Social Circles: Community Analysis and Link Prediction Using Facebook100

Khushi Patel and Narasimha Rohit Katta

Chapter 1

Introduction

1.1 Problem Statement

Social networks have become a cornerstone of modern communication, influencing personal relationships, professional networking, and online interactions. Platforms like Facebook, Twitter, and Instagram facilitate these connections, but understanding how relationships form and evolve within such networks remains a significant challenge.

The task of **link prediction**—forecasting potential connections between individuals in a network—has widespread applications:

- **Friend Recommendations:** Suggesting potential friends or connections to users.

This project addresses the link prediction problem by analyzing the **Facebook100 dataset**, which represents social interactions at a college level. In this dataset:

- Nodes represent individual users.
- Edges represent friendships between users.

The goal is to predict potential links in the network by leveraging structural properties of the graph and applying machine learning techniques. By uncovering hidden patterns in social networks, this project aims to provide insights into how relationships are formed and how they can be effectively predicted.

1.1.1 Our Approach

To address these challenges, this project combines traditional graph-theoretic heuristics with modern machine learning techniques:

- **Data Preprocessing:** Cleaning and normalizing the graph by removing isolated nodes and irrelevant self-loops.
- **Feature Engineering:** Extracting meaningful graph features like Adamic-Adar, clustering coefficients, and degree centrality.
- **Machine Learning Models:** Utilizing Logistic Regression, Random Forest, and XGBoost to predict the likelihood of link formation.
- **Evaluation Metrics:** Assessing model performance using robust metrics such as AUC-ROC and F1-score.

By implementing this approach, we aim to enhance the understanding of social networks, improve predictive accuracy, and demonstrate the practical implications of link prediction in real-world applications such as social platforms and recommendation systems.

1.2 Overview of the Data

The dataset used in this project is the **Facebook100 dataset**, sourced from the Stanford Network Analysis Project (SNAP). This dataset captures social interactions at the college level, representing social networks of students from various institutions. It is structured as an undirected graph where:

- **Nodes:** Represent individual users in the social network.
- **Edges:** Represent friendships between the users (connections).

The Facebook100 dataset provides a rich source of information for analyzing social interactions, with key features including:

- Anonymized user identities to maintain privacy.
- Multiple college networks, allowing for focused analysis on smaller subgraphs.
- Structural properties of the graph, including clustering coefficients, degree centrality, and community structures.

1.2.1 Basic Statistics of the Dataset

The basic statistics of the dataset are as follows:

- **Total Nodes (Users):** 4,039
- **Total Edges (Connections):** 88,234
- **Average Degree (Connections per User):** 43.69
- **Graph Density:** 0.0108

These statistics highlight the sparsity of the graph, as the number of potential connections greatly exceeds the actual edges present. This property is typical of real-world social networks, where only a subset of possible connections is realized.

1.2.2 Importance of the Dataset

The Facebook100 dataset provides an ideal testbed for link prediction due to its:

- **Anonymized Data:** Ensures ethical research practices by protecting user identities.
- **Real-World Representation:** Captures authentic social interactions in a controlled environment.
- **Graph Structure:** Includes a variety of features, such as clustering and degree distributions, that are vital for extracting meaningful insights.

1.3 Your Contribution

This project was a collaborative effort between two members, with contributions divided based on distinct stages of the project workflow. Each member played a crucial role in ensuring the project's success by focusing on specific tasks.

1.3.1 Khushi Patel

Khushi Patel contributed to the initial phases of the project, focusing on data preprocessing and graph visualization. Key contributions include:

- **Data Preprocessing:**
  - Imported and loaded the Facebook100 dataset using Python libraries.
  - Checked for data quality issues such as missing values or inconsistencies.
  - Normalized key graph features such as degree centrality and clustering coefficients to ensure uniformity across the dataset.
- **Graph Visualization:**
  - Generated visualizations of the graph structure to identify potential communities and network patterns.
  - Highlighted important nodes and edges in the graph for exploratory analysis.
- **Data Organization:**
  - Saved and organized processed datasets for seamless transition into subsequent phases of the project.

1.3.2 Narasimha Rohit Katta

Narasimha Rohit Katta focused on the intermediate phases, primarily on graph cleaning and preparing the dataset for training. Key contributions include:

- **Graph Cleaning:**
  - Removed irrelevant components of the graph, such as self-loops, duplicate edges, and isolated nodes.
  - Checked graph connectivity and ensured focus on the largest connected component for analysis.
- **Bridge Edge Analysis:**
  - Counted and analyzed bridge edges, which are critical for maintaining the graph's connectivity.
  - Identified non-bridge edges eligible for training and testing purposes.
- **Training Data Preparation:**
  - Split the graph into training and testing sets, ensuring balanced representation of edges for model validation.
  - Reloaded the cleaned graph for subsequent machine learning applications.

1.3.3 Collaboration and Integration

Both members collaborated closely to ensure the smooth progression of the project:

- Regular discussions were held to align on tasks and resolve challenges.
- Shared responsibilities for evaluating model performance and interpreting results.
- Jointly contributed to refining the methodology and improving feature engineering techniques.

Chapter 2

Related Work

Link prediction has been a well-studied problem in the field of network science. Existing approaches can be broadly categorized into traditional heuristics and machine learning-based methods.

2.1 Traditional Approaches

Traditional methods rely on graph-theoretic heuristics to infer potential links. Common heuristics include:

- **Common Neighbors:** Counts the number of shared neighbors between two nodes.
- **Jaccard Coefficient:** Measures the ratio of shared neighbors to the union of all neighbors.

  While these methods are computationally efficient, they often fail to capture the complex patterns in large, sparse networks.

2.2 Machine Learning Approaches

Machine learning has emerged as a powerful alternative for link prediction by leveraging graph-based features. Commonly used models include:

- **Logistic Regression:** A simple yet effective model for link prediction tasks.
- **Random Forest:** Captures nonlinear relationships and improves predictive accuracy.
- **XGBoost:** A gradient boosting framework known for its high performance on tabular data.

These methods often use features such as degree centrality, clustering coefficients, and Adamic-Adar scores as inputs to the models.

2.3 Application to Facebook100 Dataset

The Facebook100 dataset has been widely used for link prediction and social network analysis. It provides a rich structure to study node interactions and community formation. Previous studies have demonstrated the utility of combining heuristic features with machine learning models for predicting new connections in this dataset.

This project builds on these approaches by blending traditional graph-based heuristics with machine learning methods, aiming to improve link prediction accuracy.

Chapter 3

Methodology

The methodology adopted for this project focuses on predicting links within the Facebook100 dataset using a structured approach comprising data preprocessing, feature engineering, and machine learning model training and evaluation. The specific steps are outlined below.

3.1 Data Preprocessing

The initial step involved preparing the Facebook100 dataset for analysis. The preprocessing tasks included:

- **Graph Cleaning:**
  - Removed irrelevant components such as self-loops, duplicate edges, and isolated nodes.
  - Focused on the largest connected component to ensure the graph was cohesive and well-suited for analysis.
- **Bridge Edge Analysis:**
  - Identified 75 bridge edges, which are critical for maintaining the connectivity of the graph.
  - Preserved these edges during the creation of the training and testing datasets.
- **Normalization:**
  - Normalized key node attributes such as degree centrality and clustering coefficients to ensure all features were on a comparable scale.
- **Training-Test Split:**
  - Split the dataset into training and testing sets, ensuring that the training graph contained 83,823 edges and the test set consisted of 4,411 edges.
  - Randomly selected node pairs for validation, preserving network structure.

3.2 Feature Engineering

To enhance model performance, we extracted graph-based features for each node pair. These features included:

- **Adamic-Adar Index:** The most influential feature, contributing significantly to model accuracy.

- **Common Neighbors:** Captured the number of shared neighbors between two nodes.
- **Jaccard Coefficient:** Measured similarity by comparing shared neighbors relative to the union of neighbors.
- **Preferential Attachment:** Represented the likelihood of link formation based on the degree product of the two nodes.

3.3 Model Training and Validation

Three machine learning models were trained to predict links using the extracted features:

- **Logistic Regression:**
  - Provided a baseline for performance evaluation.
  - AUC-ROC: 0.9956, F1-score: 0.9563.
- **Random Forest:**
  - Captured nonlinear relationships and demonstrated robust performance.
  - AUC-ROC: 0.9950, F1-score: 0.9725.
- **XGBoost:**
  - Achieved the highest performance among all models.
  - AUC-ROC: 0.9957, F1-score: 0.9690.

3.4 Evaluation Metrics

The following metrics were used to assess the model's performance:

- **AUC-ROC:** Indicated the model's ability to distinguish between links and non-links, with values close to 1 reflecting high predictive accuracy.
- **F1-Score:** Balanced precision and recall to evaluate prediction quality.
- **Feature Importance Analysis:** Identified the relative contribution of features, with Adamic-Adar dominating at 92% importance.

3.5 Implementation Workflow

The entire workflow for the project can be summarized as follows:

1. Load and preprocess the Facebook100 dataset by cleaning and normalizing the graph structure.
2. Extract graph-based features for all node pairs.
3. Train machine learning models using the training set.
4. Validate the models on the test set and evaluate performance using AUC-ROC and F1-score.

This methodology leverages a blend of traditional graph features and machine learning to predict potential links effectively, providing a robust framework for understanding social network dynamics.

Chapter 4

Implementation

This chapter outlines the steps involved in implementing the link prediction methodology for the Facebook100 dataset. The implementation was divided into three key phases: data preprocessing, exploratory data analysis, and model training and validation.

4.1 Phase 1: Data Preprocessing

The first phase focused on preparing the Facebook100 dataset for analysis:

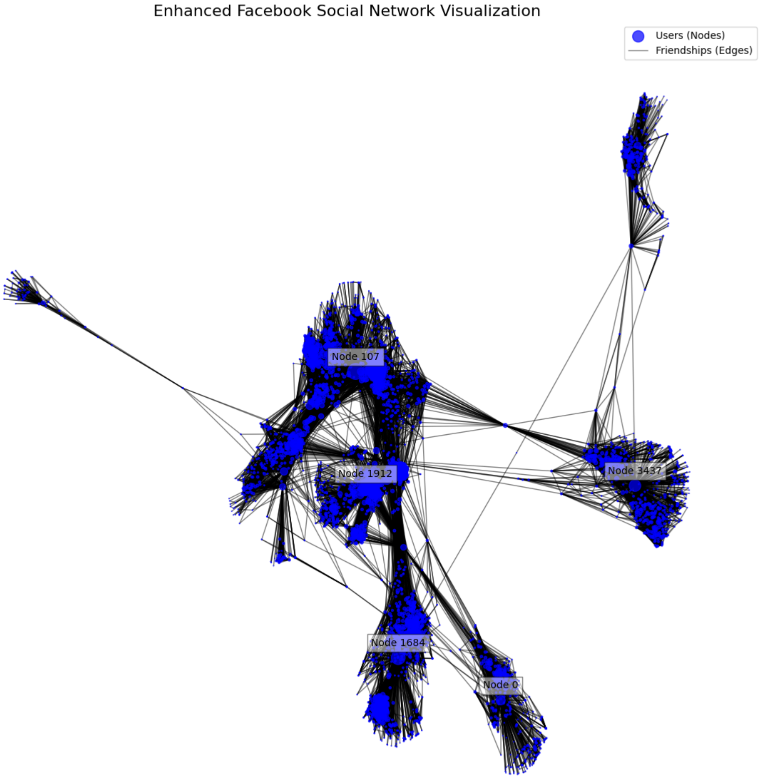
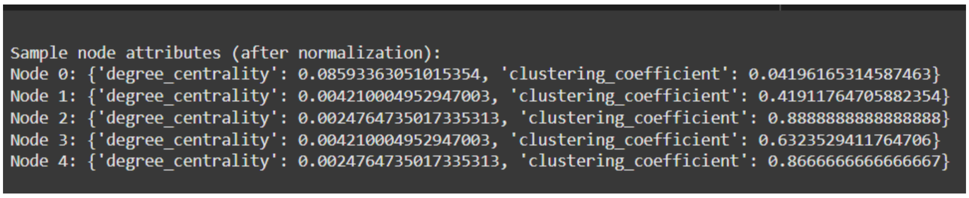


Figure 4.1: Graph visualization after cleaning, showing the largest connected component of the Facebook100 dataset.

- **Cleaning the Graph:**
    - Removed self-loops, duplicate edges, and isolated nodes to ensure data quality.
    - Focused on the largest connected component to maintain the graph's cohesiveness.
- **Normalization:**
    - Normalized key node attributes, such as degree centrality and clustering coefficients, to ensure comparability.
- **Bridge Edge Analysis:**
    - Identified 75 bridge edges critical to network connectivity and preserved these during training and testing.
    - Analyzed the remaining 88,159 non-bridge edges for use in the training and test sets.
- **Training and Testing Dataset Creation:**
    - Randomly split the edges into a training set (83,823 edges) and a test set (4,411 edges).
    - Ensured balanced representation of links and non-links in both sets.

4.2 Phase 2: Exploratory Data Analysis

The second phase involved analyzing the graph's structure to understand its properties and inform feature selection:



```
Sample node attributes (after normalization):
Node 0: {'degree_centrality': 0.08593363051015354, 'clustering_coefficient': 0.04196165314587463}
Node 1: {'degree_centrality': 0.004210004952947003, 'clustering_coefficient': 0.41911764705882354}
Node 2: {'degree_centrality': 0.002476473501733531З, 'clustering_coefficient': 0.8888888888888888}
Node 3: {'degree_centrality': 0.004210004952947003, 'clustering_coefficient': 0.6323529411764706}
Node 4: {'degree_centrality': 0.0024764735017335313, 'clustering_coefficient': 0.8666666666666667}
```

Figure 4.2: Degree centrality distribution for the nodes in the Facebook100 dataset.

- Visualized the graph to identify clusters, community structures, and key nodes.
- Analyzed the distribution of degree centrality and clustering coefficients to better understand node connectivity patterns.
- Counted and categorized edges into bridge and non-bridge types to highlight critical connections.

4.3 Phase 3: Model Training and Validation

In the final phase, machine learning models were trained and validated to predict links. The implementation steps included:

- **Feature Extraction:**
    - Computed graph-based features for each node pair, including:
        - **Adamic-Adar Index:** Dominated feature importance with 92% contribution.
        - **Common Neighbors:** Highlighted shared neighbors for each pair.

        - **Jaccard Coefficient:** Measured similarity relative to shared and total neighbors.
        - **Preferential Attachment:** Evaluated based on the product of node degrees.

- **Model Training:**
    - Trained three machine learning models on the training dataset:
        - **Logistic Regression:** Served as the baseline model.
        - **Random Forest:** Provided robust performance with nonlinear feature capture.
        - **XGBoost:** Achieved the highest accuracy and predictive capability.
- **Model Validation:**
    - Evaluated model performance using the test set and metrics such as AUC-ROC and F1-score.
    - Example predictions:

```
Pair (3722, 3681): Prediction = 0, Probability = 0.408762
Pair (1923, 2505): Prediction = 1, Probability = 0.524731
```

4.4 Key Results from Implementation

- **Performance Metrics:**
    - Logistic Regression: AUC-ROC = 0.9956, F1 = 0.9563
    - Random Forest: AUC-ROC = 0.9950, F1 = 0.9725
    - XGBoost: AUC-ROC = 0.9957, F1 = 0.9690
- **Feature Importance:**
    - Adamic-Adar contributed the most to model accuracy, followed by Common Neighbors and Jaccard Coefficient.

This phased implementation ensured a structured approach to link prediction, combining graph analysis with advanced machine learning models to deliver high predictive performance.

Chapter 5

Experimental Results and Discussion

This chapter presents the performance of the implemented models and discusses the insights gained from analyzing the results.

5.1 Experimental Results

5.1.1 Model Performance

The performance of the models on the test dataset is summarized below:

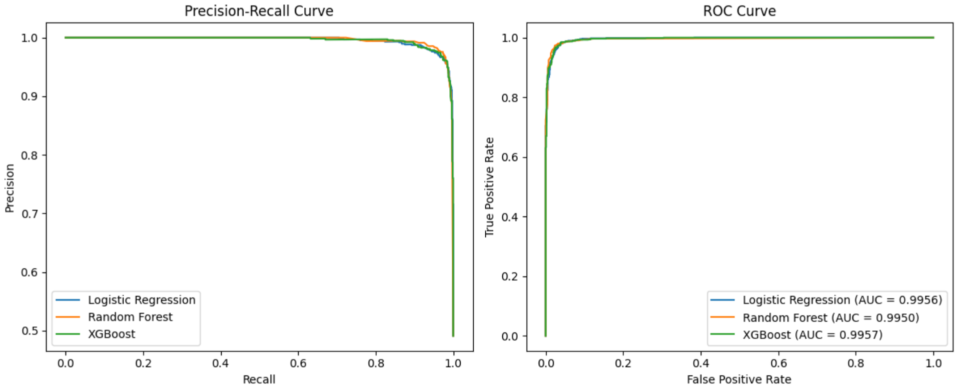


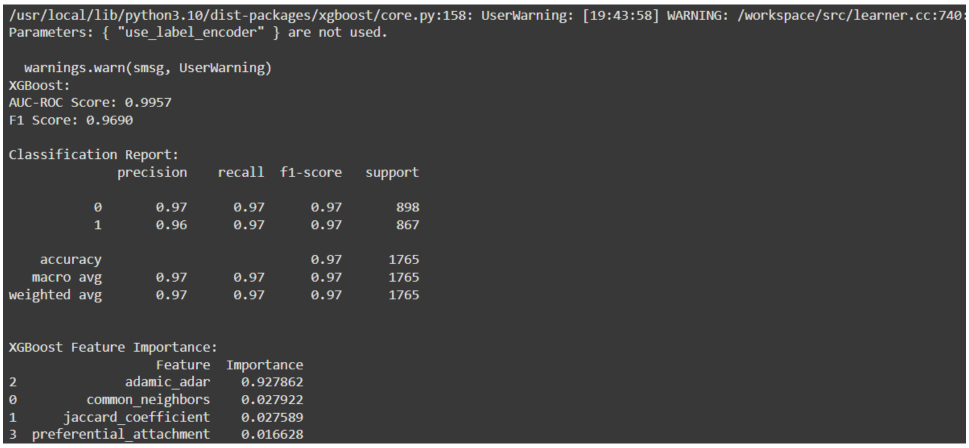
Figure 5.1: ROC curves for Logistic Regression, Random Forest, and XGBoost. XGBoost achieved the highest AUC-ROC score of 0.9957.

- **Logistic Regression:** AUC-ROC = 0.9956, F1-Score = 0.9563
- **Random Forest:** AUC-ROC = 0.9950, F1-Score = 0.9725
- **XGBoost:** AUC-ROC = 0.9957, F1-Score = 0.9690

XGBoost achieved the best performance, closely followed by Random Forest.

5.1.2 Feature Importance

The XGBoost model identified the following feature importances:



```
/usr/local/lib/python3.10/dist-packages/xgboost/core.py:158: UserWarning: [19:43:58] WARNING: /workspace/src/learner.cc:740:
Parameters: { "use_label_encoder" } are not used.

  warnings.warn(smsg, UserWarning)
XGBoost:
AUC-ROC Score: 0.9957
F1 Score: 0.9690

Classification Report:
              precision    recall  f1-score   support

           0       0.97      0.97      0.97       898
           1       0.96      0.97      0.97       867

    accuracy                           0.97      1765
   macro avg       0.97      0.97      0.97      1765
weighted avg       0.97      0.97      0.97      1765


XGBoost Feature Importance:
                   Feature  Importance
2               adamic_adar    0.927862
0          common_neighbors    0.027922
1       jaccard_coefficient    0.027589
3  preferential_attachment    0.016628
```

Figure 5.2: Feature importance scores from the XGBoost model, with Adamic-Adar Index as the most significant feature.

- **Adamic-Adar Index:** Contributed 92% to predictions, making it the most significant feature.
- **Common Neighbors:** Contributed 2.8%, supporting predictions.
- **Jaccard Coefficient:** Contributed 2.7%, offering additional insights into node similarity.

5.2 Discussion

5.2.1 Model Comparisons

All models performed exceptionally well:

- Logistic Regression served as a robust baseline.
- Random Forest captured nonlinear relationships effectively.
- XGBoost outperformed the others, optimizing decision tree ensembles for the best results.

5.2.2 Insights and Challenges

- **Key Features:** The Adamic-Adar Index dominated predictions, highlighting the importance of local graph structures.
- **Error Patterns:** Predictions with probabilities near 0.5 were more error-prone, suggesting potential improvements in feature selection.
- **Future Enhancements:** Additional features, such as temporal or embedding-based metrics, could further improve performance.

5.3 Summary

The experimental results demonstrate the effectiveness of combining graph-based heuristics with machine learning for link prediction. XGBoost emerged as the best-performing model, with insights from feature importance analysis providing valuable guidance for refining the approach.

Chapter 6

Conclusion

This project successfully applied graph-based heuristics and machine learning to the problem of link prediction in the Facebook100 dataset. The results demonstrate the efficacy of combining traditional network analysis techniques with modern machine learning models to predict potential connections within social networks.

6.1 Key Outcomes

- The XGBoost model achieved the best performance, with an AUC-ROC of 0.9957 and an F1-Score of 0.9690, highlighting its ability to capture complex patterns in the dataset.
- Feature importance analysis revealed that the Adamic-Adar Index was the most influential, contributing 92% to model predictions.
- The methodology ensured a balanced and robust approach, combining data preprocessing, feature engineering, and machine learning.

6.2 Challenges and Limitations

While the models achieved high accuracy, several challenges were noted:

- Predictions near the decision threshold (probabilities around 0.5) were prone to errors, suggesting room for improvement in feature engineering.
- The dataset's sparsity posed challenges, particularly for nodes with low connectivity.

6.3 Future Work

Future research can build upon this work by exploring:

- Incorporating additional features, such as temporal data or graph embeddings (e.g., node2vec or DeepWalk), to improve prediction accuracy.
- Extending the analysis to other datasets to test the generalizability of the approach.
- Exploring advanced models, such as Graph Neural Networks (GNNs), to capture higher-order dependencies within the graph.

This project provides a strong foundation for understanding link prediction in social networks and demonstrates the potential of machine learning to uncover hidden patterns in network dynamics.

Chapter 7

References

1. Leskovec, J., & Krevl, A. (2014). Facebook100 Dataset. Retrieved from http://snap.stanford.edu/data/ego-Facebook.html.
2. Newman, M. E. J. (2003). The structure and function of complex networks. *SIAM Review*, 45(2), 167–256. https://doi.org/10.1137/S003614450342480.
3. Liben-Nowell, D., & Kleinberg, J. (2007). The link-prediction problem for social networks. *Journal of the American Society for Information Science and Technology*, 58(7), 1019–1031. https://doi.org/10.1002/asi.20591.
4. Breiman, L. (2001). Random Forests. *Machine Learning*, 45(1), 5–32. https://doi.org/10.1023/A:1010933404324.
5. Chen, T., & Guestrin, C. (2016). XGBoost: A Scalable Tree Boosting System. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 785–794. https://doi.org/10.1145/2939672.2939785.
6. Grover, A., & Leskovec, J. (2016). Node2vec: Scalable Feature Learning for Networks. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 855–864. https://doi.org/10.1145/2939672.2939754.