PROJECT REPORT
CLUSTERING ALGORITHMS

Arun Sharma(50206920)
Prashanth Chandrashekar(50207911)
Vipin Kumar(50208397)

Algorithm Description:

We compared 4 clustering techniques K means, Hierarchical Agglomerative Clustering (HAC) using Single Link node, DBSCAN (Density Based Spatial Clustering of application with noise) and MapReduce K-means. Each one gave interesting results of Jaccard values and plots which gave some interesting findings.

K means:

K-means clustering is a type of unsupervised learning algorithm that is used to find groups in the data, with the number of groups represented by the variable *K*. The algorithm works iteratively assigning each point to one of the K groups and the data points are clustered on the basis of their similarity.

The steps are as follows:

1) Partition the dataset into K clusters
2) Initialize the centroid.
3) For each point x we calculate Euclidian distance between each K centroids.
4) Reassign x to the cluster which is closest to the centroid
5) Update the cluster centroids based on current value.

Hierarchical Agglomerative Clustering Algorithm:

It is a bottom up clustering method where the cluster consists of sub-clusters and sub-clusters. It starts with the single cluster with every single object and with successive iterations it merges the closest pair using some similarity criteria. Here we have used Euclidian distance.

The steps are as follows:

1) Assign each point as a separate cluster.
2) Evaluate Euclidian distance between clusters.
3) Construct distance matrix
4) Check the shortest value.
5) Remove the pair and merge them
6) Evaluate all the distance and check the minimum and update the merged cluster matrix.
7) Repeat until the cluster reduced to the single element.

Density Based Spatial Clustering of Application with Noise:

It is used to identify clusters of any shape in a data set containing noise and outliers. It is based on intuitive notion of cluster and noise and hence it is derived as human intuitive clustering. The key idea is of each point of a cluster, the neighborhood contains at least minimum number of specified points. For minimum distance, we calculate the Euclidian matrix and compare the value with eps.

The steps are as follows:

1) Load the dataset and set predicted label as false.
2) Initialize minpoints i.e. minimum number of points that qualifies the cluster and the epsilon value which is the distance limit.
3) For each and every point assign neighbors by checking the Euclidean distance with epsilon value.
4) For each neighbor check if the points lie in the cluster by checking the minpoints and hence expand the cluster until no new points discovered.
5) Other points can be considered as noise.

Map-Reduce K means:

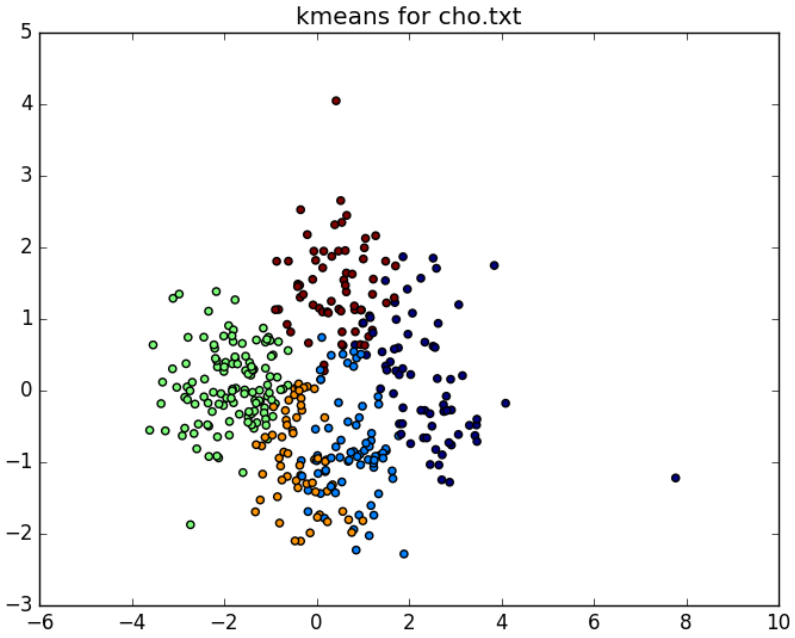
The steps are as follows:

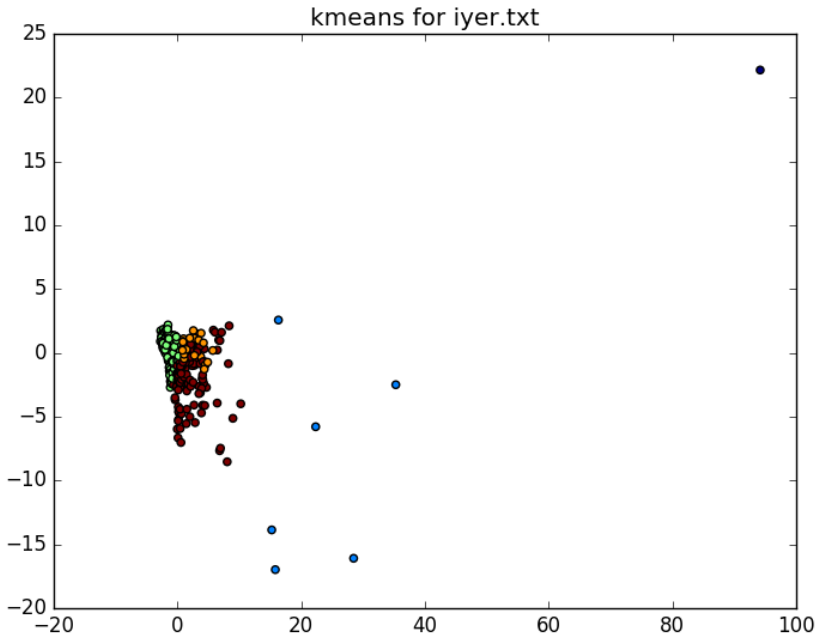
1) Partition the dataset into K clusters
2) Initialize the centroid.
3) For each point x we calculate Euclidian distance between each K centroids.
4) Reassign x to the cluster which is closest to the centroid
5) Update the cluster centroids based on current value.

Data Visualization:

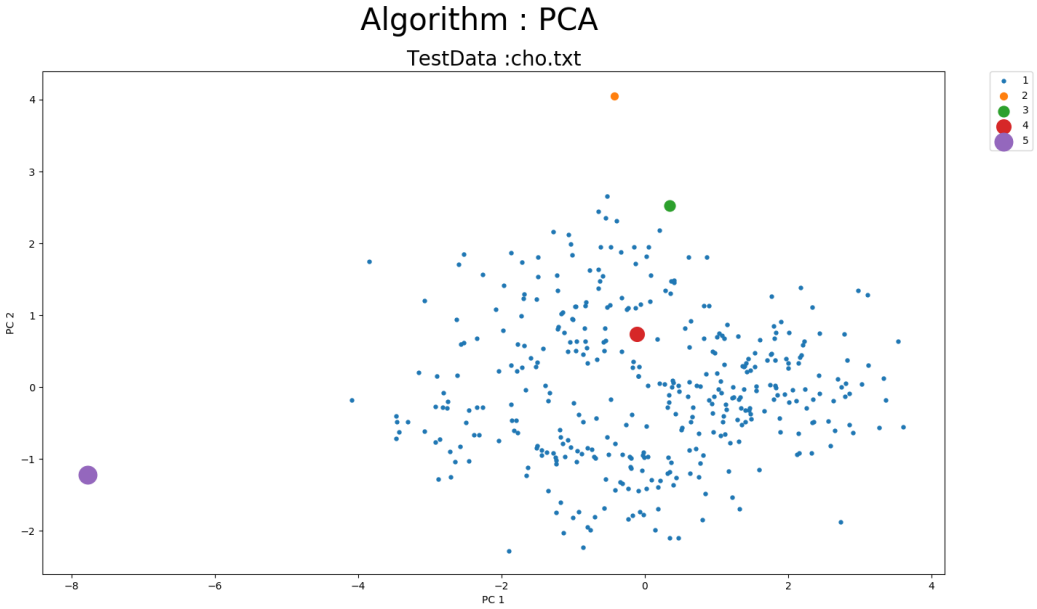
We used Principal Component Analysis for graph 2-D graph visualization of each Algorithm.

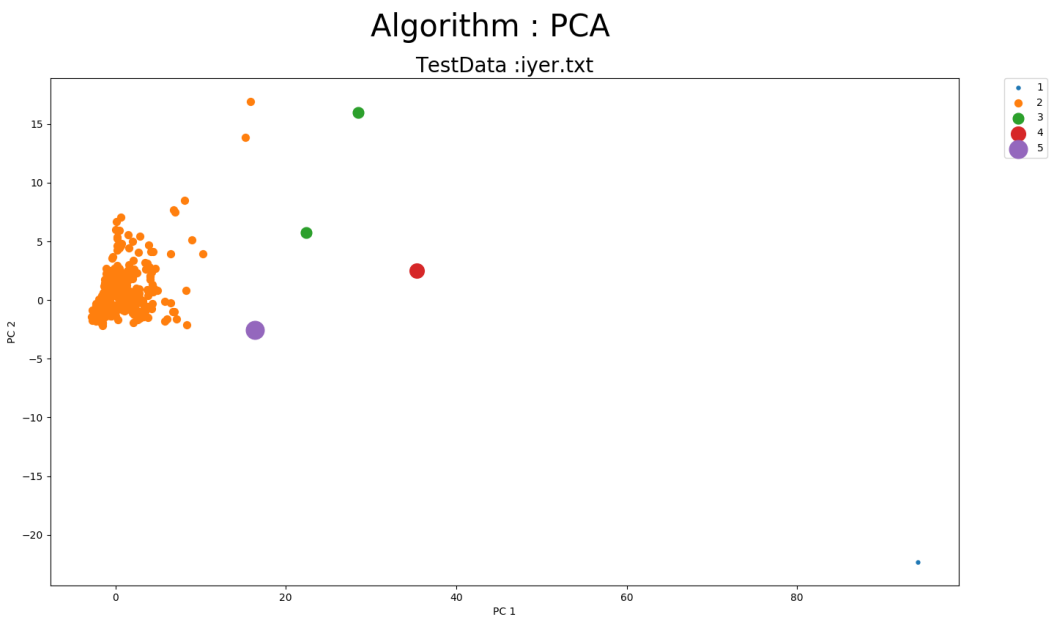
K Means:



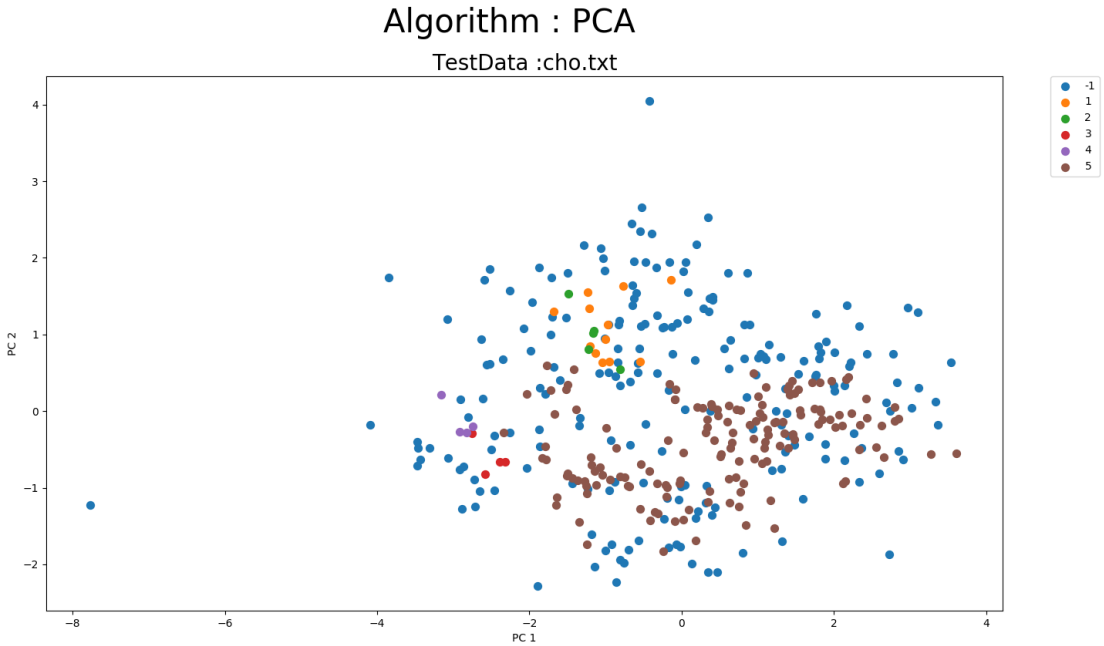

kmeans for iyer.txt
25
20
15
10
5
0
−5
−10
−15
−20
−20
0
20
40
60
80
100

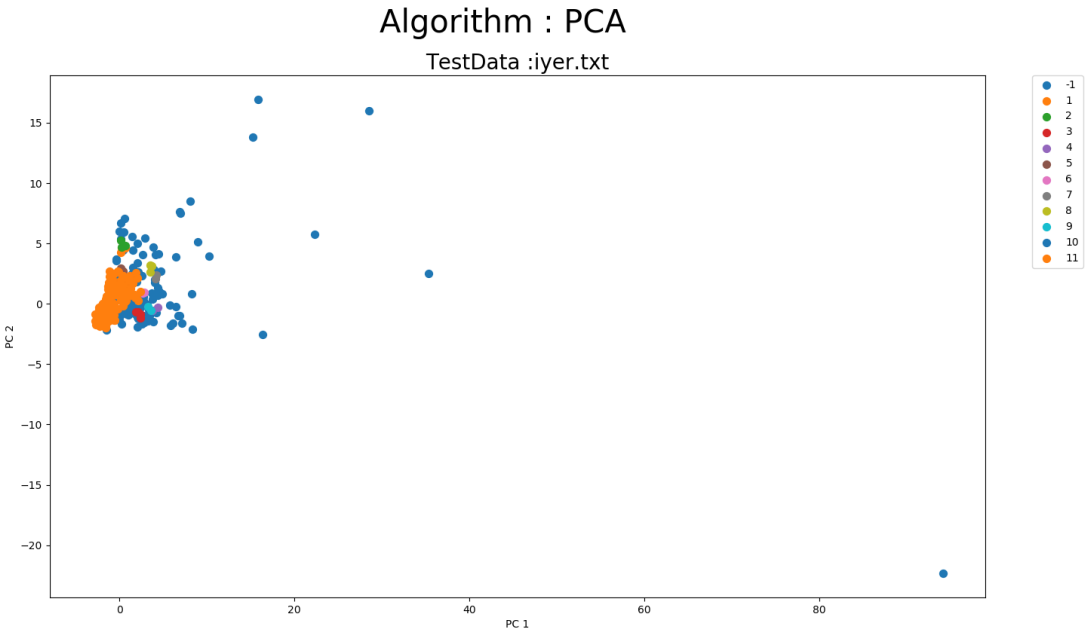
HAC:


Algorithm : PCA
TestData :cho.txt
PC 2
PC 1
1
2
3
4
5

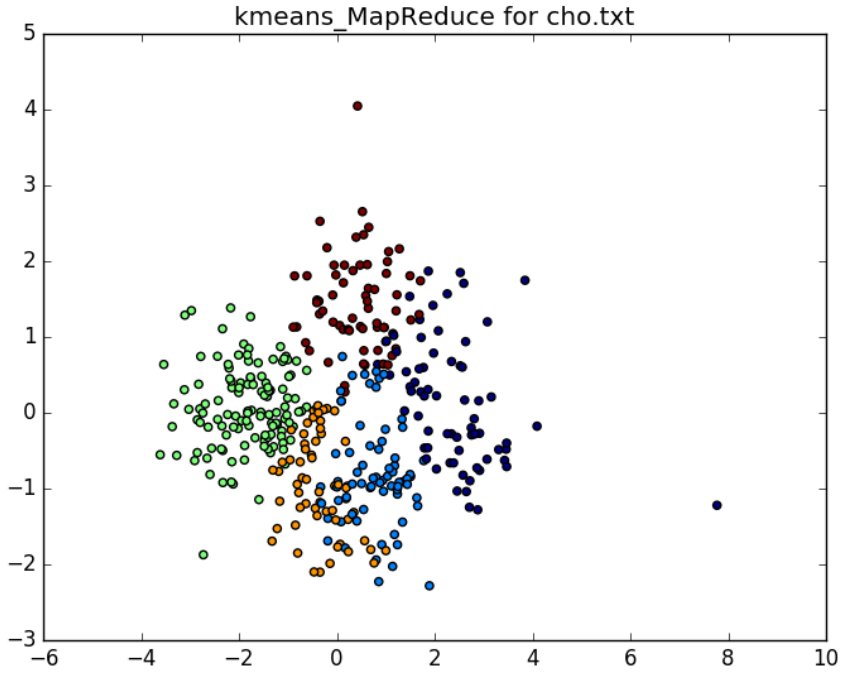

Algorithm : PCA
TestData :iyer.txt
PC 1
PC 2
1
2
3
4
5

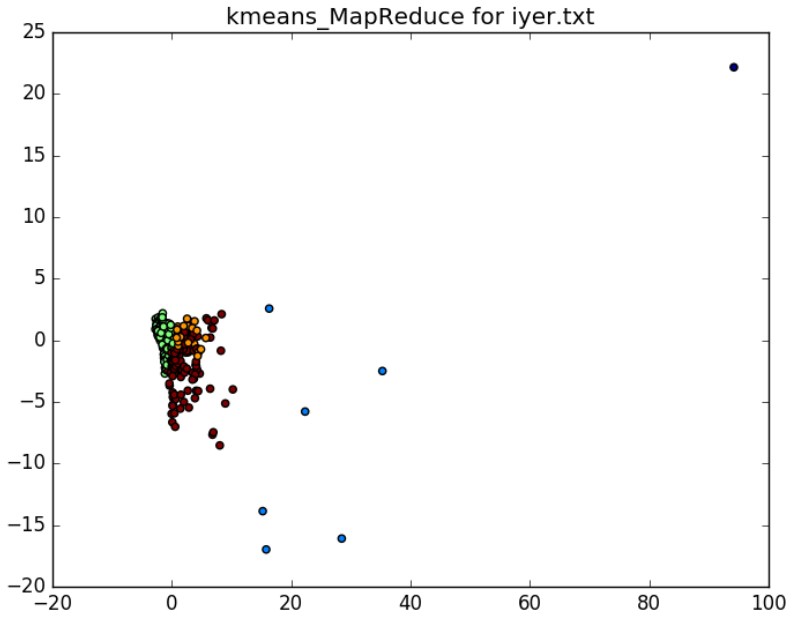
DBSCAN


Algorithm : PCA
TestData :cho.txt
PC 1
PC 2
-1
1
2
3
4
5



K Means Map-Reduce:





Data Evaluations:
For data evaluation we used Jaccard Coefficient which is

Algorithm	Iyer.txt	Cho.txt
K means	0.41077	0.406463
HAC	0.156	0.228
DBScan	0.229(2,1.149)	0.204(4,1.03)
K means MR	0.410931	0.406763

For K means and HAC we used 5 clusters. Parameters passed to DBScan with minpoints = 2 and epsilon = 1.149 for iyer.txt and minpoints = 4 and epsilon = 1.03 for cho.txt dataset.

Pros
Kmeans:

- Easier to implement

HAC:

- Can handle non-elliptical shape

Dbscan:

- Resistant to Noise
- Can handle different shapes

K means MR:

- Used for implementing large jobs

Cons:
K means:

- Cannot handle clusters deferring from size shapes and densities.

HAC:

- Sensitive to noise and outliers

Dbscan:

- Cannot handle varying densities
- Hard to set parameters

K means MR:

- Not useful for less data

Cluster Analysis
HAC:

For cho.txt
cluster 1 = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 127, 128, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 384, 385]
cluster 2 = [35]
cluster 3 = [126]
cluster 4 = [129]
cluster 5 = [383]

For iyer.txt
cluster 1 = [362]
cluster 2 = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77,

78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 262, 263, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 290, 291, 292, 293, 294, 295, 296, 297, 298, 299, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 310, 311, 312, 313, 314, 315, 316, 317, 318, 319, 320, 321, 322, 323, 324, 325, 326, 327, 328, 329, 330, 331, 332, 333, 334, 335, 336, 337, 338, 339, 340, 341, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 357, 358, 359, 360, 361, 363, 364, 365, 366, 367, 368, 369, 370, 371, 372, 373, 374, 375, 376, 377, 378, 379, 380, 381, 382, 383, 384, 385, 386, 387, 388, 389, 390, 391, 392, 393, 394, 395, 396, 397, 398, 399, 400, 401, 402, 403, 404, 405, 406, 407, 408, 409, 410, 411, 412, 413, 414, 415, 416, 417, 418, 419, 420, 421, 422, 423, 424, 425, 426, 427, 428, 429, 430, 431, 432, 433, 434, 435, 436, 437, 438, 439, 440, 442, 443, 444, 445, 446, 447, 448, 449, 450, 451, 452, 453, 454, 455, 456, 457, 458, 459, 460, 461, 462, 463, 464, 465, 466, 467, 468, 469, 470, 471, 473, 474, 475, 476, 477, 478, 479, 480, 481, 482, 483, 484, 485, 486, 487, 488, 489, 491, 492, 493, 494, 495, 496, 497, 498, 499, 500, 501, 502, 503, 504, 505, 506, 507, 509, 510, 511, 512, 513, 514, 515, 516]
cluster 3 = [441, 472]
cluster 4 = [490]
cluster 5 = [508]

DBScan:
For cho.txt
cluster 1 = [15, 16, 19, 21, 32, 33, 34, 51, 133, 145, 189, 237]
cluster 2 = [3, 6, 346, 353, 365]
cluster 3 = [343, 345, 348, 371]
cluster 4 = [282, 332, 337, 379]
cluster 5 = [24, 28, 31, 47, 69, 71, 73, 74, 75, 77, 79, 80, 81, 84, 85, 87, 88, 91, 92, 93, 94, 95, 96, 97, 99, 100, 101, 102, 103, 104, 105, 109, 110, 112, 113, 114, 116, 118, 127, 134, 135, 136, 137, 140, 142, 144, 146, 149, 152, 154, 155, 159, 161, 163, 165, 166, 167, 171, 173, 174, 181, 182, 184, 185, 192, 193, 195, 197, 199, 200, 201, 207, 208, 212, 221, 222, 223, 224, 225, 226, 229, 230, 231, 232, 233, 234, 236, 239, 240, 241, 243, 245, 251, 253, 254, 255, 256, 257, 258, 259, 263, 264, 265, 266, 267, 269, 270, 272, 273, 276, 277, 278, 279, 280, 281, 284, 285, 286, 287, 288, 290, 291, 297, 298, 299, 300, 308, 309, 313, 314, 316, 317, 320, 321, 322, 323, 324, 326, 327, 328, 329, 330, 336, 340, 341, 351, 352, 357, 359, 364, 368, 370, 381]
Outliers = [0, 1, 2, 4, 5, 7, 8, 9, 10, 11, 12, 13, 14, 17, 18, 20, 22, 23, 25, 26, 27, 29, 30, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 48, 49, 50, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 70, 72, 76, 78, 82, 83, 86, 89, 90, 98, 106, 107, 108, 111, 115, 117, 119, 120, 121, 122, 123, 124, 125, 126, 128, 129, 130, 131, 132, 138, 139, 141, 143, 147, 148, 150, 151, 153,

156, 157, 158, 160, 162, 164, 168, 169, 170, 172, 175, 176, 177, 178, 179, 180, 183, 186, 187, 188, 190, 191, 194, 196, 198, 202, 203, 204, 205, 206, 209, 210, 211, 213, 214, 215, 216, 217, 218, 219, 220, 227, 228, 235, 238, 242, 244, 246, 247, 248, 249, 250, 252, 260, 261, 262, 268, 271, 274, 275, 283, 289, 292, 293, 294, 295, 296, 301, 302, 303, 304, 305, 306, 307, 310, 311, 312, 315, 318, 319, 325, 331, 333, 334, 335, 338, 339, 342, 344, 347, 349, 350, 354, 355, 356, 358, 360, 361, 362, 363, 366, 367, 369, 372, 373, 374, 375, 376, 377, 378, 380, 382, 383, 384, 385]

Iyer.txt
cluster 1 = [331, 332, 337]
cluster 2 = [323, 328, 338]
cluster 3 = [357, 358, 371, 381]
cluster 4 = [387, 388]
cluster 5 = [399, 400]
cluster 6 = [385, 433]
cluster 7 = [438, 439]
cluster 8 = [426, 427, 468]
cluster 9 = [487, 488]
cluster 10 = [298, 299, 495]
cluster 11 = [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, 60, 61, 62, 63, 64, 65, 66, 67, 68, 69, 70, 71, 72, 73, 74, 75, 76, 77, 78, 79, 80, 81, 82, 83, 84, 85, 86, 87, 88, 89, 90, 91, 92, 93, 94, 95, 96, 97, 98, 99, 100, 101, 102, 103, 104, 105, 106, 107, 108, 109, 110, 111, 112, 113, 114, 115, 116, 117, 118, 119, 120, 121, 122, 123, 124, 125, 126, 127, 128, 129, 130, 131, 132, 133, 134, 135, 136, 137, 138, 139, 140, 141, 142, 143, 144, 145, 146, 147, 148, 149, 150, 151, 152, 153, 154, 155, 156, 157, 158, 159, 160, 161, 162, 163, 164, 165, 166, 167, 168, 169, 170, 171, 172, 173, 174, 175, 176, 177, 178, 179, 180, 181, 182, 183, 184, 185, 186, 187, 188, 189, 190, 191, 192, 193, 194, 195, 196, 197, 198, 199, 200, 201, 202, 203, 204, 205, 206, 207, 208, 209, 210, 211, 212, 213, 214, 215, 216, 217, 218, 219, 220, 221, 222, 223, 224, 225, 226, 227, 228, 229, 230, 231, 232, 233, 234, 235, 236, 237, 238, 239, 240, 241, 242, 243, 244, 245, 246, 247, 248, 249, 250, 251, 252, 253, 254, 255, 256, 257, 258, 259, 260, 261, 264, 265, 266, 267, 268, 269, 270, 271, 272, 273, 274, 275, 276, 277, 278, 279, 280, 281, 282, 283, 284, 285, 286, 287, 288, 289, 291, 293, 294, 295, 296, 297, 300, 301, 302, 303, 304, 305, 306, 307, 308, 309, 311, 312, 313, 314, 315, 316, 317, 318, 319, 321, 322, 329, 330, 335, 336, 339, 340, 341, 379, 384, 393, 394, 396, 401, 402, 403, 404, 405, 406, 408, 409, 410, 412, 413, 414, 416, 417, 418, 420, 421, 423, 425, 428, 429, 430, 431, 432, 436, 442, 443, 447, 450, 451, 452, 453, 454, 458, 459, 460, 461, 462, 465, 466, 467, 473, 474, 476, 477, 478, 479, 496, 498]
Outliers = [262, 263, 290, 292, 310, 320, 324, 325, 326, 327, 333, 334, 342, 343, 344, 345, 346, 347, 348, 349, 350, 351, 352, 353, 354, 355, 356, 359, 360, 361, 362, 363, 364, 365, 366, 367, 368, 369, 370, 372, 373, 374, 375, 376, 377, 378, 380, 382, 383, 386, 389, 390, 391, 392, 395, 397, 398, 407, 411, 415, 419, 422, 424, 434, 435, 437, 440, 441, 444, 445, 446, 448, 449, 455, 456, 457, 463, 464, 469, 470, 471, 472, 475, 480, 481, 482, 483, 484, 485, 486, 489, 490, 491, 492, 493, 494, 497, 499, 500, 501, 502, 503, 504, 505, 506, 507, 508, 509, 510, 511, 512, 513, 514, 515, 516]