Project Report: Friend Recommendation System Using Social Network Analysis

---

1. Title

Friend Recommendation System Using Social Network Analysis

---

2. Abstract

This project aims to develop a Friend Recommendation System that suggests potential friends to users in a social network based on shared connections. Using graph theory and social network analysis, the system leverages the concept of **common neighbors** to identify users with overlapping friend groups. The project demonstrates the use of the **NetworkX library** in Python to implement and visualize the recommendation system. The results highlight the effectiveness of simple graph-based methods for personalized friend recommendations.

---

3. Introduction

Social networks are a crucial part of modern digital ecosystems. Platforms like Facebook, LinkedIn, and Twitter use recommendation systems to enhance user engagement. One of the most common use cases is friend recommendation, where users are suggested connections based on various factors like shared friends, interests, or interactions. This project focuses on implementing a basic Friend Recommendation System using **common neighbors** as the similarity measure.

---

4. Objectives

1. Develop a basic Friend Recommendation System using social network analysis.
2. Use graph theory concepts to identify potential friend recommendations.
3. Visualize the social network and the recommendations for better understanding.

---

5. Methodology

1. **Dataset Preparation**:
    - A synthetic dataset representing a social network was created with users as nodes and friendships as edges.
    - Additional datasets like the **Zachary's Karate Club Graph** or real-world data from APIs (e.g., Twitter) can be used.
2. **Graph Construction**:
    - A graph was constructed using the NetworkX library in Python. Each user is represented as a node, and an edge between two nodes indicates a friendship.
3. **Friend Recommendation Algorithm**:
    - The algorithm identifies potential friends by calculating the number of **common neighbors** between a user and others in the network.
    - Recommendations are ranked based on the number of shared connections.
4. **Visualization**:
    - The social network graph was visualized using Matplotlib to provide insights into connections and recommendations.

---

6. Tools and Technologies

- **Programming Language**: Python
- **Libraries**: NetworkX, Matplotlib
- **Dataset**: Synthetic graph dataset or publicly available datasets
- **IDE**: Jupyter Notebook or any Python IDE

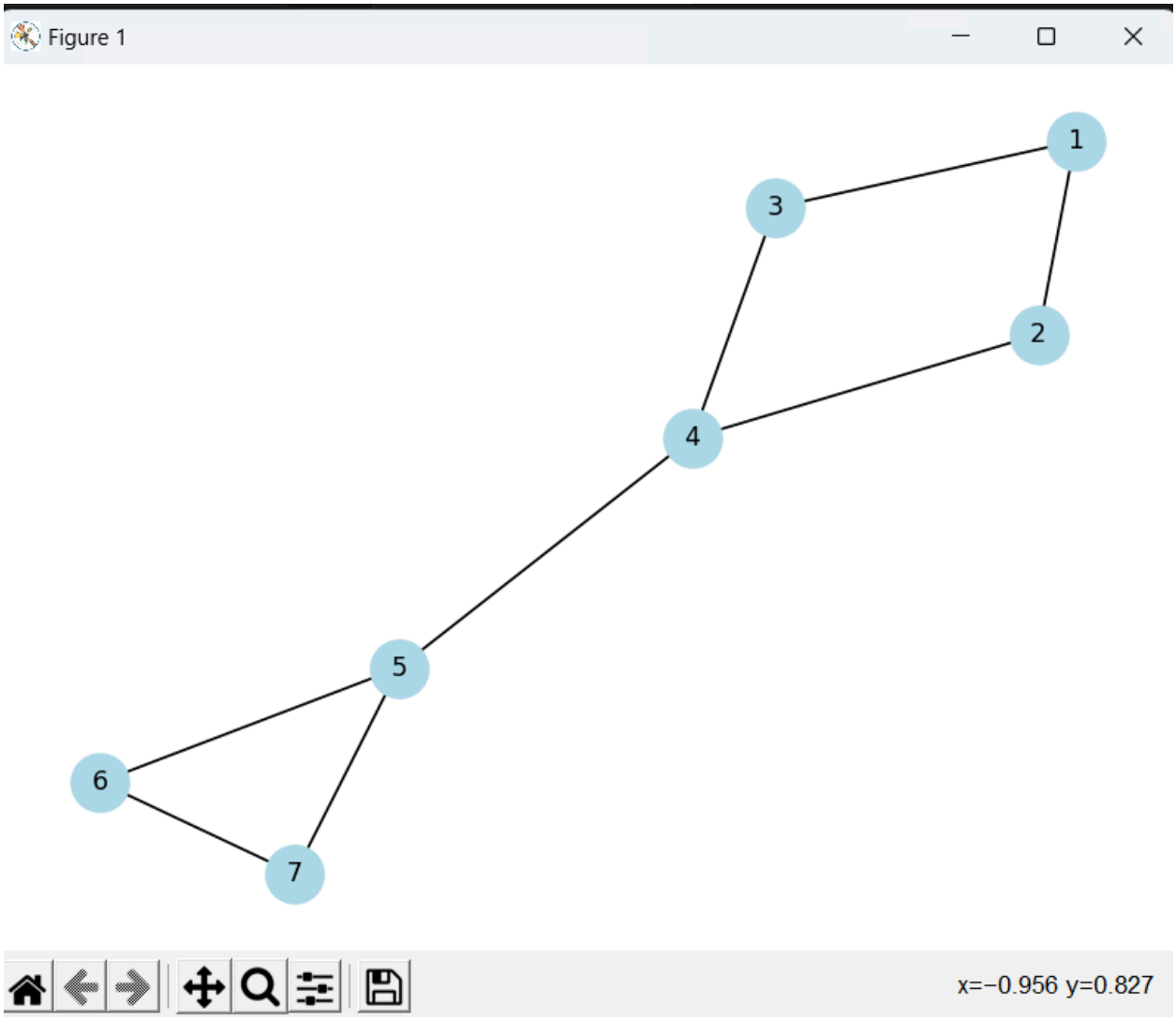
---

7. Results

Friend Recommendations:
For a given node, the system generated a ranked list of potential friends based on the number of common neighbors.
Example output for Node 1:

Node 5 (Common Neighbors: 2)
Node 6 (Common Neighbors: 1)

- **Visualization**:
  A graph of the social network was generated, highlighting existing connections and recommendations.



---

8. Discussion

The results demonstrate that simple graph-based techniques, like analyzing common neighbors, can effectively identify potential friends in a network. While the current implementation uses synthetic data, it can be scaled to real-world datasets with minor

adjustments. However, the simplicity of this approach may limit its applicability to more complex social networks where factors like user interests or activity patterns are significant.

---

9. Conclusion

The project successfully implemented a Friend Recommendation System using social network analysis. By leveraging common neighbors as a similarity metric, the system provides a foundational approach to recommending friends in a social network. Future work can involve integrating machine learning models and additional factors like user activity or interests to enhance the recommendations.

---

10. Future Scope

1. Incorporate **real-world datasets** (e.g., from APIs like Twitter or Facebook).
2. Use advanced metrics such as **Jaccard similarity**, **Adamic-Adar index**, or **Katz index**.
3. Integrate other features like user interests, geographic proximity, or interaction patterns.
4. Develop a user-friendly **web or mobile interface** for live testing.

---

11. References

1. NetworkX Documentation: https://networkx.org/
2. SNAP Datasets: http://snap.stanford.edu/data/
3. Social Network Analysis Book: Wasserman, S., & Faust, K. (1994). *Social Network Analysis: Methods and Applications*.

Let me know if you need help customizing this report or adding specific sections!