In this code, we are trying to find the smallest distance r (radius) such that if we connect all points that are within this distance of each other, we create one big, connected structure (also known as a connected graph). Additionally, we visualize this connected structure using a 2D plot.

We used numpy for data manipulation, scipy.spatial.distance\_matrix for distance calculation matrix, network to work with complex graphs, matplotlib.pyplot to plot the graphs in 2D.

After loading the data :

* We calculate the "distance matrix" (stored in dist\_matrix), which contains the Euclidean distance between every pair of points. This matrix will help us understand which points are close enough to connect based on a given radius.
* We call the function get\_min\_radius\_for\_connectivity, which will find the smallest radius where all points in the graph are connected. We also plot the graph to visualize how the points connect at that radius.
* We create a sorted list of all unique distances between points. By checking each of these distances as a potential radius, we can gradually increase the radius until we find the smallest one that creates a connected graph.
* For each radius, we create a new empty graph. Here, each point is a node, and an edge (connection) will be added between nodes if they are within the radius distance of each other.
* For each pair of points, if the distance between them is less than or equal to the current radius, we add an edge between them in the graph. This step creates connections between points that are close enough.
* We check if the graph is fully connected, meaning there’s a path between any two nodes. If it is connected, we know that this radius is large enough.
* Once we find a radius that makes the graph connected, we plot it using matplotlib. Since our data points are in 3D, we project them onto a 2D plane by only using the X and Y coordinates.