Assignment 1

Facebook Social Network Analysis: Developing a Network Recommendation System

Xavier Felixan MIJAATA (57195713)
Jawad MAHMUD (57274243)
Pongamorn TRAKARNKULPHUN (57587131)

City University of Hong Kong
SDSC3016: Social Network Analysis
Prof. KE Qing
10 October 2024

1. Introduction

The human being naturally has always been a social creature, seeking relationships and interactions with others (Flynn, 2008). Throughout history, the need for social activities has encouraged humankind to develop tools that allow connections between individuals, even worldwide. Ever since the digitalization era, social media has become one of the most important inventions that connected human lives. The rapid advancement of social media platforms has brought people ease and convenience in interacting and doing social activities.

Through social media platforms, users form interesting networks that can be analyzed to understand the flow of information and social preferences. New users come and go, building links with people they want to connect with. Individuals around the world collectively generate a complex, constantly evolving, system of nodes and links. The dynamic nature of these networks has led to numerous innovative methods for exploring human behaviours online, opening doors for application and research opportunities. Hence, understanding these digital networks has become extremely important for multiple industries around the world.

In this report, a thorough exploration was done on the "Facebook Users" dataset to gain insights into how a social media network behaves. Analyzing the patterns and properties behind the graph network will not only allow a comprehensive interpretation of the users and links, but also the development of the network recommendation system.

a. Background

With around 3 billion active users monthly, Facebook is one of the most used social media platforms worldwide, serving as a crucial hub for information sharing (Backlinko, 2024). The platform offers a wide range of features that satisfy various customer needs, including communication channels, content sharing, status updates, community engagements, etc.

In this community, users are being followed and are following other individuals, creating a huge graph data. It presents the window to generally understand social network, user importance, and ultimately, train models for link predictions.

b. Objectives

The key objective of this project is to develop a machine learning model capable of giving recommendations. By implementing statistical techniques, exploratory data analysis, and machine learning methods, this project aims to fulfil these objectives:

1. Understand the graph properties of a social network
2. Check if the network is small world
3. Design a link prediction model that recommends connections between users

We aim to utilize the “Facebook Users” dataset to detect key features in a social network and build a network recommendation system to suggest possible links in the network. This model, if implemented to an application, would allow a better user experience as well as open business opportunities for the social media platform to grow.

2. Methodology

a. Dataset

The dataset “Facebook Users” is an anonymized directed social graph data. It was taken from Kaggle, containing almost 2 million nodes and more than 9 million edges overall. The data was provided by Facebook company itself in the occasion of a recruiting competition organized 12 years ago.

b. Data Preprocessing

i. Data Sampling

Because of the size of the network and our computational power limitations, we sampled the dataset. The sample size is 50000 nodes and 241276 edges. According to Leskovec, J., and Faloutsos, C. (2006), among various graph sampling methods used in the paper, random walk sampling is the best at keeping the properties of the original graph. In this paper, we would like to preserve as many properties of the network as possible. Thus, the method we chose for data sampling was random walk sampling.

First, we randomly choose a node in the network uniformly as the starting node. Then, we perform random walk to the neighbour node such that the probability of visiting each neighbour node is equal. At each step, there is a probability of 0.15 that a visited node is chosen as the next node with equal probability. We repeat the random walk until the number of visited nodes is equal to what we consider, which in this case is 50000. Finally, to prevent the case where the random walk is stuck, we count how many iterations has the algorithm performed. If it is too large, we start the whole sampling process over again.

ii. Labelling

In this study, link prediction is considered as a classification problem. Hence, target variable is a necessity in training supervised learning models. This was done by labelling each observation so the model could predict and recommend edges.

Firstly, a column “Exist” was made as the target variable. The labelling was executed by assigning 1 to existing observations in the sample.

To balance the number of target variables, a set of edges was randomized by using existing nodes in the dataset. These imaginary edges were then labelled as 0, meaning that they did not actually exist.

After computing and concatenating the same number of edges as the initial sample, the size of the dataset doubled. Now, half of the dataset consisted of existing links, and the other half was non-existing links.

iii. Feature Engineering

The current dataset only consisted of the source and target nodes, along with the target variables. With only these two nodes, it would not provide any significant insights into the target variables. Thus, feature engineering was introduced to produce predictors that might help in predicting the response variable. Through this process, an array of features was calculated from the directed graph's overall properties.

1. Shortest Path

   The shortest path is the least number of links needed to go from the source node to the target node. Since half of the dataset was already existing edges, the shortest path would surely be 1. In that case, this edge was ignored, and the second shortest path were calculated. If no paths were found due to the network being da directed graph, -1 would be assigned.

2. Follow Back

   Follow back is a property where a user who is being followed by another user, tends to follow back that user, completing a two-way connection.

3. Preferential Attachment

   Preferential attachment indicates that a user who has more links is inclined to make more connections in the future. The score is calculated by multiplying the number of source node's successors and target node's predecessors.

4. Cosine Similarity

   Cosine similarity represents how similar a user is to another user based on their neighbours. The neighbours were

treated as a vector, and the similarity was calculated by measuring the angle between those two vectors of neighbours. The smaller the angle between the vectors, the more similar those two users are.

5. Jaccard Similarity

   Jaccard similarity is the proportion of shared connections to the overall connections of two different users. It is calculated by dividing the number of common neighbours by the number of total neighbours between the two users.

6. Eigenvector Centrality

   Eigenvector centrality measures a user's importance according to the importance of its predecessors. The score assigned to a node is based on the concept that connections to high-scoring nodes contribute more to the initial node than having equal connections with low-scoring nodes. In this directed graph, a node who has higher importance, meaning higher scores, would have high-scoring nodes pointing to that node in question.

7. PageRank

   PageRank is an algorithm that represents a user's importance based on random walking on the graph. It is a variant of the eigenvector centrality, where a node is considered important if they have high quality in-degree.

8. Hyperlink-Induced Topic Search

   Hyperlink-Induced Topic Search (HITS) algorithm calculates two values for each user node based on their roles as hubs and authorities. Hub is a node that links to many authorities, and authority is a node that many hubs link to.

9. Resource Allocation Index

   Resource allocation index calculates link value relevant to the common neighbours between two users, weighted accordingly to differentiate celebrity and normal links. The purpose was to balance out common neighbours, where if two

users followed a user with high degree (celebrity), compared to following a common user, where it could be more useful.

10. Link Weight

    Link weight determines the weight of the link going in and out of a user proportioned with the degree. The concept was derived from when a user has only 10 links, then every link would intuitively be more meaningful, compared to a user who has 100 links.

The feature engineering step was completed. The columns containing source and target node id were removed from the dataset as they were not necessary anymore. They were not meaningful for the link prediction.

iv. Dimension Reduction

Following the feature engineering, each observation had a total of 22 predictors. These predictors may or may not contribute to the response variable. To achieve a simpler model, dimension reduction was needed. Hence, the predictors were standardized first, and they were passed through PCA. The result had 15 dimensions, and it was ready to train the classification models.

c. Exploratory Data Analysis

i. Correlation

To determine whether the dataset for the network can yield any significant insights through conventional statistical methods, exploratory data analysis was carried out on the 22 attributes prior to using the dataset for training the classification model. Specifically, a correlation analysis was performed to determine which attributes have weak, strong, positive, and negative correlations with one another. To visualize the outcome of the correlation study, a correlation heatmap was employed.

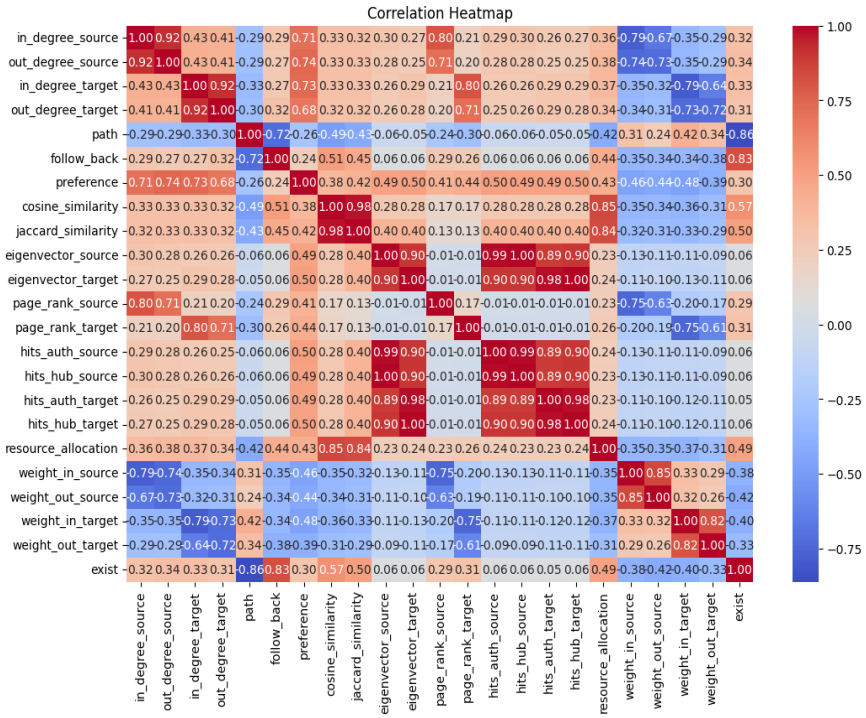

Figure 1: Correlation Heatmap of Attributes

Based on the results of the correlation heatmap, 35 pairs of attributes have strong positive correlations meaning that as the value of one attribute increases significantly the value of the other attribute also increases significantly and 14 pairs of attributes have strong negative correlations with each other meaning that as the value of one attribute decreases significantly the value of the other attribute also decreases significantly.

Attribute 1	Attribute 2	Correlation
eigenvector_target	hits_hub_target	0.999904657
eigenvector_source	hits_hub_source	0.999903295
hits_auth_source	hits_hub_source	0.987558184
eigenvector_source	hits_auth_source	0.987508229
eigenvector_target	hits_auth_target	0.98196066
hits_auth_target	hits_hub_target	0.98174965
cosine_similarity	jaccard_similarity	0.977308385
in_degree_source	out_degree_source	0.923917264
in_degree_target	out_degree_target	0.918386535
hits_hub_source	hits_hub_target	0.904161923
eigenvector_source	hits_hub_target	0.903799571
eigenvector_target	hits_hub_source	0.903735836
eigenvector_source	eigenvector_target	0.903369206
hits_auth_source	hits_hub_target	0.901060806
eigenvector_target	hits_auth_source	0.900462102
hits_auth_target	hits_hub_source	0.890692789
eigenvector_source	hits_auth_target	0.890352873
hits_auth_source	hits_auth_target	0.887371131
cosine_similarity	resource_allocation	0.853850867
weight_in_source	weight_out_source	0.850474982
jaccard_similarity	resource_allocation	0.835831214
exist	follow_back	0.831774705
weight_in_target	weight_out_target	0.816494492
in_degree_target	page_rank_target	0.804959071
in_degree_source	page_rank_source	0.804009837
out_degree_source	preference	0.744528391
in_degree_target	preference	0.733743488
out_degree_source	page_rank_source	0.71142185
out_degree_target	page_rank_target	0.70933469
in_degree_source	preference	0.706346281
out_degree_target	preference	0.684808003
cosine_similarity	exist	0.572064814
cosine_similarity	follow_back	0.51320782
exist	jaccard_similarity	0.501237801
hits_auth_source	preference	0.501220118

Figure 2: Table for Attributes with Strong Positive Correlations

A correlation coefficient close to 1 indicates a strong positive correlation, meaning that as one attribute increases, the other tends to

increase as well. Conversely, a coefficient near -1 signifies a strong negative correlation, where one attribute's increase corresponds to the decrease of the other. A coefficient close to 0 suggests little to no linear relationship between the attributes.

Starting with the strongest positive correlations, the attributes eigenvector_target and hits_hub_target exhibit an extremely high correlation coefficient of 0.999905. This value implies that these attributes move almost identically, suggesting a strong relationship in their behaviors or values. A similar pattern is observed with eigenvector_source and hits_hub_source, which display a correlation coefficient of 0.999903, reinforcing the notion of a near-perfect positive relationship between them.

Moving on to other notable strong positive correlations, hits_auth_source and hits_hub_source demonstrate a robust correlation coefficient of 0.987558. This correlation indicates a substantial positive relationship between the hits authorities and hits hubs of the source attribute. Moreover, eigenvector_source and hits_auth_source display a correlation coefficient of 0.987508, suggesting a strong positive association between the eigenvector centrality of the source and its hits authorities.

Transitioning to moderately positive correlations, attributes such as in_degree_source and out_degree_source exhibit a correlation coefficient of 0.923917, indicating a moderate positive relationship between the in-degree and out-degree attributes of the source node. Similarly, in_degree_target and out_degree_target showcase a correlation coefficient of 0.918387, pointing towards a similar moderate positive correlation between the corresponding attributes of the target node.

Exploring additional correlations, the attributes cosine_similarity and jaccard_similarity manifest a strong positive correlation with a coefficient of 0.977308. This relationship implies that changes in cosine similarity are closely aligned with alterations in Jaccard similarity within the dataset. Moreover, the correlation between eigenvector_source and hits_auth_target with a coefficient of 0.890353 indicates a moderate positive relationship between the eigenvector centrality of the source and the hits authorities of the target attribute.

Attribute 1	Attribute 2	Correlation
exist	path	-0.861070074
in_degree_target	weight_in_target	-0.788501157
in_degree_source	weight_in_source	-0.787947223
page_rank_source	weight_in_source	-0.751033152
page_rank_target	weight_in_target	-0.749522798
out_degree_source	weight_in_source	-0.735590417
out_degree_target	weight_in_target	-0.72895695
out_degree_source	weight_out_source	-0.725087375
out_degree_target	weight_out_target	-0.716900845
follow_back	path	-0.716427554
in_degree_source	weight_out_source	-0.67296332
in_degree_target	weight_out_target	-0.644822944
page_rank_source	weight_out_source	-0.62719723
page_rank_target	weight_out_target	-0.611085907

Figure 3: Table for Attributes with Strong Negative Correlations

A negative correlation near -1 implies an inverse relationship, where an increase in one attribute is associated with a decrease in the other attribute and vice versa. Understanding these negative correlations is crucial for uncovering how changes in one attribute may influence another in an opposite direction, providing valuable insights into the underlying dynamics of the dataset.

Among the most striking negative correlations is the relationship between exist and path with a coefficient of -0.861070. This strong negative correlation implies that the presence of a relationship (exist) is inversely related to the length of the path between entities (path), suggesting that as the existence of a relationship increases, the path length decreases significantly. This insight can be pivotal in understanding the network structure and connectivity within the dataset.

Moving on to attributes with moderate to strong negative correlations, pairs such as in_degree_target and weight_in_target, as well as in_degree_source and weight_in_source, both exhibit coefficients around -0.78. These correlations signify an inverse relationship between the in-degree of the target/source node and its corresponding weight in attribute, indicating that as the in-degree increases, the weight tends to decrease and

vice versa. This relationship sheds light on how certain node properties are intertwined within the network.

Further exploration reveals negative correlations between attributes like out_degree_source and weight_in_source, along with out_degree_target and weight_in_target, displaying an inverse relationship between the out-degree of nodes and their respective weight in attributes. These negative associations highlight how certain structural characteristics of nodes are linked to their inbound weights within the network topology, offering insights into information flow and node importance.

ii. Degree and Weight Distribution

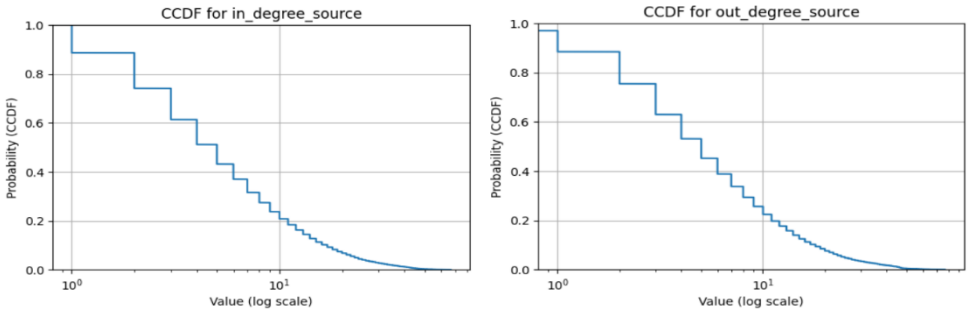

Figure 4: CCDF of links going in and out of user

Figure 4 shows the cumulative distribution function of links entering and leaving a node in log-log scale. The results indicate that both distributions follow a power law distribution. As the number of nodes in the network increases, the probability of having more than 10 edges entering and leaving the source node is slightly above 0.2.

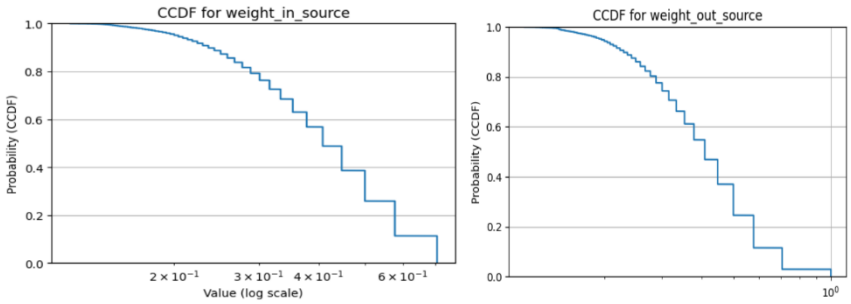

Figure 5: CCDF of weights of links going in and out of user

Figure 5 shows the complementary cumulative distribution function of edge weights connected to a node in log-log scale. It reveals a power law distribution indicating that as the network's source nodes increase, the likelihood of edge weights surpassing 0.4 and 10 for incoming and outgoing edges, respectively, approaches 0.

iii. Data Visualization

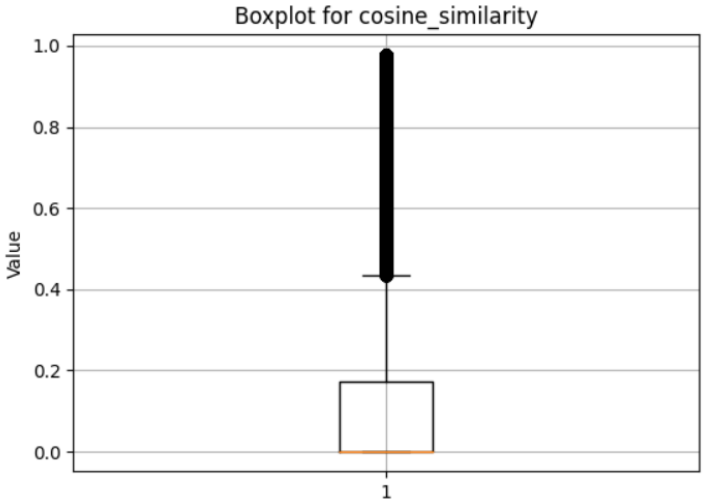

Figure 8: Boxplot for Cosine similarity of users

Figure 8 shows a boxplot depicting the cosine similarity of network nodes, with a median cosine similarity of nodes at 0 and an interquartile range varying from 0 to 0.18. The skewness of the cosine similarity distribution indicates a notable presence of outliers, suggesting that the majority of node pairs in the network exhibit a cosine similarity above the median value. This indicates that most, if not all, nodes have a significantly large number of shared neighbours above the average of their degrees.

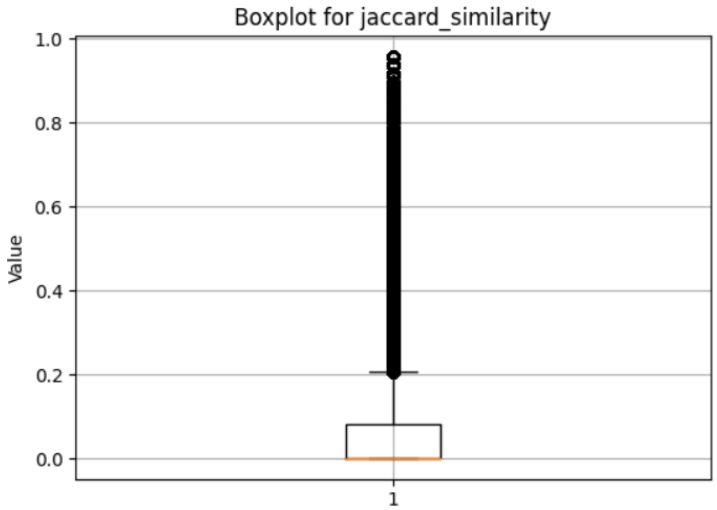

Figure 9: Boxplot for Jaccard similarity of users

In Figure 9, a boxplot displays the Jaccard similarity of nodes in the network. The median Jaccard similarity value of nodes is 0, with an interquartile range of cosine similarity values ranging from 0 to 0.15. The Jaccard similarity distribution is skewed to the right with many outliers, indicating that the cosine similarity of nodes in the network is generally higher than the median. This indicates that most, if not all, nodes have many common neighbours.

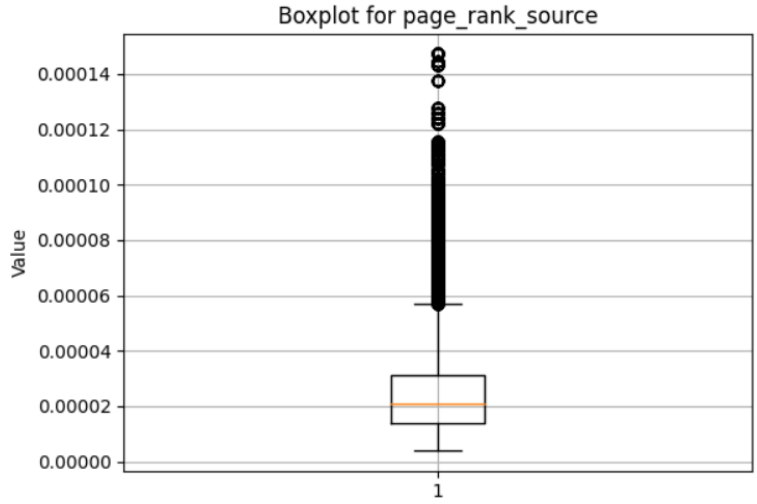

Figure 10: Boxplot for page rank score of users

Figure 10 shows boxplots depicting the distribution of page rank scores of nodes in the network. The plot reveals a positive skewness in the page rank scores, with a considerable number of outliers. This indicates that

most nodes in the network have page rank scores higher than the median value of around 0.00002, highlighting the significant importance of most nodes within the network.

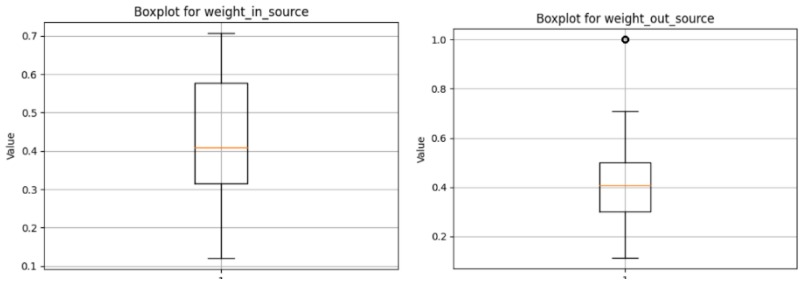

Figure 11: Boxplot for weights of links going in and out of source users

Figure 11 shows boxplots of how the weight distribution differs for edges coming into and going out from each node. According to the data, the weights of edges going into nodes are skewed positively, while the weights of edges coming out of nodes are skewed negatively. The interquartile range of weights for edges entering source nodes ranges from 0.32 to 0.57, compared to 0.3 to 0.5 for target nodes. The median weight for edges going in and out of nodes is approximately 0.42, and there are few outliers present in the data.

iv. Small World

In this paper, we define small world property of a network as having lower average shortest path length and higher clustering coefficient compared to a random graph of same size. For small world analysis, we use the original network before random walk sampling.

Even though the network is directed since the network is a social network. We assume that the network is indeed undirected. Furthermore, since the network is disconnected, but most nodes are connected, we used largest connected component of the network for the small world analysis.

To check if the network is small world, we first approximated average shortest path length and clustering coefficient of the network. Then, we generated random network using Maslov-Sneppen method and approximate average shortest path length and clustering coefficient of that

random network. The interested network's clustering coefficient is approximately 0.2, while random network's clustering coefficient is approximately 0.05. The ratio of the interested network to random network's clustering coefficient is 4. The interested network's average shortest path length is approximately 7.4713, while random network's average shortest path length is approximately 79.8135. Therefore, the network is indeed small world.

d. Link Prediction

To predict the link, we defined this problem as a classification problem using the features we got from PCA analysis as predictors and whether the edge existed in the dataset or not as the response. The dataset from PCA analysis was divided into train set and test set with the ratio of 2:8.

The models used included the following 3 models, SVM (Support Vector Machine), RF (Random Forest), and KNN (k-nearest neighbours). The kernel of SVM was set as linear for simplicity. 5-fold cross-validation was performed on every model to tune hyperparameters.

3. Results

The best hyperparameters for SVM model was C equal to 10 with the best accuracy achieved during the cross-validation being 0.99646. The test accuracy, specificity, and sensitivity of the SVM model were 0.91197, 0.99541, and 0.82830, respectively. The confusion matrix is as follows.

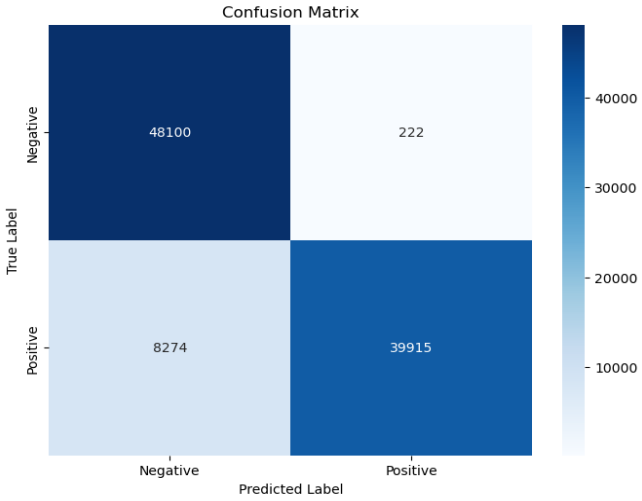

Figure 12: SVM confusion matrix

The best hyperparameters for the RF model was maximum depth being 12, minimum samples leaf being 26, minimum samples split being 133, and the number of estimators being 115 with the best accuracy achieved during the cross-validation being 0.99523. The test accuracy, specificity, and sensitivity of RF were 0.98339, 0.99439, and 0.97236, respectively. The confusion matrix is as follows.

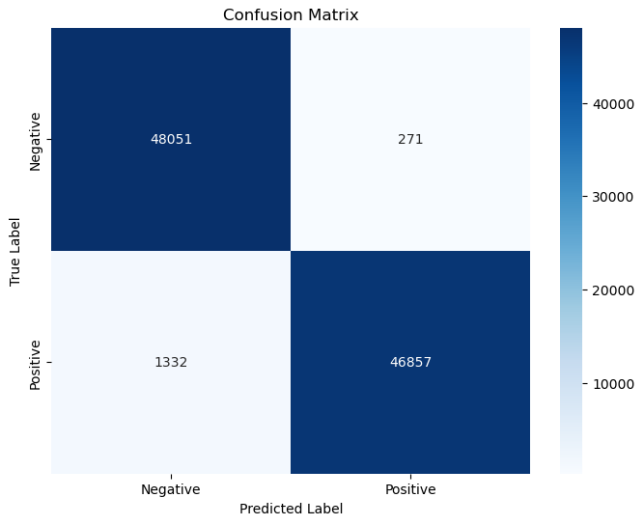

Figure 13: RF confusion matrix

The best hyperparameters for KNN classifier was K = 5 with the best accuracy achieved during the cross-validation being 0.99625. The accuracy, specificity, and sensitivity of the KNN classifier were 0.95769, 0.99385, and 0.92143, respectively. The confusion matrix is as follows.

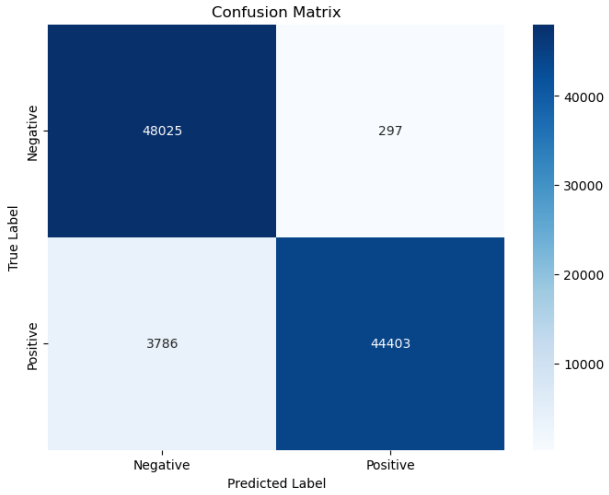

Figure 14 KNN confusion matrix

4. Discussion

i. Interpretation

From the analysis, strong positive and negative correlations between features were prevalent. This prompted the action to reduce the dimensions. In addition, the sampled network was found to be a small world, meaning that the average shortest distance between nodes were much shorter compared to a random network with same size.

Overall, all 3 classification models worked extremely well with the dataset. They all generated an extremely high accuracy on the training set, with all of them above 98% prediction accuracy. At first glance, this could be a huge sign of overfitting. However, when tested on the test set, all models still performed well, achieving over 90% accuracy. This suggested that the models indeed were able to pick up meaningful patterns from the data and delivered well on unseen data.

With a slight advantage, the Random Forest Classifier model produced the best accuracy given the test set, coming in at 98.339% accuracy. Utilizing this trained model, we could predict links between nodes in a network graph with high accuracy.

ii. Limitations

There were some limitations that might be present in this “Facebook User” analysis. Firstly, the dataset could be considered old and did not have more information on the users, for example, demography, interest, occupation, etc. This information could lead to a more comprehensive insight on the network and possibly train a better model to catch more complex underlying pattern, instead of just using the graph’s properties.

Additionally, since the original dataset was extraordinarily huge, it was not feasible to do all the feature engineering, visualizations, model training, and the overall analysis in a timely manner. By utilizing random walk to sample, the number of observations shrank, which might result to losing information from the original social network, with the worst-case scenario being not representative enough on the whole network.

Furthermore, because the features were calculated from the network properties, there could be possibility of collinearity and redundancy. These were

found when calculating the correlations between features. Although these were tuned down using dimensional reduction by PCA, there may be more optimized approach. More information on the nodes and edges would bring more perspectives into this link prediction problem. Overall, the models still worked with high accuracy, despite these problems.

iii. Suggestions

There are several suggestions to expand this project further in the future. The obvious would be to expand the predictive ability by implementing more information on the nodes and links such as occupations or demography. However, this would come with a cost of finding or making a good dataset, which could be near impossible.

Another suggestion would be to try train with more classifier models such as XGBoost Classifier or as simple as logistic regression. Here simpler or more complicated models could be explored, resulting in better comparisons, and choosing the best one for the implementation.

Moreover, another approach for simplifying dimension would be to implement RFECV (Recursive Feature Elimination with Cross Validation). This is a different approach that allows recursively training the model while selecting and removing features until no more features could be eliminated. This provides the option to retain the original values of the features as well as optimize features used tailored to each model.

5. Conclusion

Social media has been an integral part of the human lives. It has helped humankind to interact worldwide easily and conveniently. One of the most used social media platforms is Facebook. As the number of social media users increases, the social network grows into a more complex state. This invites opportunities to explore technological advancements for business opportunities.

In conclusion, our exploration on the "Facebook Users" data has brought us numerous insights on social networks. It has helped in understanding the underlying properties of a graph data, such as centralities and degrees, and how they interact with each other. These calculated values allowed for the training of several supervised learning models. The models' purpose was to make predictions on potential connections in the network.

After thorough model training, with a slight advantage, we found that the Random Forest Classifier accomplished the best accuracy on the dataset. This link prediction ability would be useful for designing a network recommendation system on real applications or social media platforms specifically, such as Facebook. That way, it will boost the user experience by providing tailored recommendations for connections, as well as open opportunities for business development of these social media platforms.

Although our analysis and model were able to provide actionable implementations on social media platforms, there were some limitations, for instance, lack of information and redundancy. Some suggestions for future studies would be to gather more detailed information on the observations or explore different approaches such as different data preprocessing procedure or classifier models.

References

Backlinko. (2024, September 4). *Facebook User & Growth Statistics*. https://backlinko.com/facebook-users

Facebook. (2012). *Facebook Recruiting Competition*. Kaggle. https://www.kaggle.com/c/FacebookRecruiting/data

Flynn, C. P. (Ed.). (2008). *SOCIAL CREATURES: A Human and Animal Studies Reader*. Lantern Books. https://books.google.com.hk/books?hl=en&lr=&id=d6nT4VGleOEC&oi=fnd&pg=PR13&dq=human+is+a+social+creatures&ots=waoeKq6c9x&sig=Fqo8ToPweB51qJqtTQvsTAebrPI&redir_esc=y#v=onepage&q=human%20is%20a%20social%20creatures&f=false

Leskovec, J., & Faloutsos, C. (2006). Sampling from large graphs. In *Proceedings of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining* (pp. 631-636). https://doi.org/10.1145/1150402.1150479

Appendix

Appendix A: Source Code for Social Network Analysis

This appendix includes a link to the Google Colab notebook that features the source code and detailed analysis for the methodology, including the data preprocessing, feature engineering, and dimension reduction, up until the model training conducted in this project.

The source code can be accessed in the Google Colab notebook here: [SDSC3016_Assignment_1.ipynb - Colab (google.com)].