DA-516 Course Project

Network Analysis for Twitch Social Networks Data

Ali Enver Arslan

Mustafa Tufan Keser

Onur Akbaş

Abstract

In the study, user-to-user Twitch network is held as a network analysis topic. Dataset consists of six different countries as six separate network structures. Germany (DE) and Spain (ES) networks are selected for the analysis to overcome the computation problems. For community detection, two different approaches are experimented with. Firstly, the Louvain method is applied and 6 communities are found. Afterward, the K-Means Clustering method is used as an alternative approach and it is seen that the Louvain method fits better in the dataset. Community structure obtained by the Louvain method is selected and a Neural Network classification model is built to predict the communities of each node based on the node embedding vectors and the node features derived from the data. The results of the node classification algorithm performed well with about 85% accuracy score. For the link prediction task, edge embedding vectors are calculated. The average Embedding method is selected among the Average, L1, L2, and Hadamard Embedding methods due to the high computation costs. In the link prediction model, the XGBoost Classifier algorithm is used due to its high performance on large datasets. 70% of the actual links are predicted correctly on the train set and 42% of the actual links are predicted correctly on the Spain network which the model never saw.

Introduction

Twitch is a live streaming platform geared towards gamers that launched in 2011. Since then, it has amassed millions of users, with a total of 3.8 million unique broadcasters as of February 2020.

Twitch offers gamers — or anyone interested in lifestyle casting about other subjects like food or music — the ability to stream their activity and let others watch in real-time. Streams can last anywhere from a minute to eight hours and beyond. One can find a stream by browsing various categories, including specific games. If users find a streamer they can like, you can follow their channel and get activity updates and notifications.

With the outputs of the project, a couple of questions can be explained. Streamers compete with each other and many of them presenting the same content. Who streaming the hottest games at the moment will raise viewers and get more popular? As a result of

the predicted links, there are possible benefits for both sides. For users, they will recommend more related streamers or content. It will help parents to avoid their children from mature contents, as well. For streamers, there are a couple of hits about what should be streamed, how they can raise viewers and their popularity, what are the hottest games, how they make money quickly in the light of predictions.

Dataset

The dataset worked on consists of Twitch user-user networks of gamers who stream in a certain language. Nodes are the users themselves and the links are mutual friendships between them. Vertex features are extracted based on the games played and liked, location, and streaming habits. Datasets share the same set of node features. These social networks were collected in May 2018.

Dataset statistics						
	DE	EN	ES	FR	PT	RU
Nodes	9,498	7,126	4,648	6,549	1,912	4,385
Edges	153,138	35,324	59,382	112,666	31,299	37,304
Density	0.003	0.002	0.006	0.005	0.017	0.004
Transitivity	0.047	0.042	0.084	0.054	0.131	0.049

The detailed information about the network of Germany is given below since it is used for model building purposes.

- Number of nodes : 9,498
- Number of edges : 153,138
- Average degree : 32.2464
- Is directed : False
- Is weighted : False
- Is bipartite : False
- Density : 0.0034

The node features data consists of the following properties:

- Days: number of days active using
- Views: total number of streamings
- New_id: Id
- Lang: streaming language
- Mature: the content of the streaming
- Partner: undefined

Since the data description does not explain the partner feature, it is removed from the analysis.

The degree distribution of the network is shown below. As can be seen from the graph as well, most of the nodes in the network have very few links while a small minority has many links.

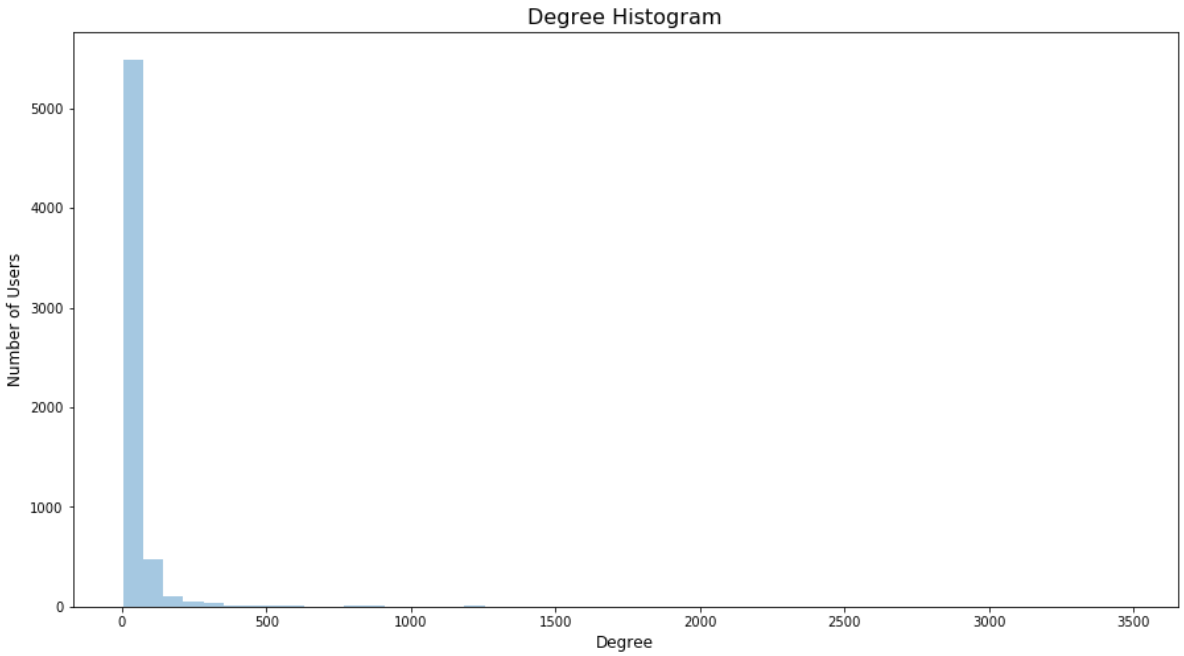


Figure 1: Degree Distribution

As a result, it can be inferred from the degree distribution that the network follows a Power-Law distribution. It can also be said that the network also is in line with the Pareto – Principle of 80 – 20 rule which is a special type of the Power-Law distribution. In other words, most of the users watch the same small group of users in the network.

Since the network has a sparse structure in which most of the nodes are connected with just a few other nodes, a sub-group is selected from the whole network for the remaining analysis.

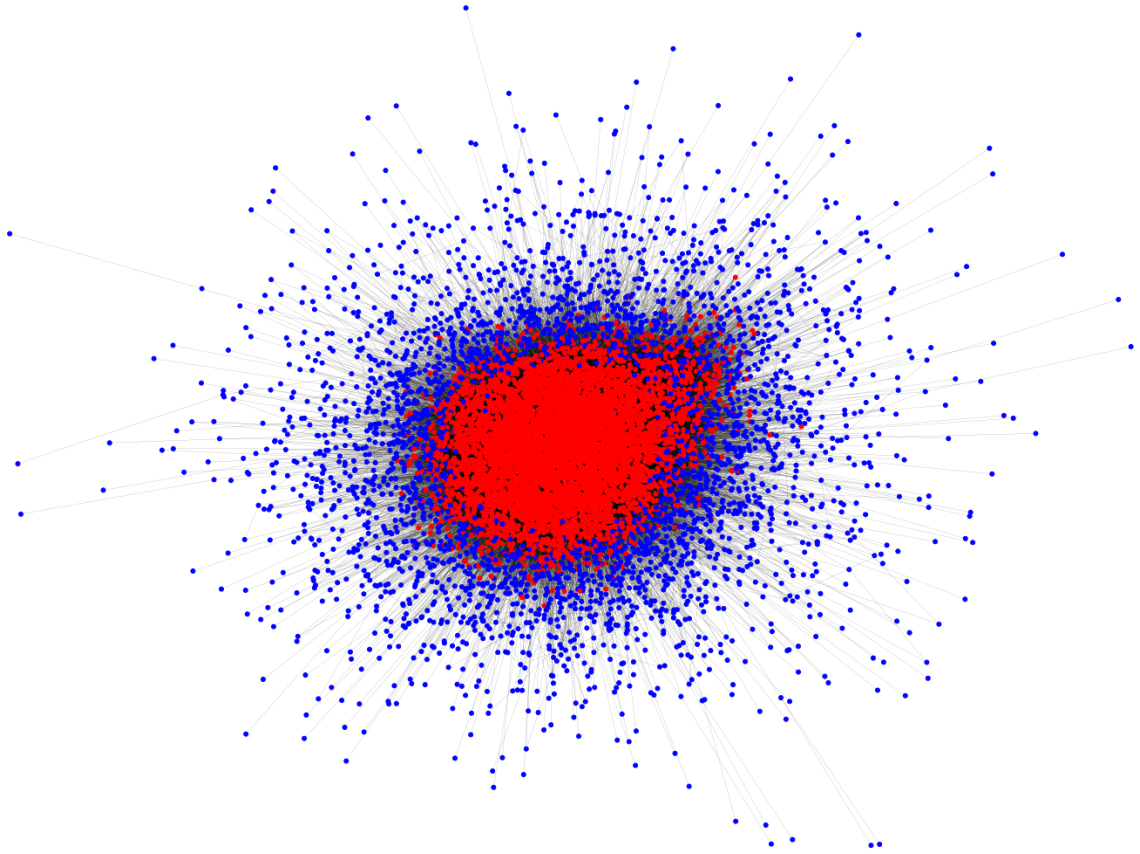


Figure 2: Network Structure

In the graph visualization above, red nodes are the ones having degree 10 and above while the blue nodes are the ones having degree less than 10. A comparison of the results of the red nodes with the whole network is given below.

Network Comparison			
	Original	Filtered (d>=10)	%
Nodes	9,498	6,216	-%34.5
Edges	153,138	138,791	- %9.3
Average Degree	32.2464	44.6560	+%34.3

As can be seen from the table above, the node elimination makes the network more compact while keeping the %91 of the links.

Community Detection

A community is defined as a subset of nodes within the network such that connections between the nodes are denser than connections with the rest of the network.

The detection of the community structure in a network is generally intended as a procedure for mapping the network into a tree. In this tree (called a dendrogram in the social sciences), the leaves are the nodes whereas the branches join nodes or (at higher level) groups of nodes, thus identifying a hierarchical structure of communities nested within each other.

One of the recent and most efficient community detection algorithms is the Girvan-Newman algorithm. However, it is a computationally expensive algorithm since it requires the repeated evaluation, for each edge in the system, of a global quantity, the betweenness, whose value depends on the properties of the whole system.

To overcome the performance issue of the german-Newman method, Cluster-Overlap Newman Girvan Algorithm Optimized (CONGO) method is used with the expectation of having better results, however, since there is no improvement achieved its results are not shown in the paper.

On the other hand, The Louvain method is a simple, efficient, and easy-to-implement method for identifying communities in large networks. The method has been used with success for networks of many different types and sizes up to 100 million nodes and billions of links. The analysis of a typical network of 2 million nodes takes 2 minutes on a standard PC. The method unveils hierarchies of communities and allows to zoom within communities to discover sub-communities, sub-sub-communities, etc. It is today one of the most widely used methods for detecting communities in large networks.

The Louvain method is a greedy optimization method that attempts to optimize the "modularity" of a partition of the network. The optimization is performed in two steps. First, the method looks for "small" communities by optimizing modularity locally. Second, it aggregates nodes belonging to the same community and builds a new network whose nodes are the communities. These steps are repeated iteratively until a maximum of modularity is attained and a hierarchy of communities is produced. Although the exact computational complexity of the method is not known, the method seems to run in time O(N log N) with most of the computational effort spent on the optimization at the first level. Exact modularity optimization is known to be NP-hard.

In this study, the Louvain method is used due to its convenience. The result of the community detection implementation is shown below.

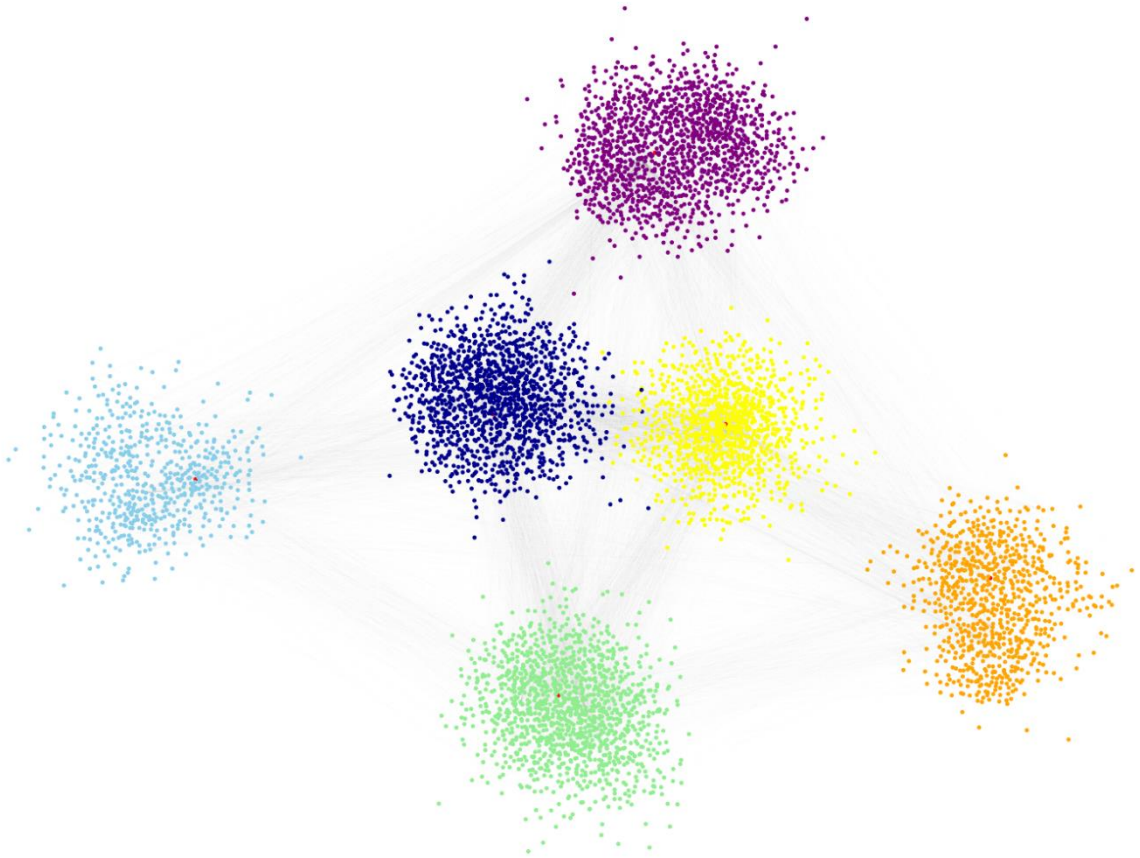


Figure 3: Communities detected using Louvain Method

There are 6 communities detected in the network. The orange, purple, green, and light blue communities are seen to be well separated from the rest of the network. On the other hand, yellow and dark blue communities are relatively closer to each other.

To measure the performance of the communities, modularity is used. The modularity score is calculated as 0.279. although the modularity.

As another community detection method, Kernighan–Lin bipartition algorithm is also implemented. However, the results derived from there are not shown here since the modularity score is just 0.003.

As an alternative approach for detecting communities, clustering algorithms might also be a good choice. To check the performance of a clustering method, the K-Means algorithm is used.

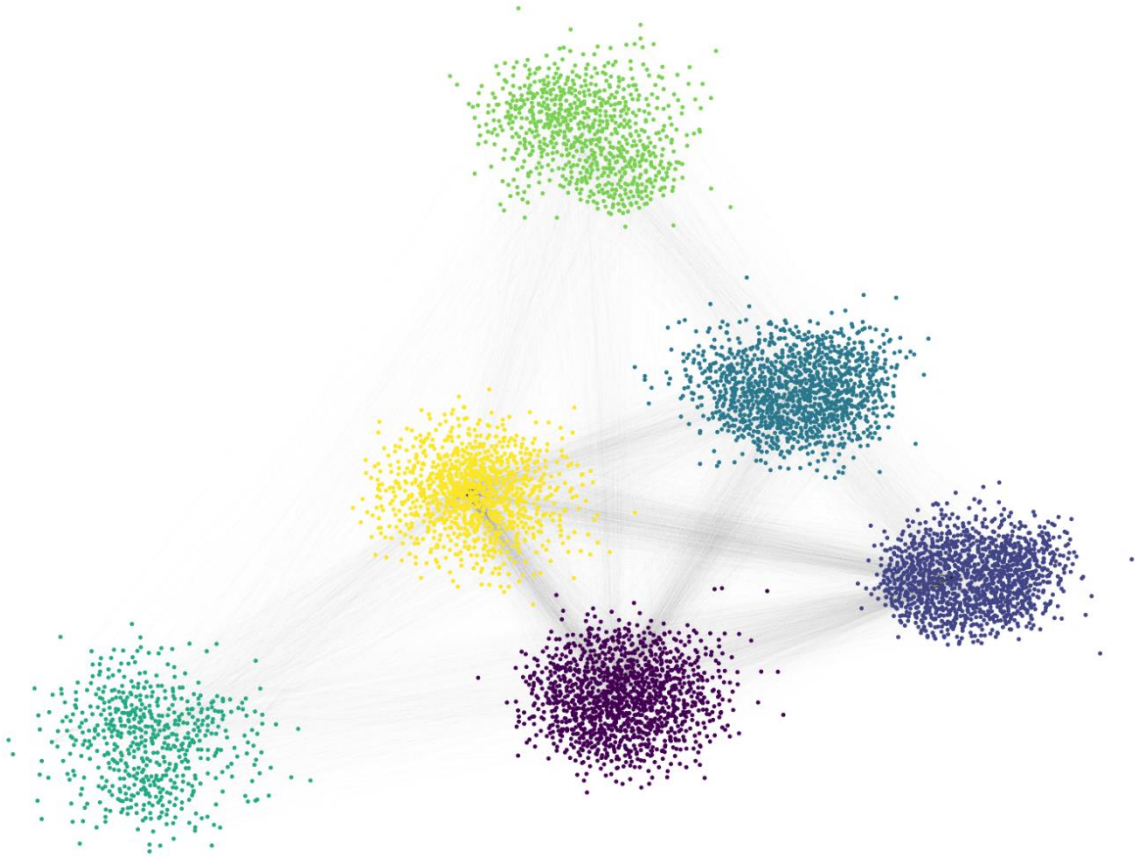


Figure 4: Communities detected using K-Means Clustering

To determine the number of clusters in K-Means, the elbow method is used, however, it did not work well. Hence, the number of clusters is selected the same as the number of communities calculated by the Louvain method. K-Means algorithm can divide communities with a modularity score of 0.278 which is quite close to Louvain. However, for the node classification analysis, the Louvain algorithm is preferred due to its slightly better modularity score.

Node & Edge Embeddings

The node2vec framework learns low-dimensional representations for nodes in a graph by optimizing a neighborhood preserving objective. Besides reducing the engineering effort, these representations can lead to greater predictive power. In this project, parameters are assigned;

- Dimensions=20
- walk_length=20

- num_walks=50
- p = 1
- q = 1
- weight_key = None
- workers=4

It is suggested to use dimensions between the cube root and the square root of the number of nodes. Since the number of nodes in the network is 6216, dimension number between 18 and 78 is meaningful. Due to computation issues, 20 is selected as the number of dimensions. Walk the length of 20 with a window of 10 and 50 number of walks is found appropriate considering the large dataset. *p* and *q* are left at their default values. Since the edges do not have weights there is no weight stated in the node embeddings as well.

For the edge embeddings, different embedding methods, such as Hadamard Embedder, Weighted L1 Embedder, Weighted L2 Embedder, Average Embedder are compared in terms of computation cost and model performance, and Average Embedder Method is used for further analysis due to its success over the model performance. After the Average Embedding method, the dataset was transformed into an edge embeddings dataset from 153138 rows to 19322436 rows and 23 columns. There are embedding methods are used for model preparation with this prepared dataset.

Node Classification

The node Classification section aims to predict the attributed classes (communities) to each node from the feature set derived from the node embeddings via node2vec and node-related features from the dataset.

After Node Embedding vectors and node-related features are gathered together, a Random Forest algorithm is implemented. The accuracy of the Random Forest model on the test set is acquired at around %80.7.

Taking the size of the dataset into account more advanced model is considered to improve the performance and an Artificial Neural Network (ANN) model is built. The structure built for the ANN model is shown below.

Layer (type)	Output Shape	Param #
input_layer_1 (Dense)	(None, 1000)	24000
input_layer_2 (Dense)	(None, 1000)	1001000
input_layer_3 (Dense)	(None, 500)	500500
output_layer (Dense)	(None, 6)	3006

Total params: 1,528,506
Trainable params: 1,528,506
Non-trainable params: 0

The ANN model performance on the train set is calculated as %92 and on the test set, the accuracy falls to %85. For a 6-Class model, it can be said that the performance of the model is highly successful.

The confusion matrix for the classification model is shown below.

The high performance of the node classification model is also an indicator of the quality of the community detection implemented in the previous step. With the given feature set, the communities can both separated from each other and they can also be predicted successfully.

Link Prediction

In the link prediction section, the purpose is to build a machine learning model that can predict the true edges among the nodes from all possible links in the whole network. This means a huge computation cost in a large dataset and model selection is considered with this information in mind. The steps of the link prediction model are explained below.

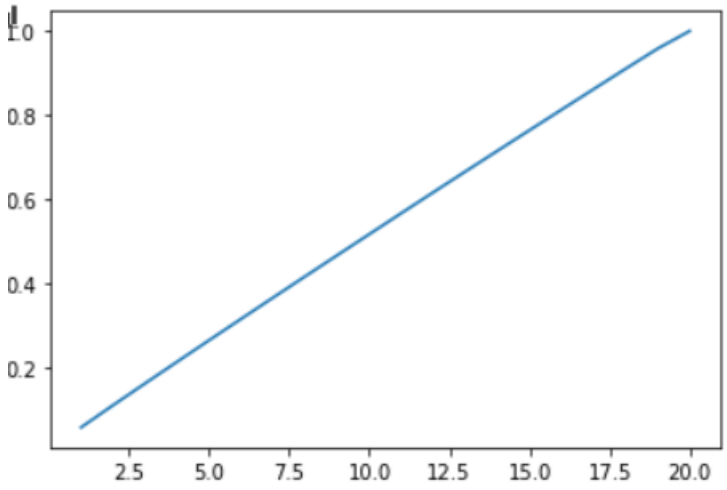
1. **Model Preparation**

    a. **Train-Test-Split:** Dataset is split into train and test data by using train test splitter in sci-kit-learn, given parameters are;

        i. test_size=0.25
        ii. random_state=42
        iii. stratify=y

    Stratify parameter is used to make more robust predictions because of the imbalances of the dataset.

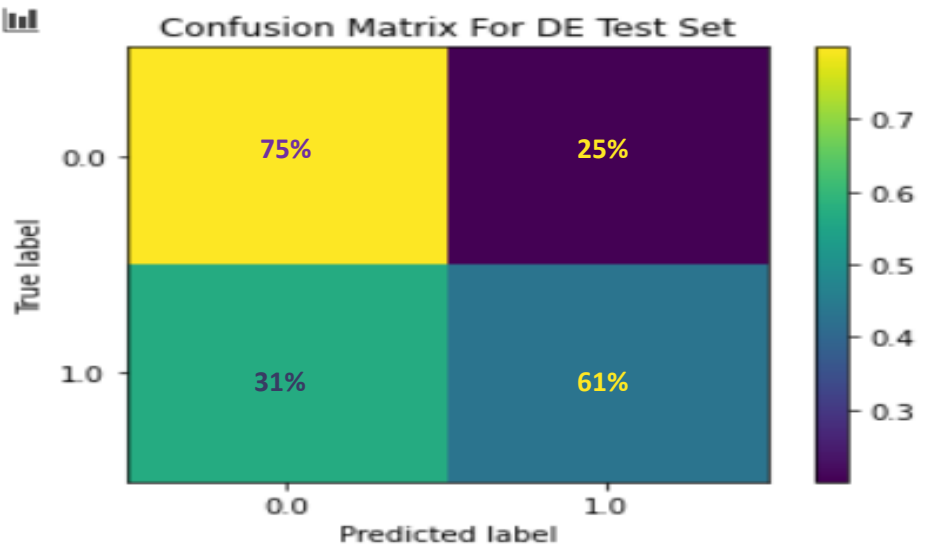
    b. **Principal Component Analysis (PCA):** Dimensionality reduction involves reducing the number of features in modeling data. PCA is a technique that can be used to perform dimensionality reduction. PCA is tried to see if it helps as a dimensionality reduction technique in this project because of the large size of this dataset and computation time concerns. According to the result of the PCA 16 components can explain around %80 of the total variance.



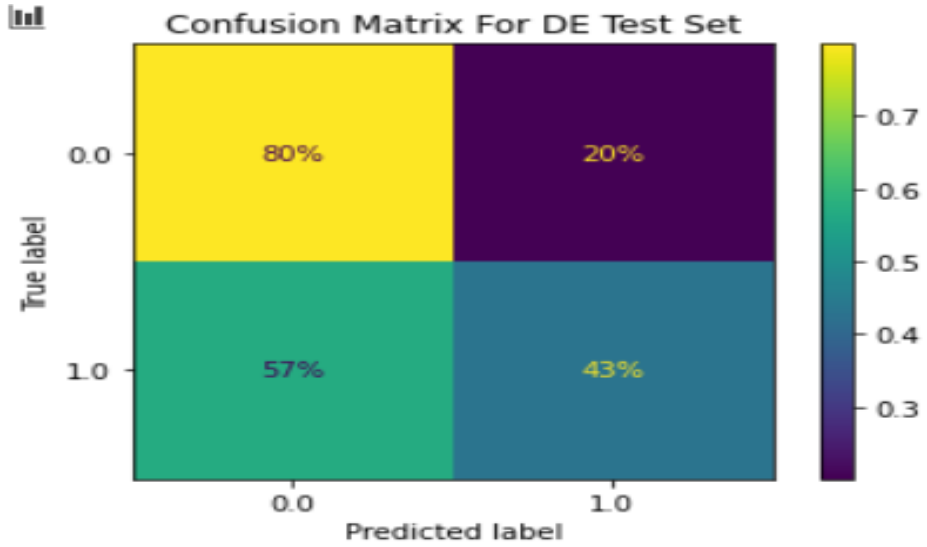
c. **Model Selection:** XGBoost classifier model is used since it provides a parallel tree boosting (also known as GBDT, GBM) that solves many data science problems in a fast and accurate way. It also has early stopping parameters that can help reduce time in calculations.

d. **Model Preparation:** As an initial model, the links between Twitch users in Germany are predicted and the results obtained are below. Precision and recall parameters are acceptable after using XGBoost classifier with the prepared dataset. In the next steps, links between Twitch users from Spain are predicted.

The result from PCA implemented XGBoost Classifier is shown below:



The result from XGBoost Classifier without PCA implementation is shown below:



The confusion matrix above belongs to the model implemented without feature scaling. The model with feature scaling gave almost the same results so, it is not shown in the paper again.

2. Comparison of the Confusion Matrix with different Zones:

The link Prediction model is built on Germany data and it is now intended to predict a network it did not see. To fulfill this purpose, the Spain network is chosen.

Results from Spain network link prediction are shown below: