Production and Evaluation of Node Embeddings (Node2Vec)

Table of Contents

Tables

Figures

1 Introduction

This project aims to visualize and study results for vector representations (node embeddings) created based on the *polbooks* network (V. Krebs, unpublished, http://www.orgnet.com). More specifically, Link Prediction and Clustering will be performed on the given node embeddings, using different metrics for each technique, while visualizations are also produced to draw conclusions.

This file has the following format:

```
...
...
node [
    id 0
    label "1000 Years for Revenge"
    value "n"
  ]
node [
    id 1
    label "Bush vs. the Beltway"
    value "c"
  ]
...
...
```

```
...
...
  edge [
    source 1
    target 0
  ]
  edge [
    source 2
    target 0
  ]
...
...
```

The nodes represent books on US politics sold by Amazon. The value values on the left take values "l", "n" and "c", representing liberal neutrals and conservatives respectively. Therefore, it is taken for granted that there are 3 communities, the so-called ground truth, in the archive.

For Link Prediction the performance of various categorizers will be compared in terms of accuracy, precision and recall, while for Clustering when applying K-Means to embeddings the performance in terms of modularity and purity will be compared.

2 Link Prediction

2.1 Parameters Selection

The original *polbooks.gml* contains 105 nodes and 441 edges. The following methods were applied to the largest coherent component of the graph, so a new subgraph was created that eventually contains 105 nodes and 441 edges.

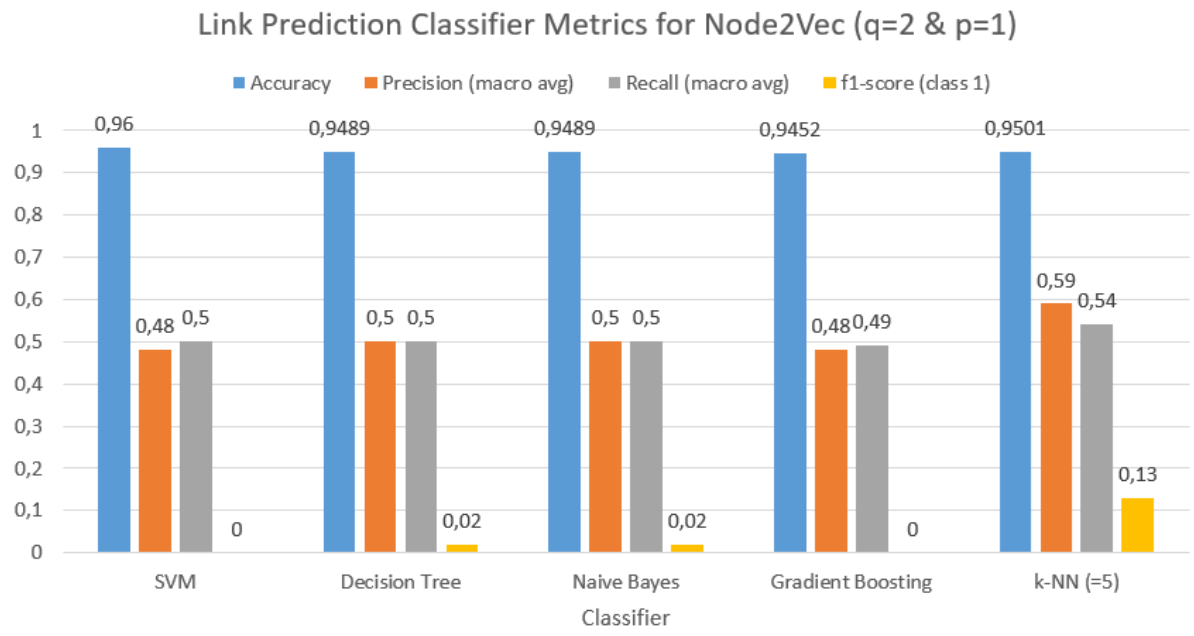
To generate the vector representations, the parameters *dimensions*, *num_walks* and *workers* were used together with values of 64, 40 and 4 respectively. The *dimensions* parameter refers to the dimension of the embedding, *num_walks* represents the number of walks per node and *workers* was set to 4 in order to run in parallel on the computer performing the experiments. Finally, three pairs of *p* and *q* values were chosen to create the three separate Node2Vec vector representations (p=2 & q=1, p=0.5 & q=1, p=1 & q=1).

After creating the embeddings, this information was passed to three dataframes with an identifier book_id for each book in the network, in order to proceed with the Link Prediction process where using a classifier we will try to predict whether an edge exists (class 1) or not (class 0).

2.2 Commenting on the Results

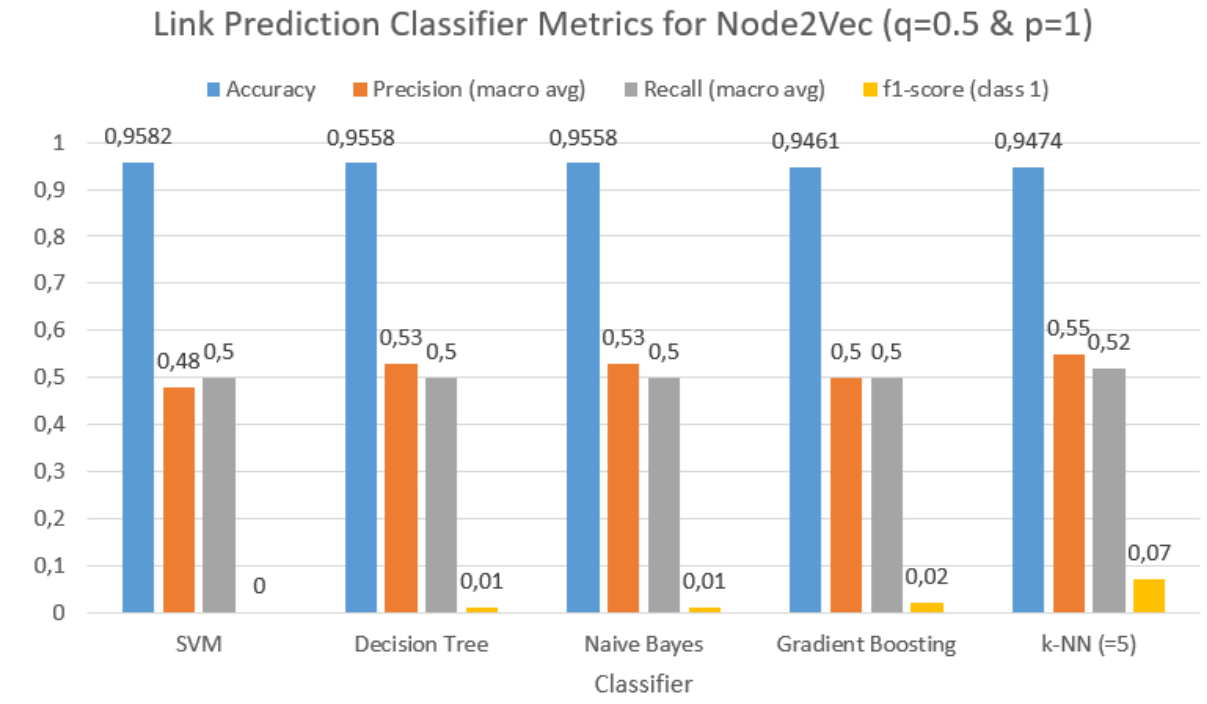
To evaluate the whole process, we are asked to measure accuracy, precision, recall for the selected classifier. As will be seen in the graphs below, we are asked to make predictions on an imbalanced dataset where class 1 has very little support, so these three metrics may not give a correct assessment of the classifier's output. Therefore, in order to draw some overall conclusion, we will add the F1-Score metric to the "disadvantaged" Class 1 as well as metrics from not only one classifier but from many. More specifically, Link Prediction was applied to SVM, Decision Tree, Naive Bayes, Gradient Boosting and k-NN (=5).

Figure 2.1: Comparison of Node Embedding Classifiers with q=2 and p=1



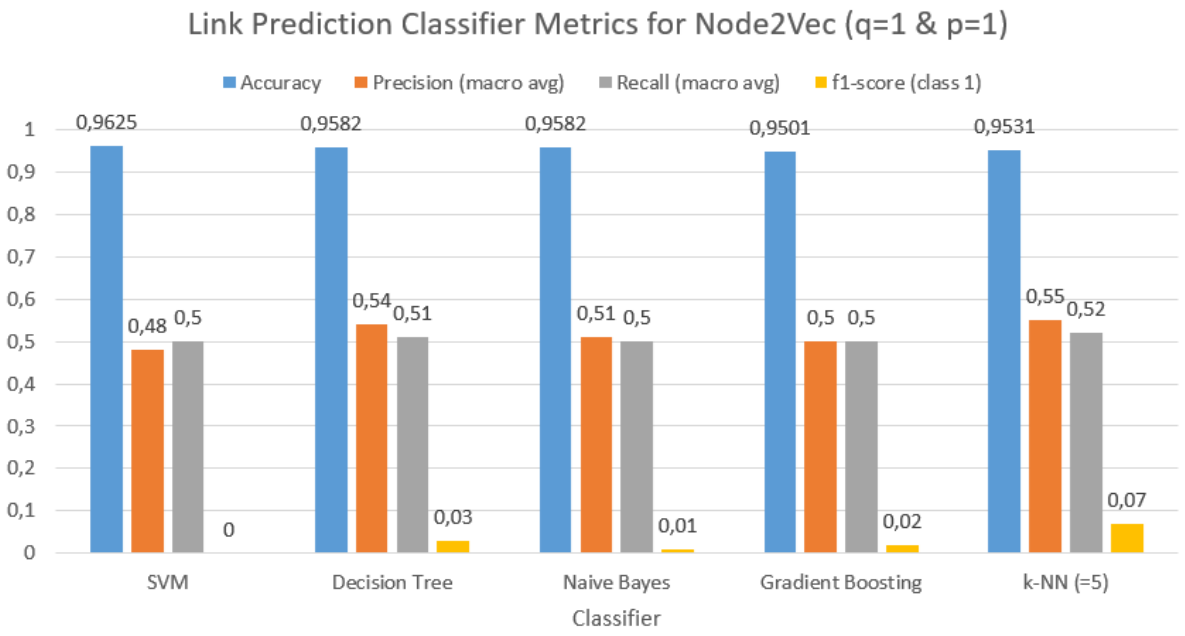
***Comment*:** The graph above shows that overall for all categorizers the Accuracy, Precision and Recall values are in the same range. Also, the only classifier that manages to predict a somewhat higher percentage of class 1 is K-nearest neighbors with 0.13.

Figure 2.2: Comparison of Node Embedding Classifiers with q=0.5 and p=1



***Comment*:** As in *Figure 2.1* the results appear without any particular differences.

Figure 2.3: Comparison of Node Embedding Classifiers with q=1 and p=1



***Comment*:** Following on from the two previous graphs, *Graph 2.3* continues to show that k-NN manages to predict significantly better in terms of Class 1, while the values of the other three metrics are in the same range.

Table 2.1: k-NN Classification Report of Node Embedding with q=2 and p=1

Metrics for k-nearest neighbors Classifier

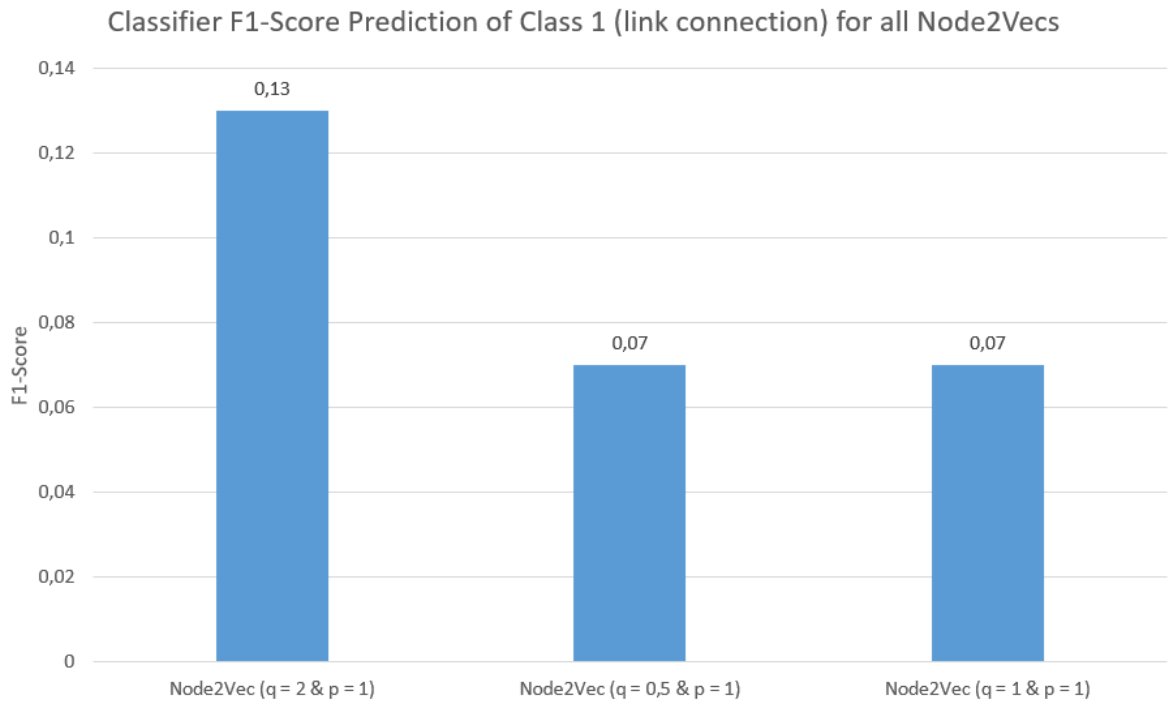
Testing Accuracy: 0.95010120920918984 2805

Classification Report:

	precision	recall	f1-score	support
0	0.96	0.99	0.97	3176
1	0.21	0.09	0.13	132
accuracy			0.95	3308
macro avg	0.59	0.53	0.55	3308
weighted avg	0.93	0.95	0.94	3308

Comment: *Table 2.1* shows the Classification Report of the Node2Vec vector representation with q=2 and p=1 where the imbalance between the two classes is clearly shown. Here, it can be seen that the support of the redundant class is 24 times larger, leading us inevitably to both low precision and recall values. Nevertheless, the accuracy remains at high values since the classifier correctly predicts the largest number of the test set. This particular classification report is the one that showed the highest values among all experiments.

Figure 2.4: Comparison of F1-Score of Class 1 Node Embeddings for k-NN (=5)



***Comment*:** As discussed previously, the F1-Score metric for the smallest class is visualized to check which classifier and on which generated node embedding out of the three, showed the highest values. Although the difference extracted from the classification reports is not large, the maximum value was 0.13 and was displayed by applying Link Prediction with k-NN to Node2Vec with q = 2 and p = 1.

3 Clustering

3.1 K-Means

Then, clustering with K-Means (K=3) is applied to the three Node2Vec embeddings, measuring Modularity and Purity, in order to try to identify clusters and gain some knowledge about whether there are any patterns for the network structures we study. In parallel, the t-SNE method is used to visualize the vector representations of the nodes in two dimensions, comparing the clustering results with the actual class of nodes.

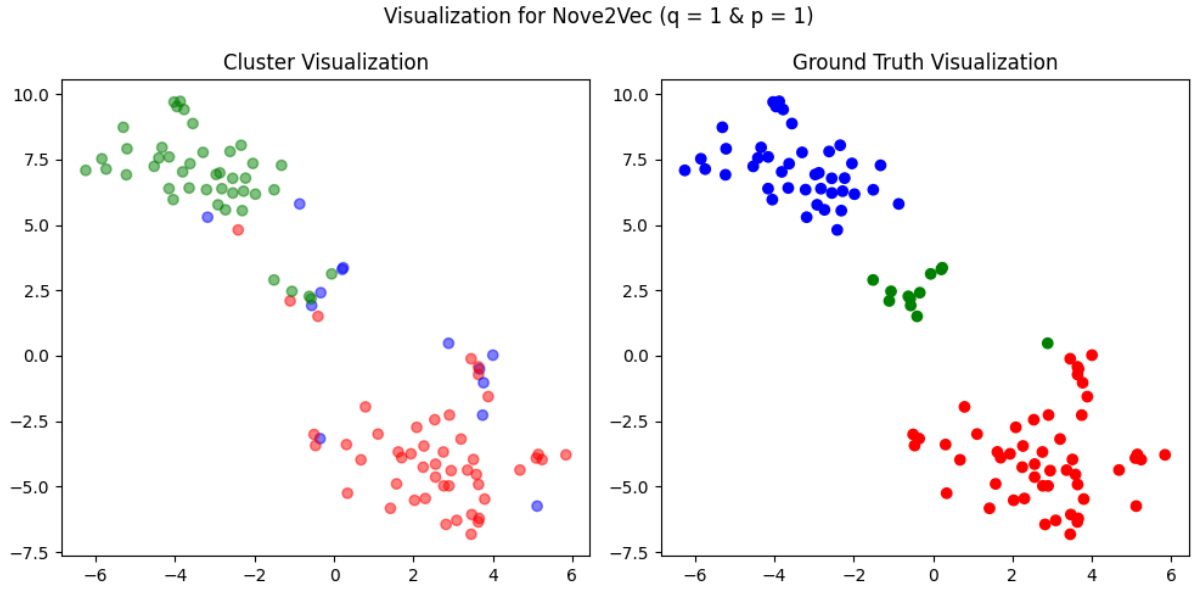
First, Modularity and Purity were measured for the ground truth communities of the graph, giving values of 0.4149 and 1 respectively. Purity measures the degree to which the detected communities contain only nodes from a single ground truth community, thus indicating whether the clustering results are homogeneous compared to ground truth and the communities are coherent. Obviously, a value of 1 for the purity of ground truths is reasonable (or even unnecessary to measure). Modularity is a measure of how well a network is partitioned into communities, where higher values indicate better community structure in the network.

To properly compare the three Node2Vec vector representations created, the parameter values p and q, and how they affect the results, need to be further analyzed. Node2Vec uses random biased walks through the network to explore the neighborhood of each node and determine its content. By adjusting its parameters, in particular the values of p and q, we can control the nature of the walks and the resulting embeddings.

The parameter p affects the likelihood of the walk returning to the current node, while q determines the direction of the walk away from the original node. Lower values of p lead to embeddings that focus on the nearby neighborhood, resembling a breadth-first search (BFS) approach. On the other hand, lower values of q lead to embeddings that capture the wider neighborhood, similar to a depth-first search (DFS) approach.

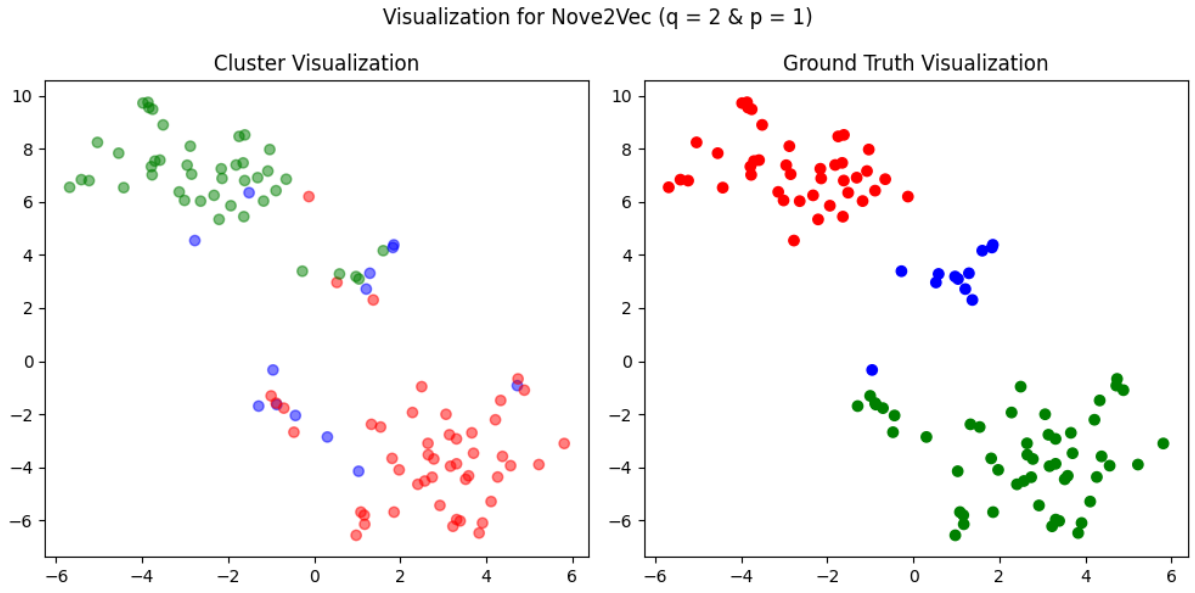
3.2 Commenting on the Results

Figure 3.1: K-Means Clustering visualization of Node Embedding with q=1 and p=1



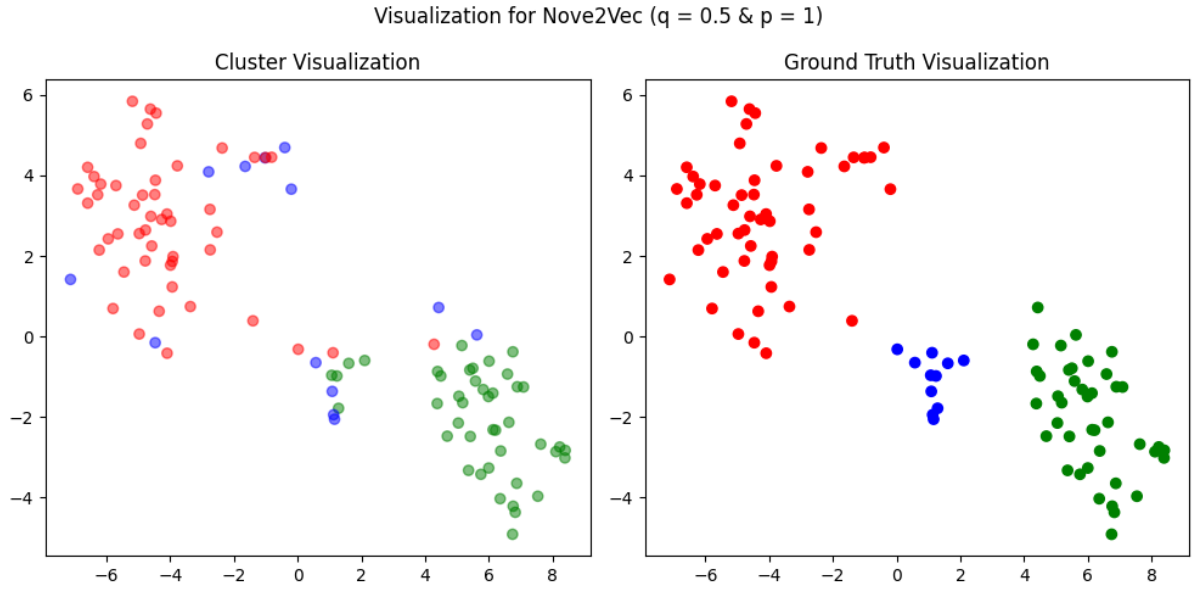
***Comment*:** the left shows the clustering result on the embeddings of the values studied, while the right shows the ground truths. Here, Modularity, as we will see below in *Figure 3.4*, is quite close to the values of the original graph while Purity is quite high reaching almost 85% which indicates clusters that are largely uniform and homogeneous compared to the ground truth.

Figure 3.2: K-Means Clustering visualization of Node Embedding with q=2 and p=1



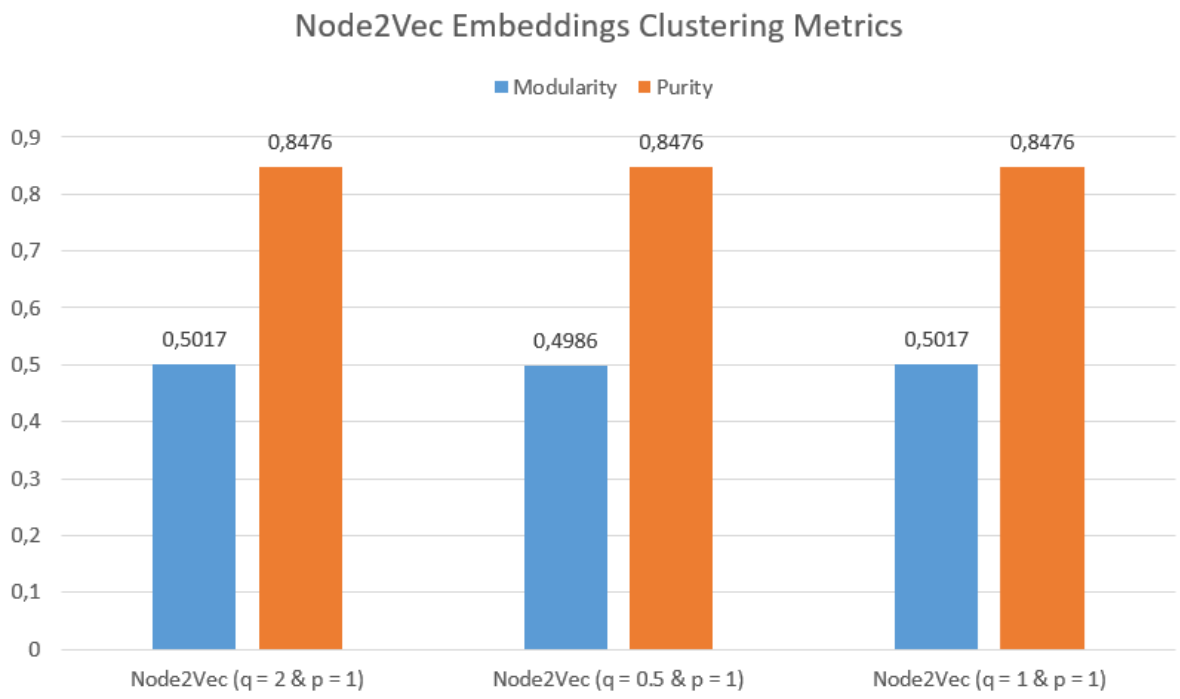
***Comment*:** The results for these values do not show any particular effect, since they are also high (0.5017, 0.8476) indicating a homogeneous cluster.

Figure 3.3: K-Means Clustering visualization of Node Embedding with q=0.5 and p=1



***Comment*:** In *Figure 3.3, although* the metric results are no different, the result of the visualizations clearly shows that the low *q* value led to a wider neighborhood search compared to the previous two graphs.

Figure 3.4: Comparison of Clustering Results of Node Embedding



***Comment*:** Regarding the Modularity and Purity metrics for the three different Node2Vec embeddings created, it is clear that there is absolutely no difference between the different values of *p* and *q*.

Finally, it is important to point out that because it seemed strange that the values look extremely similar, despite the parameter variations, different Node2Vec embeddings were tested with repeated tests for the parameter's *dimensions*, *num_walks*, *workers*, *window* and *min_count* without any significant change.