DBA5104: Introduction to Network Science & Analytics
Group 5

Project CoRe

‘CO’llaboration ‘RE’commendation using Community Detection and Link Prediction in YouTube

Name	Matriculation Number
Nathaniel Nartea Casanova	A0262708B
Anusha Mediboina	A0262847U
Prasanna Govindarajan	A0262732H
Asok Kaushik	A0262739U
Janita Bose	A0262823E

Introduction

Background

Collaborative efforts serve as a crucial strategy for expanding the reach and enhancing the appeal of user-created video content on the YouTube platform. Collaborative endeavors frequently involve a mutually beneficial exchange of content promotion on both participating channels, with the intention of attracting each other's viewership and ultimately augmenting the revenue of both YouTubers. Performing a quantitative evaluation of diverse collaboration, such as those involving YouTubers from varying levels of popularity or content categories, has the capability to offer valuable insights and inform YouTube content creators regarding strategies that can optimize their popularity and revenue potential.

Objective:

The main objective of this paper is to propose a YouTube collaboration recommendation (CoRe) tool which leverages the network by developing a Link Prediction algorithm and Community Network analysis. CoRe can be added as a new feature in YouTube Studio where it will recommend collaborations to content creators in the YouTube Studio dashboard. CoRe aims to meet the following motivations:

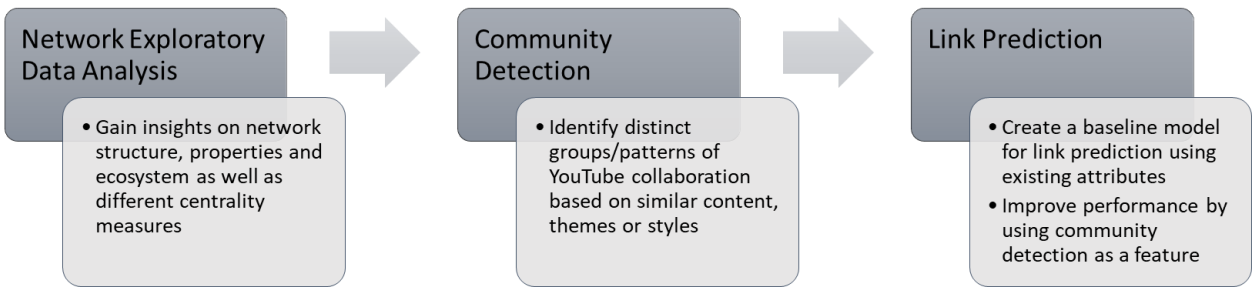
Increased reach for content creators: Collaboration between creators generates synergy, exposing their audiences to new content and boosting engagement. This process enables creators to access new demographics, explore innovative ideas, and expand reach, thereby increase revenue.

Diversification of Content: Collaborations blend diverse perspectives to produce innovative content and enhance the potential for unique creations. Link prediction on a YouTube collaboration network aims to uncover untapped or disconnected collaboration opportunities to boost network connectivity and collaboration potential.

Increased Brand Opportunities: Community detection in a YouTube collaboration network reveals the structure and composition of content ecosystems, highlighting groups of YouTubers with similar themes or styles. This information assists businesses in targeting specific communities for advertising and identifying potential influencers or emerging talent within niche content.

Strengthening the YouTube ecosystem: Channel collaborations on YouTube create a sense of community, driving growth and success through idea-sharing and engagement among creators. This interconnectedness leads to the formation of alliances, fostering knowledge and resource-sharing that enhances content quality on the platform.

Proposed solution and Methodology:



Network Ascertainment

We evaluated a 3-month collaboration network graph of 2455 YouTubers existing data recorded in the time span between 28.12.2016 and 28.03.2017 which was a part of research conducted by Multimedia Communications Lab in Germany [1]. The graph is created with channels as nodes and collaborations as directed edges connecting the nodes. The collaboration frequency is represented as weight of edges. The graph is a bi-directional graph wherein directions describe the origin of a content creator.

Network Analysis / EDA on Network

Figure 1. ***Graph network***

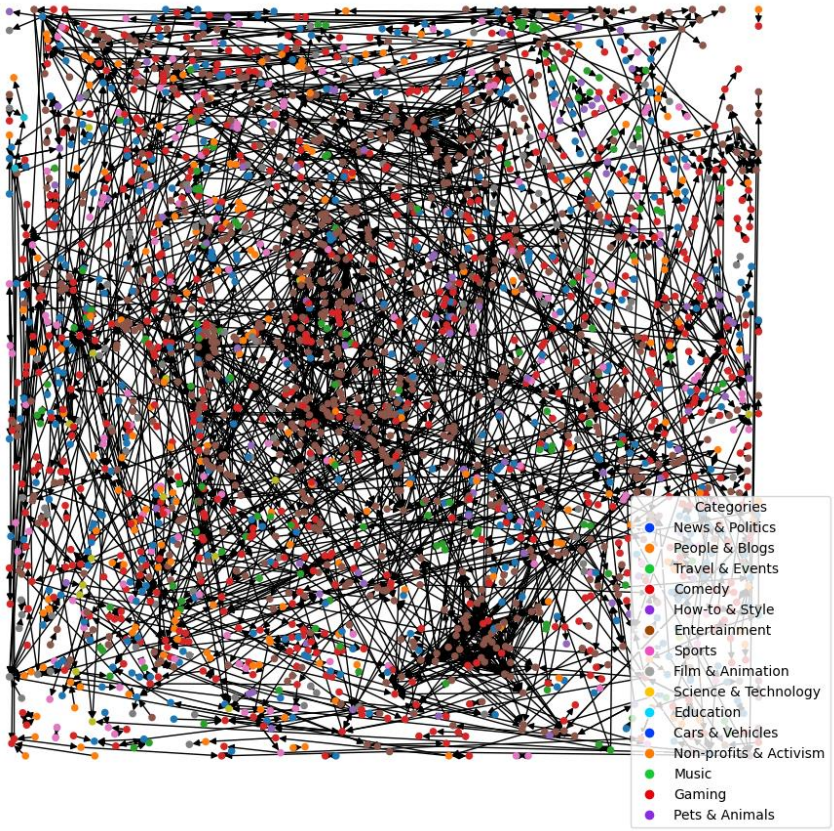


Table1: ***Graph Level Features***

Number of Nodes	**2455**
Number of Edges	3475
Diameter	24
Density	0.0006
Average Clustering Coefficient	0.17
No of connected Components	255
Average Path Length	1.08

The existing YouTube collaboration network as indicated by the graph above is sparse network as evident from very low density of the network (~0.001).

The sparsity of the network can also be explained by number of edges in relation to the number of nodes, in other words, the nature of network is such that the number of potential collaborations is significantly higher than the current collaborations.

Therefore, the sparse network formed has immense potential for new future collaboration from different communities and strongly aligns with the business objective defined in the project.

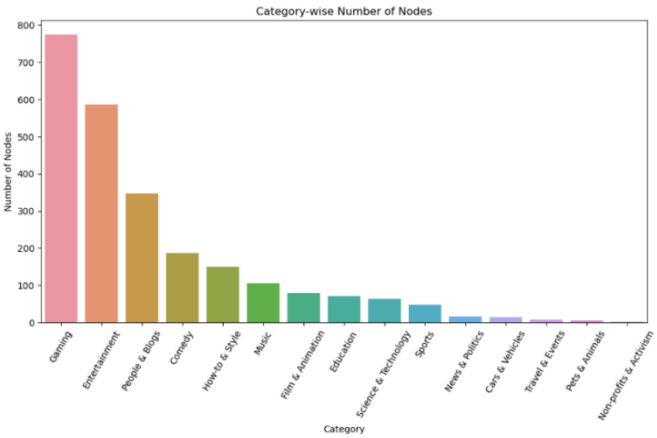
It may be noted that despite being a sparse network, it has an unusually low average path length which can be explained by the presence of interconnected communities of collaborators.

Network Characteristics

Some of the other key observations regarding the network is summarized below. The charts supporting the same is provided in appendix:

- **Degree Distribution** indicates Scale-free network
- **Betweenness Centrality**: While most of the nodes have low betweenness centrality, the maximum value is high in comparison to the mean, (53997 vs. 342) indicating that there are few nodes that have high influence over the flow of information.
- **Eigenvector Centrality:** Negligible mean and median indicating most nodes have low influence on the network
- **Page Rank Distribution & Closeness Centrality Distribution** have low mean and median indicating most nodes have low importance and are not very central to the network
- **Clustering co-efficient distribution** indicates most nodes have low local clustering.

Category-wise Distribution of Nodes:



As seen from the chart besides, majority of the nodes pertain to the 'Gaming' and 'Entertainment' category. We have introduced channel category dummies for the nodes.

Other Attributes Included:

Based on subscriber count the channels are classified from 0-6 to indicate the Popularity Index. Popularity class definitions and their number of observed channels can be referred in Appendix Table 1.

Total Views, Videos , Subscriber Count, Comment Count which are interaction statistics of videos in a channel present in the data. These are used as inputs for link prediction analysis.

Community Detection

Community detection is useful in collaboration data as it can help to identify groups of channels or creators who frequently collaborate with each other. Community detection can also reveal trends in the types of content being produced and or uncover patterns in collaboration behavior. Understanding these collaborative relationships can provide insights into the structure and dynamics of the YouTube ecosystem which can be valuable for marketing and business purposes. For this project, we have implemented two types of community detection algorithms namely, Louvain and InfoMap.

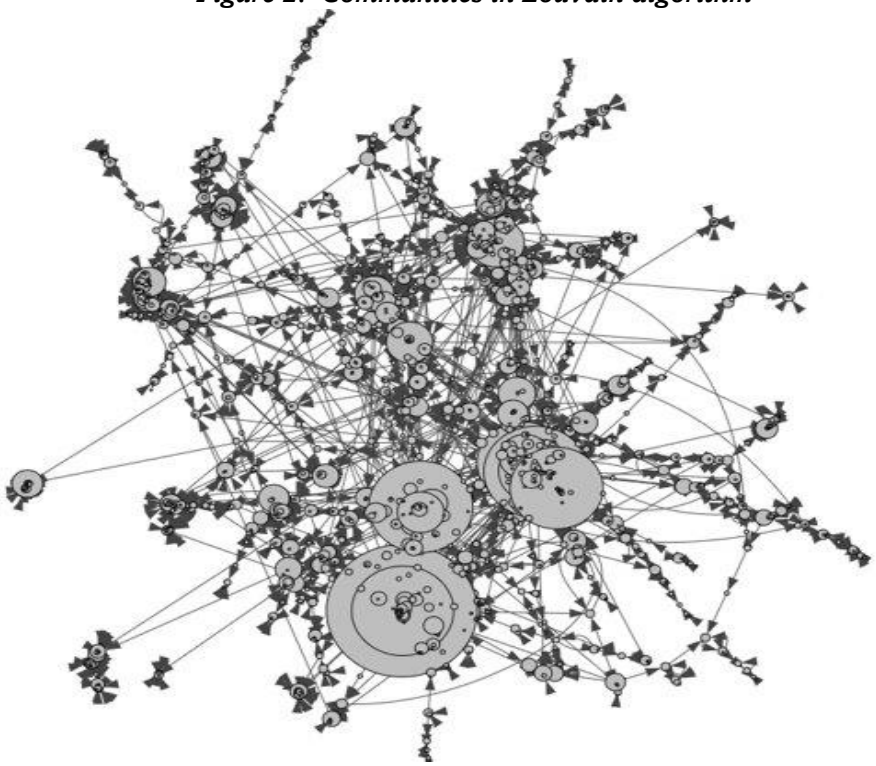
Louvain Algorithm:

An undirected version of the current 'collabs' network was used for community detection, the result of which is a list of communities, where each community is represented as a list of vertex indices. We filter out the list of communities to only include those that have at least 20 vertices.

The outcome is a list of communities that are considered "large". The number of communities detected was 26. Below is the visualization of the graph. The size of the bubbles represents twice the number of degrees of each node.

Louvain method is a widely used community detection algorithm that works by optimizing the modularity of a network, which measures the degree to which a network is divided into tightly knit groups or "communities". Modularity ranges from -1 to 1, with values closer to 1 indicating a strong community structure. A value of 0 indicates that the community structure is no better than random, while negative values indicate that the network is less modular than random. In this case, the modularity score was 0.83 which indicates a strong and dense network.

Figure 2. Communities in Louvain algorithm

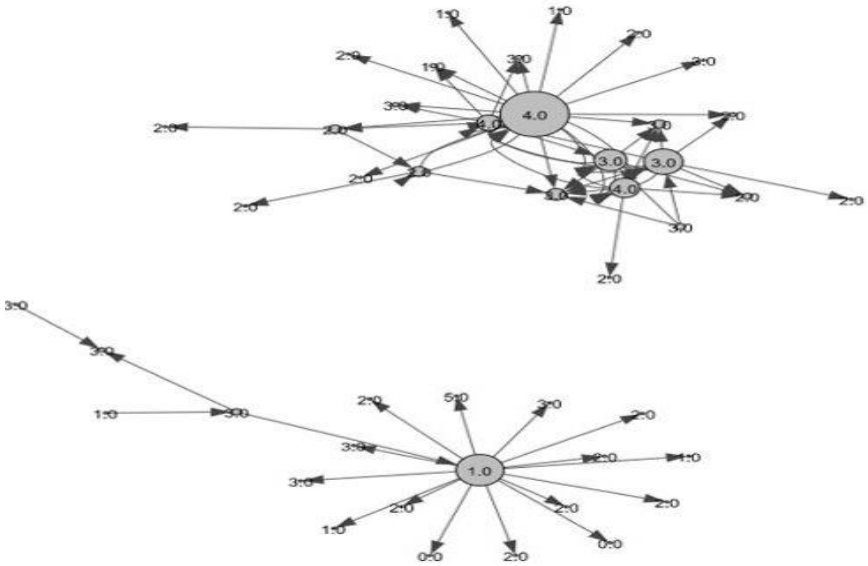


InfoMap Algorithm:

The community parameters and strategy for InfoMap were similar to that of Louvain method, except for using directed 'collabs' network. The outcome was 2 large unconnected communities with more than 20 nodes. Below is the representation of the outcome. The labels of each node represent the popularity index and the direction of the edge represents the direction of collaboration.

The Infomap algorithm works by first partitioning the network into modules based on the links between the nodes. It then constructs a map of the modules, where each module is represented by a node and the links between the modules are represented by the links between the nodes in the map. The algorithm then performs a hierarchical optimization of the map, seeking to find the partition of the network that minimizes the expected description length of the random walker's traversal. The resulting partition can be interpreted as a hierarchy of modules, where each module is composed of sub-modules that are more strongly connected to each other than to the rest of the network.

Figure 3. Communities in Infomap algorithm



Analysis

It is not surprising for the Louvain algorithm to not show any unconnected communities, as the algorithm tries to optimize the modularity of the graph by moving nodes between communities. In the process, the algorithm tends to group connected nodes together to form communities. On the other hand, the Infomap algorithm tends to optimize the information-theoretic cost of encoding a random walker's path through the network, which may result in communities that are not necessarily connected. As modularity measures the quality of the community structure based on the internal connectivity of the communities and the external connectivity between the communities, the modularity score may not be affected by the presence of unconnected components. This reconstitutes the scale-freeness of the network mentioned in EDA as it has some densely connected subgraphs that are not connected to the rest of the network.

Figures 4 and 5 represent the individual unconnected communities based on the popularity of channels and categories of collaboration. It can be observed that for both clusters, collaboration is occurring mostly on Gaming videos followed by Entertainment videos. However, the trend that both clusters exhibit is opposite to each other. Channels of community in Figure 4 show a trend where more popular channels are collaborating with lesser popular channels, whereas the channels of community in Figure 5 show an opposite trend.

Figure 4. ***Popularity and Categories of first unconnected community***

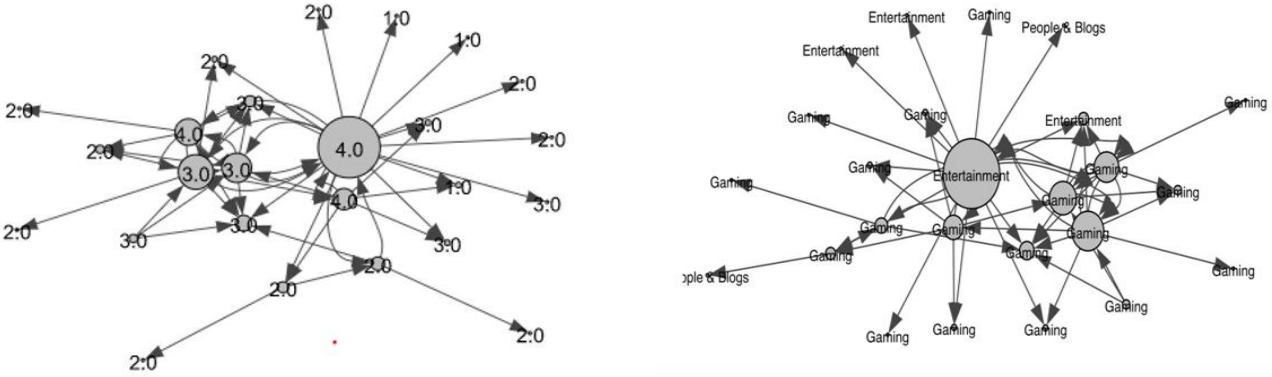
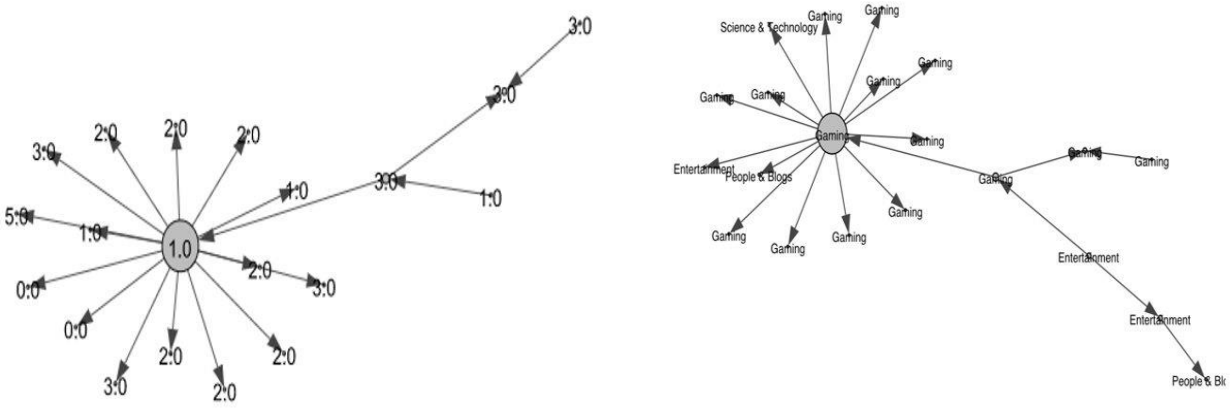


Figure 5. ***Popularity and Categories of second unconnected community***



This can be further observed in the analysis of each of the large communities based on their node level attributes.

Table2: ***Popularity of the communities***

Attribute/Community	Community1	Community2
Max popularity	4.0	4.0
Min Popularity	1.0	1.0
Avg Popularity	2.38	2.44

Table3: ***Subscriber Count of the communities***

Attribute/Community	Community1	Community2
Max Subscriber Count	1,231,535	2,316,278
Min Subscriber count	2,496	6,399
Avg Subscriber Count	175,268.67	364,050.24

Table4: ***View Count of the communities***

Attribute/Community	Community1	Community2
Max View Count	315,527,100	213,645,867
Min View Count	33,049	73,732
Avg View Count	45,132,800.9	35,844,158.08

Table5: ***Video Count of the communities***

Attribute/Community	Community1	Community2
Max Video Count	2,328	1,194
Min Video Count	43	53
Avg Video Count	821.14	437.76

Categories and count of channels in Community1: Gaming - 16, Music - 1, Entertainment - 3, Science & Technology – 1

Categories and count of channels in Community2: Entertainment - 4, Gaming - 19, People & Blogs – 2

Higher the subscriber count, larger the popularity index [Appendix Table A1]. Based on the above community structure and node attributes, below community profiles can be hypothesised:

Profile of Community 1: Popular gaming channels collaborating with lesser popular channels on gaming and other categories. Low average subscriber count is due to the existence of many low popular channels.

Profile of Community 2: A single less popular gaming channel acting as a moderator/mediator to help other popular channels collaborate for game streams hoping to improve its popularity and subscriber count in parallel. High average subscriber count is due to the existence of many popular channels.

The obtained community labels can be utilised to improve the node embeddings, which is explained in the next section.

Link Prediction

Methodology

The core of the proposed YouTube Collaboration Recommendation (CoRe) Tool will be a robust link prediction model that can recommend collaborations between channels in the YouTube platform. We have noted that the current YouTube network analyzed in this paper is sparse, and is just a sample of the overall YouTube collaboration network, and therefore there is ample space for more collaborations on the platform.

For link prediction between two YouTube channels, two methods were implemented. The first involved using local similarity measures between node pairs to predict whether a link should exist between them. This is based on the assumption that if two nodes have common neighbour structures, a link between the pair might exist in the future. The Adamic-Adar Index and the Jaccard Index were both used to compute similarity scores between node pairs. Using the similarity scores, the algorithm used to predict links between node pairs and the accuracy computation of the prediction are as follows:

1. Randomy sample f percent of edges fron original graph G, to get $edge_{delete}$
2. Let $edge_{count}$ = count of edges in $edge_{delete}$
3. Create new graph G', where $G' = G - edge_{delete}$
4. Get all possible node-pair combinations from G'
5. For each G' node-pair, compute similarity score
6. Get k node-pairs with highest similarity scores, where k = edge count in original graph G'. Let's call these $Edge_{new}$
7. Get count of $edge_{delete}$ existing in $Edge_{new}$. Let's call this $edge_{intersect}$
8. $$Accuracy = \frac{edge_{intersect}}{edge_{count}}$$

The second approach that was implemented is GraphSage [Hamilton et al], a feature-learning method that aggregates a node's local neighborhood characteristics recursively or iteratively using a convolutional neural network. According to Mutlu and Oghaz (2019), the iterative characteristic of GraphSage makes models built on this architecture generalizable on unseen nodes [3]. This feature of GraphSage makes it attractive for use in a collaboration recommendation tool business case as the actual YouTube collaboration network is dynamic with channels appearing and disappearing overtime and the network doesn't have to be retrained when a new channel enters the network.

The specific implementation in this paper utilizes a two-layer GraphSage model to solve a supervised learning task where we take the existing links in the network as positive cases of a link and non-existent links as negative cases. A two-layer model means that a single node's representation will involve a mean aggregate of the node's two-hop random neighbors. This representation will then be passed through the GraphSage model to get the final node embeddings and then through a dense net layer for binary link prediction. For the link prediction task, a fraction of existing links in the network will be randomly sampled and deleted from the network. The reduced graph will then be used for training a GraphSage link prediction model. 30% of existing and non-existing edges are sampled from the reduce graph to form the labels for training. The model is then fed samples of node pairs, represented with their node embeddings, with their corresponding binary labels if an edge exists between the node pair or not. This is all implemented using the StellarGraph library. Different combinations of features are used to test different version of the model. These features are enumerated in Table X in the appendix.

To further enhance the node embeddings, the community labels derived from the community detection in the previous section were added as a node feature and another model was trained using these features.

Note that both methods do not consider the directionality of the collaboration, so in the prediction algorithm the network was treated as an undirected graph.

Results

The results of the similarity-based and GraphSage link prediction are summarized below:

Table X: Similarity-Based Local Methods

Measure	Edges Removed	Accuracy
Adamic-Adar Index	20%	13.09%
Jaccard-Index	20%	5.18%

Table X: GraphSage Feature Learning

Features	Edges Removed	Test Accuracy
Node Attributes Only	20%	83.96%
Channel Attributes Only	20%	71.01%
Node Attributes with Community Dummies	20%	**85.25%**
Node and Channel Attributes Only	20%	84.03%
Node and Channel Attributes with Community Dummies	20%	84.96%

The similarity-based methods yielded very low accuracy which means these methods were not able to predict back most of the deleted edges in the original network. For Adamic-Adar Index, which yielded the best accuracy in this method, it was only able to predict back 13.09% of the deleted edges in our collaboration network. This is probably because our network is sparse and the similarity measures were not able to capture sufficient common neighbor information to be reliable in link prediction. This justifies our use of feature-learning methods.

For GraphSage, the overall accuracy of the link prediction are significantly better than similarity-based methods with accuracies ranging from 71-85% on the test set of deleted edges. This is means it can predict whether a link should exist or not exist between a node pair around 71-85% of the time. Using only the node centrality attributes as features (e.g. degree, PageRank, etc) proved powerful already as it can already achieve around 84% accuracy. Adding the channel attributes as features together with node centrality attributes in the model only yielded a slight improvement. However, using channel attributes is still useful in the case that the models need to do inference on a new channel in the collaboration network, where the channel's node centrality measures are not existent yet.

Concerning adding community label dummies as features, it is observed that combining community label dummies with node centrality attributes yielded the highest accuracy among all the models, albeit at a slight margin. From the pretext of the current network structure, we believe that this is possible because adding these features provided marginally better node embeddings for the neural network model. The marginal effect can be explained by the few sizable communities in the community detection exercise.

From these results, it is recommended that a GraphSage model be used for building the collaboration recommendation tool because of its superior performance over similarity-based methods.

Limitations and Future Work

For link prediction, there are several limitations that need to be mentioned. First, the model that was recommended has a cold-start problem where it cannot generate recommendations for new channels that has not done any collaborations and have low channel statistics. This is because enough data is required to generate reliable embeddings for the link prediction neural network. The collaboration recommendation will only be useful for emerging and mature channels looking to further accelerate growth. Secondly, the model inference pipeline to predict future links for the have yet to be developed. Lastly, if explainability is a concern for the tool stakeholders, additional expliration needs to be done on available explainable AI tools for the GraphSage StellarGraph library.

References:

1. Christian Koch, Moritz Lode, Denny Stohr, Amr Rizk, & Ralf Steinmetz. (2018). Collaborations on YouTube: From Unsupervised Detection to the Impact on Video and Channel Popularity.
2. William L. Hamilton, Rex Ying, & Jure Leskovec (2017). Inductive Representation Learning on Large Graphs. CoRR, abs/1706.02216.
3. Ece C. Mutlu, & Toktam A. Oghaz (2019). Review on Graph Feature Learning and Feature Extraction Techniques for Link Prediction.
4. CSIRO's Data61. (2018). StellarGraph Machine Learning Library.

Appendix

Fig A1: Betweenness Centrality

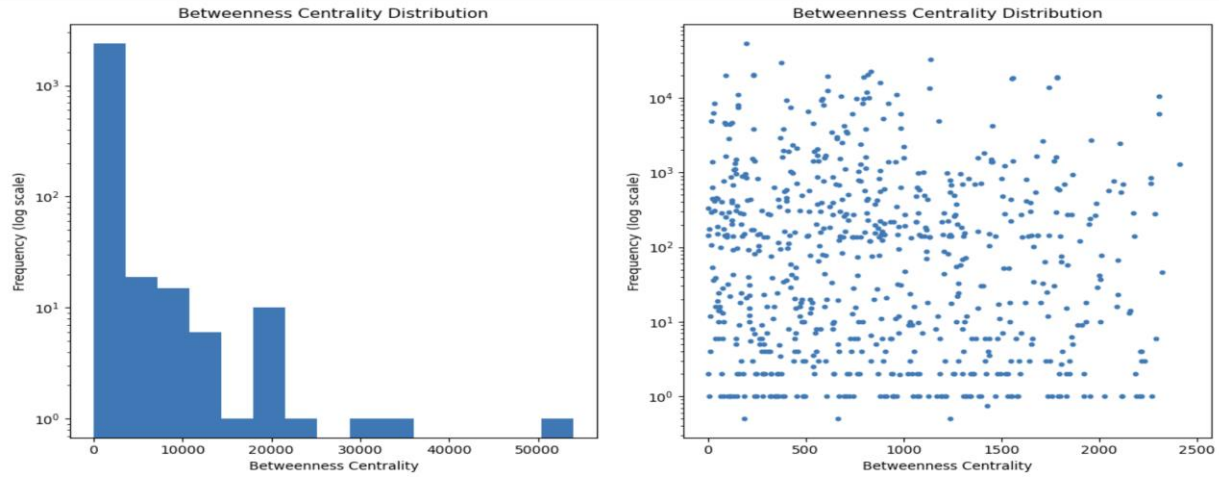


Fig A2: Eigenvector Centrality

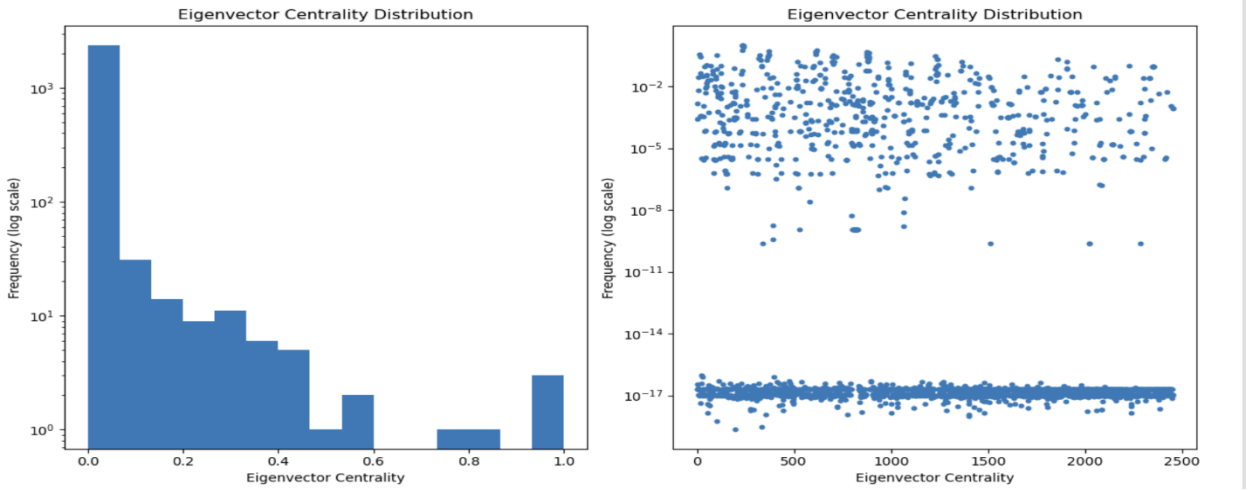


Fig A3: Closeness Centrality

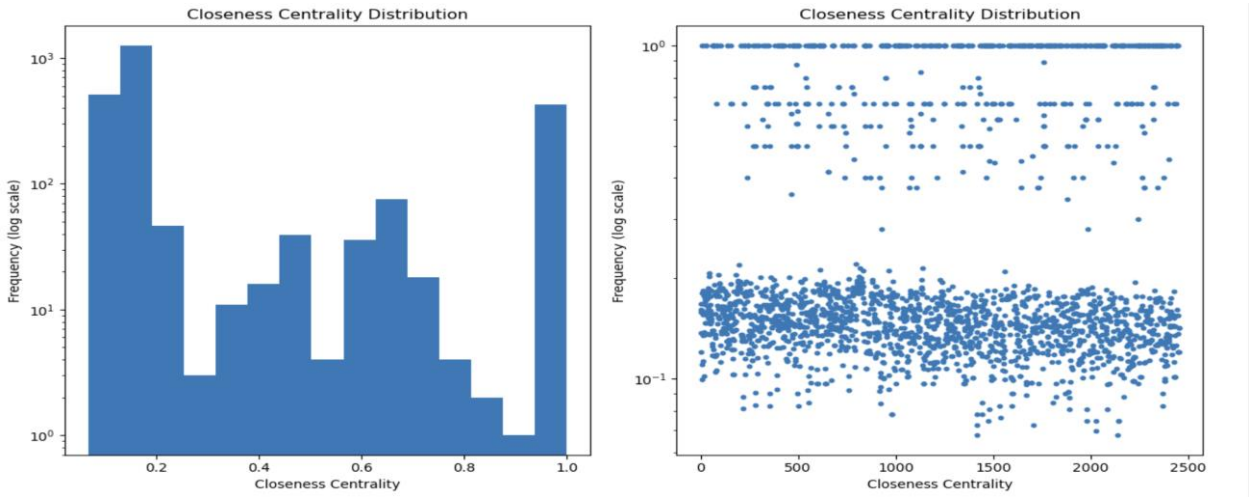


Fig A4: *Degree Distribution*

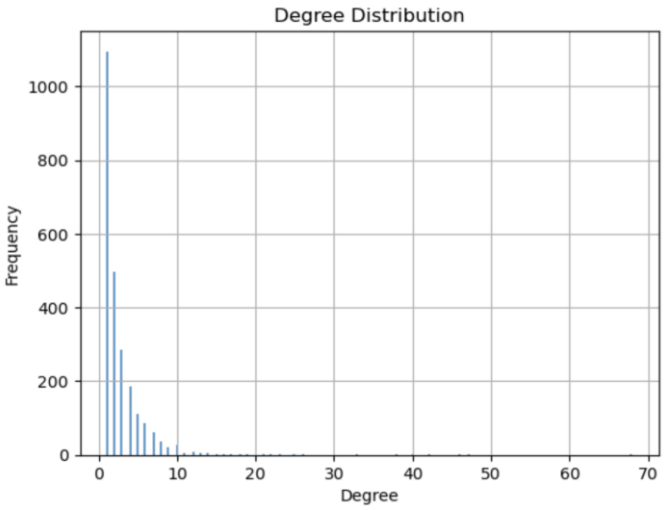
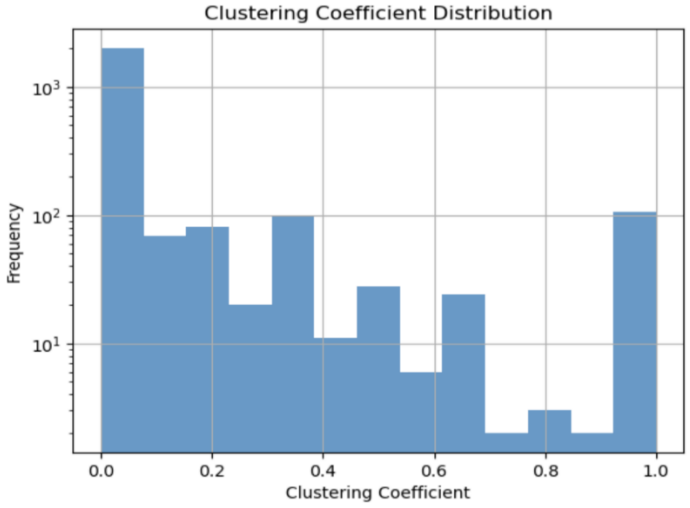


Fig A5: *Clustering Coefficient Distribution*



FigA6: PageRank Distribution

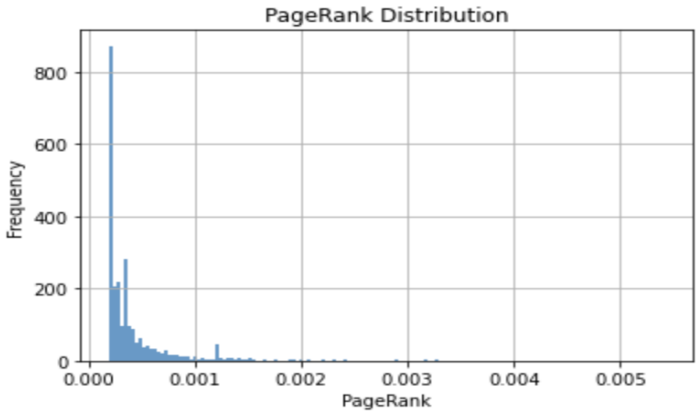


Table A1: Popularity class definitions

Popularity Class	Subscriber Range
0	$[0, 10^3)$
1	$[10^3, 10^4)$
2	$[10^4, 10^5)$
3	$[10^5, 10^6)$
4	$[10^6, 10^7)$
5	$[10^7, 5x10^7)$
6	$[5x10^7, 10^8)$

Table A2: GraphSage Model Features

Node Centrality Measures	**Channel Attributes***	**Cluster Dummies**
Degree	View Count	Cluster 1 Dummy
Closeness	Subscriber Count	Cluster 2 Dummy
PageRank	Video Count	No Cluster Dummy
Eccentricity	Comment Count	
	Popularity Index	

*Standardized by removing the mean and scaling to unit variance