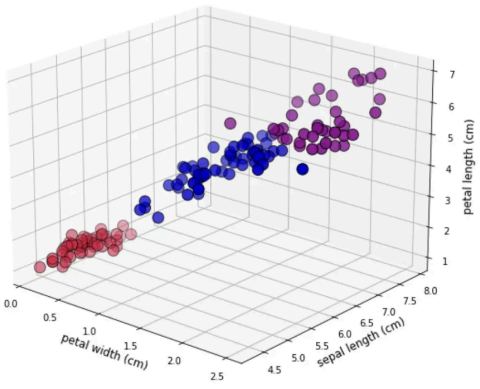
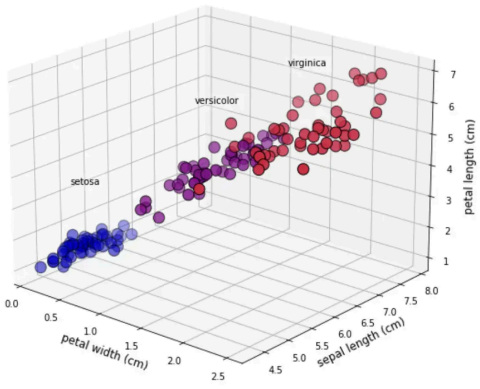
Project 2: Data Representations and Clustering

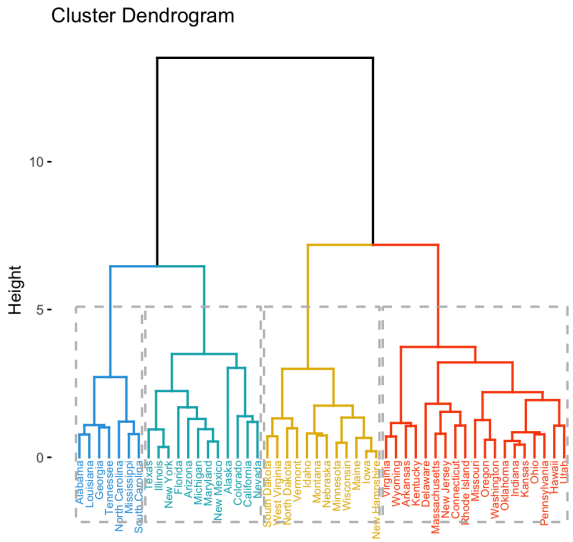
Authors: Alex Chen, Jiamin Xu

Electrical and Computer Engineering 219: Winter 2024

Professor Vwani Roychowdhury



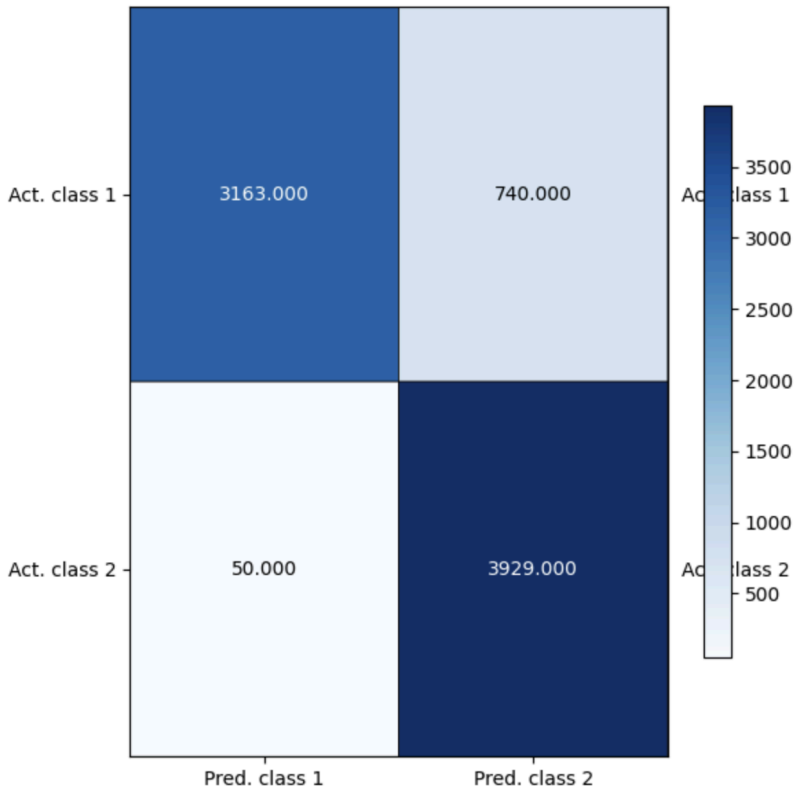




Question 1

The TF-IDF matrix has dimensions of (7882, 23522)

Question 2

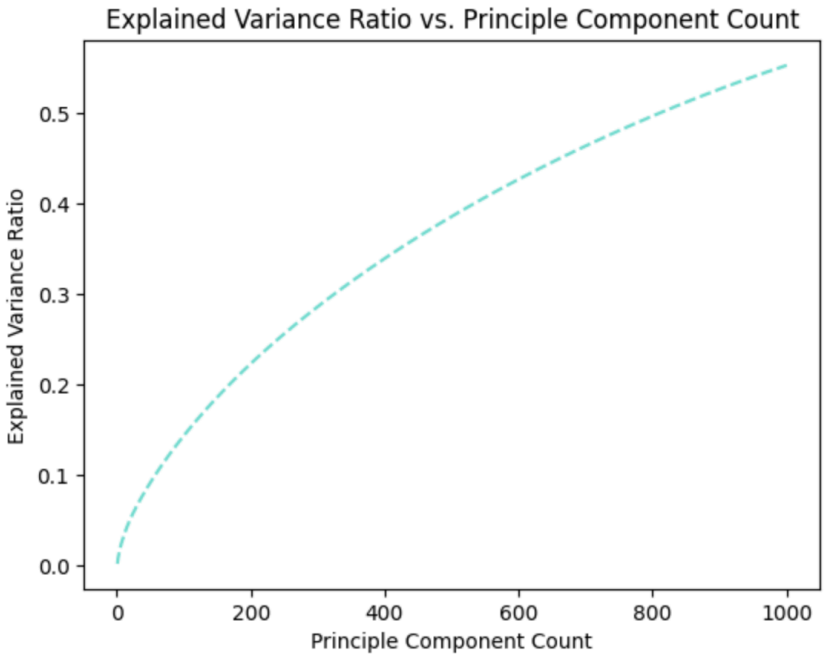


The contingency matrix does not need to be square. Since A_{ij} represents the number of datapoints of class i predicted to be in class / cluster j, if the training results in finding more clusters than true classes, the matrix will not be square.

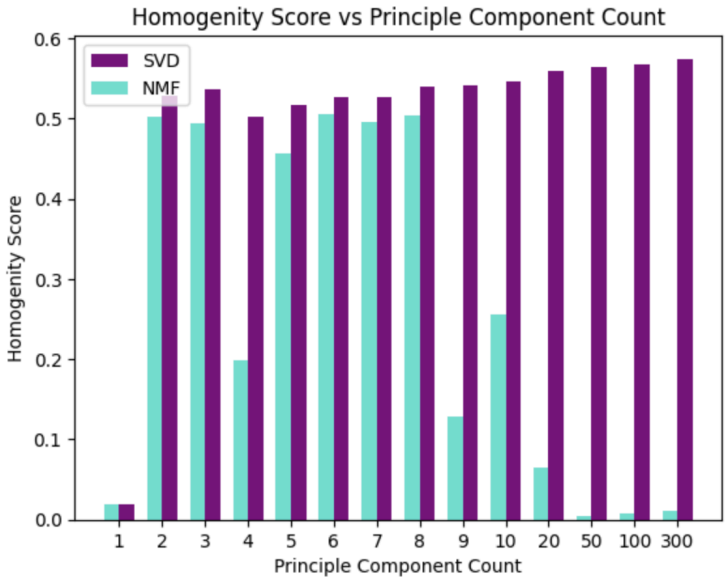
Question 3

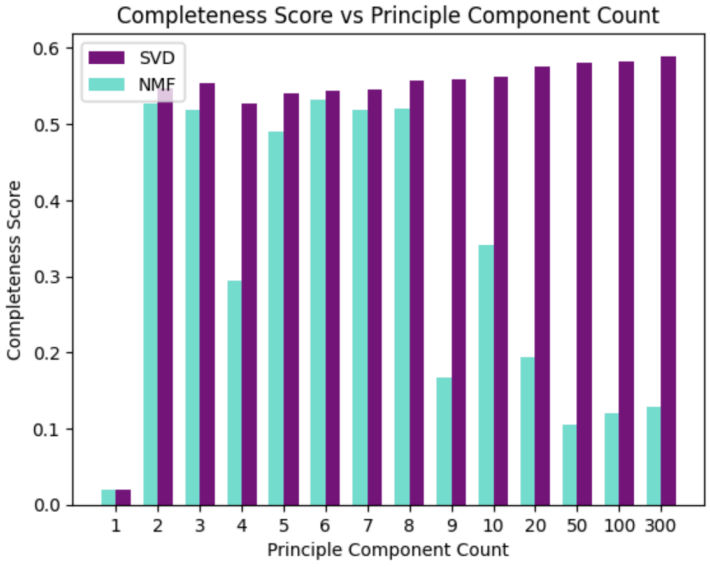
Homogeneity Score: 0.5791959247349042
Completeness Score: 0.5938590275126396
V-measure Score: 0.5864358322571914
Adjusted Rand Index Score: 0.6392240733134459
Adjusted Mutual Information score: 0.586397496683508

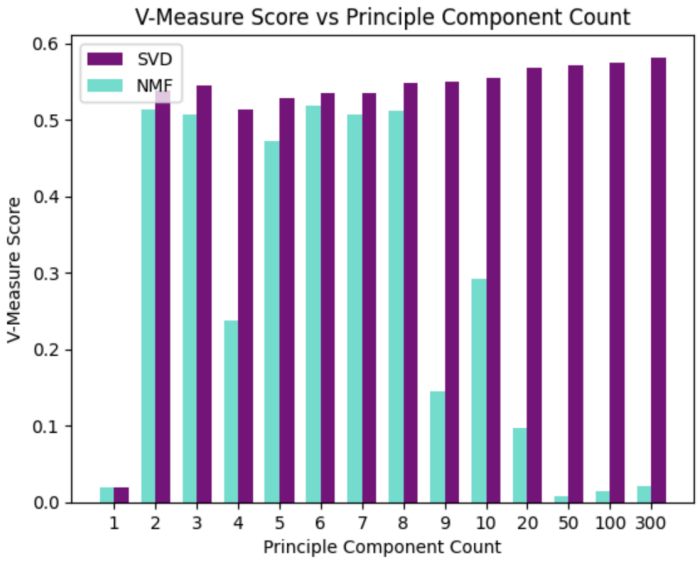
Question 4

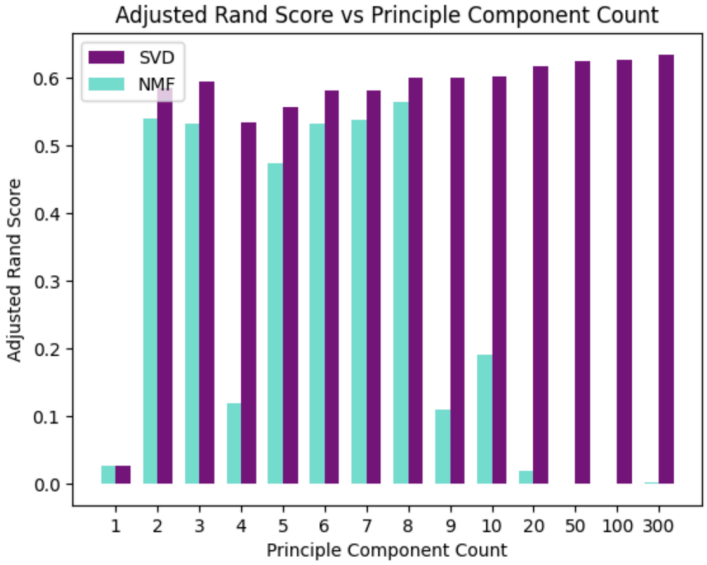


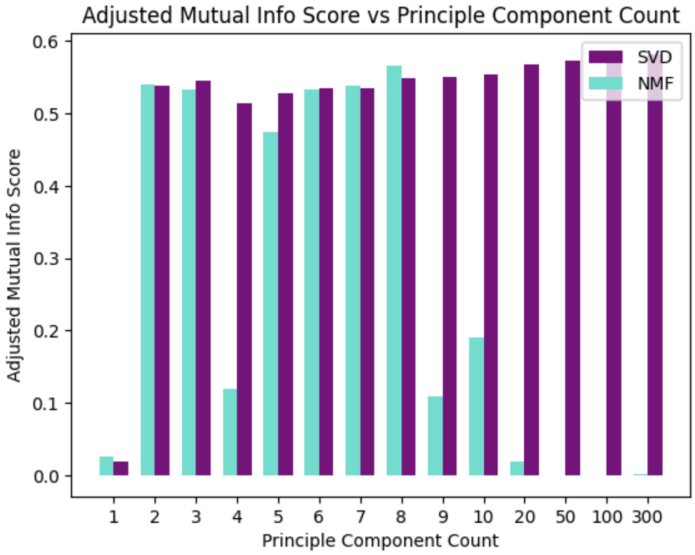
Question 5

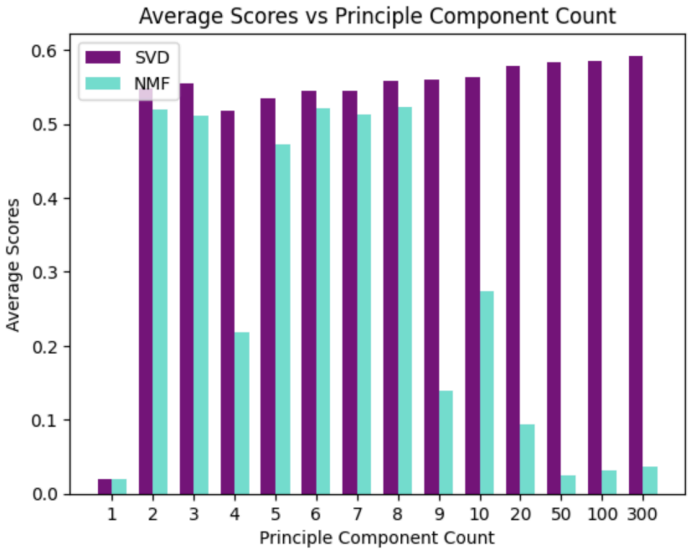





V-Measure Score vs Principle Component Count
SVD
NMF
V-Measure Score
0.0
0.1
0.2
0.3
0.4
0.5
0.6
1 2 3 4 5 6 7 8 9 10 20 50 100 300
Principle Component Count


Adjusted Rand Score vs Principle Component Count
SVD
NMF
Adjusted Rand Score
0.0
0.1
0.2
0.3
0.4
0.5
0.6
1 2 3 4 5 6 7 8 9 10 20 50 100 300
Principle Component Count


Adjusted Mutual Info Score vs Principle Component Count
SVD
NMF
Adjusted Mutual Info Score
0.0
0.1
0.2
0.3
0.4
0.5
0.6
1 2 3 4 5 6 7 8 9 10 20 50 100 300
Principle Component Count


Average Scores vs Principle Component Count
SVD
NMF
Average Scores
0.0
0.1
0.2
0.3
0.4
0.5
0.6
1 2 3 4 5 6 7 8 9 10 20 50 100 300
Principle Component Count

r for SVD: 7
r for NMF: 2

For SVD, each of the score measures has close to non-monotonic behavior as *r* increases. However, the best principle component count should not be the largest value since clustering behavior degrades and is less accurate as dimensionality increases. This is further explained in question 6. Thus, instead of choosing the dimensionality with the largest average score, which would be 300, we choose a lower dimensionality which still has a good score. This is 7.

For NMF, by visually inspecting the average scores histogram as well as using the underlying average scores, using 2 principle components yields the best score for NMF.

Question 6

As *r* increases, the measure scores for NMF increase for a while before crashing. This can be explained because as *r* increases, there is more information about the initial data representation retained. However, at a certain point, the k-means algorithm, aka Euclidian distance, performs more and more poorly. This is because as dimensionality increases, points become less and less separated from each other. As is a similar idea touched upon in one of the references, points become roughly equidistant as dimensionality increases. Furthermore, all dimensions are treated equally with Euclidian distance / k-means, and as we use more and more principle components, we are taking into account dimensions that should not be factored in. Thus, these points explain why the NMF clustering scores begin to decrease after seeing non-monotonic behavior for a certain few component counts.

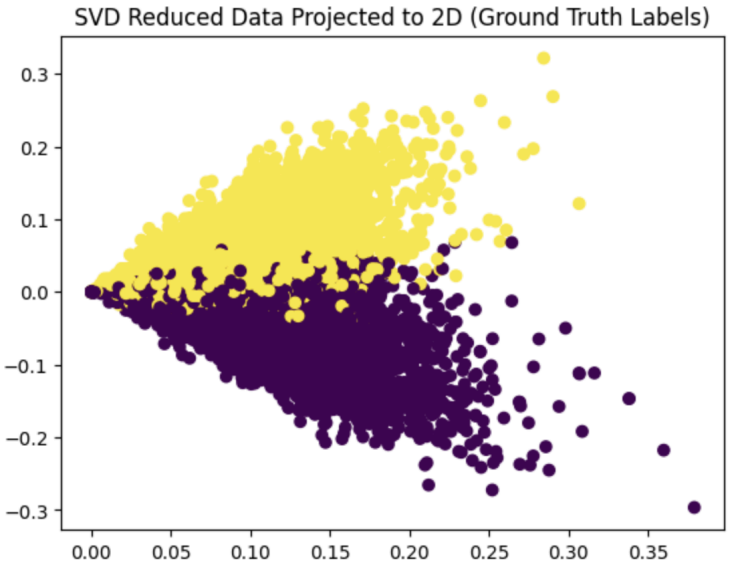
For SVD, the scores are more holistically non-monotonic. There is no drop-off point where the scores crash. This may be explained by how SVD has less restraints on the dimensionality reduction it performs. SVD allows both positive and negative values in the reduced matrix, allowing it to represent the higher dimension matrix better than NMF. This can also be explained by how SVD does a linear transformation, allowing it to conform better to the assumption of K-means that clusters are spherical, making it sensitive to linear structure of data. NMF is non-linear, which may make it align less well to K-means assumptions.

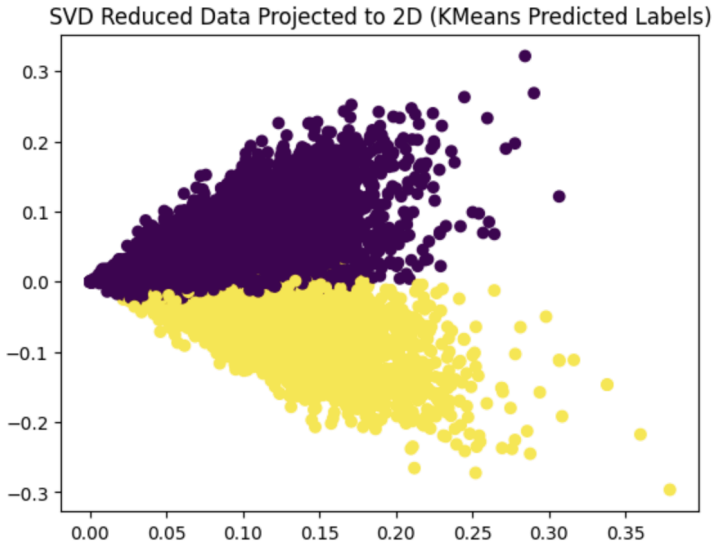
Question 7

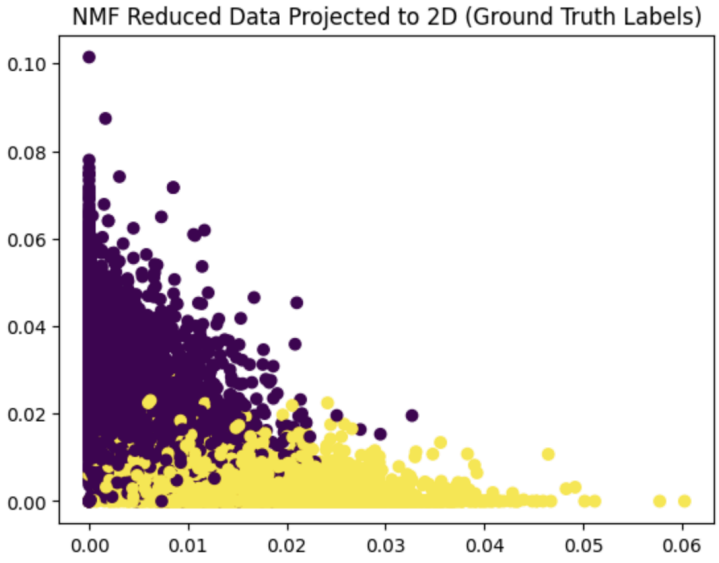
Surprisingly, both the highest average scores for each of SVD and NMF are not higher than the average score of all measures using non-reduced TF-IDF counts. However, the highest average SVD score is not significantly different from the previous average. This may be explained by SVD's ability to represent higher dimensionality data, as explained in the previous section. This may cause SVD to better represent the higher dimensionality data, and ultimately result in similar clustering. The average NMF score is noticeably lower, however. Initially, this doesn't make much sense because NMF should reduce the data to the important dimensions, after which clustering should be more effective. However, with the same reasoning as why NMF

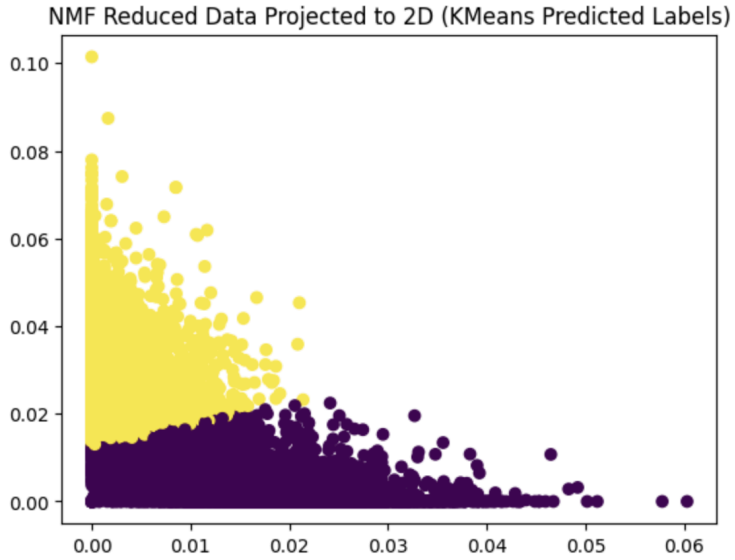
scores began to decrease once principle component count increased, while SVD did not, NMF does not represent the higher dimensionality data as well, while SVD does. Thus, it is possible that lower average clustering metric scores may be because the truly important components to use to cluster have been eliminated. In a similar vein, it could be true that clustering using TF-IDF scores is not a good way to classify documents.

Question 8









Question 9

From these results, it seems like the dimensionality reduction did a decent job of performing classification. When looking at the projections of PC2 vs PC1 for SVD and NMF, for the KMeans predicted labels, the projected principle components cause the two groups to be separated well

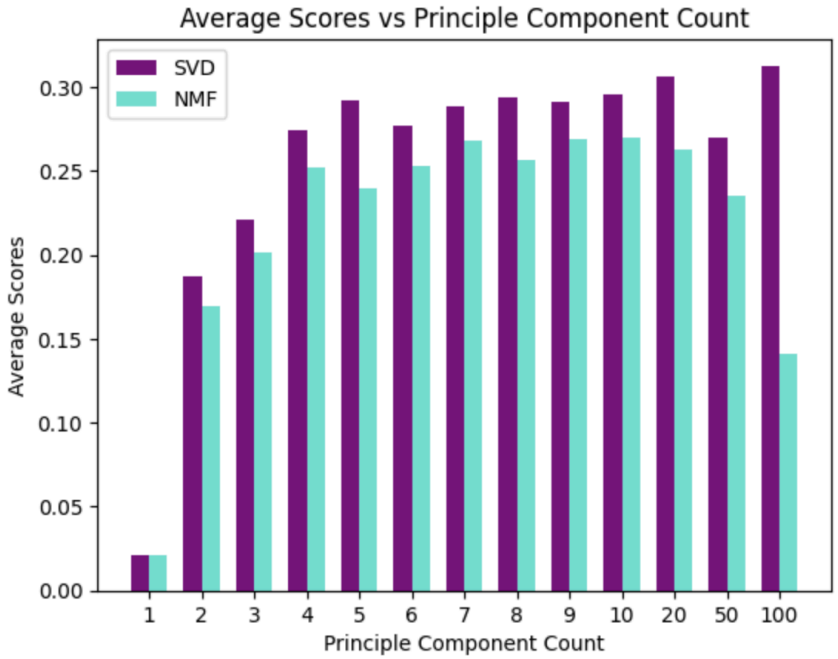
and very similarly to how they are in the ground truth labeled visualizations. Note that the color swap should be insignificant, since KMeans is performing unsupervised classification and only determining which points belong to the same class. The grouping of all colors / labels together just as they are in the ground truth visualization is the important point.

The distribution of data between different classes does overlap a bit, making it not perfectly ideal for KMeans, since the KMeans algorithm would prefer more separation between the two classes. However, for the majority of the datapoints, there is a clear separation in which class a datapoint should belong to.

Furthermore, the data is not spherical, which as previously mentioned, is an assumption of KMeans. With SVD, the data is heart-shaped, while with NMF, the data follows a curve.

Question 10

I calculated the average of the five metrics for each of many principle component counts from 1 through 100, and used these averages to determine the best value for SVD and NMF. I also took into account that the smaller the dimension, the better clustering performs. Thus, even though average scores were similar between lower and higher dimensions, I prioritized smaller dimensionality.



r for SVD: 5
r for NMF: 7

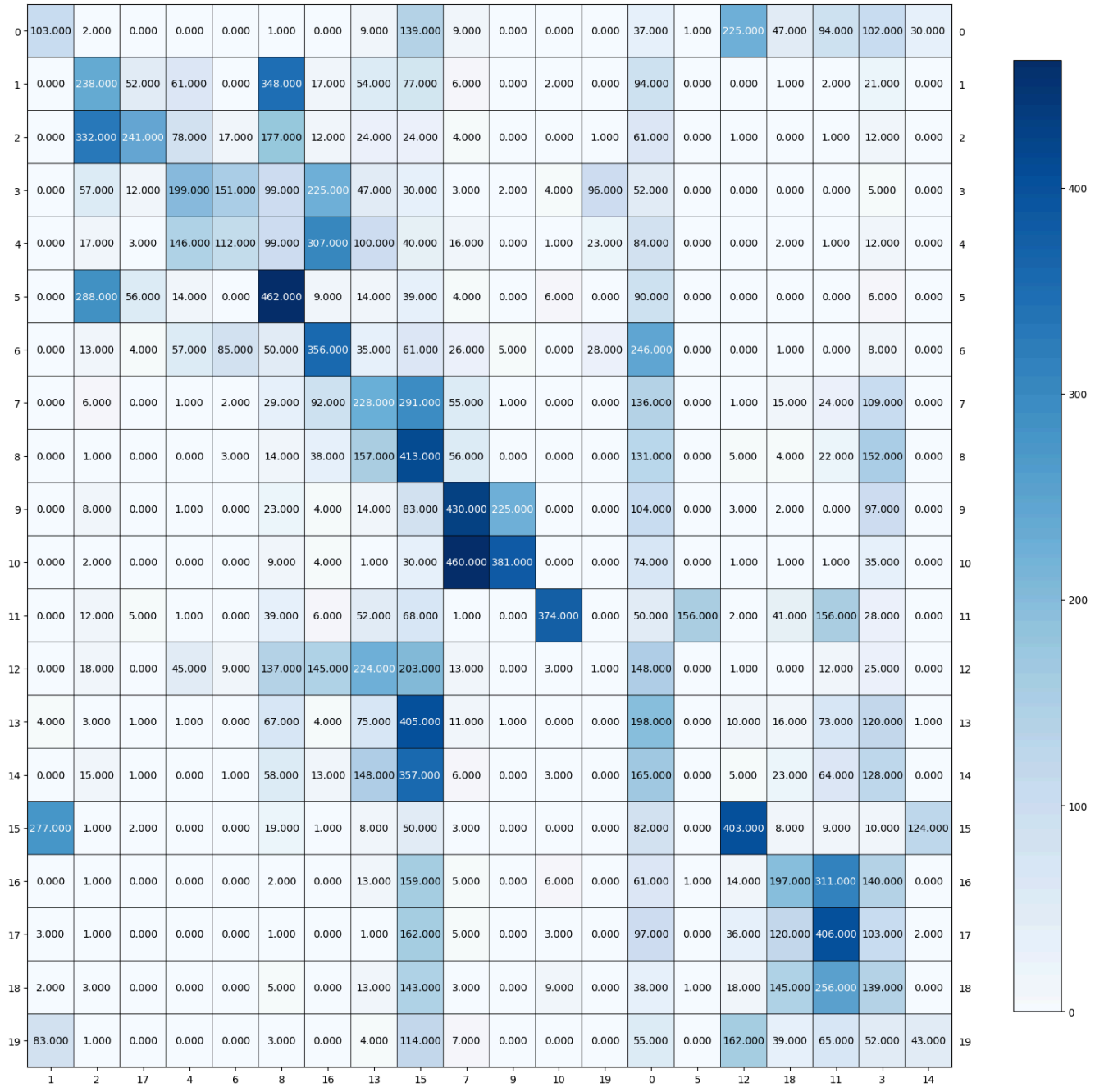
The following are the five measure scores as well as contingency matrices when performing dimensionality reduction using the selected principle component counts.

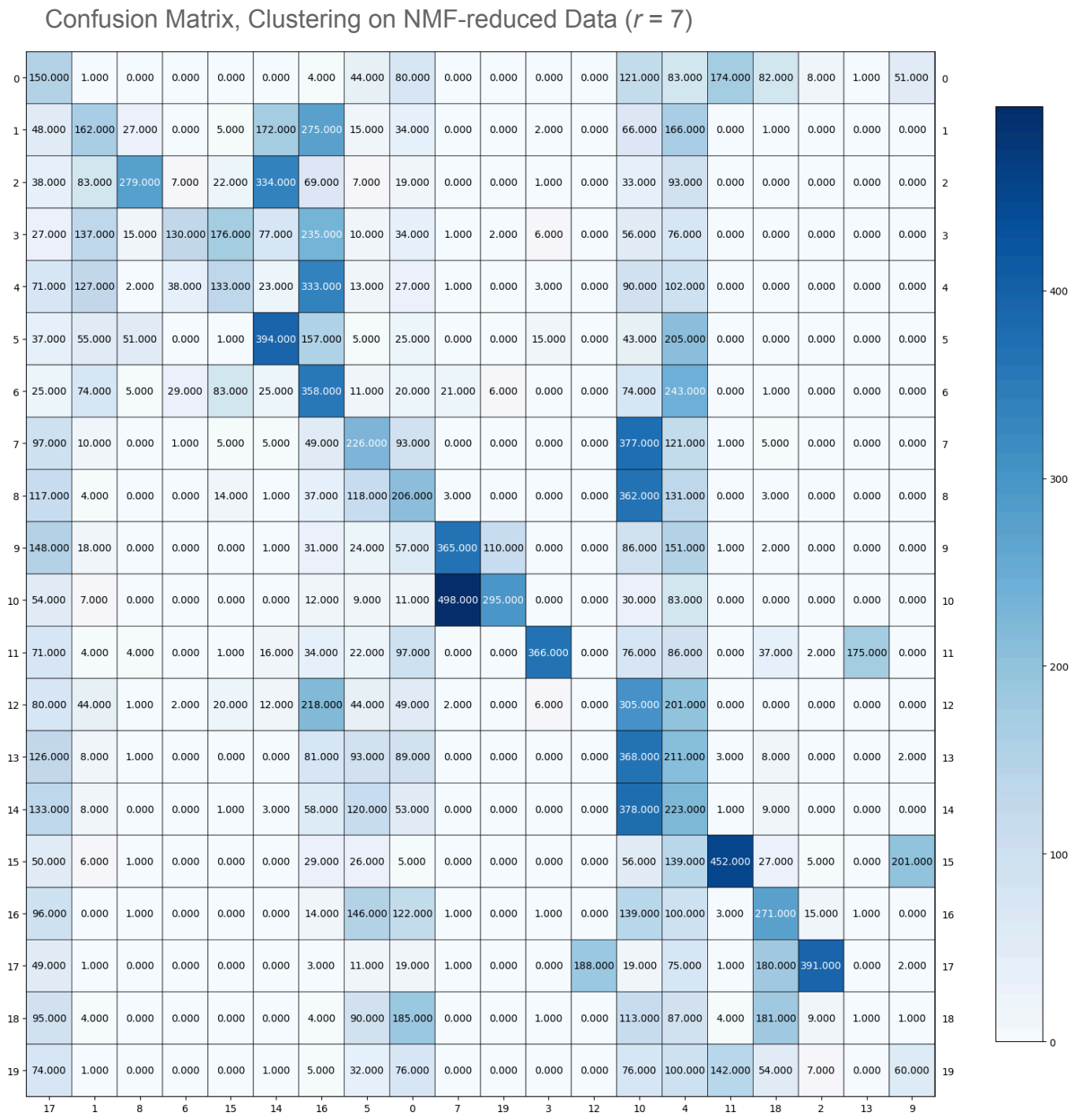
Homogeneity Score (SVD): 0.32016271714698474
Completeness Score (SVD): 0.3482890305417125
V-measure Score (SVD): 0.3336341411516645
Adjusted Rand Index Score (SVD): 0.1259875510824014
Adjusted Mutual Information score (SVD): 0.33138414757577583

Homogeneity Score (NMF): 0.29544735896178187
Completeness Score (NMF): 0.3274814544666058
V-measure Score (NMF): 0.3106407305150111
Adjusted Rand Index Score (NMF): 0.10549913689715151
Adjusted Mutual Information score (NMF): 0.30829241618235176

From these scores, it can be seen that clustering with 20 categories has noticeably poorer performance than binary clustering.

Confusion Matrix, Clustering on SVD-reduced Data ($r = 5$)




Confusion Matrix, Clustering on NMF-reduced Data (r = 7)
0 150.000 1.000 0.000 0.000 0.000 0.000 4.000 44.000 80.000 0.000 0.000 0.000 0.000 121.000 83.000 174.000 82.000 8.000 1.000 51.000 0
1 48.000 162.000 27.000 0.000 5.000 172.000 275.000 15.000 34.000 0.000 0.000 2.000 0.000 66.000 166.000 0.000 1.000 0.000 0.000 0.000 1
2 38.000 83.000 279.000 7.000 22.000 334.000 69.000 7.000 19.000 0.000 0.000 1.000 0.000 33.000 93.000 0.000 0.000 0.000 0.000 0.000 2
3 27.000 137.000 15.000 130.000 176.000 77.000 235.000 10.000 34.000 1.000 2.000 6.000 0.000 56.000 76.000 0.000 0.000 0.000 0.000 0.000 3
4 71.000 127.000 2.000 38.000 133.000 23.000 333.000 13.000 27.000 1.000 0.000 3.000 0.000 90.000 102.000 0.000 0.000 0.000 0.000 0.000 4
5 37.000 55.000 51.000 0.000 1.000 394.000 157.000 5.000 25.000 0.000 0.000 15.000 0.000 43.000 205.000 0.000 0.000 0.000 0.000 0.000 5
6 25.000 74.000 5.000 29.000 83.000 25.000 358.000 11.000 20.000 21.000 6.000 0.000 0.000 74.000 243.000 0.000 1.000 0.000 0.000 0.000 6
7 97.000 10.000 0.000 1.000 5.000 5.000 49.000 226.000 93.000 0.000 0.000 0.000 0.000 377.000 121.000 1.000 5.000 0.000 0.000 0.000 7
8 117.000 4.000 0.000 0.000 14.000 1.000 37.000 118.000 206.000 3.000 0.000 0.000 0.000 362.000 131.000 0.000 3.000 0.000 0.000 0.000 8
9 148.000 18.000 0.000 0.000 0.000 1.000 31.000 24.000 57.000 365.000 110.000 0.000 0.000 86.000 151.000 1.000 2.000 0.000 0.000 0.000 9
10 54.000 7.000 0.000 0.000 0.000 0.000 12.000 9.000 11.000 498.000 295.000 0.000 0.000 30.000 83.000 0.000 0.000 0.000 0.000 0.000 10
11 71.000 4.000 4.000 0.000 1.000 16.000 34.000 22.000 97.000 0.000 0.000 366.000 0.000 76.000 86.000 0.000 37.000 2.000 175.000 0.000 11
12 80.000 44.000 1.000 2.000 20.000 12.000 218.000 44.000 49.000 2.000 0.000 6.000 0.000 305.000 201.000 0.000 0.000 0.000 0.000 0.000 12
13 126.000 8.000 1.000 0.000 0.000 0.000 81.000 93.000 89.000 0.000 0.000 0.000 0.000 368.000 211.000 3.000 8.000 0.000 0.000 2.000 13
14 133.000 8.000 0.000 0.000 1.000 3.000 58.000 120.000 53.000 0.000 0.000 0.000 0.000 378.000 223.000 1.000 9.000 0.000 0.000 0.000 14
15 50.000 6.000 1.000 0.000 0.000 0.000 29.000 26.000 5.000 0.000 0.000 0.000 0.000 56.000 139.000 452.000 27.000 5.000 0.000 201.000 15
16 96.000 0.000 1.000 0.000 0.000 0.000 14.000 146.000 122.000 1.000 0.000 1.000 0.000 139.000 100.000 3.000 271.000 15.000 1.000 0.000 16
17 49.000 1.000 0.000 0.000 0.000 0.000 3.000 11.000 19.000 1.000 0.000 0.000 188.000 19.000 75.000 1.000 180.000 391.000 0.000 2.000 17
18 95.000 4.000 0.000 0.000 0.000 0.000 4.000 90.000 185.000 0.000 0.000 1.000 0.000 113.000 87.000 4.000 181.000 9.000 1.000 1.000 18
19 74.000 1.000 0.000 0.000 0.000 1.000 5.000 32.000 76.000 0.000 0.000 0.000 0.000 76.000 100.000 142.000 54.000 7.000 0.000 60.000 19
17 1 8 6 15 14 16 5 0 7 19 3 12 10 4 11 18 2 13 9
400
300
200
100
0

Question 11

Clustering Measures

Homogeneity Score (Cosine, 5 components): 0.5757393786334769
Completeness Score (Cosine, 5 components): 0.596170314940667
V-Measure Score (Cosine, 5 components): 0.5857767515120362
Adjusted Rand Score (Cosine, 5 components): 0.460359263242143
Adjusted Mutual Info Score (Cosine, 5 components): 0.584404126405671
Average Score (Cosine, 5 components): 0.5604899669467989

Homogeneity Score (Euclidean, 5 components): 0.006478555842729128
Completeness Score (Euclidean, 5 components): 0.006517670097977768
V-Measure Score (Euclidean, 5 components): 0.006498054109997733
Adjusted Rand Score (Euclidean, 5 components): 0.001264960852663542
Adjusted Mutual Info Score (Euclidean, 5 components): 0.003282579427816658
Average Score (Euclidean, 5 components): 0.004808364066236966

Homogeneity Score (Cosine, 20 components): 0.5731621212431922
Completeness Score (Cosine, 20 components): 0.5953023492883153
V-Measure Score (Cosine, 20 components): 0.5840224771985414
Adjusted Rand Score (Cosine, 20 components): 0.45679067578106736
Adjusted Mutual Info Score (Cosine, 20 components): 0.5826418212257898
Average Score (Cosine, 20 components): 0.5583838889473812

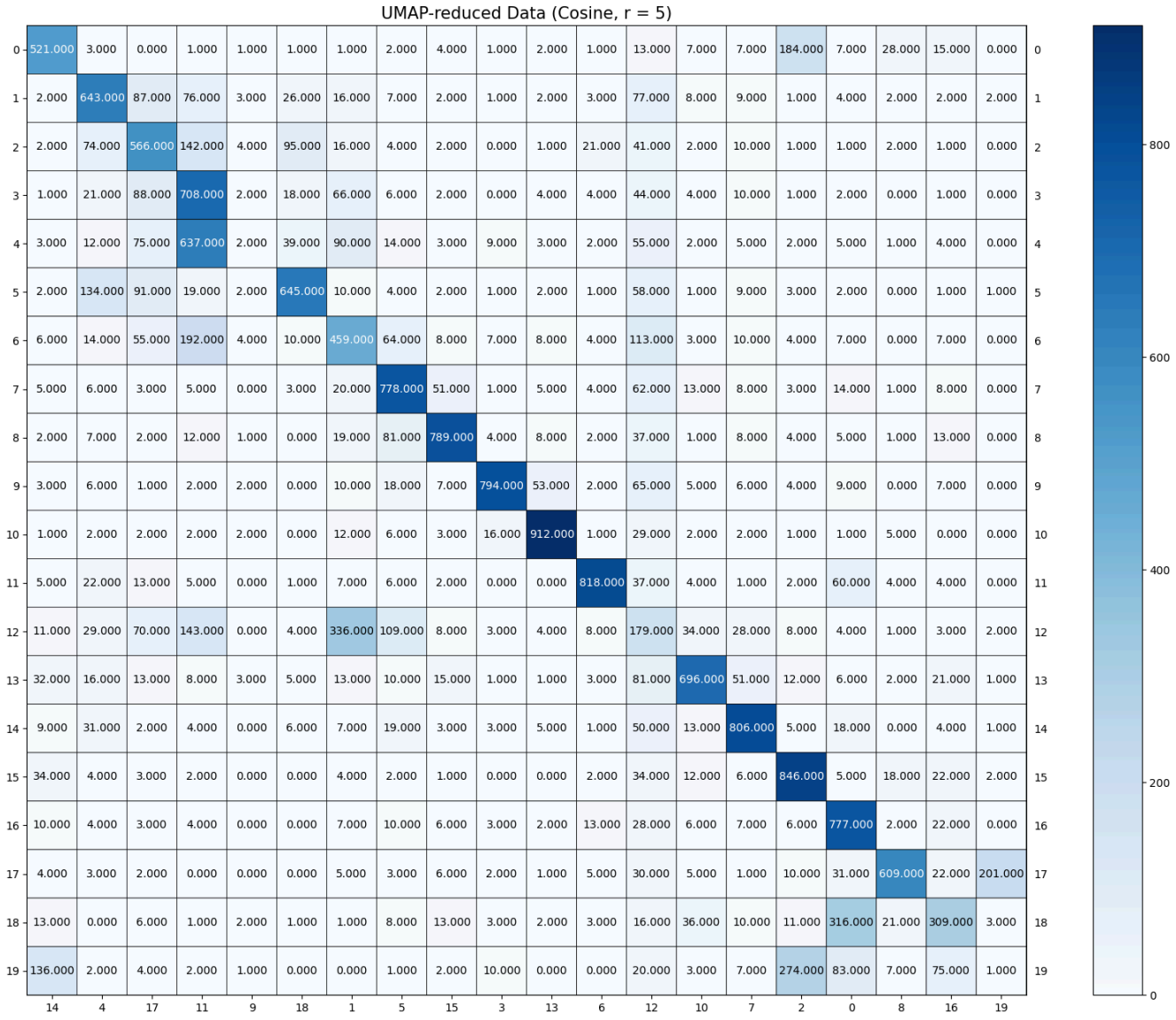
Homogeneity Score (Euclidean, 20 components): 0.007088539555509815
Completeness Score (Euclidean, 20 components): 0.007259026188717289
V-Measure Score (Euclidean, 20 components): 0.0071727699584032915
Adjusted Rand Score (Euclidean, 20 components): 0.001243368997671217
Adjusted Mutual Info Score (Euclidean, 20 components): 0.003902814334708501
Average Score (Euclidean, 20 components): 0.005333303807002023

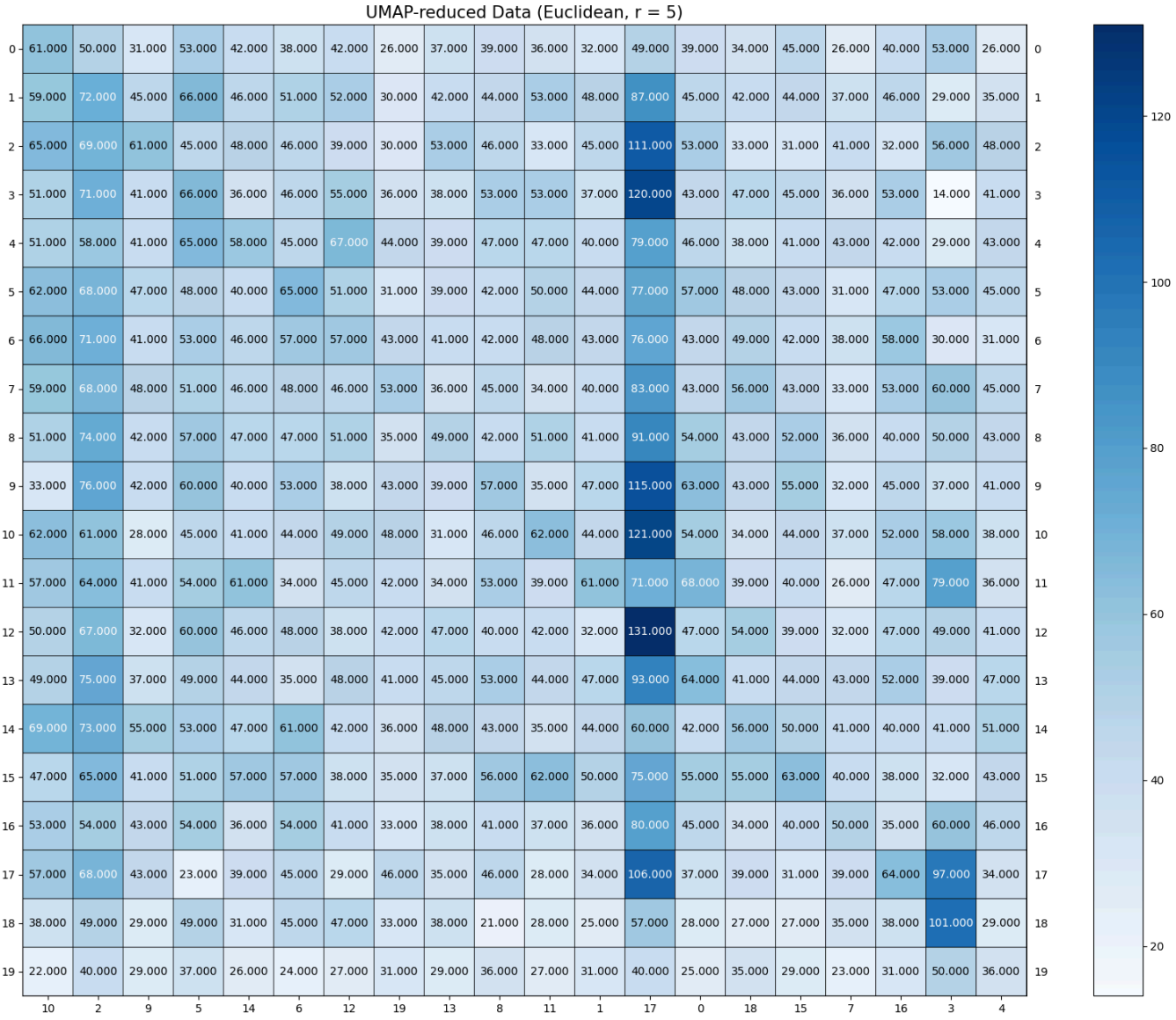
Homogeneity Score (Cosine, 200 components): 0.5726974977283901
Completeness Score (Cosine, 200 components): 0.5943566154588147
V-Measure Score (Cosine, 200 components): 0.5833260730335602
Adjusted Rand Score (Cosine, 200 components): 0.4557576821741938
Adjusted Mutual Info Score (Cosine, 200 components): 0.5819436671767985
Average Score (Cosine, 200 components): 0.5576163071143514

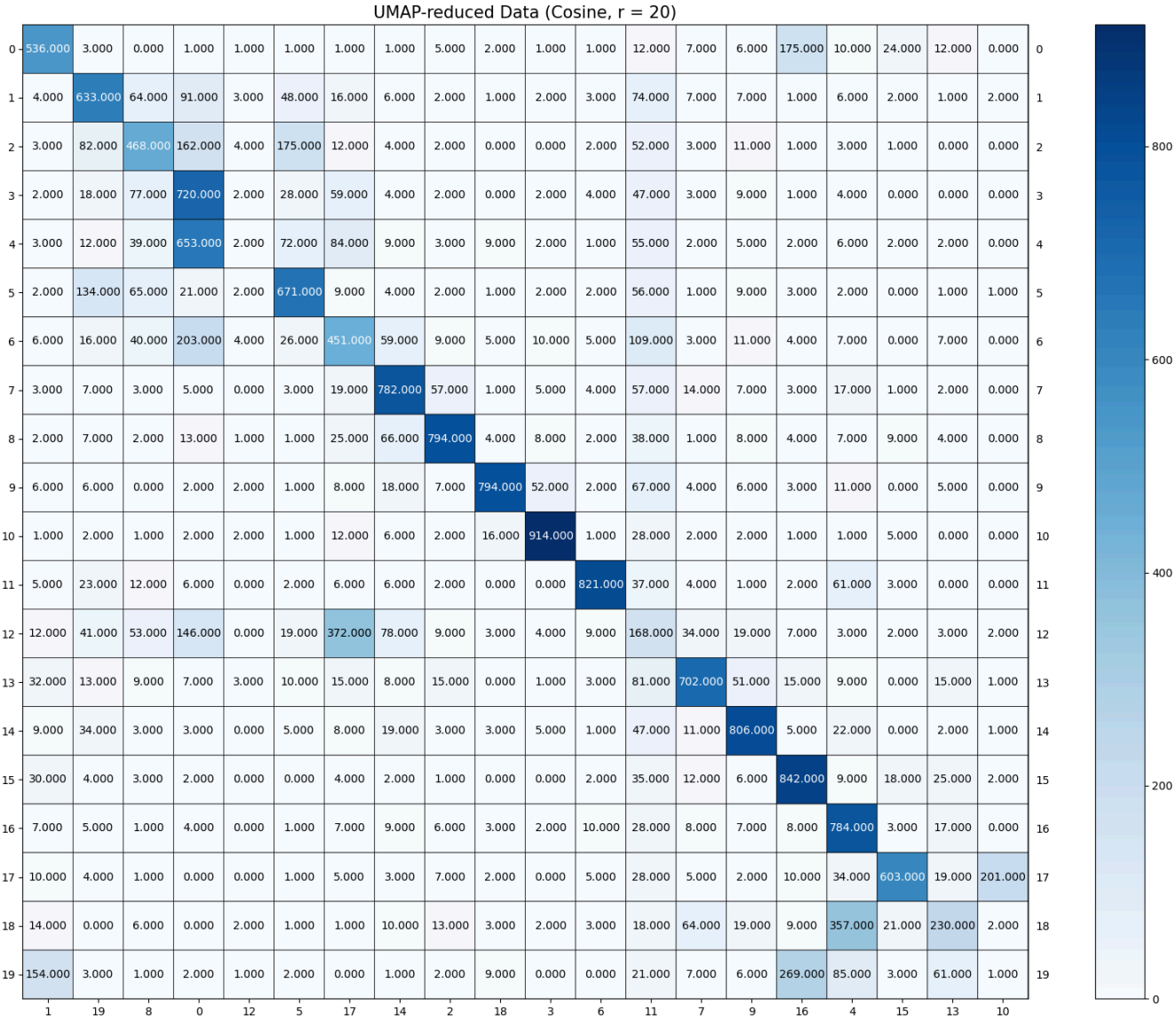
Homogeneity Score (Euclidean, 200 components): 0.0064698185786692865
Completeness Score (Euclidean, 200 components): 0.006750872490085557
V-Measure Score (Euclidean, 200 components): 0.006607358122421738
Adjusted Rand Score (Euclidean, 200 components): 0.0010911159364331501
Adjusted Mutual Info Score (Euclidean, 200 components): 0.0033377077711450333

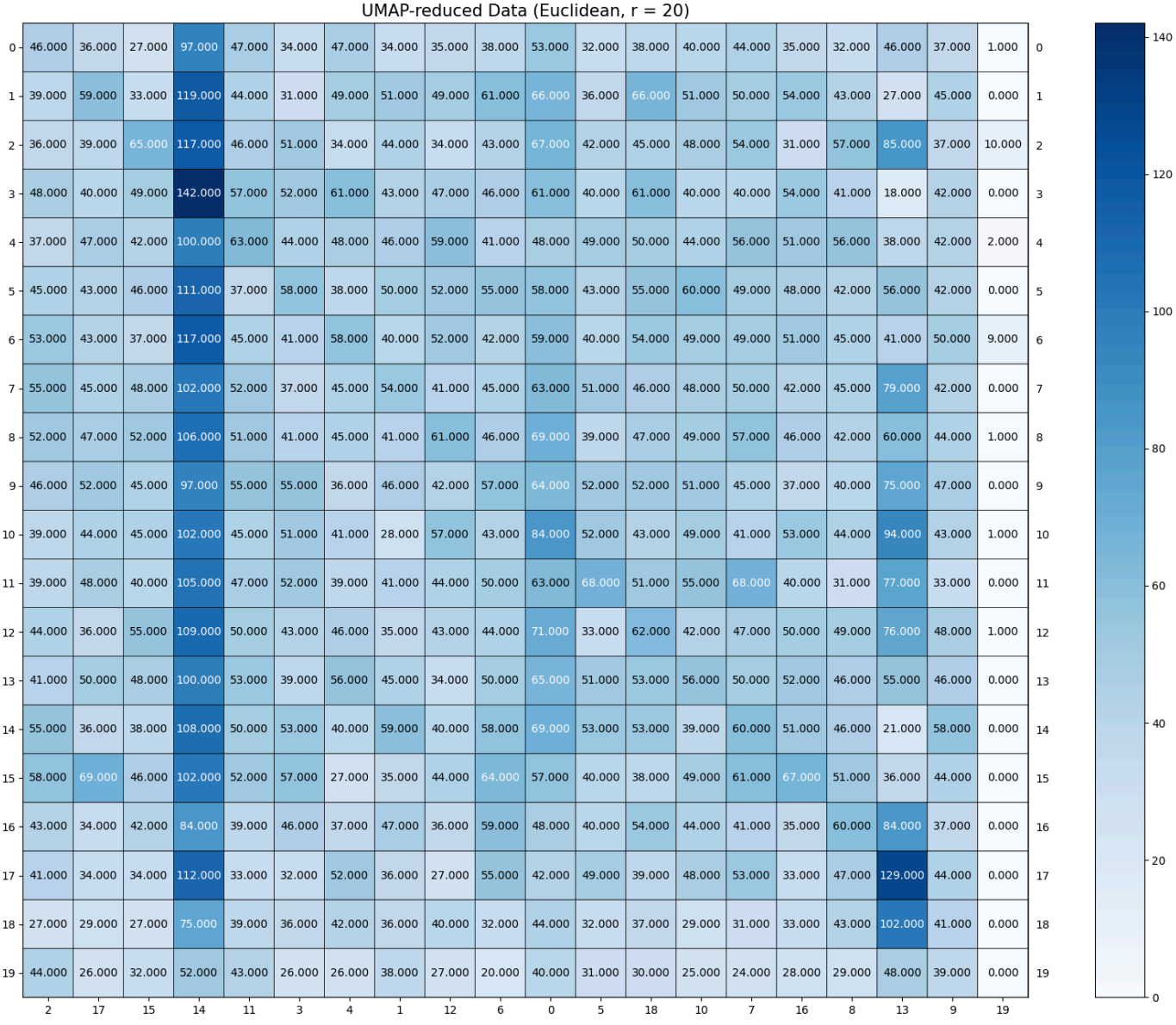
Average Score (Euclidean, 200 components): 0.004851374579750952

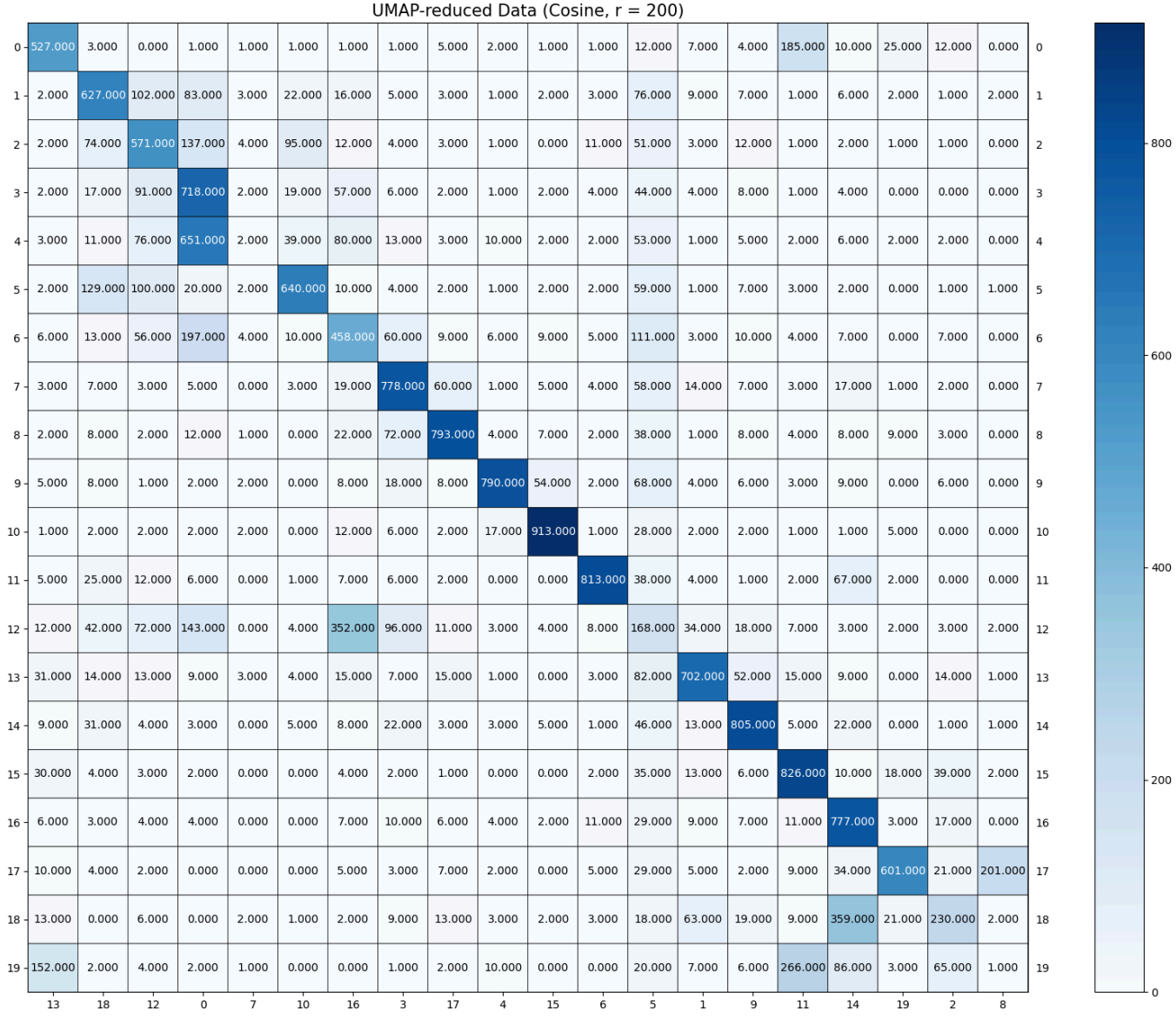
Contingency Matrices

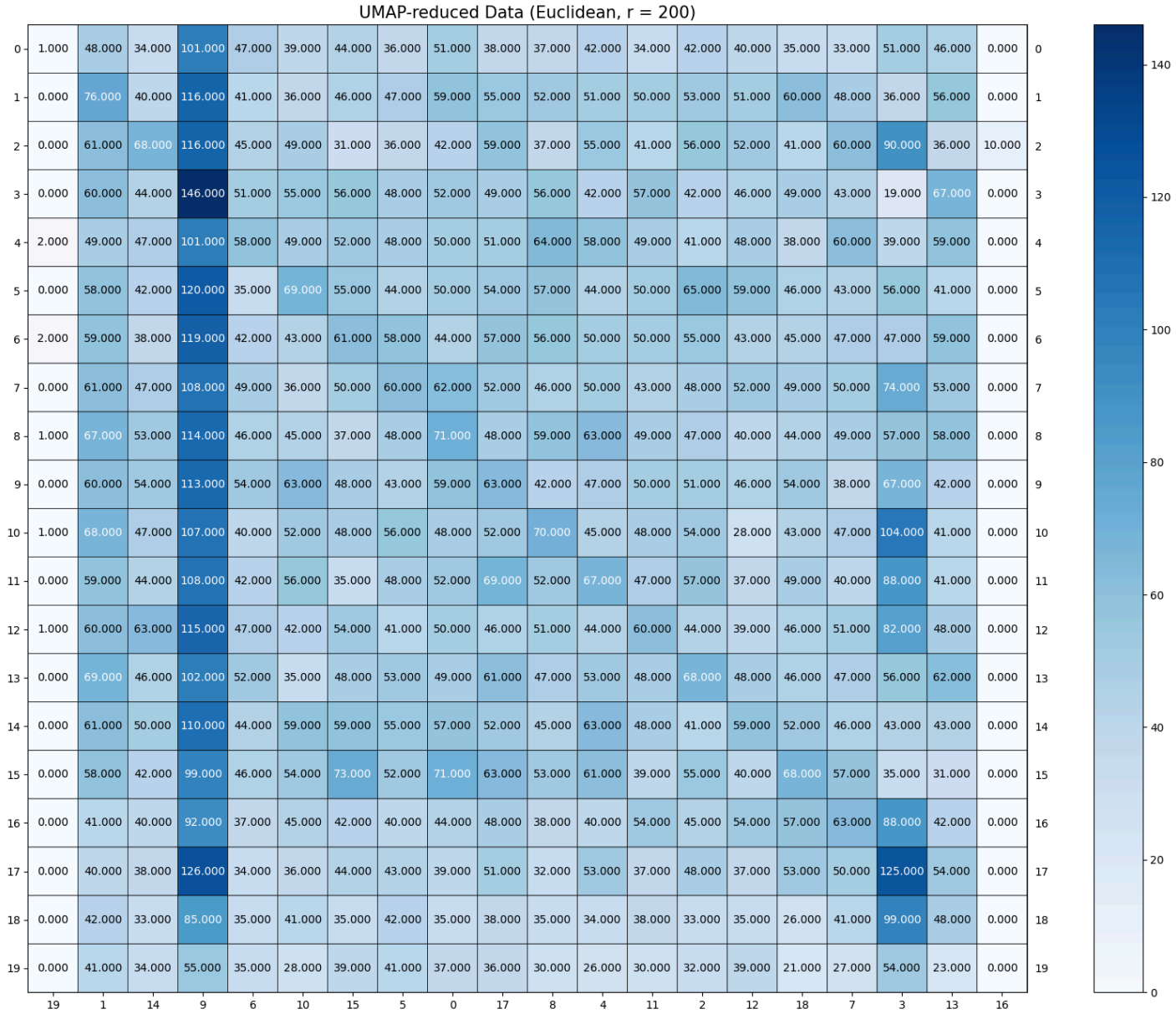



UMAP-reduced Data (Euclidean, r = 5)
10 2 9 5 14 6 12 19 13 8 11 1 17 0 18 15 7 16 3 4
0 61.000 50.000 31.000 53.000 42.000 38.000 42.000 26.000 37.000 39.000 36.000 32.000 49.000 39.000 34.000 45.000 26.000 40.000 53.000 26.000
1 59.000 72.000 45.000 66.000 46.000 51.000 52.000 30.000 42.000 44.000 53.000 48.000 87.000 45.000 42.000 44.000 37.000 46.000 29.000 35.000
2 65.000 69.000 61.000 45.000 48.000 46.000 39.000 30.000 53.000 46.000 33.000 45.000 111.000 53.000 33.000 31.000 41.000 32.000 56.000 48.000
3 51.000 71.000 41.000 66.000 36.000 46.000 55.000 36.000 38.000 53.000 53.000 37.000 120.000 43.000 47.000 45.000 36.000 53.000 14.000 41.000
4 51.000 58.000 41.000 65.000 58.000 45.000 67.000 44.000 39.000 47.000 47.000 40.000 79.000 46.000 38.000 41.000 43.000 42.000 29.000 43.000
5 62.000 68.000 47.000 48.000 40.000 65.000 51.000 31.000 39.000 42.000 50.000 44.000 77.000 57.000 48.000 43.000 31.000 47.000 53.000 45.000
6 66.000 71.000 41.000 53.000 46.000 57.000 57.000 43.000 41.000 42.000 48.000 43.000 76.000 43.000 49.000 42.000 38.000 58.000 30.000 31.000
7 59.000 68.000 48.000 51.000 46.000 48.000 46.000 53.000 36.000 45.000 34.000 40.000 83.000 43.000 56.000 43.000 33.000 53.000 60.000 45.000
8 51.000 74.000 42.000 57.000 47.000 47.000 51.000 35.000 49.000 42.000 51.000 41.000 91.000 54.000 43.000 52.000 36.000 40.000 50.000 43.000
9 33.000 76.000 42.000 60.000 40.000 53.000 38.000 43.000 39.000 57.000 35.000 47.000 115.000 63.000 43.000 55.000 32.000 45.000 37.000 41.000
10 62.000 61.000 28.000 45.000 41.000 44.000 49.000 48.000 31.000 46.000 62.000 44.000 121.000 54.000 34.000 44.000 37.000 52.000 58.000 38.000
11 57.000 64.000 41.000 54.000 61.000 34.000 45.000 42.000 34.000 53.000 39.000 61.000 71.000 68.000 39.000 40.000 26.000 47.000 79.000 36.000
12 50.000 67.000 32.000 60.000 46.000 48.000 38.000 42.000 47.000 40.000 42.000 32.000 131.000 47.000 54.000 39.000 32.000 47.000 49.000 41.000
13 49.000 75.000 37.000 49.000 44.000 35.000 48.000 41.000 45.000 53.000 44.000 47.000 93.000 64.000 41.000 44.000 43.000 52.000 39.000 47.000
14 69.000 73.000 55.000 53.000 47.000 61.000 42.000 36.000 48.000 43.000 35.000 44.000 60.000 42.000 56.000 50.000 41.000 40.000 41.000 51.000
15 47.000 65.000 41.000 51.000 57.000 57.000 38.000 35.000 37.000 56.000 62.000 50.000 75.000 55.000 55.000 63.000 40.000 38.000 32.000 43.000
16 53.000 54.000 43.000 54.000 36.000 54.000 41.000 33.000 38.000 41.000 37.000 36.000 80.000 45.000 34.000 40.000 50.000 35.000 60.000 46.000
17 57.000 68.000 43.000 23.000 39.000 45.000 29.000 46.000 35.000 46.000 28.000 34.000 106.000 37.000 39.000 31.000 39.000 64.000 97.000 34.000
18 38.000 49.000 29.000 49.000 31.000 45.000 47.000 33.000 38.000 21.000 28.000 25.000 57.000 28.000 27.000 27.000 35.000 38.000 101.000 29.000
19 22.000 40.000 29.000 37.000 26.000 24.000 27.000 31.000 29.000 36.000 27.000 31.000 40.000 25.000 35.000 29.000 23.000 31.000 50.000 36.000
120
100
80
60
40
20


UMAP-reduced Data (Cosine, r = 20)
0 536.000 3.000 0.000 1.000 1.000 1.000 1.000 1.000 5.000 2.000 1.000 1.000 12.000 7.000 6.000 175.000 10.000 24.000 12.000 0.000 0
1 4.000 633.000 64.000 91.000 3.000 48.000 16.000 6.000 2.000 1.000 2.000 3.000 74.000 7.000 7.000 1.000 6.000 2.000 1.000 2.000 1
2 3.000 82.000 468.000 162.000 4.000 175.000 12.000 4.000 2.000 0.000 0.000 2.000 52.000 3.000 11.000 1.000 3.000 1.000 0.000 0.000 2
3 2.000 18.000 77.000 720.000 2.000 28.000 59.000 4.000 2.000 0.000 2.000 4.000 47.000 3.000 9.000 1.000 4.000 0.000 0.000 0.000 3
4 3.000 12.000 39.000 653.000 2.000 72.000 84.000 9.000 3.000 9.000 2.000 1.000 55.000 2.000 5.000 2.000 6.000 2.000 2.000 0.000 4
5 2.000 134.000 65.000 21.000 2.000 671.000 9.000 4.000 2.000 1.000 2.000 2.000 56.000 1.000 9.000 3.000 2.000 0.000 1.000 1.000 5
6 6.000 16.000 40.000 203.000 4.000 26.000 451.000 59.000 9.000 5.000 10.000 5.000 109.000 3.000 11.000 4.000 7.000 0.000 7.000 0.000 6
7 3.000 7.000 3.000 5.000 0.000 3.000 19.000 782.000 57.000 1.000 5.000 4.000 57.000 14.000 7.000 3.000 17.000 1.000 2.000 0.000 7
8 2.000 7.000 2.000 13.000 1.000 1.000 25.000 66.000 794.000 4.000 8.000 2.000 38.000 1.000 8.000 4.000 7.000 9.000 4.000 0.000 8
9 6.000 6.000 0.000 2.000 2.000 1.000 8.000 18.000 7.000 794.000 52.000 2.000 67.000 4.000 6.000 3.000 11.000 0.000 5.000 0.000 9
10 1.000 2.000 1.000 2.000 2.000 1.000 12.000 6.000 2.000 16.000 914.000 1.000 28.000 2.000 2.000 1.000 1.000 5.000 0.000 0.000 10
11 5.000 23.000 12.000 6.000 0.000 2.000 6.000 6.000 2.000 0.000 0.000 821.000 37.000 4.000 1.000 2.000 61.000 3.000 0.000 0.000 11
12 12.000 41.000 53.000 146.000 0.000 19.000 372.000 78.000 9.000 3.000 4.000 9.000 168.000 34.000 19.000 7.000 3.000 2.000 3.000 2.000 12
13 32.000 13.000 9.000 7.000 3.000 10.000 15.000 8.000 15.000 0.000 1.000 3.000 81.000 702.000 51.000 15.000 9.000 0.000 15.000 1.000 13
14 9.000 34.000 3.000 3.000 0.000 5.000 8.000 19.000 3.000 3.000 5.000 1.000 47.000 11.000 806.000 5.000 22.000 0.000 2.000 1.000 14
15 30.000 4.000 3.000 2.000 0.000 0.000 4.000 2.000 1.000 0.000 0.000 2.000 35.000 12.000 6.000 842.000 9.000 18.000 25.000 2.000 15
16 7.000 5.000 1.000 4.000 0.000 1.000 7.000 9.000 6.000 3.000 2.000 10.000 28.000 8.000 7.000 8.000 784.000 3.000 17.000 0.000 16
17 10.000 4.000 1.000 0.000 0.000 1.000 5.000 3.000 7.000 2.000 0.000 5.000 28.000 5.000 2.000 10.000 34.000 603.000 19.000 201.000 17
18 14.000 0.000 6.000 0.000 2.000 1.000 1.000 10.000 13.000 3.000 2.000 3.000 18.000 64.000 19.000 9.000 357.000 21.000 230.000 2.000 18
19 154.000 3.000 1.000 2.000 1.000 2.000 0.000 1.000 2.000 9.000 0.000 0.000 21.000 7.000 6.000 269.000 85.000 3.000 61.000 1.000 19
1 19 8 0 12 5 17 14 2 18 3 6 11 7 9 16 4 15 13 10
800
600
400
200
0


UMAP-reduced Data (Euclidean, r = 20)
0 46.000 36.000 27.000 97.000 47.000 34.000 47.000 34.000 35.000 38.000 53.000 32.000 38.000 40.000 44.000 35.000 32.000 46.000 37.000 1.000 0
1 39.000 59.000 33.000 119.000 44.000 31.000 49.000 51.000 49.000 61.000 66.000 36.000 66.000 51.000 50.000 54.000 43.000 27.000 45.000 0.000 1
2 36.000 39.000 65.000 117.000 46.000 51.000 34.000 44.000 34.000 43.000 67.000 42.000 45.000 48.000 54.000 31.000 57.000 85.000 37.000 10.000 2
3 48.000 40.000 49.000 142.000 57.000 52.000 61.000 43.000 47.000 46.000 61.000 40.000 61.000 40.000 40.000 54.000 41.000 18.000 42.000 0.000 3
4 37.000 47.000 42.000 100.000 63.000 44.000 48.000 46.000 59.000 41.000 48.000 49.000 50.000 44.000 56.000 51.000 56.000 38.000 42.000 2.000 4
5 45.000 43.000 46.000 111.000 37.000 58.000 38.000 50.000 52.000 55.000 58.000 43.000 55.000 60.000 49.000 48.000 42.000 56.000 42.000 0.000 5
6 53.000 43.000 37.000 117.000 45.000 41.000 58.000 40.000 52.000 42.000 59.000 40.000 54.000 49.000 49.000 51.000 45.000 41.000 50.000 9.000 6
7 55.000 45.000 48.000 102.000 52.000 37.000 45.000 54.000 41.000 45.000 63.000 51.000 46.000 48.000 50.000 42.000 45.000 79.000 42.000 0.000 7
8 52.000 47.000 52.000 106.000 51.000 41.000 45.000 41.000 61.000 46.000 69.000 39.000 47.000 49.000 57.000 46.000 42.000 60.000 44.000 1.000 8
9 46.000 52.000 45.000 97.000 55.000 55.000 36.000 46.000 42.000 57.000 64.000 52.000 52.000 51.000 45.000 37.000 40.000 75.000 47.000 0.000 9
10 39.000 44.000 45.000 102.000 45.000 51.000 41.000 28.000 57.000 43.000 84.000 52.000 43.000 49.000 41.000 53.000 44.000 94.000 43.000 1.000 10
11 39.000 48.000 40.000 105.000 47.000 52.000 39.000 41.000 44.000 50.000 63.000 68.000 51.000 55.000 68.000 40.000 31.000 77.000 33.000 0.000 11
12 44.000 36.000 55.000 109.000 50.000 43.000 46.000 35.000 43.000 44.000 71.000 33.000 62.000 42.000 47.000 50.000 49.000 76.000 48.000 1.000 12
13 41.000 50.000 48.000 100.000 53.000 39.000 56.000 45.000 34.000 50.000 65.000 51.000 53.000 56.000 50.000 52.000 46.000 55.000 46.000 0.000 13
14 55.000 36.000 38.000 108.000 50.000 53.000 40.000 59.000 40.000 58.000 69.000 53.000 53.000 39.000 60.000 51.000 46.000 21.000 58.000 0.000 14
15 58.000 69.000 46.000 102.000 52.000 57.000 27.000 35.000 44.000 64.000 57.000 40.000 38.000 49.000 61.000 67.000 51.000 36.000 44.000 0.000 15
16 43.000 34.000 42.000 84.000 39.000 46.000 37.000 47.000 36.000 59.000 48.000 40.000 54.000 44.000 41.000 35.000 60.000 84.000 37.000 0.000 16
17 41.000 34.000 34.000 112.000 33.000 32.000 52.000 36.000 27.000 55.000 42.000 49.000 39.000 48.000 53.000 33.000 47.000 129.000 44.000 0.000 17
18 27.000 29.000 27.000 75.000 39.000 36.000 42.000 36.000 40.000 32.000 44.000 32.000 37.000 29.000 31.000 33.000 43.000 102.000 41.000 0.000 18
19 44.000 26.000 32.000 52.000 43.000 26.000 26.000 38.000 27.000 20.000 40.000 31.000 30.000 25.000 24.000 28.000 29.000 48.000 39.000 0.000 19
2 17 15 14 11 3 4 1 12 6 0 5 18 10 7 16 8 13 9 19
140 120 100 80 60 40 20 0


UMAP-reduced Data (Cosine, r = 200)
0 527.000 3.000 0.000 1.000 1.000 1.000 1.000 1.000 5.000 2.000 1.000 1.000 12.000 7.000 4.000 185.000 10.000 25.000 12.000 0.000 0
1 2.000 627.000 102.000 83.000 3.000 22.000 16.000 5.000 3.000 1.000 2.000 3.000 76.000 9.000 7.000 1.000 6.000 2.000 1.000 2.000 1
2 2.000 74.000 571.000 137.000 4.000 95.000 12.000 4.000 3.000 1.000 0.000 11.000 51.000 3.000 12.000 1.000 2.000 1.000 1.000 0.000 2
3 2.000 17.000 91.000 718.000 2.000 19.000 57.000 6.000 2.000 1.000 2.000 4.000 44.000 4.000 8.000 1.000 4.000 0.000 0.000 0.000 3
4 3.000 11.000 76.000 651.000 2.000 39.000 80.000 13.000 3.000 10.000 2.000 2.000 53.000 1.000 5.000 2.000 6.000 2.000 2.000 0.000 4
5 2.000 129.000 100.000 20.000 2.000 640.000 10.000 4.000 2.000 1.000 2.000 2.000 59.000 1.000 7.000 3.000 2.000 0.000 1.000 1.000 5
6 6.000 13.000 56.000 197.000 4.000 10.000 458.000 60.000 9.000 6.000 9.000 5.000 111.000 3.000 10.000 4.000 7.000 0.000 7.000 0.000 6
7 3.000 7.000 3.000 5.000 0.000 3.000 19.000 778.000 60.000 1.000 5.000 4.000 58.000 14.000 7.000 3.000 17.000 1.000 2.000 0.000 7
8 2.000 8.000 2.000 12.000 1.000 0.000 22.000 72.000 793.000 4.000 7.000 2.000 38.000 1.000 8.000 4.000 8.000 9.000 3.000 0.000 8
9 5.000 8.000 1.000 2.000 2.000 0.000 8.000 18.000 8.000 790.000 54.000 2.000 68.000 4.000 6.000 3.000 9.000 0.000 6.000 0.000 9
10 1.000 2.000 2.000 2.000 2.000 0.000 12.000 6.000 2.000 17.000 913.000 1.000 28.000 2.000 2.000 1.000 1.000 5.000 0.000 0.000 10
11 5.000 25.000 12.000 6.000 0.000 1.000 7.000 6.000 2.000 0.000 0.000 813.000 38.000 4.000 1.000 2.000 67.000 2.000 0.000 0.000 11
12 12.000 42.000 72.000 143.000 0.000 4.000 352.000 96.000 11.000 3.000 4.000 8.000 168.000 34.000 18.000 7.000 3.000 2.000 3.000 2.000 12
13 31.000 14.000 13.000 9.000 3.000 4.000 15.000 7.000 15.000 1.000 0.000 3.000 82.000 702.000 52.000 15.000 9.000 0.000 14.000 1.000 13
14 9.000 31.000 4.000 3.000 0.000 5.000 8.000 22.000 3.000 3.000 5.000 1.000 46.000 13.000 805.000 5.000 22.000 0.000 1.000 1.000 14
15 30.000 4.000 3.000 2.000 0.000 0.000 4.000 2.000 1.000 0.000 0.000 2.000 35.000 13.000 6.000 826.000 10.000 18.000 39.000 2.000 15
16 6.000 3.000 4.000 4.000 0.000 0.000 7.000 10.000 6.000 4.000 2.000 11.000 29.000 9.000 7.000 11.000 777.000 3.000 17.000 0.000 16
17 10.000 4.000 2.000 0.000 0.000 0.000 5.000 3.000 7.000 2.000 0.000 5.000 29.000 5.000 2.000 9.000 34.000 601.000 21.000 201.000 17
18 13.000 0.000 6.000 0.000 2.000 1.000 2.000 9.000 13.000 3.000 2.000 3.000 18.000 63.000 19.000 9.000 359.000 21.000 230.000 2.000 18
19 152.000 2.000 4.000 2.000 1.000 0.000 0.000 1.000 2.000 10.000 0.000 0.000 20.000 7.000 6.000 266.000 86.000 3.000 65.000 1.000 19
13 18 12 0 7 10 16 3 17 4 15 6 5 1 9 11 14 19 2 8
800
600
400
200
0


UMAP-reduced Data (Euclidean, r = 200)
19 1 14 9 6 10 15 5 0 17 8 4 11 2 12 18 7 3 13 16
0 20 40 60 80 100 120 140

Question 12

Overall, it can be seen from the contingency matrices that using the euclidean distance metric has poor results. There is no clear diagonal structure which indicates higher accuracy – instead, most of the boxes are shaded heavily, indicating many incorrect predictions.

The overall best combination of parameters, based on the contingency matrices and average score measures, is the cosine metric with 5 principle components, although all component count parameters performed with similar accuracy and scores.

In terms of metric, the cosine metric makes sense because of the point mentioned in the spec. Two datapoints may be far apart in euclidean distance, due to a feature such as article length, or distinct words, or any similar feature. However, their actual similarity in topic may be closer, and better represented by the cosine distance, which could be close. Imagine two points, and a right triangle drawn with the hypotenuse being the straight line connecting the points. Let the two legs of the triangle be of length x and $10x$. The hypotenuse would be the Euclidean distance, which is large. However, if x is small, this represents a small cosine. Thus, the reduced TF-IDF data could be better represented in this way, and then the cosine metric would be much more ideal than the euclidean metric.

In terms of component number, all component numbers actually have similar average scores. They are all within a 0.01 score difference of each other. This shows that the cosine metric is the most important factor, and when cosine is used, component number is less significant.

When looking at the euclidean metric, the best component number turns out to be 20. This can be explained by how with a very large dimensionality, the distance between the data points is insignificant and approaches zero, as aforementioned. On the other hand, with a dimensionality too small, the original data is not represented as well. Overall though, the observation that UMAP-reduced data using 200 components still performs close to 20 components may a result of how UMAP performs dimensionality reduction. Specifically, UMAP is effective at preserving non-linear structures in data. With more dimensions, UMAP can better represent the potentially intricate patterns in the original data and thus achieve better separation between clusters. Thus, UMAP may be better at overcoming the “curse of dimensionality.”

Question 13

Average measure scores for KMeans clustering

The following are the highest average measure scores among all parameters attempted, for each of the following data representations.

Two classes

Sparse TF-IDF, 2 clusters: 0.60
SVD-reduced, 7 components, 2 clusters : 0.592
NMF-reduced, 2 components, 2 clusters: 0.523

Among the binary classification results, the sparse TF-IDF matrix as well as SVD-reduced matrix performed the best. They both had roughly a 0.60 average measure score. The NMF representation of data does not hold up to the same levels of accuracy for this use case. However, because dimensionality reduction allows data to be represented much more compactly and efficiently, I would choose SVD over the sparse TF-IDF matrix for this.

When looking at the histograms that show average score as principle component count increase, SVD also proved to have some non-monotonic behavior, which is good because despite clustering not performing well in high dimensional data, it still seems to have decent accuracy for SVD. Lastly, the projection of the top two principle components show good separation between the two classes.

Twenty classes

SVD-reduced, 5 components, 20 clusters : 0.313
NMF-reduced, 10 components, 20 clusters: 0.269
UMAP-reduced, cosine, 5 components, 20 clusters: 0.560

When looking at the average measures for data representation with 20 categories, it seems that UMAP-reduced data, with cosine metric, performed the best. This makes sense since the cosine metric is the biggest difference between these techniques and the reason we are examining it in the first place. When there are large numbers of categories, discriminating between each one is even more crucial since they can all be very close to each other in space. By having the cosine metric, we are able to better differentiate the classes which may be far apart but actually very similar. The negative effect due to noise introduced by different categories is more neutralized with UMAP-reduced data that utilizes the cosine metric.

Thus, overall, both theoretically and practically, using UMAP with the cosine metric to perform dimensionality reduction is the best approach for performing KMeans on the 20-class text data.

Question 14

Homogeneity Score (ward): 0.5607434677659899
Completeness Score (ward): 0.5872483144495927
V-measure Score (ward): 0.5736899190143476
Adjusted Rand Index Score (ward): 0.4223134307995713
Adjusted Mutual Information score (ward): 0.5722691253079922
Average Score (ward): 0.543

Homogeneity Score (single): 0.01790367399262084
Completeness Score (single): 0.3514133864963121
V-measure Score (single): 0.034071486977306185
Adjusted Rand Index Score (single): 0.0004498557610736202
Adjusted Mutual Information score (single): 0.029327801776631867
Average Score (single): 0.086

From inspection, the *ward* linkage criteria seems to have much better performance. From brief research, it seems that *ward* has best performance on noisy data, while *single* is better when speed is a concern. This makes sense, as TF-IDF data, although dimension-reduced, is likely to have a lot of noise since the numbers represent word related metrics and can vary largely.

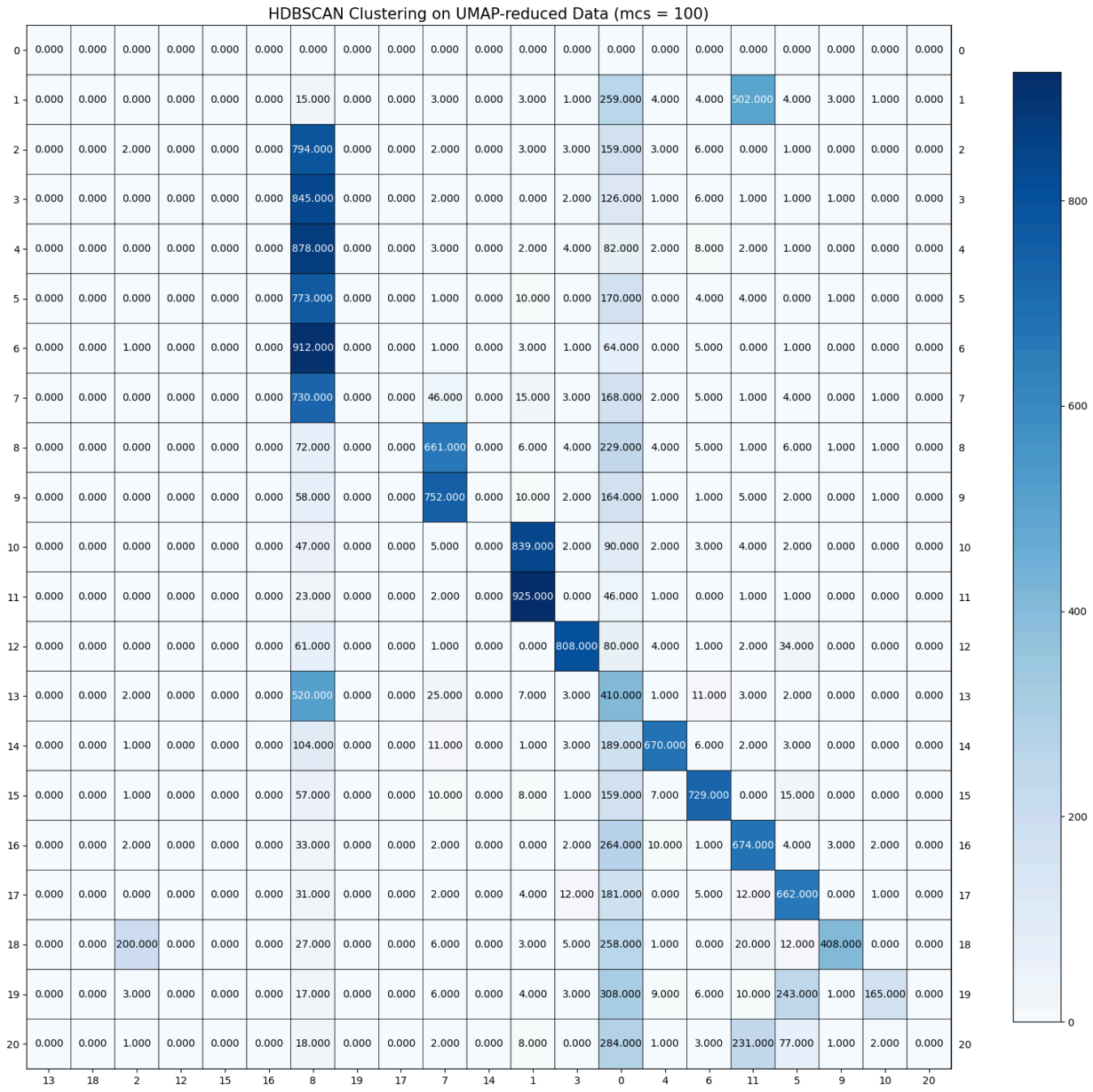
Question 15

Homogeneity Score (HDBSCAN, 20 min_cluster_size): 0.4398871968598709
Completeness Score (HDBSCAN, 20 min_cluster_size): 0.44543310350271065
V-Measure Score (HDBSCAN, 20 min_cluster_size): 0.4426427795864458
Adjusted Rand Score (HDBSCAN, 20 min_cluster_size): 0.09234221253262065
Adjusted Mutual Info Score (HDBSCAN, 20 min_cluster_size): 0.4309473624180231
Average Score (HDBSCAN, 20 min_cluster_size): 0.37025053097993427

Homogeneity Score (HDBSCAN, 100 min_cluster_size): 0.42618792757179286
Completeness Score (HDBSCAN, 100 min_cluster_size): 0.6189769470594981
V-Measure Score (HDBSCAN, 100 min_cluster_size): 0.5048016991100382
Adjusted Rand Score (HDBSCAN, 100 min_cluster_size): 0.21485770591120965
Adjusted Mutual Info Score (HDBSCAN, 100 min_cluster_size): 0.5037040832239996
Average Score (HDBSCAN, 100 min_cluster_size): 0.4537056725753077

Homogeneity Score (HDBSCAN, 200 min_cluster_size): 0.41693169363502175
Completeness Score (HDBSCAN, 200 min_cluster_size): 0.6124544120767882
V-Measure Score (HDBSCAN, 200 min_cluster_size): 0.49612415377385294
Adjusted Rand Score (HDBSCAN, 200 min_cluster_size): 0.2147842364324832
Adjusted Mutual Info Score (HDBSCAN, 200 min_cluster_size): 0.4951060773924176
Average Score (HDBSCAN, 200 min_cluster_size): 0.4470801146621127

Question 16



Out of the 21 labels, 12 clusters are given by the model, which are the clusters labeled by 2, 8, 7, 1, 3, 0, 4, 6, 11, 5, 9, and 10 (left to right). The other labels in the contingency matrix are populated with 0s, meaning no data points received that label.

-1 refers to the datapoints that have not been successfully classified by the clustering algorithm. This means that they do not fit well enough into any cluster.

There is a decently strong diagonal structure to the matrix, which is good, since that means many of the datapoints were correctly classified. However, cluster 8 did receive many false positives, as many points were predicted to be part of that class even when they were not. The same holds true for cluster 0. One initial thought is maybe the minimum cluster size is too large – some data points may belong to their own class / cluster and may have less than the min cluster size of 100 data points. However, from the lumped clusters, it doesn't seem that this is the cause since the lumped clusters existing each have over 100 data points in them. Another possibility is that because HDBSCAN looks at areas of with large variation in density and forms clusters from these datapoints. This can lead to lumping of clusters if multiple clusters are close and have significant density variation.

Question 17

In the appendix at the end of this report are the average measures for each of the dimensionality reduction and clustering method parameter combinations in examination. Since there are a lot of combinations, they are placed at the end to prevent cluttering in this report. All UMAP-reductions are performed using the "cosine" metric. All agglomerative clustering models are performed using ward linkage.

Firstly, when looking at the KMeans results, using UMAP dimensionality reduction with cosine metric noticeably outshone non-reduced data and SVD / NMF-reduced data. Out of the parameters for this combination, the best was using 5 components and 20 clusters (**0.560**). In fact, all of the top scores using KMeans (over 0.50) were performed on UMAP-reduced data. Then, out of the agglomerative clustering results, UMAP-reduced data, again, performed the best (**0.545**). Specifically, this was using 200 principle components, and 20 clusters (the only tested cluster count). Lastly, out of the HDBSCAN results, UMAP with 5 components and HDBSCAN with 100 minimum cluster size performed the best (**0.454**). Overall, it generally appears that UMAP-reduced data as well as KMeans clustering works best when using the average of measure scores as the benchmark.

Un-reduced TF-IDF matrix, SVD, NMF, and UMAP-reduced data have been compared in several ways in the previous questions. The main factor explaining why UMAP dimensionality reduction still remains the best performing dimensionality reduction method in these experiments is the same – cosine is a better representation of distance between data points in the context of news articles and TF-IDF scores. However, there are a few important points regarding why the order of performance of KMeans, Agglomerative Clustering, and HDBSCAN are the way they are.

One characteristic of agglomerative clustering is that it converges to the number of clusters parameterized initially – rather than searching for a certain number of clusters, it starts with individual data points and merges them until that number of clusters remains. Thus, this may not best represent the data in our case, where we know there are a fixed number of labels. This is one possible reason KMeans clustering performed better than agglomerative clustering, when performed on the same dimension-reduced data. Furthermore, specifically when using the ward

linkage criteria in agglomerative clustering, clusters are merged based on a minimal increase in variance – thus, the algorithm attempts to minimize the sum of squared differences of the data points and new cluster centroid. This can lead to less accurate results because TF-IDF datapoints, even after being reduced, may have a lot of overlap in data points that make it so looking at variance is not the best way to cluster.

When looking at HDBSCAN and KMeans clustering, one characteristic of HDBSCAN is that it does not prioritize spherical clusters, while KMeans does. It adapts to the cluster shape, and furthermore, identifies clusters based on areas of varying density. Thus, when data is more naturally spherical, KMeans performs better. This is a potential explanation of why KMeans performed the best – UMAP-reduced TF-IDF data may be inherently spherical, thus catering to KMeans.

Question 18

N/A

Question 19

The VGG network excels in extracting general features in its initial layers that are applicable across various image types, making it highly adaptable. Despite being trained on a dataset with completely different classes, the lower and mid-level features it learns are broad and versatile, capturing essential visual properties useful across a wide range of image recognition tasks. This universality is what gives features derived from a VGG network trained on one dataset discriminative power for a custom dataset, as these features are relevant for distinguishing between objects in new contexts.

Question 20

The code performs feature extraction by leveraging a pretrained VGG-16 model, adapting it to output 4096-dimensional feature vectors instead of class predictions. This is achieved by retaining the model's convolutional and average pooling layers while discarding most of its fully connected layers. The flower photos dataset is processed in batches, with each image resized, normalized, and fed through the modified VGG-16 network to extract features. This method utilizes the VGG-16's deep learning capabilities for efficient feature extraction without training a new model, suitable for tasks like image classification on a custom dataset.

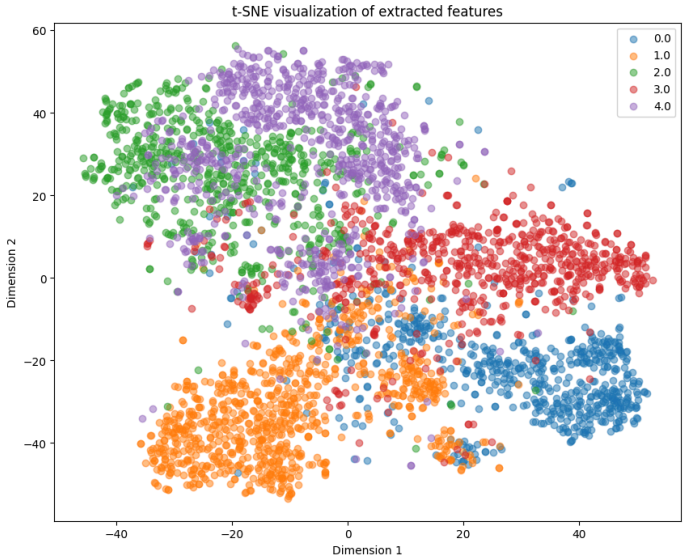
Question 21

The original images' pixels are 320 x 240. For the feature extraction, images are resized and center-cropped to 224 x 224 pixels before being passed into the VGG-16 network. The code uses the VGG-16 model such that it outputs a 4096-dimensional feature vector for each image.

Question 22

The extracted features are dense, as the proportion of non-zero elements to total elements is 1.00. With the sparse TF-IDF, this proportion is 0.0028.

Question 23



The t-SNE plot shows distinct, color-coded clusters corresponding to different classes, indicating that the feature extraction process has captured discriminative patterns. Most of the clusters are distinctly different, but the overlap between some suggests visual similarities among certain classes.

Question 24

Doing dimensionality reduction using UMAP and the K-means clustering algorithm yields a rand-score of **0.79629**, which is the best performing score. UMAP resulted in the highest rand scores across the reduction algorithms, with K-means performing the best among the 3 clustering algorithms.

UMAP + K-means: 0.7962947532273864
UMAP + Agglomerative Clustering: 0.7885879409412242
UMAP + HDBSCAN: 0.7248875808285488
SVD + K-means: 0.7029135038911329
SVD + Agglomerative Clustering: 0.6957926452054662
SVD + HDBSCAN: 0.4129367266656418
Autoencoder + K-means: 0.7041633897081595
Autoencoder + Agglomerative Clustering: 0.693344265192648
Autoencoder + HDBSCAN: 0.4486293958588156

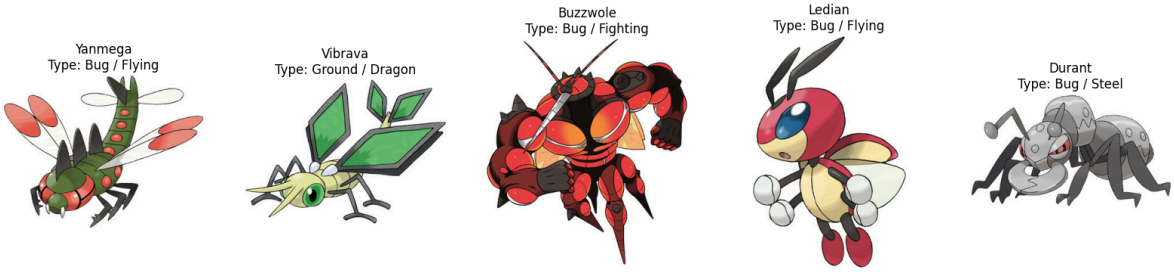
Question 25

MLP Accuracy Original VCG Features: 90%
MLP Accuracy Autoencoder + K_means: 88%
MLP Accuracy SVD + K_means: 89%
The performance of the model does not suffer with reduced-dimension representations, as demonstrated by the accuracies. This success makes sense, as demonstrated in part 24, where clustering after dimensionality reduction still resulted in effective rand-scores similar to clustering without dimensionality reduction.

Question 26

Top 5 Pokémon for Grass

Top 5 Pokémon for Bug



Top 5 Pokémon for Fire

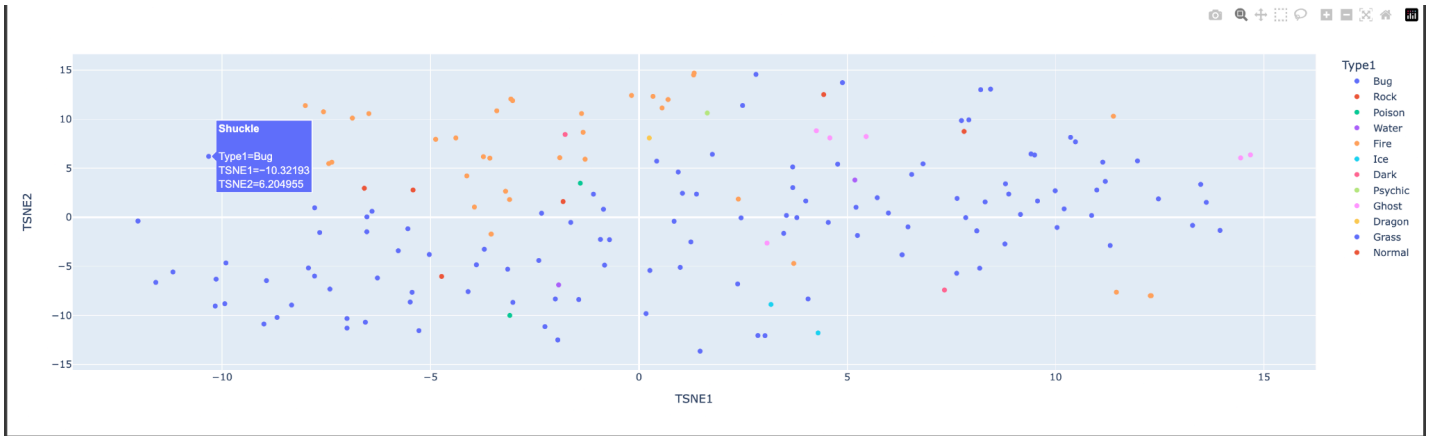
Top 5 Pokémon for Dark

Top 5 Pokémon for Dragon

The constructed queries were very effective for Fire, Bug, and Grass. For Dark and Dragon, there were a few misclassifications but understandably so, as they did take on the appearance of Dark and Dragon pokemon.

Question 27

Question 28



The t-SNE clusters the bug and grass pokemon in the same plane, which makes sense given how similar these pokemon look. The fire and dragon pokemon are typically classified just above it, which is also logical. There are various scattered outliers of various types, but overall the classification does make sense.

Appendix

Question 17 Results

Average Score [Unreduced] [KMeans, k = 10]: 0.3127298558065917
Average Score [SVD, r = 5] [KMeans, k = 10]: 0.28826121545814104
Average Score [SVD, r = 20] [KMeans, k = 10]: 0.28103632815154544
Average Score [SVD, r = 200] [KMeans, k = 10]: 0.27397170620866157
Average Score [NMF, r = 5] [KMeans, k = 10]: 0.24105909324884447
Average Score [NMF, r = 20] [KMeans, k = 10]: 0.21331247059166847
Average Score [NMF, r = 200] [KMeans, k = 10]: 0.032385718417162965
Average Score [UMAP, r = 5] [KMeans, k = 10]: 0.5001602918761598
Average Score [UMAP, r = 20] [KMeans, k = 10]: 0.501103386927099
Average Score [UMAP, r = 200] [KMeans, k = 10]: 0.4998333858310423
Average Score [Unreduced] [KMeans, k = 20]: 0.3037430666748485
Average Score [SVD, r = 5] [KMeans, k = 20]: 0.2918915174997078
Average Score [SVD, r = 20] [KMeans, k = 20]: 0.3065475948781547
Average Score [SVD, r = 200] [KMeans, k = 20]: 0.32461349858546773
Average Score [NMF, r = 5] [KMeans, k = 20]: 0.2396382853881612
Average Score [NMF, r = 20] [KMeans, k = 20]: 0.2627086703407877
Average Score [NMF, r = 200] [KMeans, k = 20]: 0.11487118434220356
Average Score [UMAP, r = 5] [KMeans, k = 20]: 0.5604899669467989
Average Score [UMAP, r = 20] [KMeans, k = 20]: 0.5583838889473812
Average Score [UMAP, r = 200] [KMeans, k = 20]: 0.5576163071143514
Average Score [Unreduced] [KMeans, k = 50]: 0.36797777775698615
Average Score [SVD, r = 5] [KMeans, k = 50]: 0.2828190172917353
Average Score [SVD, r = 20] [KMeans, k = 50]: 0.32867074172086935
Average Score [SVD, r = 200] [KMeans, k = 50]: 0.3669264234627544
Average Score [NMF, r = 5] [KMeans, k = 50]: 0.23368389238229748
Average Score [NMF, r = 20] [KMeans, k = 50]: 0.3047935570521014
Average Score [NMF, r = 200] [KMeans, k = 50]: 0.13782058586457885
Average Score [UMAP, r = 5] [KMeans, k = 50]: 0.5151200015920188
Average Score [UMAP, r = 20] [KMeans, k = 50]: 0.5253759078637479
Average Score [UMAP, r = 200] [KMeans, k = 50]: 0.5154857074957021

Average Score [Unreduced] [Agglomerative Clustering, k = 20]: N/A, sparse matrices not allowed
Average Score [SVD, r = 5] [Agglomerative Clustering, k = 20]: 0.28175825239625946
Average Score [SVD, r = 20] [Agglomerative Clustering, k = 20]: 0.3548089047105772
Average Score [SVD, r = 200] [Agglomerative Clustering, k = 20]: 0.32757379932917896
Average Score [NMF, r = 5] [Agglomerative Clustering, k = 20]: 0.23868230288974498
Average Score [NMF, r = 20] [Agglomerative Clustering, k = 20]: 0.3318282950981767
Average Score [NMF, r = 200] [Agglomerative Clustering, k = 20]: 0.1083850849800305
Average Score [UMAP, r = 5] [Agglomerative Clustering, k = 20]: 0.5432528514674988
Average Score [UMAP, r = 20] [Agglomerative Clustering, k = 20]: 0.5413746329195541
Average Score [UMAP, r = 200] [Agglomerative Clustering, k = 20]: 0.5446923226090532

Average Score [Unreduced] [HDBSCAN, mcs = 100]: N/A, not enough memory
Average Score [SVD, r = 5] [HDBSCAN, mcs = 100]: 0.2
Average Score [SVD, r = 20] [HDBSCAN, mcs = 100]: 0.2
Average Score [SVD, r = 200] [HDBSCAN, mcs = 100]: 0.2
Average Score [NMF, r = 5] [HDBSCAN, mcs = 100]: 0.1067468991884675
Average Score [NMF, r = 20] [HDBSCAN, mcs = 100]: 0.2
Average Score [NMF, r = 200] [HDBSCAN, mcs = 100]: 0.2
Average Score [UMAP, r = 5] [HDBSCAN, mcs = 100]: 0.4537056725753077
Average Score [UMAP, r = 20] [HDBSCAN, mcs = 100]: 0.4516924983408176
Average Score [UMAP, r = 200] [HDBSCAN, mcs = 100]: 0.4391426102513131
Average Score [Unreduced] [HDBSCAN, mcs = 200]: N/A, not enough memory
Average Score [SVD, r = 5] [HDBSCAN, mcs = 200]: 0.2
Average Score [SVD, r = 20] [HDBSCAN, mcs = 200]: 0.2
Average Score [SVD, r = 200] [HDBSCAN, mcs = 200]: 0.2
Average Score [NMF, r = 5] [HDBSCAN, mcs = 200]: 0.048623498974678075
Average Score [NMF, r = 20] [HDBSCAN, mcs = 200]: 0.2
Average Score [NMF, r = 200] [HDBSCAN, mcs = 200]: 0.2
Average Score [UMAP, r = 5] [HDBSCAN, mcs = 200]: 0.4470801146621127
Average Score [UMAP, r = 20] [HDBSCAN, mcs = 200]: 0.44345733152529004
Average Score [UMAP, r = 200] [HDBSCAN, mcs = 200]: 0.44260265052562636