Data Visualization

Heidi Implementation

Problem statement:

Follow up with Heidi Matrix : nearest neighbour Driven High Dimensional Data visualization

As discussed following tasks were expected to be done

1. Code check
2. Epsilon problem - Given two points p and q for what change in value of p and q bit vector remains intact
3. Modification to KNN Algorithm to find actual overlap.
4. New ordering mechanism : Instead of ordering based on algorithm like distance from centroid or connectivity come up with different solution
5. Clustering on heidi matrix (Image Segmentation)
6. In ordered dataset actual row and actual column number to be mentioned

Solution:

1. Code was corrected, the error found in code was with k-means initialization.

Results:

IRIS Dataset:

Number of Instances: 150 (50 in each of three classes)

Number of Attributes: 4 numeric, predictive attributes and the class

Attribute Information:

1. sepal length in cm

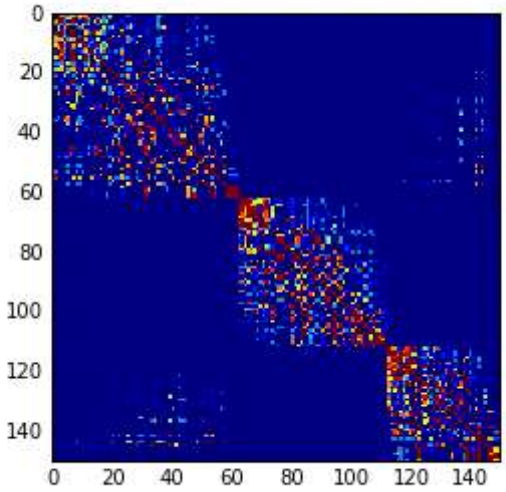
2. sepal width in cm

3. petal length in cm

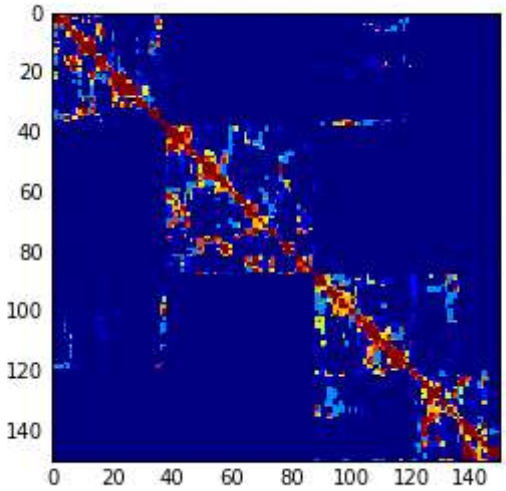
4. petal width in cm

5. class: -- Iris Setosa, Iris Versicolour, Iris Virginica

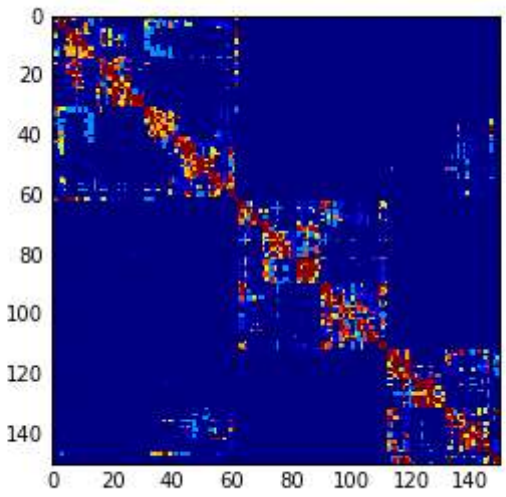
A) Distance from centroid ordering



B) Connectivity ordering



C) Minimum spanning ordering



Haberman Dataset :

Number of Instances: 306

Number of Attributes: 4 (including the class attribute)

Attribute Information:

1. Age of patient at time of operation (numerical)

2. Patient's year of operation (year - 1900, numerical)

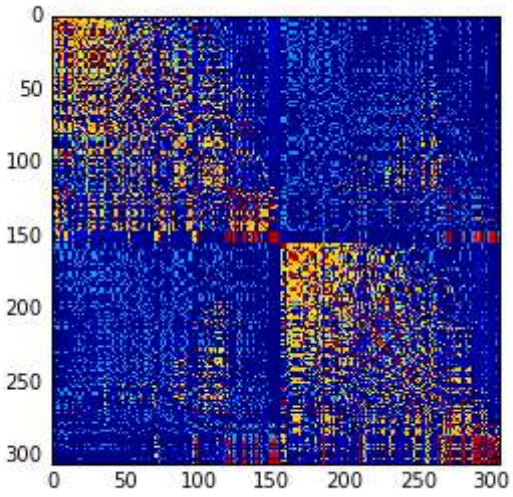
3. Number of positive axillary nodes detected (numerical)

4. Survival status (class attribute)

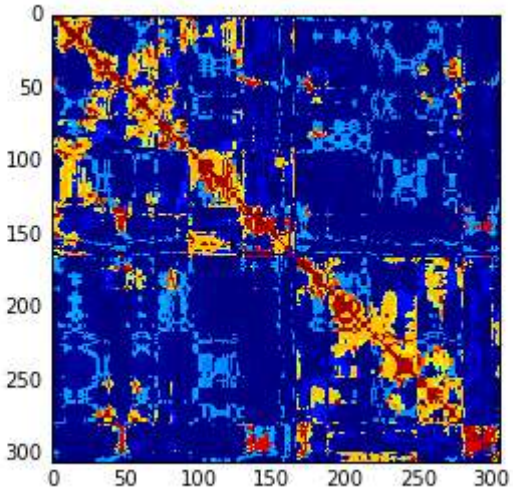
1 = the patient survived 5 years or longer

2 = the patient died within 5 year

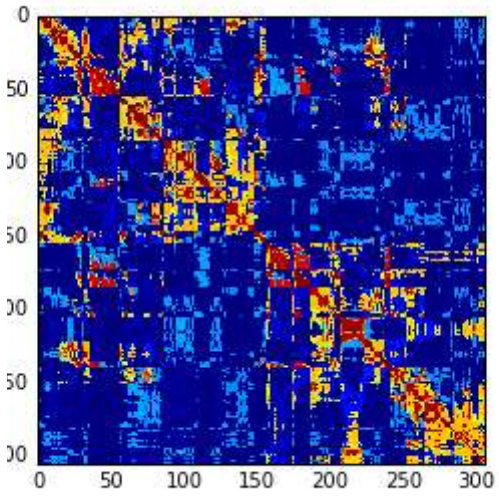
A) Distance from centroid ordering



B) Connectivity ordering



C) Minimum spanning tree



2) Modification to KNN to find actual overlap

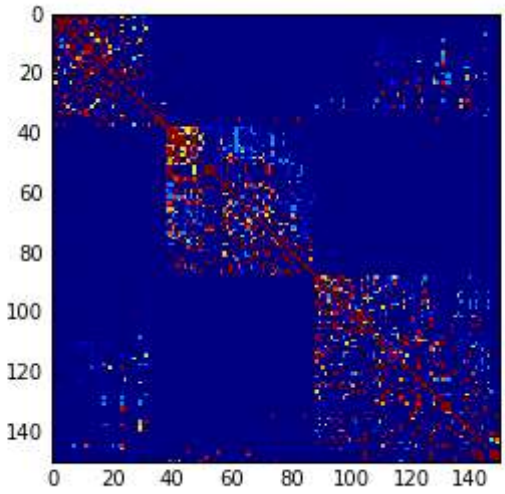
Approach is to find make a matrix similar to heidi matrix in dimension and create an overlap-matrix which represents in what all subspaces all datapoints overlap.

Then apply Knn and consider only those points which overlap in either of the subspace.

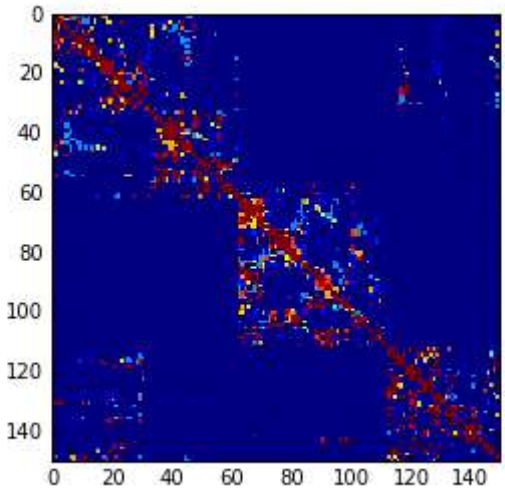
Results obtained are:

IRIS Dataset : (k=10)

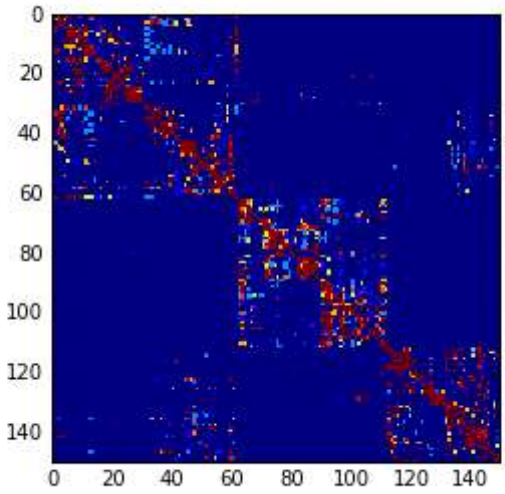
A) Distance from centroid ordering



B) Connectivity ordering

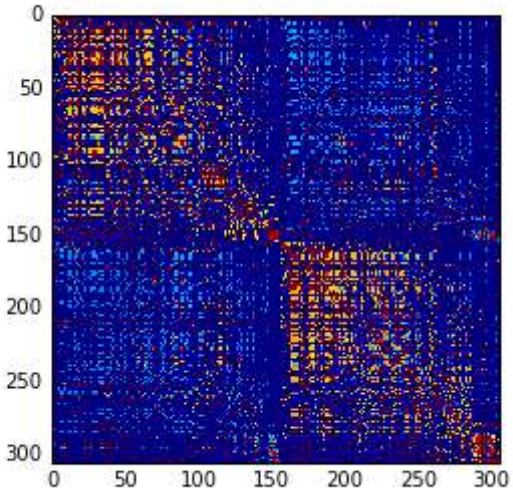


C) Minimum spanning Tree ordering

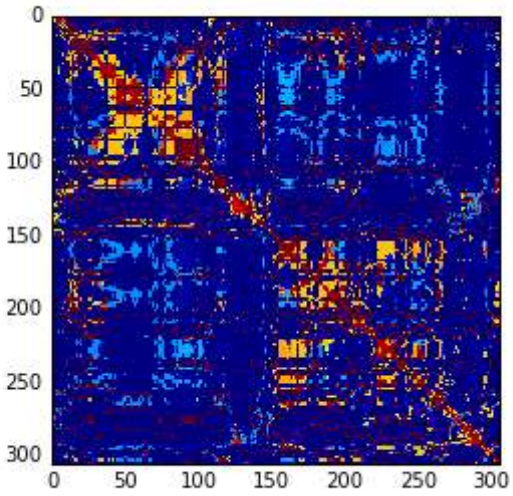


Haberman Dataset: (k=50)

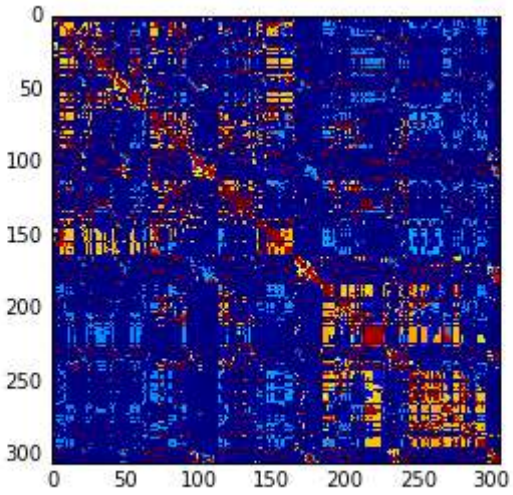
A) Distance from centroid



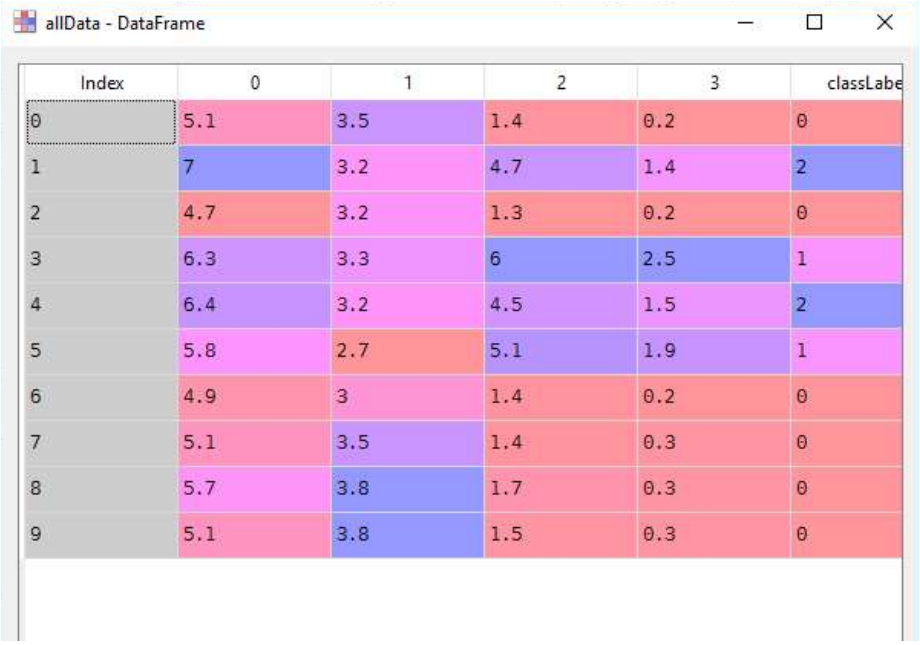
B) Connectivity ordering



C) Minimum spanning tree ordering



3) Ordering of datapoints:

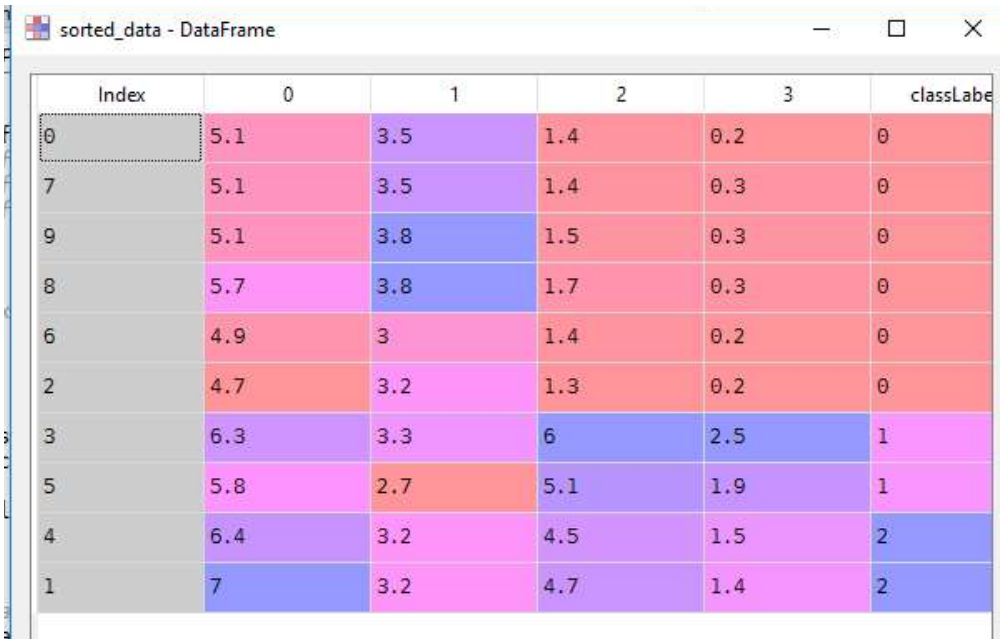


Index	0	1	2	3	classLabe
0	5.1	3.5	1.4	0.2	0
1	7	3.2	4.7	1.4	2
2	4.7	3.2	1.3	0.2	0
3	6.3	3.3	6	2.5	1
4	6.4	3.2	4.5	1.5	2
5	5.8	2.7	5.1	1.9	1
6	4.9	3	1.4	0.2	0
7	5.1	3.5	1.4	0.3	0
8	5.7	3.8	1.7	0.3	0
9	5.1	3.8	1.5	0.3	0

A) Distance from centroid :

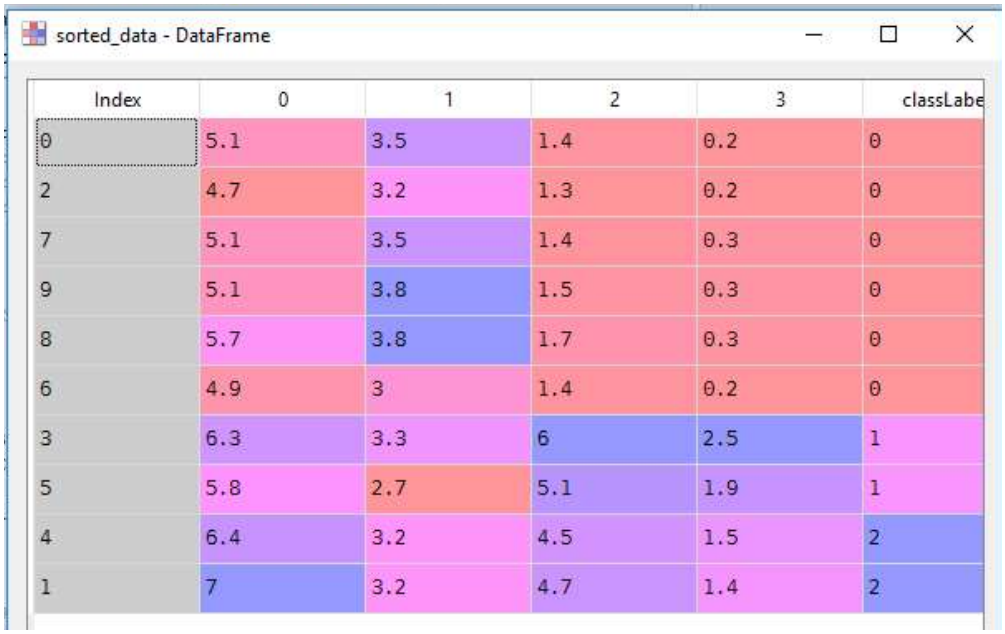
Index	0	1	2	3	classLabe
0	5.1	3.5	1.4	0.2	0
7	5.1	3.5	1.4	0.3	0
9	5.1	3.8	1.5	0.3	0
2	4.7	3.2	1.3	0.2	0
6	4.9	3	1.4	0.2	0
8	5.7	3.8	1.7	0.3	0
3	6.3	3.3	6	2.5	1
5	5.8	2.7	5.1	1.9	1
4	6.4	3.2	4.5	1.5	2
1	7	3.2	4.7	1.4	2

B) Connectivity Distance:



Index	0	1	2	3	classLabe
0	5.1	3.5	1.4	0.2	0
7	5.1	3.5	1.4	0.3	0
9	5.1	3.8	1.5	0.3	0
8	5.7	3.8	1.7	0.3	0
6	4.9	3	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
3	6.3	3.3	6	2.5	1
5	5.8	2.7	5.1	1.9	1
4	6.4	3.2	4.5	1.5	2
1	7	3.2	4.7	1.4	2

C) Minimum Spanning Tree Distance:



Index	0	1	2	3	classLabe
0	5.1	3.5	1.4	0.2	0
2	4.7	3.2	1.3	0.2	0
7	5.1	3.5	1.4	0.3	0
9	5.1	3.8	1.5	0.3	0
8	5.7	3.8	1.7	0.3	0
6	4.9	3	1.4	0.2	0
3	6.3	3.3	6	2.5	1
5	5.8	2.7	5.1	1.9	1
4	6.4	3.2	4.5	1.5	2
1	7	3.2	4.7	1.4	2