Homework 2

Akash Choudhuri

Problem 1:

The table for the first iteration is:

Point/ Cluster Center	C1	C2	C3	Assignment
A1	8	7	3	C3
A2	7	2	8	C2
A3	8	7	1	C3
A4	6	3	3	C2
A5	10	5	7	C2
A6	2	7	5	C1
A7	7	2	6	C2
A8	3	6	10	C1
A9	5	4	2	C3

New Centroids are:

C1= (5+2+3)/3,(1-1+1)/3 = (3.33,0.33)

C2= (-1+2-2+0+1)/5,(4+6+6+5+4)/5 = (0,5)

C3= (7+5+3+5)/4, (5+7+6+6)/4 = (5,6)

Problem 2

MIN

Points	A	B	C	D	E
A	0				
B	5.5	0			
C	3.5	7.5	0		
D	5.0	7.0	4.0	0	
E	1.0	6.5	6.0	4.5	0

A,E is merged first

B= min(5.5,6.5)= 5.5

C= min(3.5,6.0)= 3.5

D= min(5.0,4.5)= 4.5

The Updated Table is

Points	A,E	B	C	D
A,E	0			
B	5.5	0		
C	3.5	7.5	0	
D	4.5	7.0	4.0	0

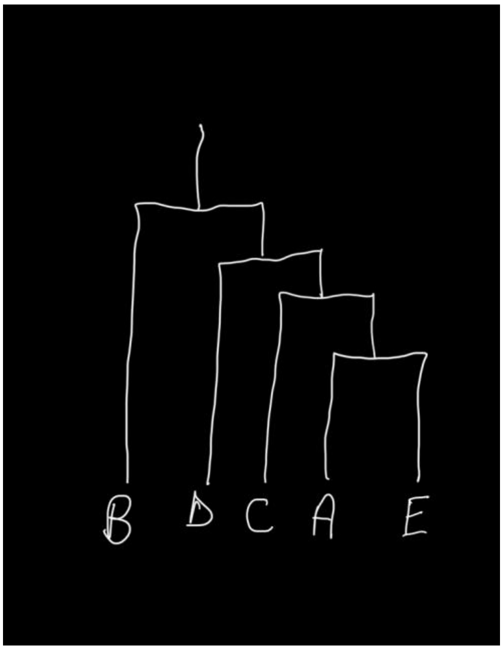
C is merged with (A,E)
B= min(5.5,7.5) = 5.5
D= min(4.5,4.0) = 4.0

The updated table is

Points	A,E,C	B	D
A,E,C	0		
B	5.5	0	
D	4.0	7.0	0

D is merged with (A,E,C)

Finally B is merged with (A,E,C,D).
The resultant Dendogram is:



MAX

Points	A	B	C	D	E
A	0				
B	5.5	0			
C	3.5	7.5	0		

D	5.0	7.0	4.0	0	
E	1.0	6.5	6.0	4.5	0

A,E is merged first
B= max(5.5,6.5)= 6.5
C= max(3.5,6.0)= 6.0
D= max(5.0,4.5)= 5.0

The Updated Table is

Points	A,E	B	C	D
A,E	0			
B	6.5	0		
C	6.0	7.5	0	
D	5.0	7.0	4.0	0

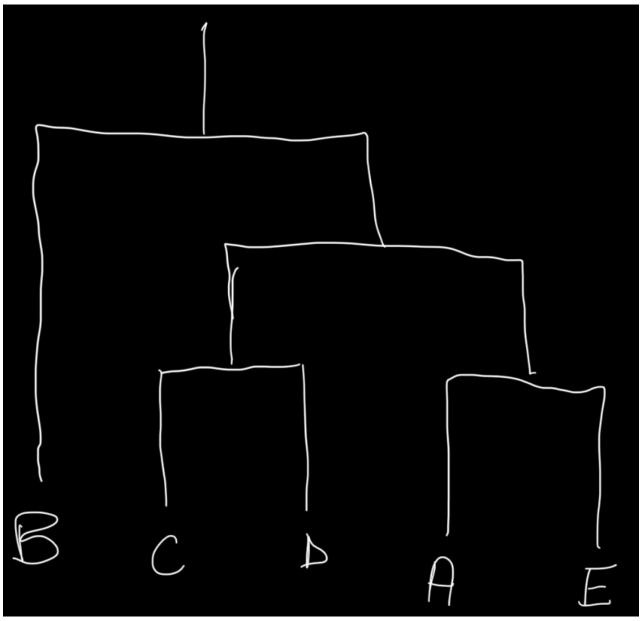
C is merged with D
B= max(7.5,7.0)=7.5
(A,E)= MAX(6.0,5.0)=6.0

The updated table is

Points	A,E	C,D	B
A,E	0		
C,D	6.0	0	
B	6.5	7.5	0

(A,E) is merged with (C,D).
Finally B is merged with (A,E,C,D).

The resultant dendogram is:



AVERAGE

Points	A	B	C	D	E
A	0				
B	5.5	0			
C	3.5	7.5	0		
D	5.0	7.0	4.0	0	
E	1.0	6.5	6.0	4.5	0

A,E is merged
B= AVG(5.5,6.5) = 6.0
C= AVG(3.5,6)= 4.75
D= AVG((5.0,4.5)=4.75

The Updated Table is

Points	A,E	B	C	D
A,E	0			
B	6.0	0		
C	4.75	7.5	0	
D	5.0	7.0	4.0	0

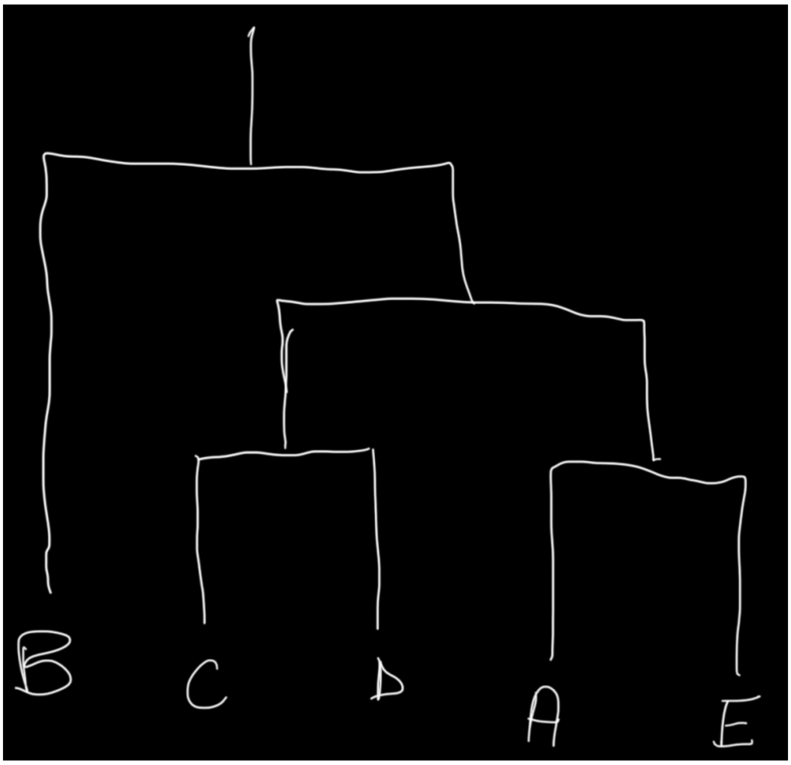
C,D is merged
B=AVG(7.5,7.0)=7.25
(A,E)= AVG(4.75,5.0)= 4.875

The updated table is

Points	A,E	C,D	B
A,E	0		
C,D	4.875	0	
B	6.0	7.25	0

(A,E) and (C,D) are merged.
B is merged with (A,E,C,D).

The resultant Dendogram is:

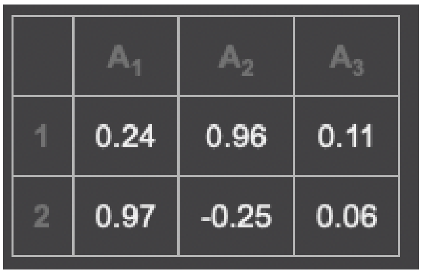


Problem 3:

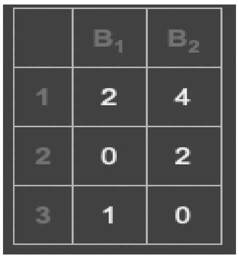
The transformation can simply be obtained by:

P= $U^T X$

=

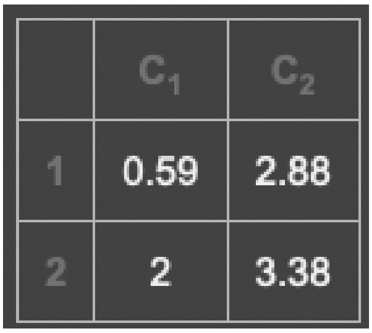


	A_1	A_2	A_3
1	0.24	0.96	0.11
2	0.97	-0.25	0.06



	B_1	B_2
1	2	4
2	0	2
3	1	0

=



	C_1	C_2
1	0.59	2.88
2	2	3.38

So,
y11= 0.59
y12=2.88
y21=2
y22=3.38

Problem 4

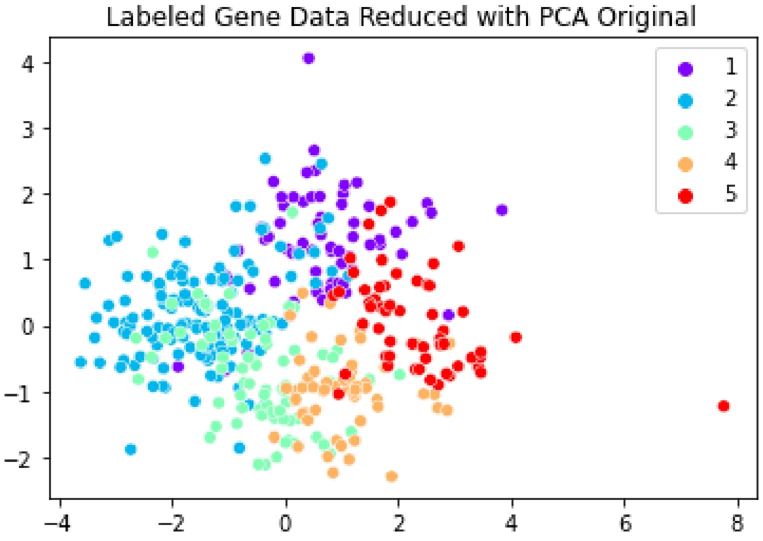
For the first part, we take the RHS of the required proof
1-g(z)
= 1-$\frac{1}{1+e^{-z}}$
= $\frac{e^{-z}}{1+e^{-z}}\frac{e^{z}}{e^{z}}$
= $\frac{1}{e^{z}+1}$
= $\frac{1}{1+e^{z}}$
= g(-z)

For the second part, we need to first show that the function is bijective (ie, one-to-one and onto).
To prove that the function is one-to-one, if we choose 2 numbers z $_1$ and z_2, we get g(z_1) = $\frac{1}{1+e^{-z_1}}$ and g(z_2) = $\frac{1}{1+e^{-z_2}}$, we can only have g(z_1)= g(z_2) if z $_1$= z_2, so the function is one-to-one.
To prove that the function is onto, we need to prove that the inverse exists, which is the original proof itself
Let x=g(z)=$\frac{1}{1+e^{-z}}$
or $e^{-z} = \frac{1}{x} - 1$
Taking logarithm in both sides, we get
z= -ln($\frac{1-x}{x}$)
or z=ln($\frac{x}{1-x}$)
This corresponds to $g^{-1}(y) = \ln\left(\frac{y}{1-y}\right)$, which proves the statement.

Problem 5

Cho.txt

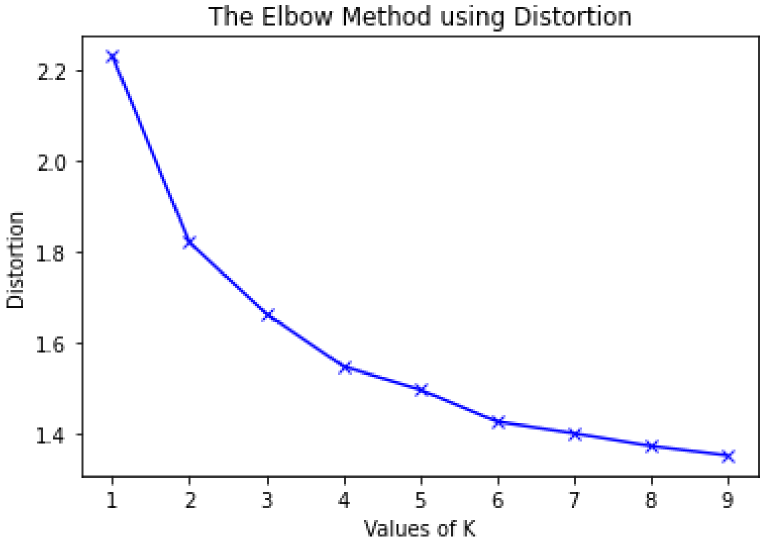
The ground Truth Cluster visualization after converting it to 2 dimensions using PCA looks like:



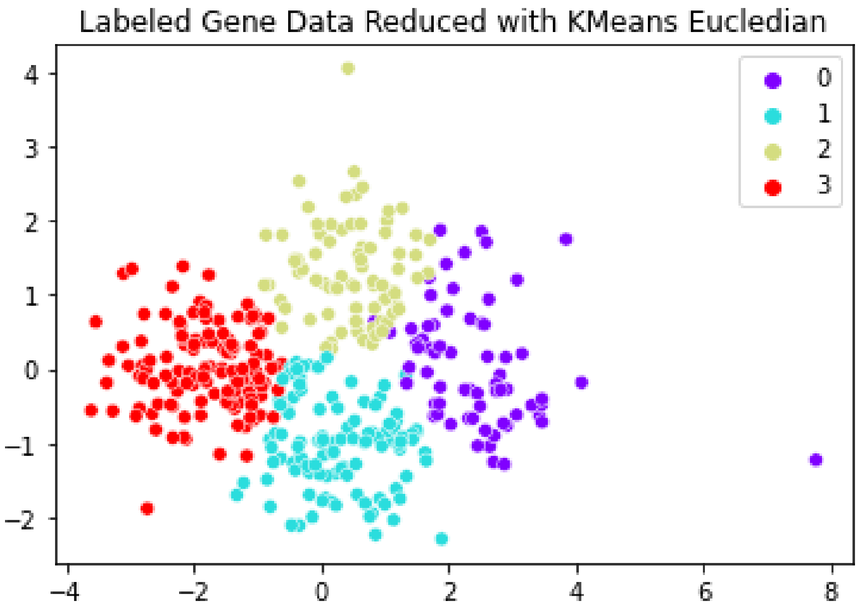
K-Means

- Euclidean Distance

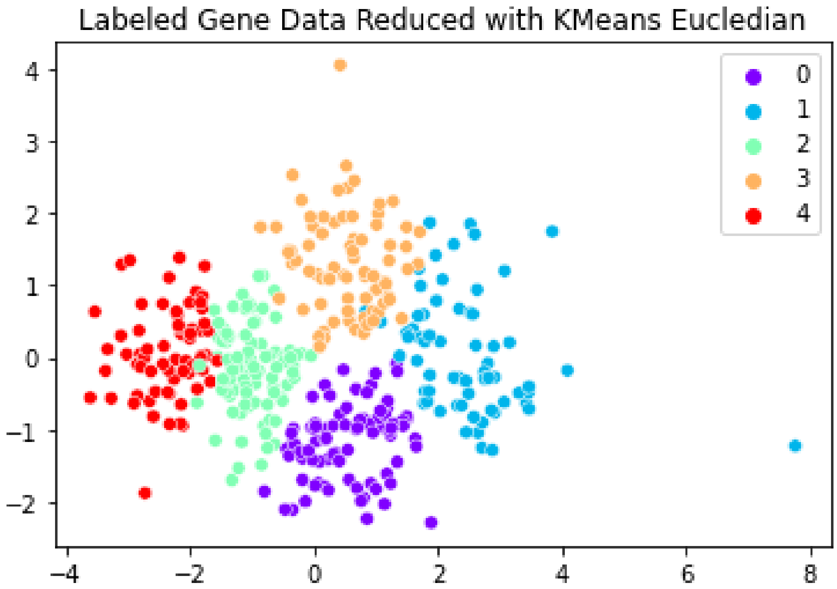
  We initially used Euclidian Distance as a distance metric. The elbow plot plotting inertia with different values of k was:



K=4 was the optimal number of cluster number. No empty clusters were found. The generated plot is, as follows:



The plot for k=5 looks like:



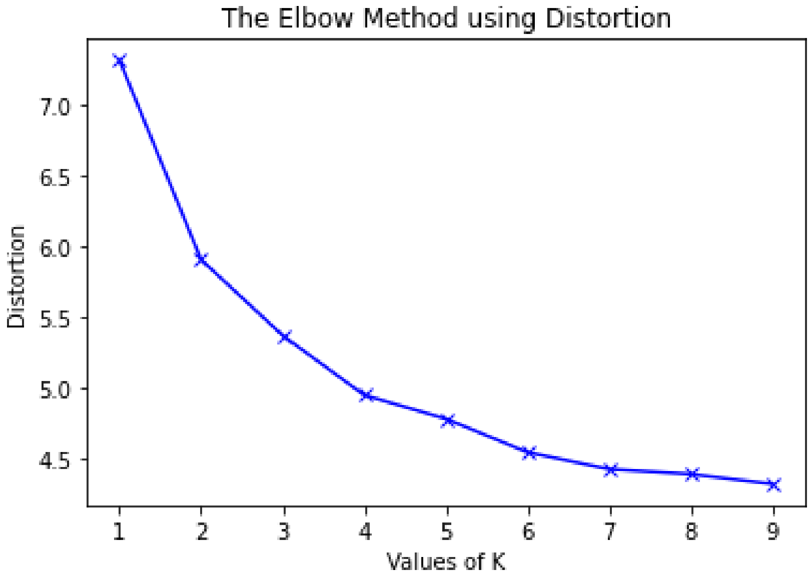
We can compare the ground truth with the output of K=5 here. The results are:

Rand index= 0.799
Jaccard Coefficient= 0.239

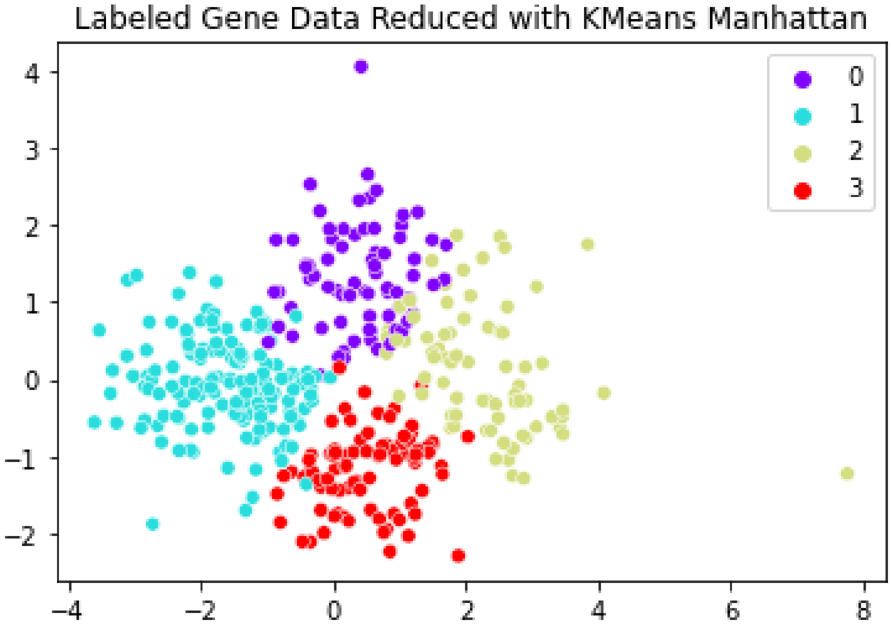
For k=4,
Rand index= 0.794
Jaccard Coefficient= 0.205
I did not find any empty clusters as cluster center initialization was one of the points.
The method does not detect outliers but instead just assigns it to the nearby cluster.
But the PCA plots of the ground truth clusters and K means for K=5 look very similar.

- Manhattan Distance

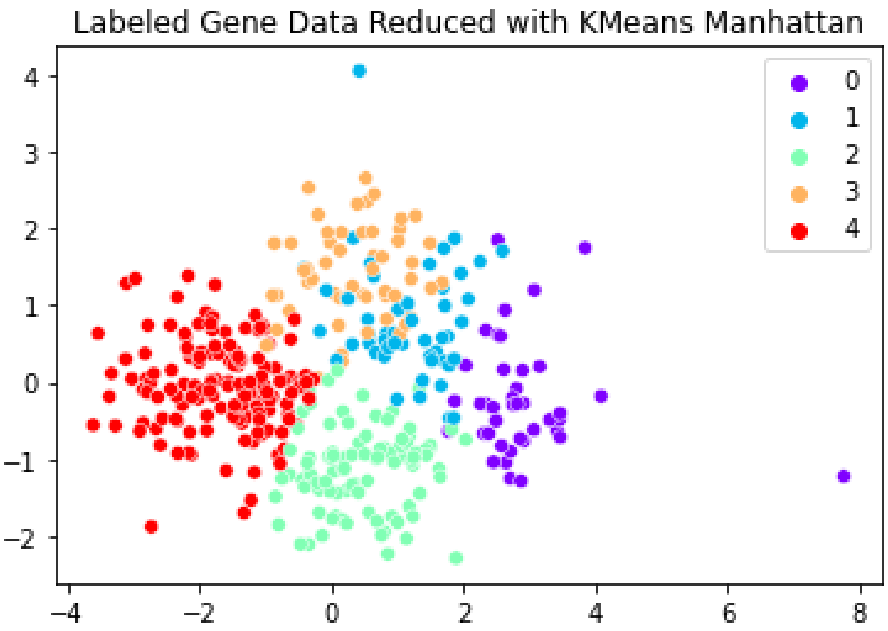
We initially used Euclidian Distance as a distance metric. The elbow plot plotting inertia with different values of k was:



K=4 was the optimal number of cluster number. No empty clusters were found. The generated plot is, as follows:



The plot for k=5 looks like:



We can compare the ground truth with the output of K=5 here. The results are:

Rand index= 0.794
Jaccard Coefficient= 0.105

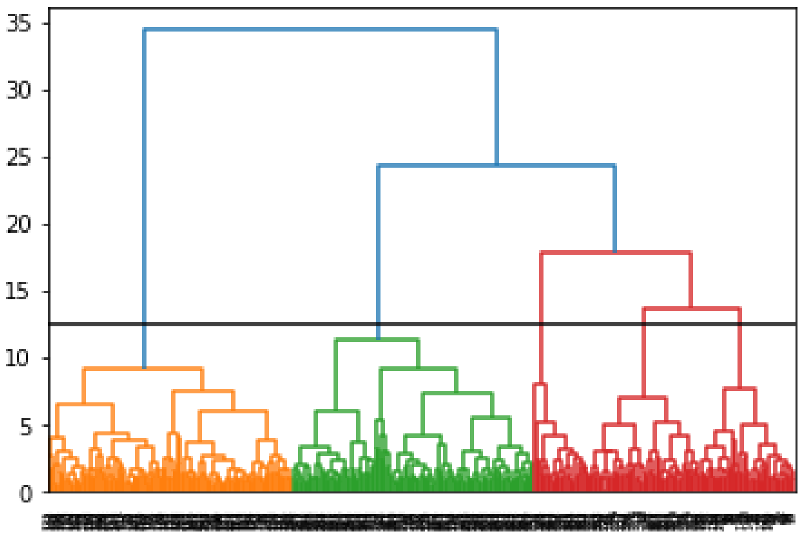
For k=4,
Rand index= 0.797
Jaccard Coefficient= 0.047

I did not find any empty clusters as cluster center initialization was one of the points. The method does not detect outliers but instead just assigns it to the nearby cluster. But the PCA plots of the ground truth clusters and K means for K=5 look very similar but not as good as Euclidean.

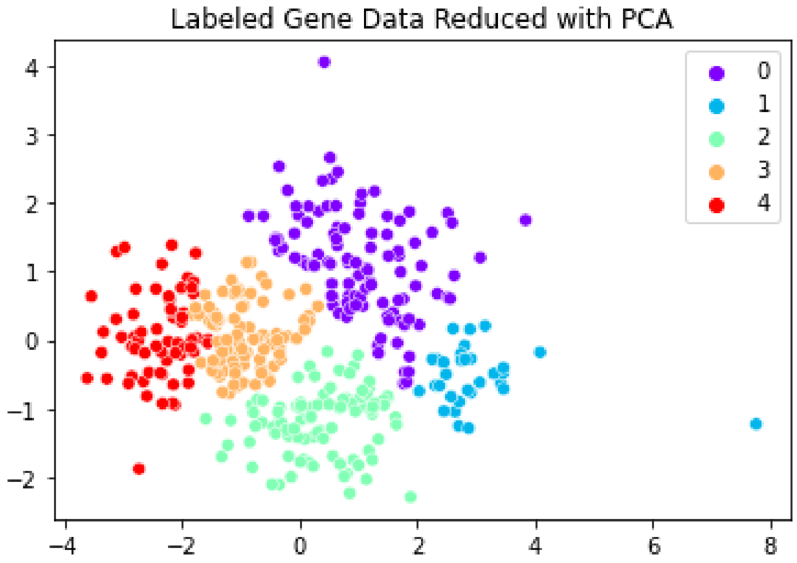
Hierarchical Agglomerative Clustering

- Method: Ward variance minimization algorithm, Distance: Euclidian

The Data in its original dimension is used to generate dendrograms to understand the optimal number of clusters. The dendrogram is, as follows:



We decide to find the number of clusters using the dendrogram by finding the largest horizontal space that doesn't have any vertical lines (the space with the longest vertical lines). This means that there's more separation between the clusters. We can draw a horizontal line that passes through that longest distance at 12.5. 5 seems a good indication of the number of clusters that have the most distance between them. Using this number, we assign clusters to data points. The result in 2D using PCA is given as follows:

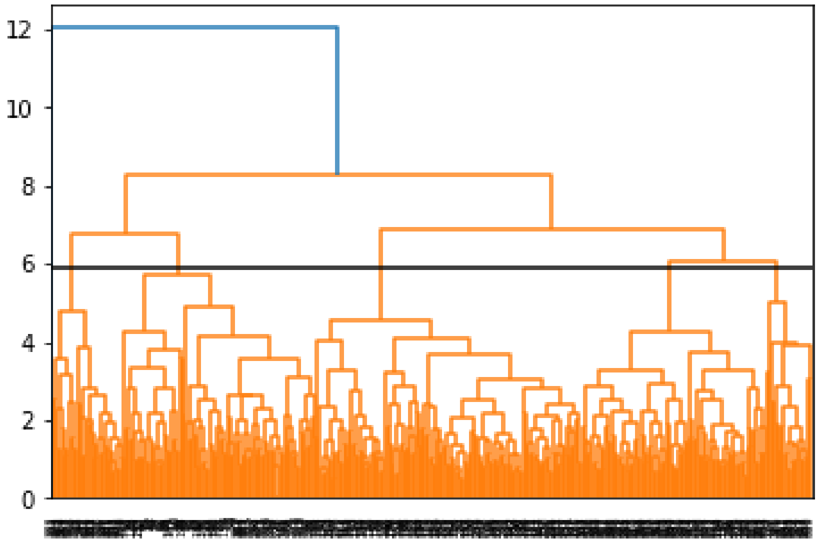


We see that in the 2D space, out cluster assignments seem to be more pronounced and better than the ground truths. This means that we still need more features to effectively cluster the data in the way ground truth has done.
Rand index= 0.758
Jaccard Coefficient= 0.170

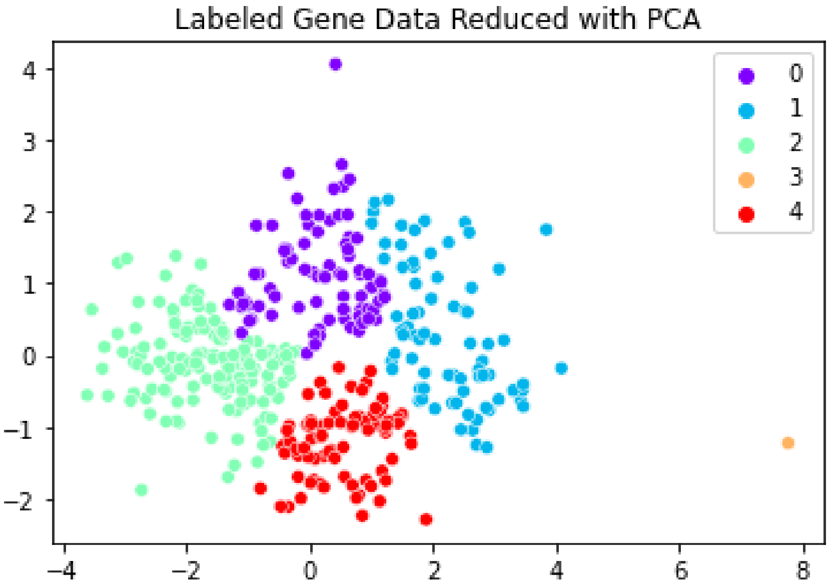
The reason why Jaccard Coefficient is low is because the cluster numbers do not correspond to the actual numbers. So, the only way we can compare is by checking other performance metrics (shown later).

- Method: Complete, Distance: Minkowski

The Data in its original dimension is used to generate dendrograms to understand the optimal number of clusters. The dendrogram is, as follows:



We decide to find the number of clusters using the dendrogram by finding the largest horizontal space that doesn't have any vertical lines (the space with the longest vertical lines). This means that there's more separation between the clusters. We can draw a horizontal line that passes through that longest distance at 5.9. 5 seems a good indication of the number of clusters that have the most distance between them. Using this number, we assign clusters to data points. The result in 2D using PCA is given as follows:



Notice that Cluster 3 has only 1 point. It seems to be weird to use 1 whole cluster for just an outlier point. So, the cluster assignment is not a good as the previous method.

Rand index= 0.785
Jaccard Coefficient= 0.130

Model/Metric	**Silhoutte Score**	**Calinski-Harabasz Index**	**Davies- Bouldin Index**
Ground Truth	0.086	71.39	1.97
Ward	0.161	106.77	1.54
Complete	0.224	100.69	1.22

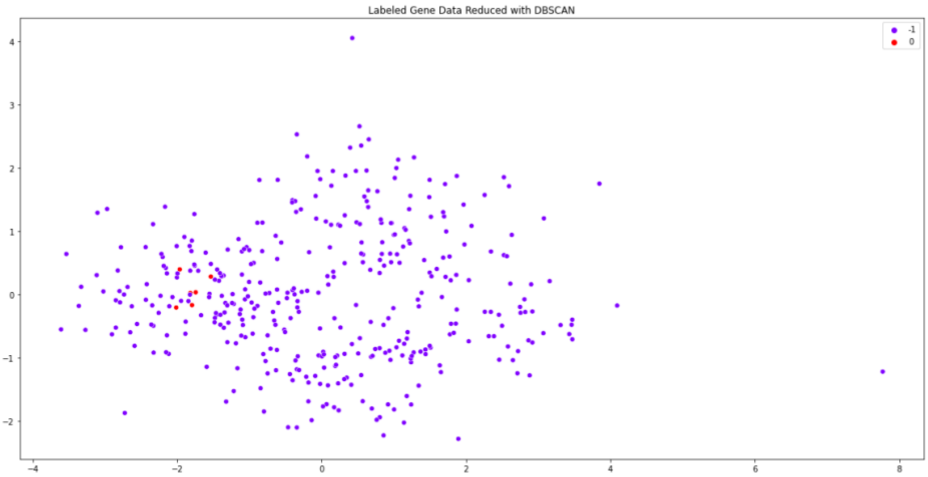
So, by this, we can understand that the Ward Method does better than Complete compared to the true ground truth clusters. The problem of both the methods of HAC is that both of them are heavily biased towards globular clusters.

Density Based

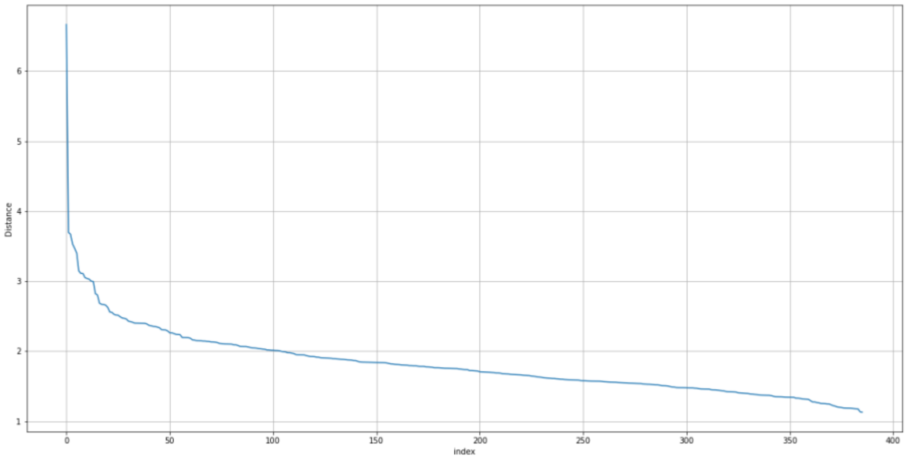
We choose the optimal value of the hyperparameters by taking the Minimum Silhoutte Score. The best value is given below and the candidate values are given in silscore.csv.

Score	Parameters
-0.11902	Eps: 0.7, min_sample: 4

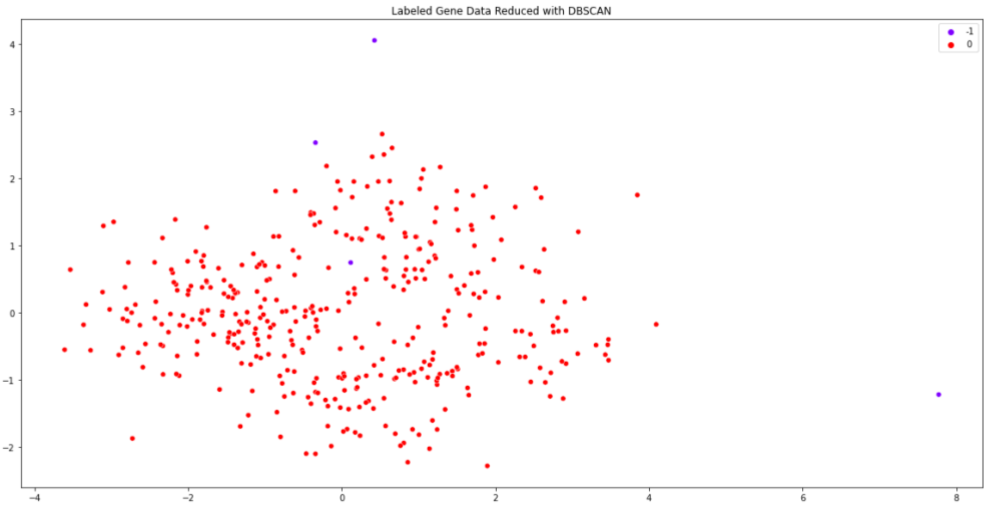
With this value the clustering is:



Another way to choose the optimal eps would be to check the elbow plot. We keep the min_samples to its default value and the elbow plot thus generated is given as follows:



The elbow is seen at 2.5. Setting the eps at that value the resultant clustering plot is:



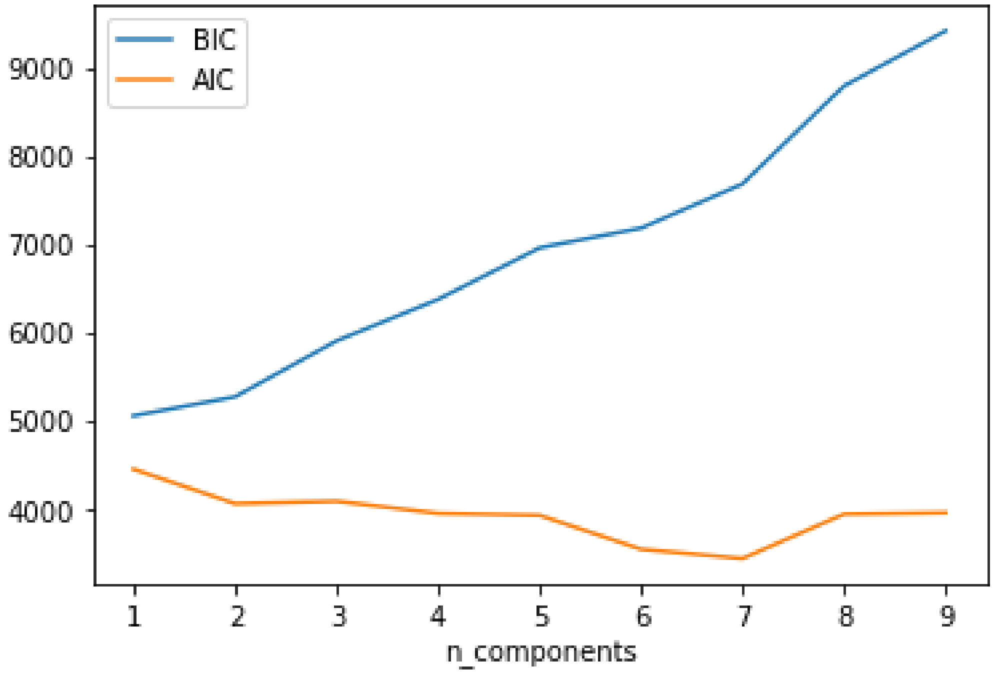
We can see that density based clustering does not do a good job as the clusters are not dense enough but has a more hierarchal relationship. Either way, the performance metrics are:

Model/Metric	**Silhoutte Score**	**Calinski-Harabasz Index**	**Davies- Bouldin Index**
Ground Truth	0.086	71.39	1.97
DBSCAN	0.446	6.34	2.26

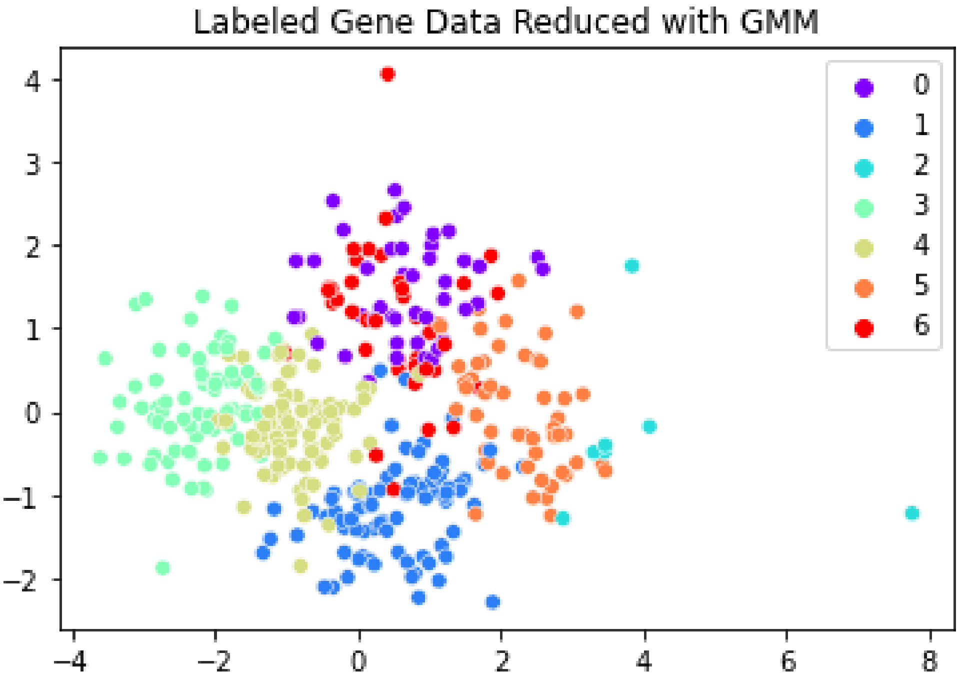
Rand index= 0.238
Jaccard Coefficient= 0

Gaussian Mixture Model

GMM is a generative model determining the optimal number of components for a given dataset. A generative model is inherently a probability distribution for the dataset, and so we can simply adjust the model likelihoods using some analytic criterion such as the Akaike information criterion (AIC) or the Bayesian information criterion (BIC). This will help us in choosing the optimal number of n_components. The plots is:

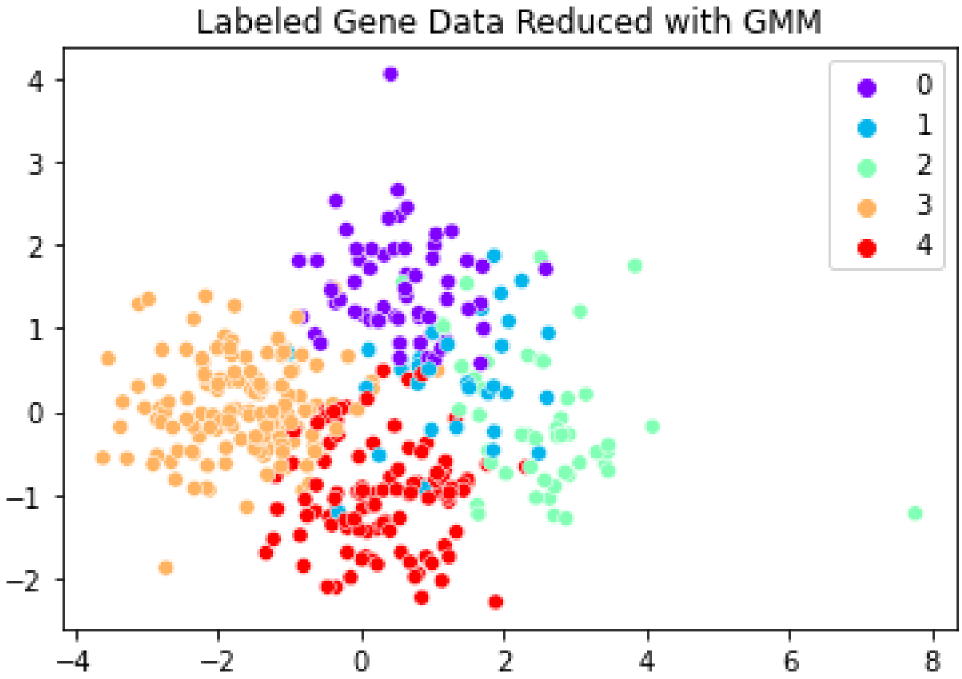


We see that is we have 7 components, AIC is minimized. The resultant Clusters are:



Model/Metric	Silhoutte Score	Calinski-Harabasz Index	Davies- Bouldin Index
Ground Truth	0.086	71.39	1.97
GMM	0.147	86.06	1.66

As we can see the performance metrics values are very close to the ground truths.
If we set n_components=5, the plot is:



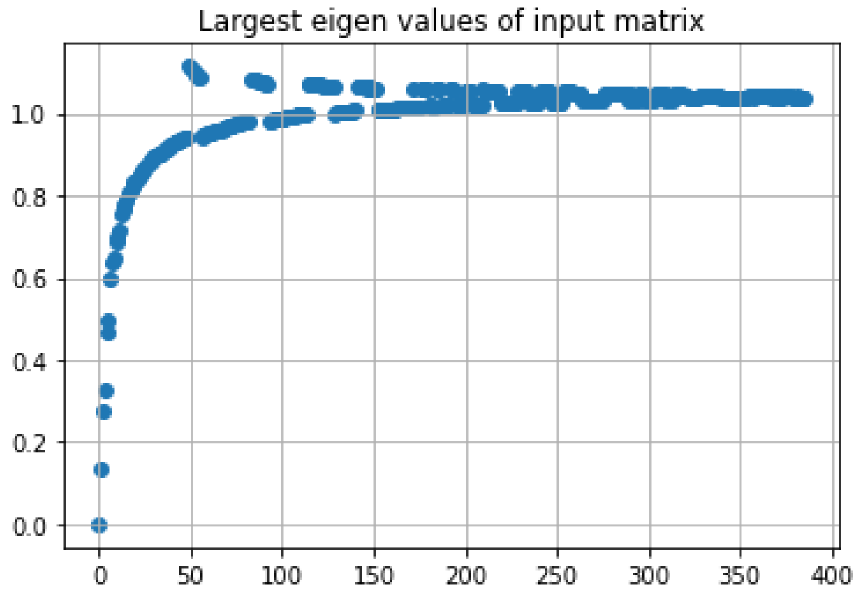
The other Performance metrics are:
Rand index= 0.7814
Jaccard Coefficient= 0.058

For n_components=5,
Rand index= 0.786
Jaccard Coefficient= 0.105

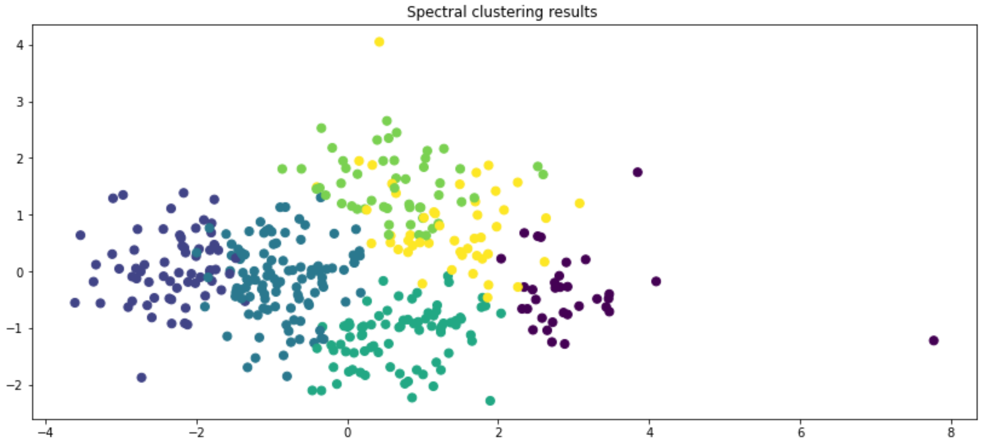
This algorithm does a good job and better than K-Means as clusters do intersect.

Spectral Clustering

We choose the optimal cluster by identifying the maximum gap which corresponds to the number of clusters by eigengap heuristic. The plot is:



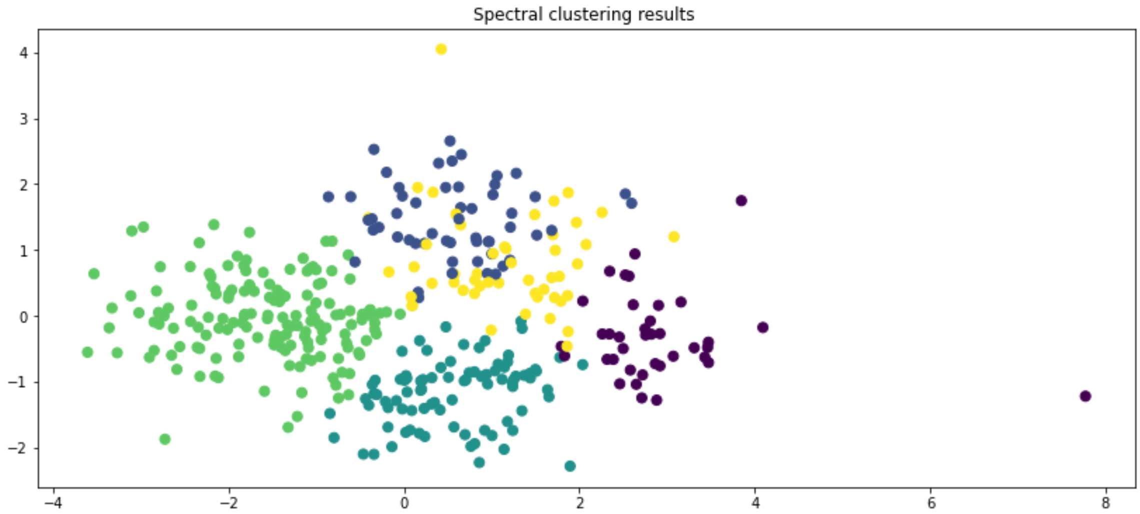
We see that is we have 6 clusters, maximum eigen gap is minimized. The resultant Clusters are:



Model/Metric	Silhoutte Score	Calinski-Harabasz Index	Davies- Bouldin Index
Ground Truth	0.086	71.39	1.97
SC	0.189	105.66	1.56

As we can see the performance metrics values are very close to the ground truths.

If we set number of clusters=5, the plot is:

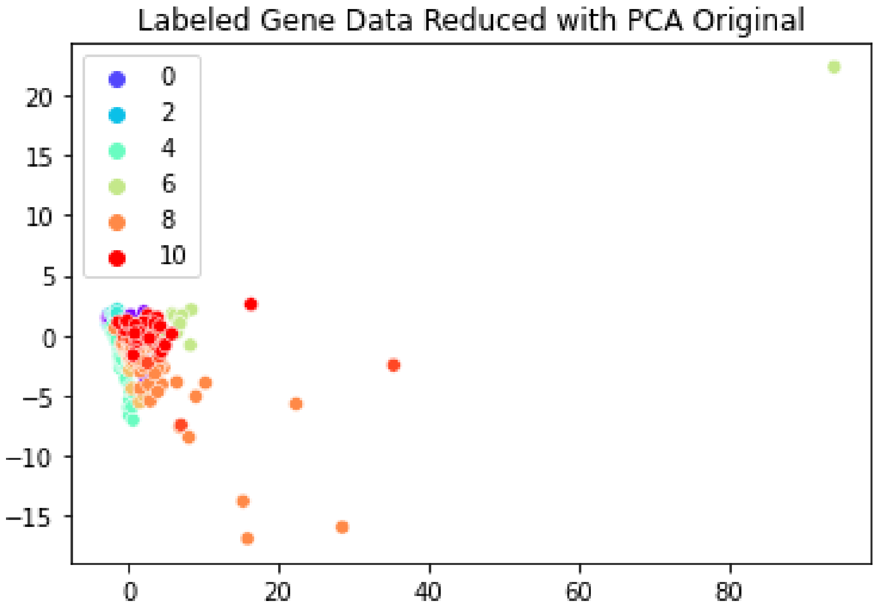


The other Performance metrics are:
Rand index= 0.779
Jaccard Coefficient= 0.153

For n_components=5,
Rand index= 0.799
Jaccard Coefficient= 0.112
This method performs really well as well. But not as good as GMM. It is not susceptible to outliers. But it might overfit to globular clusters.

Iyer.txt

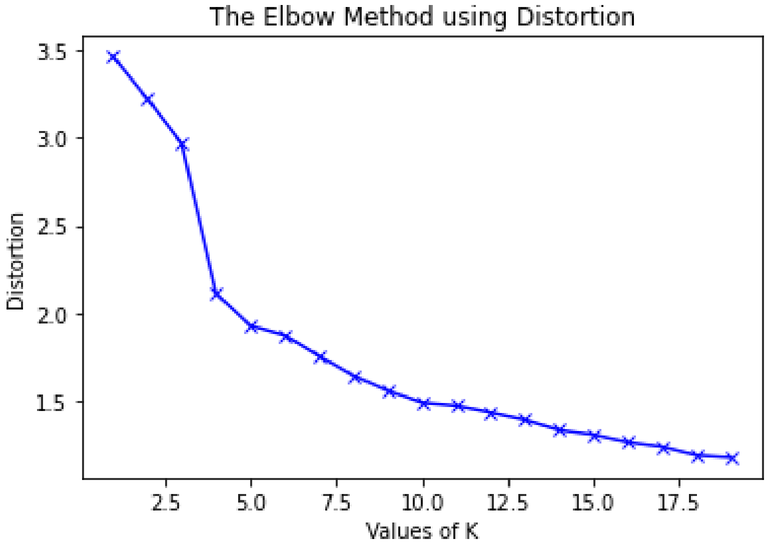
The ground Truth Cluster visualization after converting it to 2 dimensions using PCA looks like:



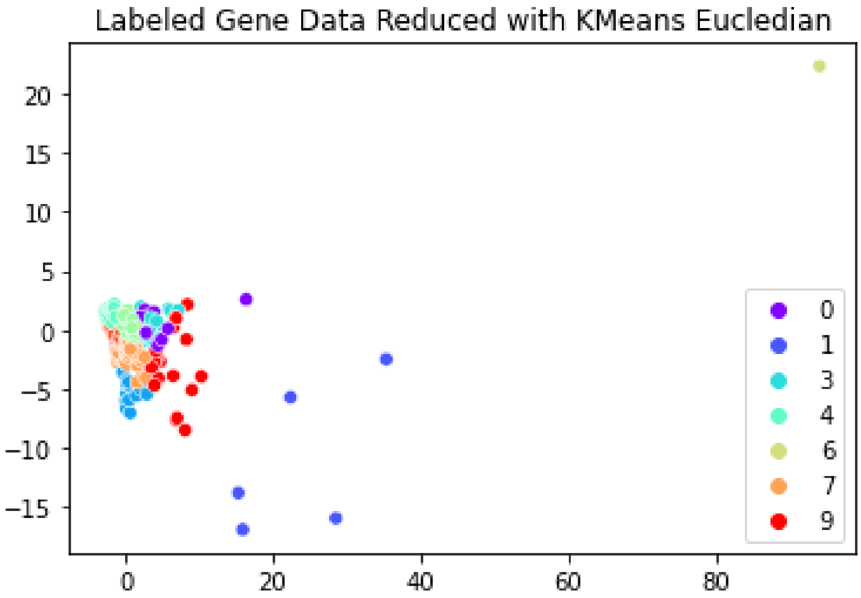
K-Means

- ### Euclidean Distance

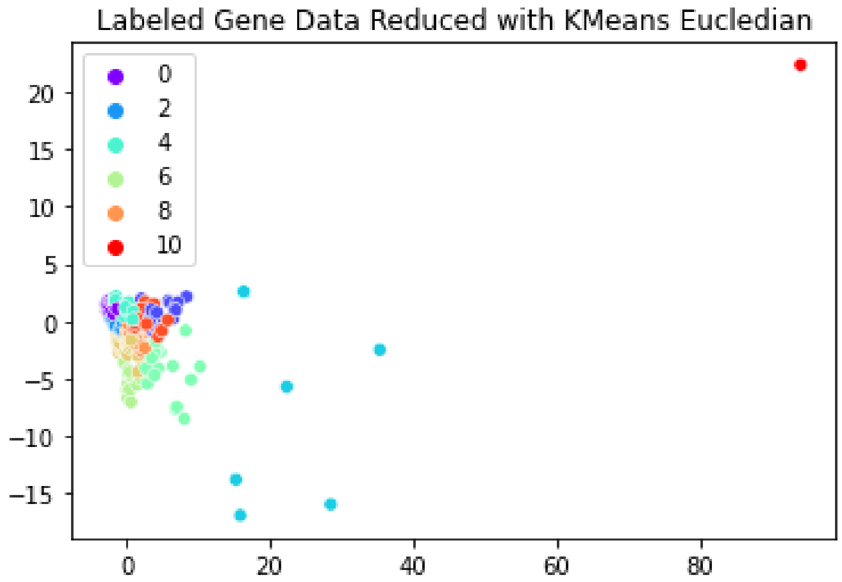
We initially used Euclidian Distance as a distance metric. The elbow plot plotting inertia with different values of k was:



K=10 was the optimal number of cluster number. No empty clusters were found. The generated plot is, as follows:



The plot for k=11 looks like:



We can compare the ground truth with the output of K=11 here. The results are:

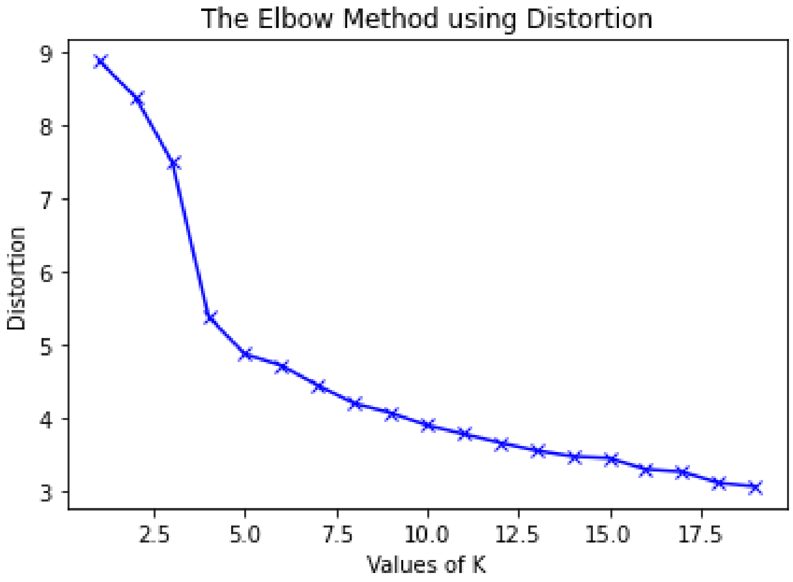
Rand index= 0.8040
Jaccard Coefficient= 0.112

For k=10,
Rand index= 0.777
Jaccard Coefficient= 0.032

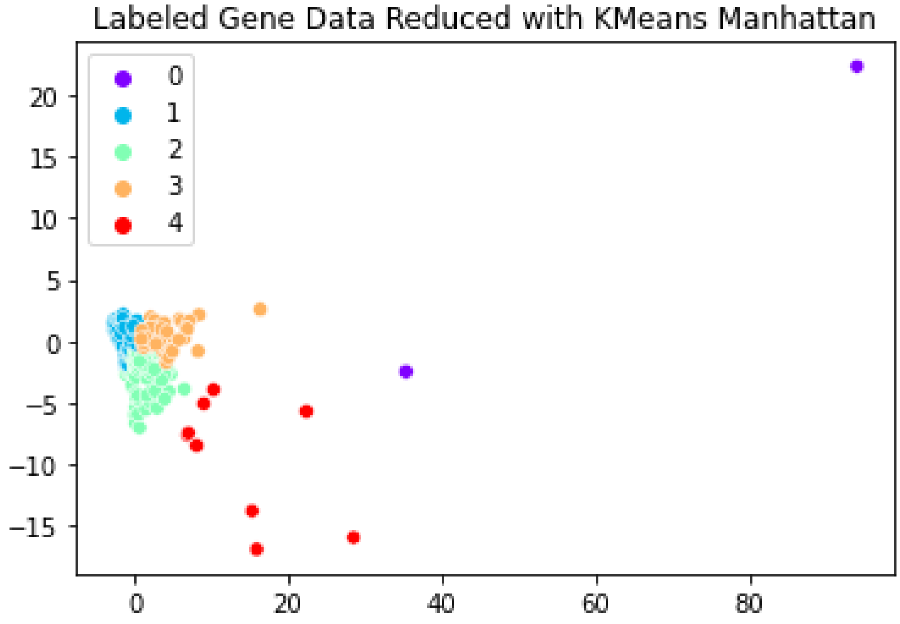
I did not find any empty clusters as cluster center initialization was one of the points. The method does not detect outliers but instead just assigns it to the nearby cluster. But if the right number of clusters is given it does detect outliers. But the PCA plots of the ground truth clusters and K means for K=11 look very similar.

- Manhattan Distance

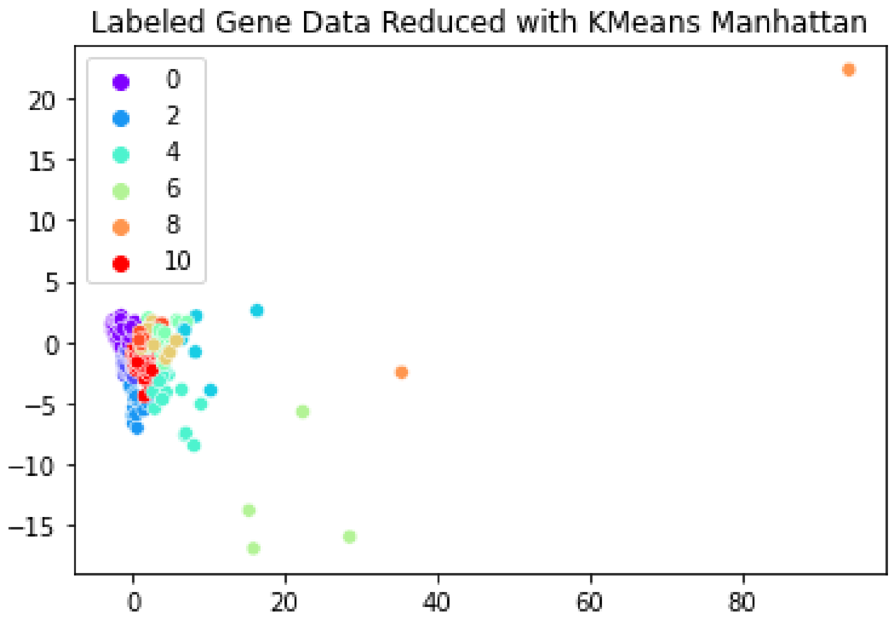
We initially used Euclidian Distance as a distance metric. The elbow plot plotting inertia with different values of k was:



K=5 was the optimal number of cluster number. No empty clusters were found. The generated plot is, as follows:



The plot for k=11 looks like:



We can compare the ground truth with the output of K=11 here. The results are:

Rand index= 0.708
Jaccard Coefficient= 0.098

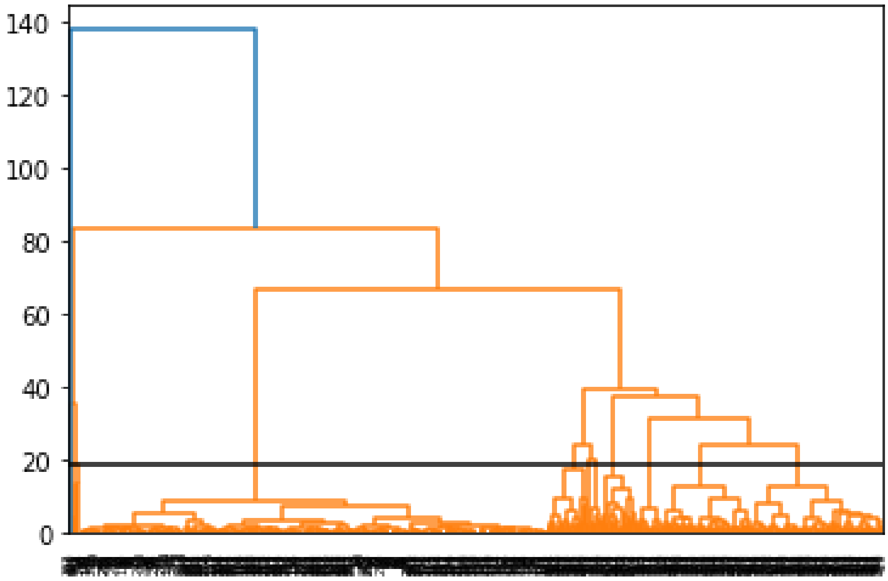
For k=5,
Rand index= 0.617
Jaccard Coefficient= 0.037

I did not find any empty clusters as cluster center initialization was one of the points. The method does not detect outliers but instead just assigns it to the nearby cluster. But if right number of clusters are given, it does detect outliers. But the PCA plots of the ground truth clusters and K means for K=11 look very similar but not as good as Euclidean.

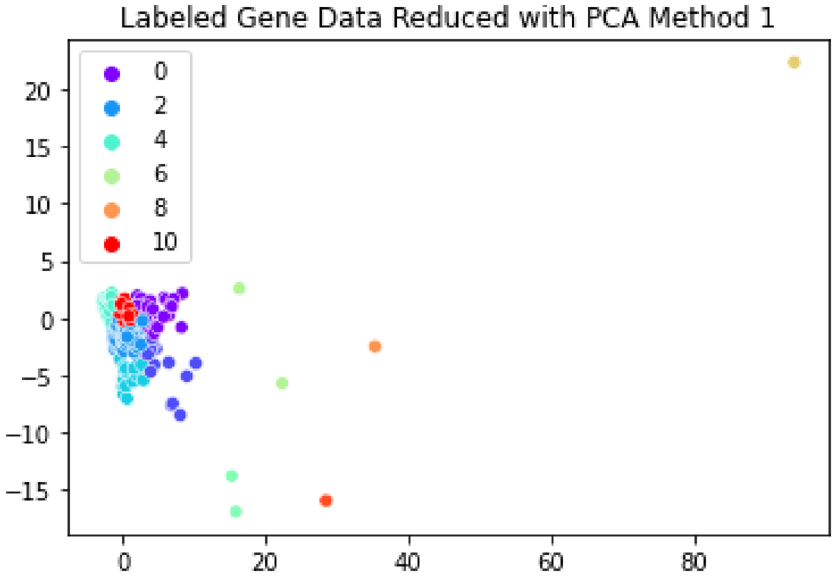
Hierarchical Agglomerative Clustering

- ### Method: Ward variance minimization algorithm, Distance: Euclidian

The Data in its original dimension is used to generate dendrograms to understand the optimal number of clusters. The dendrogram is, as follows:



We decide to find the number of clusters using the dendrogram by finding the largest horizontal space that doesn't have any vertical lines (the space with the longest vertical lines). This means that there's more separation between the clusters. We can draw a horizontal line that passes through that longest distance at 19. 11 seems a good indication of the number of clusters that have the most distance between them. Using this number, we assign clusters to data points. The result in 2D using PCA is given as follows:



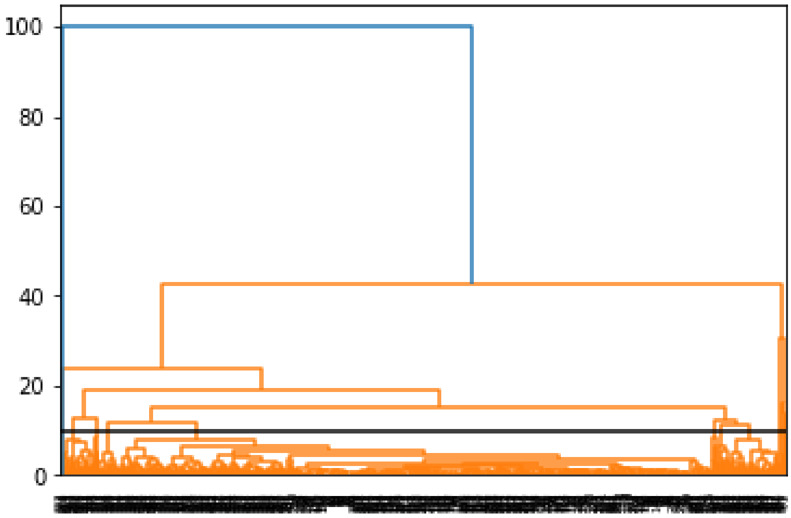
Rand index= 0.719
Jaccard Coefficient= 0.026

We see that in the 2D space, outlier points are present and needs to be taken care of, which this model does.

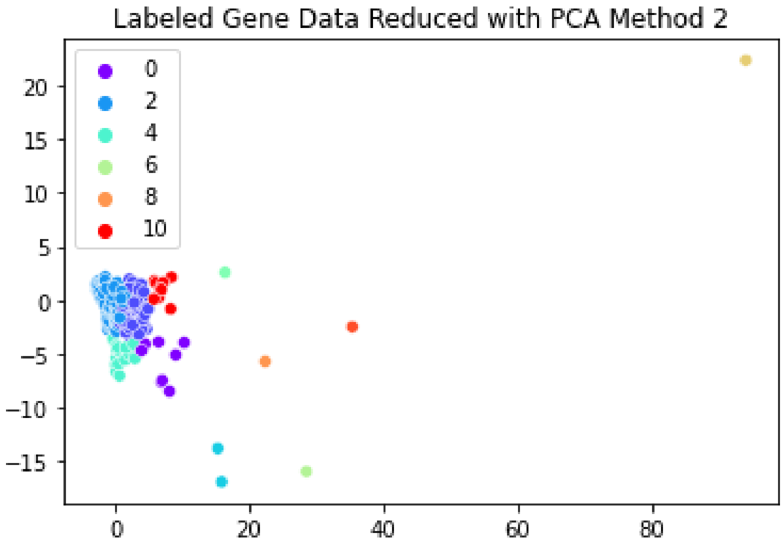
The reason why Jaccard Coefficient is low is because the cluster numbers do not correspond to the actual numbers. So, the only way we can compare is by checking other performance metrics (shown later).

- Method: Complete, Distance: Minkowski

The Data in its original dimension is used to generate dendrograms to understand the optimal number of clusters. The dendrogram is, as follows:



We decide to find the number of clusters using the dendrogram by finding the largest horizontal space that doesn't have any vertical lines (the space with the longest vertical lines). This means that there's more separation between the clusters. We can draw a horizontal line that passes through that longest distance at 10.
The resultant clusters in 2D are given by:



Rand index= 0.515
Jaccard Coefficient= 0.0067

Thus, it is easily understood that both the above methods of HAC are less susceptible to outliers. But they are more biased towards globular clusters.

Model/Metric	**Silhoutte Score**	**Calinski-Harabasz Index**	**Davies- Bouldin Index**
Ground Truth	-0.033	20.23	2.39
Ward	0.161	387.64	0.98
Complete	0.513	325.90	0.62

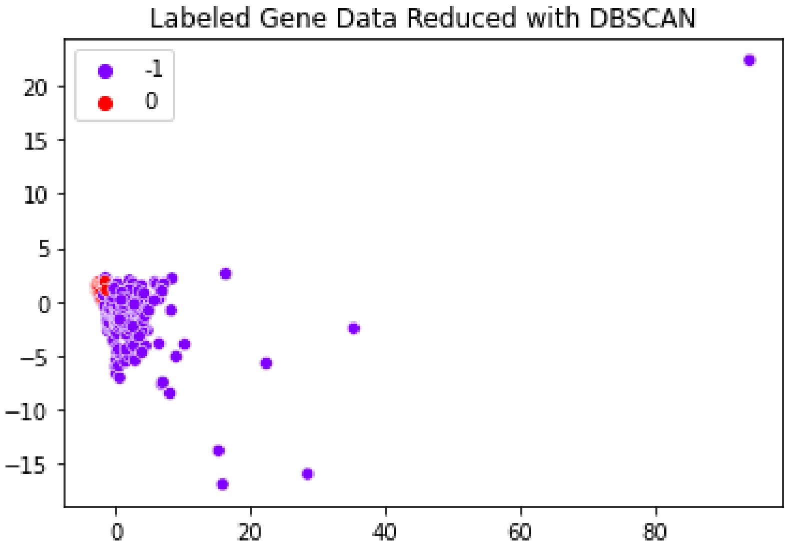
We can see that none of the above methods perform anywhere close as the ground truths. Thus, some features may be missing and may be required to do effective clustering corresponding to the ground truths.

Density Based

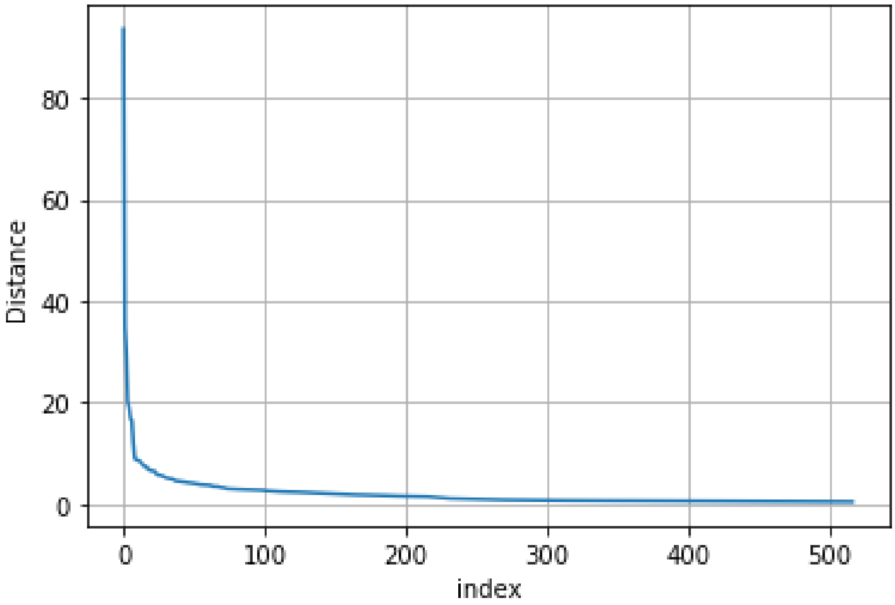
We choose the optimal value of the hyperparameters by taking the Minimum Silhoutte Score. The best value is given below and the candidate values are given in silscore1.csv.

Score	Parameters
0.3713	Eps: 0.7, min_sample: 4

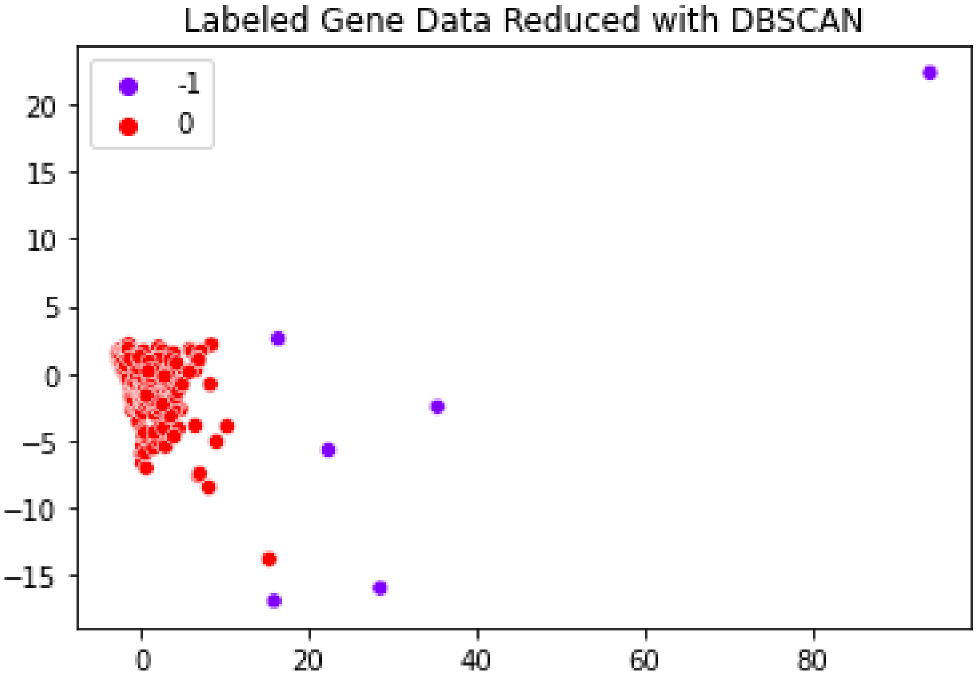
With this value the clustering is:



Another way to choose the optimal eps would be to check the elbow plot. We keep the min_samples to its default value and the elbow plot thus generated is given as follows:



The elbow seems to have happened at 10, so we choose that value and the resultant cluster is, as follows:



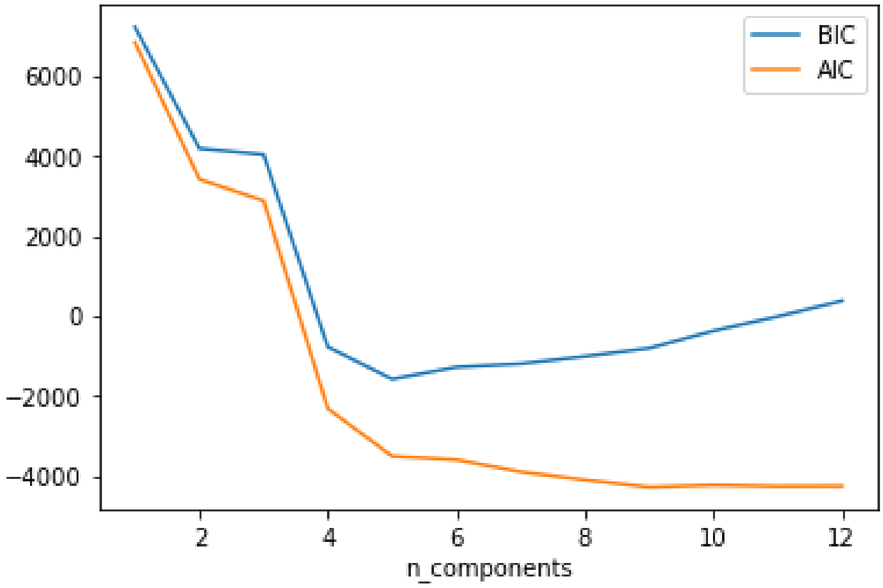
This looks like a better cluster. The performance metrics are, as follows:

Model/Metric	**Silhoutte Score**	**Calinski-Harabasz Index**	**Davies- Bouldin Index**
Ground Truth	0.086	71.39	1.97
DBSCAN	0.88	306.46	0.83

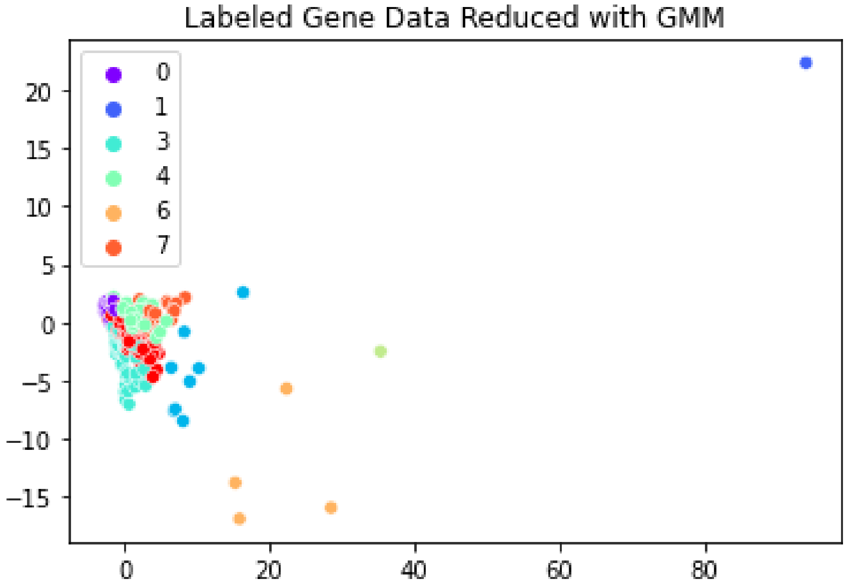
Rand index= 0.172
Jaccard Coefficient= 0
Density based clustering does not do well as most points are clustered together and no fine graining can be done.

Gaussian Mixture Model

GMM is a generative model determining the optimal number of components for a given dataset. A generative model is inherently a probability distribution for the dataset, and so we can simply adjust the model likelihoods using some analytic criterion such as the Akaike information criterion (AIC) or the Bayesian information criterion (BIC). This will help us in choosing the optimal number of n_components. The plots is:

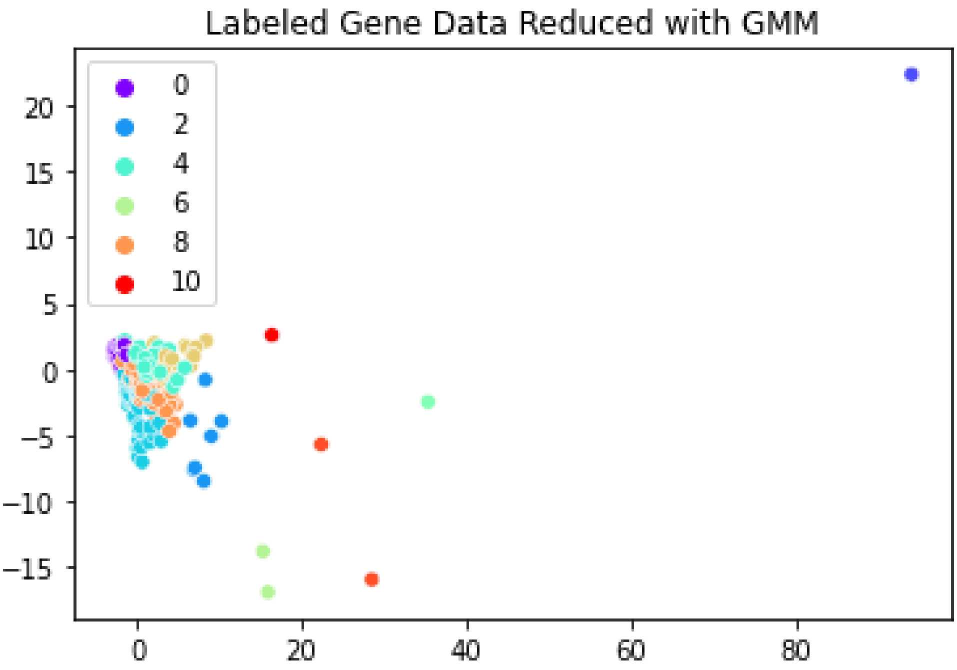


We see that is we have 9 components, AIC is minimized. The resultant Clusters are:



Model/Metric	**Silhoutte Score**	**Calinski-Harabasz Index**	**Davies- Bouldin Index**
Ground Truth	0.086	71.39	1.97
GMM	0.414	398.45	1.01

As we can see the performance metrics values are very close to the ground truths.
If we set n_components=11, the plot is:



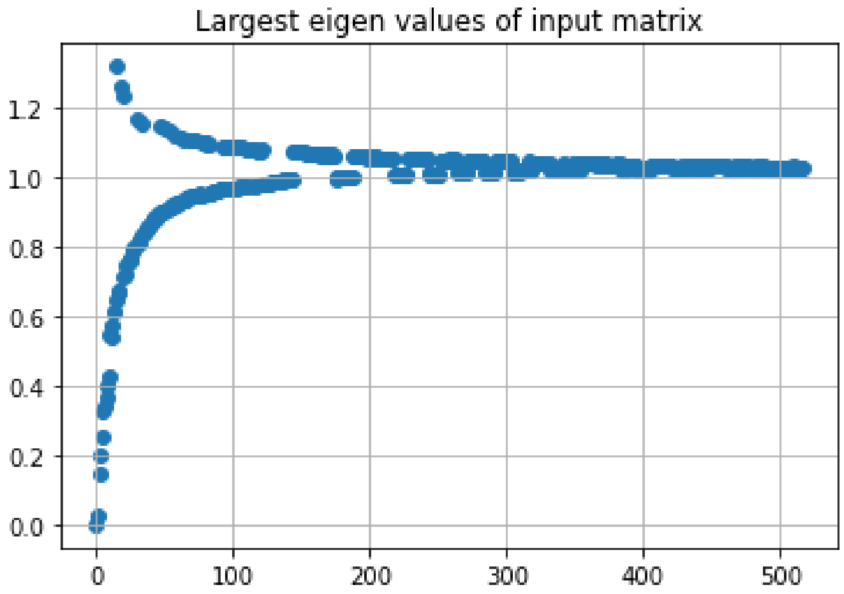
The other Performance metrics are:
Rand index= 0.7613
Jaccard Coefficient= 0.102

For n_components=9,
Rand index= 0.7613
Jaccard Coefficient= 0.102

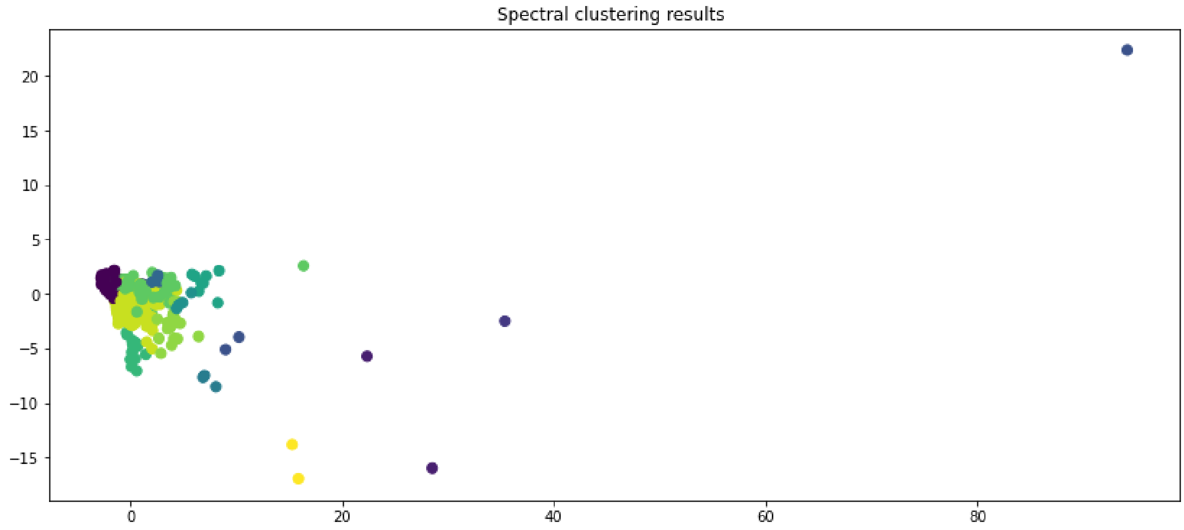
This algorithm does a good job and detects outliers as well as intersecting clusters.

Spectral Clustering

We choose the optimal cluster by identifying the maximum gap which corresponds to the number of clusters by eigengap heuristic. The plot is:



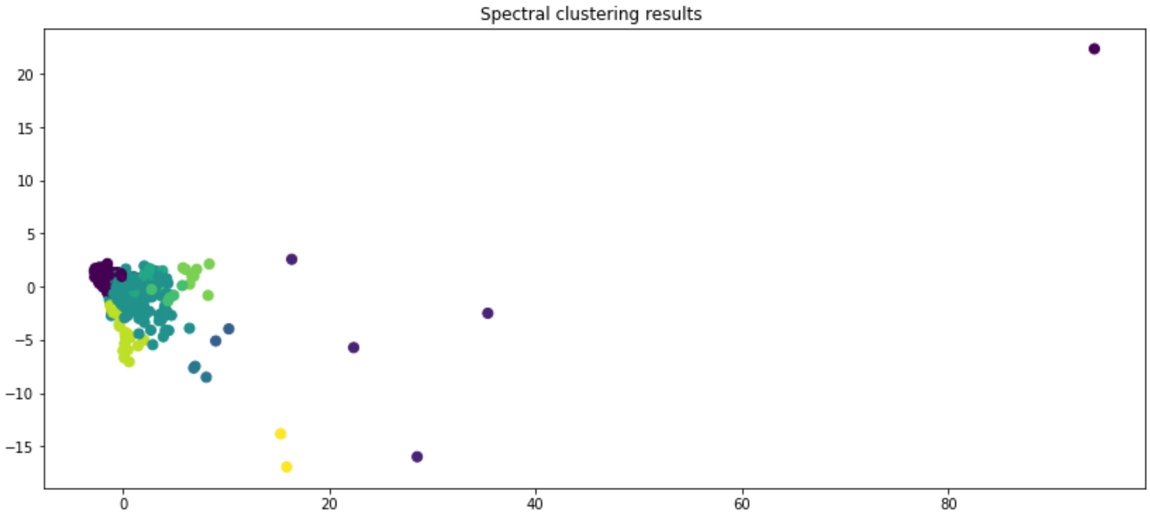
We see that is we have 14 clusters, maximum eigen gap is minimized. The resultant Clusters are:



Model/Metric	**Silhoutte Score**	**Calinski-Harabasz Index**	**Davies- Bouldin Index**
Ground Truth	0.086	71.39	1.97
SC	0.470	61.78	1.57

As we can see the performance metrics values are very close to the ground truths.

If we set number of clusters=5, the plot is:



The other Performance metrics are:
Rand index= 0.717
Jaccard Coefficient= 0.011

For n_components=5,
Rand index= 0.676
Jaccard Coefficient= 0.002
This method performs not very well as the graph is not fully connected. It is not susceptible to outliers. But it might overfit to globular clusters.

The final Ranking Table is, as follows:

Rank	Cho.txt	Iyer.txt
1	K Means (Euclidian)	K-Means (Euclidean)
2	Gaussian Mixture Model	Heirarchal (Ward)
3	Heirarchal (Ward)	Gaussian Mixture Model
4	Spectral Clustering	Spectral Clustering
5	DBSCAN	DBSCAN
Outlier	No	Yes

Final Points:

- Distance Metric matters in K-Means. Here Eucledian did best.
- GMM and HAC perform pretty well if the data has some inherent hidden relationship.
- Density based models do not perform well if there are hidden relationship other than density.
- In HAC, Wards Algo significantly outperformed the simpler method.