Project 2: Data Representations and Clustering

Hexi Meng (406200552),Zhanhong Liu(206152835)

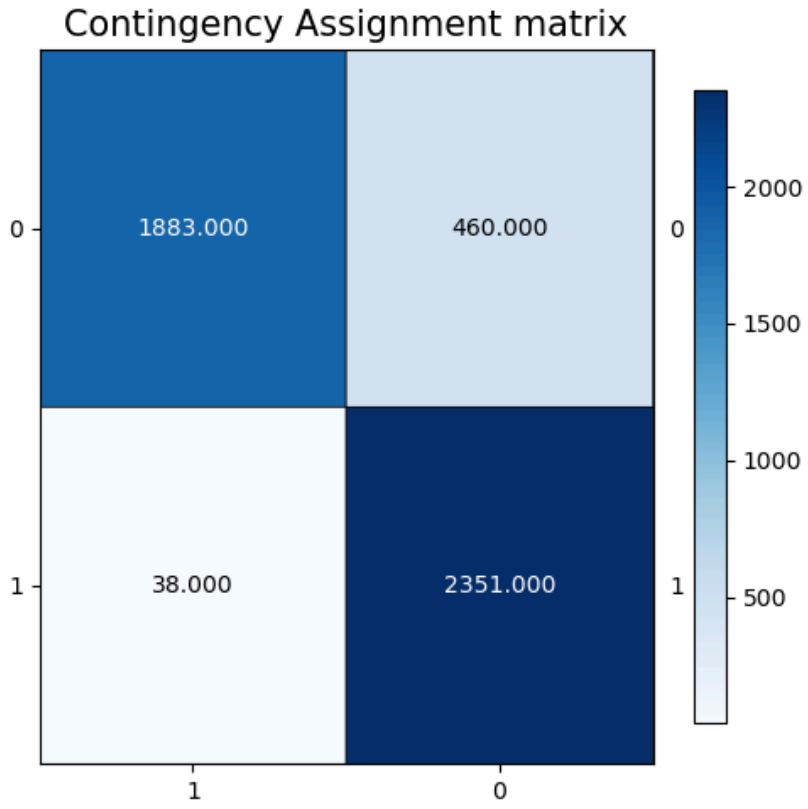
- **Part 1 - Clustering on Text Data**

Question 1

The dimension of the TF-IDF matrix is 4732 X 17131

Question 2

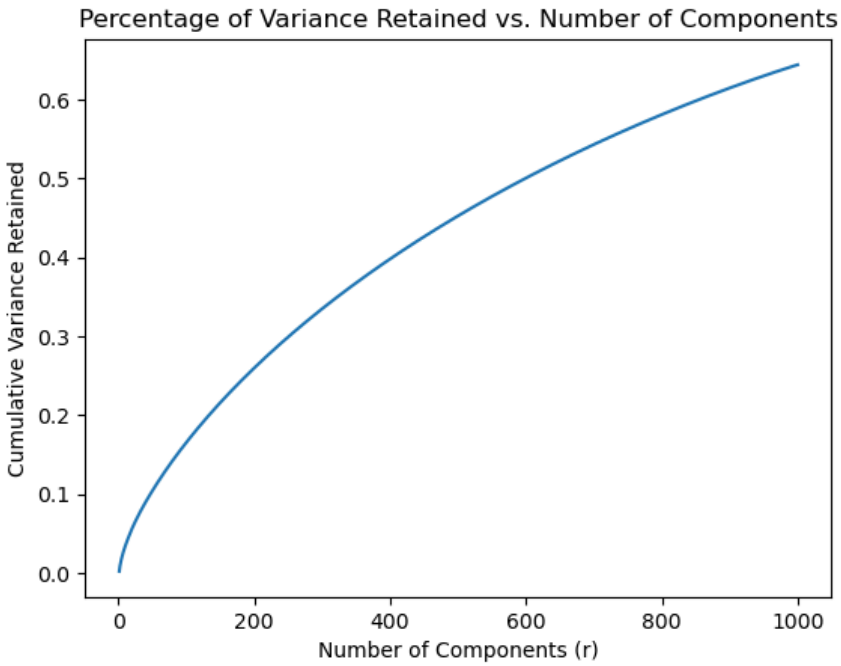
Contingency matrix:



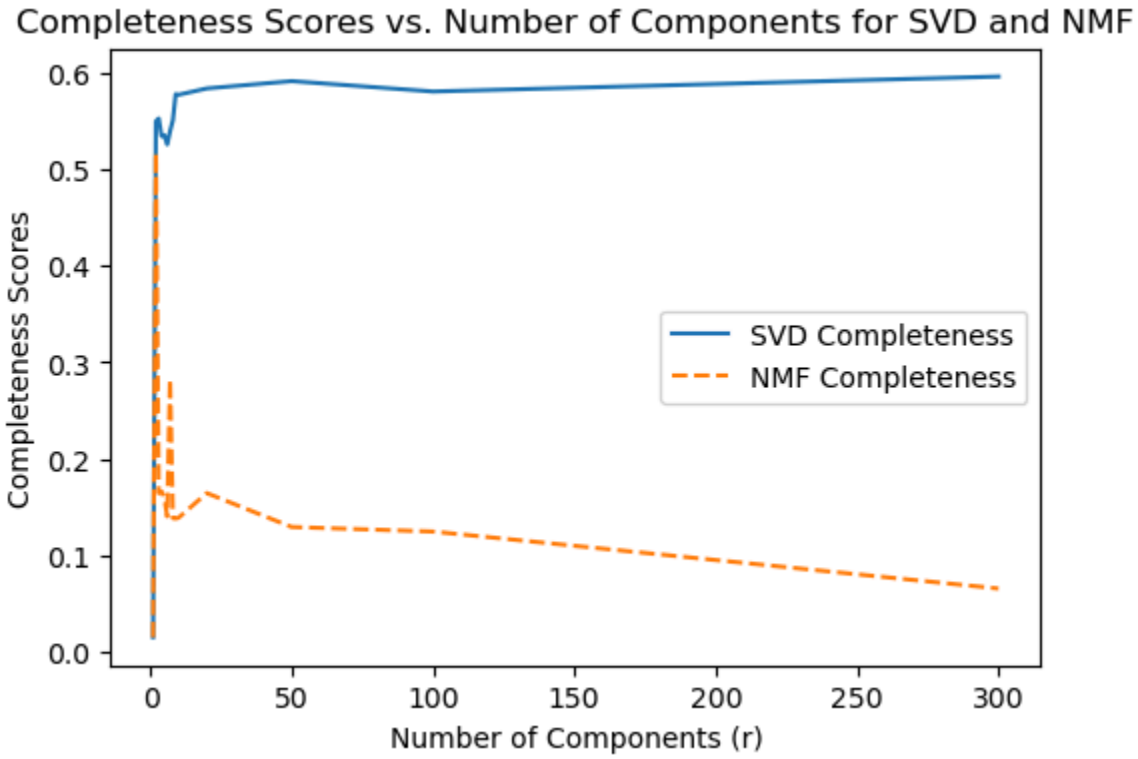
Question 3

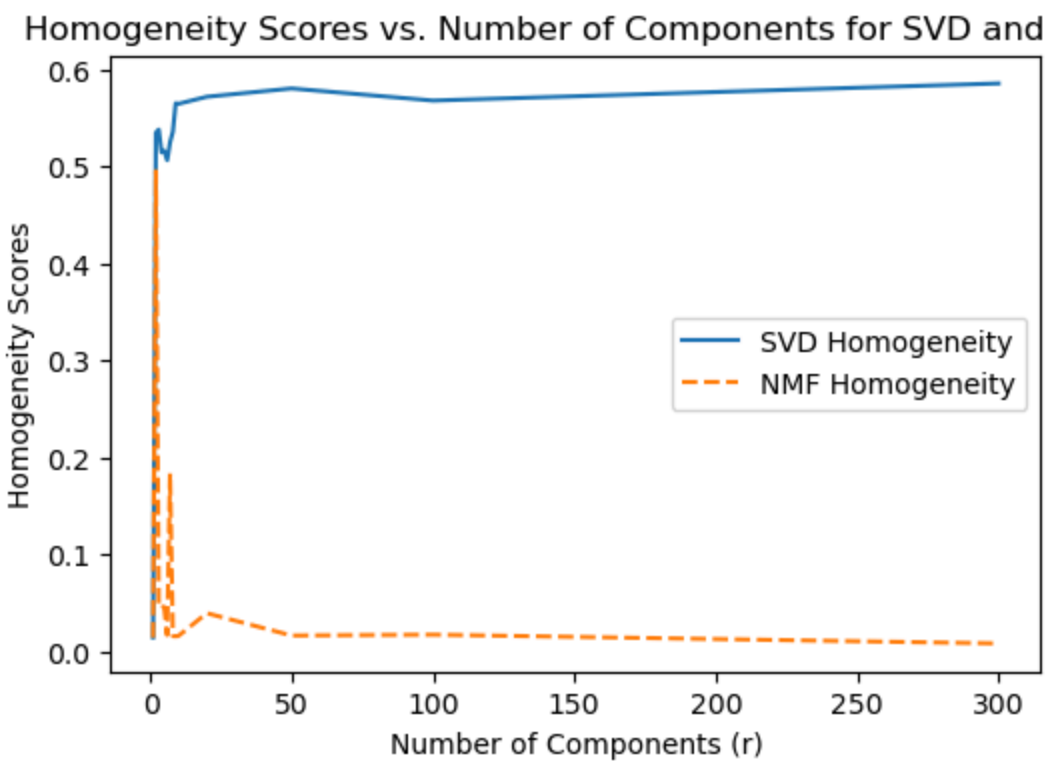
```
  Contingency Table:
[[ 460 1883]
 [2351   38]]
Clustering Measures:
Homogeneity: 0.561109514051486
Completeness: 0.5758533836929259
V-measure: 0.5683858513410838
Adjusted Rand Index: 0.6232608800818095
Adjusted Mutual Information Score: 0.5683191657107082
```

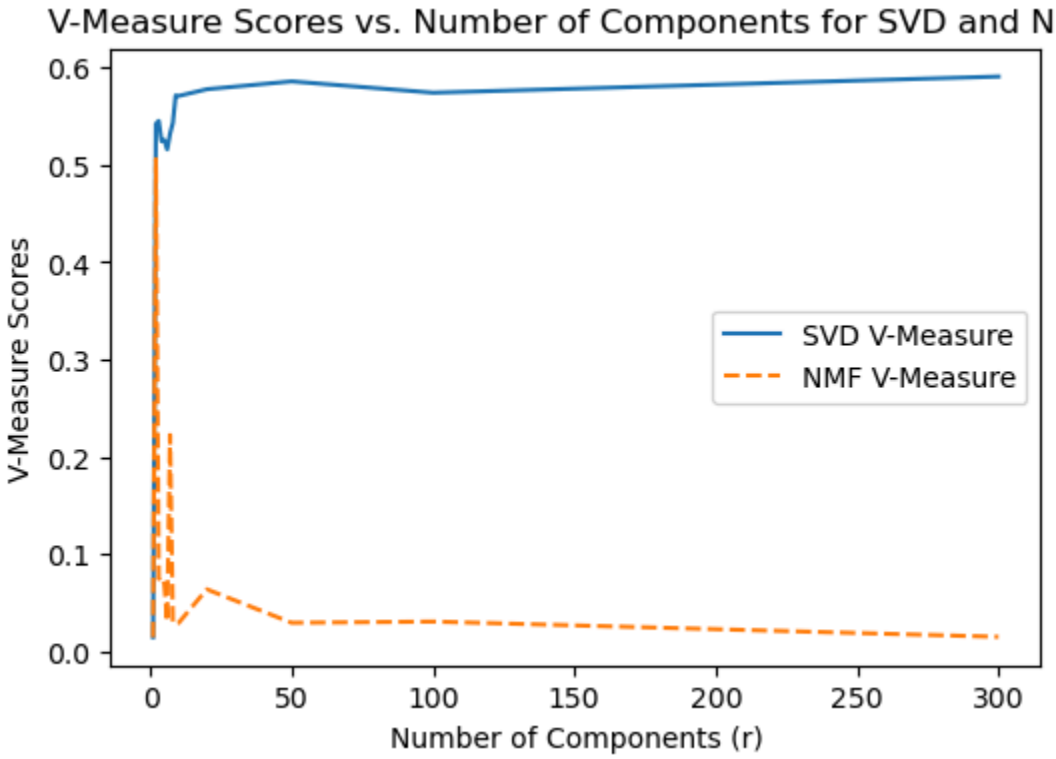
Question 4

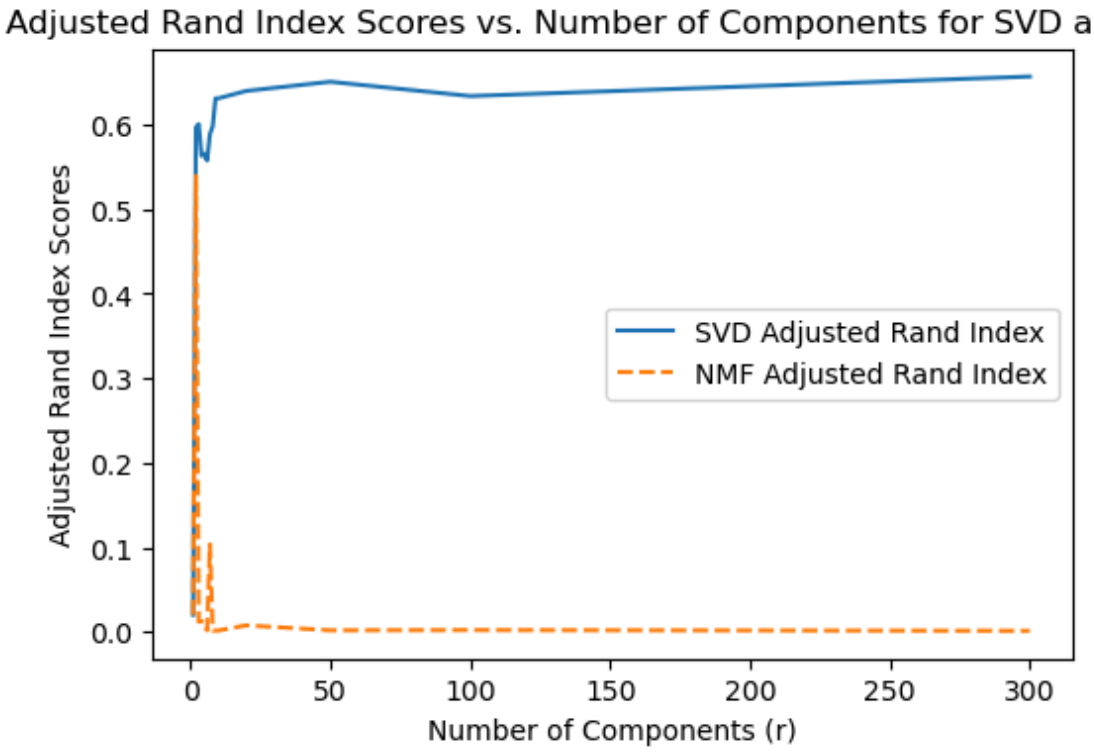

Percentage of Variance Retained vs. Number of Components
Cumulative Variance Retained
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0
200
400
600
800
1000
Number of Components (r)

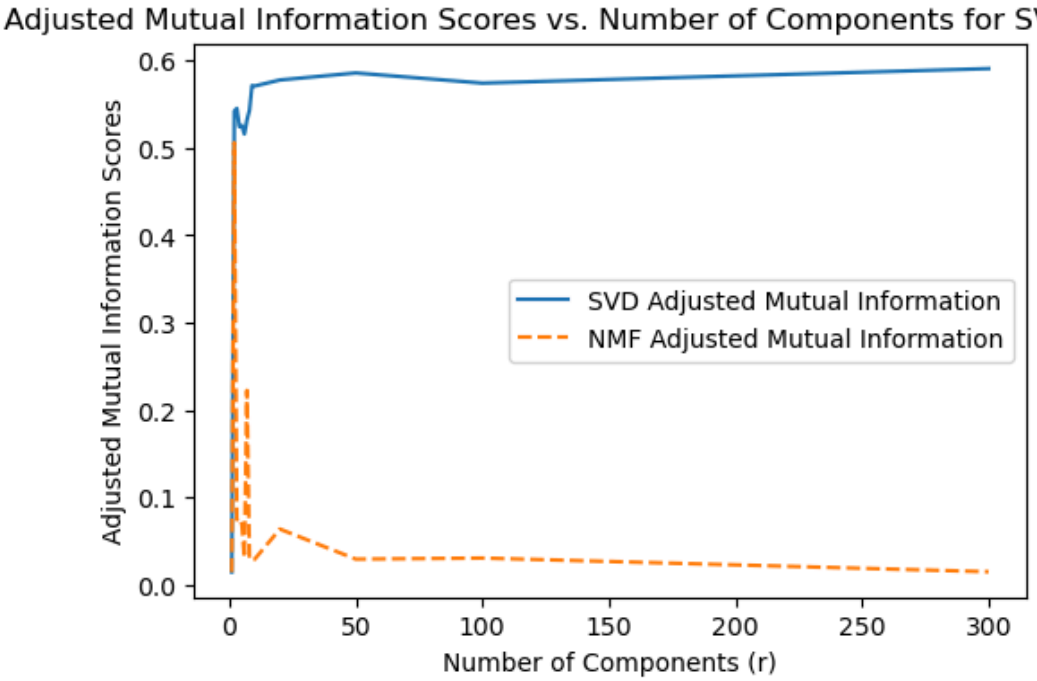
Question 5

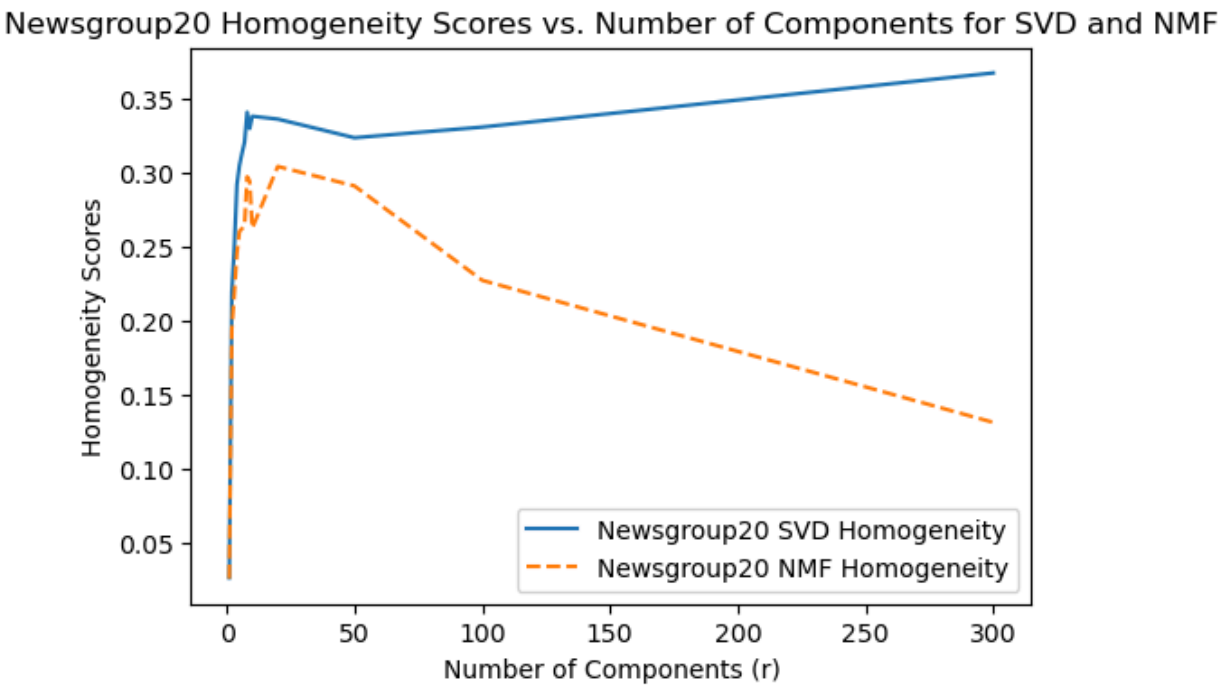

Completeness Scores vs. Number of Components for SVD and NMF
Completeness Scores
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0
50
100
150
200
250
300
Number of Components (r)
SVD Completeness
NMF Completeness

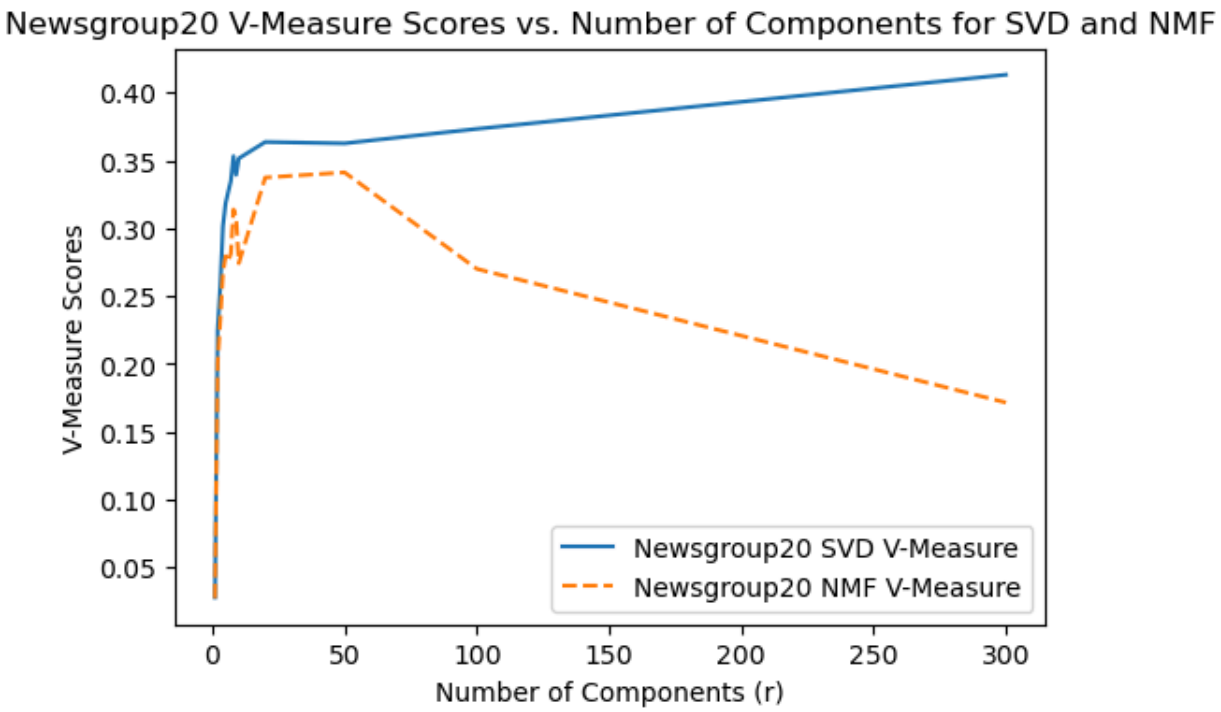

Homogeneity Scores vs. Number of Components for SVD and NMF
Homogeneity Scores
Number of Components (r)
SVD Homogeneity
NMF Homogeneity
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0
50
100
150
200
250
300

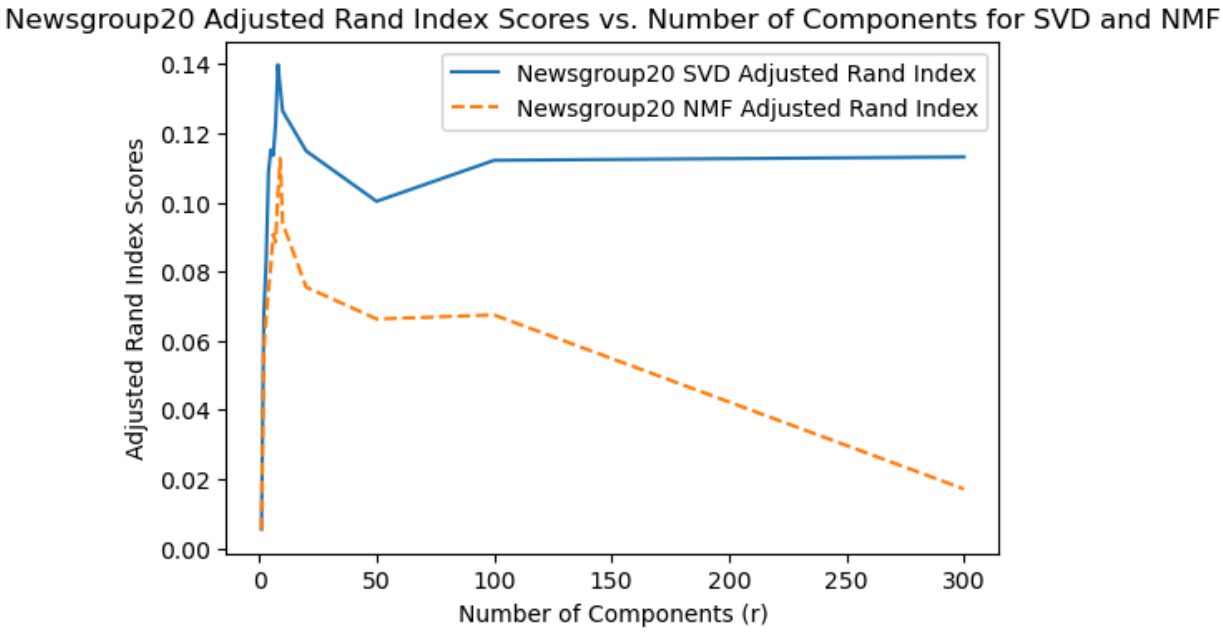

V-Measure Scores vs. Number of Components for SVD and NMF
V-Measure Scores
Number of Components (r)
SVD V-Measure
NMF V-Measure
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0
50
100
150
200
250
300

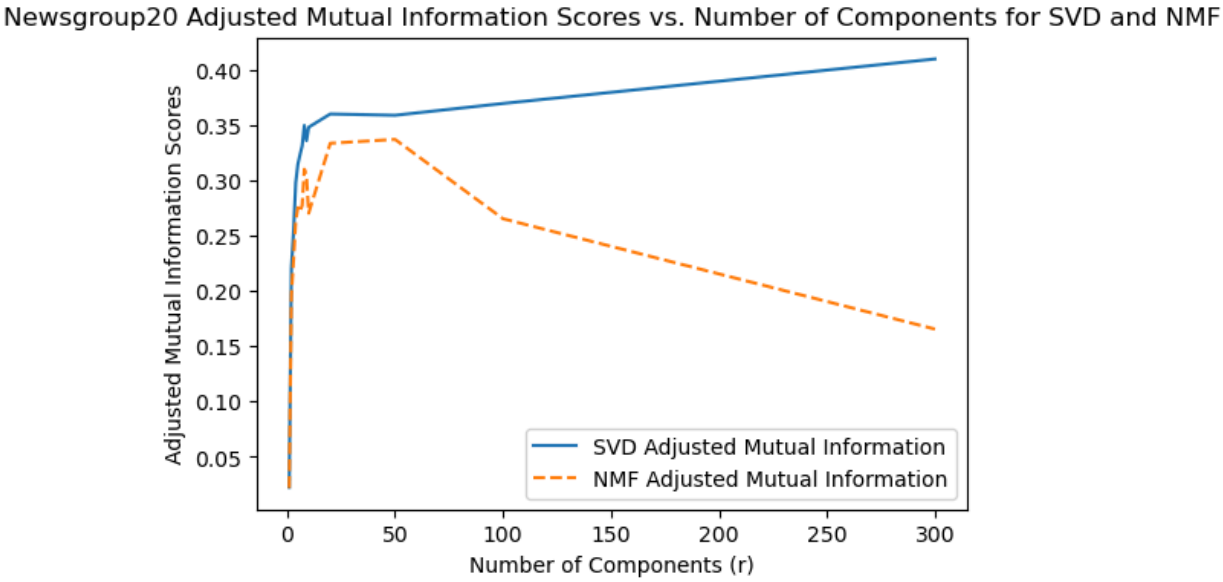

Adjusted Rand Index Scores vs. Number of Components for SVD and NMF
Adjusted Rand Index Scores
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0
50
100
150
200
250
300
Number of Components (r)
SVD Adjusted Rand Index
NMF Adjusted Rand Index


Adjusted Mutual Information Scores vs. Number of Components for SVD and NMF
Adjusted Mutual Information Scores
0.0
0.1
0.2
0.3
0.4
0.5
0.6
0
50
100
150
200
250
300
Number of Components (r)
SVD Adjusted Mutual Information
NMF Adjusted Mutual Information


Newsgroup20 Homogeneity Scores vs. Number of Components for SVD and NMF
Homogeneity Scores
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0
50
100
150
200
250
300
Number of Components (r)
Newsgroup20 SVD Homogeneity
Newsgroup20 NMF Homogeneity


Newsgroup20 V-Measure Scores vs. Number of Components for SVD and NMF
V-Measure Scores
0.40
0.35
0.30
0.25
0.20
0.15
0.10
0.05
0
50
100
150
200
250
300
Number of Components (r)
Newsgroup20 SVD V-Measure
Newsgroup20 NMF V-Measure





Question 6

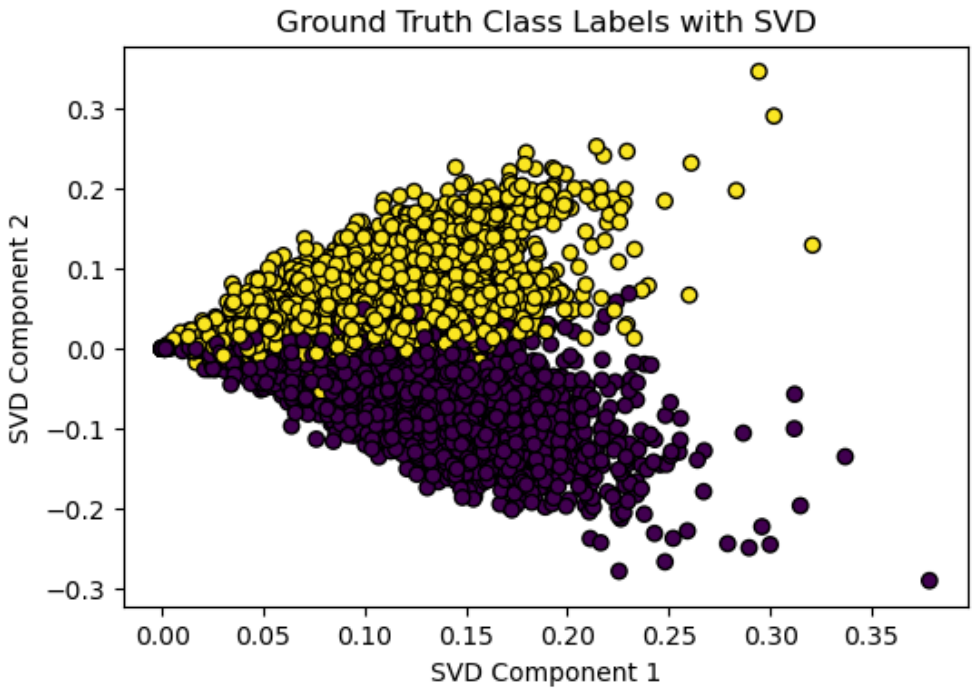
When r is large the dimensions of performa clustering is high (the Euclidean distance is not a good metric in high-dimensional space due to the curse of dimensionality)therefore the points are now have similar distance to each other which makes it harder to perform clustering.When r

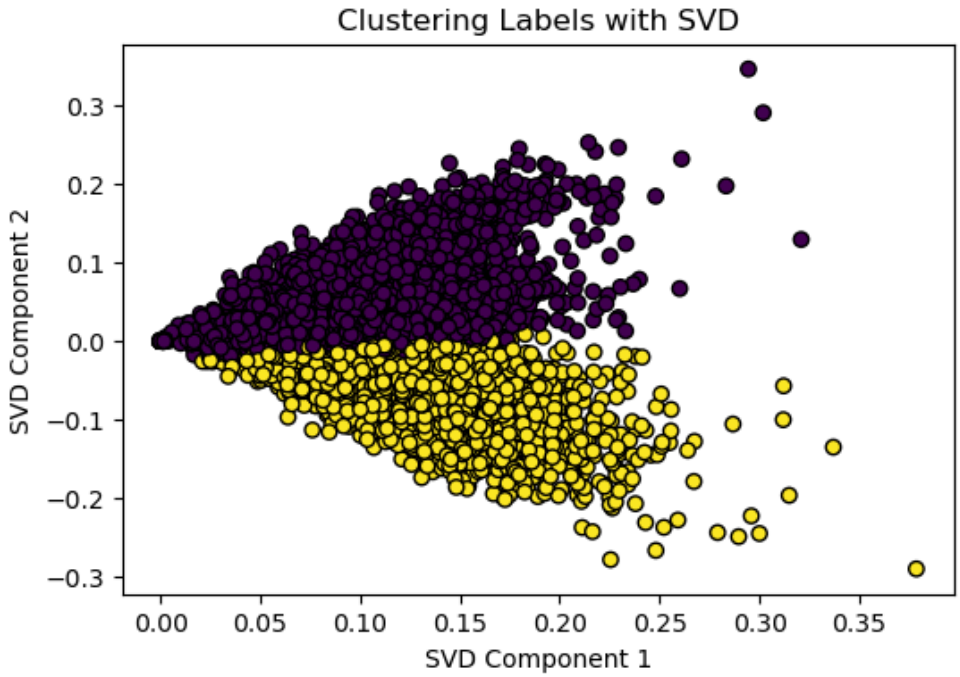
is small, lots of information are lost when projecting TF-IDF vector into lower space.So it shows a non-monotonic behavior.

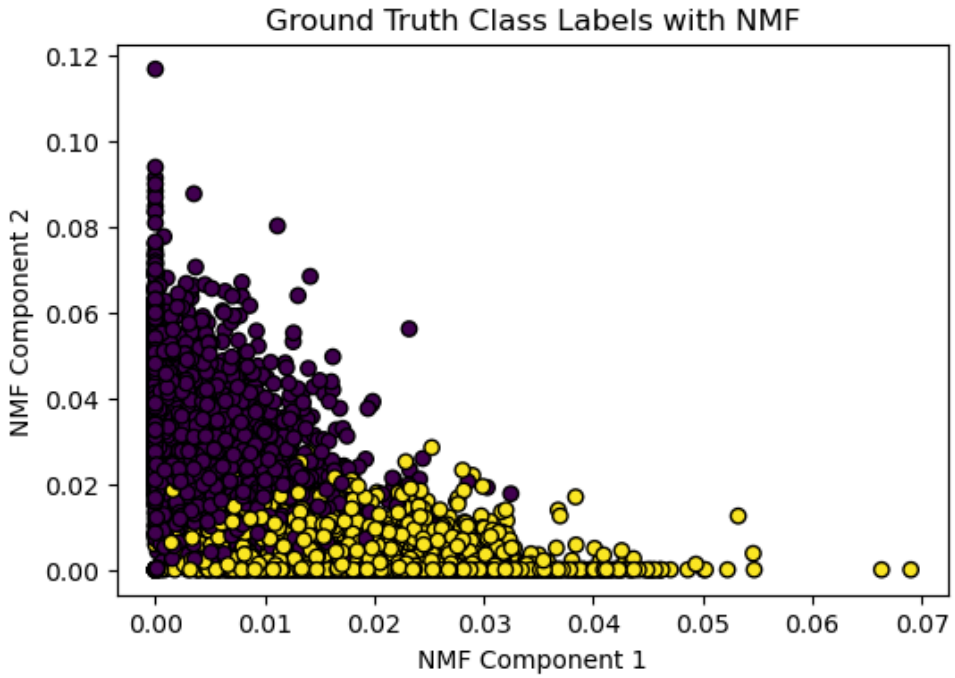
Question 7

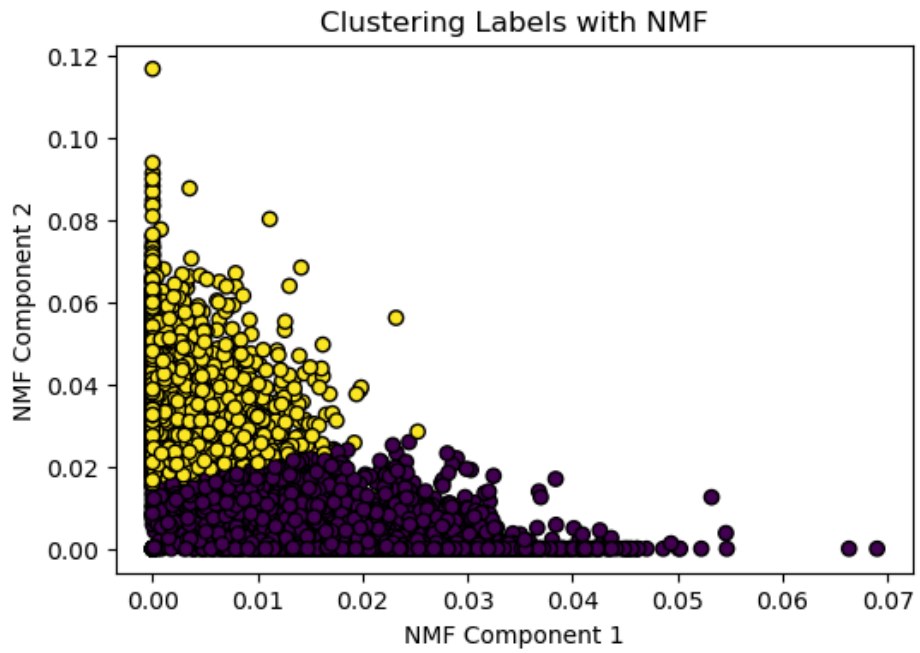
No, because they result in lower Adjusted Mutual Information Score comparing to question 3

Question 8




Clustering Labels with SVD
SVD Component 2
SVD Component 1
0.3
0.2
0.1
0.0
−0.1
−0.2
−0.3
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35


Ground Truth Class Labels with NMF
NMF Component 2
NMF Component 1
0.12
0.10
0.08
0.06
0.04
0.02
0.00
0.00
0.01
0.02
0.03
0.04
0.05
0.06
0.07



Question 9

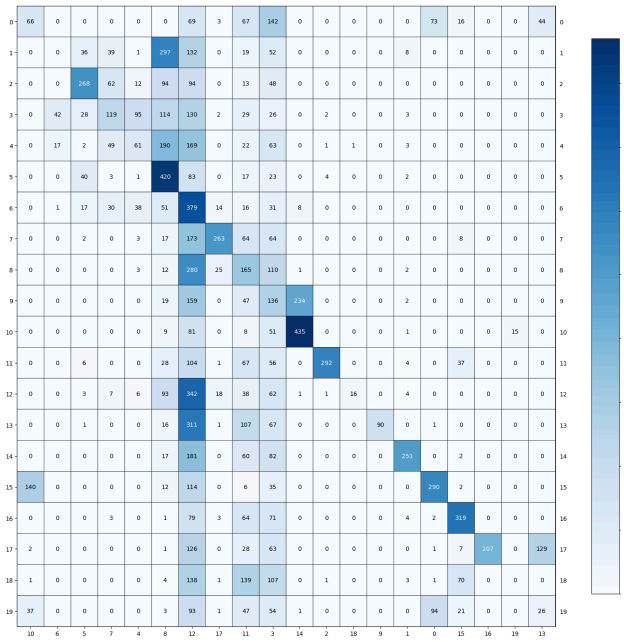
-Most of the labels are successfully separated but the edges are not very clear,Most predicted labels are true.

-Many points are at the zero boundary for NMF because it has no negative values, and for SVD , different labels take the equal proportion of the space and their distributions are symmetric.

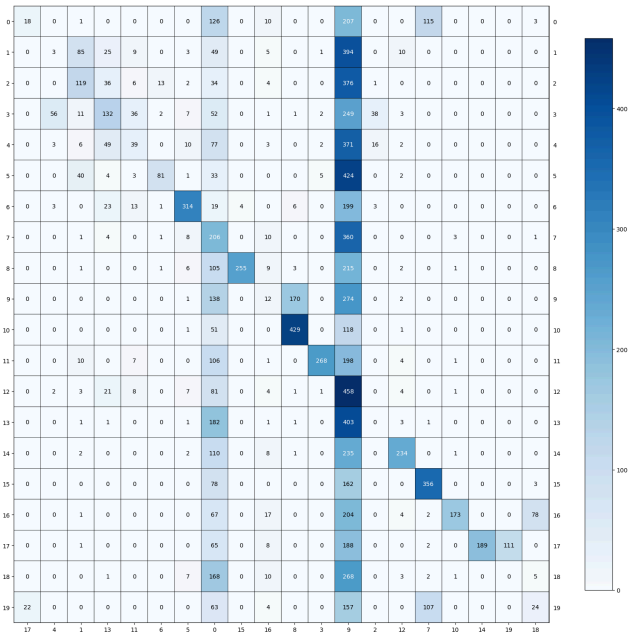
-the SVD distribution seems ideal because there's a clear spherical distribution therefore linearly separable ,however for NMF the distribution is not spherical therefore it is not ideal for k mean clustering.

(next page)

Question 10



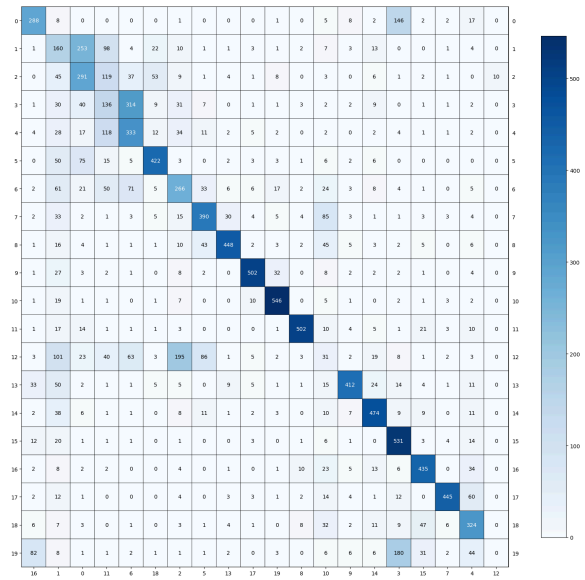
SVD of Newsgroup20 with the Optimal 'r' (r = 300) Clustering Measures:
Homogeneity: 0.33093212252249127
Completeness: 0.4087992685411551
V-measure: 0.3657673886988702
Adjusted Rand Index: 0.11435006625961701
Adjusted Mutual Information Score: 0.36188797365088676



NMF of Newsgroup20 with the Optimal 'r' (r = 50) Clustering Measures:
Homogeneity: 0.2760010367197732
Completeness: 0.42097610555511633
V-measure: 0.33341076635089767
Adjusted Rand Index: 0.05119369194370699
Adjusted Mutual Information Score: 0.3289290465346211

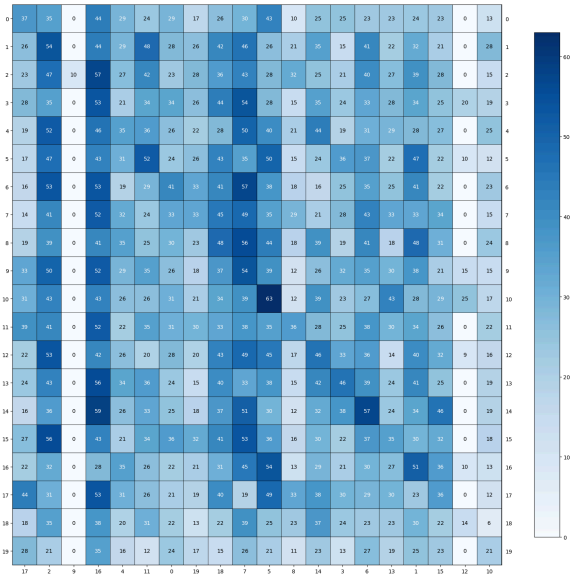
Question 11

Cosine UMAP with n_component = 5 Clustering Measures:



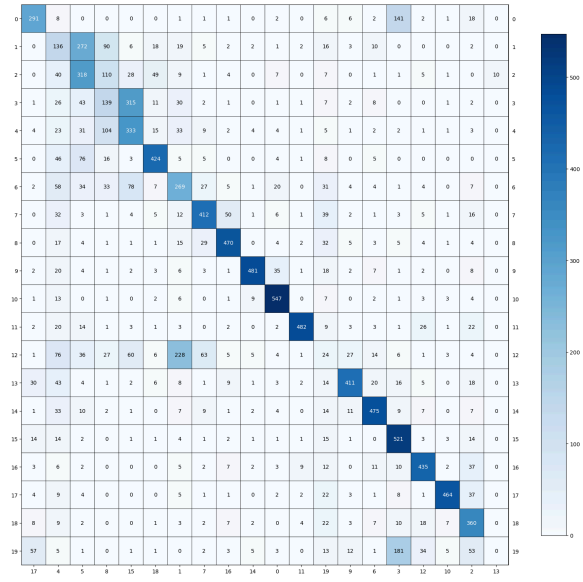
Homogeneity: 0.5770667444770883
Completeness: 0.5903068377495534
V-measure: 0.583611708006879
Adjusted Rand Index: 0.45677957879668624
Adjusted Mutual Information Score: 0.5813344734016105

Euclidean UMAP with n_component = 5 Clustering Measures:



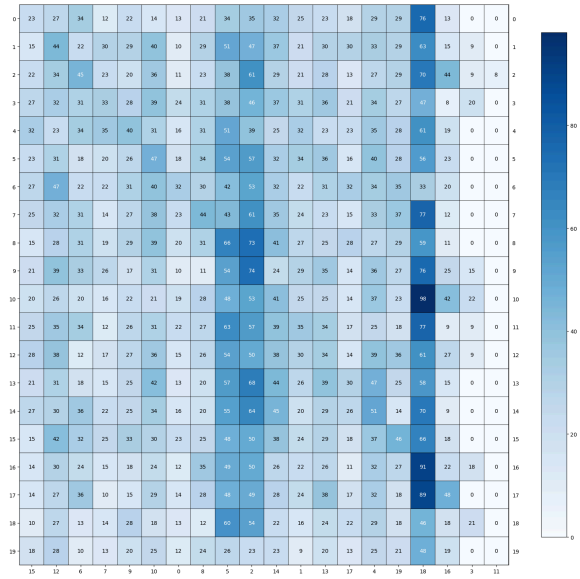
Homogeneity: 0.01275507650748453
Completeness: 0.01320749303732113
V-measure: 0.0129773429299724
Adjusted Rand Index: 0.0013720104823701118
Adjusted Mutual Information Score: 0.00753595935524953

Cosine UMAP with n_component = 20 Clustering Measures:



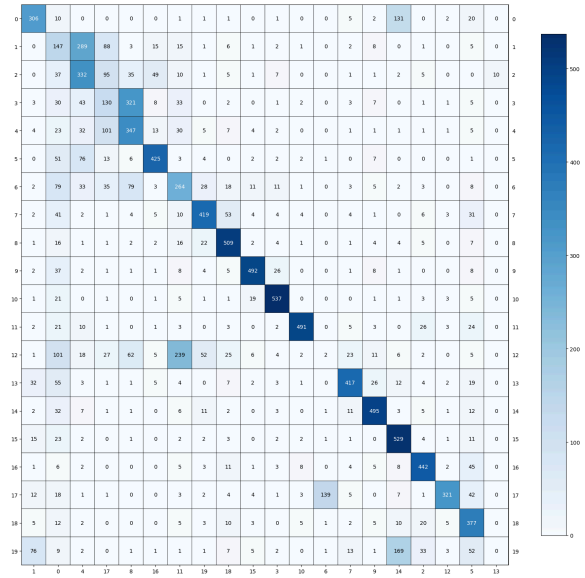
Homogeneity: 0.5734160062729408
Completeness: 0.5871000141322371
V-measure: 0.5801773339914031
Adjusted Rand Index: 0.46166778799191227
Adjusted Mutual Information Score: 0.5778801880434606

Euclidean UMAP with n_component = 20 Clustering Measures:



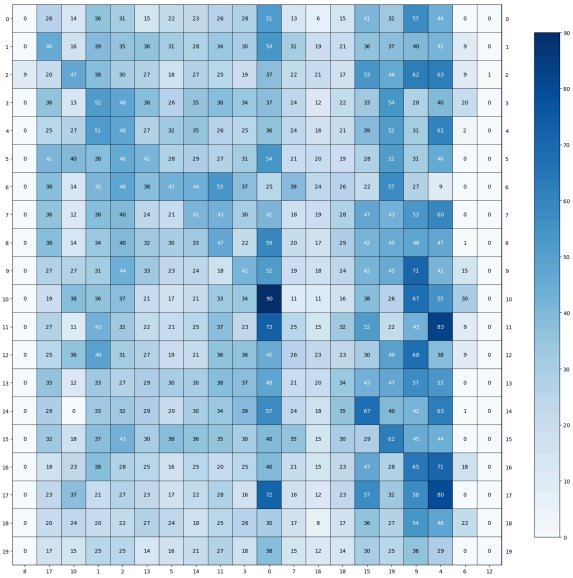
Homogeneity: 0.014085203915666752
Completeness: 0.014767848244392987
V-measure: 0.014418450621028912
Adjusted Rand Index: 0.0018522404383776554
Adjusted Mutual Information Score: 0.00897296704573849

Cosine UMAP with n_component = 200 Clustering Measures:



Homogeneity: 0.5833682475856476
Completeness: 0.6014797448522641
V-measure: 0.592285570726685
Adjusted Rand Index: 0.46209208109713695
Adjusted Mutual Information Score: 0.5900448729633321

Euclidean UMAP with n_component = 200 Clustering Measures:



Homogeneity: 0.01693875706137612
Completeness: 0.017938734668412944
V-measure: 0.01742441061363389
Adjusted Rand Index: 0.003113127036629564
Adjusted Mutual Information Score: 0.012160461251485016

Question 12

By visualize the contingency matrix we find out that the Cosine UMAP Clustering works better with weight are mainly distributed on the diagonal,suggesting that most clustering matches true label whereas the Euclidean UMAP has a relative scattered weight distribution and low performance matrices score.

Question 13

	sparse TF-IDF	SVD（r=300）	NMF(r = 50)	UMAP(cosine ,n=200)
Homogeneity	0.3436469263424915	0.3309321225224912 7	0.2760010367197732	0.5833682475856476
Completeness	0.4079307553523886	0.4087992685411551	0.4209761055551163 3	0.6014797448522641
V-measure	0.3757018853398814	0.3657673886988702	0.3334107663508976 7	0.5922855707266685
Adjusted Rand Index	0.1190587509207104	0.1143500662596170 1	0.0511936919437069 9	0.4620920810971369 5
Adjusted Mutual Information Score	0.3687670257855133	0.3618879736508867 6	0.3289290465346211	0.5900448729633321

Based on the performance we find out that Umap is the best approach

Question 14

```
Ward Clustering Measures:
Homogeneity: 0.013976375782589631
Completeness: 0.014937010997410577
V-measure: 0.014440735037853817
Adjusted Rand Index: 0.0015377327427691992
Adjusted Mutual Information Score: 0.008849482554577561

Single Clustering Measures:
Homogeneity: 0.01574796257757014
Completeness: 0.3479495687789924
V-measure: 0.030132163765726735
Adjusted Rand Index: 0.00011626602589998092
Adjusted Mutual Information Score: 0.020410685964017707
```

Question 15

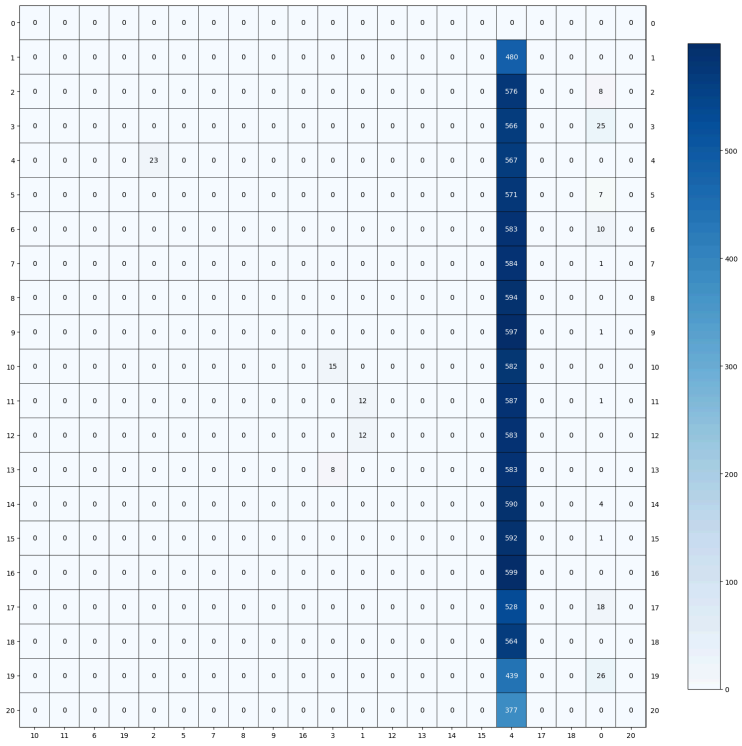
```
HDBScan Clustering Measures (min_cluster_size = 20):
Homogeneity: 0.008554711965201765
Completeness: 0.2670311623686569
V-measure: 0.016578314729074086
Adjusted Rand Index: 7.98755459160745e-05
Adjusted Mutual Information Score: 0.014131130347647838

HDBScan Clustering Measures (min_cluster_size = 100):
Homogeneity: 0.0006842273408473208
Completeness: 0.0020501908581130003
V-measure: 0.0010260293320235475
Adjusted Rand Index: 1.7327162541385735e-05
Adjusted Mutual Information Score: 0.0001825825629761095

HDBScan Clustering Measures (min_cluster_size = 200):
Homogeneity: 0.0006767333256027381
Completeness: 0.0020945449315391243
V-measure: 0.0010229563584904723
Adjusted Rand Index: 5.430188443566419e-05
Adjusted Mutual Information Score: 0.00017253267426471997
```

Question 16

contingency matrix（min_cluster_size = 20）



Question17

```
{
  'None': {
    'kmean': {
      10: 0.31634736393460405,
      20: 0.31634736393460405,
      50: 0.31634736393460405
    },
    'agglo': {
      20: 0.36960007528083527
    },
    'hdbscan': {
      100: 0.2,
      200: 0.2
    }
  },
  'SVD': {
    'kmean': {
      10: 0.31634736393460405,
      20: 0.31634736393460405,
      50: 0.31634736393460405
    },
    'agglo': {
```

```
      20: 0.33616216587869163
    },
    'hdbscan': {
      100: 0.2,
      200: 0.2
    }
  },
  'NMF': {
    'kmean': {
      10: 0.31634736393460405,
      20: 0.31634736393460405,
      50: 0.31634736393460405
    },
    'agglo': {
      20: 0.18694528574285818
    },
    'hdbscan': {
      100: 0.2,
      200: 0.2
    }
  },
  'UMAP': {
    'kmean': {
      10: 0.31634736393460405,
      20: 0.31634736393460405,
      50: 0.31634736393460405
    },
    'agglo': {
      20: 0.013596060188171496
    },
    'hdbscan': {
      100: 0.007802540854748778,
      200: 0.0029273012865734293
    }
  }
}
```

No dimensionality reduction and agglomerative Clustering has the highest reported performance metric, making it the best performer in this specific set of results.

- **Part 2 - Deep Learning and Clustering of Image Data**

Question 19

One might anticipate that features obtained from a VGG would exhibit strong discriminative capabilities for a custom dataset, owing to its robust ability to intricately capture and differentiate details across images. VGG, a Convolutional Neural Network (CNN) architecture, is designed to enhance network depth through the utilization of simple and uniform convolutional layers, contributing to its effectiveness. The increased depth allows VGG to systematically capture and organize a hierarchy of features, encompassing

a spectrum of complexities, ranging from fundamental edges and shapes to more elaborate textures and patterns. Another rationale for augmenting the depth lies in the recognition that visual data encompasses features at various levels of abstraction. Consequently, by emphasizing deeper networks, VGG is better positioned to encapsulate the essence of images and discern subtle distinctions within them.

Question 20

The helper code executes feature extraction through a sequence of steps. Initially, it loads the dataset. Subsequently, it instantiates an object of the 'FeatureExtractor' class, responsible for the actual feature extraction process. The code then transforms the dataset images to prepare them for the feature extraction method, storing them in a Dataloader. Lastly, the code iterates through the Dataloader, employing the 'FeatureExtractor' instance to extract features from each batch of images. These extracted features are then saved onto the disk. Notably, the 'FeatureExtractor' class utilizes the VGG16 model, leveraging its convolutional layers, average pooling, flattening, and the initial fully-connected layer to capture hierarchical image features from the dataset.

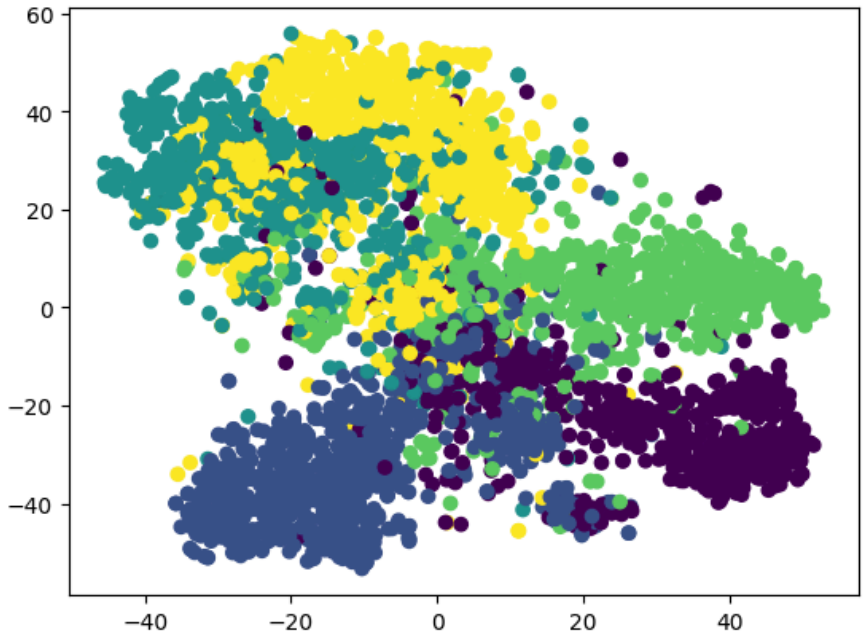
Question 21

The original images possess varying pixel dimensions, but they are uniformly resized to 224x224 pixels before undergoing feature extraction using VGG16. Following the feature extraction process, a feature vector of dimensions 4096 is generated for each image sample. In essence, the VGG16 network extracts 4096 features for every image, providing a comprehensive representation of the image content.

Question 22

The sparsity value I computed for the features extracted from VGG16 is 0.0, indicating high density. However, when evaluating the sparsity of TF-IDF features in text, the result is 0.9824, indicating significant sparsity.

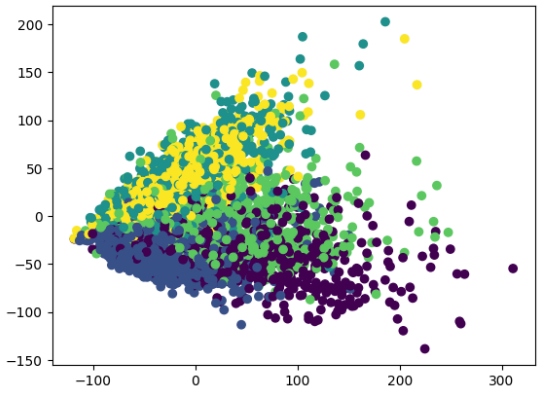
Question 23



It appears that there are 5 distinct classes or labels, as evidenced by the presence of 5 different colored points (cyan, yellow, blue, green, and purple) on the graph. The blue, purple, and green clusters exhibit clear boundaries, while the yellow and cyan clusters seem to be partially mixed. This suggests potential similarities between the yellow and cyan data points or labels. Additionally, if clusters were outlined, they would take on an oval shape, indicating that the data points within each cluster may display some degree of elongation or directionality.

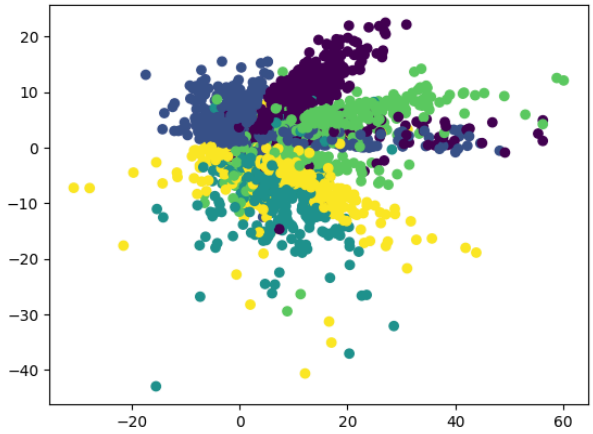
Helper plot

```
f_pca = PCA(n_components=2).fit_transform(f_all)
plt.scatter(*f_pca.T, c=y_all)
```



MLP Classifier

```
X_em =Autoencoder(2).fit_transform(f_all)
plt.scatter(*X_em.T, c=y_all)
```



Question 24

```
Rand score of clustering w/o dimensionality reduction:
{'kmeans': 0.19635566765122872, 'agglomerative': 0.2184499487113686, 'hdbscan':
(0.015014212771105666, {'min_cluster_size': 10, 'min_samples': 1})}
Rand score of clustering with SVD:
{'kmeans': 0.2098699434994347, 'agglomerative': 0.16422464508301501, 'hdbscan':
(0.02643295458142522, {'min_cluster_size': 10, 'min_samples': 1})}
Rand score of clustering with UMAP:
{'kmeans': 0.39981871984057604, 'agglomerative': 0.3862922170893482, 'hdbscan':
(0.4167541162586394, {'min_cluster_size': 100, 'min_samples': 1})}
Rand score of clustering with Autoencoder:
{'kmeans': 0.22650562888522663, 'agglomerative': 0.21755943468455324, 'hdbscan':
(0.11194370301529104, {'min_cluster_size': 100, 'min_samples': 5})}
```

The combination of dimensionality reduction and clustering with the best rand score is UMAP dimensionality reduction with n_components = 50 and HDBScan clustering with min_cluster_size = 100 & min_samples = 1. Their rand score is 0.4167541162586394.

Question 25

```
100%|██████████| 100/100 [00:10<00:00,  9.96it/s]

Original VGG features test accuracy:
0.8950953678474114

100%|██████████| 100/100 [00:06<00:00, 16.00it/s]
Reduced-dimension features (dimensionality reduction using UMAP with n_components =
50) test accuracy:
0.26430517711171664
```

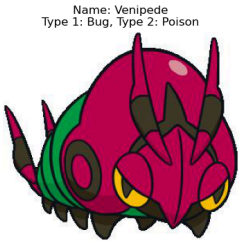
The MLP classifier achieved a test accuracy of 0.8950953678474114 on the original VGG features, whereas the test accuracy dropped significantly to 0.26430517711171664 when using reduced-dimension features (obtained through UMAP with n_components = 50). This decline in performance suggests that the dimensionality reduction process led to a loss of information, as the reduced-dimension features may not fully capture the essential patterns present in the data. The discrepancy in results aligns with the clustering outcomes discussed in Question 24. Clustering, an unsupervised machine learning process operating on unlabeled data, aims to identify patterns without prior knowledge of categories. Consequently, clustering algorithms can make reasonably accurate predictions even in the presence of information loss during dimensionality reduction. In contrast, classification, being a supervised machine learning process, heavily relies on labeled datasets for training. The substantial drop in accuracy after dimensionality reduction highlights the sensitivity of classification models to the loss of valuable information. Supervised learning depends on having access to accurate labels during training, and any loss of relevant features can significantly impact classification performance.

- **Part 3 - Clustering using both image and text**

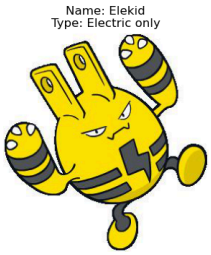
Question 26

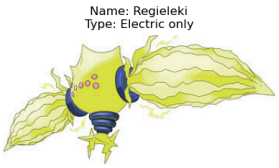
Testing queries results

The query: type: Bug

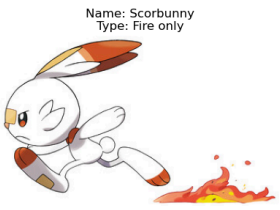


The query: electric type Pokemon

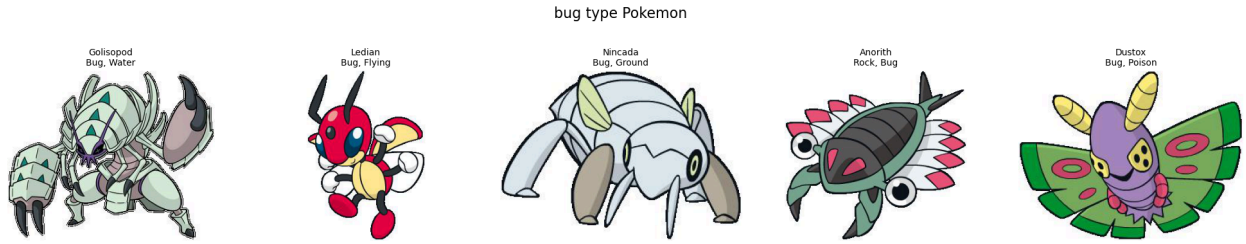




The query: Pokemon with fire abilities

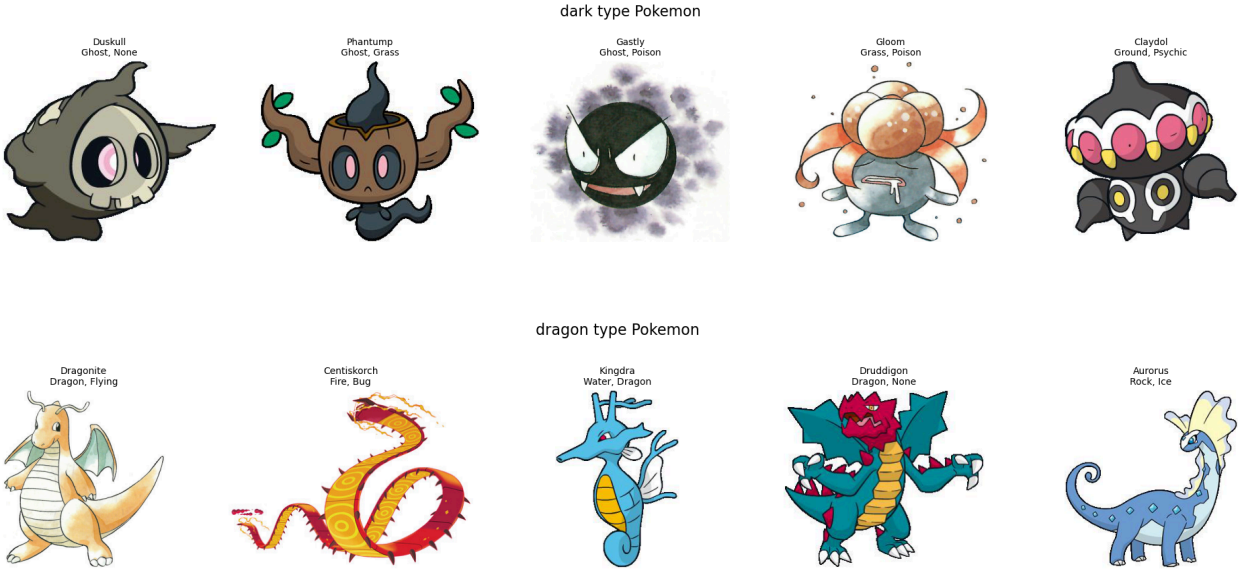


After testing these queries ,we found out that ”electric type Pokemon” and ”Pokemon with fire abilities” queries are better with zero identification error.So I used the queries “..... type Pokemon” to find the top five most relevant Pokemon for **type Bug, Fire and Grass**, and the result is shown below:



As a result ,it successfully selects the top 5 pokemon for each of the three types without misidentification. Our query works effectively.

Results for dark and dragon type



We can see that the CLIP interprets the dark as the color “dark” so it gives five pokemon which are in relative dark color but actually dark type. And for the dragon type, it interprets it as the morphological characteristics ‘dragon’ so it gives several false interpretations of the pokemon that have a dragon looking shape but not dragon types.

Question 27

Bergmite
Type 1: Ice, Type 2:

Bergmite's Top Predicted Types:
1: Ground (Probability: 0.20)
2: Poison (Probability: 0.20)
3: Electric (Probability: 0.21)
4: Water (Probability: 0.22)
5: Grass (Probability: 0.23)

Tapu Fini
Type 1: Water, Type 2: Fairy

Tapu Fini's Top Predicted Types:
1: Ghost (Probability: 0.19)
2: Poison (Probability: 0.20)
3: Fighting (Probability: 0.20)
4: Ground (Probability: 0.21)
5: Rock (Probability: 0.23)

Zangoose
Type 1: Normal, Type 2:

Zangoose's Top Predicted Types:
1: Electric (Probability: 0.20)
2: Dark (Probability: 0.20)
3: Ice (Probability: 0.21)
4: Flying (Probability: 0.21)
5: Fairy (Probability: 0.21)

Seadra
Type 1: Water, Type 2:

Seadra's Top Predicted Types:
1: Rock (Probability: 0.20)
2: Ground (Probability: 0.20)
3: Poison (Probability: 0.20)
4: Psychic (Probability: 0.21)
5: Ghost (Probability: 0.21)

Ralts
Type 1: Psychic, Type 2: Fairy

Ralts's Top Predicted Types:
1: Ghost (Probability: 0.21)
2: Rock (Probability: 0.21)
3: Ice (Probability: 0.21)
4: Dark (Probability: 0.21)
5: Steel (Probability: 0.21)

Toxel
Type 1: Electric, Type 2: Poison

Toxel's Top Predicted Types:
1: Ice (Probability: 0.19)
2: Psychic (Probability: 0.20)
3: Water (Probability: 0.21)
4: Electric (Probability: 0.21)
5: Ghost (Probability: 0.22)

Gliscor
Type 1: Ground, Type 2: Flying

Gliscor's Top Predicted Types:
1: Fighting (Probability: 0.19)
2: Electric (Probability: 0.19)
3: Fire (Probability: 0.19)
4: Ground (Probability: 0.20)
5: Flying (Probability: 0.22)

Meltan
Type 1: Steel, Type 2:

Meltan's Top Predicted Types:
1: Grass (Probability: 0.21)
2: Dark (Probability: 0.21)
3: Fighting (Probability: 0.21)
4: Water (Probability: 0.22)
5: Fire (Probability: 0.24)

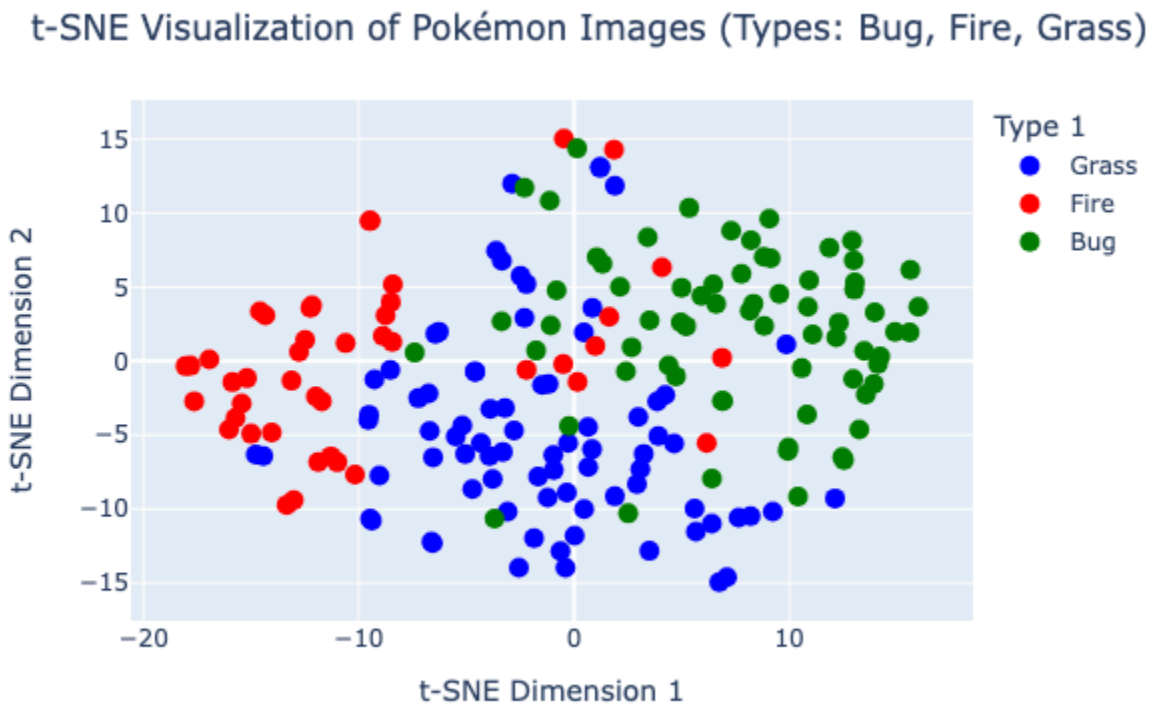
Herdier
Type 1: Normal, Type 2:

Herdier's Top Predicted Types:
1: Dragon (Probability: 0.22)
2: Electric (Probability: 0.22)
3: Water (Probability: 0.22)
4: Bug (Probability: 0.22)
5: Bug (Probability: 0.22)

Bewear
Type 1: Normal, Type 2: Fighting

Bewear's Top Predicted Types:
1: Psychic (Probability: 0.21)
2: Rock (Probability: 0.22)
3: Ground (Probability: 0.22)
4: Poison (Probability: 0.22)
5: Water (Probability: 0.23)

Question 28



The Bug-type (green) Pokémon appear to be more tightly clustered than the other types, which suggests that Bug-type Pokémon might have more in common with each other in the feature space used for this analysis.
The Fire-type (red) Pokémon are somewhat dispersed but still form a recognizable cluster, indicating some commonality but perhaps with more variation within the type than seen in the Bug-type.
The Grass-type (blue) Pokémon are the most dispersed, suggesting a higher degree of variation within this type in the feature space. They appear to intermingle more with the Fire and Bug types, indicating some shared characteristics across these types.
There are a few outliers of each type that are positioned far from their main clusters. These might be Pokémon that have unusual features not typical for their type or they could be dual-type Pokémon that share characteristics with other types.But in general the embedding model effectively interprets the Pokémon type based on the image.