Illinois Institute of Technology
Department of Computer Science

CS 579: Online Social Network Analysis

Project 2 - Explainable graph neural network

Team Members:
Shubham Modi ID:A20492276
Oleksandr Shashkov ID:A20229995

1. Project Objectives

The goal of this project is to re-implement GNN Explaner described in[1], run experimental explanations on two different datasets not used in the original work, and analyze explanations obtained as a result of such experiments

2. Introduction

In many domains of human knowledge and activities, the data can be represented as graphs. Graphs are versatile and powerful but complex representations of the world. Graph Neural Networks (GNN) have emerged as state-of-the-art machine learning models. As many other machine learning models, GNN lacks transparency of its internal workings therefore making explainability of the predictions a significant concern. The ability to understand GNN predictions is a very desirable feature as it may boost confidence in GNN models, improve transparency of the data processing and enable machine learning practitioners and researchers with better analysis and troubleshooting tools. GNNExplainer: Generating Explanations for Graph Neural Networks, Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik & Jure Leskovec [1] is one of the most influential research papers on the subject of GNN explainability.

3. Previous work and literature review

In the original work [1] authors introduced a novel and general, model-agnostic approach for providing interpretable explanations for predictions of any GNN-based model on any graph-based machine learning task. Given a graph instance, GNN Explainer identifies a compact subgraph structure and a small subset of node features that have a defining role in GNN's prediction (Single-instance explanation). Furthermore, GNN Explainer is capable of generating quality explanations for an entire class of graph instances (Multi-instance explanation).

According to the authors GNNEXLAINER is an optimization task that maximizes the mutual information between a GNN's prediction and distribution of possible subgraph structures.

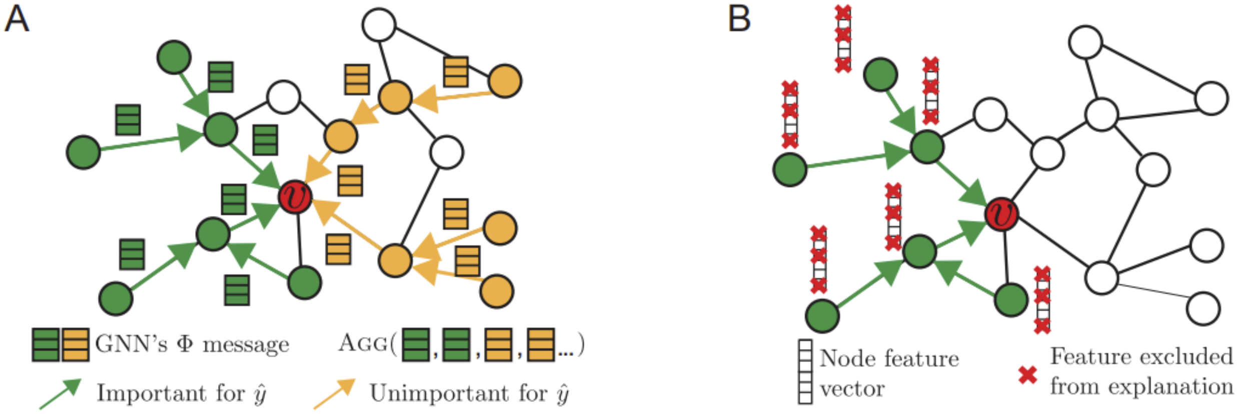


***Figure 1**: Illustration of the explanations.*
***A** - based on the importance of connections with certain neighbors.*
***B** - based on the connectivity and the importance of certain features [1]*

<u>The key insight </u>of the original paper [1] was that the computation graph of node v, which is defined by the GNN's neighborhood-based aggregation, fully determines all the information the GNN uses to generate prediction $\hat{y}$ at node v. In other words, the neighborhood, its state and state of the node defines everything.

The GNN Explainer is not a solution that has answers to all possible questions related to the explainability of the GNNs. It does have limitations and areas for improvement. According to authors [1], GNN Explainer was quite capable in highlighting a compact feature representation. However, the gradient-based approaches struggle to cope with the introduced noise, giving high importance scores to some irrelevant feature dimensions.

The second influential source [2] introduced the PGExplainer model that adopts a deep neural network to parameterize the generation process of explanations, which enables PGExplainer a natural approach to explaining multiple instances collectively and it is based on probabilistic approach.. The authors of this work claimed that the PGExplainer model demonstrates better generalization ability.

4. GNNExplainer re-implementation

4.1. Graph Neural Network

The Graph Neural Network used for this project is made from 3 GCN convolutional layers. GCN is a type of convolutional neural network that can work directly on graphs and take advantage of their structural information. It solves the problem of classifying nodes (such as Users, Publications) in a graph (such as a Twitch or Cora dataset), where labels are only available for a small subset of nodes (semi-supervised learning). The general idea of GCN: For each node, we get the feature information from all its neighbors and of course, the feature of itself. Assume we use the average() function. We will do the same for all the nodes. Finally, we feed these average values into a neural network. The initial layer has neurons equal to the number of features for each node and the output layer has neurons equal to the number of classes in the dataset. The last layer is a linear layer which concatenates the output of all 3 hidden layers and transforms it into probabilities for class prediction.

The GNN model makes use of ReLU activation function for calculating the weights and L2 normalization for penalty.

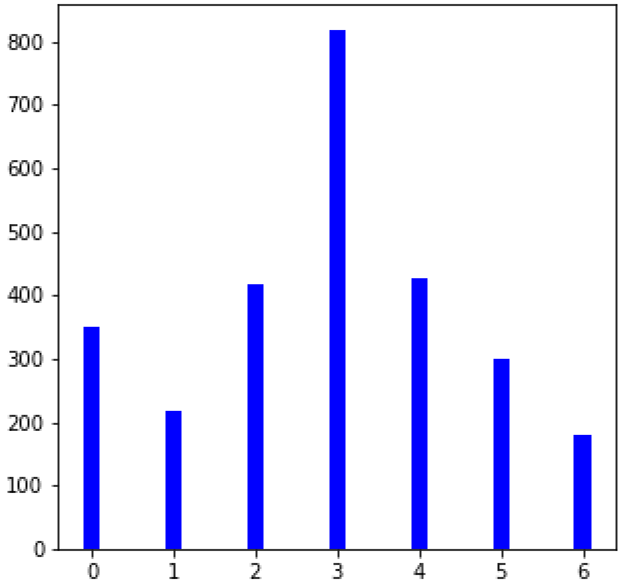
The model uses CrossEntropyLoss as the loss function which provides acceptable results for classification tasks.

4.2. Graph Neural Network Explainer

The original codebase for GNNExplainer was used to build a model for the experiments [1]. We also re-used a codebase used by other researchers [2],[3] to replicate the study and to adapt the GNNExplainer to the datasets we wanted to explain. The implementation was tailored to conduct node classification in a single instance graph setup. GNN Explainer core functionality is the optimization task to maximize Mutual Information (MI) between the original state of the nodes in the graph and a state of the subgraph produced as a result of the explanation (i.e. a subset of the nodes and a subset of the features).

5. GNN Explanations for experimental datasets

5.1. Dataset 1 - Cora



The Cora dataset consists of 2708 scientific publications classified into one of seven classes. The citation network consists of 5429 links. Each publication in the dataset is described by a 0/1-valued word vector indicating the absence/presence of the corresponding word from the dictionary. All words with document frequency less than 10 were removed. The dictionary has 1433 unique words.

The dataset represents a perfect taks for GNN to classify the nodes based on the connectivity of the graph and by the state of features for each node (dictionary). Hence, the experiments with explaining predictions based on this dataset will be a single-instance graph explanation. And the goal is to obtain an explanation for predicting a class that would be assigned to a node within a single graph.

5.2. Dataset 2 - Twitch

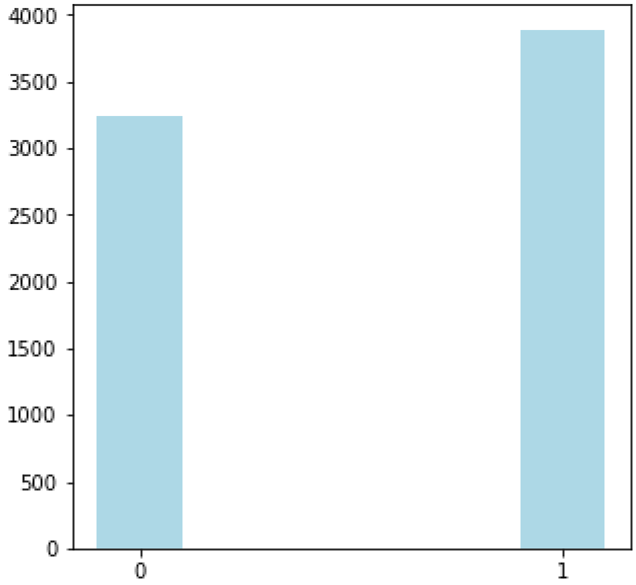


Figure 3: *Twitch Dataset split as per Labels*

The Twitch gamer dataset is a user-user network where nodes correspond to Twitch users and links to mutual friendships. Node features are games liked, location and streaming habits. All datasets have the same set of node features enabling transfer learning across networks. The associated task is binary classification of whether a streamer uses explicit language.

The labels are classified as 1/0 where 1 represents the user node which streams mature content and 0 represents the users who don’t stream mature content.

This dataset becomes an ideal piece of information for Graph Neural Networks given the importance of the data and the applications of the knowledge obtained from this dataset. As we need to classify a particular node/user in this case we consider the graph and train the model on the graph to understand the flow of information in the graph.

6. Results and discussion

Fig 4. demonstrates the explanation of the predicted label for the node index 1505 in the **Twitch dataset**. The classification of nodes is defined by the connectivity to it's neighbours and aggregating the flow of information from it's neighbours.

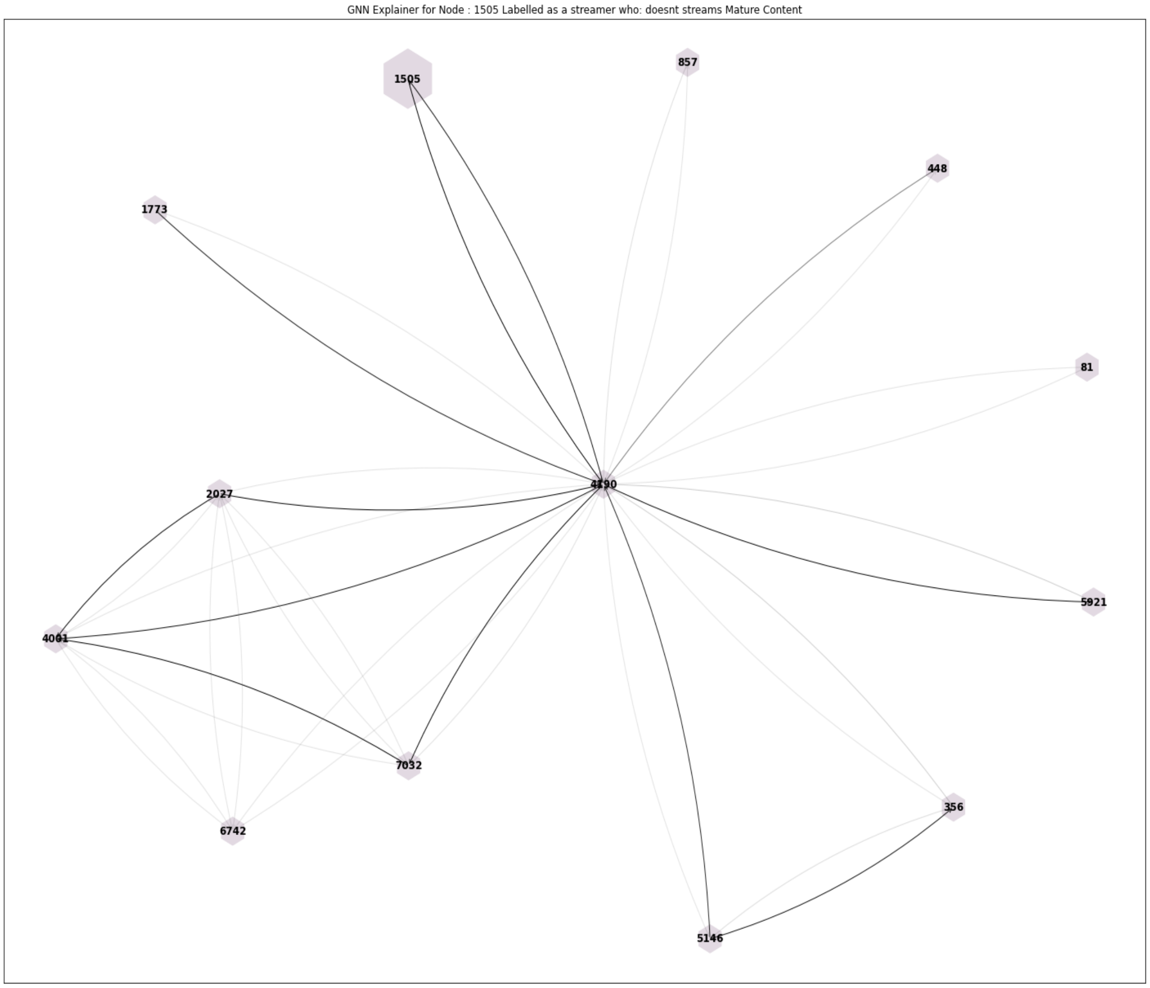

GNN Explainer for Node : 1505 Labelled as a streamer who: doesnt streams Mature Content
1505
857
448
1773
81
4290
2027
5921
4061
7032
356
6742
5146

Figure 4*: Explanation of the node with index 1505 using mutual information*

When analyzing the result obtained from the GNN Explainer we believe that the target node i.e. node 1505 is influenced by it's neighbouring nodes and the features important for prediction of this node are a result of the aggregation of the same information from the neighbouring nodes which in this case are nodes {4290, 2027, 4062, 7032, 5146, 5921}. The Edges are given the opacity between 0 and 1 which define the importance of the information passed on by that node. The node 1505 was marked as a Twitch user who does not make use of explicit language in this streams because it is believed that his/her friends/ followers do not make use of explicit language in their content either.

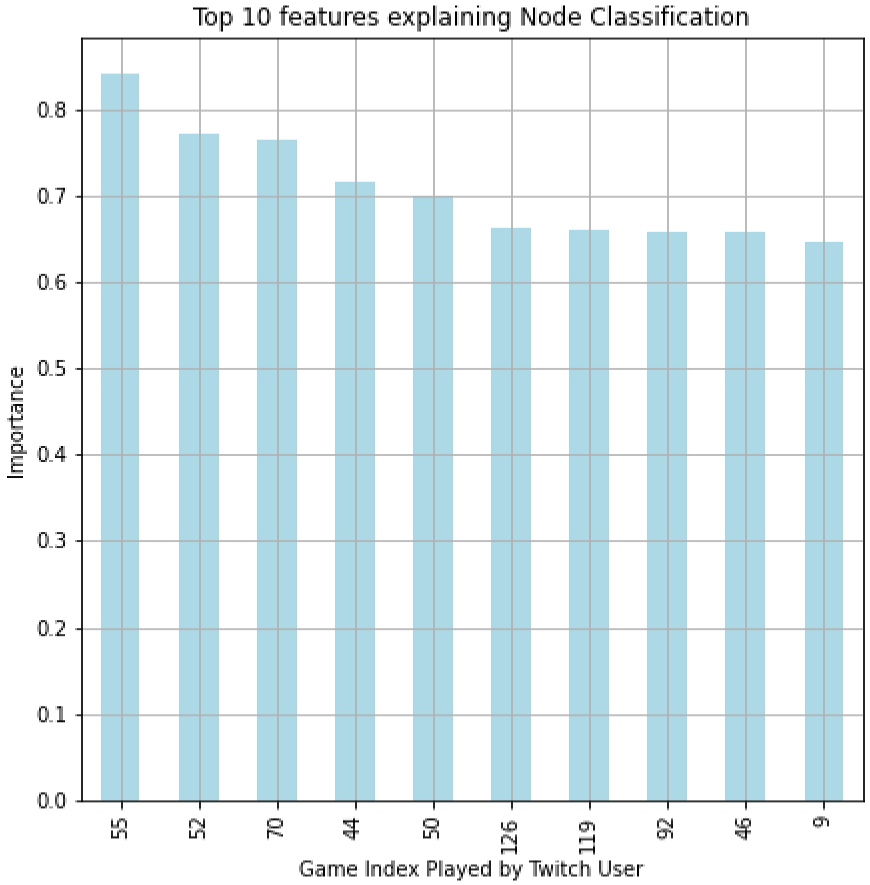


Figure 5: *Important node features contributing to node classification*

Figure 5, represents the most important node features of the target node which were almost important in the nodes which influenced the node 1505 to be predicted as a user who doesn't stream content with explicit language. This explains the type of games played the target user and his/her followers. There are 128 such features for each node which provide information about a particular streamer which is used to aggregate this information.

Fig. 6 and 7 demonstrate the explanation of the predicted topic of the publication number 400 in the **Cora dataset**. As expected, the classification of nodes is defined by the connectivity and features of the neighbors within the close neighborhood (3 hops).

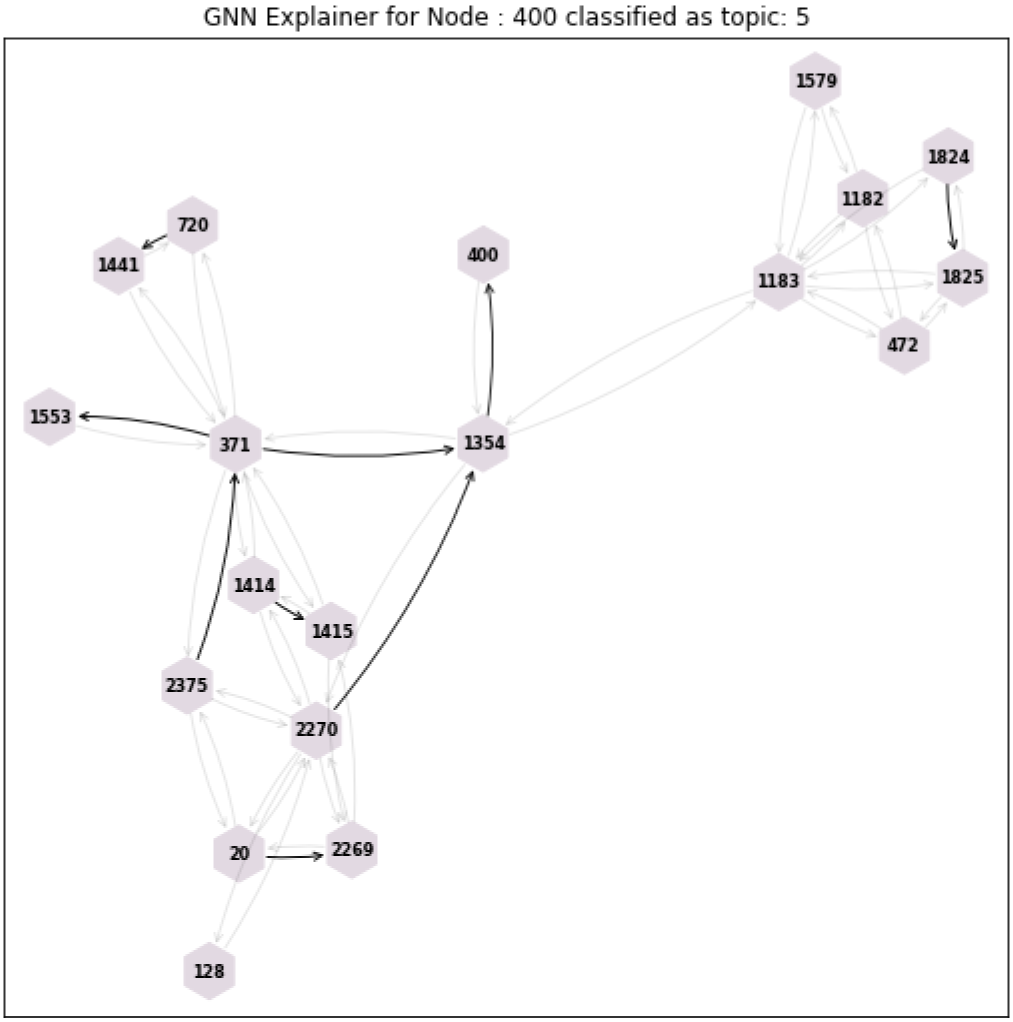


Figure 6. *Explanation of the node classification from the connectivity standpoint*

The edges highlighted in bold contributed the most to the classification of the node 400. GNN explainer also produced the feature importance mask array. The features with the highest weight in this array are the words from the embedding dictionary that had the greatest influence on classifying the node. Figure 6 displays top 10 influencers with this regard.

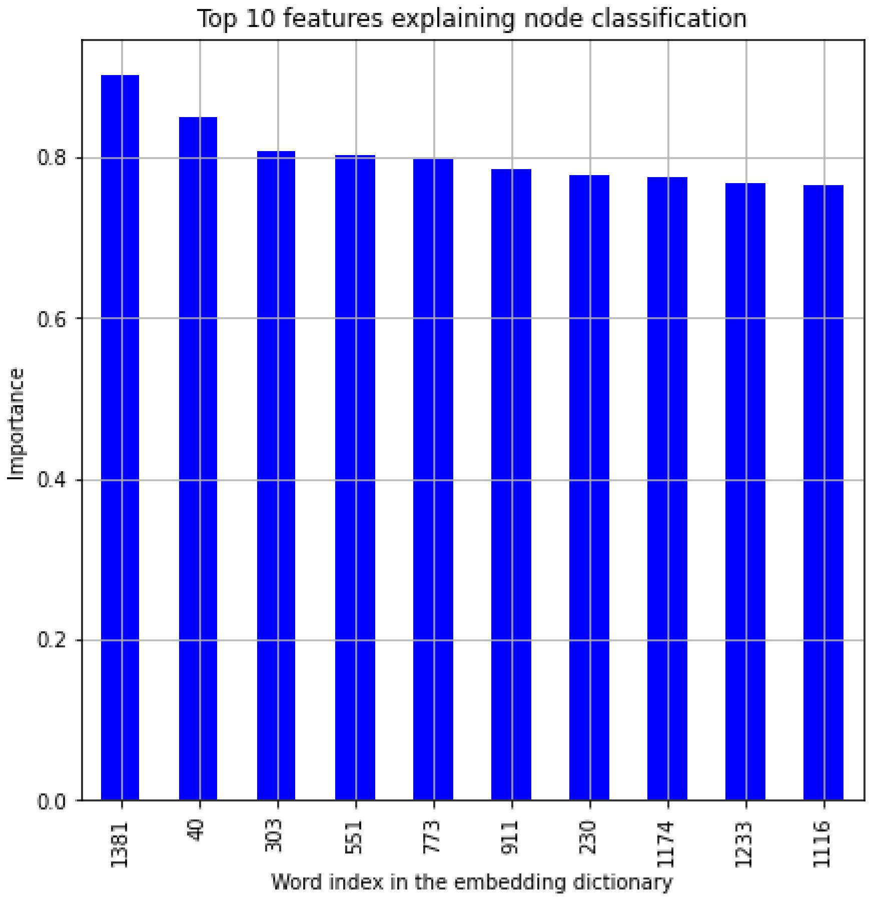


Figure 7. *Top features explaining node classification*

7. Team Effort

Project activity	**Team member contribution percentage**		**Comments**
	Shubham Modi	Oleksandr Shashkov	
Initial research	60%	40%	We made sure we both get sufficient knowledge of Graph Neural Networks in general and working of GNN explainer through reading papers, walking through previous work and sharing knowledge obtained in status meetings.
Datasets selection	60%	40%	Shubham tried out different datasets such as Facebook, Politifact, Gossipcop, webKB, Reddit Binary before finalizing the Twitch Dataset and Oleksandr tried Cora dataset which worked for his task.
Implementation	60%	40%	
Analysis	60%	40%	
Report	30%	70%	Oleksandr worked on the report while Shubham worked on creating and presenting the Presentation infront of the class.
Presentation	70%	30%	
Overall logistics	60%	40%	

8. References

[1] GNNExplainer: Generating Explanations for Graph Neural Networks, Rex Ying, Dylan Bourgeois, Jiaxuan You, Marinka Zitnik & Jure Leskovec, arXiv:1903.03894, 2019, retrieved from: https://arxiv.org/pdf/1903.03894

[2] Parameterized Explainer for Graph Neural Network,
Dongsheng Luo, Wei Cheng, Dongkuan Xu, Wenchao Yu, Bo Zong, Haifeng Chen, Xiang Zhang, arXiv:2011.04573, 2020, retrieved from https://arxiv.org/pdf/2011.04573

[3] GNNExplainer Tutorial, 2020, retrieved from https://github.com/OpenXAIProject/GNNExplainer-Tutorial

[4] gnn-explainer experiments, Anshul Yadav, 2020, retrieved from https://github.com/anshul3899/GNNExplainer-Experiments

[5] The Cora Dataset. Orbifold Consulting. Retrieved from https://graphsandnetworks.com/the-cora-dataset/

[6] Twitch Dataset. Retrieved from https://pytorch-geometric.readthedocs.io/en/latest/modules/datasets.html

[7] Torch Geometric GNN Explainer https://pytorch-geometric.readthedocs.io/en/latest/_modules/torch_geometric/nn/models/gnn_explainer.html

[8] Sci-kit Learn https://scikit-learn.org/stable/

[9] Torch Geometric GNN Explainer https://pytorch-geometric.readthedocs.io/en/latest/_modules/torch_geometric/nn/models/gnn_explainer.html

[10] Matplotlib https://matplotlib.org/

[11] Pandas https://pandas.pydata.org/

[12] PyTorch
https://pytorch.org/