Graph-based Clustering and K-mean Clustering

ABSTRACT : This research focuses on exploring and comparing two distinct approaches to clustering analysis—Graph-Based Clustering and K-Means Clustering—applied to a specific dataset. The dataset chosen for this study is related to Titanic survival predictions, a classic dataset in machine learning. Graph-Based Clustering methods, including techniques such as k-Nearest Neighbours (KNN) and Epsilon Neighbourhood Graph, leverage the inherent relationships between data points to form clusters. Additionally, minimum spanning trees are employed through algorithms like Prim's and Kruskal's to visualize and analyse the resulting clusters. On the other hand, K-Means Clustering, a centroid-based algorithm, partitions the data into clusters based on the minimization of within-cluster variance. The research involves implementing these clustering techniques, evaluating their performance, and comparing the obtained clusters. The study also considers the application of visualization tools to enhance the interpretation of the clustering results. Through this investigation, insights into the strengths and limitations of each clustering approach are gained, contributing to a comprehensive understanding of their applicability in real-world scenarios.

INDEX TERM : Graph-Based Clustering, K-Means Clustering, Titanic Dataset, k-Nearest Neighbors (KNN), Epsilon Neighborhood Graph, Minimum Spanning Tree, Prim's Algorithm, Kruskal's Algorithm, Centroid-Based Clustering, Data Visualization, Machine Learning, Data Clustering Unsupervised Learning, Feature Engineering, Dimensionality Reduction, Predictive Modeling, Titanic Survival Prediction, Data Analysis, Evaluation Metrics (e.g., Silhouette Score, Confusion Matrix)

I.INTRODUCTION

Graphs : Graphs are represented as set of vertices and edges. A graph is a mathematical representation of a set of objects, where some pairs of the objects are connected by links. In the context of your project, a graph may represent relationships or connections between different data points in the Titanic dataset.

Clustering: Clustering is a technique in machine learning and data analysis that involves grouping similar data points together based on certain characteristics or features. The goal is to create homogeneous groups or clusters, making it easier to understand patterns and trends in the data.

Graph-Based Clustering: Graph-based clustering is a clustering approach that utilizes the connections or relationships between data points, represented as a graph. It often involves algorithms that partition the graph into subsets, where nodes within the same subset are more strongly connected.

K-Means Clustering: K-Means is a popular centroid-based clustering algorithm. It partitions the data into 'k' clusters, where each cluster is represented by its centroid. Data points are assigned to the cluster whose centroid is closest to them.

K-Nearest Neighbours (KNN):

KNN is a supervised or unsupervised algorithm that classifies a data point based on its neighbours. In the context of graph-based clustering, it may be used to define connections between data points.

Epsilon Neighbourhood Graph: An epsilon neighbourhood graph is a graph representation where edges are formed between points that are within a certain distance (epsilon) of each other. It's commonly used in graph-based clustering algorithms.

Minimum Spanning Tree: A minimum spanning tree is a tree that spans all the nodes in a graph with the minimum possible sum of edge weights. It's often employed in graph-based clustering to identify essential connections between data points.

Data Visualization: Data visualization involves creating graphical representations of data to better understand patterns, trends, and relationships. In your project, it likely includes visualizing clusters, graphs, and other relevant information.

Accuracy: Accuracy is a metric used to measure the performance of a clustering model. It represents the ratio of correctly predicted instances to the total instances.

Confusion Matrix: A confusion matrix is a table used to evaluate the performance of a classification algorithm. It compares the predicted values with the actual values, providing insights into the model's accuracy, precision, recall, and other metrics.

Graph Construction Methods:

- K-Nearest Neighbours (K-NN)
- Epsilon Neighborhood Graph
- Minimum Spanning Tree

K-Nearest Neighbours (K-NN) Graph:

K-Nearest Neighbors Graph is a data structure and methodology that leverages the concept of proximity to establish relationships among data points. It is particularly prominent in clustering, classification, and pattern recognition tasks. The primary idea behind the KNN Graph is to connect each data point to its k nearest neighbors, creating a network that reflects the intrinsic relationships in the data.

Connectivity Based on Proximity: In a KNN Graph, each data point is linked to its k nearest neighbors. The notion of "closeness" is often defined using distance metrics, such as Euclidean distance, Manhattan distance, or other similarity measures.

Graph Construction: The construction of a KNN Graph involves computing pairwise distances or similarities between data points. The k nearest neighbors for each point are then identified, and edges are established accordingly. This process results in a graph representation of the data, where nodes represent data points, and edges signify proximity relationships

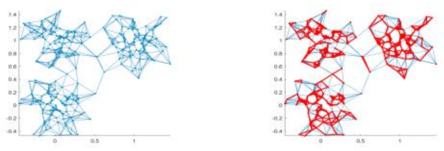


Fig 01: K-NN Graph

Epsilon Neighborhood Graph :

For considering graph:- first define epsilon value from data object and using epsilon value (consider as radius) and draw a circle. And data object is treated as center of circle. Whatever object which are fallen within the radius of epsilon. All those objects can be consider as neighbors. And those nearest neighbor only that will be joining them that is the edges will be available only between those nodes .

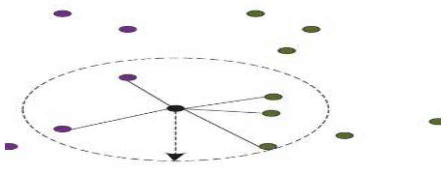


Fig 02: Epsilon Neighborhood graph

Consider only those points. That are present inside the circle .

Advantage of K-NN Graph :

- Flexibility: KNN Graphs adapt well to datasets with varying densities and non-uniform structures.
- Local and Global Patterns: By adjusting the parameter k, KNN Graphs can capture both local and global patterns in the data.
- Intuitive Interpretation: The edges in a KNN Graph directly represent proximity, providing an intuitive interpretation of relationships among data points.

Disadvantage of K-NN Graph :

- Difficulty Handling Missing Data: KNN Graphs may face challenges in handling missing data. Imputation strategies are required to deal with missing values, and the choice of imputation method can impact the graph structure.
- Boundary Effects: The definition of neighborhood in KNN Graphs is sensitive to the data distribution, especially near cluster boundaries. This sensitivity might lead to the creation of edges that do not accurately represent the true relationships between points.

Advantage of Epsilon Neighborhood Graph :

- Flexibility in Capturing Local Structures: The Epsilon Neighborhood Graph is effective in capturing local structures within a dataset. By considering points within a specified distance (ε) as neighbors, it can represent the local relationships and density variations in the data.
- Robust to Noise: It tends to be robust to noise and outliers in the dataset. The graph construction process allows for some level of flexibility in including or excluding points based on their proximity, which can help mitigate the impact of noisy data.
- Adaptability to Varying Densities: Epsilon Neighborhood Graphs adapt well to datasets with varying densities. In regions of high data density, the graph can capture more connections, while in sparse regions, it may have fewer edges, providing a representation that aligns with the underlying density of the data

Disadvantage of Epsilon Neighborhood graph :

- Sensitivity to Parameter ε: The performance of the Epsilon Neighborhood Graph is highly dependent on the choice of the distance parameter ε. Selecting an inappropriate value can lead to either over-smoothing (connecting distant points) or over-focusing (isolating points), impacting the quality of the graph representation.

Steps in Graph based clustering :

1. Determine a Minimal Spanning Tree (MST)
2. Delete branches iteratively
3. Each connected component is cluster .

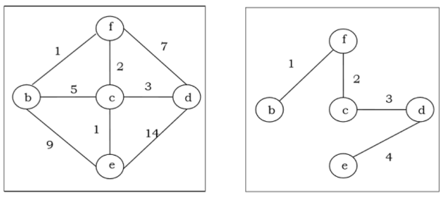


Fig 03: Minimum Spanning Tree(MST)

Determine a Minimal Spanning Tree :

Spanning Tree: Spanning tree is a sub-graph of a graph having all vertices, but only n-1 edges .

Graph : G(V,E) G→ Graph

V → vertex, E → Edges

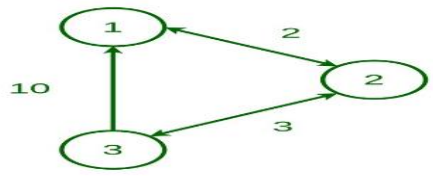


Fig 04: Graph

Spanning Tree : S(V^1 , E^1)

S→ Spanning Tree, V^1→ vertex of ST,

E^1→ Edges of ST .

Relationship between graph and spanning tree : S is subset of G . V^1=V, $|E^1|=|V|-1$. E^1 is subset of E .

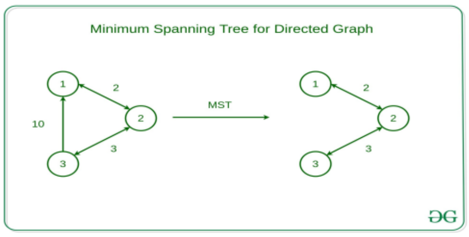


Fig 05: Spanning Tree .

A graph can have more than one spanning tree . Spanning tree should not contains any cycle . Spanning tree should not be disconnected

How many Spanning Tree are possible in given graph :

$^{|E|}C_{|V|-1}$ - No of cycles

If you have a weighted graph and you get different spanning tree. The cost of spanning tree may be variant . you can get different cost of spanning tree . Can I find out Minimum cost spanning tree. Which gives minimum?

➔ Yes we can find out.

First method is : you can try all possible spanning tree and from that you can check which tree has minimum cost . After that you can pickup the answer .

Greedy method are available for finding minimum cost Spanning Tree

Two methods are possible :

1. Prim's Algorithms
2. Kruskal's Algorithms

Prim's Algorithms : Prim's algorithm is a greedy algorithm used to find the minimum spanning tree (MST) of a connected, undirected graph. The minimum spanning tree is a subset of the edges that connects all the vertices in the graph without any cycles and with the minimum possible total edge weight.

Algorithm Steps :

1. Start with an arbitrary vertex (can be randomly chosen).
   - This vertex becomes the first vertex in the minimum spanning tree.
2. Add the shortest edge that connects a vertex in the MST to a vertex outside the MST.
   - This ensures that the tree grows while minimizing the total edge weight.

3. Repeat step 2 until all vertices are included in the MST. Continue adding the shortest edges until all vertices are connected in a tree-like structure with no cycles

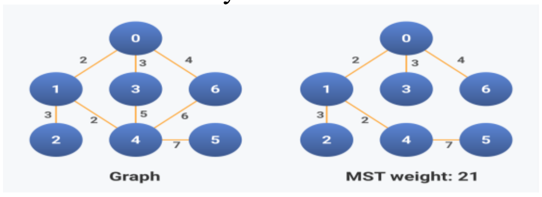
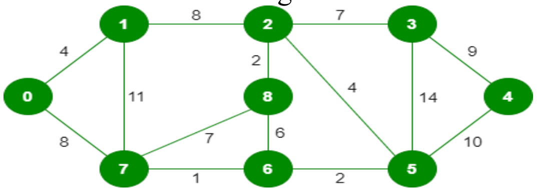


Fig 06 : Prim's Spanning Tree .

Kruskal's Methods: This also follow greedy approach . Always select a Minimum cost edge but if it is forming a cycle don't included in the solution discard that edge.



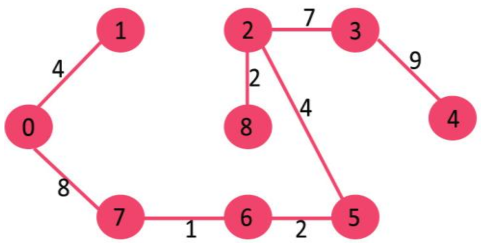


Fig 07: Kruskal's Spanning Tree

Time Taken By Kruskal's Algorithms:

It select a minimum cost of edge. i.e |E|. How many edge it will selected that is |V| -1.

Total time will be O(|V||E|). V=vertex

→O(n.e) n=no of verted, e=edges

- ➔ O(n2) if n==e .
- ➔ Total time take=O(n2). If both are n .

Kruskal's algorithms can be improved: when we have to always select a minimum cost edge then there is one data structure which will always give you a minimum value that is Min Heap . If min heap is used, then min heap will always give you a minimum value. Whenever you delete a value, you get a minimum value. If all these edges are kept in a min heap whenever you delete. You will getting a next minimum edge. so, you don't have a to search. So, From min heap whenever you delete the time taken is log(n). so much time it will take for finding a minimum cost edges is log(n). so how many time you have to do that. That is n times number.

Using Min Heap time Complexity is O(log(n)).

It may be Finding spanning tree for all the components but not the entire graph. So Kruskal's algorithms may working tree. For those components .

Strategies to delete branches:

1. Delete the branch with max weight.
2. Delete Inconsistent branches .
3. Delete by Analysis of Weights .

K-Means Clustering Algorithm: K-Means Clustering is an unsupervised learning algorithm that is used to solve the clustering problems in machine learning or data science. In this topic, we will learn what is K-means clustering algorithm, how the algorithm works, along with the Python implementation of k-means clustering .

What is K-Means Algorithm?

K-Means Clustering is an Unsupervised Learning algorithm, which groups the unlabeled dataset into different clusters. Here K defines the number of pre-defined clusters that need to be created in the process, as if K=2, there will be two clusters, and for K=3, there will be three clusters, and so on.It is an iterative algorithm that divides the unlabeled dataset into k different clusters in such a way that each dataset belongs only one group that has similar properties . It allows us to cluster the data into different groups and a convenient way to discover the categories of groups in the unlabeled dataset on its own without the need for any training. It is a centroid-based algorithm, where each cluster is associated with a centroid. The main aim of this algorithm is to minimize the sum of distances between the data point and their corresponding clusters. The algorithm takes the unlabeled dataset as input, divides the dataset into k-number of clusters, and repeats the process until it does not find the best clusters. The value of k should be predetermined in this algorithm.The k-means clustering algorithm mainly performs two tasks: Determines the best value for K-center points or centroids by an iterative process. Assigns each data point to its closest k-center. Those data points which are near to the particular k-center, create a cluster.

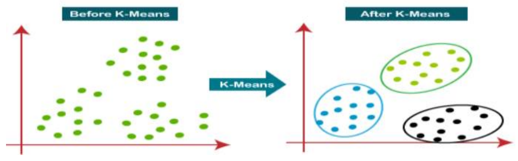


Fig 09 : K-mean clustering .

How does the K-Means Algorithm Work?The working of the K-Means algorithm is explained in the below steps:

1. Select the number K to decide the number of clusters.
2. Select random K points or centroids. (It can be other from the input dataset).
3. Assign each data point to their closest centroid, which will form the predefined K clusters.
4. Calculate the variance and place a new centroid of each cluster.
5. Repeat the third steps, which means reassign each datapoint to the new closest centroid of each cluster.
6. If any reassignment occurs, then go to step-4 else go to FINISH.
7. The model is ready.

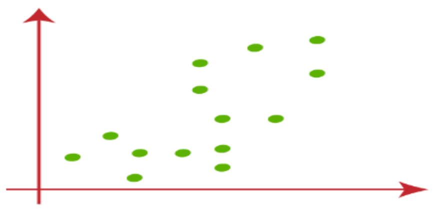


Fig 10: x-y axis scatter plot of these two variables .Let's take number k of clusters, i.e., K=2, to identify the dataset and to put them into different clusters. It means here we will try to group these datasets into two different clusters. We need to choose some random k points or centroid to form the cluster. These points can be either the points from the dataset or any other point. So, here we are selecting the below two points as k points, which are not the part of our dataset.

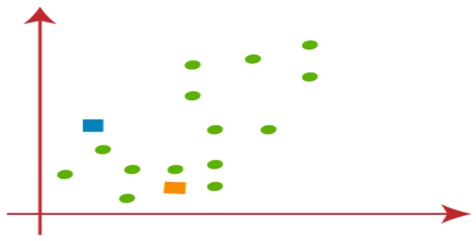
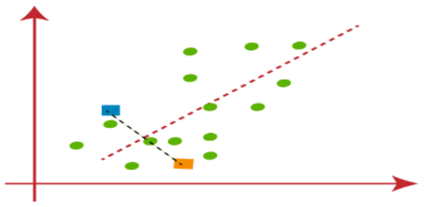
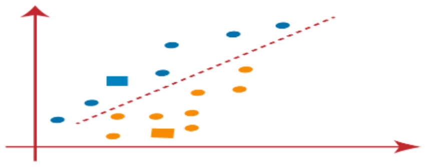


Fig 11 : Finding K point

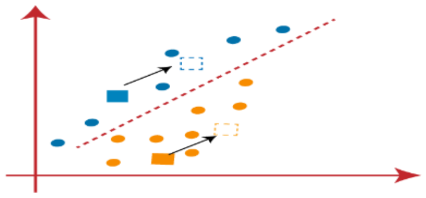
Now we will assign each data point of the scatter plot to its closest K-point or centroid. We will compute it by applying some mathematics that we have studied to calculate the distance between two points. So, we will draw a median between both the centroids.



From the above image, it is clear that points left side of the line is near to the K1 or blue centroid, and points to the right of the line are close to the yellow centroid. Let's color them as blue and yellow for clear visualization.



As we need to find the closest cluster, so we will repeat the process by choosing **a new centroid**. To choose the new centroids, we will compute the center of gravity of these centroids, and will find new centroids .



As our model is ready, so we can now remove the assumed centroids, and the two final cluster .

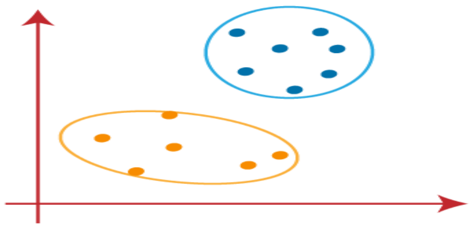


Fig : Final Cluster .

Compare graph-based clustering and K-means clustering: To evaluate the performance of both clustering models, typically without ground truth labels, internal evaluation metrics are used. One common metric is the silhouette score, which measures how similar an object is to its own cluster compared to other clusters.

The silhouette scores for graph-based clustering and K-means clustering are -0.4645 and -0.4945, respectively. While negative values indicate that objects may be assigned to the wrong clusters, the absolute values help understand the degree of separation.

For a deeper understanding of clustering, a confusion matrix can be analyzed. It's important to note that the confusion matrix is not a common metric for clustering, given that clustering is unsupervised. However, a confusion matrix can be created based on ground truth labels if available.

```
[[264   2]
 [152   0]]
```

Graph Based Clustering Graph :

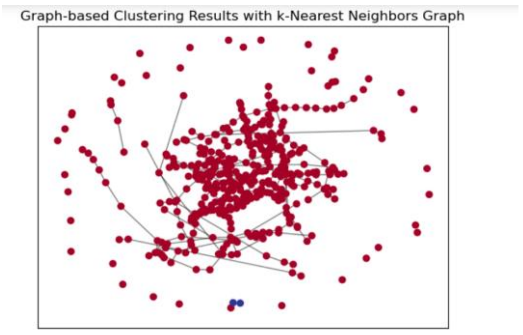


Fig 15: Graph-based Clustering (sex and Embarked)

K-mean Clustering :

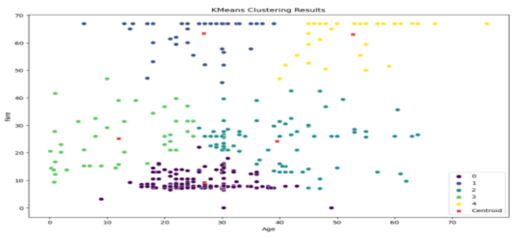


Fig16 : K-mean clustering of (sex and Embarked)

How to choose the value of "K number of clusters" in K-means Clustering? The performance of the K-means clustering algorithm depends upon highly efficient clusters that it forms. But choosing the optimal number of clusters is a big task. There are some different ways to find the optimal number of clusters, but here we are discussing the most appropriate method to find the number of clusters or value of K. The method is given below:

Elbow Method: The Elbow method is one of the most popular ways to find the optimal number of clusters. This method uses the concept of WCSS value. **WCSS** stands for **Within Cluster Sum of Squares**, which defines the total variations within a cluster. The formula to calculate the value of WCSS (for 3 clusters) is given below:

$$WCSS= \sum_{P_i \text{ in } Cluster1} distance(P_i\ C_1)^2 + \sum_{P_i \text{ in } Cluster2} distance(P_i\ C_2)^2 + \sum_{P_i \text{ in } CLuster3} distance(P_i\ C_3)^2$$

Conclusion: The project aimed to explore and implement graph-based clustering techniques, specifically focusing on K-nearest neighbours (KNN) and Epsilon neighbourhood graph, along with traditional K-means clustering. The key steps involved data loading, analysis, visualization, and preprocessing, followed by the application of graph-based clustering using KNN and Epsilon neighbourhood graph. Additionally, the project incorporated the use of minimum spanning tree in the graph-based clustering approach. Furthermore, K-means clustering was applied for comparison, and the performance of both clustering models was evaluated using internal metrics such as silhouette score and confusion matrix.

The silhouette scores indicated the degree of similarity within clusters, and the confusion matrices, although unconventional for clustering, provided insights based on ground truth labels. Notably, the project revealed that both clustering methods resulted in similar confusion matrices, implying comparable clustering outcomes. The silhouette scores, while negative, helped quantify the separation of objects within clusters. In summary, the project provided a comprehensive exploration of graph-based clustering and K-means clustering, offering insights into their applications, strengths, and limitations. The analysis of clustering performance contributes to a better understanding of how these techniques can be applied to real-world datasets for effective pattern recognition and grouping of data points.

REFERENCES :

[1]https://www.javatpoint.com/k-means-clustering-algorithm-in-machine-learning(JavaTpoint),“K-mean clustering”

[2] Abdul barik . (Minimum spanning tree, prim’s, Kruskal’s algorithms)

[3] Geeks for Geeks for data preprocessing

[4] Kaggle for “titanic_Dataset”

[5] Sklearn website for “model implementation” .

[6] of Introduction to Data Mining by Tan et al (Second edition) is Chapter 8 Section 4 (pp. 676-701).