**Sparse Matrices : scipy.sparse.csgraph**

A graph is just a collection of nodes connected by connections. Graphs can represent almost anything: social network connections, in which each node is a person connected to acquaintances; images, in which each node is a pixel connected to neighbouring pixels; points in a high-dimensional distribution, in which each node is connected to its nearest neighbours; and practically anything else you can think of.

A sparse matrix, let's call it G, is a particularly efficient way to describe graph data. G[i, j] is the value of the link between nodes I and 'j' in the matrix G of size N x N. A sparse graph is one in which the majority of nodes have only a few connections.

Several techniques employed in scikit-learn inspired the construction of the sparse graph submodule, including the following:

* Isomap is a manifold learning technique that seeks for the shortest pathways through a network.
* A clustering approach based on a minimum spanning tree is known as hierarchical clustering.
* A projection approach based on sparse graph laplacians is Spectral Decomposition.

**Example 1 : Constructing a Dense, Masked and Sparse Representation**

*import numpy as np*

*G\_dense = np.array([ [0, 2, 1],*

*[2, 0, 0],*

*[1, 0, 0] ])*

*G\_masked = np.ma.masked\_values(G\_dense, 0)*

*from scipy.sparse import csr\_matrix*

*G\_sparse = csr\_matrix(G\_dense)*

*print(G\_sparse.data)*

*Output: [2 1 2 1]*