REPORT - NODE2VEC Clustering

Bolla Nehasree - CS22BTECH1012
Nethi Keerthana - CS22BTECH11043

1. Data Preprocessing

- The transaction data from Payments.xlsx was converted into a directed graph using NetworkX to model transactions between nodes. Each edge had a weight (transaction amount), but raw data often contains noise, such as one-off transactions that don't reflect meaningful patterns. Filtering edges with fewer than 2 transactions (min_transactions=2) removed such noise, ensuring the graph captured more significant relationships.
- Isolated nodes were removed because they don't contribute to clustering.

2. Node2Vec Embeddings

- Node2Vec generates node embeddings by simulating random walks on the graph and learning representations using a SkipGram model. The parameters p=1.0, q=1.0, and dimensions=128 were chosen to balance local and global exploration without bias, and 128 dimensions provide a good trade-off between expressiveness and computational efficiency.
- **Parameters explanation**:
  - p=1.0, q=1.0: Equal weighting ensures unbiased exploration, suitable for a transaction graph where no prior assumption about local/global structure was made.
  - walk_length=10, num_walks=80: Ensures sufficient walks to capture the graph's structure.
  - window_size=5, epochs=10, learning_rate=0.01: Standard values for SkipGram training, balancing training time and model performance.

3. Dimensionality Reduction (PCA)

- The 128-dimensional embeddings are too high-dimensional to visualize directly. PCA reduced them to 2D for scatter plots, allowing visual inspection of the clustering results. This step doesn't affect the clustering itself but helps in interpreting the results.

4. Optimal Number of Clusters (k)

- **Silhouette Score**: Measures how well-separated clusters are. The plot peaked at k=2, indicating 2 clusters as optimal.

5. Clustering and Evaluation (KMeans and Agglomerative)

- KMeans and Agglomerative Clustering were applied to group nodes into clusters based on their embeddings. KMeans is a partitioning method that minimizes variance within clusters, while Agglomerative Clustering is hierarchical, merging nodes based on similarity. Both were evaluated using:
  - **Silhouette Score**: Measures cluster cohesion and separation (0.4283 for KMeans, 0.4472 for Agglomerative).

6. Aggregation policy:

- Average Weight: Using **average weight aggregation,** computes the total sum of transaction amounts divided by the number of transactions. Average weight offers a more representative measure of typical interaction strength between nodes, allowing the Node2Vec algorithm to generate random walks that better reflect meaningful relationships in the graph. This leads to higher-quality node embeddings that preserve the core structure of transactional behavior, resulting in improved clustering performance. Moreover, average weighting reduces sensitivity to noise and outliers, making it more robust across different types of nodes and interactions.

7. Hyperparameter Tuning

- **p and q:**
  - High p (e.g., p =1, q = 0.25): This discourages revisiting previous nodes and walks behave more like DFS, which is good for role similarity.

  - High q (e.g., q = 1, p = 0.25): It walks stay close (BFS) and walks get stuck in small clusters, leading to bad generalization.

- **Low or high dimensions:**
  - Low dimensions (64): It can't encode enough structure.

  - High dimensions (256): It causes overfitting or unnecessary noise; hard to train, especially with a limited context size.

- **Long walk lengths (40):**

  - Long walks lead to noisy contexts, embedding becomes diluted over unrelated nodes.

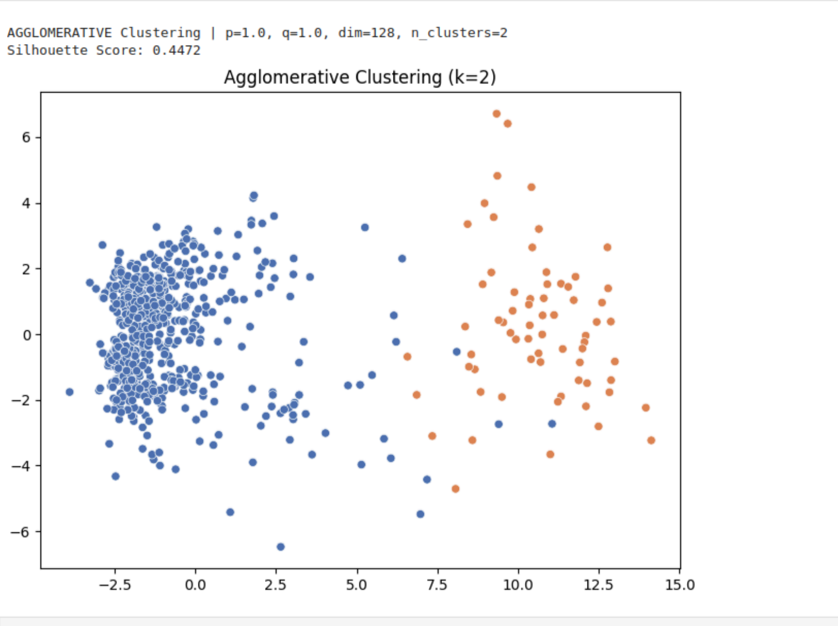
- **Learning rate and epochs:**
  - High learning rate (0.025): Causes instability or divergence.

  - Too low (0.001): Converge too slowly.

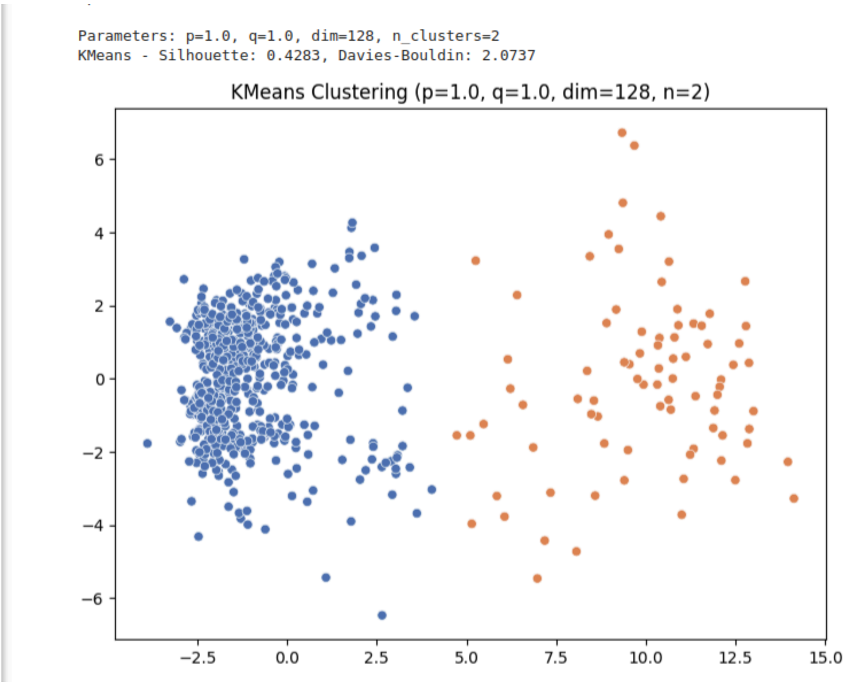
  - Too many epochs (30): Overfit on frequent co-occurrence pairs, hurting generalization to unseen patterns.

- **Aggregation policy:**
  - Sum of edge weights: Biased toward rich nodes, which may dominate embeddings and obscure community structure.

  - Edge counts: Ignores transactional volume—two nodes with 10 tiny payments vs one node with 1 huge payment will look the same, which is misleading.
- **Clustering Results:**



```
AGGLOMERATIVE Clustering | p=1.0, q=1.0, dim=128, n_clusters=2
Silhouette Score: 0.4472
```



```
Parameters: p=1.0, q=1.0, dim=128, n_clusters=2
KMeans - Silhouette: 0.4283, Davies-Bouldin: 2.0737
```