Large-Scale Data Mining: Models and Algorithms ECE 219 Winter 2023
Prof. Vwani Roychowdhury
UCLA, Department of ECE

Project 2: Data Representations and Clustering

Sudeeksha Agrawal, Tazeem Khan, Vamsi Krishna Pamidi

UID: 305928941,105946724,805945580

Introduction:

In this project, we will be exploring feature extraction and clustering techniques for text and image data. The goal is to use these techniques to automatically separate a document set or an image dataset into groups that match known labels (or categories). The project will begin by exploring feature extraction from text data and continue with clustering techniques using K-means, DBSCAN, HDBSCAN and Hierarchical clustering. Then, the project will shift its focus to image data and explore how to use deep learning or deep neural networks (DNNs) to obtain image features. The pre-trained networks could be considered as experienced agents that have learned to discover features that are salient for image understanding, and can be used for transfer learning. The project will conclude by using a common set of multiple evaluation metrics (contingency matrix, homogeneity score, completeness score, V-measure score, adjusted Rand Index score, and adjusted mutual information score) to compare the groups extracted by the unsupervised learning algorithms with the corresponding ground truth human labels.

Part1: Clustering on test data

Clustering with Sparse Text Representations:

QUESTION 1: Report the dimensions of the TF-IDF matrix you obtain.

Answer 1:

The 20 Newsgroups dataset has been given to us, but we will only be focusing on 8 of the 20 categories. This will be transformed into a two-class clustering problem by merging all of the "comp" related classes into one and all of the "rec" related classes into another and doing feature extraction similar to Project1.

- During feature extraction, the headers and footers will be removed from the dataset.
- No stemming or lemmatization will be applied.
- TfidfVectorizer() will be used to convert the text into a TF-IDF matrix, while removing English stop-words, setting a minimum document frequency of 3 to remove rare words that appear in less than 3 documents and highlighting the discriminatory words in each document against the more common words to better differentiate between the classes during clustering.

The **dimensions of the TF-IDF matrix thus obtained are 7882 rows and 23,522 columns.**

QUESTION 2: Report the contingency table of your clustering result. You may use the provided plotmat.py to visualize the matrix. Does the contingency matrix have to be square-shaped?

Answer 2:

Clustering is the process of grouping similar data based on their features without the use of pre-labelled classes. The contingency matrix, M, can be used to evaluate the accuracy of the clustering algorithm, where M_{ij} represents the true class, i and the predicted class/cluster, j.

After obtaining the TF-IDF matrix, we applied K-means clustering algorithm on the data with parameters: random_state = 0, max_iter = 1000 and n_init = 30. When we compare the predicted labels with expected results (we took ‘comp.sys.ibm.pc.hardware’, ‘comp.graphics’, ‘comp.sys.mac.hardware’, and ‘comp.os.ms-windows.misc’ as one class, and considered ‘rec.autos’, ‘rec.motorcycles’, ‘rec.sport.baseball’, and ‘rec.sport.hockey’ as second class), we obtain contingency matrix as shown below.

The results show that the majority of the documents in the "comp" class were correctly assigned to the "comp" cluster (3228 documents), and similarly, most of the documents in the "rec" class were assigned to the "rec" cluster (3923 documents). However, there were some misclassifications, such as 56 "rec" documents being placed in the "comp" cluster and 675 "comp" documents being placed in the "rec" cluster. Despite these misclassifications, the clustering algorithm appears to have accurately grouped the majority of the sample points into their respective clusters.

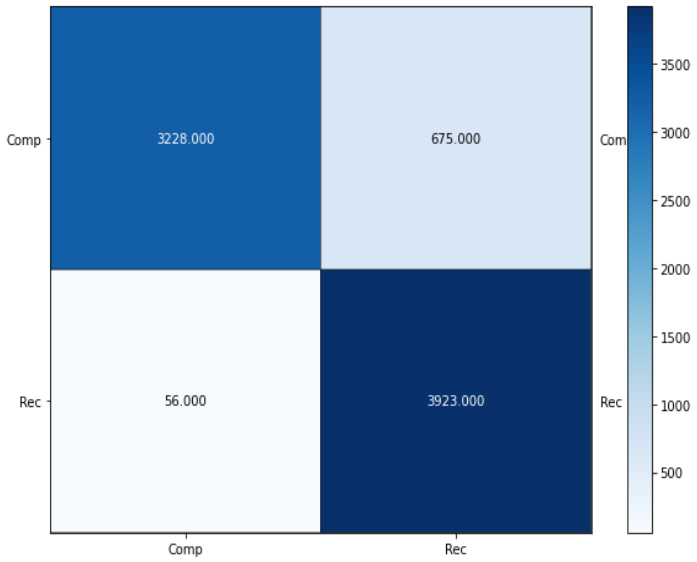

Fig1: Contingency Matrix

The shape of the contingency matrix for this particular question must be square because there are only two classes and the k-means algorithm creates two clusters. The contingency matrix expands on the concept of a confusion matrix, which displays the number of samples in the ground truth clusters on the rows and the number of samples that the clustering algorithm separates into each cluster on the columns. The diagonal entries in the matrix represent the accurate categorizations.

In a general sense, the contingency matrix displays the ground truth clusters or classes on the rows and the predicted clusters on the columns. For example, if the data has 20 classes and the clustering algorithm creates 40 clusters, the contingency matrix will only have non-zero entries in the 20x40 rectangle. If the clustering solution only creates 10 clusters, the non-zero entries will only be in the 20x10 rectangle. Typically, the contingency matrix is displayed as a square with zeros in the remaining entries.

QUESTION 3: Report the 5 clustering measures explained in the introduction for K-means clustering.

Answer 3:

5 clustering measures explained in the introduction for K-means clustering are:

- **Homogeneity** is a measure of how “pure” the clusters are. If each cluster contains only data points from a single class, the homogeneity is satisfied.
- **Completeness** indicates how much of the data points of a class are assigned to the same cluster.
- **V-measure** is the harmonic average of homogeneity score and completeness score.
- **Adjusted Rand Index** is similar to accuracy, which computes similarity between the clustering labels and ground truth labels. This method counts all pairs of points that both fall either in the same cluster and the same class or in different clusters and different classes.
- **Adjusted mutual information score** measures the mutual information between the cluster label distribution and the ground truth label distributions.

The values of these 5 clustering methods using sklearn.metrics are obtained as below:

- Homogeneity score: 0.597026
- Completeness score: 0.609257
- V-measure score: 0.603080

- Adjusted Rand Index score: 0.663391
- Adjusted mutual information score: 0.603043

Clustering with Dense Text Representations:

QUESTION 4: Report the plot of the percentage of variance that the top r principle components retain v.s. r, for r = 1 to 1000.

Answer 4:

The K-means algorithm has limitations when it comes to high-dimensional data and when clusters are not of equal size or shape. To improve the results, a better representation of the data must be found before applying the K-means algorithm. To determine the effective dimension of the data, the top singular values of the TF-IDF matrix were inspected. The explained variance ratio of the "Truncated SVD" object was used to calculate the percentage of variance retained by the top r principal components for r=1 to 1000. The results, shown in Figure 2, reveal that about 50% of the variance is covered by around 1000 features. The concavity of the plot suggests that as the number of components increases, the variance also increases, but at a slower pace. The difference in variance explained between r=50 and r=100 is greater than that between r=950 and r=1000. This makes sense because the components are selected based on the descending order of their variance explanation ratio.

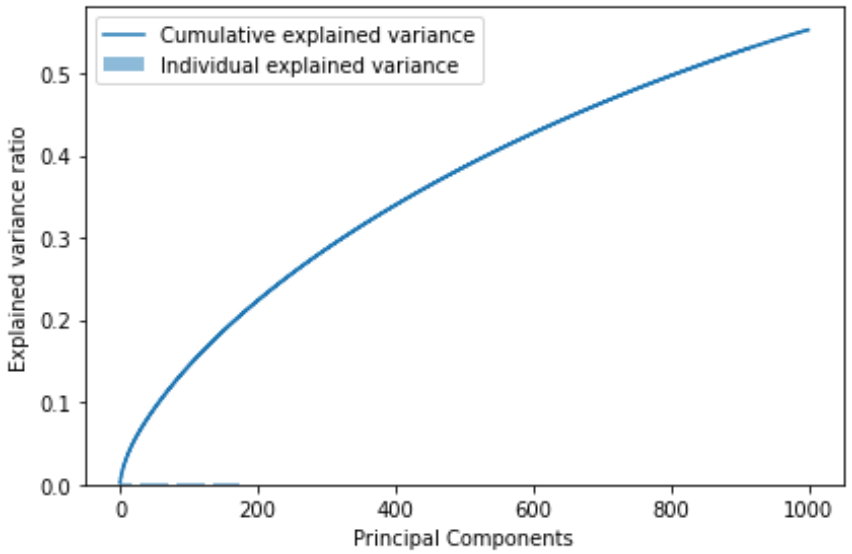

Fig2: Percentage of Variance retained by top r Principal Components

QUESTION 5: Let r be the dimension that we want to reduce the data to (i.e. n components). Try r = 1 − 10, 20, 50, 100, 300, and plot the 5 measure scores v.s. r for both SVD and NMF. Report a good choice of r for SVD and NMF respectively.

Answer 5:

As the number of components (r) increases, the amount of complex information and patterns in the data that can be used for clustering also increases. However, this also leads to a decrease in the performance of K-means clustering. The reason for this is that as the number of components grows, the Euclidean distance between sample points becomes more constant, and which is an important metric for K-means clustering.

To determine the best rank for truncation methods such as Singular Value Decomposition (SVD) and Non-negative Matrix Factorization (NMF), five measures (homogeneity, completeness, V-measure, adjusted rand index (ARI), and adjusted mutual information score (AMI)) were calculated. The rank was selected from the following options: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 50, 100, 300]. The aim was to find the rank that provides just enough information without being too large or vast, as this can negatively impact K-means clustering performance. The results obtained are shown in the images below.

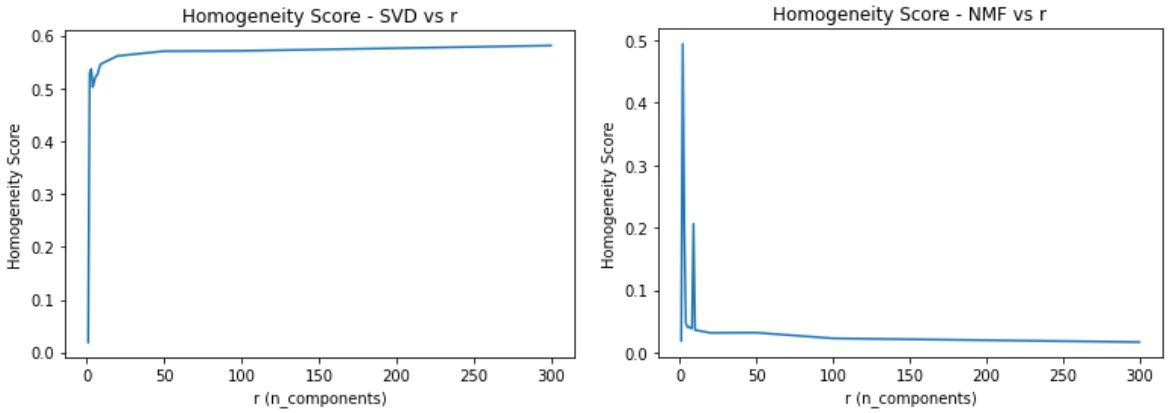

Fig3: Homogeneity Score by r Principal Components for SVD and NMF

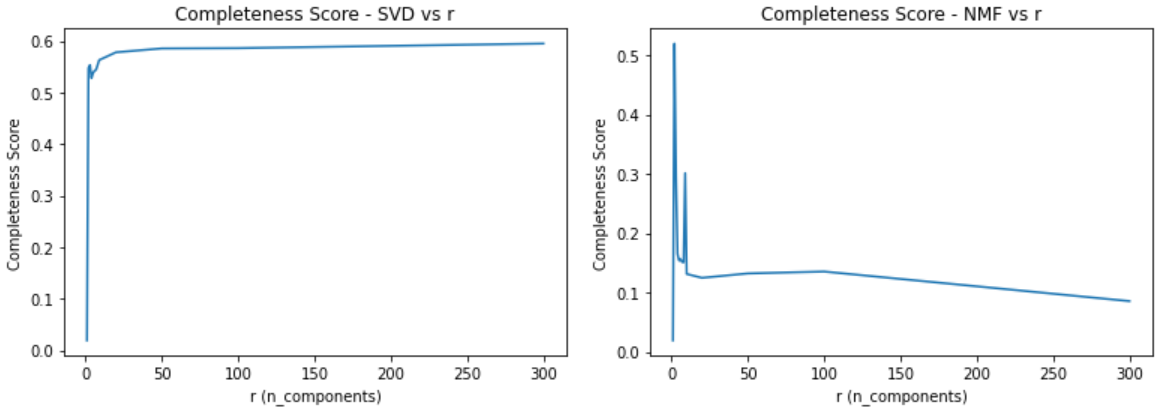

Fig4: Completeness Score by r Principal Components for SVD and NMF

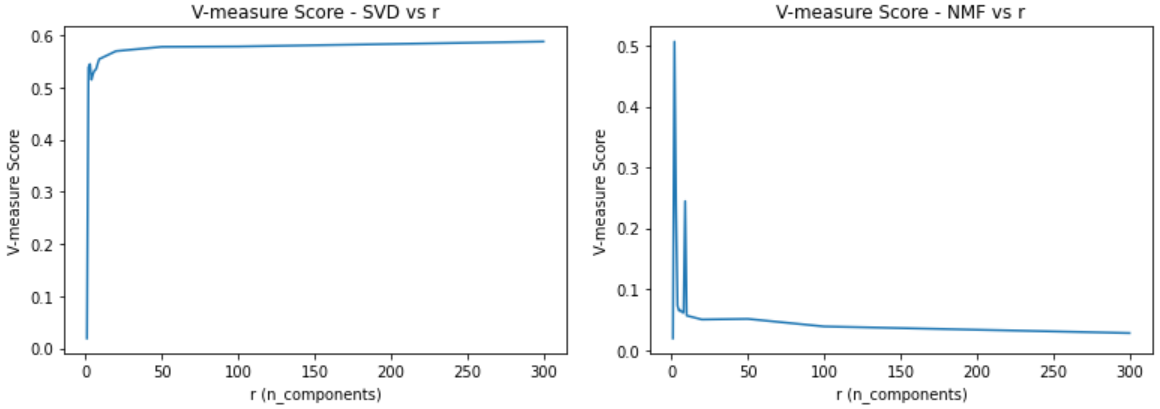

Fig5: V-measure Score by r Principal Components for SVD and NMF

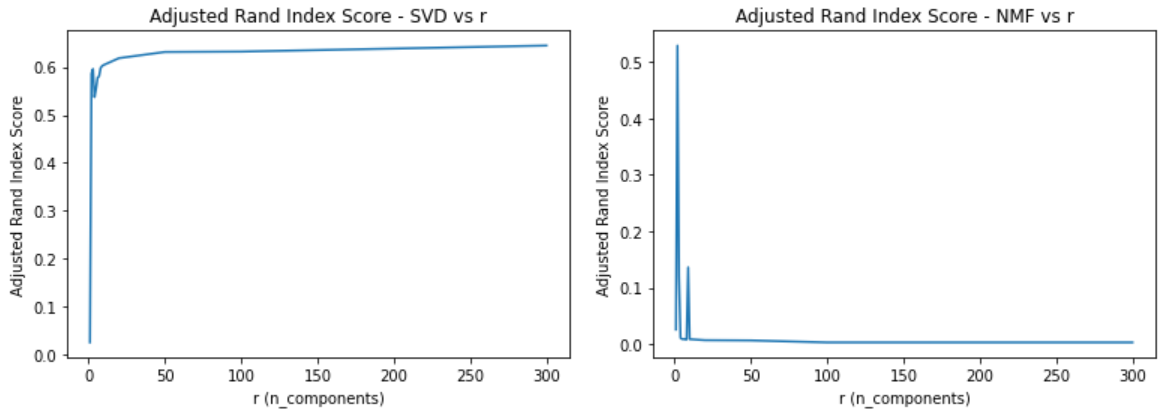

Fig6: Adjusted Rand Index Score by r Principal Components for SVD and NMF

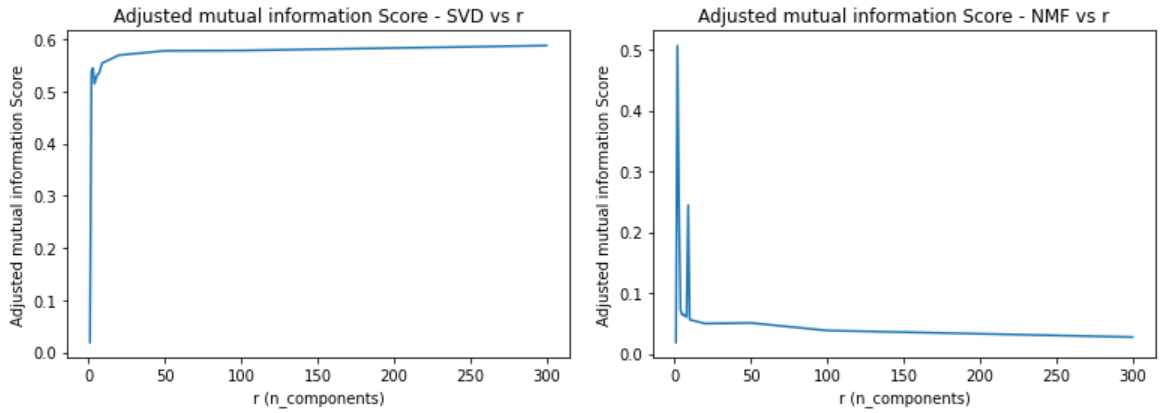

Fig7: Adjusted Mutual Information Score by r Principal Components for SVD and NMF

A good choice for r is obtained by taking average of all the 5 metrics and plotting them against r for both SVD and NMF as shown below:

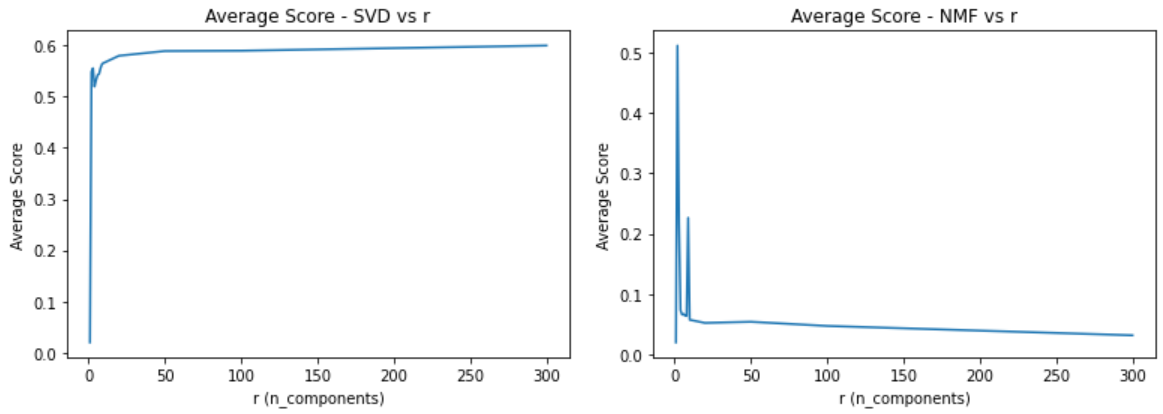

Fig8: Average Measure Score by r Principal Components for SVD and NMF

This shows that **r=300 for SVD** with an average measure value =0.5993426580991089 and **r=2 for NMF** with average measure= 0.5114096565949626 give the best results.

In analysing the results of the clustering algorithms applied on the data, it was observed that there is a balance between maintaining the information in the data and achieving better k-means performance in lower dimensions. For the Non-negative Matrix Factorization (NMF) truncation method, the best results were obtained with r = 2, where the information preservation and k-means performance are in harmony. On the other hand, in the case of Singular Value Decomposition (SVD), r = 300 had the best scores, but the difference in the scores between r = 100 (Avg. score=0.5892717934077527) and the higher dimensions was minimal. As a result, it is more logical

to choose r = 100 for SVD, as increasing the dimensions further does not lead to significant improvements in the results.

QUESTION 6: How do you explain the non-monotonic behaviour of the measures as r increases?

Answer 6:

The non-monotonic behaviour of measures as r increases can be explained as:

- The results of clustering metrics do not always improve with an increase in the number of components (r).
- Intuitively, it might seem that increasing r would lead to a better identification of higher-level features and improved clustering properties. This is because a higher r value indicates a larger amount of semantic information and patterns in the data that can be used for clustering.
- However, as we enter higher dimensions, the data becomes sparser along each dimension, which leads to a convergence of Euclidean distances between sample points to a constant value.
- This equidistant feature space without normalization makes it difficult for the clustering algorithm to find centroids with enough inter-cluster distances.
- This effect is more pronounced in the results from NMF, as NMF only allows positive entries in the reduced-feature matrix. In contrast, SVD allows for more accurate representation of higher-dimensional feature matrices, as it is more deterministic and considers the geometry in the feature space that is important for clustering.
- Additionally, SVD sorts the features based on their importance in explaining variance, meaning that the more important features are given a higher priority. As a result, with increasing dimensions, there is a marginal increase in information, but not much added noise. This is why SVD shows only a marginal improvement when r is increased from 20 to 200, while NMF exhibits a more non-monotonic behaviour.
- Moreover, the outcomes obtained from SVD are distinct and the outcomes obtained from NMF are non-unique and dependent on chance, with no assurance that the optimal feature matrix will be reached each time the function is executed.

QUESTION 7: Are these measures on average better than those computed in Question 3?

Answer 7:

We can compare the measures computed in Q3 without dimensionality reduction with the measures we obtain in Q5 with dimensionality reduction using SVD/NMF for best r (no of components) in the table below.

Measure	Q3: No Dimensionality Reduction	Q5: SVD for best r=300	Q5: SVD with r=100	Q5: NMF with best r=2
Homogeneity score	0.597026	0.5812549696357695	0.5712651570193955	0.49436629542018734
Completeness score	0.609257	0.5949452135583514	0.5860582025920705	0.5201076638663424
V-measure score	0.603080	0.5880204186037479	0.5785671365670185	0.5069103975543827
Adjusted Rand Index score	0.663391	0.6445104262991748	0.6319404101570526	0.5287998174841438
Adjusted mutual information score	0.603043	0.587982262398501	0.5785280607032259	0.5068641086497566

We observe that the values obtained in Q5 (post dimensionality reduction using SVD, NMF) for the measures are slightly less or similar to the measures we get in Q3. This observation can be explained as:

- The main difference between Singular Value Decomposition (SVD) and Non-negative Matrix Factorization (NMF) is that SVD does not limit the values in the reduced-rank feature matrix, while NMF only allows positive values. This allows SVD to better represent the higher dimensional feature matrix, providing a more complete factorization with less information loss.

- SVD is also more deterministic than NMF as it is based on the relevant geometric basis, while NMF disregards this, which is a crucial aspect for higher-dimensional clustering. As a result, SVD scores are generally higher than those of NMF.
- Comparing SVD with no dimensionality reduction, from a k-means point of view, SVD with a higher value of r adds very little additional information that could be useful for clustering, but it also does not mask any important information with excessive noise. SVD performance increases only slightly with increasing r beyond r = 100, and no dimensionality reduction can be considered as a very large r.
- Thus, scores without dimensionality reduction are very similar but slightly better than those of SVD with r = 100 or r = 300. This could be because some of the less important features removed by SVD.

QUESTION 8: Visualize the clustering results for:

• SVD with your optimal choice of r for K-Means clustering;

• NMF with your choice of r for K-Means clustering.

Answer 8:

We can visualise the clustering data points compared to the dataset labels, using SVD and NMF with optimal r for K-means clustering as below.

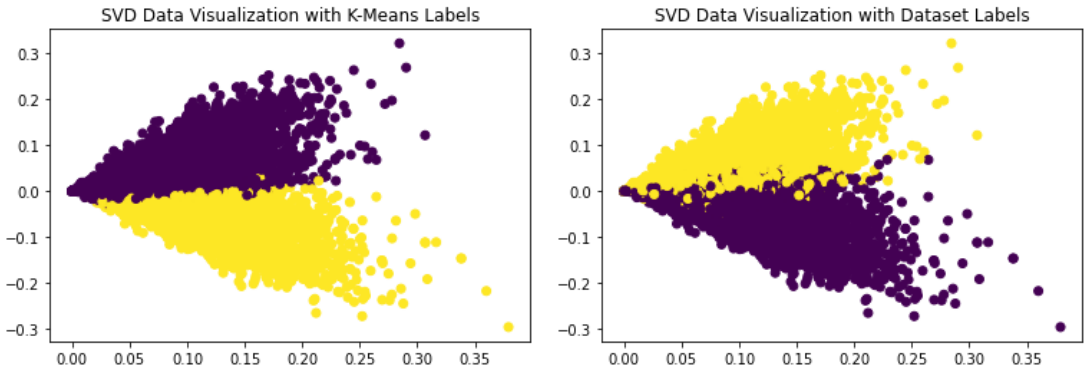

Fig9: Plot for SVD feature matrix for r=300

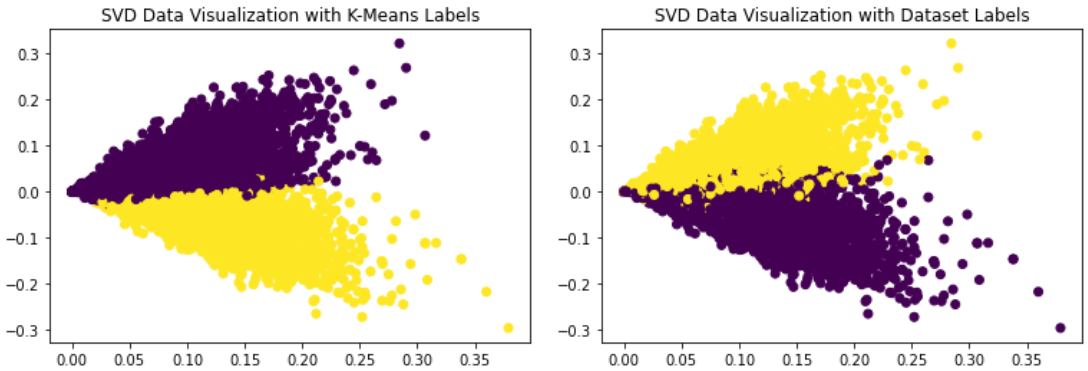

Fig9: Plot for SVD feature matrix for r=100

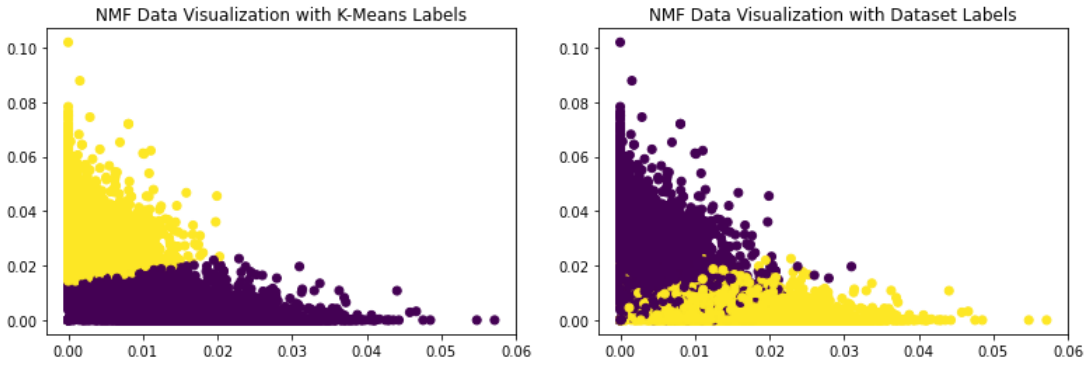

Fig9: Plot for NMF feature matrix for r=2

QUESTION 9: What do you observe in the visualization? How are the data points of the two classes distributed? Is distribution of the data ideal for K-Means clustering?

Answer 9:

We can interpret multiple observations from the visualizations above which show that the data distribution isn't ideal for K-Means clustering:

- K-means algorithm assumes that the clusters being considered have a Gaussian distribution, with equal variance and a convex and isotropic shape. However, the results obtained from SVD and NMF show 2 clusters with unequal variance and irregular, elongated shapes more prominent in NMF that has the yellow clusters more tightly packed than the purple one. So, centroids get affected by noise and outliers in data.
- K-means++ initializes the centroids to be far from each other, but the results from both SVD and NMF show clusters with high overlap, resulting in a minimal Euclidean distance between the centroids, making it difficult to define a decision boundary. The low Homogeneity and V-measure scores reflect this deviation from the assumptions made by the K-means algorithm.
- The K-means method utilizes Euclidean distance, which assumes convex and isotropic clusters, but the results from SVD and NMF indicate that this is not the case, as the clusters have irregular shapes that do not resemble a spherical structure. NMF's structure are extremely contradicting the assumption as their size of blobs is uneven across classes.

The results suggest that the distribution of the dataset is not ideal for K-means clustering as it doesn't fully meet the assumptions and requirements. However, the dataset is still reasonable to use K-means on as it can be divided into two clusters by an axis with decent accuracy and its shape is not overly complicated. Logarithmic transformations can improve the clustering results by spreading the data points more uniformly and smoothing out outliers. However, the high dimensionality of the dataset poses a challenge for distance-based similarity measures, including K-means, due to the curse of dimensionality.

QUESTION 10: Load documents with the same configuration as in Question 1, but for ALL 20 categories. Construct the TF-IDF matrix, reduce its dimensionality using BOTH NMF and SVD (specify settings you choose and why), and perform K-Means clustering with k=20 .Visualize the contingency matrix and report the five clustering metrics (DO BOTH NMF AND SVD).

Answer 10:

We perform similar steps as above but now we do them for all the 20 categories in the dataset as follows:

- Eliminate headers and footers during import.
- Do not apply stemming or lemmatization.
- Utilize TfidfVectorizer() to change the text into a TF-IDF matrix that emphasizes the unique words in each document and helps distinguish between classes during clustering.
- Remove English stop-words and set the minimum document frequency to 3 to eliminate infrequent words.

- Decrease the dimensionality using SVD or NMF and experiment with various values of r (number of components) in the range: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 20, 50, 100, 300], and finding the optimal r.
- Use k-means clustering to divide the data into k=20 clusters with parameters: random_state = 0, max_iter = 1000 and n_init = 30.

Measure	r=1	r=2	r=3	r=4	r=5	r=6	r=7	r=8	r=9	r=10	r=20	r=50	r=100	r=300
Homogeneity score	0.02	0.21	0.25	0.31	0.32	0.31	0.32	0.33	0.32	0.32	0.34	0.32	0.32	0.29
Completeness score	0.03	0.22	0.27	0.33	0.35	0.34	0.35	0.35	0.35	0.35	0.38	0.39	0.38	0.37
V-measure score	0.03	0.22	0.26	0.32	0.34	0.33	0.33	0.34	0.33	0.34	0.36	0.35	0.35	0.32
Adjusted Rand Index score	0.01	0.07	0.08	0.12	0.13	0.12	0.12	0.12	0.13	0.12	0.12	0.10	0.10	0.09
Adjusted mutual information score	0.02	0.21	0.25	0.32	0.33	0.32	0.33	0.34	0.33	0.33	0.35	0.35	0.35	0.32
Average Measure	**0.02**	**0.19**	**0.22**	**0.28**	**0.29**	**0.28**	**0.29**	**0.30**	**0.29**	**0.29**	**0.31**	**0.30**	**0.30**	**0.28**

We get a **TF-IDF matrix of shape (18846, 45365)** whose dimensionality reduced by SVD gives below measures for various values of r.

We observe that the best average measure values of **SVD are for r=20** and hence that's used to apply k-means clustering with k=20 and get the contingency matrix below:

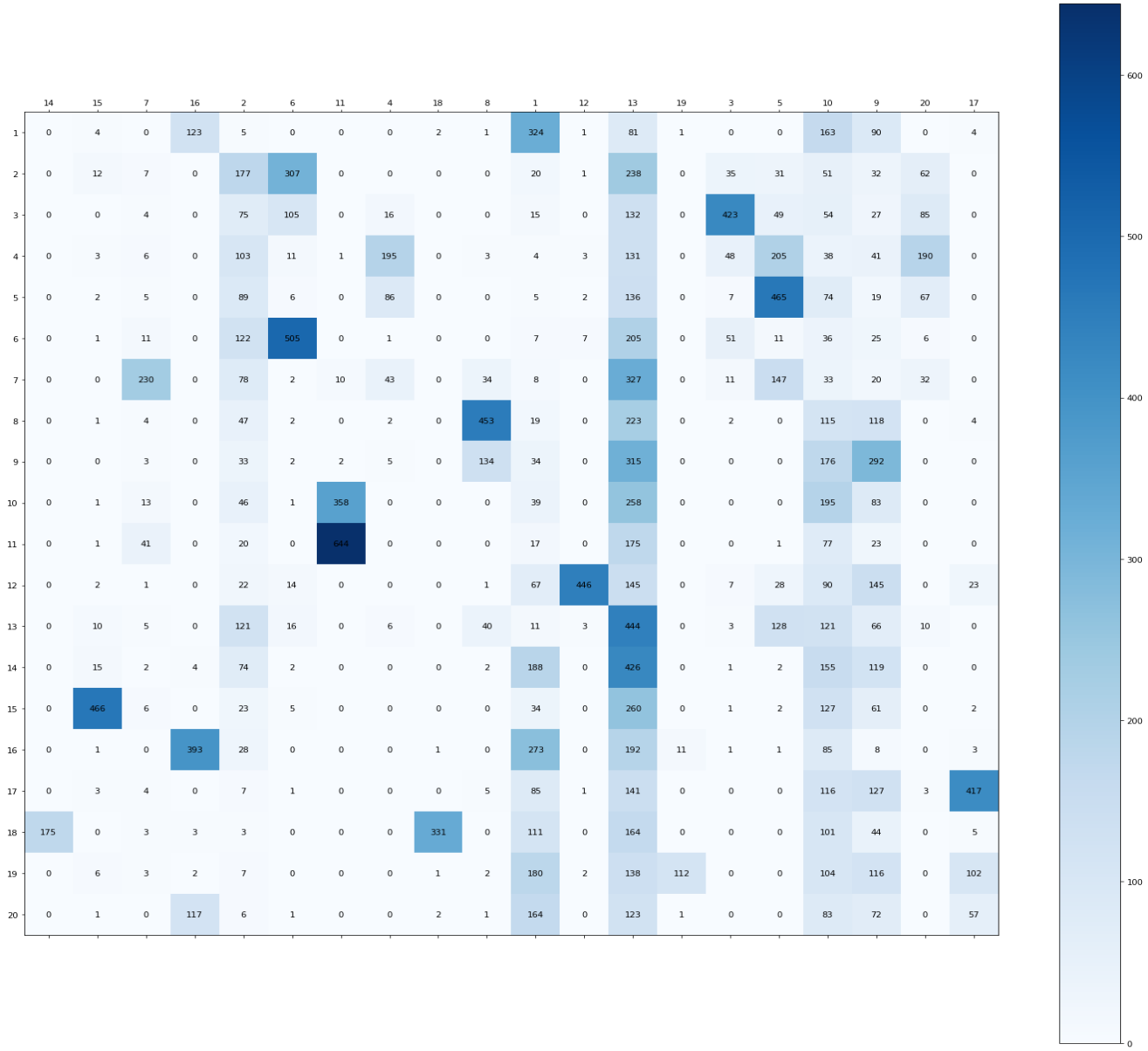

Fig10: Plot for SVD contingency matrix for r=20

The metrics for this SVD(r=20) with K-means clustering (k=20) are as follows:

- Homogeneity score: 0.336158
- Completeness score: 0.378021
- V-measure score: 0.355862
- Adjusted Rand Index score: 0.120619
- Adjusted mutual information score: 0.353652

The average value of metrics for SVD, r=20 and K-means clustering (k=20) are: 0.308862

Similarly, dimensionality reduction by NMF on the TF-IDF matrix of shape (18846, 45365), gives below measures for various values of r.

Measure	r=1	r=2	r=3	r=4	r=5	r=6	r=7	r=8	r=9	r=10	r=20	r=50	r=100	r=300
Homogeneity score	0.02	0.19	0.22	0.23	0.27	0.28	0.29	0.3	0.3	**0.33**	0.33	0.15	0.14	0.07
Completeness score	0.03	0.2	0.24	0.27	0.29	0.31	0.32	0.34	0.34	**0.38**	0.39	0.21	0.21	0.09
V-measure score	0.03	0.2	0.23	0.25	0.28	0.3	0.31	0.32	0.32	**0.35**	0.36	0.17	0.17	0.08
Adjusted Rand Index score	0.01	0.06	0.06	0.07	0.1	0.1	0.11	0.1	0.11	**0.12**	0.1	0.03	0.02	0.01
Adjusted mutual information score	0.02	0.19	0.22	0.25	0.27	0.29	0.31	0.32	0.31	**0.35**	0.36	0.17	0.16	0.07
Average Measure	**0.02**	**0.17**	**0.19**	**0.21**	**0.24**	**0.26**	**0.27**	**0.28**	**0.28**	**0.31**	**0.31**	**0.15**	**0.14**	**0.06**

We observe that the best average measure values of **NMF are for r=10** and hence that's used to apply k-means clustering with k=20 and get the contingency matrix below:

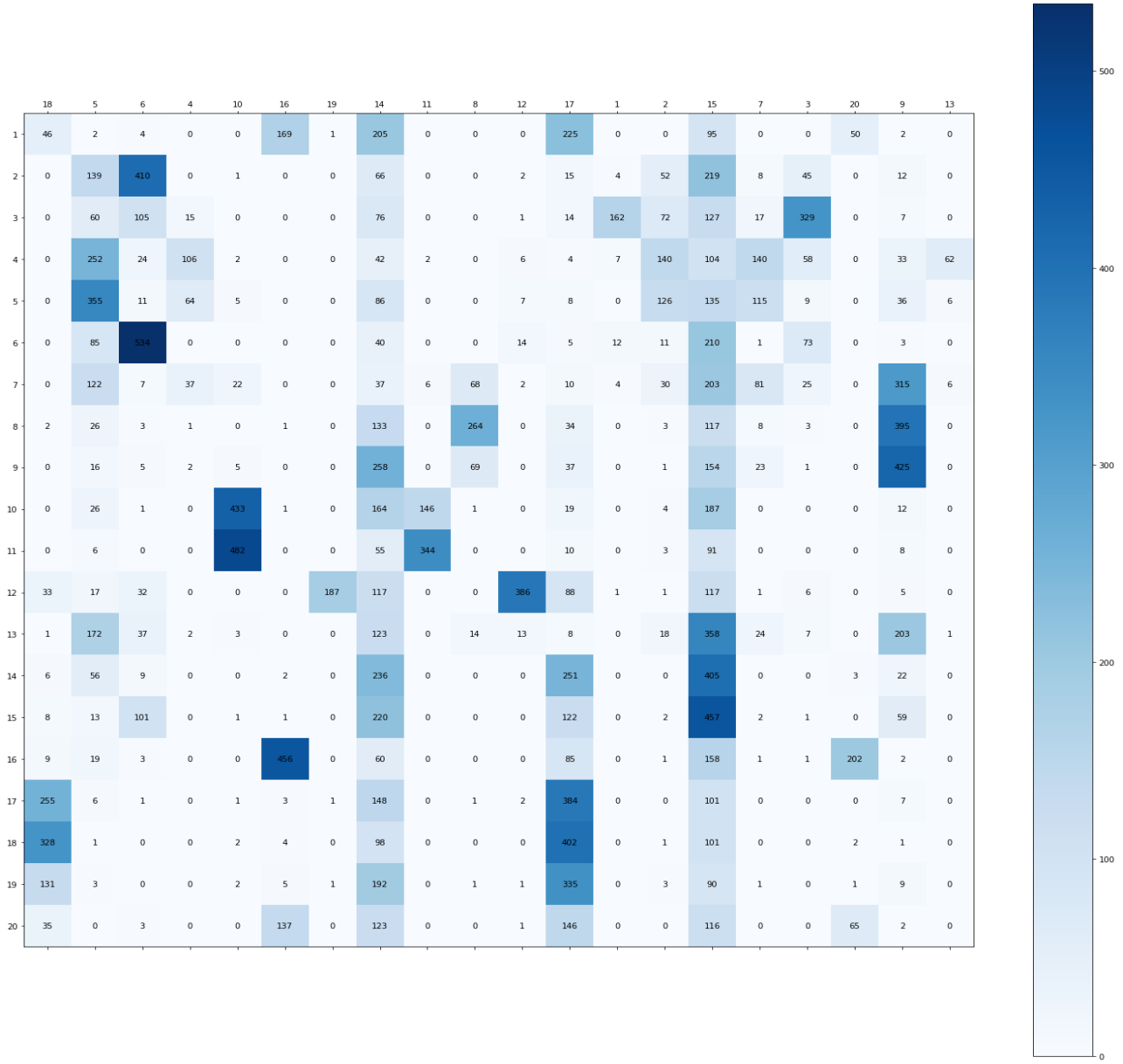

Fig11: Plot for NMF contingency matrix for r=10

The metrics for this NMF(r=10) with K-means clustering (k=20) are as follows:

- Homogeneity score: 0.332317
- Completeness score: 0.378952
- V-measure score: 0.354105
- Adjusted Rand Index score: 0.122884
- Adjusted mutual information score: 0.351868

The average value of metrics for NMF, r=10 and K-means clustering (k=20) are: 0.308025

QUESTION 11: Reduce the dimension of your dataset with UMAP. Consider the following settings: n components = [5, 20, 200], metric = "cosine" vs. "euclidean". Report the permuted contingency matrix and the five clustering evaluation metrics for the different combinations (6 combinations).

Answer 11:

If we choose UMAP for dimensionality reduction of our dataset, for r= [5, 20, 200], n_init=30,random_state=0 and max_iters=1000 , we get the below values for the clustering metrics for Euclidean and cosine distance metrics respectively:

Measure	Euclidean			Cosine		
	r=5	r=20	r=200	r=5	r=20	r=200
Homogeneity score	0.0132	0.0149	0.0135	0.5646	0.5464	0.5732
Completeness score	0.0143	0.0159	0.0145	0.5984	0.5841	0.5934
V-measure score	0.0137	0.0154	0.0140	0.5810	0.5646	0.5831
Adjusted Rand Index score	0.0021	0.0031	0.0031	0.4463	0.4233	0.4558
Adjusted mutual information score	0.0104	0.0122	0.0107	0.5796	0.5632	0.5817
Average Measure	**0.0108**	**0.0123**	**0.0112**	**0.5540**	**0.5363**	**0.5575**

We observe that the best average measure values of **UMAP are 0.0123 for r=20 with Euclidean distance and are 0.5575 for r=200 with Cosine distance.**

The results of the clustering metrics show that **UMAP with cosine distance is more effective** compared to Euclidean distance. This is due to the fact that it takes into consideration the angles between the sample points, instead of only their magnitudes, to cluster them. This helps solve the problem where higher frequency words dominate the clustering. Additionally, in high-dimensional spaces, Euclidean distances tend to converge to a constant value between all sample points because of sparsity in each dimension.

We can see the respective contingency matrix for all 6 combinations of r and distance metrics below:

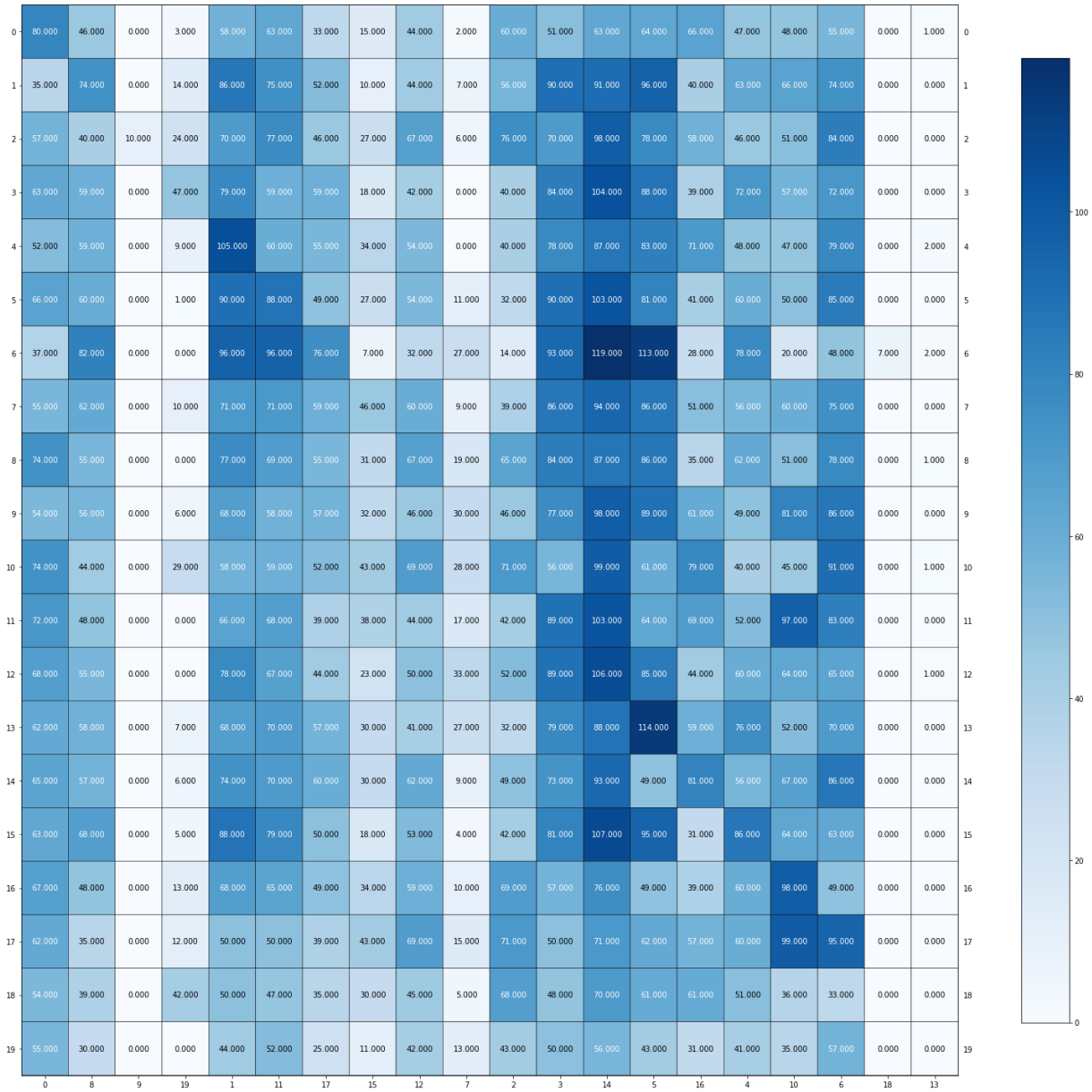

Fig12: Plot for UMAP contingency matrix for r=5, Euclidean distance

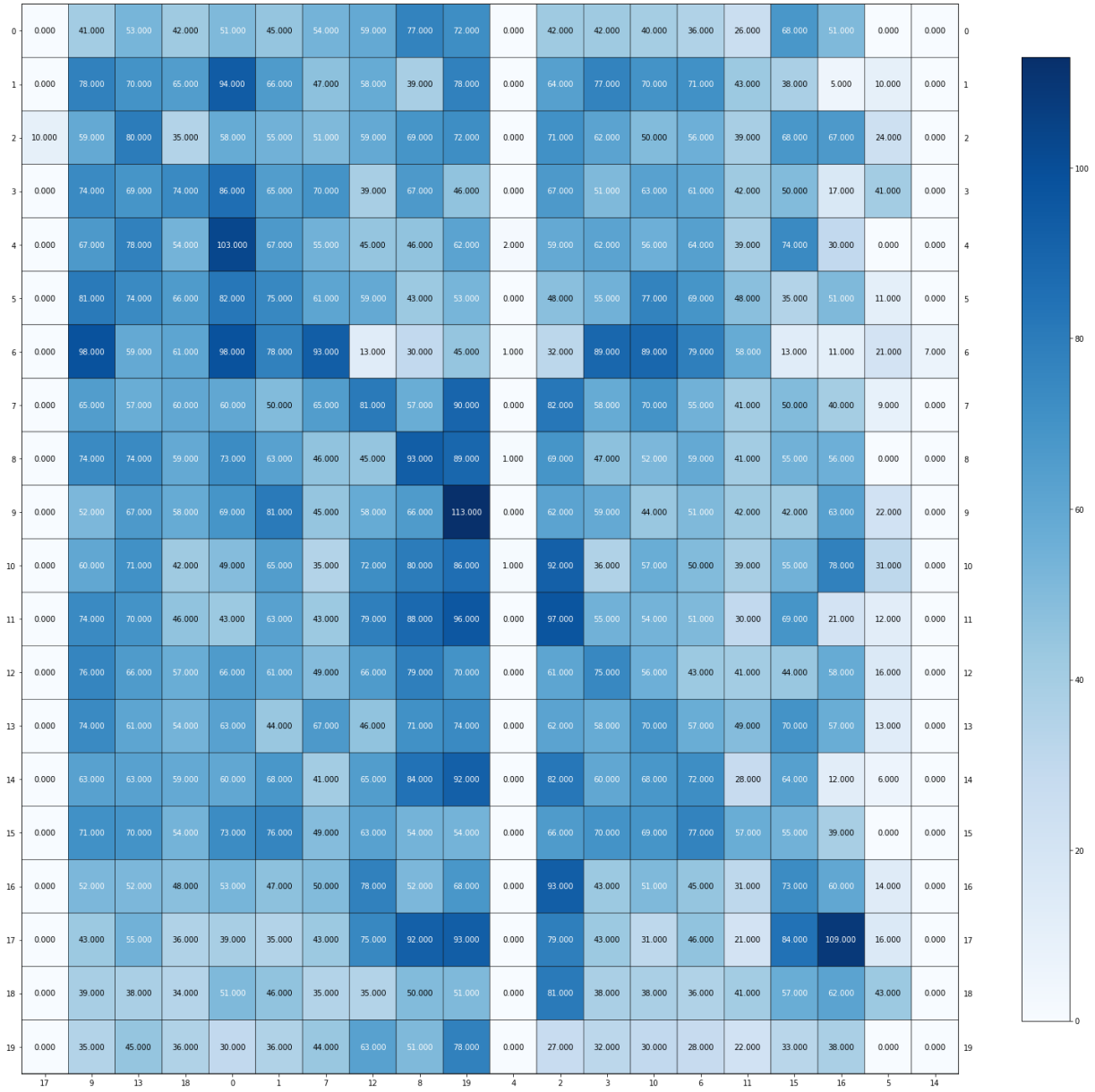

Fig13: Plot for UMAP contingency matrix for best r=20, Euclidean distance

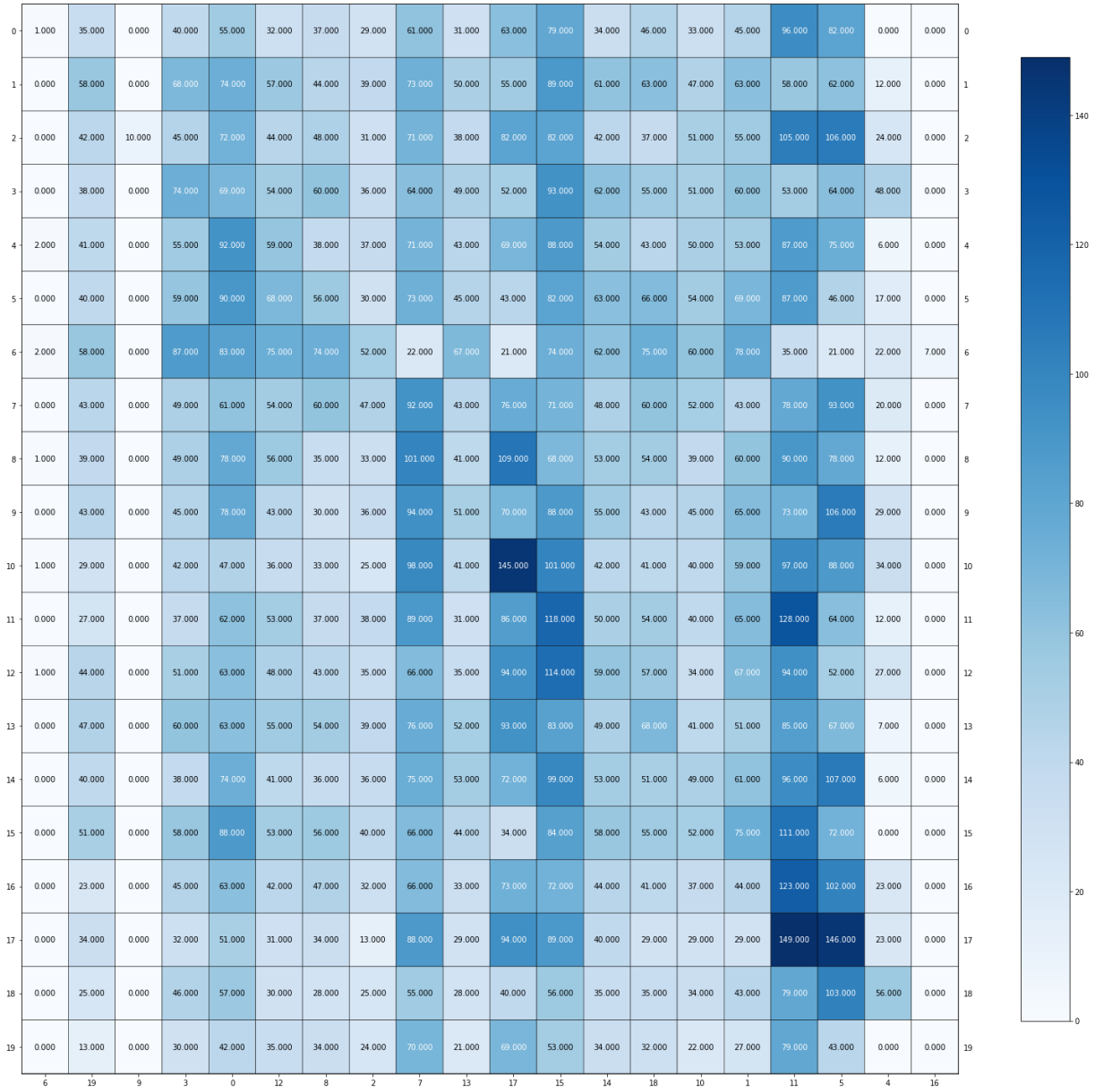

Fig14: Plot for UMAP contingency matrix for r=200, Euclidean distance

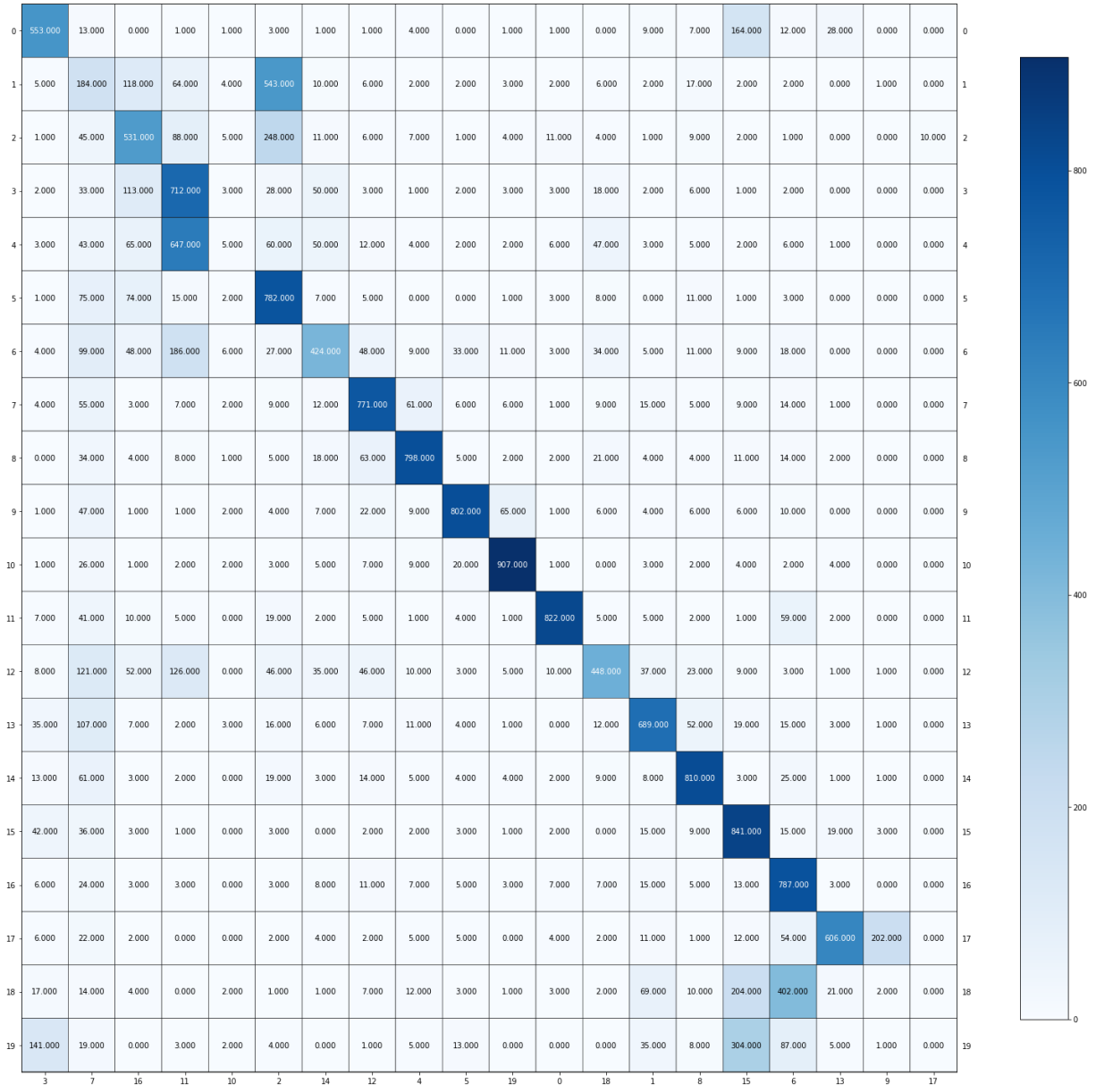

Fig15: Plot for UMAP contingency matrix for r=5, Cosine distance

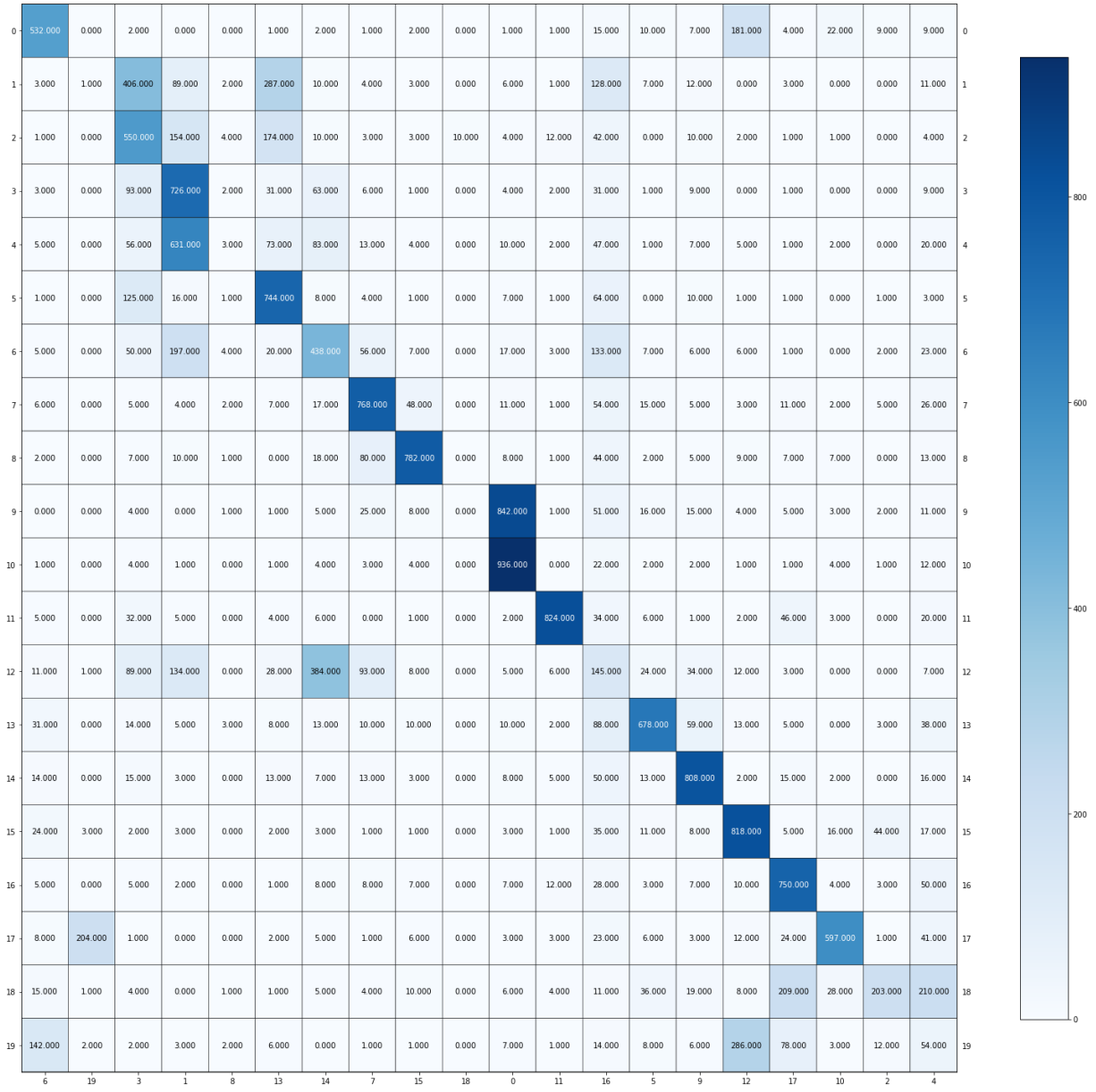

Fig16: Plot for UMAP contingency matrix for r=20, Cosine distance

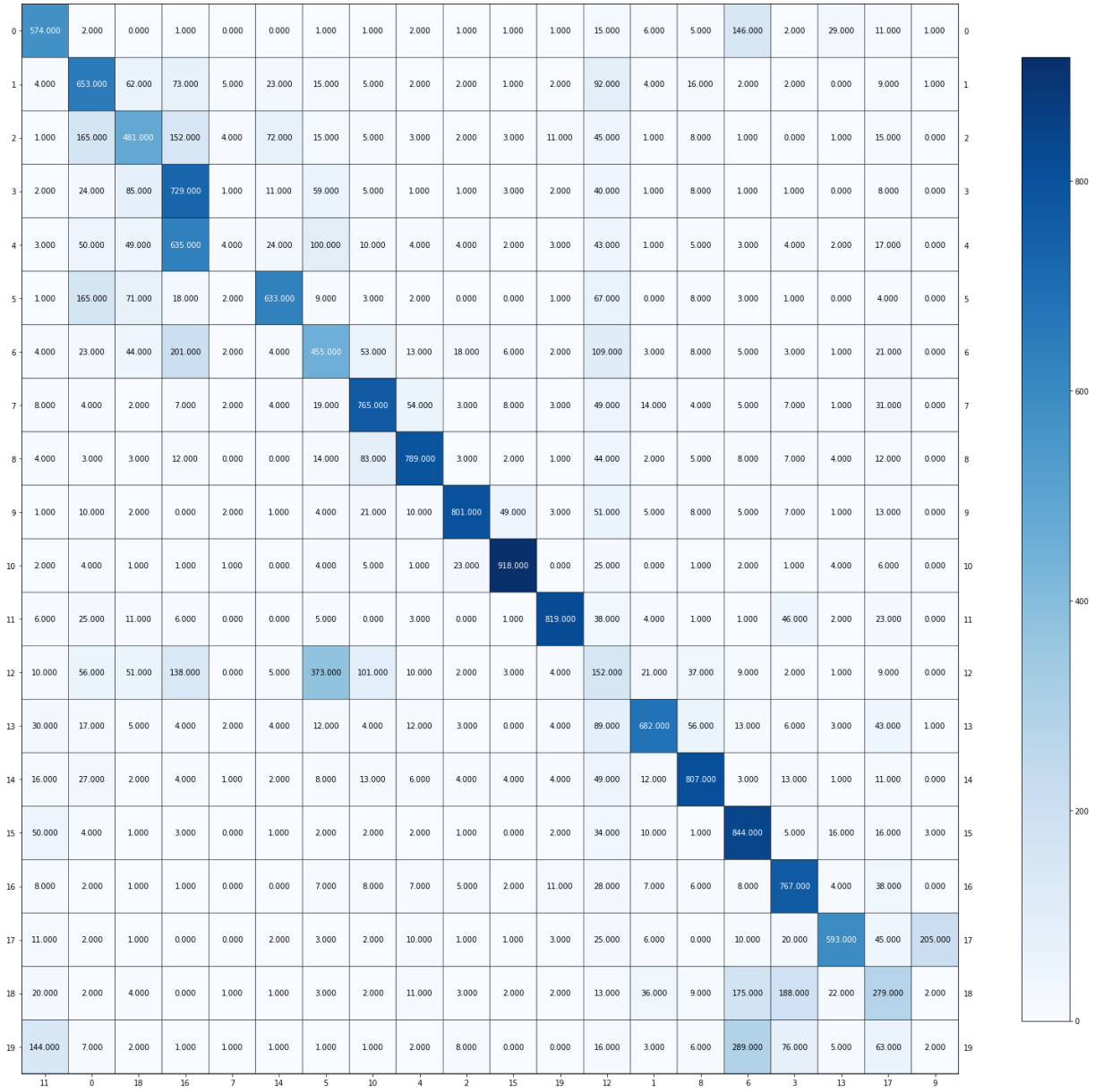

Fig17: Plot for UMAP contingency matrix for best r=200, Cosine distance

QUESTION 12: Analyse the contingency matrices. Which setting works best and why? What about for each metric choice?

Answer 12:

After examining both Euclidean and Cosine UMAP contingency tables, we can see that Cosine UMAP performs better. The analysis of the Euclidean UMAP's contingency table shows that the matrix has no clear structure and is not diagonal, lacking patterns and appearing to be random. For this reason, only the Cosine UMAP contingency matrix needs to be explained. The Cosine UMAP matrix has a good structure overall, as can be seen from its diagonal centroid matrix. However, closer examination reveals that some classes or clusters do not follow this pattern. This could be due to the use of common words across classes, which may have lost their semantic meaning after the removal of identifying information like headers and footers.

Further as previously described, the limitations of applying K-means on this dataset include its sensitivity to outliers and noisy samples which weakens its clustering ability. It operates based on the assumption of having well-defined clusters and centroids, and only works well with isotropic, globular, and convex clusters. It is not suitable for handling datasets with irregular shapes and disparate sizes.

QUESTION 13: So far, we have attempted K-Means clustering with 4 different representation learning techniques (sparse TF-IDF representation, PCA-reduced, NMF-reduced, and UMAP-reduced). Compare and contrast the clustering results across the 4 choices, and suggest an approach that is best for the K-Means clustering task on the 20-class text data. Choose any choice of clustering metrics for your comparison.

Answer 13:

If we compare the clustering metrics across the 4 different representation learning techniques (for the optimal r) we tried on the entire 20 groups' dataset, we get the following results:

Metric	TF-IDF Sparse	PCA/SVD, r=20	NMF , r=10	UMAP Cosine, r=200
Homogeneity score	0.326881	0.336158	0.332317	**0.5732**
Completeness score	0.374454	0.378021	0.378952	**0.5934**
V-measure score	0.349054	0.355862	0.354105	**0.5831**
Adjusted Rand Index score	0.114896	0.120619	0.122884	**0.4558**
Adjusted mutual information score	0.346801	0.353652	0.351868	**0.5817**
Average Measure	**0.302417**	**0.308862**	**0.308025**	**0.5575**

The findings from the comparison of clustering metrics above can be summarised as:

• The best results were obtained using UMAP, with an Average Measures Score of ~56%. This is due to the fact that UMAP preserves semantic and non-linear dependencies, in addition to the overall structure of the data. It clusters similar embedding together and keeps different ones far apart, improving the clustering outcome. UMAP also performs well with larger sample sizes and higher dimensions.

• In comparison to PCA and NMF, UMAP is a local-density based dimensionality reduction method that allows for a more flexible interpretation and works well even if the data lacks prior distributions. PCA and NMF, on the other hand, treat all clusters as a whole and may lose local structure.

• SVD and NMF provided similar results, with SVD having a slightly higher Average Measure score of 30.88% compared to NMF's 30.80%. This could be due to SVD allowing for negative entries in its reduced-rank feature matrix, allowing for a more accurate representation of higher dimensional feature matrices. SVD is also more deterministic and considers the geometry in the feature space, important for clustering in higher dimensions.

• The least Average Measure score was obtained using sparse representation, with a score of 30.2%. This is due to the large and sparse nature of the dataset in 20 classes, causing a lot of noise that makes it difficult for the k-means algorithm to accurately classify the data points.

Hence, we can conclude that the **UMAP reduction technique with cosine distance metric** for r=200 is the best representation learning technique for K-means clustering task on the 20 class newsgroup dataset we are provided.

Clustering Algorithms that do not explicitly rely on the Gaussian distribution per cluster:

QUESTION 14: Use UMAP to reduce the dimensionality properly, and perform Agglomerative clustering with n_clusters=20. Compare the performance of "ward" and "single" linkage criteria. Report the five clustering evaluation metrics for each case.

Answer 14:

If we choose UMAP with cosine distance metric for dimensionality reduction of our dataset, for r= [5, 20, 200], and then apply Agglomerative clustering with 20 clusters, we get the below values for the clustering metrics for the ward and single linking respectively:

Measure	Ward linking			Single linking		
	r=5	r=20	r=200	r=5	r=20	r=200
Homogeneity score	0.5514	**0.5562**	0.5581	0.01688	0.0184	**0.0195**
Completeness score	0.5883	**0.5895**	0.5841	0.3507	0.3551	**0.3644**
V-measure score	0.5693	**0.5724**	0.5708	0.03203	0.0351	**0.0369**
Adjusted Rand Index score	0.4180	**0.4290**	0.4274	0.00058	0.0005	**0.0005**
Adjusted mutual information score	0.5678	**0.5709**	0.5694	0.02735	0.0301	**0.0317**
Average Measure	**0.5390**	**0.5436**	**0.5420**	**0.0855**	**0.0878**	**0.0906**

We observe that the best average measure values of **Agglomerative Clustering are 0.5436 for r=20 with Ward linking and are 0.0906 for r=200 with Single linking.** Overall **Agglomerative Clustering with Ward linkage performs better** than the single linkage clustering.

We can see the respective contingency matrix for all 6 combinations of r and distance metrics below:

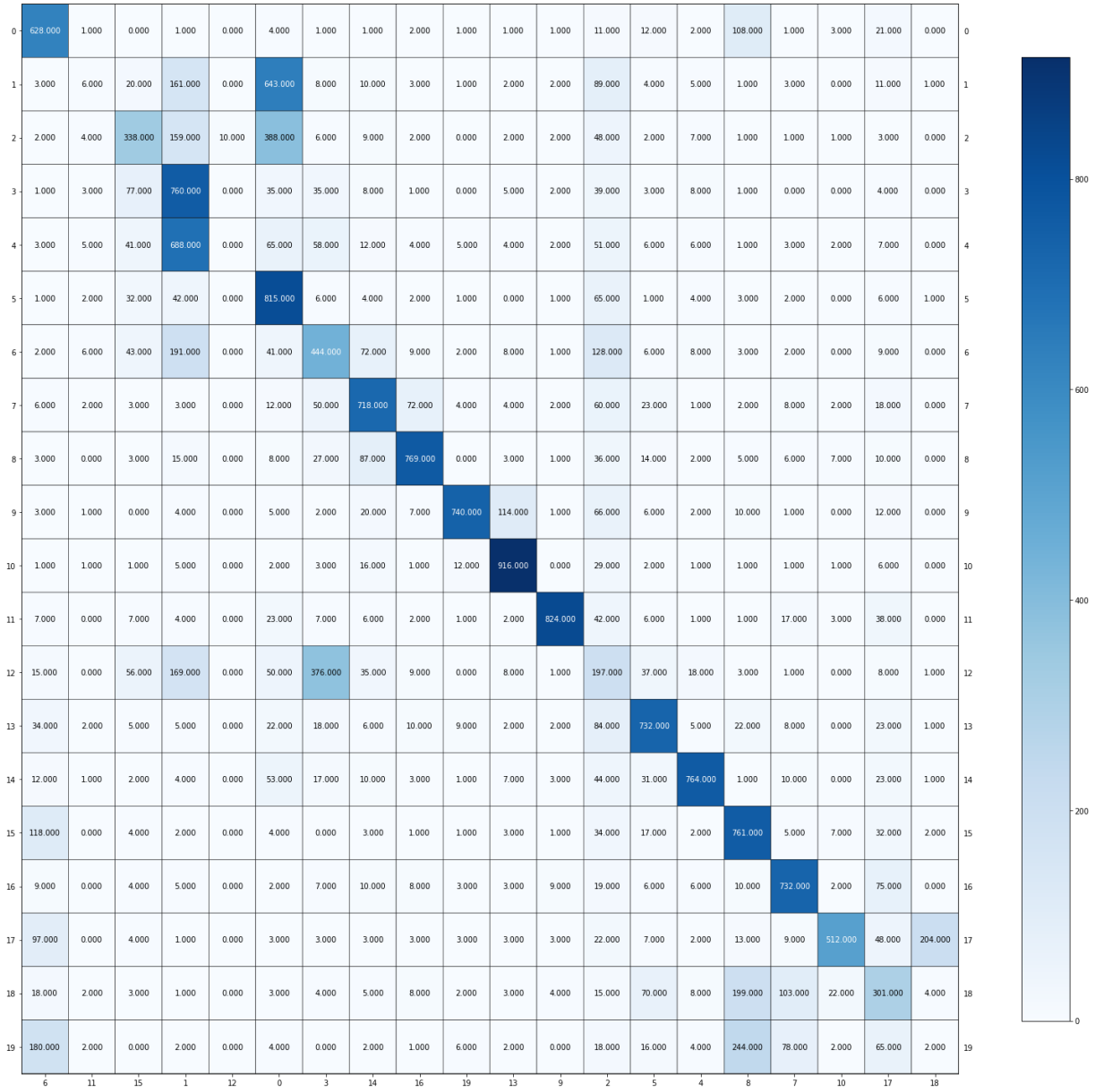

Fig18: Plot for Agglomerative clustering contingency matrix for r=5, Ward linking

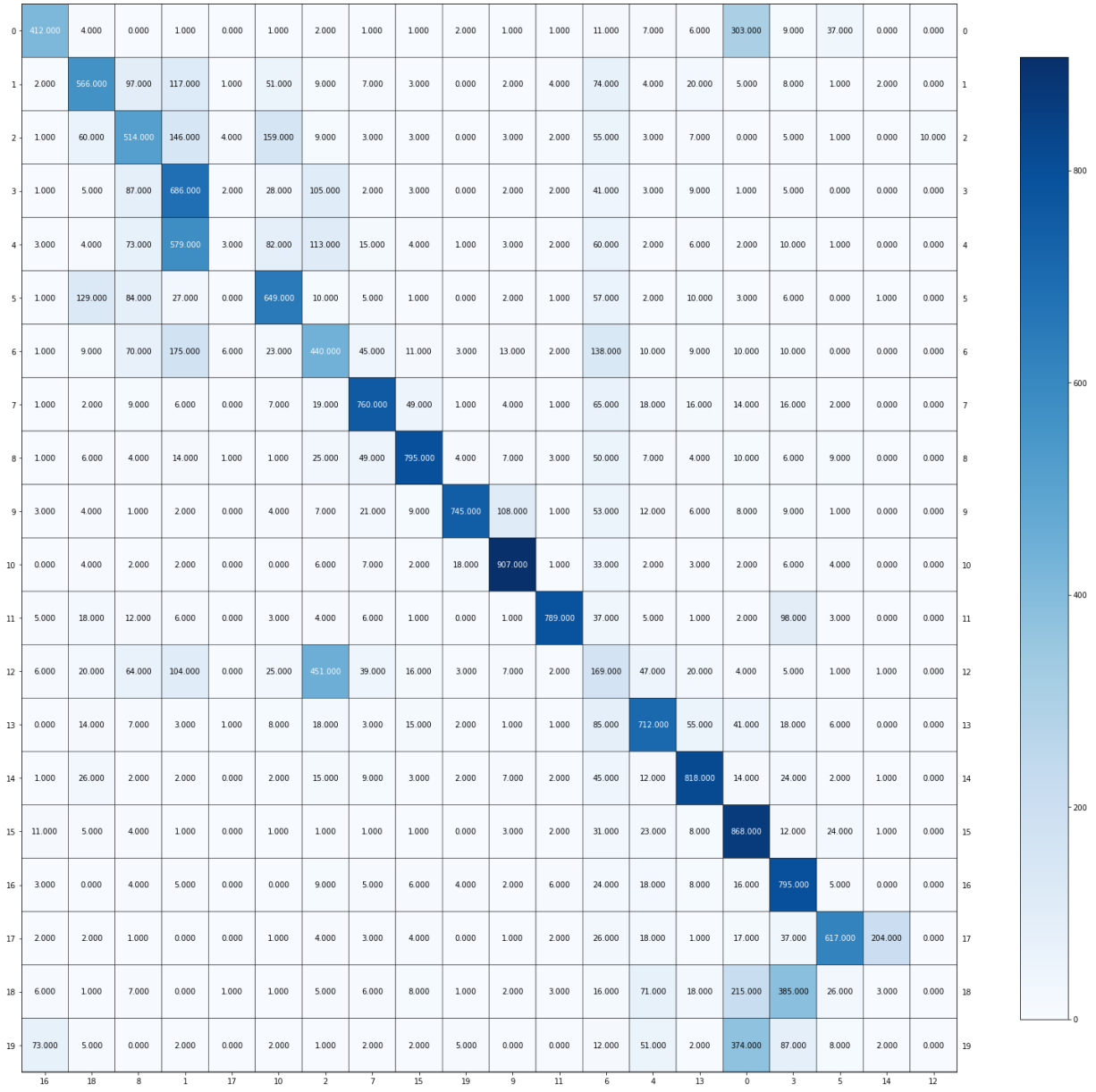

Fig19: Plot for Agglomerative clustering contingency matrix for best r=20, Ward linking

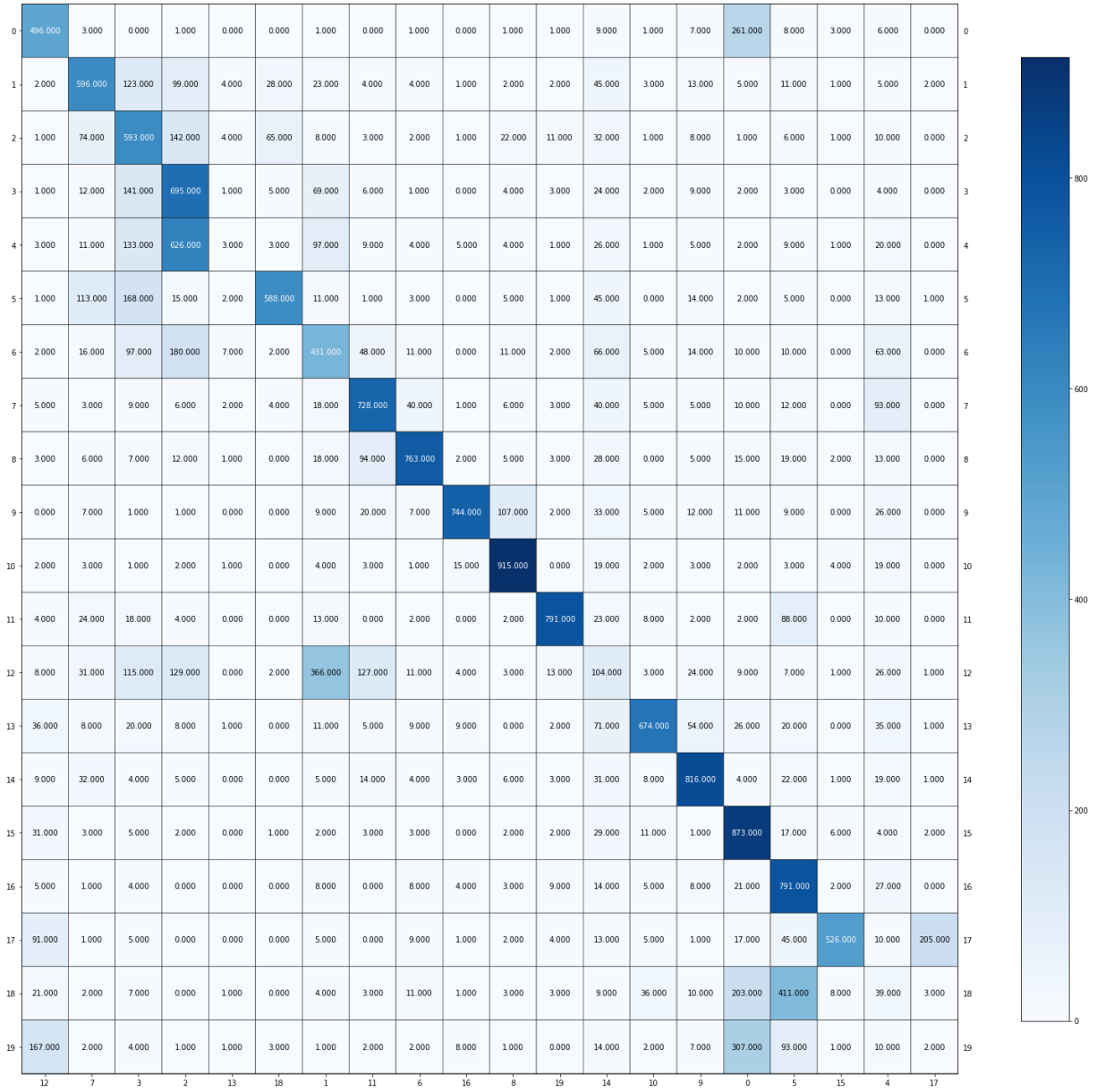

Fig20: Plot for Agglomerative clustering contingency matrix for r=200, Ward linking

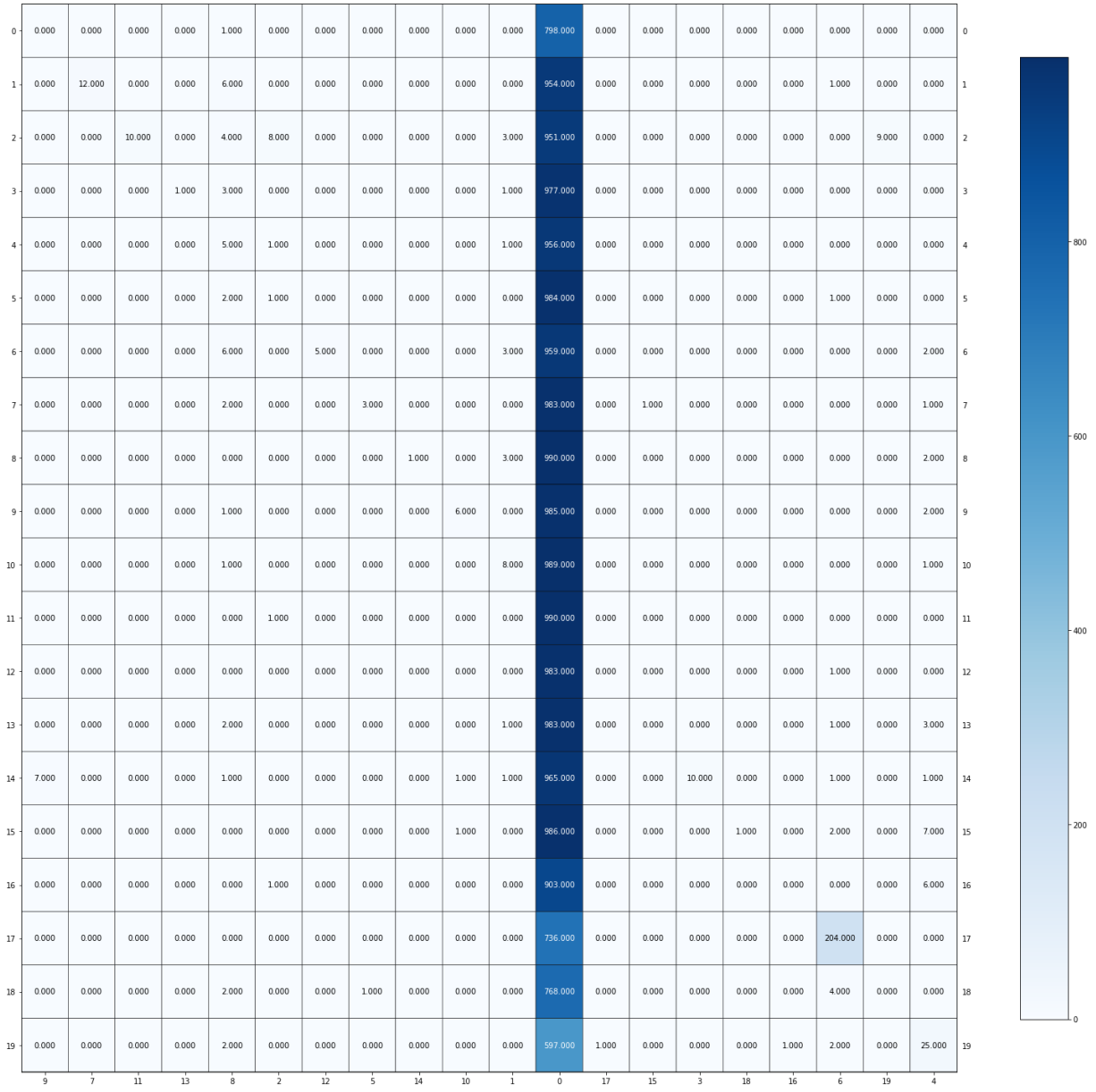

Fig21: Plot for Agglomerative clustering contingency matrix for r=5, Single linking

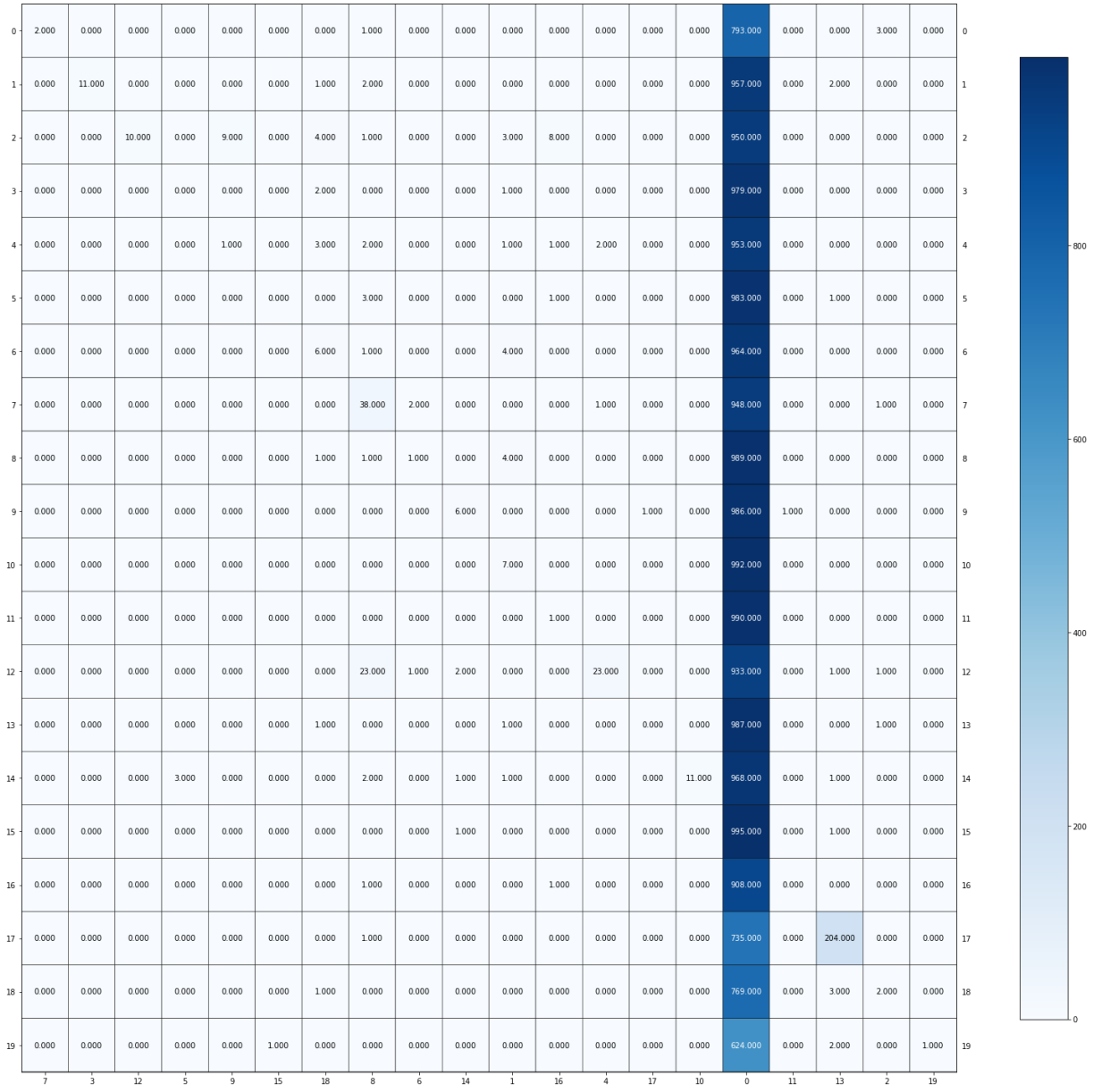

Fig22: Plot for Agglomerative clustering contingency matrix for r=20, Single linking

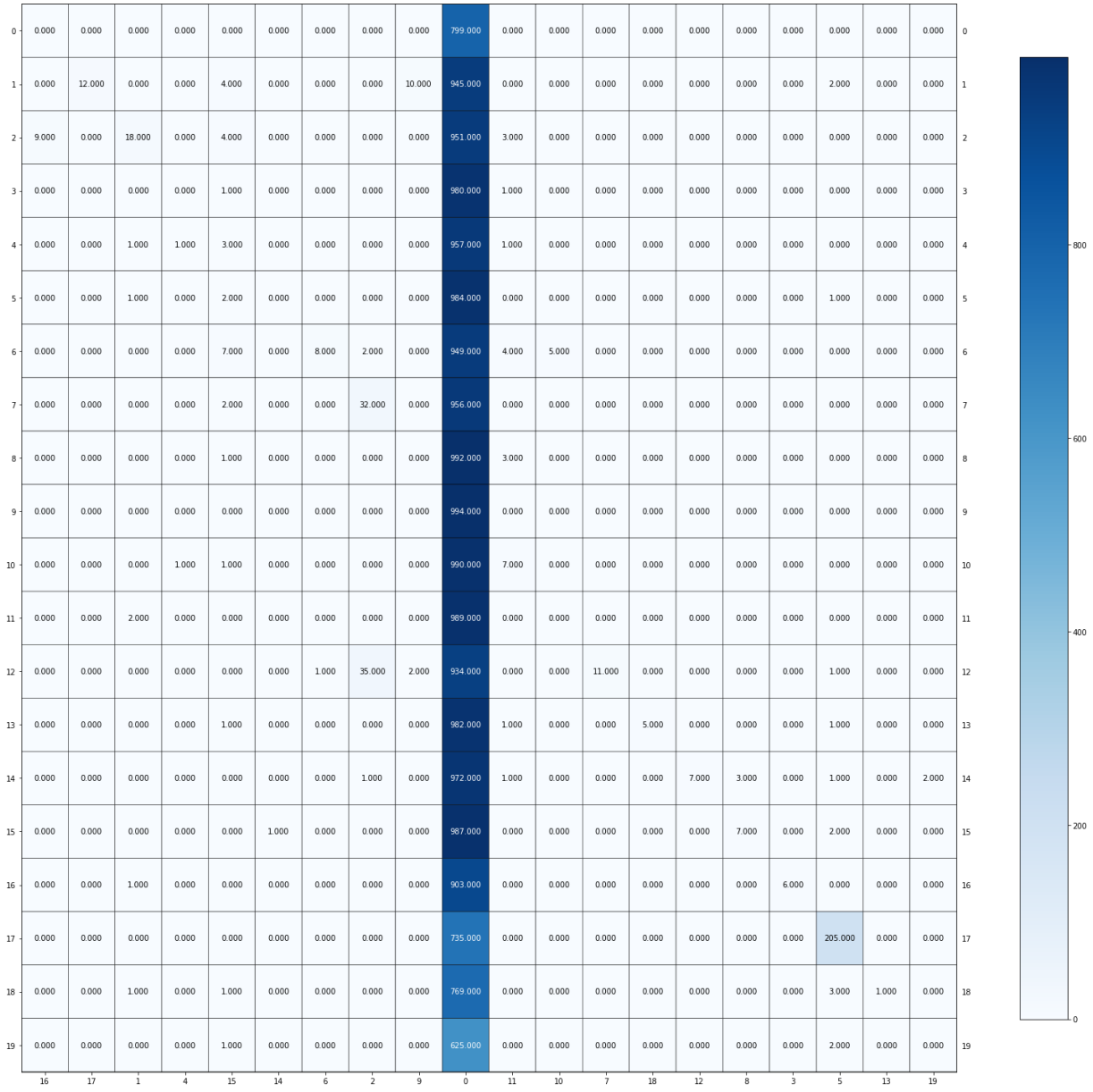

Fig23: Plot for Agglomerative clustering contingency matrix for best r=200, Single linking

Comparing Ward and Single linkage clustering, we find that Ward consistently performs better in terms of Average score and Adjusted Rand Index Score. Ward's clustering is a bottom-up method, starting with all documents as separate clusters and merging them based on a minimum variance criterion. Single linkage clustering merges clusters based on the closest pair of elements, but this can result in the formation of long chains and merging of groups too soon even if they are overall dissimilar. Ward's method minimizes the total within-cluster variance, whereas single linkage can result in a large cluster with documents far away from it categorized as different clusters.

QUESTION 15: Apply HDBSCAN on UMAP-transformed 20-category data. Use min_cluster_size=100. Vary the min cluster size among 20, 100, 200 and report your findings in terms of the five clustering evaluation metrics - you will plot the best contingency matrix in the next question. Feel free to try modifying other parameters in HDBSCAN to get better performance.

Answer 15:

We use UMAP with cosine distance metric on varying no of components (r) = [5,20,200] for dimensionality reduction and then we applied HDBSCAN for clustering on it with min cluster size varying between 20,100,200. Below are the 5 clustering evaluation metrics for all these 9 combinations:

1. UMAP (cosine, r=5) and HDBSCAN (min_cluster_size=20)
   - Homogeneity score: 0.00041764777356824743
   - Completeness score: 0.11934206294879736
   - V-measure score: 0.0008323825530800732
   - Adjusted Rand Index score: 6.011761711987771e-07
   - Adjusted mutual information score: 0.00043861800508464093

2. UMAP (cosine, r=20) and HDBSCAN (min_cluster_size=20)
   - Homogeneity score: 0.4297813947777303
   - Completeness score: 0.44716649686575993
   - V-measure score: 0.4383016198617358
   - Adjusted Rand Index score: 0.0807420958518301
   - Adjusted mutual information score: 0.42644275117884267

3. UMAP (cosine, r=200) and HDBSCAN (min_cluster_size=20)
   - Homogeneity score: 0.42408328243046084
   - Completeness score: 0.4412286913936648
   - V-measure score: 0.43248612618124666
   - Adjusted Rand Index score: 0.08152401654283277
   - Adjusted mutual information score: 0.4203707052895268

4. UMAP (cosine, r=5) and HDBSCAN (min_cluster_size=100)
   - Homogeneity score: 0.41296355343513375
   - Completeness score: 0.6203796936781066
   - V-measure score: 0.4958549900935972
   - Adjusted Rand Index score: 0.22523767868786387
   - Adjusted mutual information score: 0.49493053223241285

5. UMAP (cosine, r=20) and HDBSCAN (min_cluster_size=100)
   - Homogeneity score: 0.40849591114439354
   - Completeness score: 0.6191939167581709
   - V-measure score: 0.4922461550824574
   - Adjusted Rand Index score: 0.21827259060357856
   - Adjusted mutual information score: 0.4913116175147385

6. UMAP (cosine, r=200) and HDBSCAN (min_cluster_size=100)
   - Homogeneity score: 0.39328443415928427
   - Completeness score: 0.602606657757921
   - V-measure score: 0.4759472603790383
   - Adjusted Rand Index score: 0.19794497630685662
   - Adjusted mutual information score: 0.4748680806205346

7. UMAP (cosine, r=5) and HDBSCAN (min_cluster_size=200)
   - Homogeneity score: 0.41039046347344543
   - Completeness score: 0.6046303978259363
   - V-measure score: 0.48892502342517286
   - Adjusted Rand Index score: 0.2068320492723218
   - Adjusted mutual information score: 0.48789111970680654

8. UMAP (cosine, r=20) and HDBSCAN (min_cluster_size=200)
   - Homogeneity score: 0.41631182267323186
   - Completeness score: 0.6144277947444438
   - V-measure score: 0.4963301124914484
   - Adjusted Rand Index score: 0.21212907780591406
   - Adjusted mutual information score: 0.49531044642121097

9. UMAP (cosine, r=200) and HDBSCAN (min_cluster_size=200)
    - Homogeneity score: 0.4168883622503015
    - Completeness score: 0.6160190931329017
    - V-measure score: 0.4972588580180511
    - Adjusted Rand Index score: 0.21300598956646372
    - Adjusted mutual information score: 0.49624055108855136

From these 9 combinations, when we compare the average of the 5 evaluation metrics for each, we obtain that the **best out of these is when UMAP with cosine distance has 5 components and HDBSCAN has 100 min size of clusters,** for which average score = 0.44987328962542283 **.**

QUESTION 16: Contingency matrix: Plot the contingency matrix for the best clustering model from Question 15. How many clusters are given by the model? What does "-1" mean for the clustering labels? Interpret the contingency matrix considering the answer to these questions.

Answer 16:

Based on Q15, we found that the best clustering model is UMAP cosine with r=5 and HDBSCAN with min_clusters=100 having an average score of 0.44987328962542283 for the 5 clustering metrics. We plot its contingency matrix as shown below.

From the contingency matrix we see that the model has given **10 major clusters** (the count of columns which are not all zero in the matrix), without including the outliers.

The **-1** in the clustering labels signifies samples that have not been assigned to any cluster by the algorithms. This could be due to **outliers or noisy data**.

The contingency matrix shows that most of the columns are empty because there are fewer clusters than the actual number of classes in the dataset (20). This results in a smaller prominent diagonal and one of the columns being the cluster of outliers. The low metrics for HDBSCAN are due to the lower number of clusters compared to the actual number of classes. Unlike k-means or Agglomerative Clustering, the number of clusters is not specified, which may result in fewer or more clusters than expected. The poor performance in the Adjusted Rand Index Score is due to considering only pairwise elements in the clusters to calculate accuracy.

The clustering results might be affected by hyper-parameter sensitivity and the use of excessive smoothing, leading to the loss of low-density clusters. HDBSCAN struggles to identify clusters in high dimensions if they are sparse or have wild variations, which is common in textual data. We experimented with the minimum cluster size hyper-parameter and see increasing it to around 100 results in the formation of more micro clusters for low-density classes.

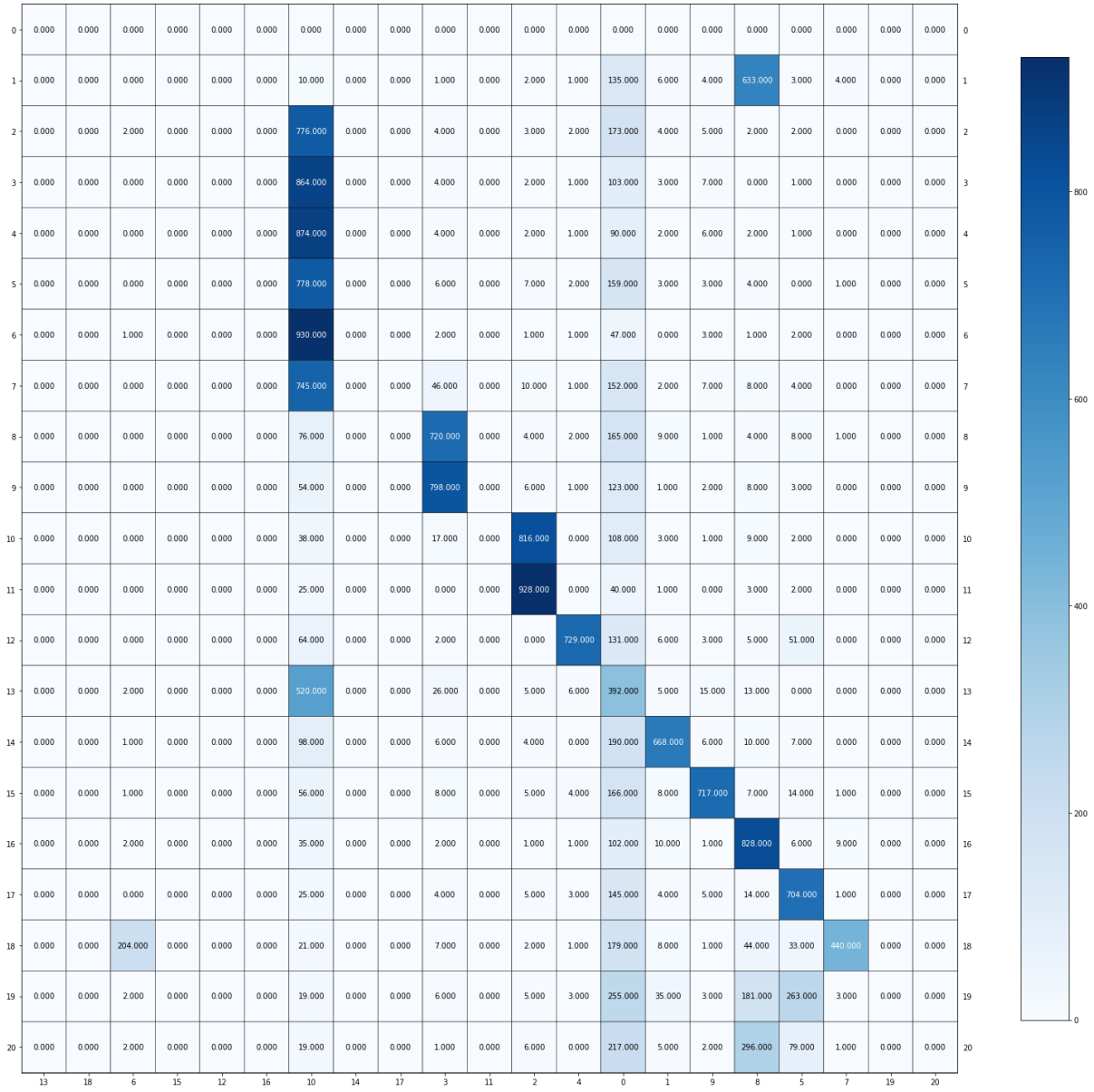

Fig24: Plot for HDBSCAN-UMAP clustering contingency matrix for best min cluster size=100, r=5

QUESTION 17: Based on your experiments, which dimensionality reduction technique and clustering methods worked best together for 20-class text data and why? Follow the table below.

Module	Alternatives	Hyperparameters
Dimensionality Reduction	None	N/A
	SVD	r = [5,20,200]
	NMF	r = [5,20,200]
	UMAP	n_components = [5,20,200]
Clustering	K-Means	k = [10,20,50]
	Agglomerative Clustering	n_clusters = [20]
	HDBSCAN	min_cluster_size = [100,200]

Answer 17:

Based on the experiments we conducted for various dimensionality reduction and clustering techniques using the table above, we can report the 5 clustering evaluation metrics for all as follows:

Dimensionality Reduction: SVD and Clustering: K-Means

- Kmeans Cluster_Size 10, SVD Component Numbers 5
    - Homogeneity score for K-Means number of clusters: 10 and number of components for SVD r: 5 is: 0.270789611796396
    - Completeness score for K-Means number of clusters: 10 and number of components for SVD r: 5 is: 0.417480258233419
    - V-measure score for for K-Means number of clusters: 10 and number of components for SVD r: 5 is: 0.3285028794149874
    - Adjusted Rand Index score for K-Means number of clusters: 10 and number of components for SVD r: 5 is: 0.10607246793487007
    - Adjusted mutual information score for K-Means number of clusters: 10 and number of components for SVD r: 5 is: 0.3272589720408019

- Kmeans Cluster_Size 10, SVD Component Numbers 20
    - Homogeneity score for K-Means number of clusters: 10 and number of components for SVD r: 20 is: 0.2558179157227848
    - Completeness score for K-Means number of clusters: 10 and number of components for SVD r: 20 is: 0.40490810506537567
    - V-measure score for for K-Means number of clusters: 10 and number of components for SVD r: 20 is: 0.31354220732377386
    - Adjusted Rand Index score for K-Means number of clusters: 10 and number of components for SVD r: 20 is: 0.09248102379965287
    - Adjusted mutual information score for K-Means number of clusters: 10 and number of components for SVD r: 20 is: 0.31225721825289565

- Kmeans Cluster_Size 10, SVD Component Numbers 200
    - Homogeneity score for K-Means number of clusters: 10 and number of components for SVD r: 200 is: 0.2731128001964655
    - Completeness score for K-Means number of clusters: 10 and number of components for SVD r: 200 is: 0.4980928072344627
    - V-measure score for for K-Means number of clusters: 10 and number of components for SVD r: 200 is: 0.35278664996923337
    - Adjusted Rand Index score for K-Means number of clusters: 10 and number of components for SVD r: 200 is: 0.08913660794542383
    - Adjusted mutual information score for K-Means number of clusters: 10 and number of components for SVD r: 200 is: 0.35150178115745556

- Kmeans Cluster_Size 20, SVD Component Numbers 5
    - Homogeneity score for K-Means number of clusters: 20 and number of components for SVD r: 5 is: 0.32208368615323685
    - Completeness score for K-Means number of clusters: 20 and number of components for SVD r: 5 is: 0.3496612217280083
    - V-measure score for for K-Means number of clusters: 20 and number of components for SVD r: 5 is: 0.33530637561278187
    - Adjusted Rand Index score for K-Means number of clusters: 20 and number of components for SVD r: 5 is: 0.1264834813291519
    - Adjusted mutual information score for K-Means number of clusters: 20 and number of components for SVD r: 5 is: 0.33306505746576154

- Kmeans Cluster_Size 20, SVD Component Numbers 20
    - Homogeneity score for K-Means number of clusters: 20 and number of components for SVD r: 20 is: 0.33615762894770923

    - Completeness score for K-Means number of clusters: 20 and number of components for SVD r: 20 is: 0.3780205265079458
    - V-measure score for for K-Means number of clusters: 20 and number of components for SVD r: 20 is: 0.35586214143836575
    - Adjusted Rand Index score for K-Means number of clusters: 20 and number of components for SVD r: 20 is: 0.12061901723196018
    - Adjusted mutual information score for K-Means number of clusters: 20 and number of components for SVD r: 20 is: 0.3536521768005214

- Kmeans Cluster_Size 20, SVD Component Numbers 200
    - Homogeneity score for K-Means number of clusters: 20 and number of components for SVD r: 200 is: 0.33851033351914606
    - Completeness score for K-Means number of clusters: 20 and number of components for SVD r: 200 is: 0.41641595173795165
    - V-measure score for for K-Means number of clusters: 20 and number of components for SVD r: 200 is: 0.37344335588341826
    - Adjusted Rand Index score for K-Means number of clusters: 20 and number of components for SVD r: 200 is: 0.11314406691084686
    - Adjusted mutual information score for K-Means number of clusters: 20 and number of components for SVD r: 200 is: 0.37118319146184625

- Kmeans Cluster_Size 50, SVD Component Numbers 5
    - Homogeneity score for K-Means number of clusters: 50 and number of components for SVD r: 5 is: 0.37195797979002765
    - Completeness score for K-Means number of clusters: 50 and number of components for SVD r: 5 is: 0.30025158501507176
    - V-measure score for for K-Means number of clusters: 50 and number of components for SVD r: 5 is: 0.3322802258052982
    - Adjusted Rand Index score for K-Means number of clusters: 50 and number of components for SVD r: 5 is: 0.09956087221440743
    - Adjusted mutual information score for K-Means number of clusters: 50 and number of components for SVD r: 5 is: 0.32720064216238487

- Kmeans Cluster_Size 50, SVD Component Numbers 20
    - Homogeneity score for K-Means number of clusters: 50 and number of components for SVD r: 20 is: 0.4183223813981078
    - Completeness score for K-Means number of clusters: 50 and number of components for SVD r: 20 is: 0.34961423360247657
    - V-measure score for for K-Means number of clusters: 50 and number of components for SVD r: 20 is: 0.3808946100874512
    - Adjusted Rand Index score for K-Means number of clusters: 50 and number of components for SVD r: 20 is: 0.13410828162695695
    - Adjusted mutual information score for K-Means number of clusters: 50 and number of components for SVD r: 20 is: 0.3760753126138631

- Kmeans Cluster_Size 50, SVD Component Numbers 200
    - Homogeneity score for K-Means number of clusters: 50 and number of components for SVD r: 200 is: 0.42027012154807813
    - Completeness score for K-Means number of clusters: 50 and number of components for SVD r: 200 is: 0.37723379349214203
    - V-measure score for for K-Means number of clusters: 50 and number of components for SVD r: 200 is: 0.39759075599018107
    - Adjusted Rand Index score for K-Means number of clusters: 50 and number of components for SVD r: 200 is: 0.12071839945173347
    - Adjusted mutual information score for K-Means number of clusters: 50 and number of components for SVD r: 200 is: 0.3926866710565535

- **Out of these combinations, the best model is K-means(number of clusters =50) and SVD (r=200) with average score= 0.34169994830773764**

  The contingency matrix for this best combination of K-means SVD is:

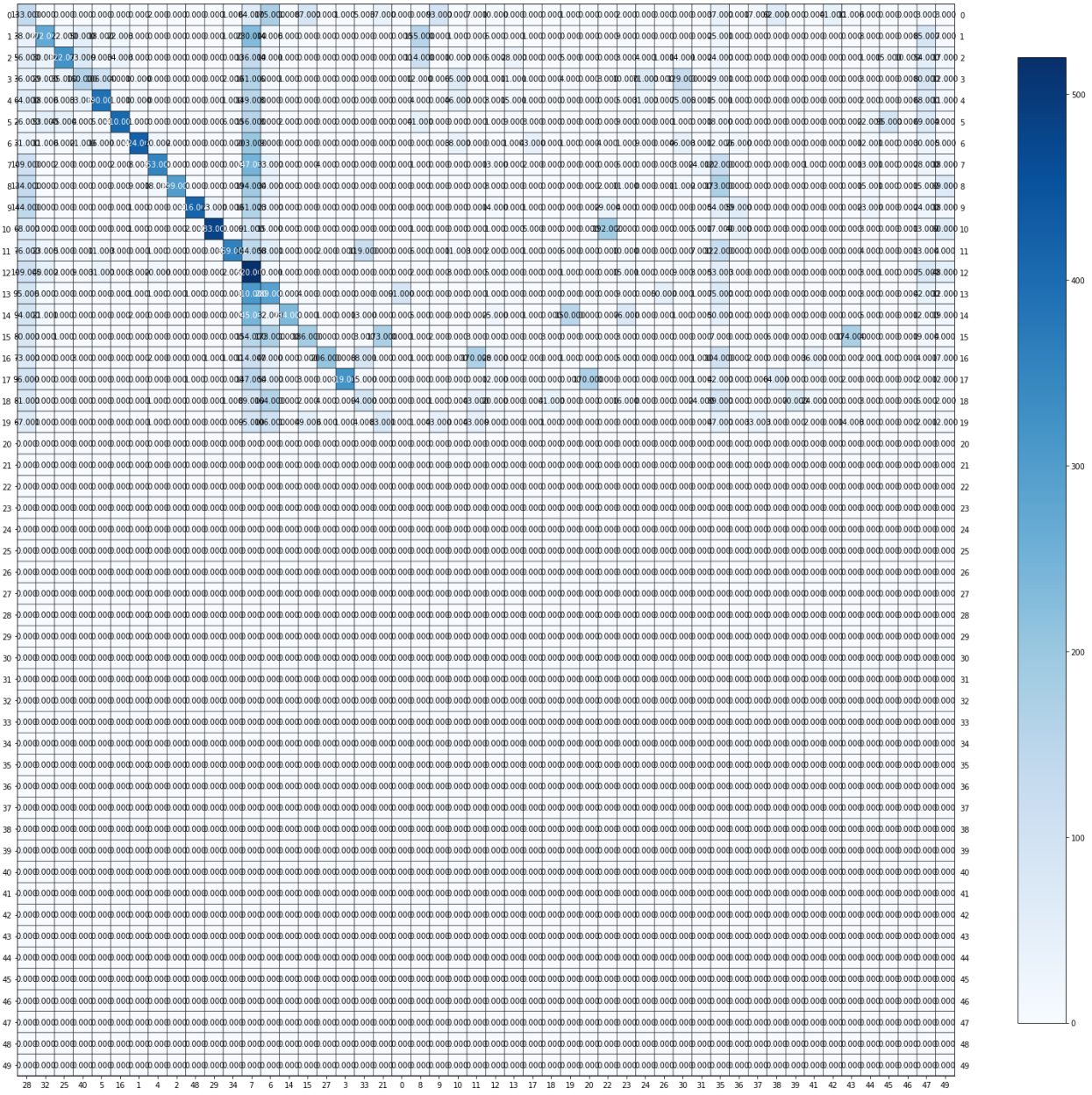

Fig25: Plot for SVD K-means clustering contingency matrix for best n_size=50, r=200

<u>Dimensionality Reduction: NMF and Clustering: K-Means</u>

- Kmeans Cluster_Size 10, NMF Component Numbers 5
  - Homogeneity score for K-Means number of clusters: 10 and number of components for NMF r: 5 is: 0.23480940679752516
  - Completeness score for K-Means number of clusters: 10 and number of components for NMF r: 5 is: 0.3512631744779117

    - V-measure score for for K-Means number of clusters: 10 and number of components for NMF r: 5 is: 0.28146649498421394
    - Adjusted Rand Index score for K-Means number of clusters: 10 and number of components for NMF r: 5 is: 0.09323405771724347
    - Adjusted mutual information score for K-Means number of clusters: 10 and number of components for NMF r: 5 is: 0.28015160205683987

- Kmeans Cluster_Size 10, NMF Component Numbers 20
    - Homogeneity score for K-Means number of clusters: 10 and number of components for NMF r: 20 is: 0.25861185216610005
    - Completeness score for K-Means number of clusters: 10 and number of components for NMF r: 20 is: 0.47679386487021275
    - V-measure score for for K-Means number of clusters: 10 and number of components for NMF r: 20 is: 0.33533746512724055
    - Adjusted Rand Index score for K-Means number of clusters: 10 and number of components for NMF r: 20 is: 0.06169374479407774
    - Adjusted mutual information score for K-Means number of clusters: 10 and number of components for NMF r: 20 is: 0.3340199977046619

- Kmeans Cluster_Size 10, NMF Component Numbers 200
    - Homogeneity score for K-Means number of clusters: 10 and number of components for NMF r: 200 is: 0.02671335785231646
    - Completeness score for K-Means number of clusters: 10 and number of components for NMF r: 200 is: 0.050316463415388306
    - V-measure score for for K-Means number of clusters: 10 and number of components for NMF r: 200 is: 0.034898735865087314
    - Adjusted Rand Index score for K-Means number of clusters: 10 and number of components for NMF r: 200 is: 0.005047113879916799
    - Adjusted mutual information score for K-Means number of clusters: 10 and number of components for NMF r: 200 is: 0.03294147623738231

- Kmeans Cluster_Size 20, NMF Component Numbers 5
    - Homogeneity score for K-Means number of clusters: 20 and number of components for NMF r: 5 is: 0.2651214476733179
    - Completeness score for K-Means number of clusters: 20 and number of components for NMF r: 5 is: 0.28696497684952593
    - V-measure score for for K-Means number of clusters: 20 and number of components for NMF r: 5 is: 0.27561108809962587
    - Adjusted Rand Index score for K-Means number of clusters: 20 and number of components for NMF r: 5 is: 0.09802369182001616
    - Adjusted mutual information score for K-Means number of clusters: 20 and number of components for NMF r: 5 is: 0.2731696870546707

- Kmeans Cluster_Size 20, NMF Component Numbers 20
    - Homogeneity score for K-Means number of clusters: 20 and number of components for NMF r: 20 is: 0.32845496046568884
    - Completeness score for K-Means number of clusters: 20 and number of components for NMF r: 20 is: 0.39247138018569905
    - V-measure score for for K-Means number of clusters: 20 and number of components for NMF r: 20 is: 0.35762092295402365
    - Adjusted Rand Index score for K-Means number of clusters: 20 and number of components for NMF r: 20 is: 0.09597919718772172
    - Adjusted mutual information score for K-Means number of clusters: 20 and number of components for NMF r: 20 is: 0.3553533578183094

- Kmeans Cluster_Size 20, NMF Component Numbers 200

    - Homogeneity score for K-Means number of clusters: 20 and number of components for NMF r: 200 is: 0.0907254609500196
    - Completeness score for K-Means number of clusters: 20 and number of components for NMF r: 200 is: 0.12543730987535945
    - V-measure score for for K-Means number of clusters: 20 and number of components for NMF r: 200 is: 0.1052943364421039
    - Adjusted Rand Index score for K-Means number of clusters: 20 and number of components for NMF r: 200 is: 0.010916348099520504
    - Adjusted mutual information score for K-Means number of clusters: 20 and number of components for NMF r: 200 is: 0.1018626494845147

- Kmeans Cluster_Size 50, NMF Component Numbers 5
    - Homogeneity score for K-Means number of clusters: 50 and number of components for NMF r: 5 is: 0.3069473362856869
    - Completeness score for K-Means number of clusters: 50 and number of components for NMF r: 5 is: 0.2496685317221516
    - V-measure score for for K-Means number of clusters: 50 and number of components for NMF r: 5 is: 0.2753607835175613
    - Adjusted Rand Index score for K-Means number of clusters: 50 and number of components for NMF r: 5 is: 0.07674361816684346
    - Adjusted mutual information score for K-Means number of clusters: 50 and number of components for NMF r: 5 is: 0.2698185226384036

- Kmeans Cluster_Size 50, NMF Component Numbers 20
    - Homogeneity score for K-Means number of clusters: 50 and number of components for NMF r: 20 is: 0.41542319305130826
    - Completeness score for K-Means number of clusters: 50 and number of components for NMF r: 20 is: 0.3465406594960813
    - V-measure score for for K-Means number of clusters: 50 and number of components for NMF r: 20 is: 0.3778683904982612
    - Adjusted Rand Index score for K-Means number of clusters: 50 and number of components for NMF r: 20 is: 0.12833896867353864
    - Adjusted mutual information score for K-Means number of clusters: 50 and number of components for NMF r: 20 is: 0.37303111004632555

- Kmeans Cluster_Size 50, NMF Component Numbers 200
    - Homogeneity score for K-Means number of clusters: 50 and number of components for NMF r: 200 is: 0.16489279016510686
    - Completeness score for K-Means number of clusters: 50 and number of components for NMF r: 200 is: 0.1619222366718426
    - V-measure score for for K-Means number of clusters: 50 and number of components for NMF r: 200 is: 0.16339401313952223
    - Adjusted Rand Index score for K-Means number of clusters: 50 and number of components for NMF r: 200 is: 0.022731901794821253
    - Adjusted mutual information score for K-Means number of clusters: 50 and number of components for NMF r: 200 is: 0.15611116998254368

- **Out of these combinations, the best model is K-means(number of clusters =50) and NMF (r=20) with average score= 0.32824046435310295**

  The contingency matrix for this best combination of K-means NMF is:

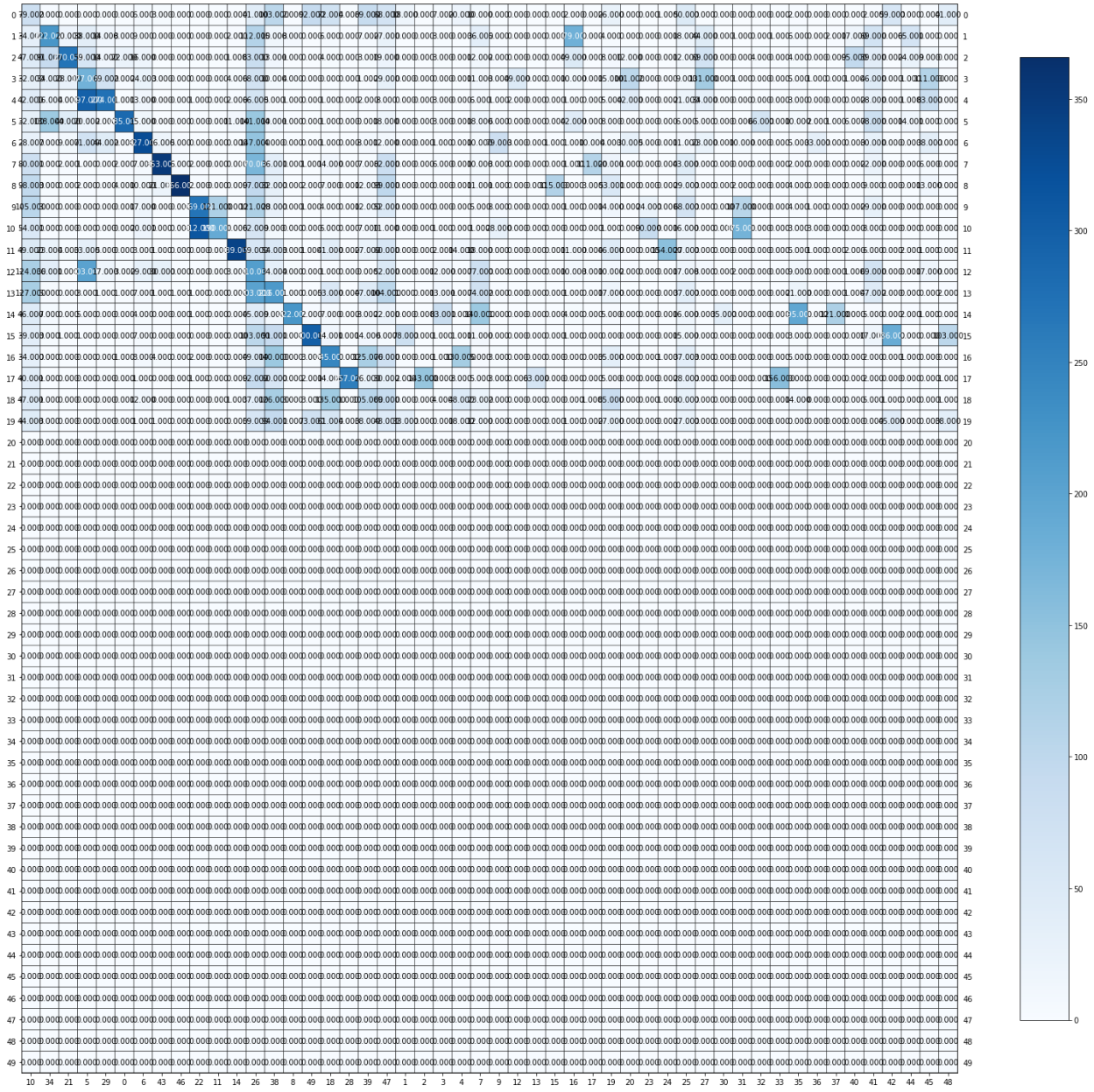

Fig26: Plot for NMF K-means clustering contingency matrix for best n_size=50, r=20

Dimensionality Reduction: UMAP and Clustering: K-Means

- Kmeans Cluster_Size 10
- UMAP Component Numbers 5
    - Homogeneity score for K-Means number of clusters: 10 and number of components for UMAP r: 5 is: 0.45469488342138226
    - Completeness score for K-Means number of clusters: 10 and number of components for UMAP r: 5 is: 0.6442757634680574
    - V-measure score for for K-Means number of clusters: 10 and number of components for UMAP r: 5 is: 0.5331332442599845
    - Adjusted Rand Index score for K-Means number of clusters: 10 and number of components for UMAP r: 5 is: 0.32875681780748195
    - Adjusted mutual information score for K-Means number of clusters: 10 and number of components for UMAP r: 5 is: 0.5322841560819805

- Kmeans Cluster_Size 10
- UMAP Component Numbers 20
    - Homogeneity score for K-Means number of clusters: 10 and number of components for UMAP r: 20 is: 0.46042158394839333
    - Completeness score for K-Means number of clusters: 10 and number of components for UMAP r: 20 is: 0.6535182841213933
    - V-measure score for for K-Means number of clusters: 10 and number of components for UMAP r: 20 is: 0.5402336914932243
    - Adjusted Rand Index score for K-Means number of clusters: 10 and number of components for UMAP r: 20 is: 0.33495511973931136
    - Adjusted mutual information score for K-Means number of clusters: 10 and number of components for UMAP r: 20 is: 0.5393981223674799
- Kmeans Cluster_Size 10
- UMAP Component Numbers 200
    - Homogeneity score for K-Means number of clusters: 10 and number of components for UMAP r: 200 is: 0.45792910133297954
    - Completeness score for K-Means number of clusters: 10 and number of components for UMAP r: 200 is: 0.6499597731850681
    - V-measure score for for K-Means number of clusters: 10 and number of components for UMAP r: 200 is: 0.5373020736699833
    - Adjusted Rand Index score for K-Means number of clusters: 10 and number of components for UMAP r: 200 is: 0.3317749096002451
    - Adjusted mutual information score for K-Means number of clusters: 10 and number of components for UMAP r: 200 is: 0.5364601472137548
- Kmeans Cluster_Size 20
- UMAP Component Numbers 5
    - Homogeneity score for K-Means number of clusters: 20 and number of components for UMAP r: 5 is: 0.5670938806900967
    - Completeness score for K-Means number of clusters: 20 and number of components for UMAP r: 5 is: 0.590228347938683
    - V-measure score for for K-Means number of clusters: 20 and number of components for UMAP r: 5 is: 0.5784298893531662
    - Adjusted Rand Index score for K-Means number of clusters: 20 and number of components for UMAP r: 5 is: 0.4500749504152894
    - Adjusted mutual information score for K-Means number of clusters: 20 and number of components for UMAP r: 5 is: 0.577030289065333
- Kmeans Cluster_Size 20
- UMAP Component Numbers 20
    - Homogeneity score for K-Means number of clusters: 20 and number of components for UMAP r: 20 is: 0.575073852683973
    - Completeness score for K-Means number of clusters: 20 and number of components for UMAP r: 20 is: 0.5888697641352526
    - V-measure score for for K-Means number of clusters: 20 and number of components for UMAP r: 20 is: 0.5818900487908386
    - Adjusted Rand Index score for K-Means number of clusters: 20 and number of components for UMAP r: 20 is: 0.4565905437716971
    - Adjusted mutual information score for K-Means number of clusters: 20 and number of components for UMAP r: 20 is: 0.5805234983276074
- Kmeans Cluster_Size 20
- UMAP Component Numbers 200
    - Homogeneity score for K-Means number of clusters: 20 and number of components for UMAP r: 200 is: 0.5686040584908657
    - Completeness score for K-Means number of clusters: 20 and number of components for UMAP r: 200 is: 0.5898662748423672
    - V-measure score for for K-Means number of clusters: 20 and number of components for UMAP r: 200 is: 0.5790400465020468

    - Adjusted Rand Index score for K-Means number of clusters: 20 and number of components for UMAP r: 200 is: 0.44795999243674783
    - Adjusted mutual information score for K-Means number of clusters: 20 and number of components for UMAP r: 200 is: 0.5776448277715267
- Kmeans Cluster_Size 50
- UMAP Component Numbers 5
    - Homogeneity score for K-Means number of clusters: 50 and number of components for UMAP r: 5 is: 0.6253435686790321
    - Completeness score for K-Means number of clusters: 50 and number of components for UMAP r: 5 is: 0.49589799182380134
    - V-measure score for for K-Means number of clusters: 50 and number of components for UMAP r: 5 is: 0.553148635997386
    - Adjusted Rand Index score for K-Means number of clusters: 50 and number of components for UMAP r: 5 is: 0.37123199723977396
    - Adjusted mutual information score for K-Means number of clusters: 50 and number of components for UMAP r: 5 is: 0.549804713342901
- Kmeans Cluster_Size 50
- UMAP Component Numbers 20
    - Homogeneity score for K-Means number of clusters: 50 and number of components for UMAP r: 20 is: 0.6243376469598874
    - Completeness score for K-Means number of clusters: 50 and number of components for UMAP r: 20 is: 0.4950163383228586
    - V-measure score for for K-Means number of clusters: 50 and number of components for UMAP r: 20 is: 0.5522066119184379
    - Adjusted Rand Index score for K-Means number of clusters: 50 and number of components for UMAP r: 20 is: 0.37551651749917037
    - Adjusted mutual information score for K-Means number of clusters: 50 and number of components for UMAP r: 20 is: 0.548855347254154
- Kmeans Cluster_Size 50
- UMAP Component Numbers 200
    - Homogeneity score for K-Means number of clusters: 50 and number of components for UMAP r: 200 is: 0.6266359408271271
    - Completeness score for K-Means number of clusters: 50 and number of components for UMAP r: 200 is: 0.5036450023765612
    - V-measure score for for K-Means number of clusters: 50 and number of components for UMAP r: 200 is: 0.5584488738040102
    - Adjusted Rand Index score for K-Means number of clusters: 50 and number of components for UMAP r: 200 is: 0.3960239696604094
    - Adjusted mutual information score for K-Means number of clusters: 50 and number of components for UMAP r: 200 is: 0.5551085694243425

- **Out of these combinations, the best model is K-means(number of clusters =20) and UMAP (r=20) with average score= 0.5565895415418738**

  The contingency matrix for this best combination of K-means UMAP is:

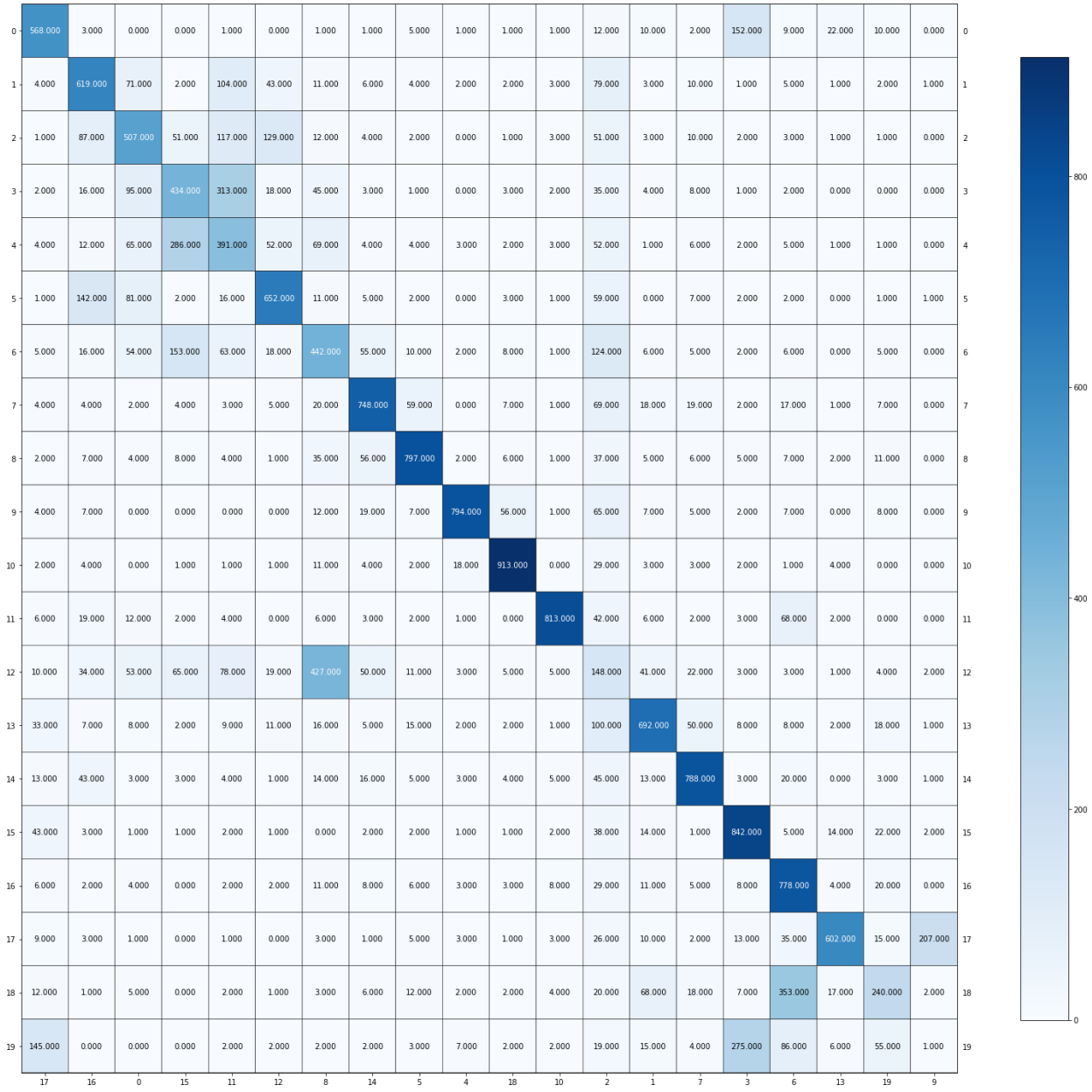

Fig27: Plot for UMAP K-means clustering contingency matrix for best n_size=20, r=20

Dimensionality Reduction: SVD and Clustering: Agglomerative

- Agglomerative Cluster_Size 20
- SVD Component Numbers 5
    - Homogeneity score for Agglomerative number of clusters: 20 and number of components for SVD r: 5 is: 0.31621596978615263
    - Completeness score for Agglomerative number of clusters: 20 and number of components for SVD r: 5 is: 0.35005368051467983
    - V-measure score for for Agglomerative number of clusters: 20 and number of components for SVD r: 5 is: 0.3322755704426335
    - Adjusted Rand Index score for Agglomerative number of clusters: 20 and number of components for SVD r: 5 is: 0.11655998591052552
    - Adjusted mutual information score for Agglomerative number of clusters: 20 and number of components for SVD r: 5 is: 0.33000280772828616

- Agglomerative Cluster_Size 20
- SVD Component Numbers 20
    - Homogeneity score for Agglomerative number of clusters: 20 and number of components for SVD r: 20 is: 0.3607240545132558
    - Completeness score for Agglomerative number of clusters: 20 and number of components for SVD r: 20 is: 0.40382787033806405
    - V-measure score for for Agglomerative number of clusters: 20 and number of components for SVD r: 20 is: 0.38106091157151406
    - Adjusted Rand Index score for Agglomerative number of clusters: 20 and number of components for SVD r: 20 is: 0.15174901635733087
    - Adjusted mutual information score for Agglomerative number of clusters: 20 and number of components for SVD r: 20 is: 0.3789396648976684
- Agglomerative Cluster_Size 20
- SVD Component Numbers 200
    - Homogeneity score for Agglomerative number of clusters: 20 and number of components for SVD r: 200 is: 0.31722561498107443
    - Completeness score for Agglomerative number of clusters: 20 and number of components for SVD r: 200 is: 0.42942831830364525
    - V-measure score for for Agglomerative number of clusters: 20 and number of components for SVD r: 200 is: 0.36489638985726963
    - Adjusted Rand Index score for Agglomerative number of clusters: 20 and number of components for SVD r: 200 is: 0.09329067079444811
    - Adjusted mutual information score for Agglomerative number of clusters: 20 and number of components for SVD r: 200 is: 0.36249951544343134

- **Out of these combinations, the best model is Agglomerative (n_clusters =20) and SVD(r=20) with average score=0.3352603035355667**

  The contingency matrix for this best combination of Agglomerative SVD is:

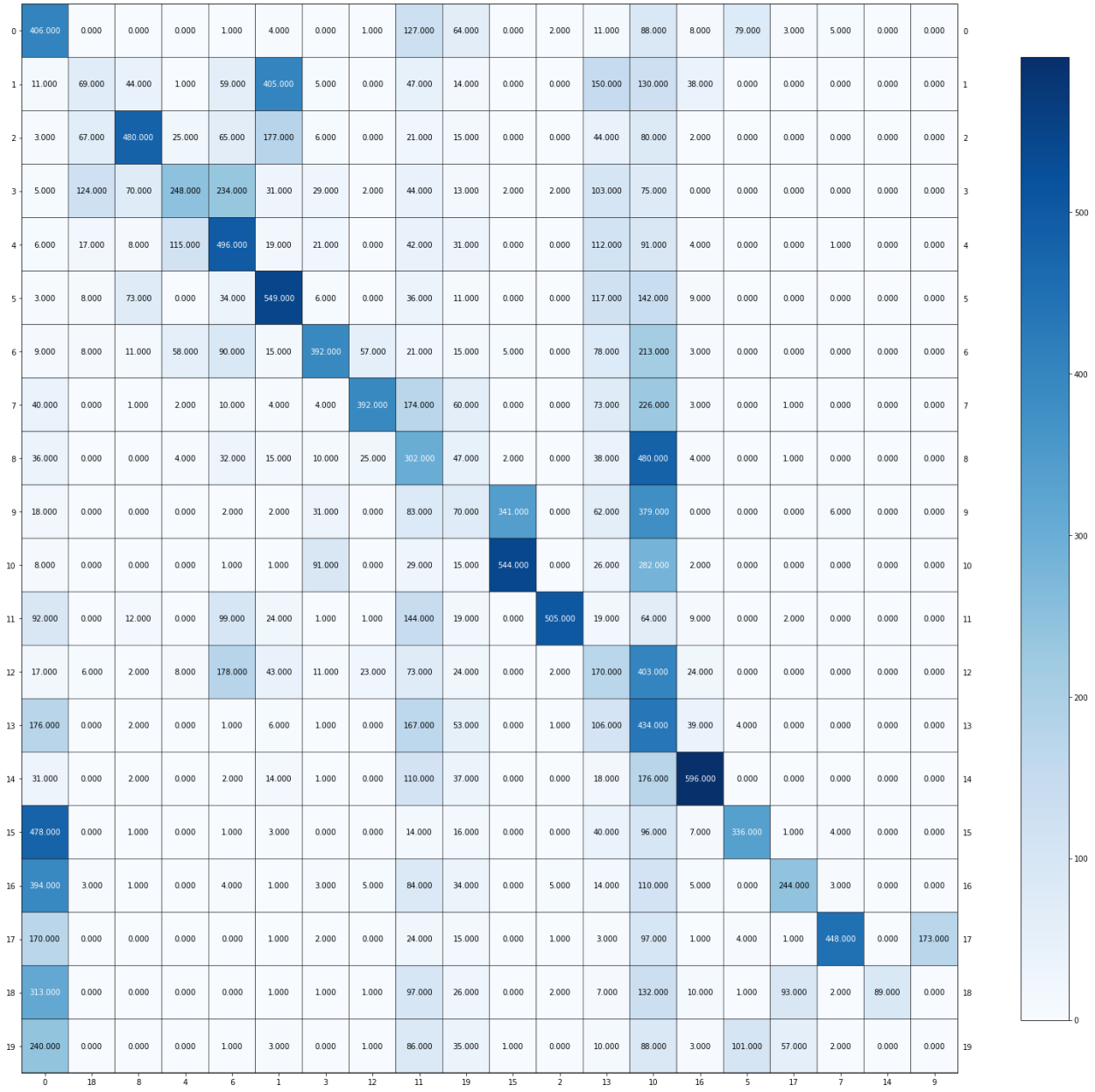

Fig28: Plot for SVD Agglomerative clustering contingency matrix for best n_cluster =20, r=20

Dimensionality Reduction: NMF and Clustering: Agglomerative

- Agglomerative Cluster_Size 20
- NMF Component Numbers 5
    - Homogeneity score for Agglomerative number of clusters: 20 and number of components for NMF r: 5 is: 0.2716107553377085
    - Completeness score for Agglomerative number of clusters: 20 and number of components for NMF r: 5 is: 0.2981826591319834
    - V-measure score for for Agglomerative number of clusters: 20 and number of components for NMF r: 5 is: 0.2842771264768713
    - Adjusted Rand Index score for Agglomerative number of clusters: 20 and number of components for NMF r: 5 is: 0.09885699946876714
    - Adjusted mutual information score for Agglomerative number of clusters: 20 and number of components for NMF r: 5 is: 0.28185074708101315

- Agglomerative Cluster_Size 20
- NMF Component Numbers 20
    - Homogeneity score for Agglomerative number of clusters: 20 and number of components for NMF r: 20 is: 0.361414804564651
    - Completeness score for Agglomerative number of clusters: 20 and number of components for NMF r: 20 is: 0.4156386916239138
    - V-measure score for for Agglomerative number of clusters: 20 and number of components for NMF r: 20 is: 0.38663483850103203
    - Adjusted Rand Index score for Agglomerative number of clusters: 20 and number of components for NMF r: 20 is: 0.14959175515373935
    - Adjusted mutual information score for Agglomerative number of clusters: 20 and number of components for NMF r: 20 is: 0.3844995859631181
- Agglomerative Cluster_Size 20
- NMF Component Numbers 200
    - Homogeneity score for Agglomerative number of clusters: 20 and number of components for NMF r: 200 is: 0.16104447680960302
    - Completeness score for Agglomerative number of clusters: 20 and number of components for NMF r: 200 is: 0.2138016867767599
    - V-measure score for for Agglomerative number of clusters: 20 and number of components for NMF r: 200 is: 0.18371046115850687
    - Adjusted Rand Index score for Agglomerative number of clusters: 20 and number of components for NMF r: 200 is: 0.04593667270461881
    - Adjusted mutual information score for Agglomerative number of clusters: 20 and number of components for NMF r: 200 is: 0.1806037699150735

- **Out of these combinations, the best model is Agglomerative (n_clusters =20) and NMF(r=20) with average score= 0.3395559351612909**

  The contingency matrix for this best combination of Agglomerative NMF is:

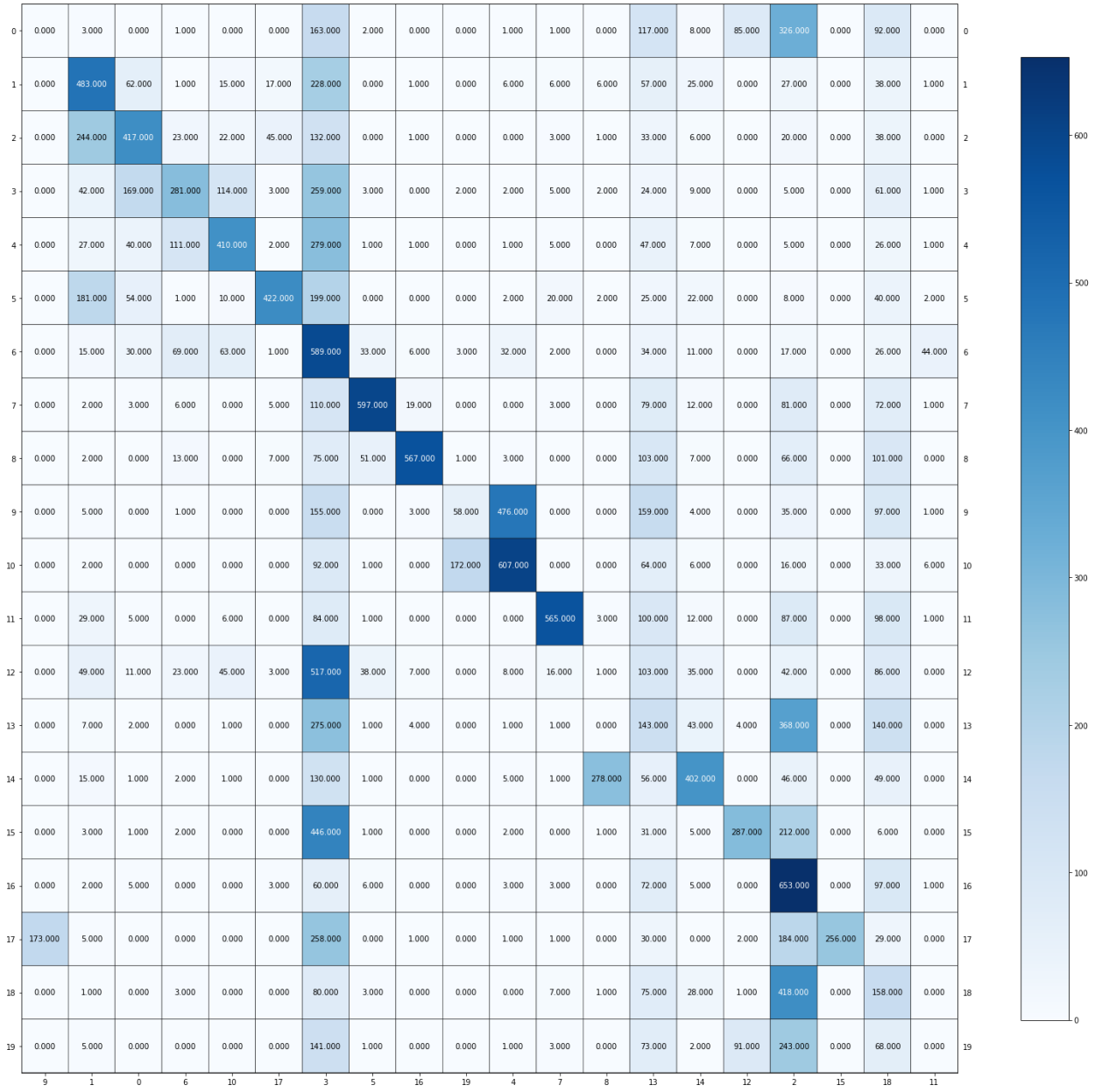

Fig29: Plot for NMF Agglomerative clustering contingency matrix for best n_cluster =20, r=20

Dimensionality Reduction: UMAP and Clustering: Ward Agglomerative

- n_components=5
  - Homogeneity score for Ward and r: 5 is: 0.5581261201222677
  - Completeness score Ward and r: 5 is: 0.5841421774465988
  - V-measure score for Ward and r: 5 is: 0.5708378807184529
  - Adjusted Rand Index score for Ward and r: 5 is: 0.42741165149002364
  - Adjusted mutual information score for Ward and r: 5 is: 0.5694078427389649
- n_components=20
  - Homogeneity score for Ward and r: 20 is: 0.5581261201222677
  - Completeness score Ward and r: 20 is: 0.5841421774465988
  - V-measure score for Ward and r: 20 is: 0.5708378807184529
  - Adjusted Rand Index score for Ward and r: 20 is: 0.42741165149002364
  - Adjusted mutual information score for Ward and r: 20 is: 0.5694078427389649

- n_components=200
    - o Homogeneity score for Ward and r: 200 is: 0.5581261201222677
    - o Completeness score Ward and r: 200 is: 0.5841421774465988
    - o V-measure score for Ward and r: 200 is: 0.5708378807184529
    - o Adjusted Rand Index score for Ward and r: 200 is: 0.42741165149002364
    - o Adjusted mutual information score for Ward and r: 200 is: 0.5694078427389649

- **Out of these combinations, the best model is Agglomerative (Ward linkage) and UMAP(n_components=5) with average score= 0.5419851345032616**

  The contingency matrix for this best combination of Agglomerative UMAP is:

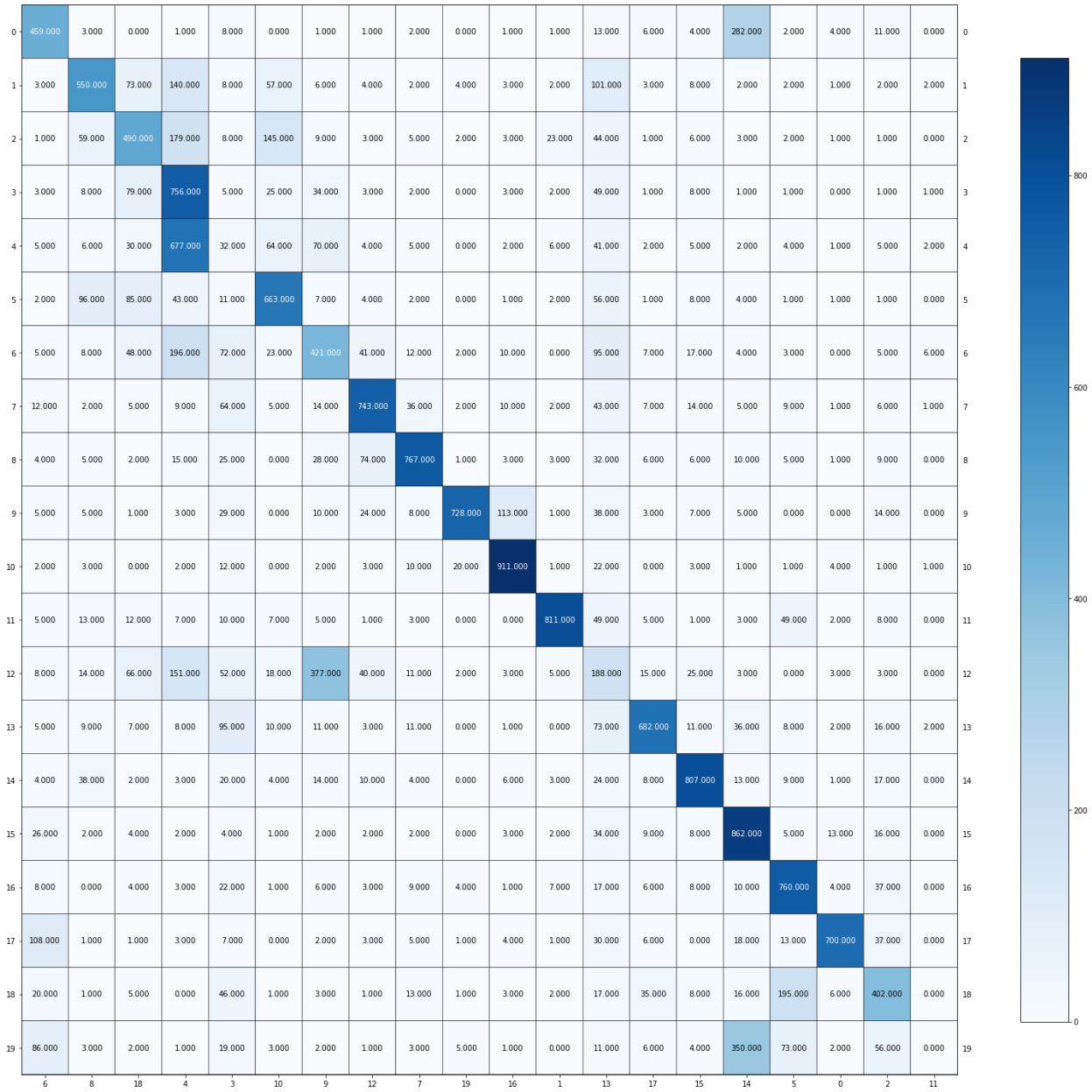

Fig30: Plot for UMAP Agglomerative clustering (Ward linkage) contingency matrix for, r=5

<u>Dimensionality Reduction: SVD and Clustering: HDBSCAN</u>

- Hdbscan Cluster_Size 100
- SVD Component Numbers 5
    - Homogeneity score for HDBSCAN number of clusters: 100 and number of components for SVD r: 5 is: 0.0
    - Completeness score for HDBSCAN number of clusters: 100 and number of components for SVD r: 5 is: 1.0
    - V-measure score for for HDBSCAN number of clusters: 100 and number of components for SVD r: 5 is: 0.0
    - Adjusted Rand Index score for HDBSCAN number of clusters: 100 and number of components for SVD r: 5 is: 0.0
    - Adjusted mutual information score for HDBSCAN number of clusters: 100 and number of components for SVD r: 5 is: -1.6056889920887454e-16
- Hdbscan Cluster_Size 100
- SVD Component Numbers 20
    - Homogeneity score for HDBSCAN number of clusters: 100 and number of components for SVD r: 20 is: 0.0
    - Completeness score for HDBSCAN number of clusters: 100 and number of components for SVD r: 20 is: 1.0
    - V-measure score for for HDBSCAN number of clusters: 100 and number of components for SVD r: 20 is: 0.0
    - Adjusted Rand Index score for HDBSCAN number of clusters: 100 and number of components for SVD r: 20 is: 0.0
    - Adjusted mutual information score for HDBSCAN number of clusters: 100 and number of components for SVD r: 20 is: -1.6056889920887454e-16
- Hdbscan Cluster_Size 100
- SVD Component Numbers 200
    - Homogeneity score for HDBSCAN number of clusters: 100 and number of components for SVD r: 200 is: 0.0
    - Completeness score for HDBSCAN number of clusters: 100 and number of components for SVD r: 200 is: 1.0
    - V-measure score for for HDBSCAN number of clusters: 100 and number of components for SVD r: 200 is: 0.0
    - Adjusted Rand Index score for HDBSCAN number of clusters: 100 and number of components for SVD r: 200 is: 0.0
    - Adjusted mutual information score for HDBSCAN number of clusters: 100 and number of components for SVD r: 200 is: -1.6056889920887454e-16
- Hdbscan Cluster_Size 200
- SVD Component Numbers 5
    - Homogeneity score for HDBSCAN number of clusters: 200 and number of components for SVD r: 5 is: 0.0
    - Completeness score for HDBSCAN number of clusters: 200 and number of components for SVD r: 5 is: 1.0
    - V-measure score for for HDBSCAN number of clusters: 200 and number of components for SVD r: 5 is: 0.0
    - Adjusted Rand Index score for HDBSCAN number of clusters: 200 and number of components for SVD r: 5 is: 0.0
    - Adjusted mutual information score for HDBSCAN number of clusters: 200 and number of components for SVD r: 5 is: -1.6056889920887454e-16
- Hdbscan Cluster_Size 200
- SVD Component Numbers 20
    - Homogeneity score for HDBSCAN number of clusters: 200 and number of components for SVD r: 20 is: 0.0

    - Completeness score for HDBSCAN number of clusters: 200 and number of components for SVD r: 20 is: 1.0
    - V-measure score for for HDBSCAN number of clusters: 200 and number of components for SVD r: 20 is: 0.0
    - Adjusted Rand Index score for HDBSCAN number of clusters: 200 and number of components for SVD r: 20 is: 0.0
    - Adjusted mutual information score for HDBSCAN number of clusters: 200 and number of components for SVD r: 20 is: -1.6056889920887454e-16
- Hdbscan Cluster_Size 200
- SVD Component Numbers 200
    - Homogeneity score for HDBSCAN number of clusters: 200 and number of components for SVD r: 200 is: 0.0
    - Completeness score for HDBSCAN number of clusters: 200 and number of components for SVD r: 200 is: 1.0
    - V-measure score for for HDBSCAN number of clusters: 200 and number of components for SVD r: 200 is: 0.0
    - Adjusted Rand Index score for HDBSCAN number of clusters: 200 and number of components for SVD r: 200 is: 0.0
    - Adjusted mutual information score for HDBSCAN number of clusters: 200 and number of components for SVD r: 200 is: -1.6056889920887454e-16

- **Out of these combinations, the best model is HDBSCAN(min cluster size=100) and SVD(r=5) with average score= 0.19999999999999998**

  The contingency matrix for this best combination of HDBSCAN SVD is:

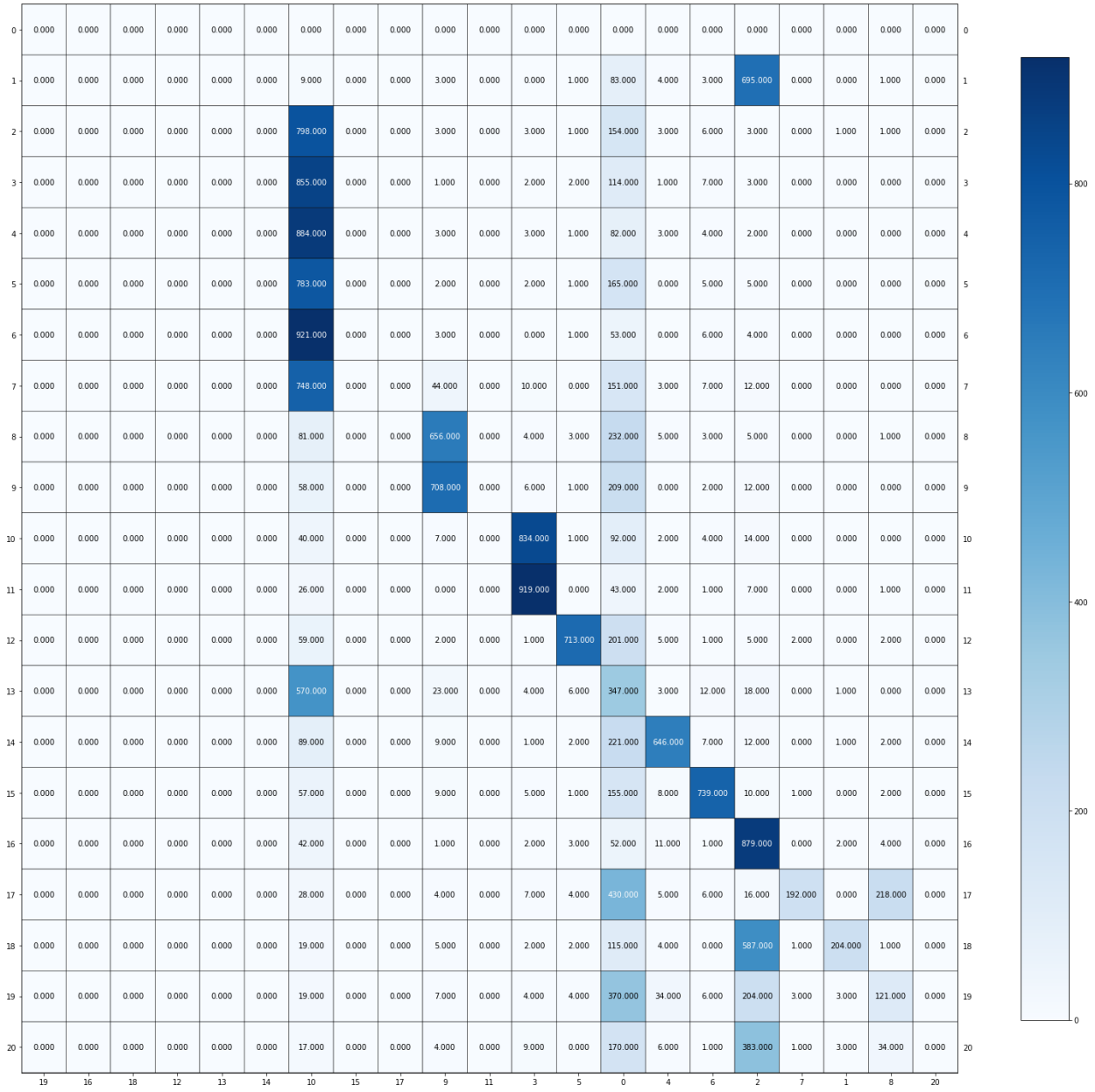

Fig31: Plot for SVD HDBSCAN clustering contingency matrix for min cluster size=100, r=5

Dimensionality Reduction: NMF and Clustering: HDBSCAN

- Hdbscan Cluster_Size 100
- NMF Component Numbers 5
    - Homogeneity score for HDBSCAN number of clusters: 100 and number of components for NMF r: 5 is: 0.05352509921582616
    - Completeness score for HDBSCAN number of clusters: 100 and number of components for NMF r: 5 is: 0.26201120551020535
    - V-measure score for for HDBSCAN number of clusters: 100 and number of components for NMF r: 5 is: 0.08889104398157056
    - Adjusted Rand Index score for HDBSCAN number of clusters: 100 and number of components for NMF r: 5 is: 0.005504626310355915
    - Adjusted mutual information score for HDBSCAN number of clusters: 100 and number of components for NMF r: 5 is: 0.08785902261704848

- Hdbscan Cluster_Size 100
- NMF Component Numbers 20
    - Homogeneity score for HDBSCAN number of clusters: 100 and number of components for NMF r: 20 is: 0.0
    - Completeness score for HDBSCAN number of clusters: 100 and number of components for NMF r: 20 is: 1.0
    - V-measure score for for HDBSCAN number of clusters: 100 and number of components for NMF r: 20 is: 0.0
    - Adjusted Rand Index score for HDBSCAN number of clusters: 100 and number of components for NMF r: 20 is: 0.0
    - Adjusted mutual information score for HDBSCAN number of clusters: 100 and number of components for NMF r: 20 is: -1.6056889920887454e-16
- Hdbscan Cluster_Size 100
- NMF Component Numbers 200
    - Homogeneity score for HDBSCAN number of clusters: 100 and number of components for NMF r: 200 is: 0.0
    - Completeness score for HDBSCAN number of clusters: 100 and number of components for NMF r: 200 is: 1.0
    - V-measure score for for HDBSCAN number of clusters: 100 and number of components for NMF r: 200 is: 0.0
    - Adjusted Rand Index score for HDBSCAN number of clusters: 100 and number of components for NMF r: 200 is: 0.0
    - Adjusted mutual information score for HDBSCAN number of clusters: 100 and number of components for NMF r: 200 is: -1.6056889920887454e-16
- Hdbscan Cluster_Size 200
- NMF Component Numbers 5
    - Homogeneity score for HDBSCAN number of clusters: 200 and number of components for NMF r: 5 is: 0.04792836309613274
    - Completeness score for HDBSCAN number of clusters: 200 and number of components for NMF r: 5 is: 0.23419766395698405
    - V-measure score for for HDBSCAN number of clusters: 200 and number of components for NMF r: 5 is: 0.07957231590180865
    - Adjusted Rand Index score for HDBSCAN number of clusters: 200 and number of components for NMF r: 5 is: 0.010484976997200262
    - Adjusted mutual information score for HDBSCAN number of clusters: 200 and number of components for NMF r: 5 is: 0.07878922877073667
- Hdbscan Cluster_Size 200
- NMF Component Numbers 20
    - Homogeneity score for HDBSCAN number of clusters: 200 and number of components for NMF r: 20 is: 0.0
    - Completeness score for HDBSCAN number of clusters: 200 and number of components for NMF r: 20 is: 1.0
    - V-measure score for for HDBSCAN number of clusters: 200 and number of components for NMF r: 20 is: 0.0
    - Adjusted Rand Index score for HDBSCAN number of clusters: 200 and number of components for NMF r: 20 is: 0.0
    - Adjusted mutual information score for HDBSCAN number of clusters: 200 and number of components for NMF r: 20 is: -1.6056889920887454e-16
- Hdbscan Cluster_Size 200
- NMF Component Numbers 200
    - Homogeneity score for HDBSCAN number of clusters: 200 and number of components for NMF r: 200 is: 0.0
    - Completeness score for HDBSCAN number of clusters: 200 and number of components for NMF r: 200 is: 1.0
    - V-measure score for for HDBSCAN number of clusters: 200 and number of components for NMF r: 200 is: 0.0

- Adjusted Rand Index score for HDBSCAN number of clusters: 200 and number of components for NMF r: 200 is: 0.0
- Adjusted mutual information score for HDBSCAN number of clusters: 200 and number of components for NMF r: 200 is: -1.6056889920887454e-16

- **Out of these combinations, the best model is HDBSCAN(min cluster size=100) and NMF(r=20) with average score= 0.19999999999999998**

The contingency matrix for this best combination of HDBSCAN NMF is:

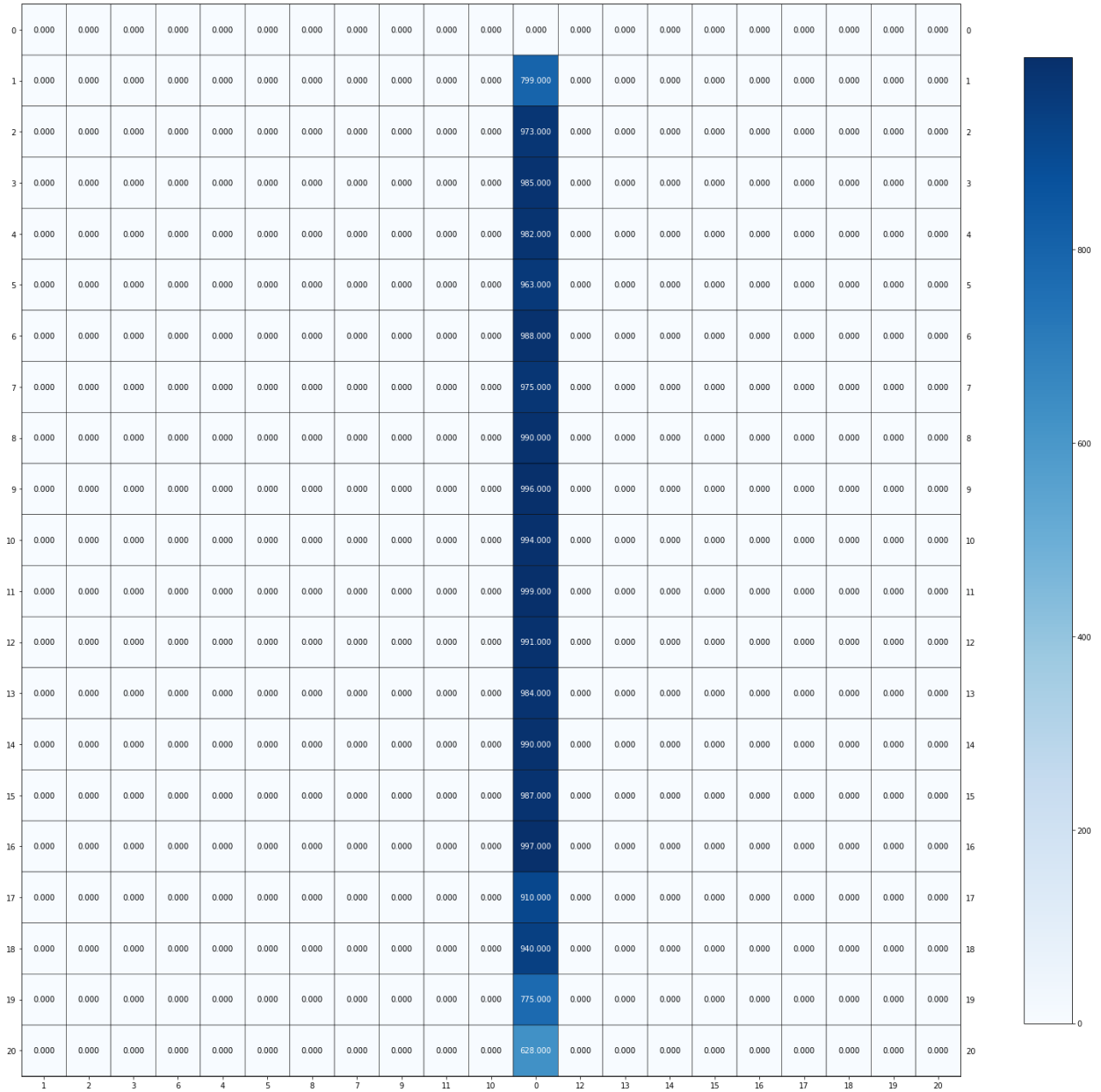

Fig32: Plot for NMF HDBSCAN clustering contingency matrix for min cluster size=100, r=20

<u>Dimensionality Reduction: UMAP and Clustering: HDBSCAN</u>

- Cluster_Size 20
- Umap Component Numbers 5

    - Homogeneity score for HDBSCAN number of clusters: 20 and number of components r: 5 is: 0.430917114056084
    - Completeness score for HDBSCAN number of clusters: 20 and number of components r: 5 is: 0.4480049803384388
    - V-measure score for for HDBSCAN number of clusters: 20 and number of components r: 5 is: 0.4392949373816441
    - Adjusted Rand Index score for HDBSCAN number of clusters: 20 and number of components r: 5 is: 0.07917173521282128
    - Adjusted mutual information score for HDBSCAN number of clusters: 20 and number of components r: 5 is: 0.42591218486922944
- Cluster_Size 20
- Umap Component Numbers 20
    - Homogeneity score for HDBSCAN number of clusters: 20 and number of components r: 20 is: 0.000390276104151654
    - Completeness score for HDBSCAN number of clusters: 20 and number of components r: 20 is: 0.10792763450387441
    - V-measure score for for HDBSCAN number of clusters: 20 and number of components r: 20 is: 0.000777739830615873
    - Adjusted Rand Index score for HDBSCAN number of clusters: 20 and number of components r: 20 is: 2.6250072145015002e-06
    - Adjusted mutual information score for HDBSCAN number of clusters: 20 and number of components r: 20 is: 0.000384132682354234
- Cluster_Size 20
- Umap Component Numbers 200
    - Homogeneity score for HDBSCAN number of clusters: 20 and number of components r: 200 is: 0.43343505381416986
    - Completeness score for HDBSCAN number of clusters: 20 and number of components r: 200 is: 0.4437075053409833
    - V-measure score for for HDBSCAN number of clusters: 20 and number of components r: 200 is: 0.43851112786149044
    - Adjusted Rand Index score for HDBSCAN number of clusters: 20 and number of components r: 200 is: 0.08271384440331317
    - Adjusted mutual information score for HDBSCAN number of clusters: 20 and number of components r: 200 is: 0.4266594621700962
- Cluster_Size 100
- Umap Component Numbers 5
    - Homogeneity score for HDBSCAN number of clusters: 100 and number of components r: 5 is: 0.4074222716704813
    - Completeness score for HDBSCAN number of clusters: 100 and number of components r: 5 is: 0.6012164181108816
    - V-measure score for for HDBSCAN number of clusters: 100 and number of components r: 5 is: 0.4857020880002561
    - Adjusted Rand Index score for HDBSCAN number of clusters: 100 and number of components r: 5 is: 0.19895017371978235
    - Adjusted mutual information score for HDBSCAN number of clusters: 100 and number of components r: 5 is: 0.484554948067443
- Cluster_Size 100
- Umap Component Numbers 20
    - Homogeneity score for HDBSCAN number of clusters: 100 and number of components r: 20 is: 0.4040666196573172
    - Completeness score for HDBSCAN number of clusters: 100 and number of components r: 20 is: 0.610272950839044
    - V-measure score for for HDBSCAN number of clusters: 100 and number of components r: 20 is: 0.48620981668527596
    - Adjusted Rand Index score for HDBSCAN number of clusters: 100 and number of components r: 20 is: 0.2155905771051507

    - Adjusted mutual information score for HDBSCAN number of clusters: 100 and number of components r: 20 is: 0.48526555785006165
- Cluster_Size 100
- Umap Component Numbers 200
    - Homogeneity score for HDBSCAN number of clusters: 100 and number of components r: 200 is: 0.40970534708752254
    - Completeness score for HDBSCAN number of clusters: 100 and number of components r: 200 is: 0.6198924916382902
    - V-measure score for for HDBSCAN number of clusters: 100 and number of components r: 200 is: 0.49334460289450793
    - Adjusted Rand Index score for HDBSCAN number of clusters: 100 and number of components r: 200 is: 0.2200515944160935
    - Adjusted mutual information score for HDBSCAN number of clusters: 100 and number of components r: 200 is: 0.4924128037520018
- Cluster_Size 200
- Umap Component Numbers 5
    - Homogeneity score for HDBSCAN number of clusters: 200 and number of components r: 5 is: 0.4163071997553647
    - Completeness score for HDBSCAN number of clusters: 200 and number of components r: 5 is: 0.6114435864948351
    - V-measure score for for HDBSCAN number of clusters: 200 and number of components r: 5 is: 0.495350372303337
    - Adjusted Rand Index score for HDBSCAN number of clusters: 200 and number of components r: 5 is: 0.21381927389595956
    - Adjusted mutual information score for HDBSCAN number of clusters: 200 and number of components r: 5 is: 0.4943308135082532
- Cluster_Size 200
- Umap Component Numbers 20
    - Homogeneity score for HDBSCAN number of clusters: 200 and number of components r: 20 is: 0.41464318531280137
    - Completeness score for HDBSCAN number of clusters: 200 and number of components r: 20 is: 0.6025153974017078
    - V-measure score for for HDBSCAN number of clusters: 200 and number of components r: 20 is: 0.4912290134974424
    - Adjusted Rand Index score for HDBSCAN number of clusters: 200 and number of components r: 20 is: 0.2128020173028053
    - Adjusted mutual information score for HDBSCAN number of clusters: 200 and number of components r: 20 is: 0.490205682489206
- Cluster_Size 200
- Umap Component Numbers 200
    - Homogeneity score for HDBSCAN number of clusters: 200 and number of components r: 200 is: 0.41262595806948077
    - Completeness score for HDBSCAN number of clusters: 200 and number of components r: 200 is: 0.604698384973462
    - V-measure score for for HDBSCAN number of clusters: 200 and number of components r: 200 is: 0.490530384235994
    - Adjusted Rand Index score for HDBSCAN number of clusters: 200 and number of components r: 200 is: 0.20982188737460572
    - Adjusted mutual information score for HDBSCAN number of clusters: 200 and number of components r: 200 is: 0.48950202767579426

- **Out of these combinations, the best model is HDBSCAN(min cluster size=100) and UMAP(r=200) with average score= 0.44708136795768316**

  The contingency matrix for this best combination of HDBSCAN UMAP is:

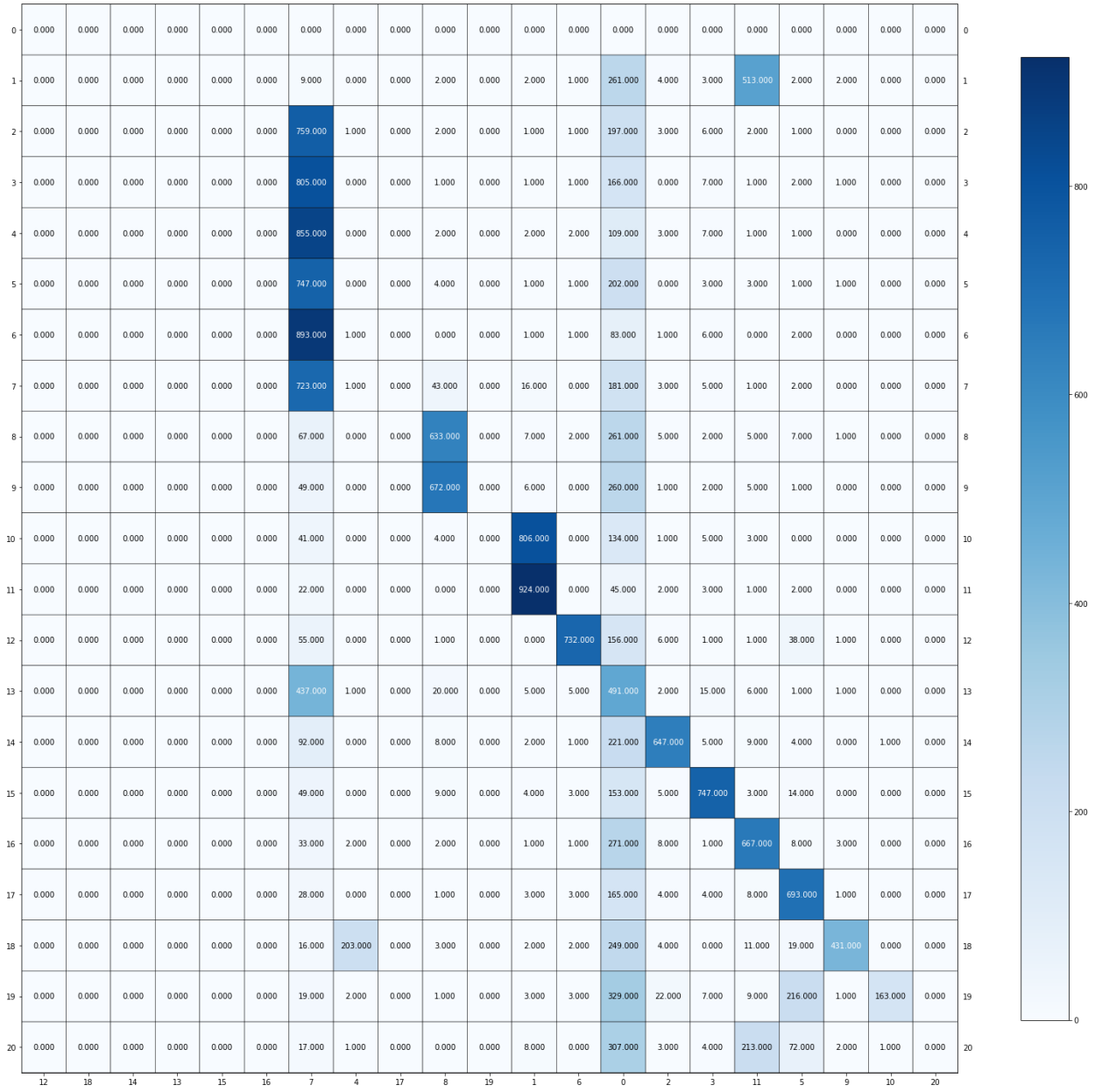

Fig33: Plot for UMAP HDBSCAN clustering contingency matrix for min cluster size=100, r=200

Dimensionality Reduction: None and Clustering: K-Means

- Running only k-means on r=[10,20,50], the best model is K-means(no of cluster =50) with average score= 0.33907105823854683

  The contingency matrix for this best combination of K-means is:

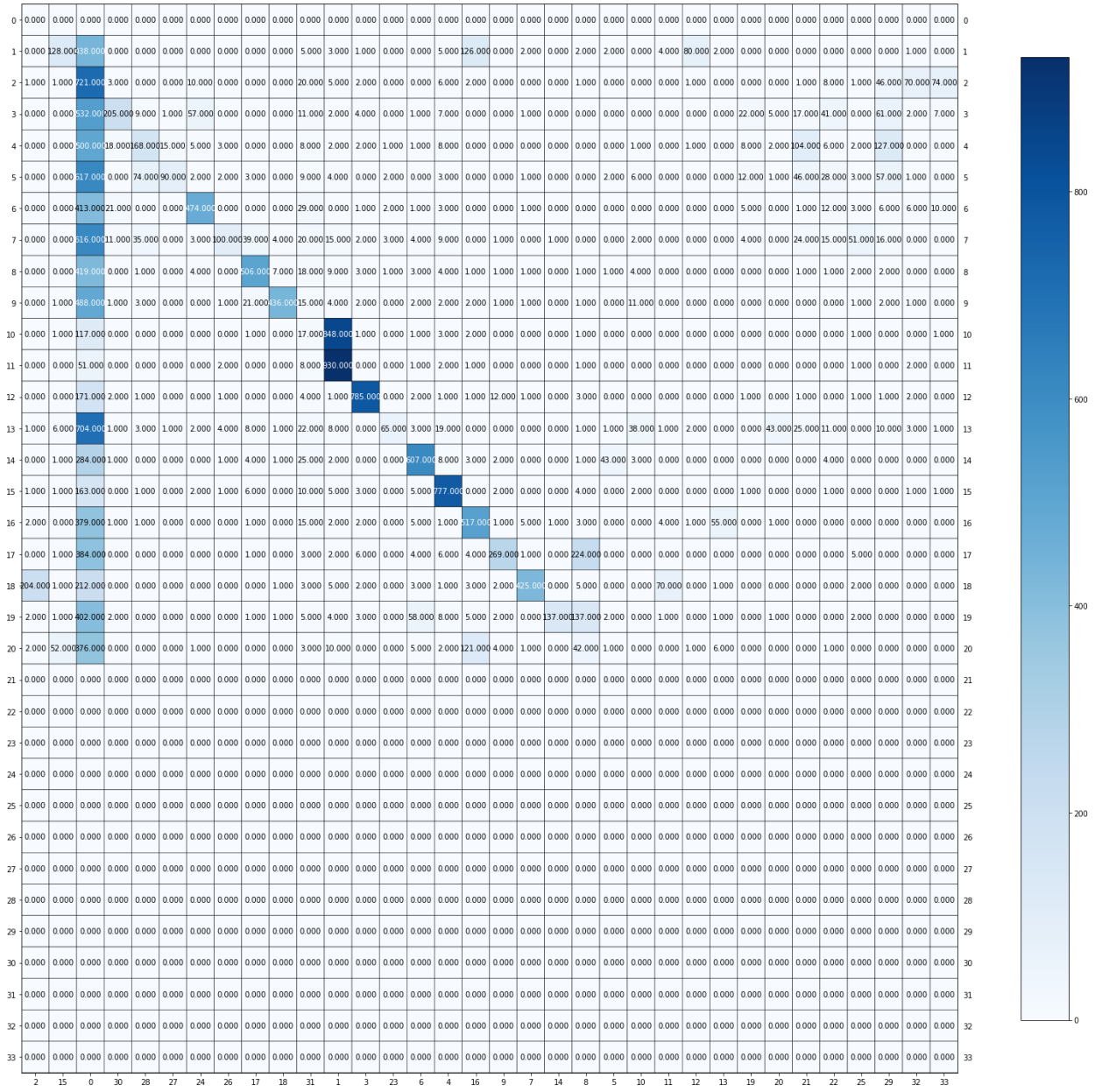

Fig34: Plot for K-means clustering contingency matrix for no of cluster =50

Dimensionality Reduction: None and Clustering: Agglomerative

- Running only Agglomerative on r=20 the best model is having average score= 0.34378284551429156

  The contingency matrix for this best Agglomerative clustering is:

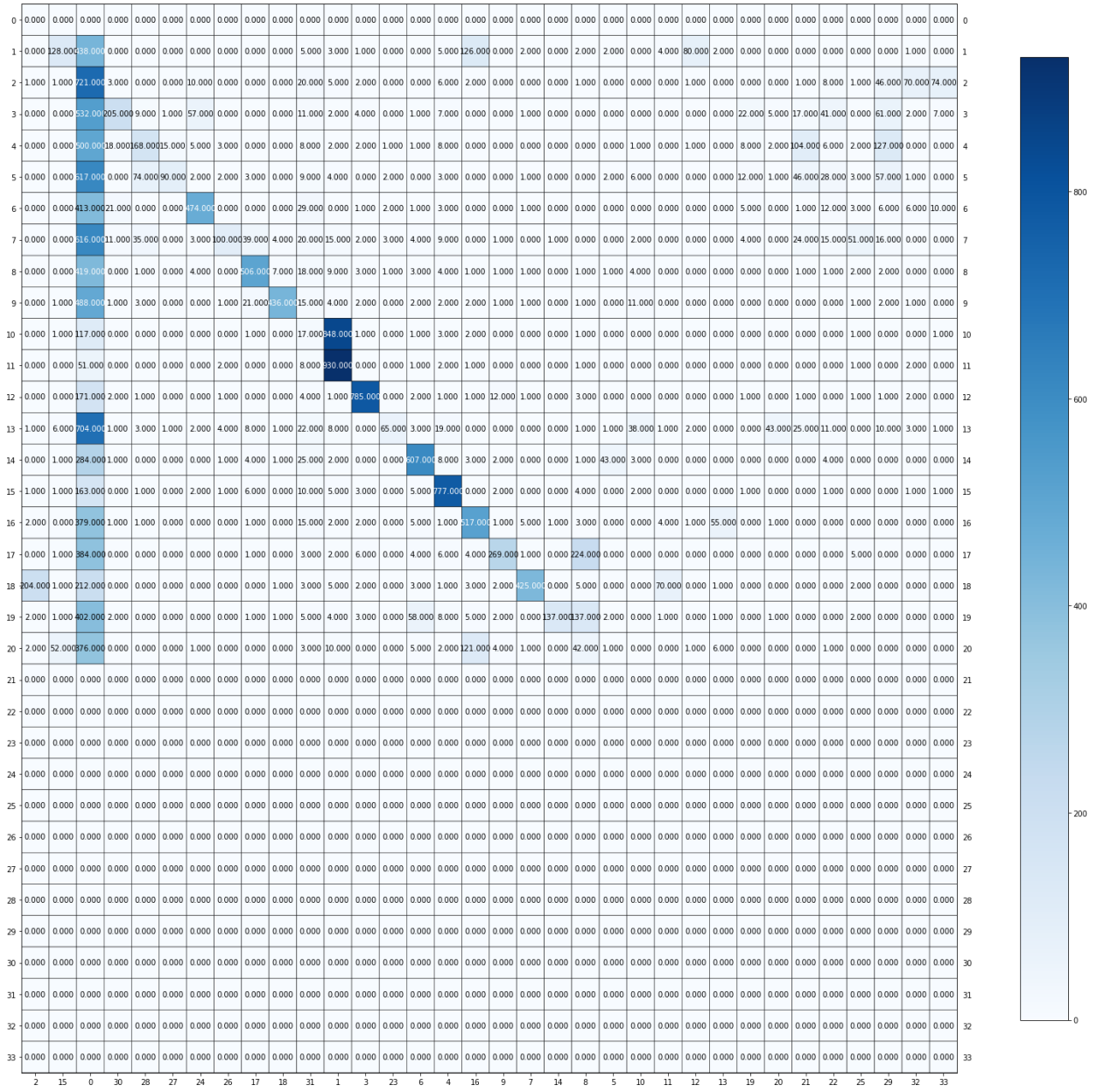

Fig35: Plot for Agglomerate clustering contingency matrix for no of cluster =20

Dimensionality Reduction: None and Clustering: HDBSCAN

- Running only HDBSCAN on min_cluster_size=[100,200], the best model is with min cluster size=100 for average score= 0.19999999999999998

  The contingency matrix for this best combination of HDBSCAN is:

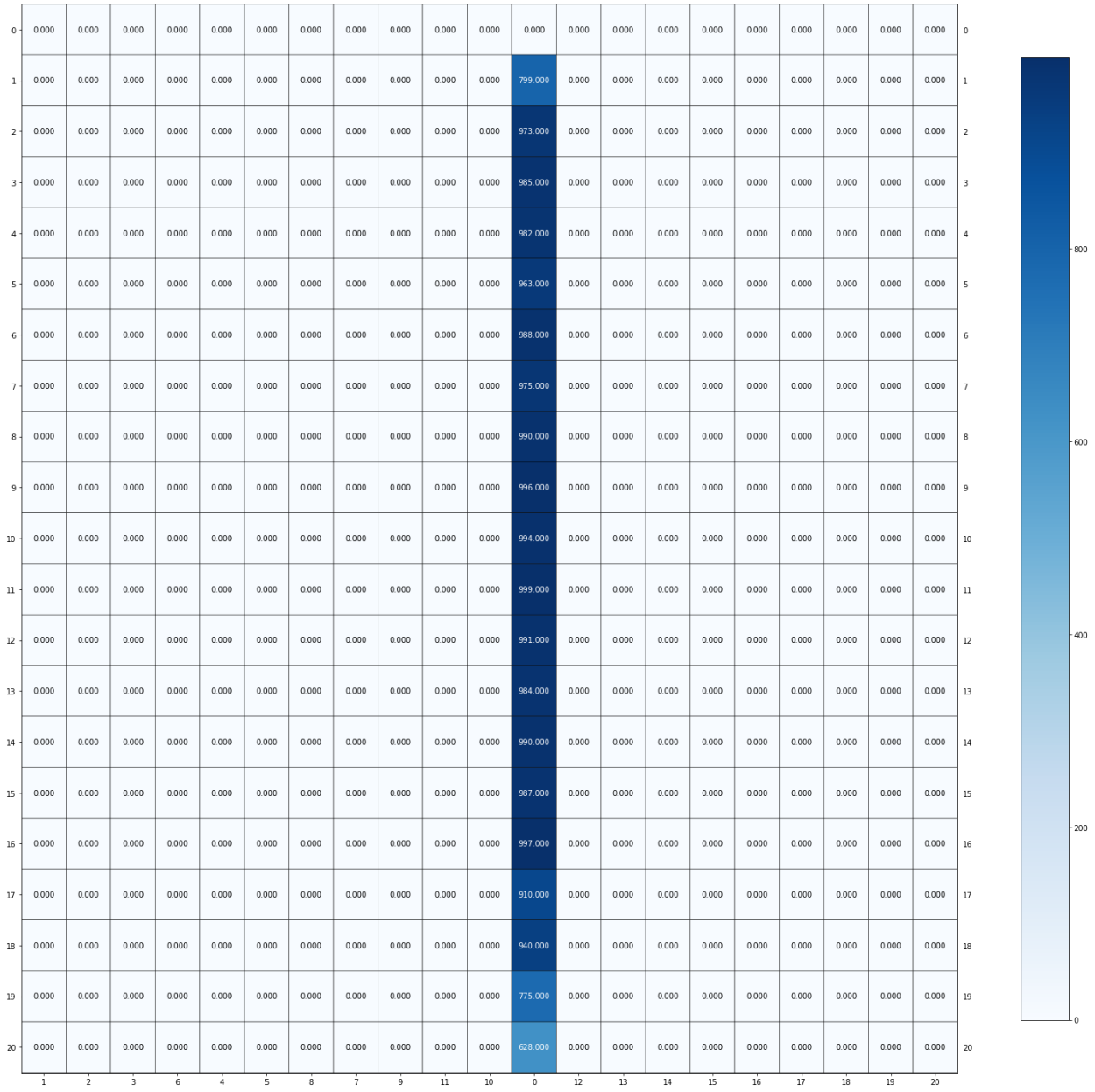

Fig36: Plot for HDBSCAN clustering contingency matrix for min cluster size=100

Based on all the variations from the table we conducted for different dimensionality reduction and clustering algorithms we get the following as the best combinations in terms of average clustering evaluation metric score:

- Best Average K-Means(k=50) SVD(r=200) values: 0.34169994830773764
- Best Average K-Means(k=50) NMF(r=20) values: 0.32824046435310295
- Best Average K-Means(k=20) UMAP(n_comp=20) values: 0.5565895415418738
- Best Average Agglomerative(n_cluster=20) Clustering SVD(r=20) values: 0.3352603035355667
- Best Average Agglomerative(n_cluster=20) Clustering NMF(r=20) values: 0.3395559351612909
- Best Average Ward Agglomerative UMAP(n_comp=5) values: 0.5419851345032616
- Best Average HDBSCAN(min_cluster_size=100) SVD(r=5) values: 0.19999999999999998

- `Best Average HDBSCAN(min_cluster_size=100)  NMF(r=20) values: 0.19999999999999998`
- `Best Average HDBSCAN(min_cluster_size=100) UMAP(n_components=200) values:  0.44708136795768316`
- `Best Average HDBSCAN(min_cluster_size=100)  values: 0.19999999999999998`
- `Best Average Agglomerative(n_clusters=20) Clustering values: 0.34378284551429156`
- `Best Average K-means(k=50) values:  0.33907105823854683`

Among all the models evaluated the best average clustering metric score is obtained for K-Means clustering (cluster size=20) with UMAP cosine reduction(r=20) at `0.5565895415418738`

***UMAP with cosine and Agglomerative with Ward linkage have been used for all combinations above.*

QUESTION 18: Extra credit: If you can find creative ways to further enhance the clustering performance, report your method and the results you obtain.

Answer 18:

We tried to apply the technique of **normalisation** on the best model obtained in Q17 above before feeding to UMAP and get an improvement in accuracy above results in Q17.

Part2: Deep learning and Clustering of Image data [1], [2], [3]

QUESTION 19: In a brief paragraph discuss: If the VGG network is trained on a dataset with perhaps totally different classes as targets, why would one expect the features derived from such a network to have discriminative power for a custom dataset?

Answer 19:

Discriminative power refers to a test's ability to differentiate between multiple groups being evaluated. Transfer learning is a technique in machine learning where knowledge gained from solving one problem is utilized to address a related but different problem. In transfer learning, the base network is trained on a base dataset and task, and then the learned features are transferred to a target network to train it on a target dataset and task. This approach is effective if the features are general and applicable to both base and target tasks, rather than specific to the base task. If the tasks are similar enough, the later-layer features of the network should work well for the new task. The ImageNet dataset includes 1000 classes, including the "daisy" class which is present in both ImageNet and the Tf_Flowers dataset. The network can learn about edges and corners of various flowers and plants present in the ImageNet dataset, and use this knowledge to classify flowers in the Tf_Flowers dataset. Imagenet is a good representation for classifying the Tf_Flowers dataset due to the presence of several flowers and plants in it.

QUESTION 20: In a brief paragraph explain how the helper code base is performing feature extraction.

Answer 20:

The VGG16 model comes with pre-trained weights from the ImageNet dataset. It consists of a series of convolutional layers followed by one or multiple dense layers. The first part of the model, up until the final max pooling layer (labeled 7 x 7 x 512), is used for feature extraction while the rest of the network is used for classification. To use the model, the input image must be resized to the model's expected size of 224x224 and transformed into a 4D NumPy array with dimensions [samples, rows, columns, channels]. Only one sample is used. The pixel values of the image must also be scaled appropriately for use with the VGG model.

The VGG16 model takes in an image of fixed size, 224 x 224, in RGB format as its input. The image is then processed through a series of convolutional (conv) layers, using filters with a small receptive field of 3x3. In some configurations, 1x1 convolution filters are also utilized to perform a linear transformation on the input channels. The convolution stride is set at 1 pixel, with the input to the conv layer padded to preserve the spatial resolution, i.e. the padding is 1 pixel for 3x3 conv layers. The image is then passed through five max-pooling layers, which follow some of the conv layers, to perform spatial pooling. This is done over a 2x2 pixel window with a stride of 2.

After the convolutional layers, the VGG16 model has three fully-connected (FC) layers. The first two FC layers have 4096 channels each, while the third FC layer performs 5-way classification for the tf_flowers dataset and contains 5 channels, one for each class. The final layer is a soft-max layer.

QUESTION 21: How many pixels are there in the original images? How many features does the VGG network extract per image; i.e what is the dimension of each feature vector for an image sample?

Answer 21:

The size of original images varies for each image like it is 263*320*3 for one image in daisy class.

The dataset consists of 3670 images, which are resized to 150,5286 pixels in size (224 x 224 x 3), with the width and height being equal to 224 and the number of RGB color channels being 3. The features extracted by VGG from each image in the final layer of the network are equal to 25,088. However, these features are then processed through a fully connected layer to reduce their number to 4096 per image. Thus, each feature vector for an image sample is of dimension (4096,).

QUESTION 22: Are the extracted features dense or sparse? (Compare with sparse TF-IDF features in text.)

Answer 22:

The features extracted from the image have a high density and close intervals, as shown by the fact that the 4096 features have no zero values when using sum of all zero values in the feature matrix. This indicates that the **extracted features are dense** rather than sparse.

In contrast, the TF-IDF features in text data tend to be sparse, as the vocabulary size across all documents is much larger than the words present in any individual document. Although adjusting the min_df value can increase the density, it remains relatively sparse compared to the extracted image features.

QUESTION 23: In order to inspect the high-dimensional features, t-SNE is a popular off-the-shelf choice for visualizing Vision features. Map the features you have extracted onto 2 dimensions with t-SNE. Then plot the mapped feature vectors along x and y axes. Colour-code the data points with ground-truth labels. Describe your observation.

Answer 23:

Mapping the extracted features onto 2 dimensions using t-SNE gives us the below plot with colour coding based on ground truth labels.

Fig25: Scatter Plot for mapped feature vectors for t-SNE

t-SNE refers to T-Distributed Stochastic Neighbour Embedding, which is a technique used for nonlinear reduction of dimensions. This method is unsupervised and non-parametric, resulting in improved clustering of the given data. t-SNE preserves the local similarities only. In the t-SNE scatter plot above, distinct clusters can be observed. t-SNE has the ability to uncover the structure in the data that other dimensionality reduction methods like PCA are unable to find.

QUESTION 24: Report the best result (in terms of rand score) within the table below. For HDBSCAN, introduce a conservative parameter grid over min cluster size and min samples.

Module	Alternatives	Hyperparameters
Dimensionality Reduction	None	N/A
	SVD	r = 50
	UMAP	n_components = 50
	Autoencoder	num_features = 50
Clustering	K-Means	k = 5
	Agglomerative Clustering	n_clusters = 5
	HDBSCAN	min_cluster_size & min_samples

Answer 24:

We tried various combinations of dimensionality reduction and clustering with parameter grid for HDBSCAN, as in the below table:

Module	**Alternatives**	**Hyper-parameters**
Dimensionality Reduction	None	N/A
	SVD	r=50
	UMAP	n_components=50, distance metric= cosine
	Auto encoder	num_features=50
Clustering	K-Means	k=5
	Agglomerative Clustering	n_clusters=5
	HDBSCAN	min_cluster_size=[2,3,5,7,9,11] min_samples=[15,30,45]

Now to compare the various combinations of models we compare the Adjusted Rand Index Score for all the possible iterations:

Dimensionality Reduction	Clustering	Adjusted Rand Index Score
None	K-Means	0.1961
	Agglomerative Clustering	0.1886
SVD	K-Means	0.1933
	Agglomerative Clustering	0.2173
UMAP	**K-Means**	**0.4666**
	Agglomerative Clustering	0.4540
Auto encoder	K-Means	0.2118
	Agglomerative Clustering	0.2048

We show the results for **HDBSCAN Clustering** in the below table for various values of hyper-parameters:

Dimensionality Reduction	Hyper-parameters		Adjusted Rand Index Score
	min cluster size	min samples	
None	2	15	-0.0018
		30	-0.0008
		45	0.0000
	3	15	-0.0018
		30	0.0000
		45	0.0000
	5	15	0.0000
		30	0.0000
		45	0.0000
	7	15	0.0000
		30	0.0000
		45	0.0000
	9	15	0.0000
		30	0.0000
		45	0.0000
	11	15	0.0000
		30	0.0000
		45	0.0000
SVD	2	15	0.0133
		30	0.0095
		45	0.0061
	3	15	0.0096
		30	0.0000
		45	0.0000
	5	15	0.0000
		30	0.0000
		45	0.0000
	7	15	0.0000
		30	0.0000
		45	0.0000
	9	15	0.0000
		30	0.0000
		45	0.0000
	11	15	0.0000
		30	0.0000
		45	0.0000
UMAP	2	15	0.0949

		30	0.0949
		45	0.0949
	3	15	0.0949
		30	0.0949
		45	0.0949
	5	15	0.0949
		30	0.0949
		45	0.0949
	7	15	0.0949
		30	0.0949
		45	0.0949
	9	15	0.0949
		30	0.0949
		45	0.0949
	11	15	0.0949
		30	0.0949
		45	0.0949
Auto encoder	2	15	0.0089
		30	0.0081
		45	0.0077
	3	15	-0.0010
		30	0.0000
		45	0.0000
	5	15	-0.0001
		30	0.0000
		45	0.0000
	7	15	0.0000
		30	0.0000
		45	0.0000
	9	15	0.0000
		30	0.0000
		45	0.0000
	11	15	0.0000
		30	0.0000
		45	0.0000

From the above tables for comparing the Average Rand Index Score of various combination of dimensionality reduction and clustering we observe that the **UMAP with K-Means Clustering gives best ARI score of 0.466**.

These results are similar to what we observed in the textual data where the best combination was UMAP with K-means clustering too. The results can be summarised as:

- The sparse representation method is not as effective as it should be because the dataset is too large, which leads to a lot of noise and makes it challenging for any algorithm to accurately cluster or classify the data points. It hence has lowest ARI Scores.
- The performance of SVD is not ideal for this task, while UMAP outperforms all other feature reduction techniques in terms of ARI scores. UMAP tries to preserve the information entropy and nonlinear dependencies besides the global structure of data, which helps in clustering the data, while SVD is linear projections that are susceptible to outliers and noise.
- The hierarchical clustering methods and density-based methods generally do not work well with large datasets and high dimensions. HDBSCAN is not effective in this case due to the varying density of the clusters and the high dimensionality of the data. Agglomerative clustering provides better results because the clusters are of equal size.
- K-means performs well because the densities of the various clusters are different and the sizes are equal. K-means also scales well to large datasets, so has highest ARI among clustering algorithms.

QUESTION 25: Report the test accuracy of the MLP classifier on the original VGG features. Report the same when using the reduced-dimension features (you have freedom in choosing the dimensionality reduction algorithm and its parameters). Does the performance of the model suffer with the reduced-dimension representations? Is it significant? Does the success in classification make sense in the context of the clustering results obtained for the same features in Question 24.

Answer 25:

On the original VGG features, we get an average test accuracy of the MLP classifier around **91.5%.**

Now, after using the best model, found by Grid Search on the various dimensionality reduction techniques from previous Q24, we get the accuracy of MLP classifier as below:

Feature Reduction Technique	Test Accuracy of MLP classifier
UMAP with n_components=25	84.64%
UMAP with n_components=50	85.73%
UMAP with n_components=100	83.22%
SVD with r=25	89.43%
SVD with r=50	89.98%
SVD with r=100	91.07%

From the above table we can conclude that the **accuracy drops by using feature reduction** techniques.

This is because the neural network uses the features that may seem redundant and removing these features results in a loss of valuable information necessary for the classification. Unlike textual data, the image features are densely packed, so losing these features has a major impact on the accuracy. As shown, the **reduction in accuracy is significant** for most feature reduction techniques.

The classification method performs better than the clustering method on the tf flowers dataset, which makes sense as supervised learning with outputs is expected to perform better than an unsupervised problem. Additionally, the accuracy measurement for clustering is different from classification accuracy, the adjusted Rand index, which measures the similarity between the clusters and is evaluated between pairs of clusters. This is not directly comparable to classification accuracy.

References:

[1] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. arXiv preprint arXiv:1409.1556.

[2]Deng, J., Dong, W., Socher, R., Li, L. J., Li, K., & Fei-Fei, L. (2009, June). Imagenet: A large-scale hierarchical image database. In 2009 IEEE conference on computer vision and pattern recognition (pp. 248-255). Ieee.

[3]https://www.tensorflow.org/datasets/catalog/tf_flowers