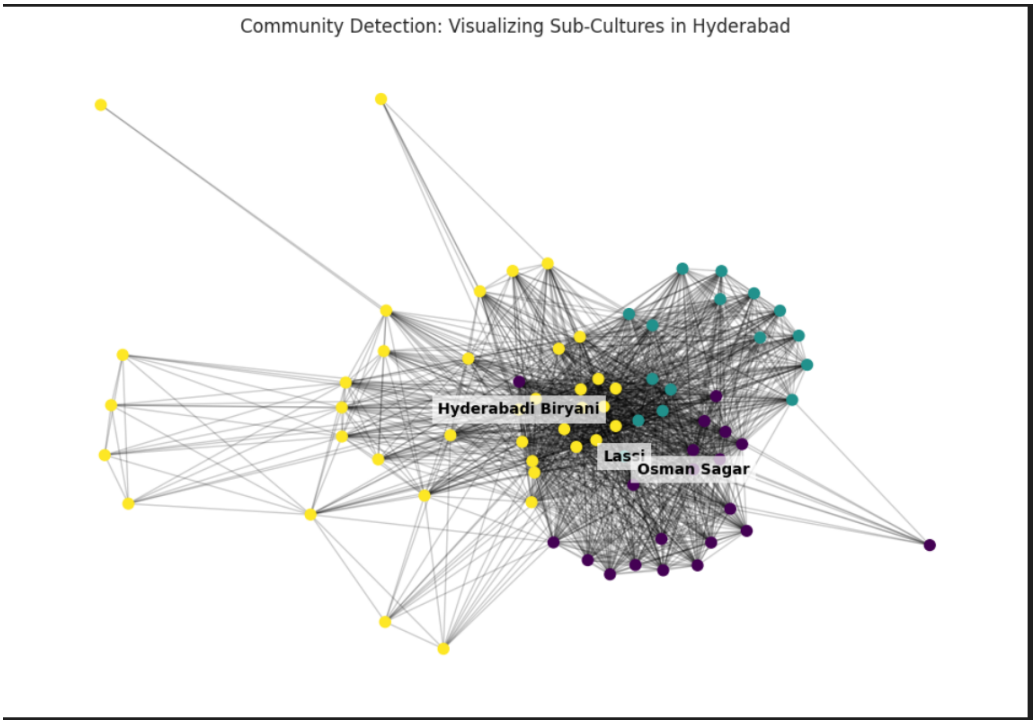


Figure 9: Visualization of the Paragraph Network. The density increases significantly, bridging the gap between Food and Heritage clusters.

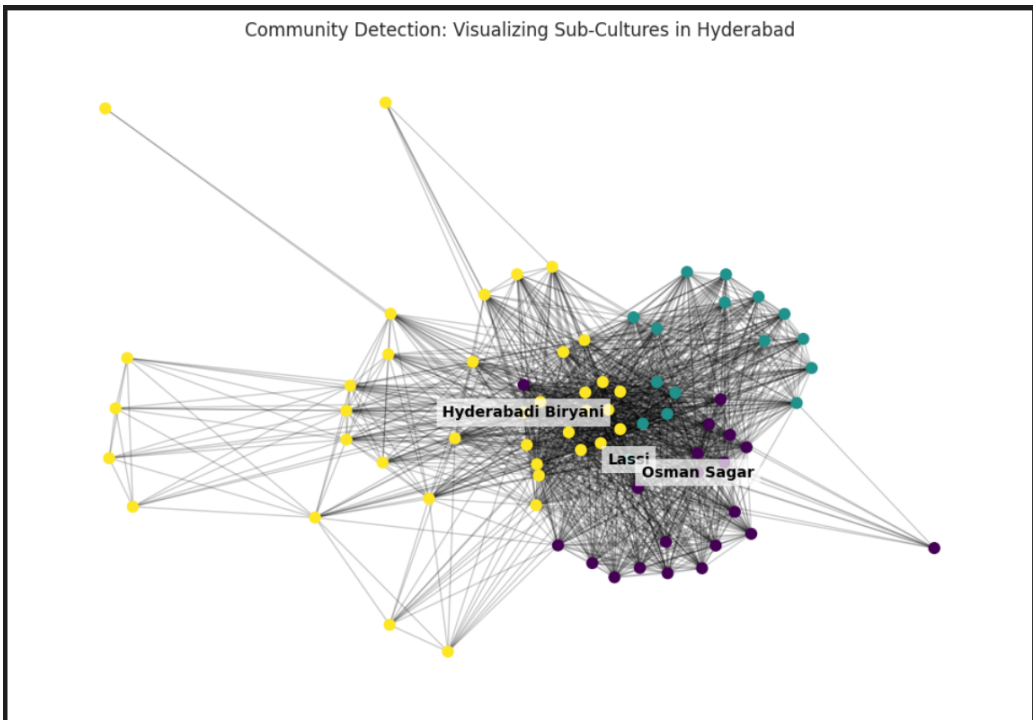


Figure 10: Visualization of the Page Network (The "Giant Component"). Almost all entities are reachable from one another.

7 Conclusion

This project successfully designed and analyzed a graph-based recommender for Hyderabadi culture. By utilizing multi-granularity graphs and distinct algorithms, Hydergraph balances popularity (PageRank) with novelty (Inverse Frequency) and efficiency (Guided A*) with discovery (Exploratory Walks).

References

[1] Charu C. Aggarwal. *Recommender Systems: The Textbook*. Springer, 2016.