**Categorize all the community detection techniques and then Compare their functionalities.**

**What is Community Detection?**

* **Community detection** is the process of finding **groups of nodes** in a network where **nodes within a group are more connected** to each other than to the rest of the network.
* These groups, called communities or clusters, often represent meaningful structures like friend circles, customer segments, or protein groups.
* By detecting communities, we can simplify analysis by focusing on **groups of nodes rather than individual nodes**, making it easier to understand the overall structure and behavior of the network.
* It helps in identifying **hidden structures** or **functional modules** in social networks, biological networks, recommendation systems, etc.
* It plays a key role in **network analysis**, allowing us to simplify complex graphs and **understand patterns** in real-world data.

**Community Detection Techniques**

**1. Graph Partitioning Methods:** These methods aim to divide the graph into a fixed number of parts, minimizing the connections between groups.

**Min-Cut (Minimum Cut)**

* Min-Cut is a graph partitioning method that divides a graph into two or more parts by **cutting the least number of edges** (i.e., minimizing the total weight of edges that are removed).
* It Find a small set of edges whose removal splits the graph into groups.
* The total weight (or number) of edges that connect different groups is called the “cut”. Min-Cut tries to make this as **small as possible**.
* It is efficient and works on both weighted and unweighted graphs.
* It may result in **unbalanced partitions**, like one node in one group and all others in the second.

**Normalized Cut (N-Cut)**

* Normalized Cut is an improvement over Min-Cut. It not only minimizes the cut between groups but also considers the **size and volume** of each group to avoid unbalanced splits.
* **Balances Partitions**: It prevents the problem where one group is much smaller by dividing the cut by the total connections (volume) in each part.
* **Better for Clustering**: Especially in image segmentation and social network analysis where balanced clusters are desired.
* **Used in Spectral Clustering**: The Normalized Cut value is often minimized using eigenvalue methods (spectral techniques).

**2. Modularity-Based Methods:** These methods try to find the division of the network that results in the highest modularity score, which measures the strength of communities.

**Louvain Algorithm:**

* Louvain begins by assigning each node to its own community and then repeatedly moves nodes to neighboring communities to increase the modularity score, which indicates stronger communities.
* Once modularity can no longer be improved, it compresses communities into single nodes and repeats the process, allowing the detection of hierarchical community structures.
* This method is very fast and can handle very large graphs, such as social networks with millions of users.
* It does not require the user to predefine the number of communities — the algorithm determines it automatically.

**FastGreedy Algorithm**

* This method also maximizes modularity but uses a greedy approach, merging communities that produce the largest modularity increase at each step.
* It starts with each node as an individual community and merges them until no modularity gain is possible.
* It works well for networks where clear modular divisions exist, such as citation networks or collaboration graphs.
* The algorithm is slower than Louvain, so it's better for moderate-sized networks.
* It gives accurate community structures but can struggle with very large graphs.

**3. Hierarchical Clustering Methods:** These methods create a nested set of communities, represented as a tree-like structure called a dendrogram.

**Girvan–Newman Algorithm**

* It works by calculating edge betweenness centrality, which measures the number of shortest paths passing through each edge. Edges with high betweenness likely connect different communities.
* The algorithm removes the edge with the highest betweenness, splitting the graph into smaller parts gradually, creating a hierarchy of communities.
* This method helps identify clear community boundaries and produces a tree like structure, so you can decide the level of granularity for communities.
* It is particularly useful in small networks like academic collaboration graphs or organizational charts.
* However, it is computationally expensive and slow for large networks, as betweenness needs to be recalculated after each edge removal.

**Edge-Betweenness Concept**

* Edge betweenness is the key metric used in Girvan–Newman; it indicates how important an edge is as a bridge connecting different parts of the network.
* Removing edges with high betweenness gradually disconnects communities.
* Helps in detecting bottlenecks or weak links in networks.
* Computing edge betweenness is costly in large graphs, which limits its use.
* Often combined with hierarchical methods for community detection.

**4. Label Propagation Methods:** These algorithms use a fast iterative process where nodes update their community label based on their neighbors.

**Label Propagation Algorithm (LPA)**

* Each node starts with a unique label (often its own ID). During each iteration, nodes update their label to the one that appears most frequently among their neighbors.
* This process repeats until no labels change, and nodes sharing the same label form a community.
* It is very fast and scalable, suitable for massive social networks like Twitter or Facebook.
* Does not require predefining the number of communities or other parameters, making it easy to use.
* However, LPA results can be unstable and different runs can produce slightly different community assignments.