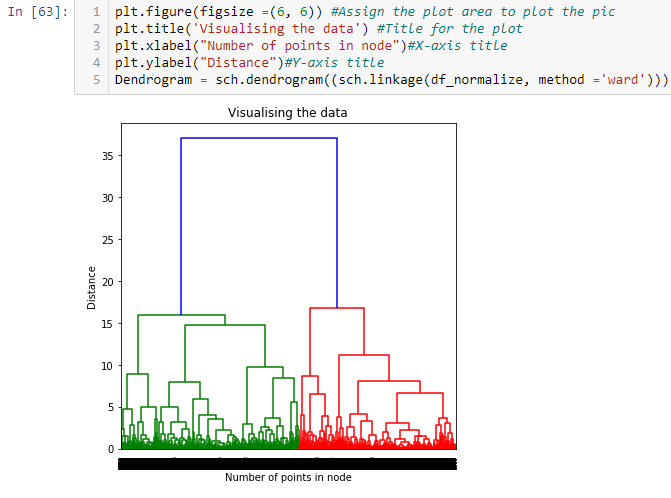
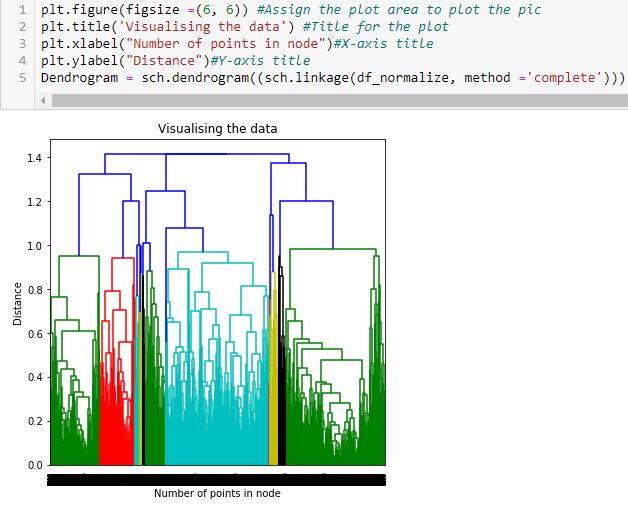
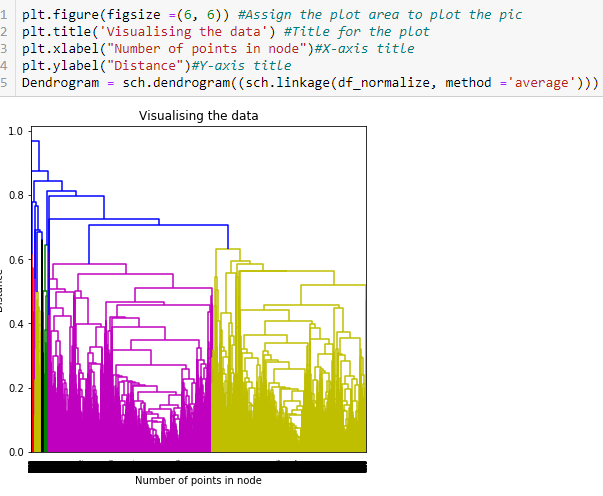
**Additional notes for Clustering**

**Closer look at dendrograms**

**Below snippets from submitted Python notebook**







The vertical axis of the dendrogram represents the distance or dissimilarity between clusters. The horizontal axis represents the objects and clusters. The objective to see dendograms is to get similarity between clusters.

**The 3 techniques used above & difference between them:**

|  |  |  |
| --- | --- | --- |
| **Ward** | **Average** | **Complete** |
| With this method, groups are formed so that the pooled within-group sum of squares is minimized | The average distance between each of the members, weighted so that the two groups have an equal influence on the final result | The distance between two groups as the distance between their two farthest-apart members |
| Requires the Distance Method to be Euclidean only. | Also known as Weighted Pair-Group | Also known as Furthest Neighbour |
| The two clusters are fused which result in the least increase in the pooled within-group sum of squares. | The two clusters have an equal influence on the final result. | This method usually yields clusters that are well separated and compact. |
| Clusters of similar sizes. Quite Cohesive inside. | Not very well understood on arbitory cut of dendogram. | Compact clusters in shape. |

Using dendrograms obtained above, choosing a suitable **k** for each linkage type. Lets experiment with different distance measures as mentioned below:

1. Euclidean
2. Manhattan
3. Cosine  
   Also calculating the cluster quality for each case.

**Agglomerative Clustering**

|  |  |  |  |
| --- | --- | --- | --- |
| **Cluster Count** | **Linkage** | **Distance Measures** | **Cluster Quality(Silhouette Score)** |
| 2 | Average | Cityblock | 0.322 |
| 3 | Average | Cityblock | 0.180 |
| 4 | Average | Cityblock | 0.157 |
| 5 | Average | Cityblock | 0.141 |
| 6 | Average | Cityblock | 0.149 |
| 7 | Average | Cityblock | 0.288 |

D

Based on Silhouette score, below are the Centroids for 17 features with Cluster K=2:

[[0.05047774 0.55055618 0.01070062 0.00771334 0.00935694 0.01390726

0.27128377 0.11130371 0.19517606 0.06138275 0.01740152 0.0211123

0.09231093 0.02011944 0.00701671 0.08586862 0.58425919]

[0.16529907 0.49386008 0.05115388 0.06096643 0.00104429 0.21719876

0.11963553 0.09339091 0.03936698 0.22101697 0.13722023 0.01846365

0.25338292 0.35699312 0.02023626 0.26591655 0.32984332]]

|  |  |  |  |
| --- | --- | --- | --- |
| **Cluster Count** | **Linkage** | **Distance Measures** | **Cluster Quality(Silhouette Score)** |
| 2 | Average | Euclidean | 0.367 |
| 3 | Average | Euclidean | 0.298 |
| 4 | Average | Euclidean | 0.264 |
| 5 | Average | Euclidean | 0.233 |
| 6 | Average | Euclidean | 0.190 |
| 7 | Average | Euclidean | 0.155 |

Based on Silhouette score, below are the Centroids for 17 features with Cluster K=2:

[[0.05054753 0.55058602 0.01072442 0.00774396 0.00935334 0.01402141

0.27122935 0.11130614 0.19511081 0.06142952 0.0174662 0.02111318

0.09236027 0.02030849 0.00702406 0.08597881 0.58409608]

[0. 0. 0. 0. 0. 0.00882712

0. 0. 0. 0.441042 0.03764996 0.

0.45613655 0.01260156 0.00737994 0. 0.77182427]]

|  |  |  |  |
| --- | --- | --- | --- |
| **Cluster Count** | **Linkage** | **Distance Measures** | **Cluster Quality(Silhouette Score)** |
| 2 | Average | Cosine | 0.415 |
| 3 | Average | Cosine | 0.274 |
| 4 | Average | Cosine | 0.231 |
| 5 | Average | Cosine | 0.228 |
| 6 | Average | Cosine | 0.204 |
| 7 | Average | Cosine | 0.167 |

Based on Silhouette score, below are the Centroids for 17 features with Cluster K=2:

[[5.06023762e-02 5.50492932e-01 1.07362103e-02 7.75178951e-03

9.36484915e-03 1.40136804e-02 2.71525747e-01 1.11405718e-01

1.95350936e-01 6.13492257e-02 1.74530289e-02 2.11384856e-02

9.24674567e-02 2.02926649e-02 7.03249987e-03 8.51015292e-02

5.84806776e-01]

[5.48611746e-03 5.74038920e-01 1.05025160e-03 1.26355023e-03

0.00000000e+00 1.93486571e-02 2.78609886e-02 2.78609886e-02

0.00000000e+00 1.52866887e-01 2.89583661e-02 5.08017104e-04

4.28404678e-02 3.14563665e-02 7.64196062e-04 7.32242444e-01

7.03911508e-02]]

**Best Cluster at K=2, with Linkage as Average, distance measure cosine, Silhouette score as 0.415**

**DBSCAN Information**

Lets analyse the results and see how the clustering changes as the above parameters are varied.

**When creating clusters without data normalization post Min-Max Scaler**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S.No.** | **Epsilon** | **Min-Points** | **No. of Clusters** | **Noise** | **Core points** | **Cluster Quality(Silhouette Score)** |
| 1. | 0.1 | 5 | 62 | 5032 | 3310 | -0.498 |
| 2. | 0.3 | 3 | 3 | 285 | 8568 | -0.146 |
| 3. | 0.8 | 5 | 1 | 6 | 8938 | NA |

**When creating clusters with data normalization post Min-Max Scaler**

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **S.No.** | **Epsilon** | **Min-Points** | **No. of Clusters** | **Noise** | **Core points** | **Cluster Quality(Silhouette Score)** |
| 1. | 0.1 | 5 | 22 | 1839 | 6415 | -0.392 |
| 2. | 0.3 | 3 | 1 | 25 | 8910 | NA |
| 3. | 0.8 | 5 | 1 | 0 | 8950 | NA |

**Note : NA:** Not Available

**Inference:**

1. All clusters are built around core points hence it’s the core part.
2. By adjusting our Min-points and epsilon (radius of our neighbourhoods), we get different number of clusters.
3. As the Epsilon & Min-points increases, number of noise points decreases and core points increases.
4. Noise is highest when epsilon and Min-points are lowest.
5. Cluster gets better Silhouette score when dataset is normalized.
6. Cluster gets lower noise points & higher core points when dataset is normalized.
7. If the number of clusters is 1, we cannot calculate Silhouette score.
8. Based on Silhouette score, none of the clusters are satisfactory with DBSCAN.