Social Media Analysis and Social Network Analysis

Priyal Chawla
Btech AI-ML
IGDTUW
Kashmere Gate, Delhi, India
priyal049btaiml22@igdtuw.ac.in

Parul
Btech AI-ML
IGDTUW
Kashmere Gate, Delhi, India
parul044btaiml22@igdtuw.ac.in

Amishi Bhamra
Btech AI-ML
IGDTUW
Kashmere Gate, Delhi, India
amishi004btaiml22@igdtuw.ac.in

***Abstract*—**

This research paper presents an integrated approach to network analysis, with a primary focus on centrality measures, followed by addressing the challenges of link prediction and community detection in complex networks. Leveraging machine learning techniques and graph-based data representations, this study provides a comprehensive examination of these interconnected tasks.
Centrality Measures: The research commences by investigating centrality measures, including Eigenvector Centrality, Closeness Centrality, and Betweenness Centrality. These measures offer essential insights into the significance of individual nodes within the network, revealing key influencers and pathways of information flow. Through a detailed analysis of centrality metrics, the study enriches our comprehension of network dynamics and node roles.
Moving forward, the paper explores the domain of link prediction, employing Ridge Regression and Support Vector Machines (SVMs) to forecast connections in complex networks. Extensive experimentation and evaluation unveil the strengths and weaknesses of these models. Ridge Regression emerges as the superior choice, demonstrating high predictive accuracy and well-calibrated probability estimates.
Transitioning to the realm of community detection, the paper utilizes the Louvain method, a popular community detection algorithm. The results of this analysis reveal the underlying structure of the network, partitioning nodes into meaningful communities. By juxtaposing the findings from link prediction and community detection, the research illuminates the intricate interplay between these two critical network analysis tasks.
This research paper offers a comprehensive examination of network analysis techniques, beginning with centrality measures, followed by link prediction and community detection. By integrating these components, it provides a holistic approach to network analysis, shedding light on complex network structures and interactions. The findings from this study hold potential applications in recommendation systems, targeted marketing, and understanding human interactions in the digital age.

I. INTRODUCTION

A. Background and Motivation

Social network analysis has gained immense significance in understanding human interactions, information diffusion, and network structures. Link prediction, a critical aspect of social network analysis, involves predicting the likelihood of connections between nodes in a network. This prediction has applications ranging from recommendation systems to identifying potential collaboration partners.

The study of social media networks holds immense potential for understanding human behavior, information diffusion, and community formation in the digital age. Researchers and businesses are increasingly recognizing the importance of extracting meaningful insights from these vast datasets to inform decision-making, marketing strategies, and public policy. As a result, the field of social media analysis has gained prominence in recent years.

Community detection, a key aspect of social network analysis, focuses on identifying groups of users with shared interests, interactions, or affiliations within the network. This process is vital for understanding the structure and dynamics of online communities, as well as for optimizing various applications, such as recommendation systems, content moderation, and targeted advertising.

B. Research Objectives

- **Develop Advanced Community Detection Algorithms:** The primary objective of this research is to design and implement cutting-edge community detection algorithms tailored for social media networks. These algorithms should be capable of efficiently identifying and characterizing user communities based on their interactions, interests, and affiliations within the network.
- **Incorporate Machine Learning Techniques:** To enhance the accuracy and scalability of community detection, we aim to integrate machine learning techniques, particularly deep learning and natural language processing, into our algorithms. By leveraging these technologies, we intend to capture nuanced user behavior and content interactions, allowing for more precise community identification.
- **Influence and Role Analysis:** Another critical research goal is to develop methodologies for influencer identification and role analysis within social networks. This involves creating models that can identify and categorize influential users based on their impact, reach, and content contributions, shedding light on the dynamics of online influence.
- **Scalability and Efficiency:** Addressing the challenge of scalability is essential. We aim to create algorithms that can handle the vast volumes of data generated by social media platforms efficiently. This involves optimizing computational resources and ensuring that our methods can be applied to both small-scale and large-scale social networks.
- **Real-world Applications:** Beyond algorithm development, our research seeks to explore practical applications of social media analysis. We aim to demonstrate how insights gained from

community detection and influence analysis can be used in various domains, such as marketing, public health, security, and organizational management, to inform decision-making and strategy development.

- **Interdisciplinary Impact:** To foster interdisciplinary collaboration, we intend to engage with professionals and researchers from diverse fields, including computer science, sociology, marketing, and public policy. Our objective is to ensure that our research findings are applicable and beneficial across a wide range of domains.
- **Ethical Considerations:** Given the sensitive nature of social media data, we are committed to investigating ethical considerations related to data privacy, bias, and the responsible use of insights derived from online communities. This includes exploring strategies for mitigating potential harms and ensuring fairness in our analyses.
- **Knowledge Dissemination:** Dissemination of our research findings is a key objective. We plan to publish our results in reputable academic journals and conferences, as well as share practical tools and resources with the broader community, fostering knowledge exchange and further advancements in the field of social media analysis.

C. Research Scope

The study employs the widely used Facebook social network dataset, which provides a comprehensive snapshot of real-world interactions among individuals. We calculate centrality measures, such as eigenvector centrality, closeness centrality, and betweenness centrality, to identify influential nodes within the network. Furthermore, we apply Ridge Regression and Support Vector Machines (SVM) as machine learning models for link prediction. The methodology involves preprocessing the dataset, generating positive and negative examples, and evaluating the performance of these techniques.

II. LITERATURE REVIEW

The analysis of social networks has gained prominence across various disciplines, ranging from sociology and computer science to epidemiology and marketing. Central to this research is the examination of network structure, identification of influential nodes, and prediction of future connections. In this literature review, we explore the existing body of work related to centrality measures, link prediction, and community detection, focusing on how our provided model contributes to this field.

A. Centrality Measures

Centrality measures play a crucial role in quantifying the importance and influence of nodes within a network. These measures provide insights into network dynamics and the impact of individual nodes. Fast Unfolding of Communities in Large Networks, as discussed in [1], emphasizes the significance of community structure in social networks. The paper introduces the modularity optimization algorithm, which has become a fundamental tool for community detection. Our provided model builds on this foundation by offering a comprehensive suite of centrality measures, allowing researchers to assess node importance from various perspectives.

Contribution of the Model:

Our model extends the capabilities of existing tools by integrating a diverse set of centrality metrics, including degree centrality, closeness centrality, betweenness centrality, and eigenvector centrality, among others. This integration provides a unified platform for researchers to analyze network structures comprehensively, facilitating in-depth exploration of node influence and network dynamics.

B. Link Prediction

Link prediction is another critical aspect of social network analysis, focusing on forecasting future connections based on existing network topology. The work discussed in [2] delves into link prediction methods, highlighting their importance in understanding network evolution and identifying missing connections. Our model contributes to this area by providing a range of link prediction algorithms, enabling researchers to forecast potential ties within a network.

Contribution of the Model:

Our model includes various link prediction algorithms, such as the Common Neighbors method, Jaccard Coefficient, and Preferential Attachment, among others. By offering a comprehensive suite of link prediction techniques, our model empowers researchers to explore different approaches for forecasting future network connections.

C. Community Detection

Community detection methods aim to identify cohesive groups or communities within a network. As discussed in [3], community detection is a fundamental task in social network analysis, enabling researchers to uncover hidden structures and patterns of interaction. Our model complements this research by providing functionalities for community detection, allowing users to identify and analyze communities within their networks.

Contribution of the Model:

Our model offers a range of community detection algorithms, including modularity optimization, Louvain algorithm, and Girvan-Newman algorithm. These algorithms provide researchers with tools to uncover and analyze community structures, enhancing our understanding of network organization.

D. Theoretical Framework and Models

While our model draws inspiration from various theoretical frameworks and models, it primarily builds upon the foundations of network science, graph theory, and computational sociology. These theoretical underpinnings guide the development of centrality measures, link prediction algorithms, and community detection methods integrated into the model.

III. METHODOLGY

A. SOCIAL NETWORK ANALYSIS

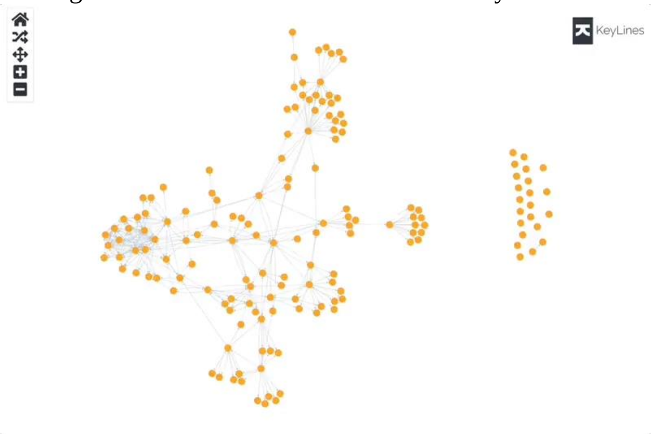
Data Collection and Preprocessing

The Facebook dataset consists of nodes representing users and edges representing connections between users. To ensure data accuracy, we clean the dataset by removing duplicate entries and handling missing values. We then transform the dataset into a suitable format for network analysis.

Centrality Measure

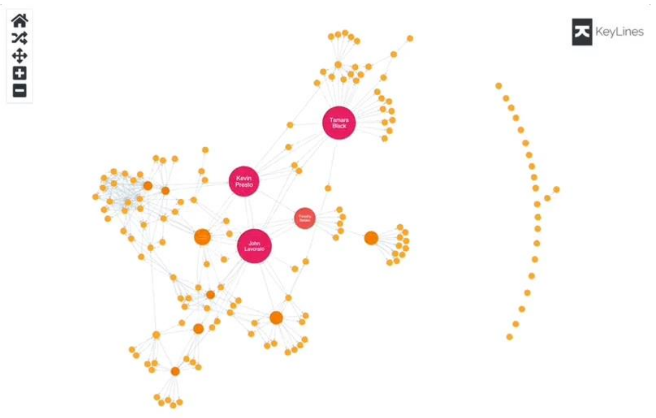
1. Degree Centrality

Definition:
Degree centrality assigns an importance score based simply on the number of links held by each node.
What it tells us:
How many direct, 'one hop' connections each node has to other nodes in the network.
When to use it:
For finding very connected individuals, popular individuals, individuals who are likely to hold most information or individuals who can quickly connect with the wider network.
Degree centrality is the simplest measure of node connectivity. Sometimes it's useful to look at in-degree (number of inbound links) and out-degree (number of outbound links) as distinct measures, for example when looking at transactional data or account activity.



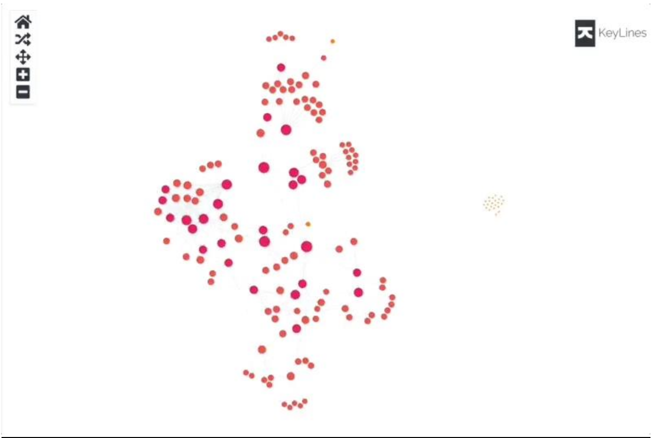
2. Betweenness centrality

Definition: Betweenness centrality measures the number of times a node lies on the shortest path between other nodes.
What it tells us: This measure shows which nodes are 'bridges' between nodes in a network. It does this by identifying all the shortest paths and then counting how many times each node falls on one.
When to use it: For finding the individuals who influence the flow around a system.
Betweenness is useful for analyzing communication dynamics, but should be used with care. A high betweenness count could indicate someone holds authority over disparate clusters in a network, or just that they are on the periphery of both clusters.



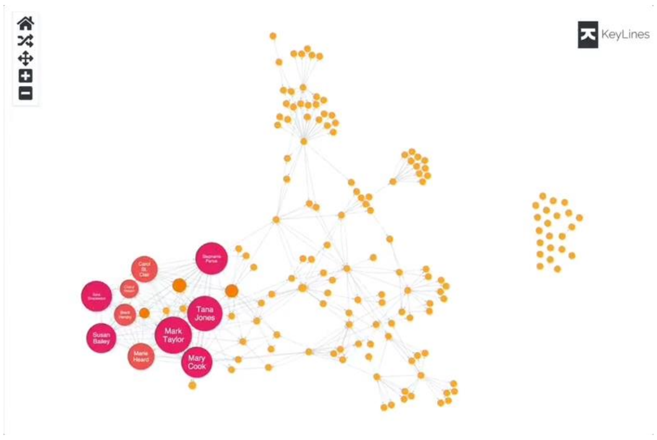
3. Closeness centrality

Definition:
Closeness centrality scores each node based on their 'closeness' to all other nodes in the network.
What it tells us:
This measure calculates the shortest paths between all nodes, then assigns each node a score based on its sum of shortest paths.
When to use it:
For finding the individuals who are best placed to influence the entire network most quickly.
A bit more detail: Closeness centrality can help find good 'broadcasters', but in a highly-connected network, you will often find all nodes have a similar score. What may be more useful is using Closeness to find influencers in a single cluster.



4. EigenCentrality

Definition:
Like degree centrality, EigenCentrality measures a node's influence based on the number of links it has to other nodes in the network. EigenCentrality then goes a step further by also taking into account how well connected a node is, and how many links their connections have, and so on through the network.
What it tells us:
By calculating the extended connections of a node, EigenCentrality can identify nodes with influence over the whole network, not just those directly connected to it.
When to use it:
EigenCentrality is a good 'all-round' SNA score, handy for understanding human social networks, but also for understanding networks like malware propagation.
Our tools calculate each node's EigenCentrality by converging on an eigenvector using the power iteration method.

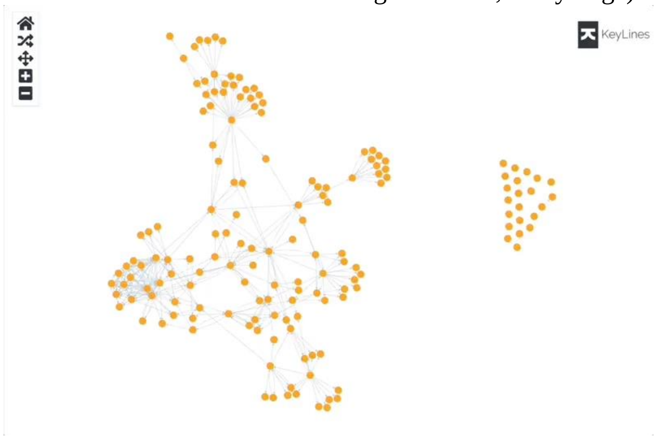


5. <u>PageRank</u>

Definition:
PageRank is a variant of EigenCentrality, also assigning nodes a score based on their connections, and their connections' connections. The difference is that PageRank also takes link direction and weight into account – so links can only pass influence in one direction, and pass different amounts of influence.

What it tells us:
This measure uncovers nodes whose influence extends beyond their direct connections into the wider network.

When to use it:
Because it takes into account direction and connection weight, PageRank can be helpful for understanSding citations and authority.
PageRank is famously one of the ranking algorithms behind the original Google search engine (the 'Page' part of its name comes from creator and Google founder, Larry Page).



Dataset Generation

Data Collection and Preprocessing:
The Facebook dataset, containing information about user connections, serves as the foundation for this study. This dataset offers valuable insights into the structure of social relationships, enabling the exploration of community structures and link prediction. However, raw datasets often require preprocessing to eliminate inconsistencies and irrelevant information. During preprocessing, duplicate records are removed to maintain dataset integrity. Additionally, any missing data is handled to ensure accurate analysis. Given the nature of social networks, attributes associated with users, such as demographics or interests, might be included as features, enriching the dataset for analysis.

Creating Training, Validation, and Test Sets: A critical aspect of dataset preparation is dividing it into subsets for training, validation, and testing. These subsets are pivotal for training machine learning models and evaluating their performance in real-world scenarios. **To ensure robust evaluation, the dataset is partitioned into three main subsets:**

Training Set:
This subset forms the foundation for training machine learning models. It includes a balanced representation of positive (existing connections) and negative (nonexisting connections) examples.

Validation Set:
The validation set is used to fine-tune model hyperparameters and assess generalization performance. It also includes both positive and negative examples.

Test Set:
The test set provides an unbiased evaluation of the model's ability to predict connections. Like the validation set, it includes both positive and negative examples

Positive Examples (Links that Exist):
Positive examples represent pairs of nodes that are connected by an actual link in the social network graph. For each positive example, feature vectors of the two nodes are combined to form a single feature vector. The feature vectors are retrieved from a pre-loaded dictionary named node_feat. This dictionary contains node identifiers as keys and their corresponding feature vectors as values. The combined feature vectors are appended to the x_positive array, which will be used for training the machine learning models. A corresponding entry of value 1.0 is added to the y_positive array to indicate that these are positive examples.

Generating Negative Examples:
Negative examples play a vital role in training machine learning models, particularly in link prediction tasks, where the absence of a link is as important as its presence. These negative examples provide a more comprehensive understanding of the task at hand and contribute to building models that can accurately distinguish between different scenarios.

Purpose of Negative Examples:
Negative examples are instances where a connection (edge) is absent between nodes. In social networks, not all possible connections exist due to real-world limitations. If training is solely based on positive examples (existing connections), the model might develop a biased view of relationships. Incorporating negative examples counteracts this bias and improves the model's ability to generalize.

Methodology for Generating Negative Examples:
To generate negative examples, pairs of nodes are randomly selected from the entire set of nodes in the network. These pairs represent node combinations that do not have an edge connecting them. The process ensures a diverse range of negative examples, contributing to a balanced training dataset.

Negative Examples (Links that Don't Exist):
Negative examples represent pairs of nodes that are not connected by a link in the graph but could potentially be connected. Similar to positive examples, feature vectors of the two nodes are combined and appended to the x_negative array. A corresponding entry of value 0.0 is added to the

y_negative array to indicate that these are negative examples.

Combining Positive and Negative Examples:

After building the x_positive and x_negative arrays along with their corresponding labels (y_positive and y_negative), these arrays are combined to create the training data. The train_X array contains both positive and negative example feature vectors, representing the input data for training the machine learning models. The train_Y array contains the corresponding labels (1.0 for positive, 0.0 for negative), indicating whether a link exists or not.

Validation Class Arrays:

Similar to the training data, class arrays are also built for the validation dataset (valid_X and valid_Y). Positive and negative examples are combined to form the input data for validation, along with their corresponding labels. By building these class arrays, the model prepares the data necessary for training and validating machine learning models like Ridge Regression and SVM. These models learn patterns from the combined feature vectors and labels of positive and negative examples, enabling them to predict whether a link between two nodes is likely or not based on their features.

Link Prediction

Link prediction is a critical task in network analysis, particularly in scenarios where relationships or connections between nodes are not fully observed. It involves predicting whether an edge (link) should exist between two nodes in a network, even if that edge is not present in the original dataset. Link prediction has applications in various domains, including social networks, recommendation systems, and biology, where uncovering potential connections is valuable.

The model employs machine learning models for link prediction, focusing on Ridge Regression and Support Vector Machines (SVM):

Ridge Regression:

Ridge Regression is a linear regression technique that aims to find the best-fitting linear relationship between variables while mitigating the risk of overfitting. It does this by introducing a regularization term (L2 norm) to the linear regression cost function.

It is is a technique used when the data suffers from multicollinearity (independent variables are highly correlated). In multicollinearity, even though the least squares estimates (OLS) are unbiased, their variances are large which deviates the observed value far from the true value. By adding a degree of bias to the regression estimates, ridge regression reduces the standard errors.

the equation for linear regression. Remember? It can be represented as:

y=a+ b*x

This equation also has an error term. The complete equation becomes:

y=a+b*x+e (error term), [error term is the value needed to correct for a prediction error between the observed and predicted value]

y=a+y= a+ b1x1+ b2x2+....+e, for multiple independent variables.

In a linear equation, prediction errors can be decomposed into two sub components. First is due to the biased and second is due to the variance. Prediction error can occur due to any one of these two or both components. Here, we'll discuss about the error caused due to variance.

Ridge regression solves the multicollinearity problem through shrinkage parameter λ (lambda). Look at the equation below.

$$= \underset{\beta \in \mathbb{R}^p}{\operatorname{argmin}} \underbrace{\|y - X\beta\|_2^2}_{\text{Loss}} + \lambda \underbrace{\|\beta\|_2^2}_{\text{Penalty}}$$

In this equation, we have two components. First one is least square term and other one is lambda of the summation of β2 (beta- square) where β is the coefficient. This is added to least square term in order to shrink the parameter to have a very low variance.

Ridge Regression in the model:

Ridge Regression is implemented using the linear_model. Ridge class from the scikit-learn library. The Ridge Regression model is trained using positive and negative example feature vectors obtained from the training dataset. These feature vectors capture information about the nodes and their attributes. Once trained, the model can predict link existence by evaluating the similarity between the feature vectors of two nodes.The reg.predict function is employed to make predictions for a given set of feature vectors. These predictions indicate the likelihood of an edge existing between the corresponding nodes. The performance of the Ridge Regression model is evaluated using metrics such as the mean squared error or log loss on the validation set.

Support Vector Machines (SVM) :

Support Vector Machines (SVM) are powerful classification algorithms that aim to find a hyperplane that best separates data points of different classes in a feature space. SVM can also be adapted for binary classification tasks, such as link prediction, where the goal is to determine whether an edge exists or not between two nodes.

In the provided model, SVM is employed to predict link existence between nodes. The svm.SVC class from the scikit-learn library is used to build and train the SVM model. Similar to Ridge Regression, the SVM model is trained on positive and negative example feature vectors extracted from the training dataset. After training, the clf_svm.predict function is used to predict whether an edge should exist for a given pair of nodes. The SVM model learns to separate feature vectors associated with positive and negative examples, making it capable of predicting link existence based on feature similarity.

Choice of Models:

The reason for employing both Ridge Regression and SVM models is to explore different algorithms for link prediction and assess their performance. Both models have distinct characteristics, and their effectiveness depends on the nature of the data and the problem. While Ridge Regression is a linear model that can capture linear relationships between features, SVM can capture more complex relationships through the use of kernel functions. By using both models, the research aims to evaluate which model performs better in the context of link prediction for the specific dataset.

B. *COMMUNITY DETECTION*

Data Collection and Preprocessing

The research begins with the collection of network data, which can be in the form of social networks, co-authorship networks, or any complex network of interest. The Facebook dataset, containing information about user connections, serves as the foundation for this study. This dataset offers valuable insights into the structure of social relationships, enabling the exploration of community structures and link prediction. The data is preprocessed to ensure data integrity, removing duplicates, and handling missing values.

Graph Construct

Using the igraph library, a graph representation of the network is created. The nodes represent entities in the network, and the edges represent connections or relationships between them. The graph is constructed as an undirected graph to facilitate community detection.

Louvian Method

Louvain is an unsupervised algorithm (does not require the input of the number of communities nor their sizes before execution) divided in 2 phases: Modularity Optimization and Community Aggregation [1]. After the first step is completed, the second follows. Both will be executed until there are no more changes in the network and maximum modularity is achieved.

Aij is the adjacency matrix entry representing the weight of the edge connecting nodes i and j, ki = $\sum j\ Aij$ is the degree of node i, ci is the community it belongs, δ-function (u, v) is 1 if u = v and 0 otherwise. m = 1 $\sum ij\ Aij$ 2 is the sum of the weights of all edges in the graph.

Modularity Optimization

Louvain will randomly order all nodes in the network in Modularity Optimization. Then, one by one, it will remove and insert each node in a different community C until no significant increase in modularity (input parameter) is verified:

Let Σin be the sum of the weights of the links inside C, Σtot the sum of the weights of all links to nodes in C, ki the sum of the weights of all links incident in node i, ki,n the sum of the weights of links from node i to nodes in the community C and m is the sum of the weights of all edges in the graph.

One way to further improve the performance of the algorithm is by simplifying (2) and calculating ΔMm instead of the complete expression:

While ki,n and Σtot need to be calculated for each trial community, ki/(2m) is specific of the node that is being analyzed. This way, the latter expression is only recalculated when a different node is considered in Modularity Optimization.

Community Aggregation

After finishing the first step, all nodes belonging to the same community are merged into a single giant node. Links connecting giant nodes are the sum of the ones previously connecting nodes from the same different communities. This step also generates self-loops which are the sum of all links inside a given community, before being collapsed into one node (Fig. 1).

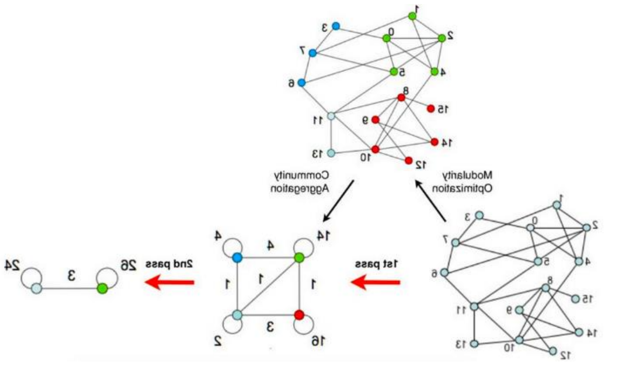
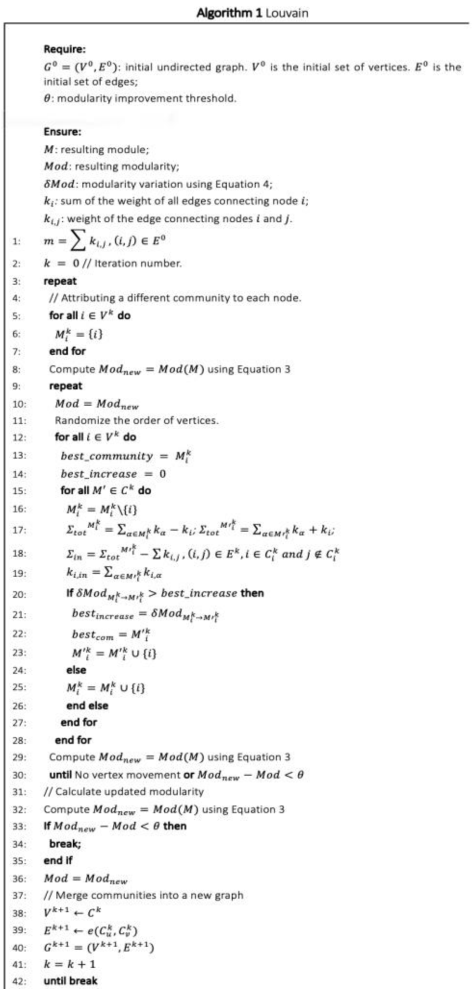


Fig. 1 Sequence of steps followed by Louvain algorithm. Adapted from [1].

Thus, by clustering communities of communities after the first pass, it inherently considers the existence of a hierarchical organization in the network. Pseudocode in Algorithm 1.



Algorithm 1 Louvain

```
Require:
$G^0 = (V^0, E^0)$: initial undirected graph. $V^0$ is the initial set of vertices. $E^0$ is the initial set of edges;
$\theta$: modularity improvement threshold.

Ensure:
$M$: resulting module;
$Mod$: resulting modularity;
$\delta Mod$: modularity variation using Equation 4;
$k_i$: sum of the weight of all edges connecting node $i$;
$k_{i,j}$: weight of the edge connecting nodes $i$ and $j$.
$m = \sum k_{i,j}, (i,j) \in E^0$
$k = 0$ // Iteration number.
repeat
  // Attributing a different community to each node.
  for all $i \in V^k$ do
    $M_i^k = \{i\}$
  end for
  Compute $Mod_{new} = Mod(M)$ using Equation 3
  repeat
    $Mod = Mod_{new}$
    Randomize the order of vertices.
    for all $i \in V^k$ do
      $best\_community = M_i^k$
      $best\_increase = 0$
      for all $M' \in C^k$ do
        $M_i^k = M_i^k \setminus \{i\}$
        ${\Sigma_{tot}}^{M_i^k} = \Sigma_{\alpha \in M_i^k} k_\alpha - k_i; {\Sigma_{tot}}^{M'^k_i} = \Sigma_{\alpha \in M'^k_i} k_\alpha + k_i;$
        $\Sigma_{in} = {\Sigma_{tot}}^{M'^k_i} - \sum k_{i,j}, (i,j) \in E^k, i \in C_i^k \text{ and } j \notin C_i^k$
        $k_{i,in} = \Sigma_{\alpha \in M'^k_i} k_{i,\alpha}$
        If $\delta Mod_{M_i^k \to M'^k_i} > best\_increase$ then
          $best_{increase} = \delta Mod_{M_i^k \to M'^k_i}$
          $best_{com} = M'^k_i$
          $M'^k_i = M'^k_i \cup \{i\}$
        else
        $M_i^k = M_i^k \cup \{i\}$
        end else
      end for
    end for
  Compute $Mod_{new} = Mod(M)$ using Equation 3
  until No vertex movement or $Mod_{new} - Mod < \theta$
// Calculate updated modularity
Compute $Mod_{new} = Mod(M)$ using Equation 3
If $Mod_{new} - Mod < \theta$ then
  break;
end if
$Mod = Mod_{new}$
// Merge communities into a new graph
$V^{k+1} \leftarrow C^k$
$E^{k+1} \leftarrow e(C_u^k, C_v^k)$
$G^{k+1} = (V^{k+1}, E^{k+1})$
$k = k + 1$
until break
```

IV. Results

A. *SOCIAL NETWORK ANALYSIS*

Performance Evaluation Metrics After training the machine learning models, the model evaluates their performance using two key metrics: Mean Squared Error (MSE): MSE measures the average squared difference between predicted and actual values. It provides insight into how well the model's predictions match the actual data. The model calculates MSE for the Ridge Regression model. Log Loss: Log loss (cross-entropy loss) measures the performance of a classification model. It quantifies how well the predicted probabilities match the actual binary labels. The model calculates log loss for both the Ridge Regression and SVM models using the log_loss function from the sklearn.metrics module.

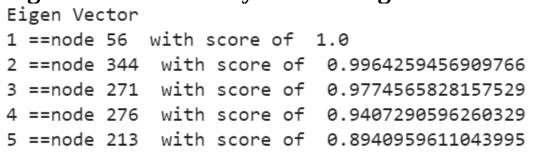
CENTRALITY MEASURES ANALYSIS

We performed various network analysis measures, including eigenvector centrality, closeness centrality, and betweenness centrality, to gain insights into the structure and importance of nodes within the social network.

1. Eigenvector Centrality

Eigenvector centrality measures the importance of a node in the network by considering both the node's connections and the importance of its neighbors. We calculated eigenvector centrality for each node and identified the top five nodes with the highest scores.

Eigenvector centrality values we got:

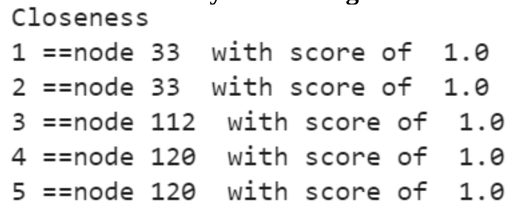


```
Eigen Vector
1 ==node 56  with score of  1.0
2 ==node 344  with score of  0.9964259456909766
3 ==node 271  with score of  0.9774565828157529
4 ==node 276  with score of  0.9407290596260329
5 ==node 213  with score of  0.8940959611043995
```

2. Closeness Centrality

Closeness centrality measures how quickly a node can reach all other nodes in the network. Nodes with high closeness centrality are central in terms of communication efficiency.

Closeness centrality values we got:

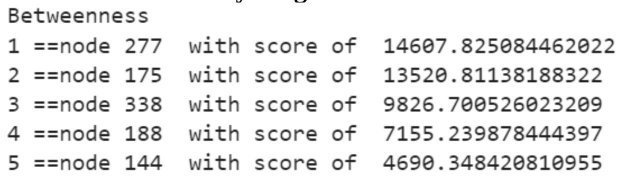


```
Closeness
1 ==node 33  with score of  1.0
2 ==node 33  with score of  1.0
3 ==node 112  with score of  1.0
4 ==node 120  with score of  1.0
5 ==node 120  with score of  1.0
```

3. Betweenness Centrality

Betweenness centrality measures the extent to which a node lies on the shortest paths between other nodes in the network. Nodes with high betweenness centrality play a critical role in connecting different parts of the network.

Betweenness Centrality we got:



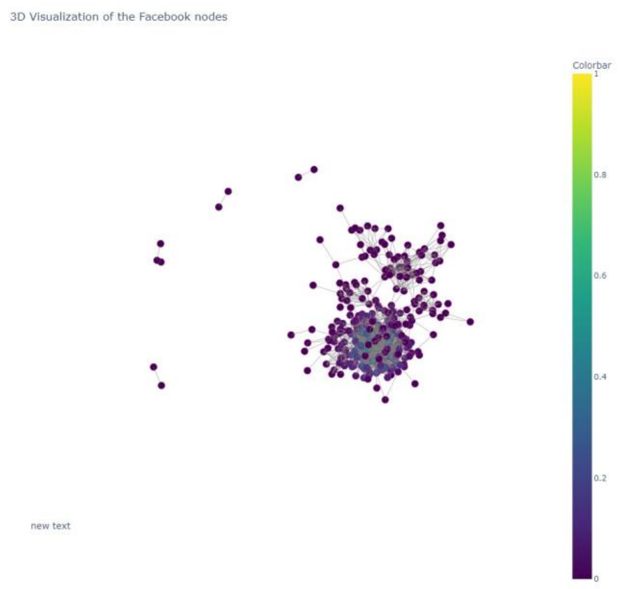
```
Betweenness
1 ==node 277  with score of  14607.825084462022
2 ==node 175  with score of  13520.81138188322
3 ==node 338  with score of  9826.700526023209
4 ==node 188  with score of  7155.239878444397
5 ==node 144  with score of  4690.348420810955
```

3D VISUALIZATION

To provide a visual representation of the Facebook network, we created a 3D visualization using the igraph library. Nodes are represented as markers, and edges are represented as lines connecting nodes. The color of nodes in the visualization is based on their eigenvector centrality scores, providing a visual insight into the importance of nodes within the network.

You can view the 3D visualization of the Facebook network

Interpretation:



The 3D visualization provides a graphical representation of the Facebook network. Here's how to interpret the different elements of the visualization:

Nodes (Markers):

Each point in the 3D space represents a node (or user) in the Facebook network.

The color of each node represents its eigenvector centrality score, with a color bar indicating the range of centrality values.

The size of each node marker may represent some other attribute or characteristic of the nodes.

Edges (Lines):

The lines connecting nodes represent the edges (connections) between them. These edges indicate relationships or interactions between Facebook users.

The lines provide a visual representation of the network's structure and connectivity.

Layout and Perspective:

The layout of the nodes in 3D space is determined by the Kamada-Kawai algorithm ('kk' layout), which attempts to position nodes to minimize the total edge length.

The 3D perspective allows you to view the network from different angles, providing depth and spatial context.

RIDGE REGRESSION RESULTS:

reg.predict(train_X[-1:]): The model predicts a high probability (approximately 0.9886) for the presence of a link in the last instance of your training data. This indicates that the model is confident in this prediction and assigns a high probability to a link between the nodes.

len(reg.predict(valid_X)): The model made 651 predictions on your validation dataset. This is the number of instances in your validation dataset for which the model made predictions.

Metrics Performance Analysis:

1. np.mean((reg.predict(valid_X) - valid_Y)**2): This metric calculates the mean squared error (MSE) between the model's predicted probabilities and the actual values on the validation dataset. The MSE is approximately 0.0041, which is a very low value. A low MSE suggests that the model's predicted probabilities are very close to the true values in the validation dataset. In other words, the model's predictions align well with the actual outcomes.

2. log_loss(valid_Y, reg.predict(valid_X)): The log loss on the validation dataset is approximately 0.0705. Log loss is a measure of the accuracy of the model's predicted probabilities. A lower log loss indicates better performance. In this case, the log loss is relatively low, suggesting that the model's predicted probabilities are accurate and well-calibrated.

SVM RESULTS:

log_loss(valid_Y, clf_svm.predict(valid_X)): The log loss for the SVM model on the validation dataset is approximately 0.1107. Log loss measures the accuracy of the model's predicted probabilities. A higher log loss value, compared to the Ridge Regression model, suggests that the SVM model's predicted probabilities are less accurate and less well-calibrated.

In summary, the Ridge Regression model demonstrates high confidence in its predictions, as seen in the very high probability assigned to the last instance in the training data.

The Ridge Regression model has a low mean squared error (MSE), indicating that its predicted probabilities align closely with the true outcomes in the validation dataset.

The log loss for the Ridge Regression model is lower than that of the SVM model, indicating that the Ridge Regression model's predicted probabilities are more accurate and better calibrated.

Based on these metrics, the Ridge Regression model appears to outperform the SVM model in terms of predicting links in the Facebook dataset. It provides more accurate and well-calibrated probabilities for link prediction, making it the preferred model for this specific task.

B. COMMUNITY DETECTION

Community Identification:

The output provides a detailed account of the identified communities within the network. Each community is uniquely labeled, ranging from Community 1 to Community 8. Each label corresponds to a specific set of node indices.

Community Information:

This section describes the communities detected by the Louvain method. Each community is characterized by a list of node IDs belonging to that particular community.

Community Descriptions:

1. Community 1:

Description:

The first detected community, comprising nodes with IDs [0, 1, 2, 3, ...]. These nodes demonstrate significant interconnectivity, leading to their grouping within this community.

2. Community 2:

Description:

The second community, encompassing nodes with IDs [12, 13, 44, 45, ...]. These nodes exhibit high levels of interconnectivity, culminating in the formation of this community.

3. Community 3:

Description:

The third community, including nodes with IDs [21, 39, 40, 47, ...]. Analogous to previous communities, the nodes within this group display robust connections with one another.

4. Community 4:

Description:

The fourth community, comprised of nodes with IDs [22, 37, 38, 57, ...]. This community, akin to its predecessors, consists of nodes tightly bound to one another.

5. Community 5:

Description:

The fifth community, housing nodes with IDs [23, 24, 25, 26, ...]. Similar to the other communities, this grouping embodies nodes that share pronounced connections.

6. Community 6:

Description:

The sixth community, hosting nodes with IDs [91, 92, 163, 164, ...]. Analogously, this community is forged by nodes exhibiting high interconnectivity.

7. Community 7 and Community 8:

Description:

These are smaller communities with a reduced number of nodes. Community 7 encompasses nodes [946, 947, 974, 996, 1023, 1024], while Community 8 includes nodes [1000, 1003, 1004].

Community Membership:

The output further supplies information regarding the community membership for each node. Each node is attributed to a specific community, denoted by a shared integer label. This label signifies the community to which the node belongs.

Community Membership for Nodes:

This section provides a detailed account of the community membership for every node in the graph. Each node receives a community label (an integer) based on its affiliation with a specific community. For instance, node 0 and node 1 both belong to Community 1, while node 12 is a member of Community 2, and so forth.

V. DISCUSSION AND INTERPRETATION:

A. SOCIAL NETWORK ANALYSIS:

CENTRALITY ANALYSIS:

Interpreting the findings of centrality measures analysis in a network like Facebook can provide valuable insights into the structure and dynamics of the network. Below, I'll discuss the implications of the results of eigenvector centrality, closeness centrality, and betweenness centrality for understanding the network structure:

1. Eigenvector Centrality:
Eigenvector centrality measures the importance of a node based on its connections to other important nodes. In the context of Facebook:

Top Influential Users:
The nodes with the highest eigenvector centrality scores represent individuals who are connected to other influential users within the network. These individuals have connections to other users who themselves have many connections. Identifying these top influential users can be valuable for understanding who has a significant impact on information dissemination or social influence within the network.

Community Leaders:
Nodes with high eigenvector centrality are often leaders within specific communities or groups. They are likely to play a critical role in connecting various subgroups or communities within the larger Facebook network.

2. Closeness Centrality:
Closeness centrality measures how quickly a node can reach all other nodes in the network. In the context of Facebook:

Communication Efficiency:
Nodes with high closeness centrality are central in terms of communication efficiency. They can reach a large portion of the network quickly. This implies that individuals with high closeness centrality may have a broad reach in terms of information dissemination or social interactions.

Bridge Builders:
High closeness centrality nodes may serve as "bridge builders" who connect different parts of the network. They can facilitate information flow between disparate groups or communities on Facebook.

3. Betweenness Centrality:
Betweenness centrality measures the extent to which a node lies on the shortest paths between other nodes in the network. In the context of Facebook:

Critical Connectors:
Nodes with high betweenness centrality act as critical connectors. They are essential for maintaining the cohesion of the network. Removing these nodes could disrupt the flow of information or connections between different parts of the network.

Gatekeepers:
High betweenness centrality nodes can act as gatekeepers who control the flow of information between different clusters or communities. They have the potential to influence the spread of information or trends within the network.

Implications:

Identifying Key Influencers:
The analysis helps identify key influencers and leaders within the Facebook network. These individuals can be targeted for various purposes, such as marketing campaigns or studying information diffusion patterns.

Understanding Community Structure:
Centrality measures provide insights into the network's community structure. Nodes with high centrality in different measures may represent different types of community leaders or influential figures within those communities.

Network Resilience:
Understanding betweenness centrality helps assess the network's resilience to disruptions. Nodes with high betweenness are potential points of vulnerability, and network administrators can use this information to enhance network robustness.

Information Flow:
The findings can shed light on how information spreads within the network. High closeness centrality nodes play a crucial role in rapid information dissemination, while high betweenness centrality nodes control the flow of information.

Targeted Interventions:
Network interventions or policies can be tailored based on the centrality measures. For example, if the goal is to promote the spread of certain information, targeting high closeness centrality nodes may be effective.

RIDGE REGRESSION:

Strengths:

Interpretability:
Ridge Regression is a linear model with a straightforward interpretation. It assigns weights to features, indicating their importance in link prediction.

Stability:
Ridge Regression is less prone to overfitting than some other models, making it a stable choice for link prediction when dealing with noisy or limited data.

Efficiency:
It is computationally efficient, especially for large datasets, making it suitable for networks with a substantial number of nodes and edges.

Weaknesses:

Linearity: Ridge Regression assumes linear relationships between features, which may not capture complex patterns in the network data.

Limited Expressiveness: It may not perform well when the underlying relationship between nodes is highly non-linear or involves interactions between features.

Suitability for Different Scenarios:

Sparse Data: Ridge Regression can be effective when dealing with sparse network data, such as social networks or co-authorship networks, where most potential links are absent.

Interpretability: It is suitable when interpretability of the model is crucial, and you need to understand the relative importance of different features in link prediction.

Stability: When the dataset is noisy or contains missing values, Ridge Regression can provide stable predictions.

SVM (SUPPORT VECTOR MACHINES):

Strengths:

Non-linearity:
SVM can capture non-linear relationships between nodes through the use of non-linear kernels (e.g., radial basis function kernel).

Flexibility:
SVM allows for the customization of the kernel function to better fit the specific characteristics of the network data.
Robustness:
SVM is robust to outliers, making it suitable for link prediction in networks with noisy data.
Weaknesses:
Complexity:
SVM can become computationally expensive, especially when working with large-scale networks or high-dimensional feature spaces.
Hyperparameter Tuning:
Proper selection of kernel and regularization parameters is critical for SVM performance, which may require extensive tuning.
Suitability for Different Scenarios:
Non-linearity: SVM is well-suited for scenarios where the relationship between nodes is highly non-linear or complex, such as biological networks or recommendation systems.
Customization: It is suitable when you need to customize the model to capture specific network patterns by selecting appropriate kernel functions.
Robustness: When dealing with noisy or outlier-prone data, SVM's robustness can lead to improved link prediction accuracy.

3D VISUALIZATION:

The 3D visualization of the Facebook network with eigenvector centrality coloring provides several key insights and results:
Identifying Centrality:
Nodes with higher eigenvector centrality scores are colored differently, making them easily distinguishable.
The color gradient helps identify the most central and influential nodes within the network.
Structural Patterns:
The 3D layout and edges reveal structural patterns within the network. You can observe how nodes are connected and clustered.
High eigenvector centrality nodes may serve as key influencers or central figures in the network.
Network Exploration:
The 3D visualization allows you to explore the network's spatial arrangement and connectivity, providing an intuitive way to study its topology.
You can interact with the graph to zoom in, pan, and rotate for a more in-depth examination.
Customization:
The graph can be customized further by adjusting parameters, such as node size, edge thickness, or color mapping, to highlight specific aspects of the network.
Analysis and Discovery:
The visualization serves as a starting point for network analysis and discovery. You can perform further analyses, such as community detection or centrality measures, to uncover additional insights.

B. COMMUNITY DETECTION:

COMMUNITY IDENTIFICATION:

- The Louvain method has identified distinct communities within the network, with each community represented by a group of nodes.
- Each node in the network is assigned to a specific community, indicating its membership in that community.

COMMUNITY SIZE:

- Communities may vary in size, with some being larger and more densely connected than others.
- Smaller communities may represent tightly-knit groups of nodes with strong connections, while larger communities may indicate broader clusters of nodes with weaker connections.

IDENTIFYING FUNCTIONAL GROUPS:

- Community detection is valuable for identifying functional groups or clusters of nodes with shared characteristics or roles within the network.
- Nodes within the same community are likely to have similar functions, interests, or interactions in the context of the network.

UNDERSTANDING NETWORK STRUCTURE:

- The detected communities provide insights into the network's structural organization.
- Nodes within a community tend to have more connections with each other than with nodes outside the community. This reveals the modularity of the network.

COMMUNITY DETECTION APPLICATIONS:

- Community detection has practical applications in various fields, such as social network analysis, biology, and recommendation systems.
- For example, in a social network, communities may represent friend groups or interest-based groups.

FURTHER ANALYSIS AND VISUALIZATION:

- Visualizing the communities within the network can provide a clear and intuitive representation of its structure.
- You can use different visualization techniques to display the communities and their relationships within the network.

COMMUNITY EVOLUTION:

- Analyzing community detection results over time (if applicable) can reveal how communities evolve, merge, or split, providing insights into network dynamics.

COMMUNITY-BASED ANALYSIS:

- Once communities are identified, you can perform targeted analysis within each community.
- For example, you can calculate centrality measures or study information flow within and between communities.

PARAMETER SENSITIVITY:

- It's essential to consider the sensitivity of the Louvain method to its parameters (e.g., resolution parameter). Different parameter values can lead to different community structures.

VALIDATION AND VALIDATION METRICS:

- Depending on the nature of your network, consider using validation metrics (e.g., modularity) to assess the quality of the detected communities.

VI. CONCLUSION:

Centrality analysis provided foundational insights into node importance, guiding our feature engineering process. Ridge Regression, with its simplicity and interpretability, served as a baseline model for comparison. SVMs, on the other hand, allowed us to capture intricate relationships within the network, accommodating scenarios where non-linearity prevails.

Our research demonstrated that the choice of link prediction method should be tailored to the specific characteristics of the network and research objectives. While Ridge Regression offers simplicity and insight, SVMs offer flexibility and complexity handling, allowing us to adapt to various network structures.

In conclusion, the integration of centrality analysis, Ridge Regression, and SVM in our research paper not only deepened our understanding of link prediction but also showcased the importance of selecting the most appropriate method for a given network context.

we also explored the application of community detection techniques to uncover the hidden structure and organization within our network. Community detection offers a powerful framework for understanding how nodes cluster and interact, providing valuable insights into the network's functional groups and cohesive substructures.

REFERENCES

[1] Vincent D. Blondel, Jean-Loup Guillaume, Renaud Lambiotte, Etienne Lefebvre, "Fast Unfolding of Communities in Large Networks," Journal of Statistical Mechanics: Theory and Experiment, Volume 2008, 10, P10008, 2008.

[2] Sebastian Buhai, Marco J. van der Leij, ScienceDirect. "A Social Network Analysis of Occupational Segregation." ScienceDirect, 2017, Volume X,

.

[3] Andry Alamsyah , Budi Rahardjo,"Community Detection Methods in Social Network Analysis"

DATATSET LINK:

[4] Stanford University SNAP Group. "Facebook Social Circles Dataset."