**Dash Report:**

Dash gives data scientists the ability to showcase their results in interactive web applications.

Dash is an open-source framework for building data visualization interfaces.

Plotly also offers a commercial companion to Dash called [Dash Enterprise](https://plotly.com/dash/). This paid service provides companies with support services such as hosting, deploying, and handling authentication on Dash applications. But these features live outside of Dash’s open-source ecosystem

**Spectral cluster Analysis:**

Spectral Clustering is a variant of the clustering algorithm that uses the connectivity between the data points to form the clustering.

It uses eigenvalues and eigenvectors of the data matrix to forecast the data into lower dimensions space to cluster the data points.

It is based on the idea of a graph representation of data where the data point are represented as nodes and the similarity between the data points are represented by an edge.

Algorithm

* Project data into Rn𝑅𝑛 matrix
* Define an Affinity matrix A , using a Gaussian Kernel K or an Adjacency matrix
* Construct the Graph Laplacian from A (i.e. decide on a normalization)
* Solve the Eigenvalue problem
* Select k eigenvectors corresponding to the k lowest (or highest) eigenvalues to define a k-dimensional subspace
* Form clusters in this subspace using k-means

Affinity Matrix:

We first create an undirected graph G = (V, E) with vertex set V = {v1, v2, …, vn} = 1, 2, …, n observations in the data.

* **Epsilon-neighbourhood Graph:**

A parameter epsilon is fixed beforehand. Then each point is connected to all the points which lie in it’s epsilon-radius. If all the distances between any two points are similar in scale then typically the weights of the edges ie the distance between the two points are not stored since they do not provide any additional information. Thus, in this case, the graph built is an undirected and unweighted graph.

* **K-Nearest Neighbours:**

A parameter k is fixed beforehand. Then, for two vertices u and v, an edge is directed from u to v only if v is among the k-nearest neighbours of u. Note that this leads to the formation of a weighted and directed graph because it is not always the case that for each u having v as one of the k-nearest neighbours, it will be the same case for v having u among its k-nearest neighbours. To make this graph undirected, one of the following approaches are followed:-

* + Direct an edge from u to v and from v to u if either v is among the k-nearest neighbours of u **OR**u is among the k-nearest neighbours of v.
  + Direct an edge from u to v and from v to u if v is among the k-nearest neighbours of u **AND**u is among the k-nearest neighbours of v.
* **Fully-Connected Graph:**

To build this graph, each point is connected with an undirected edge-weighted by the distance between the two points to every other point. Since this approach is used to model the local neighbourhood relationships thus typically the Gaussian similarity metric is used to calculate the distance.

S(xi,xj)=exp(−||xi−xj||22σ2)𝑆(𝑥𝑖,𝑥𝑗)=𝑒𝑥𝑝(−||𝑥𝑖−𝑥𝑗||22𝜎2)

Thus, when we create an adjacency matrix for any of these graphs, Aij ~ 1 when the points are close and Aij → 0 if the points are far apart.

Consider the following graph with nodes 1 to 4, weights (or similarity) wij and its adjacency matrix: