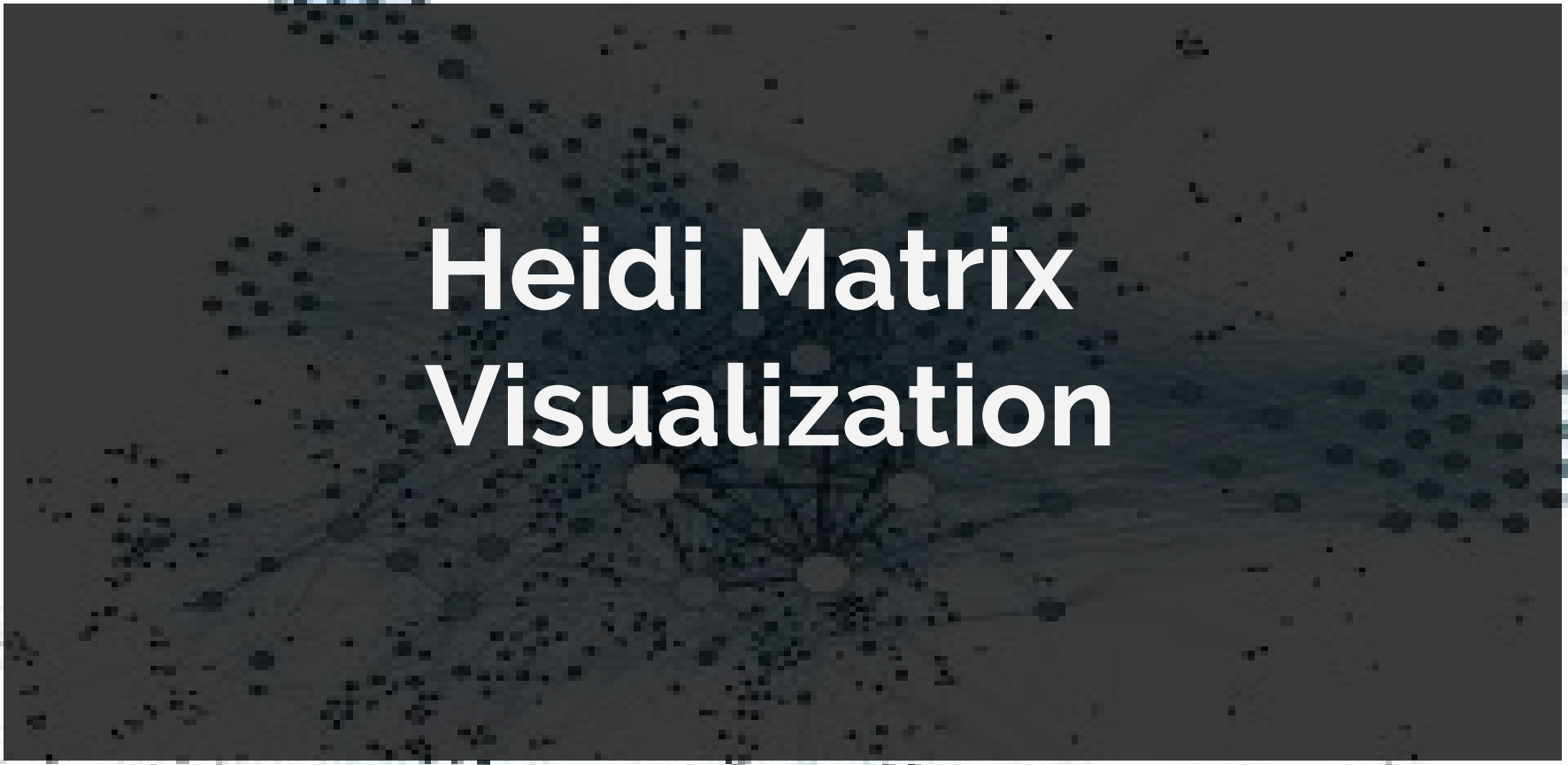
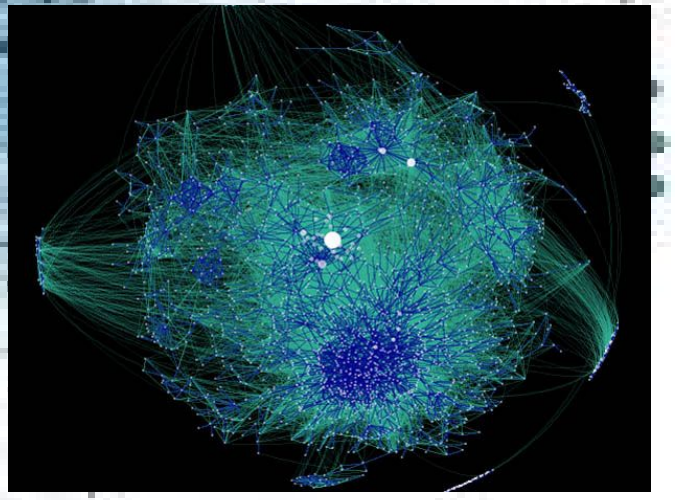


Heidi Matrix Visualization

Data Visualization. Why?

If data is high dimensional, then identifying patterns is difficult task owing to complexity and high dimensionality issues.

As the dimension of dataset increased, data in subspaces are more prominent than original dimensional space.



Heidi Matrix

Heidi matrix gives insight of

1. How the clusters are placed with respect to each other
2. Characteristics of placement of points within a cluster in all the subspaces.
3. Characteristics of overlapping clusters in various subspaces

Heidi matrix Input/output

INPUT:

- d dimensional dataset

OUTPUT:

- 2-D colored matrix called Heidi matrix

Heidi Algorithm:

Given n X d dataset . (n : no. of rows ; d : no. of columns)

1. compute set of all possible subspaces. (2^n -1)
2. For each subspace
   compute knn for every pair of points
3. Merge these (2^n -1) matrices into one (bitwise)
4. Grouping and ordering of points
5. Draw image

Implementation

Heidi approach was implemented in python and the pseudo code is as follows.

1. compute set of all possible subspaces. (2^n -1)

This step was done via binary couting.

2. pick one subspace

compute knn for every pair of points

For computing knn sklearn inbuilt method NearestNeighbors was used,

3. merge these (2^n -1) matrices into one

This step was done using basic binary maniputation.

4. Represent above computed 2-D matrix as image

Heidi approach was implemented on

Haberman's Survival Dataset.

Number of Instances: 306

Number of Attributes: 4

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)

heidi_matrix - NumPy array

	5	6	7	8	9	10	11	12	13	14	15
0	16576	16639	179	140	192	115	76	12	51	3084	51
1	16580	16639	187	128	192	127	76	48	63	0	3
2	16576	28927	8370	140	192	115	76	12	12339	12	12351
3	16588	16639	2239	128	204	127	76	0	63	0	51
4	16576	16639	176	143	192	115	76	63	48	12	15
5	32767	16588	204	12531	3276	207	13311	240	204	51	0
6	16588	29695	8447	192	204	3327	204	192	16383	0	12339
7	204	8447	16383	192	204	255	204	192	8447	192	8243
8	243	192	192	16383	192	192	243	1023	192	255	12
9	19660	16588	204	192	16383	204	204	192	204	243	0
10	16588	16639	255	192	204	4095	204	192	3327	192	51
11	29695	16588	204	12531	204	204	13311	243	204	255	12

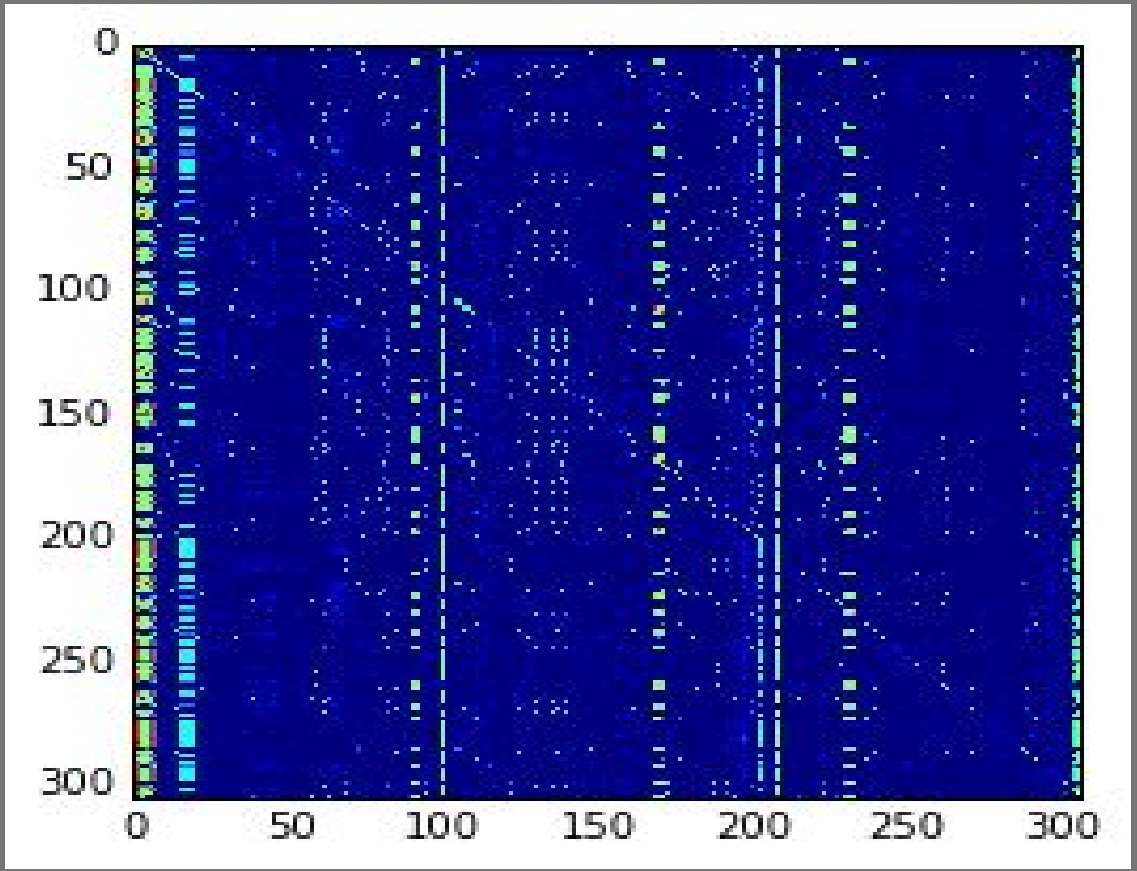
Format | Resize | Background color

OK | Cancel

Heidi Matrix

305 X 305 Heidi Matrix

(305 : no of rows/ instances in test dataset,.

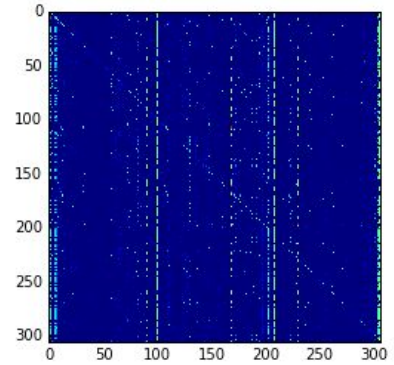


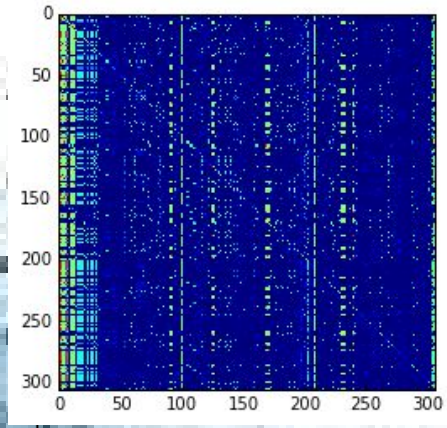
Visual representation of data via Heidi Matrix

This image is generated output when heidi matrix algorithm was implemented in Python

Heidi Matrix with varying value of k

k=[5 10 15 20]



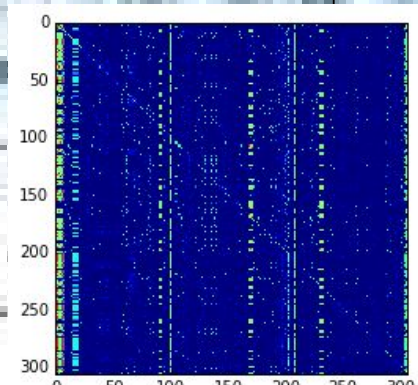


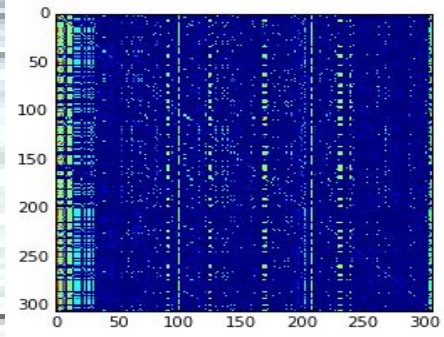
k=5

k=10

k=15

k=20





Summary

Heidi generated 2-d representation of higher dimensional data

Presents spatial overlap among clusters in various subspaces.

Presents nearest neighbour proximity information among datapoints.