Comparing Unsupervised Learning Algorithms

0. Data Setup (Load and Preprocessing)

Travel Review Dataset: Because distance-based algorithms would be used to cluster the data, all feature columns were normalized (to between 0-1) using sklearn MinMaxScaler[1,20] to avoid the dominance of large value attributes when calculating the distance. Whilst the feature values all ranged between 0-4, we found normalizing the data helped with the clustering outcome.

ICMLA Dataset: Clustering requires a distance measure between samples. Therefore, we had to pre-process the text data in each feature column. First, we use the natural language tool kit (nltk) library to perform tokenization by breaking down the text into sentence, and then word units. Then, we cleaned the data by removing all tokens containing non-letter characters using the re (regular expression) package, and used Snowball stemming to produce root forms[2]. We then convert them into a numerical representation by constructing an ordered vector of term frequency - inverse document frequency (TF-IDF) for each unique word in the corpus[2].

The resulted numerical data contained 105 feature columns, and such high dimensional data could cause storage, processing time and visualization problems. Consequently, the sklearn PCA[3] (*Figure 0.1*) was used to analyse how many features were important, then the number of features were reduced accordingly (eg, for 'combine' column, 31 features explained at least 90% of the variance in that column, so the number of features were reduced to 31).

In addition to the title, keywords and abstract columns, we created a 'combine' column that includes texts from those three columns. This allowed us to try different feature columns to ensure the best one was chosen for our clustering algorithm. Lastly, the gold cluster 'session' was converted into numbers (0-23) using the sklearn LabelEncoder[4] for model evaluation.

1. K-means Clustering

1.1. Travel Reviews Dataset

Using sklearn KMeans[5] (Euclidean) and manual k-means algorithm[2] (Manhattan), fifty eight models were trained on Dataset 1 using k=1-29 and two distance measures. The Euclidean distance[6], which measures the straight-line distance between two points, is the common metric to use for k-means. The Manhattan (or Cityblock) distance[6], which measures the absolute or distance in blocks between any two points in a city, was included as a comparison.

Because no gold clusters was provided for this dataset, the resulting models were evaluated using internal measures only. The elbow method[7], which observes the marginal reduction of the sum of within-cluster variances, was used to determine the appropriate number of clusters. The silhouette, Davies-Bouldin[8] and Calinksi-Harabasz[9] scores on the other hand, were used to assess the quality of the resulting clusters, ie how compact and well separated they were.

From the inertia plot (*Figure 5.1.1a*), it can be seen that the inertia dropped very quickly as k was increased to 7 (Euclidean) and 8 (Manhattan), but started to flatten out after those points (with inertia reduction of less than 5 per additional cluster). This suggested a good choice of k=7 for Euclidean and k=8 for Manhattan. Overall, the Euclidean models produced better separated clusters than the Manhattans (as indicated by the lower Davies-Bouldin scores on *Figure 5.1.1b*). Ignoring k=2-3, the Euclideans produced the lowest score at k=9 (1.64) and Manhattan at k=11 (1.89). Similarly, the Calinksi-Harabasz scores suggested the Euclideans formed more

dense and better separated clusters than the Manhattans, though the scores were decreasing as k was increased (*Figure 5.1.1c*). The k-means algorithm was designed for spherical-shaped clusters of similar size and density[10]. Unfortunately, this dataset was neither of those. Therefore, none of the models produced perfectly good clusters, as noted by the low silhouette scores (less than 0.2 for k>=3). Nevertheless, we still think the k-means models plot reasonably well on the silhouette plots (*Figure 5.1.1d*) and the t-SNE projections[11]. Whilst not perfect, the clusters formed were quite dense and reasonably well separated, especially for the Euclidean (*Figure 5.1.1e*). At k=7, the Euclidean also had lower Davies score (1.76 vs 2.14), higher Calinksi (174 vs 156) and silhouette scores (0.16 vs 0.13) than the Manhattan. Therefore, we chose the Euclidean as the optimal model with k=7 as the optimal number of clusters for this dataset.

1.2. ICMLA 2014 Accepted Papers Dataset

Using two k-means algorithms, ninety eight k-means models were trained on this dataset using k=1-49 and two distance measures. The Euclidean distance was again used as the base metric. Here, the Cosine metric[2] was used as a comparison as it tends to work well with text or sparse data (even if two points were far away by the Euclidean distance, they might still be considered close by this distance measure if the angle between the two points was narrow). The gold clusters (showing 24 clusters) were provided for this dataset, so the resulting models were evaluated using external (and one internal) measures. The completeness score[12] was used to measure the degree to which all points of a given class belong to the same cluster, the homogeneity score[13] was used to measure the degree to which all clusters contain only points of a single class, and lastly, the silhouette was used to check the clusters' quality at different k's.

The 'combine' column, which included texts from 'keywords', 'title', 'abstract', was used as the X input because it gave the best overall outcome in our testing. Ignoring k=2, both models showed increasing completeness and homogeneity scores as k was increased (*Figure 5.2.1c and d*). Overall, the scores indicated that the clusters produced by the Euclideans were more complete and homogeneous. This is in line with the labelling on *Table 5.2.1b and c*, where more data points from the same session such as Neural Networks I and II were grouped together by the Euclidean, and more of its clusters contain only data points from a single class. The Euclideans also had better silhouette scores than the Cosines and plot better on the silhouette plots (*Figure 5.2.1a and b*). On the t-SNE projections, both models showed well separated clusters at lower k's, though as k gets larger, the additional clusters were harder to identify (*Figure 5.2.1f and g*). At k=24, the Euclidean produced better separated clusters than the Cosine (*Figure 5.2.1e*).

When the text data was converted into numerics, it resulted in 105 feature columns. Combined with the low number of samples (105 only), it created very sparse data that were hard for the k-means to cluster. Even though the PCA had been used to reduce the number of features to 31, the resulting clusters were still hard to identify. Nevertheless, whilst not perfect, we still think the k-means models still produce reasonably well separated clusters. Of the two models, we chose the Euclidean with k=24 as the optimal model and the optimal number of clusters for k-means. It had the better completeness and homogeneity scores compared to the Cosine or those with lower k's, and it showed better separated clusters on both the silhouette and the t-SNE plots.

2. Hierarchical Clustering

1.1. Travel Reviews Dataset

Using sklearn Agglomerative Clustering[14], over 260 models were trained on this dataset using k=1-29, three similarity measures (Euclidean, Manhattan and Cosine) and three distance

metrics (single-link, average-link and complete-link). The Cosine similarity looks at the angle between two vectors, whereas the Euclidean (the straight line) and Manhattan (the distance in blocks) look at the distance between two points. The single-link uses the smallest distance, the average-link uses the average distance, and the complete-link uses the largest distance, when measuring distance between a point in a cluster and a point in other clusters[10].

Again, without the gold clusters, the resulting models were evaluated using three internal measures: the silhouette score, the Davies-Bouldin score and the Calinksi-Harabasz score.

The complete-link models produced more dense and better separated clusters than the average or single-links, as indicated by the higher Calinski scores across different k's (*Figure 5.1.2b-d*). When paired with complete-link, all three similarity measures seemed to work well. Overall, the Calinski scores were decreasing as k was increasing, but the Manhattan models seemed to do better at lower k's, whereas the Cosines were better at higher k's (with the Euclideans somewhere in between). The Davies-Bouldin scores were quite good for all models (mostly less than two), with the single and average-link produced slightly better scores than the complete-link (*Figure 5.1.2e-g*). When paired with the single-link, both Manhattan and Euclidean models produced better scores (0.54 at k=5) than the Cosine (0.65 for k=5). Similar to k-means however, all models produced low silhouette scores (less than 0.3 for k>=3). On the t-SNE projections, both Manhattan and Euclidean complete-link models showed well separated clusters at k=3 (*Figure 5.1.2h*), but did not produce good clusters when k was increased (the additional clusters were small and hard to identify). At k=7, the Cosine complete-link model showed better separated clusters than the Euclidean or Manhattan. The single and complete-link though, produced neither well balanced nor well separated clusters (*Figure5.1.2i*).

The complete-link usually favors globular shapes and is less susceptible to noise and outliers[16]. Therefore, when combined with the cosine similarity measure, it seemed to produce reasonably good clusters for this dataset. Whilst at k=3 both the Euclidean and Manhattan complete-link models produced better separated, higher Calinski score and lower Davies score, they only identified three clusters, and we feel such low number of clusters would not be too meaningful for the business (though consultation with domain experts would clarify this). Therefore, for this algorithm, we propose the Cosine complete-link with k=7 as the optimal model. It had the highest Calinski score at k=7 (132), relatively low Davies score (1.94) and well separated clusters on the silhouette coefficient plot (*Figure 5.1.2a)* and the t-SNE projection.

1.2. ICMLA 2014 Accepted Papers Dataset

Using scipy Hierarchy Clustering[15], over 260 models were trained on Dataset 2 using k=1-49, three similarity measures (Euclidean, Cityblock and Cosine) and three distance metrics (single-link, average-link and complete-link). Having the gold clusters provided, the resulting models were evaluated using external measures: the Completeness and the Homogeneity scores.

Overall, the single-link models produced higher completeness scores than the average and complete-link models at lower k's, but the scores were increasing and started so converge as k was increased, and all models produced similar scores a higher k's (*Table 5.2.2a*). At k=24, the Cosine-average, Cityblock-average and Cityblock-complete models produced the highest score at 0.61, followed closely by Cityblock-single (0.60). On the contrary, the homogeneity scores for the single-link models were mostly lower than the average and complete-links. However, the scores were increasing as k was increased (*Table 5.2.2b*), and at k=24, the Cityblock-complete had the highest score (0.60), followed by Cosine-average (0.59) and Euclidean-complete (0.58).

The single-link metric[16] is good at handling non-elliptical shapes, but is sensitive to noise and outliers. The complete-link[16] on the other hand, is less susceptible to noise and outliers, but can break a large clusters and favors globular shapes (and the average-link[16] is the intermediate approach between the two). This dataset contains sparse data with a combination of globular (in the middle) and non-globular shapes (at the top), and some outliers (*Figure 5.2.2h*). Therefore, the average-link seemed to work better for this dataset as it seemed to handle the different cluster shapes and the outliers reasonably well. At lower k's, the average and the complete-link showed more balanced clusters than the single-link (*Figure 5.2.2i*). At k=24, all models produced reasonably well separated, but different clusters (*Figure 5.2.2g*). On the dendogram[17], the average and complete-link also showed better groupings than the single-link as more points from the same session such as Feature Extraction and Selection were grouped together (*Figure 5.2.2j-l*). Of the three similarity measures, the Cosine seemed to plot better than the Euclidean or Cityblock (*Figure 5.2.2m-n*). Instead of measuring the actual distance, the Cosine measured the angle between points, and for our sparse data, it seemed to work quite well. Accordingly, for this algorithm, we chose the Cosine-average with k=24 as the optimal model. It had better completeness (0.61) and homogeneity (0.59) scores compared to most other models, and at k=24, it produced reasonably well separated clusters on the t-SNE plot and the dendogram.

3. DBSCAN Clustering

1.1. Travel Reviews Dataset

Using the sklearn DBSCAN algorithm[18], over 100 models were trained on this dataset using three distance measures (Euclidean, Cityblock and Cosine) and different epsilons (0.01-5) and min samples (2-10) values. The epsilon sets the maximum radius of the neighborhood, whereas the minimum samples set the minimum number of points in the neighborhood for a point to be considered a core point[19]. Without the gold clusters, the resulting models were evaluated using internal measures only: the Davies-Bouldin score and the Calinksi-Harabasz score.

The highest Calinksi-Harabasz score (62) was produced by the Euclidean model (with eps=0.22 and min samples=7). It was able to identify five clusters. The lowest Davies-Bouldin score (0.40), produced by the Euclidean model (eps=0.63 and min samples=4), identified only two clusters. Other combinations (the Cityblock, eps=0.40, min samples=8 and the Cosine, eps=0.01, min samples=8 identified eight; the Cosine, eps=0.40, min samples=8 had six; the Euclidean, eps=0.20, min samples=5 had 13; the Euclidean, eps=0.20, min samples=8 had ten; and the Cosine, eps=0.01, min samples=6 had 11 clusters) were able to identify more clusters, but with lower Calinski or higher Davies scores (*Figure 5.1.3a-f*). On the t-SNE projections, none of the models produced good clustering, with one cluster scattered around the other clusters (*Figure 5.1.3g*). Even the optimal model had this problem (*Figure 5.1.3h*).

The DBSCAN algorithm works well with clusters of different shapes and sizes, and is resistant to noise[19]. Unfortunately, it struggles with data of varying densities[19], hence it did not work well on this dataset. Of the different combinations, we chose the Euclidean (eps=0.22 and min samples=7) as the optimal model. It had low Davies score (3.63), the highest Calinski score (62), and it showed five reasonably well, though not perfectly separated clusters on the plot.

1.2. ICMLA 2014 Accepted Papers Dataset

Over 100 models were trained on this dataset using the same three distance measures and different epsilon (0.01-5) and min samples (2-15) values. Having the gold clusters, the resulting models were evaluated using two external measures: completeness and homogeneity scores.

The highest homogeneity score (0.48) was produced by the Cityblock model (with eps=4.63 and min samples=2), and it was able to identify 20 clusters. The highest completeness score (0.75), produced by the Cosine model (eps=0.33 and min samples=5), but only found two clusters. Other combinations (the Cityblock, eps=4.60, min samples=2, identified 20 clusters; the Euclidean, eps=1, min samples=2, 20 clusters; and the Cosine, eps=0.42, min samples=2, 15 clusters) identified more clusters, but with lower completeness or homogeneity scores (*Figure 5.2.3a-f*). Overall, both scores were decreasing and less clusters were identified as the minimum sample was increased (because more points were required to form a cluster). However, there was more fluctuations in both scores when the epsilon was varied. Again, due to uniformly sparse data, DBSCAN did not work very well on this dataset. The clusters formed were not clearly separated (*Figure 5.2.3g*) and the labelling seemed to group many different classes into one (*Table 5.2.3a*). Nonetheless, we chose the Cityblock (with eps=4.63 and min samples=2) as the optimal model for this algorithm. It had reasonable homogeneity (0.48) and completeness (0.58) scores, plotted reasonably well on the projection and produced 20 clusters that seemed to match better with the gold clusters compared to the other models.

4. The Best Model

4.1. Travel Reviews Dataset

	K-means	Agglomerative	DBSCAN
Parameters	Euclidean	Cosine, complete-link	Euclidean, eps=0.22, min sample=7
No. of clusters	7	7	5
Calinski-Harabasz	174	132	62
Davies-Bouldin	1.75	1.94	3.63

Of the three models, k-means produced the highest Calinski score and the lowest Davies score. It produced seven reasonably dense and well separated clusters on the t-SNE projection, and the elbow method confirmed that for this model, k=7 is the optimal number of clusters for this dataset. Accordingly, we chose k-means Euclidean with k=7 as the best model for this dataset.

4.2. ICMLA 2014 Accepted Papers Dataset

	k-means	Hierarchical	DBSCAN
Parameters	Euclidean	Cosine, average-link	Cityblock, eps=4.63, min sample=2
No. of clusters	24	24	20
Completeness	0.61	0.61	0.58
Homogeneity	0.60	0.59	0.48

Overall, all models produced similar completeness scores. However, the DBSCAN model had notably lower homogeneity scores than the other two models. This is not surprising because DBSCAN, which formed clusters based on the density of the data points, struggled with the uniformly sparse dataset. The k-means and the hierarchical on the other hand, worked better and had higher completeness and homogeneity scores. Whilst both models had similarly high completeness and homogeneity scores, plot well on the t-SNE projections and showed reasonably well grouping on the labelling table (*Table 5.2.1b* and *Table 5.2.2c*), we feel that the dendogram made the hierarchical model more interpretable than the k-means, and it provides more flexibility for the business with regards to choosing the number of clusters (*Figure 5.2.2j*). Therefore, we chose the hierarchical Cosine-average-link as the optimal model for this dataset.

5. Appendices

5.0. Data Setup (Load and Preprocessing)

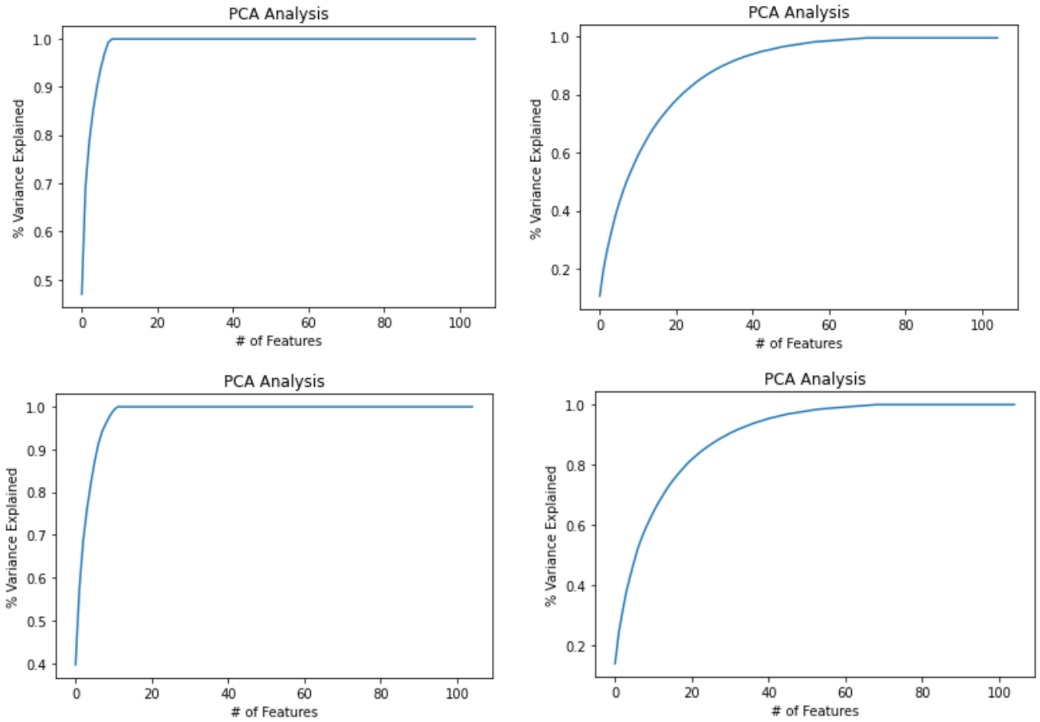


Figure 0.1. PCA analysis of the title (top-left), abstract (top-right), keywords (bottom-left) and combine (bottom-right) columns

5.1. Travel Reviews Dataset

5.1.1. K-Means Clustering

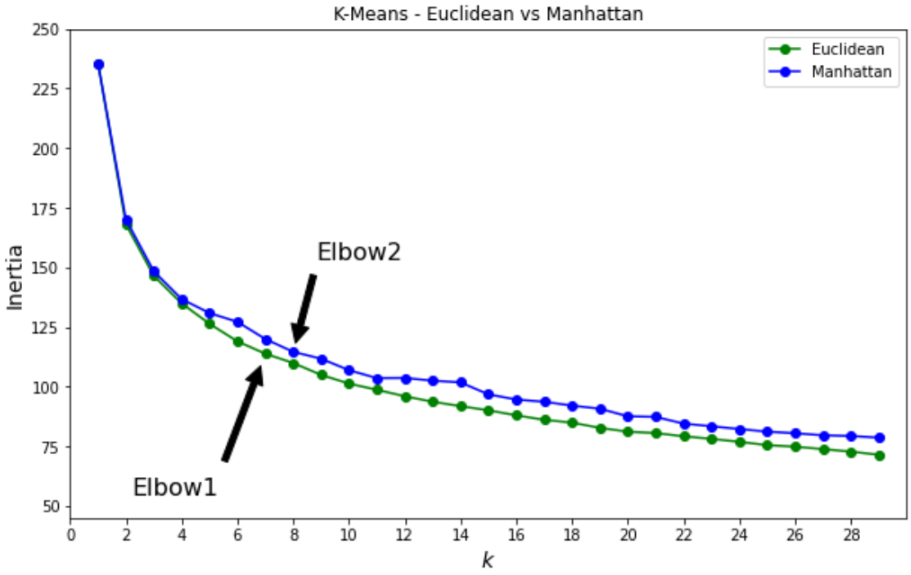


Figure 5.1.1a. Inertia plot: K-Means Euclidean (green) vs Manhattan (blue)

Euclidean

	inertia	difference
0	235.629278	NaN
1	168.075115	67.554163
2	146.428224	21.646890
3	134.887950	11.540275
4	126.321167	8.566783
5	118.973156	7.348011
6	113.794057	5.179099
7	109.906226	3.887831
8	104.918161	4.988065
9	101.262739	3.655422
10	98.670106	2.592633
11	95.967277	2.702829
12	93.653296	2.313981
13	91.815698	1.837598

Manhattan

	inertia	difference
0	235.629278	NaN
1	170.059027	65.570251
2	148.336010	21.723017
3	136.686933	11.649077
4	130.828513	5.858420
5	127.150791	3.677722
6	119.957103	7.193688
7	114.576860	5.380243
8	111.696253	2.880607
9	106.892404	4.803849
10	103.548975	3.343429
11	103.614208	-0.065233
12	102.493515	1.120693
13	101.785898	0.707618

Table 5.1.1a. Inertia difference: K-Means Euclidean (left) vs Manhattan (right)

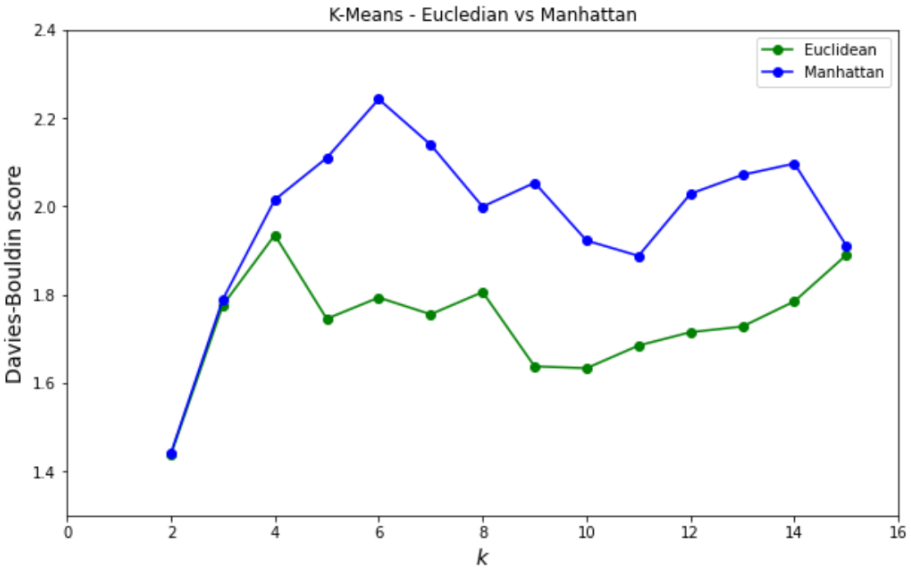


Figure 5.1.1b. Davies-Bouldin score: K-Means Euclidean (green) vs Manhattan (blue)

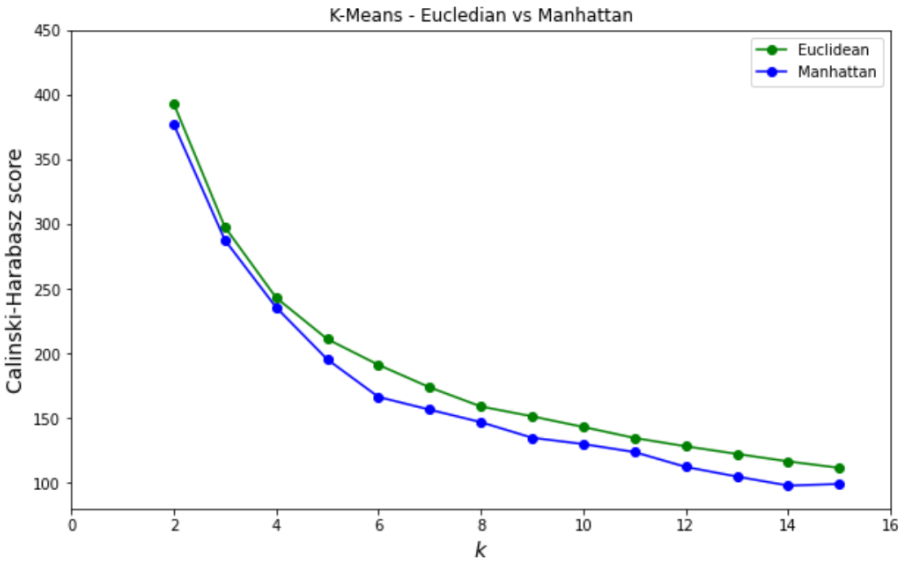


Figure 5.1.1c. Calinski-Harabasz score: K-Means Euclidean (green) vs Manhattan (blue)

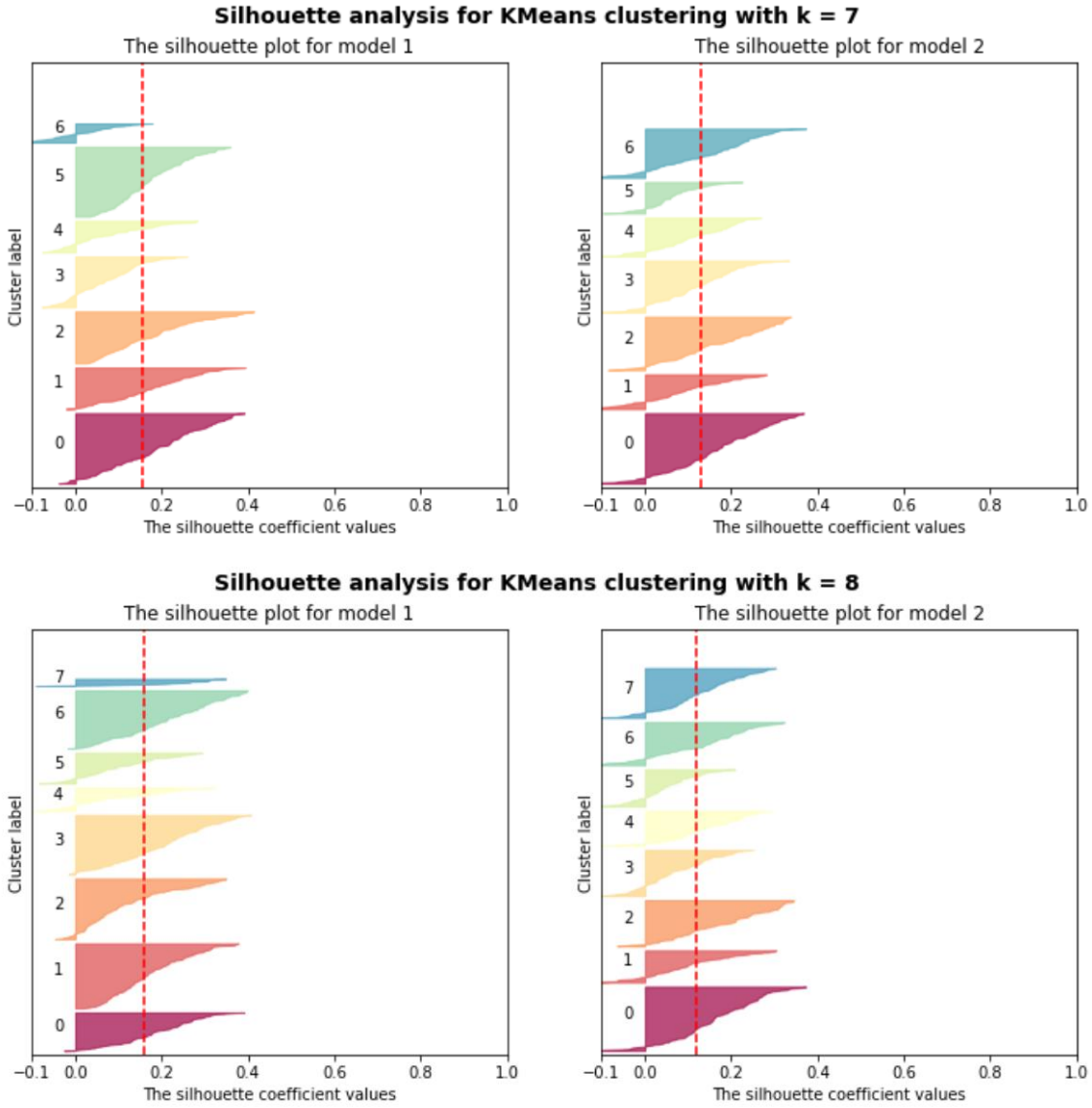


Figure 5.1.1d. Silhouette Analysis: K-Means Euclidean (left) vs Manhattan (right)

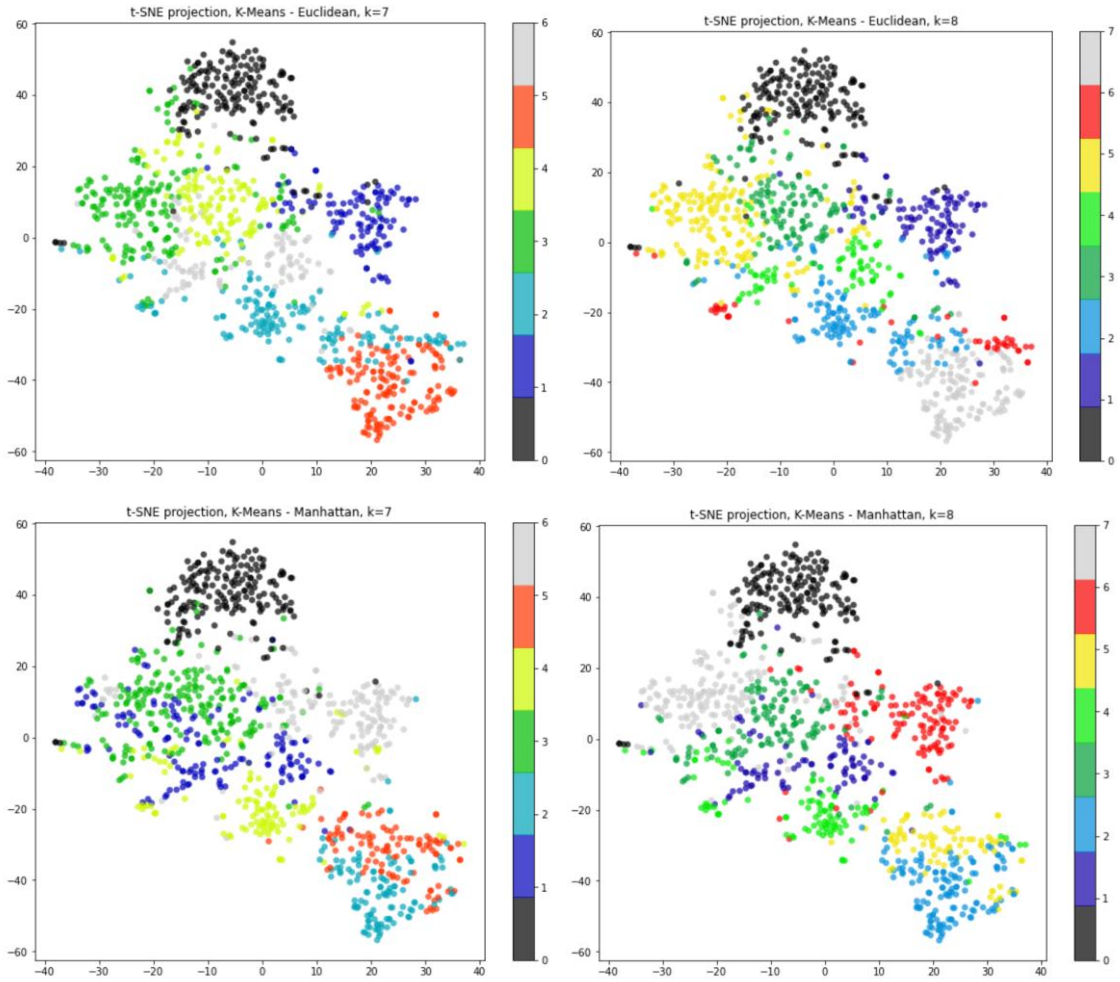


Figure 5.1.1e. t-SNE projection: K-Means Euclidean (top) vs Manhattan (bottom)

5.1.2. Agglomerative Clustering

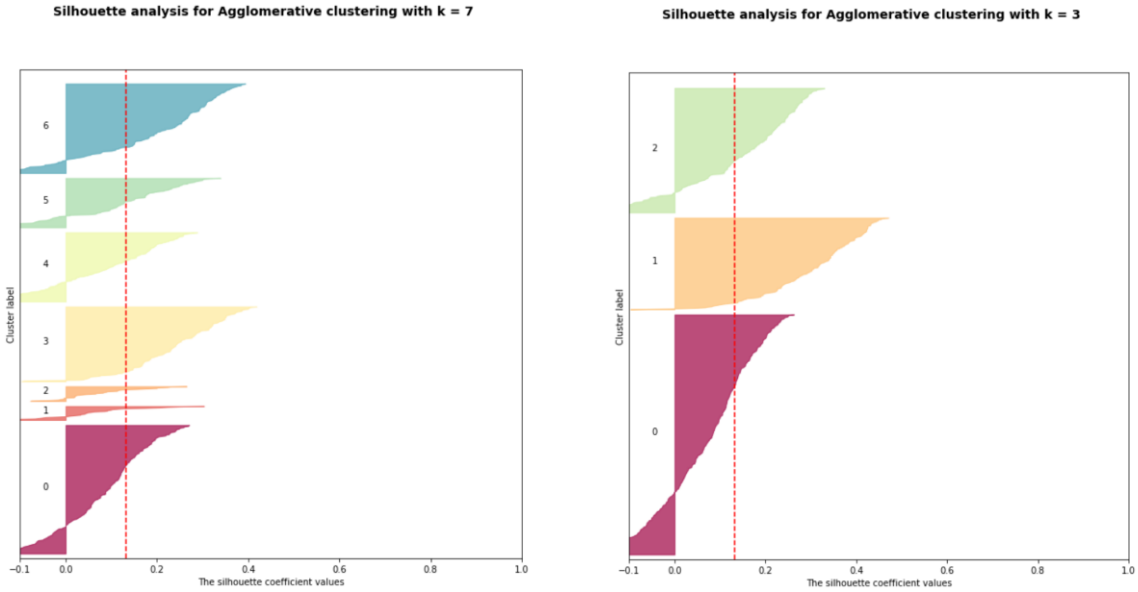


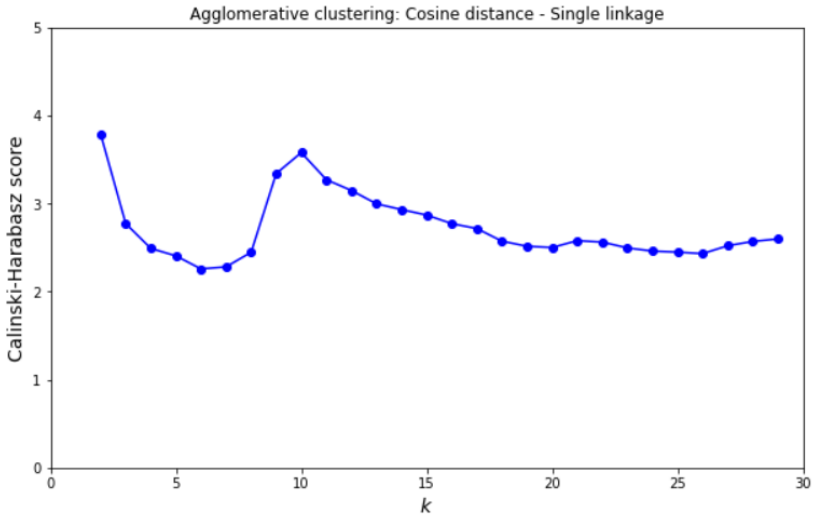
Figure 5.1.2a. Cosine complete-link, k=7 (left) and Manhattan complete-link, k=3 (right)

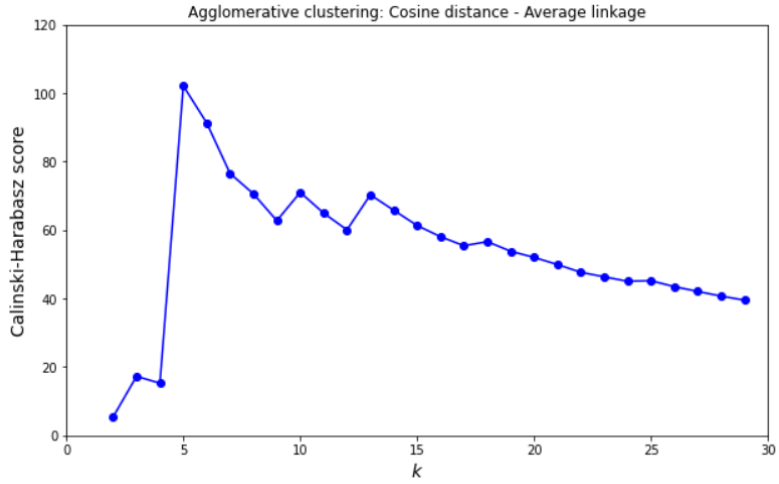
	k	cosine-single	cosine-average	cosine-complete	euclidean-single	euclidean-average	euclidean-complete	manhattan-single	manhattan-average	manhattan-complete
0	2	3.787220	5.416734	181.780777	9.342267	9.342267	267.709413	3.809731	9.342267	309.543969
1	3	2.771441	17.285063	106.188634	6.609191	6.609191	195.405860	3.600932	6.591341	237.505569
2	4	2.489744	15.251828	84.043466	5.693640	5.693640	142.258031	5.556358	5.693640	168.381686
3	5	2.405053	102.109272	117.559678	5.138337	6.795051	123.811982	5.138337	5.346979	151.712328
4	6	2.257358	91.304075	118.517489	4.621251	6.312727	126.434424	4.584204	5.309232	124.403883
5	7	2.280077	76.529386	131.648893	4.247459	6.410286	108.257734	4.186017	59.903870	112.785392
6	8	2.445759	70.575933	123.321970	3.992196	9.193597	110.947673	3.939241	58.199306	105.846029
7	9	3.339324	62.736421	114.941378	3.769467	54.878018	100.053279	3.833316	53.746329	98.712930
8	10	3.577378	70.994958	104.182362	3.589807	49.943263	90.626563	3.766919	48.284952	95.745509
9	11	3.269711	64.961400	95.910786	3.452469	47.858980	88.255684	3.589841	43.976831	89.153463
10	12	3.147112	60.045221	92.061090	3.303838	47.237609	82.399388	3.459862	40.617888	85.635203
11	13	2.995780	70.288671	86.599496	3.299422	43.524787	77.911347	3.350556	37.770654	82.506105
12	14	2.930421	65.777036	80.613782	3.199715	40.738998	79.376761	3.265861	37.932504	78.212664
13	15	2.868038	61.328728	76.471149	3.125539	41.596325	78.680588	3.149814	36.413904	74.148561

Table 5.1.2a. Calinski-Harabasz scores for different combinations of Agglomerative models

	k	cosine-single	cosine-average	cosine-complete	euclidean-single	euclidean-average	euclidean-complete	manhattan-single	manhattan-average	manhattan-complete
0	2	0.489301	1.298459	1.654703	0.598190	0.598190	1.729692	0.487803	0.598190	1.278057
1	3	0.642505	1.380152	1.864555	0.560328	0.560328	2.044680	0.507675	0.561395	2.033082
2	4	0.653362	1.345352	1.832398	0.541941	0.541941	2.057209	0.548818	0.541941	1.979288
3	5	0.652696	1.382374	1.841618	0.536151	0.769110	1.878246	0.536151	0.717163	1.842000
4	6	0.670213	1.433394	2.069629	0.547533	0.807978	1.829682	0.555094	0.751653	1.943182
5	7	0.662316	1.264549	1.938279	0.561141	0.897734	1.704912	0.571738	0.827202	1.843471
6	8	0.643367	1.450093	1.813936	0.567250	0.940719	1.740446	0.576565	0.877864	2.161831
7	9	0.636534	1.392667	1.879945	0.578468	0.909047	1.687928	0.572871	0.925141	1.936362
8	10	0.718422	1.406619	1.883133	0.586728	0.990788	1.575225	0.568647	0.951838	2.001326
9	11	0.699059	1.406017	1.835464	0.592014	1.068394	1.666171	0.581936	0.922115	1.893624
10	12	0.698200	1.386730	1.815625	0.606792	1.122929	1.601295	0.587823	0.920500	1.933102
11	13	0.708108	1.438003	1.855772	0.600908	1.079189	1.598570	0.592855	0.966014	1.908000
12	14	0.704017	1.341894	1.751963	0.606410	1.083124	1.658490	0.596015	1.014607	2.034362
13	15	0.701388	1.279295	1.715870	0.609351	1.132318	1.646710	0.611054	1.023663	1.978714

Table 5.1.2b. Davies-Bouldin scores for different combinations of Agglomerative models





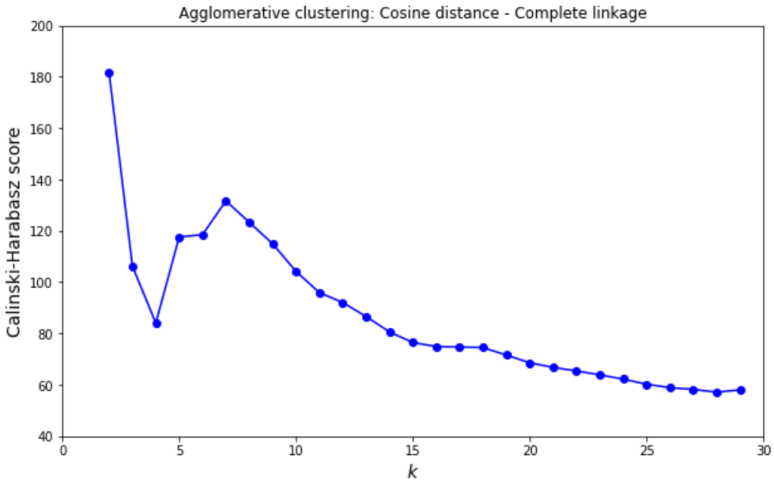


Figure 5.1.2b. Calinski Harabasz scores: Cosine models

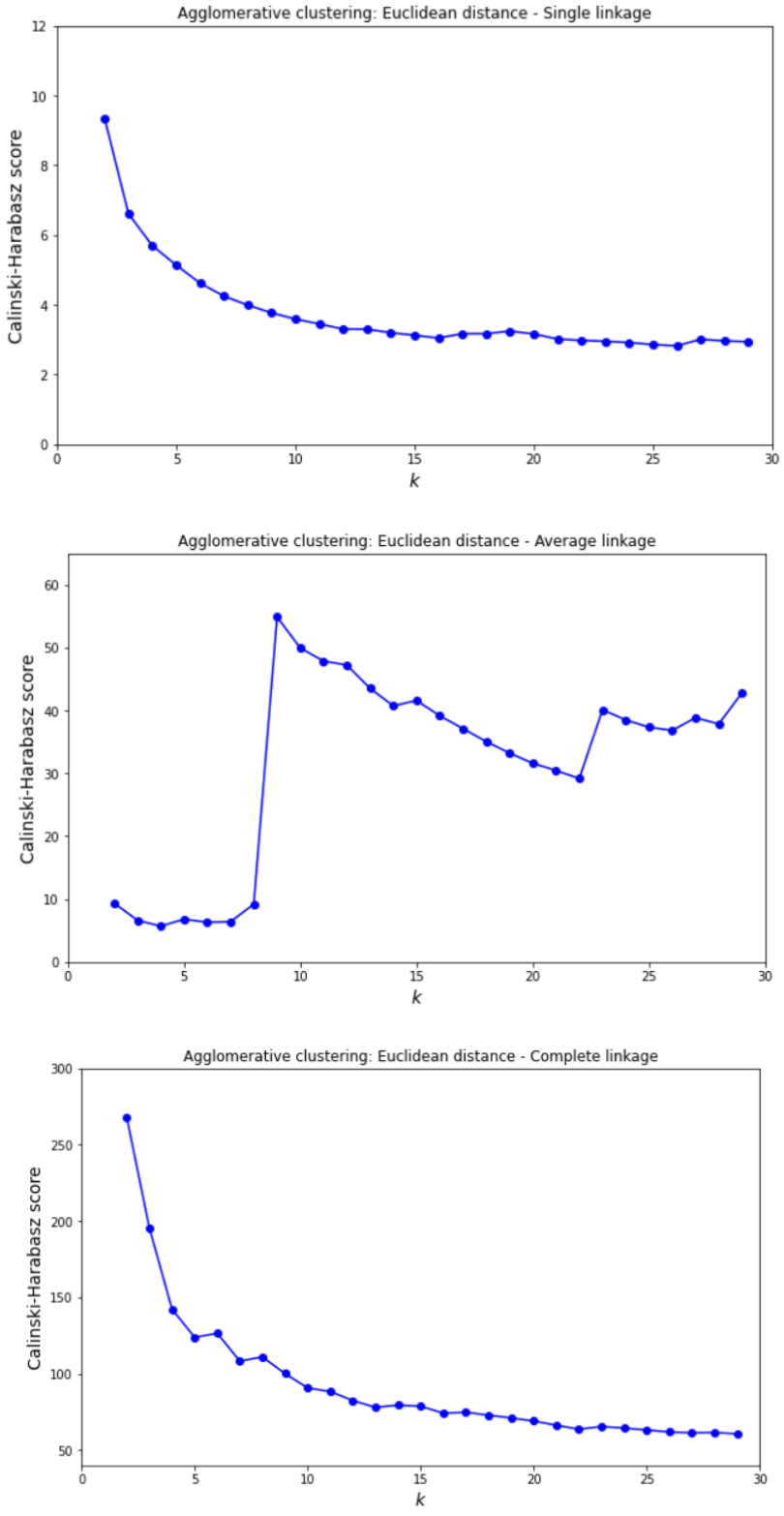


Figure 5.1.2c. Calinski Harabasz scores: Euclidean models

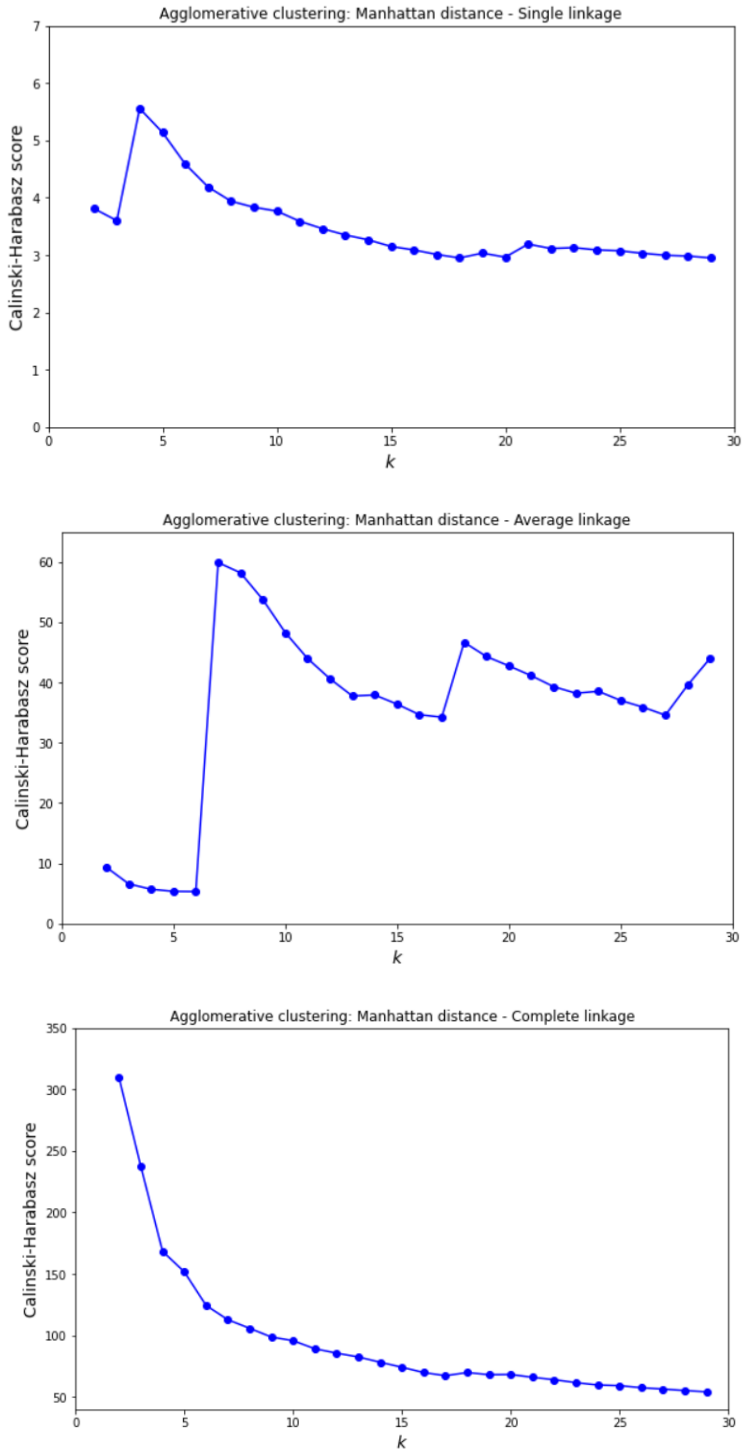


Figure 5.1.2d

Figure 5.1.2d. Calinski Harabasz scores: Manhattan models

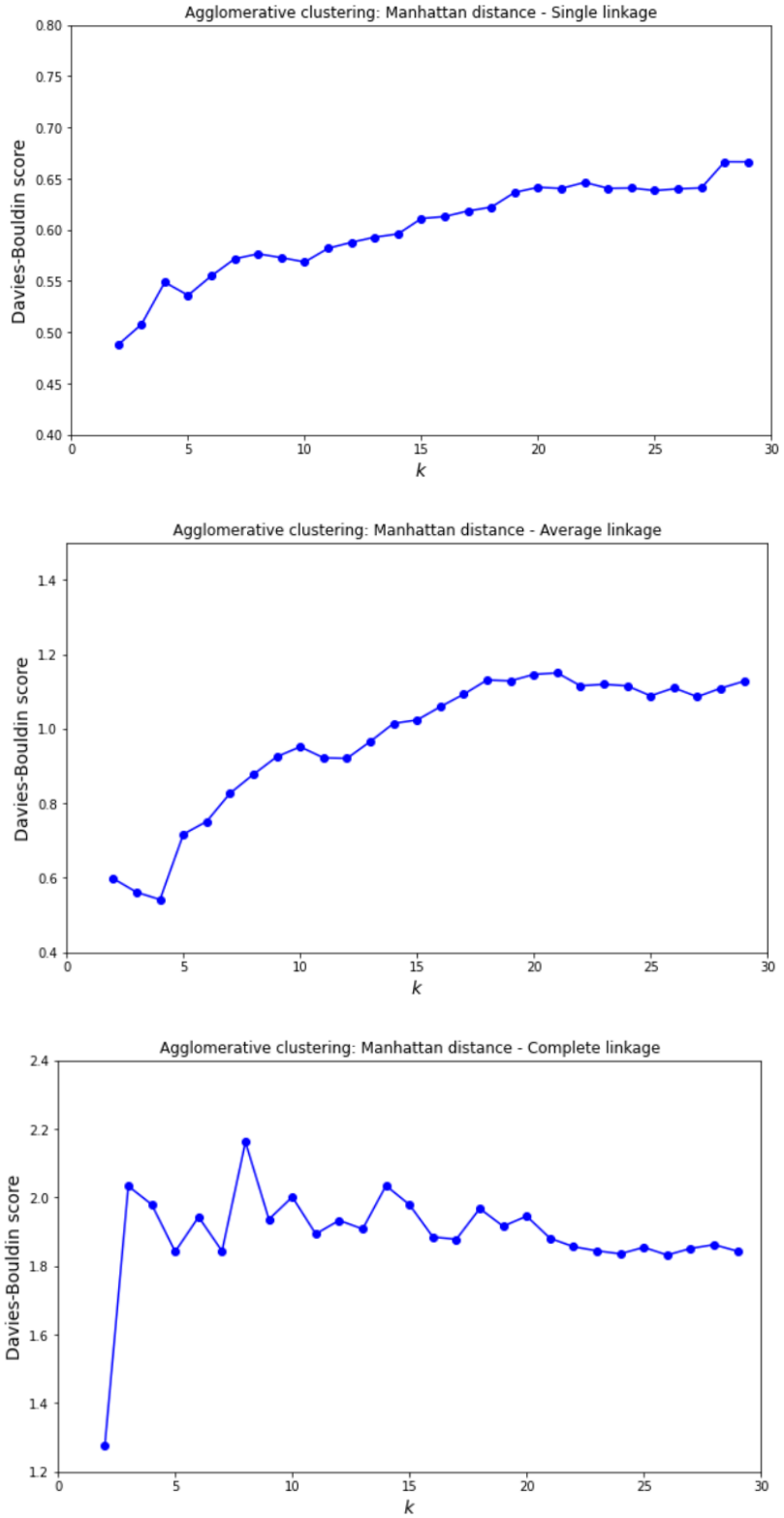


Figure 5.1.2e. Davies-Bouldin scores: Manhattan models

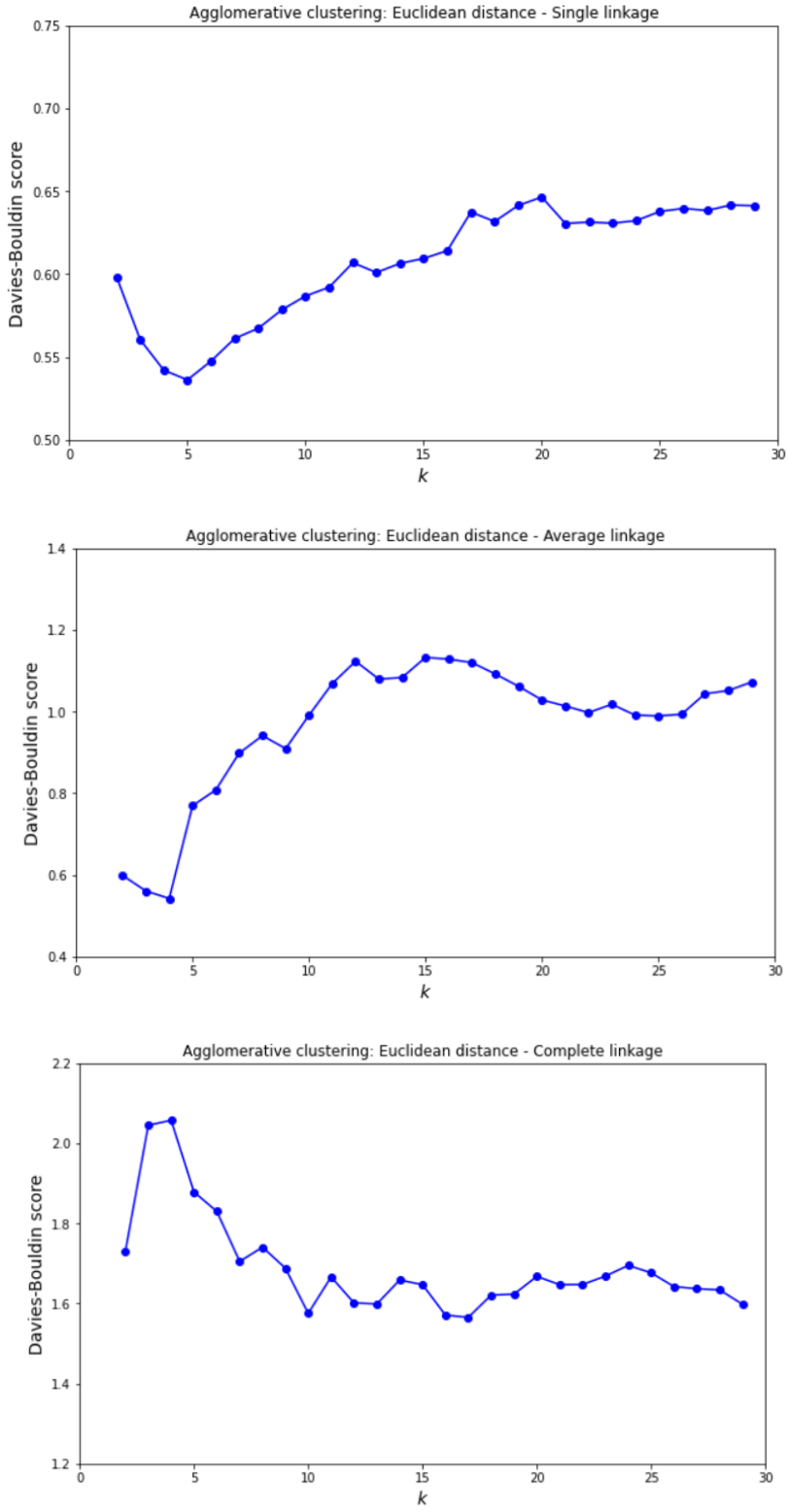


Figure 5.1.2f. Davies-Bouldin scores: Euclidean models

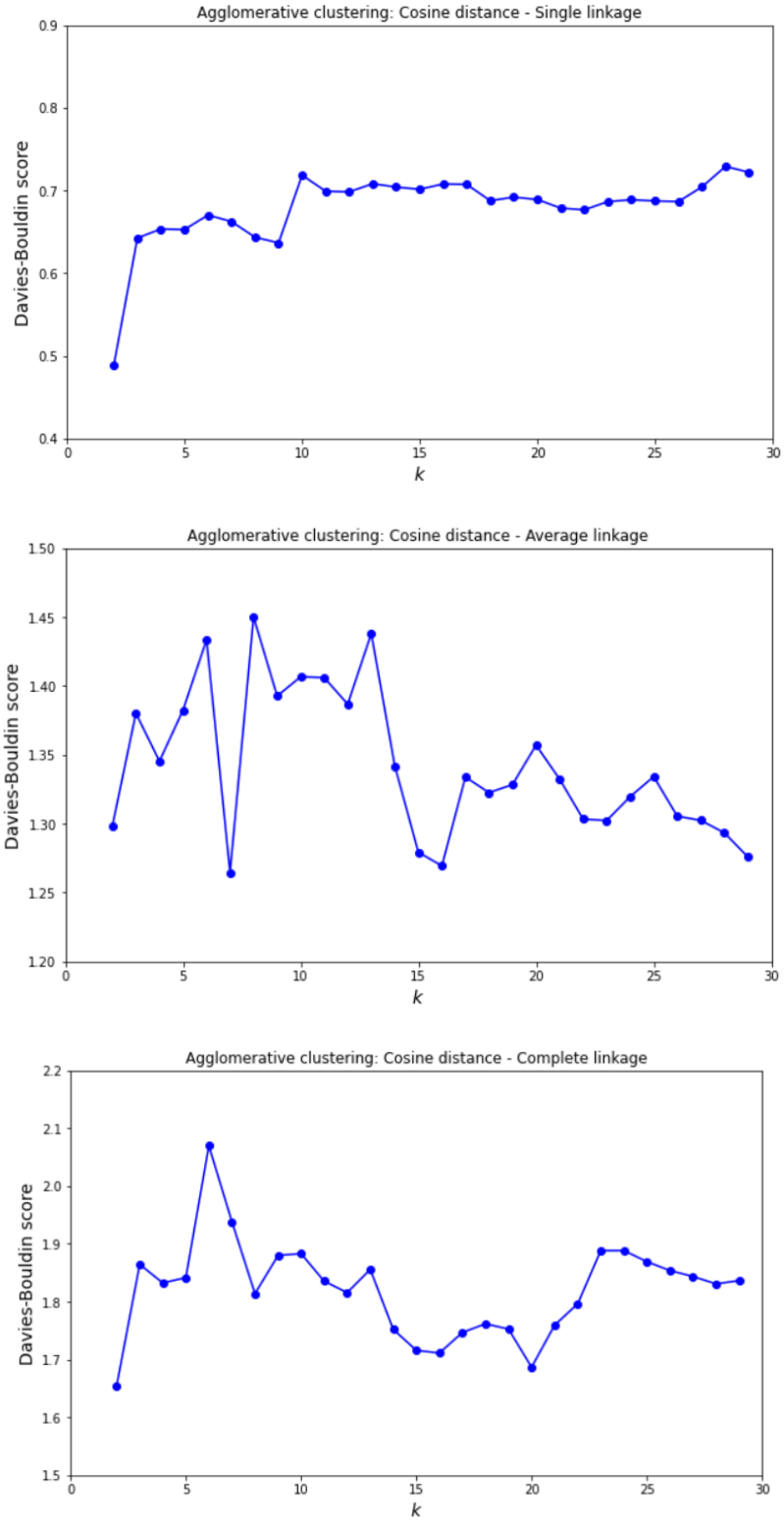


Figure 5.1.2g. Davies-Bouldin scores: Cosine models

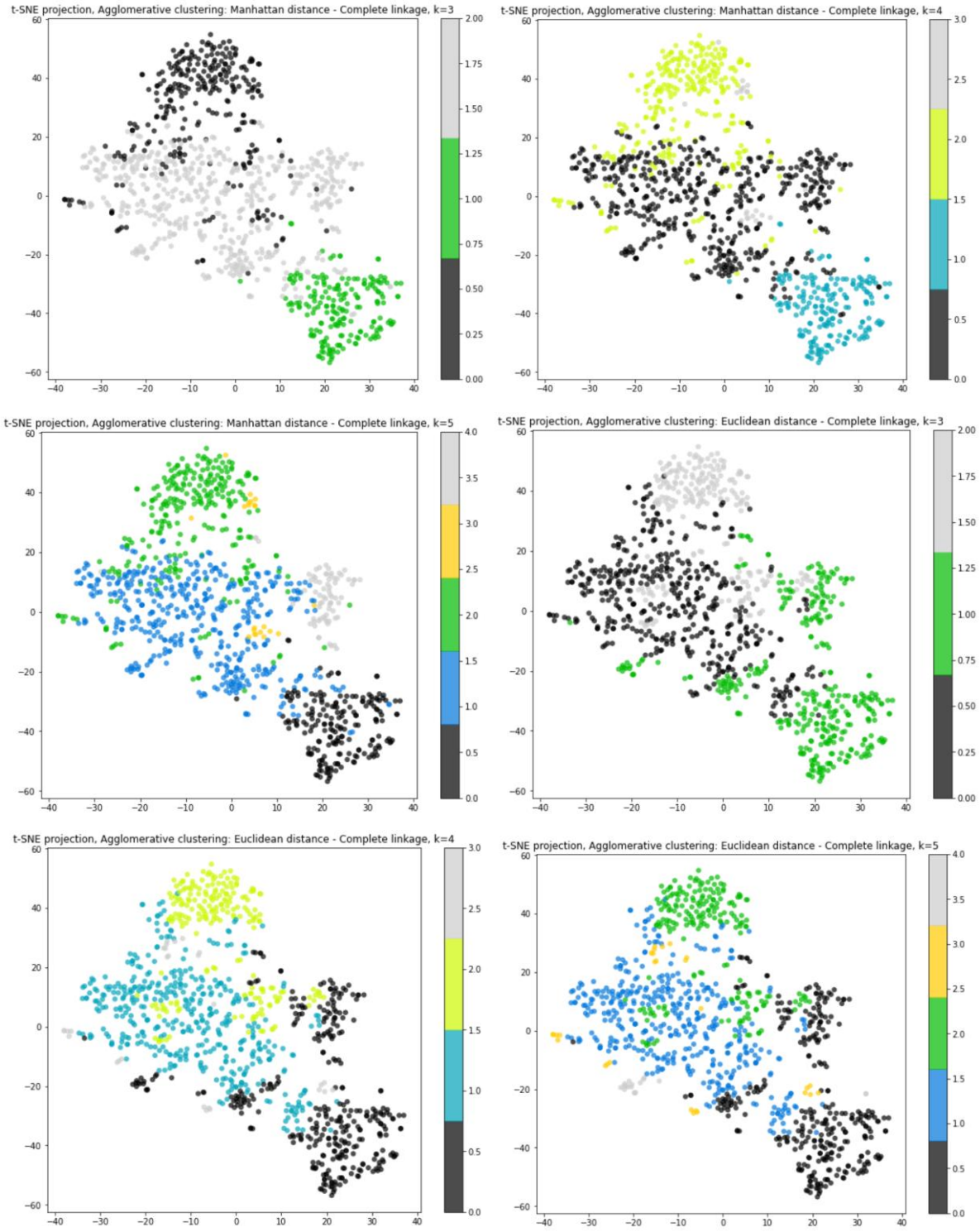


Figure 5.1.2h. t-SNE projections: Manhattan-complete, k=3-5 and Euclidean-complete, k=3-5

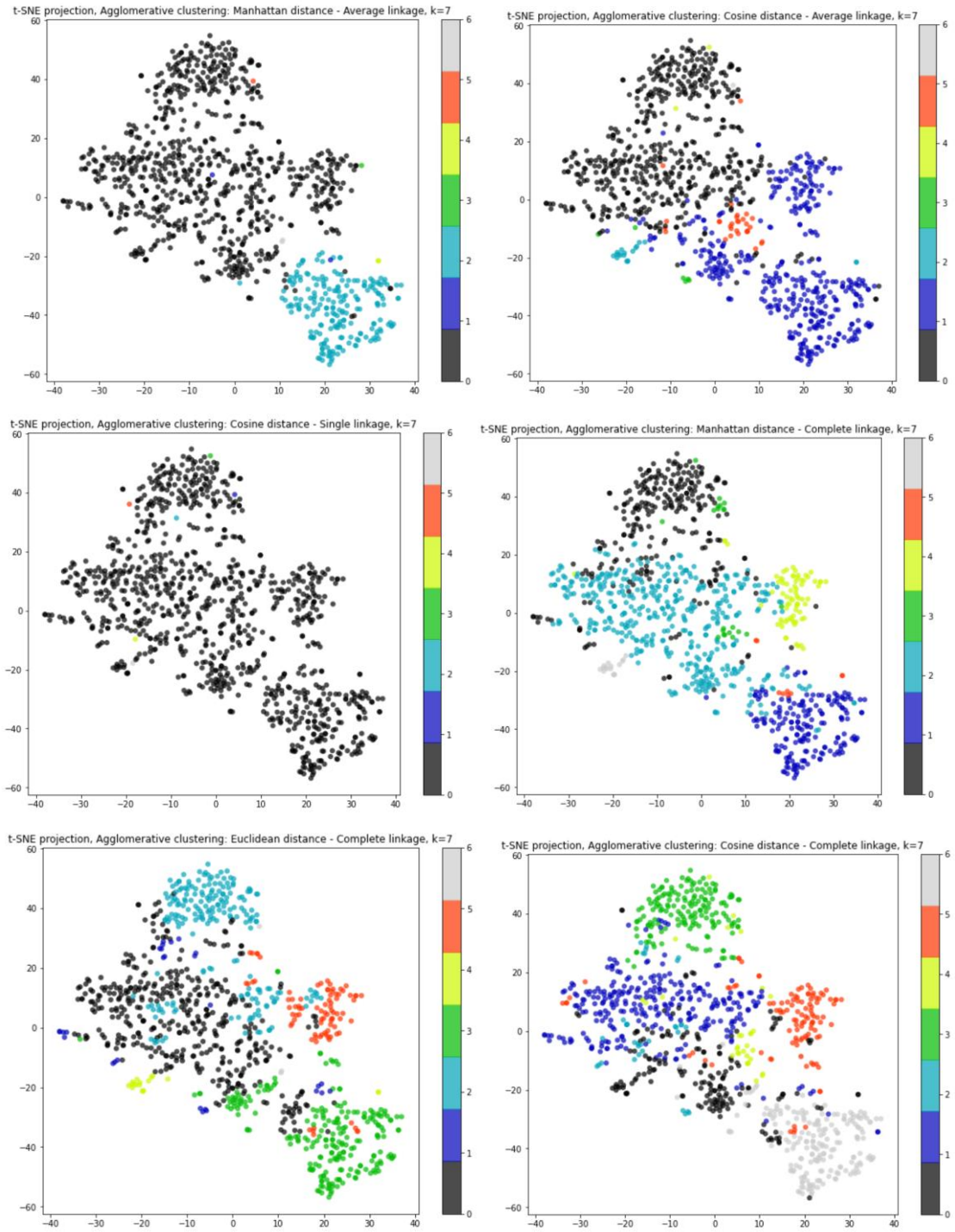


Figure 5.1.2i. t-SNE projections (k=7): Manhattan and Cosine-average (top) vs Cosine-single and Manhattan-complete (middle), Euclidean and Cosine-complete (bottom)

5.1.3. DBSCAN

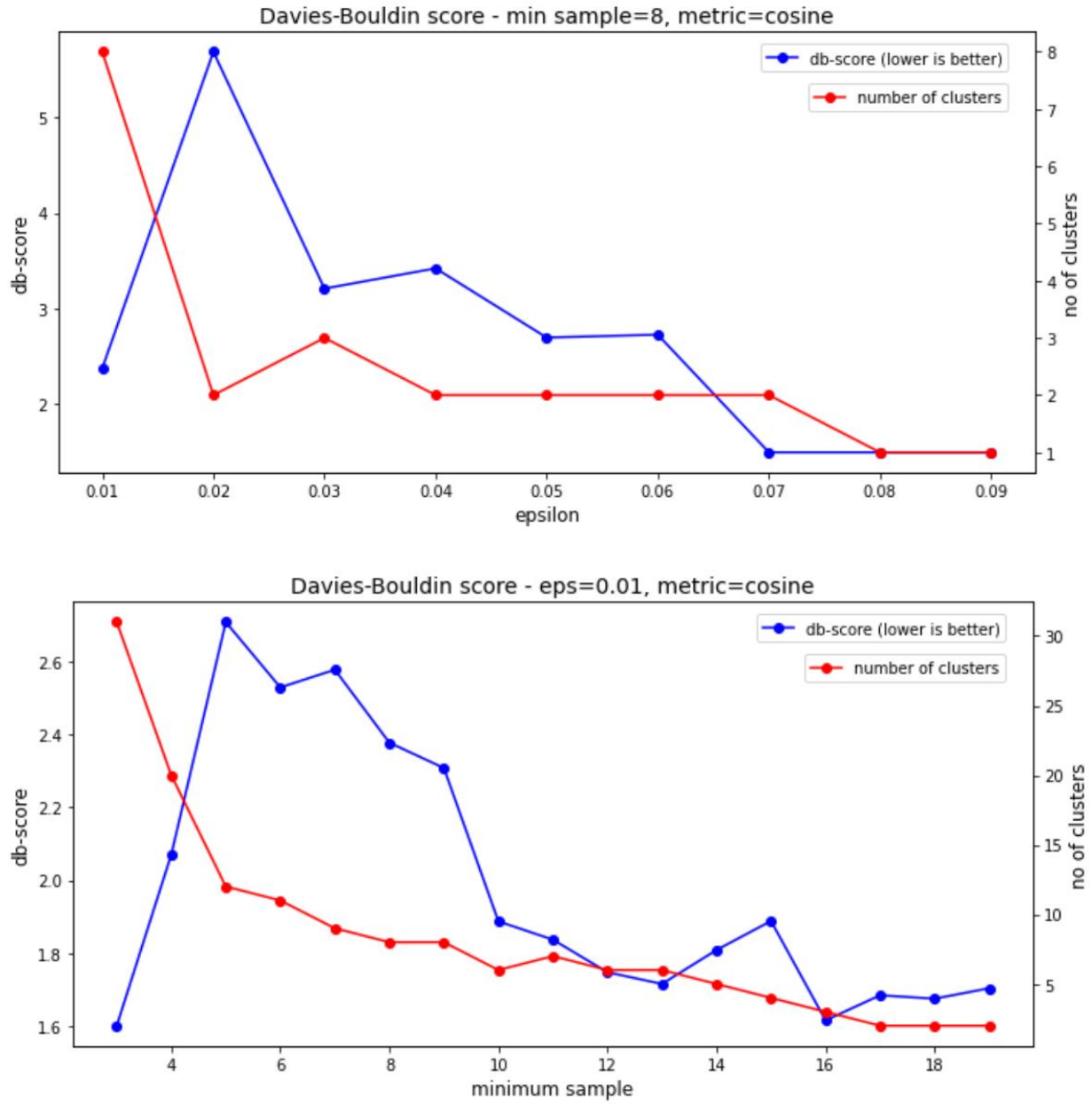


Figure 5.1.3a. Davies-Bouldin scores: Cosine distance

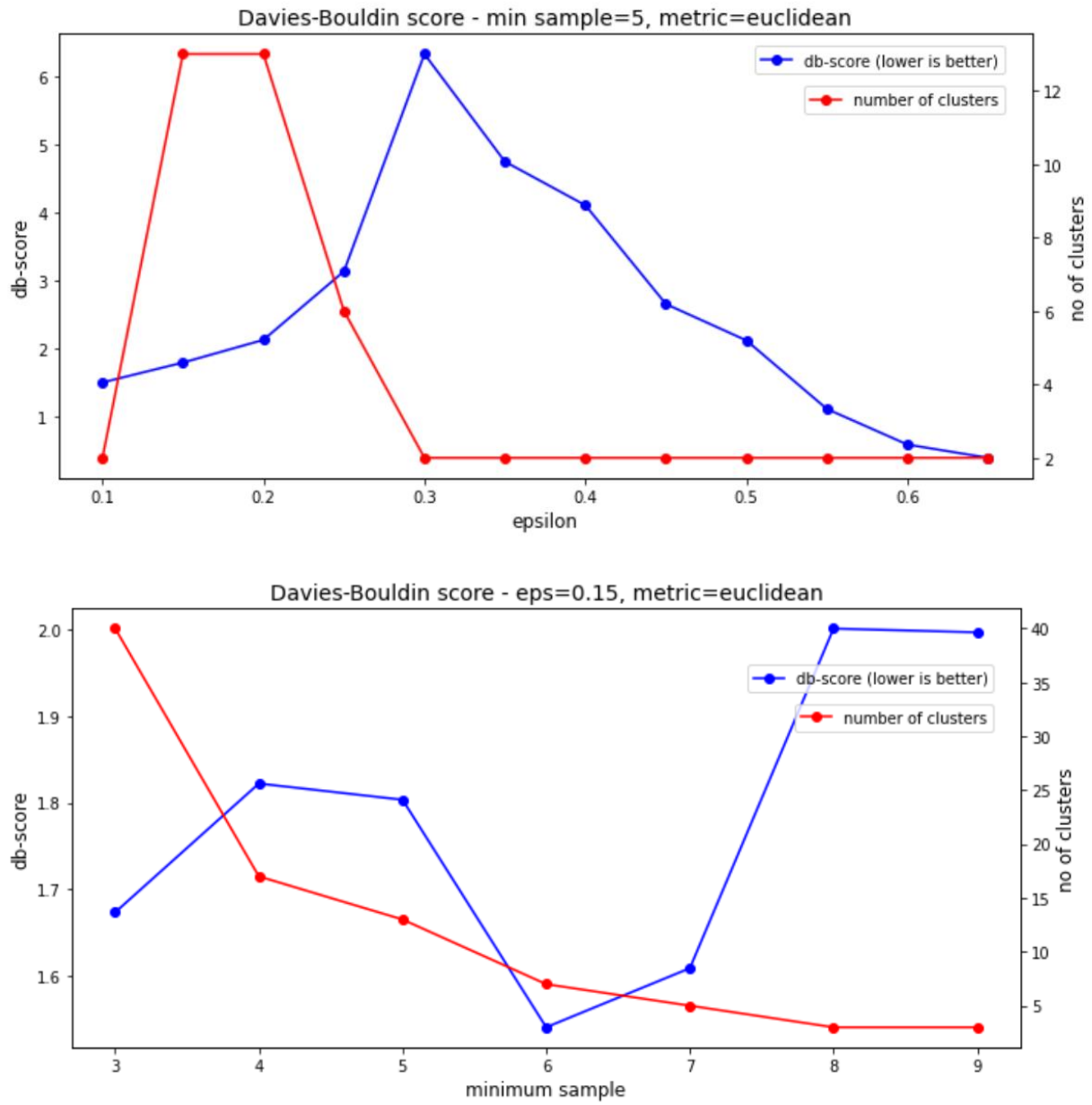


Figure 5.1.3b. Davies-Bouldin scores: Euclidean distance

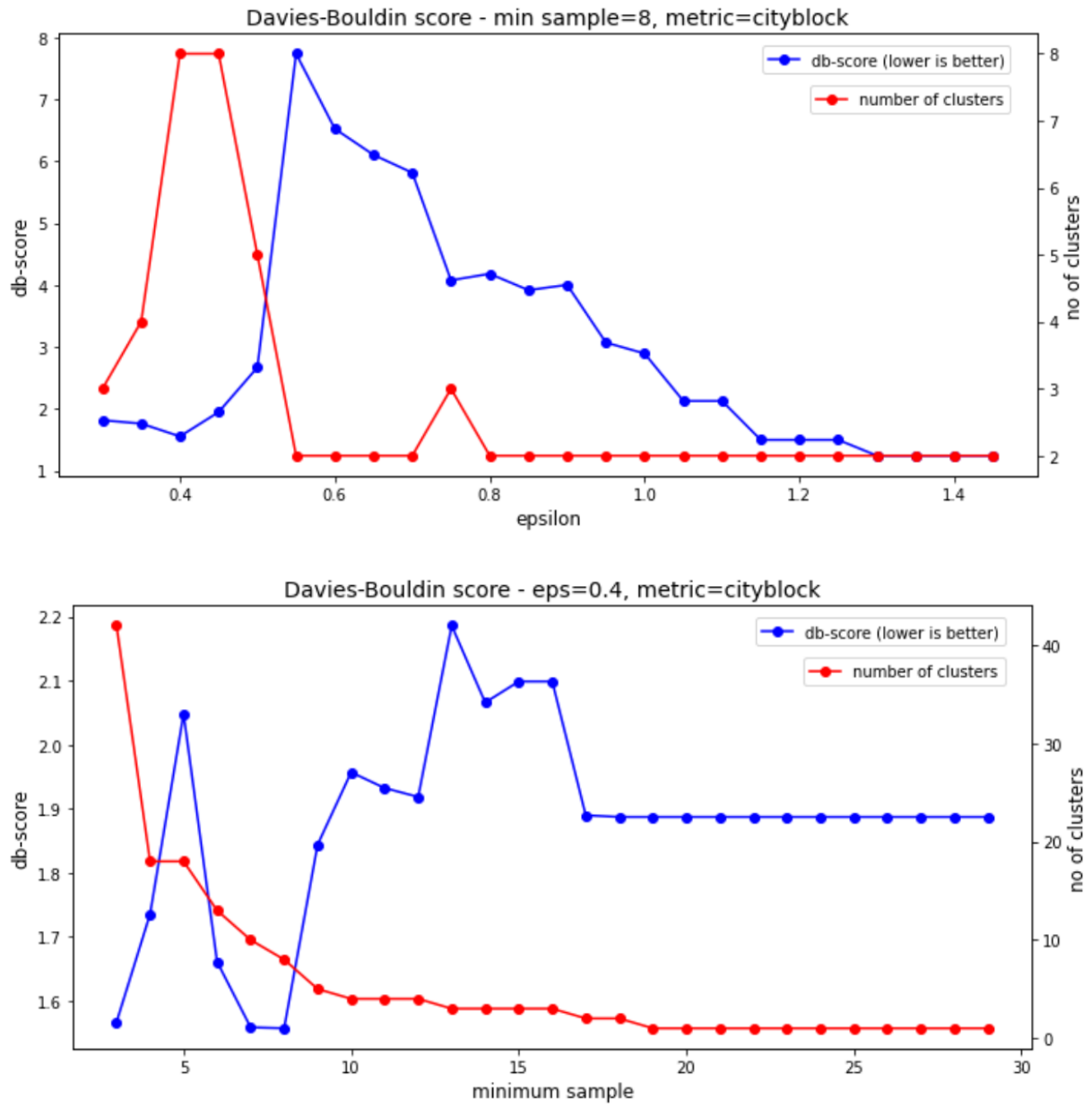


Figure 5.1.3c. Davies-Bouldin scores: Cityblock distance

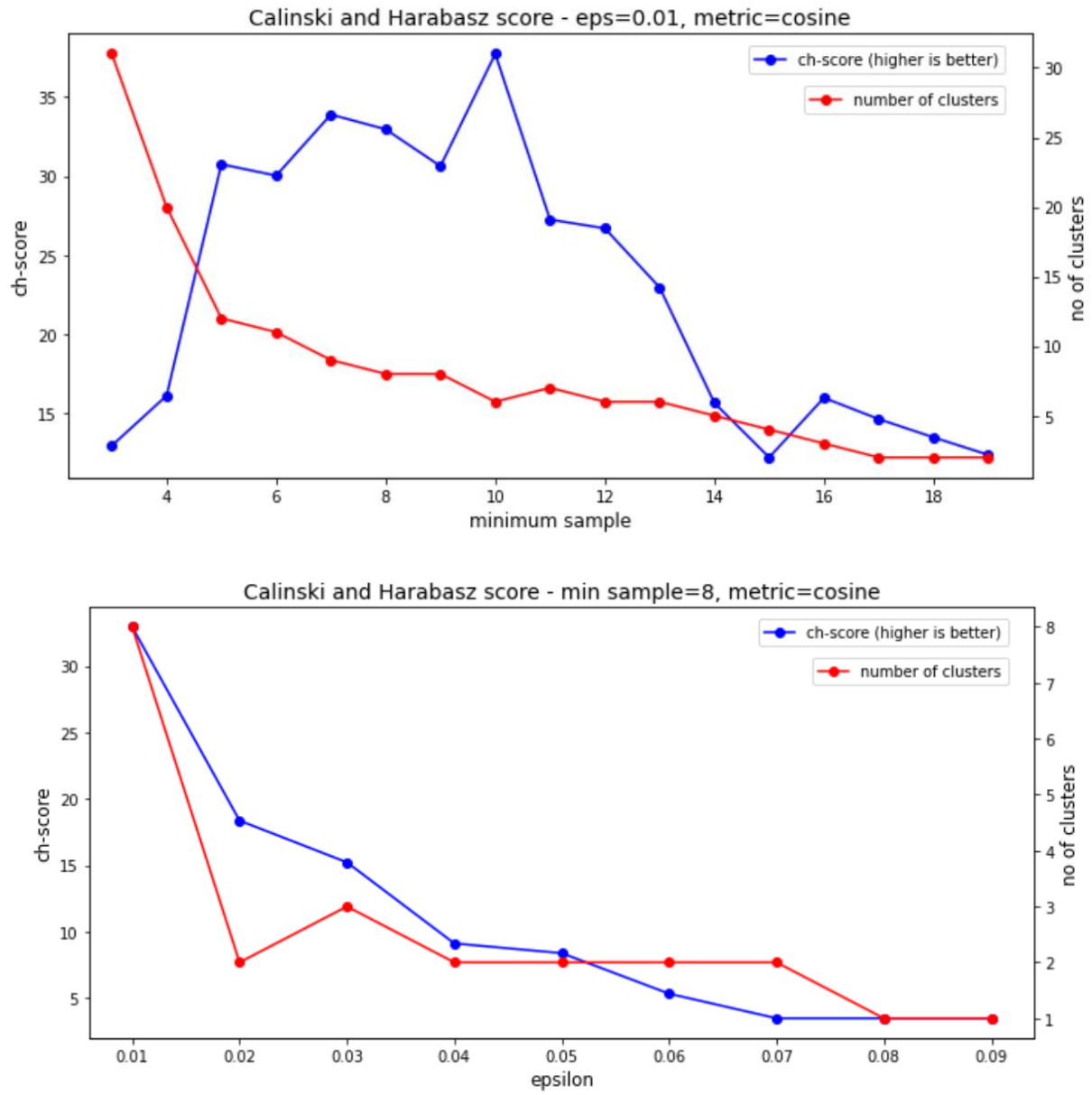


Figure 5.1.3d. Calinski-Harabasz scores: Cosine distance

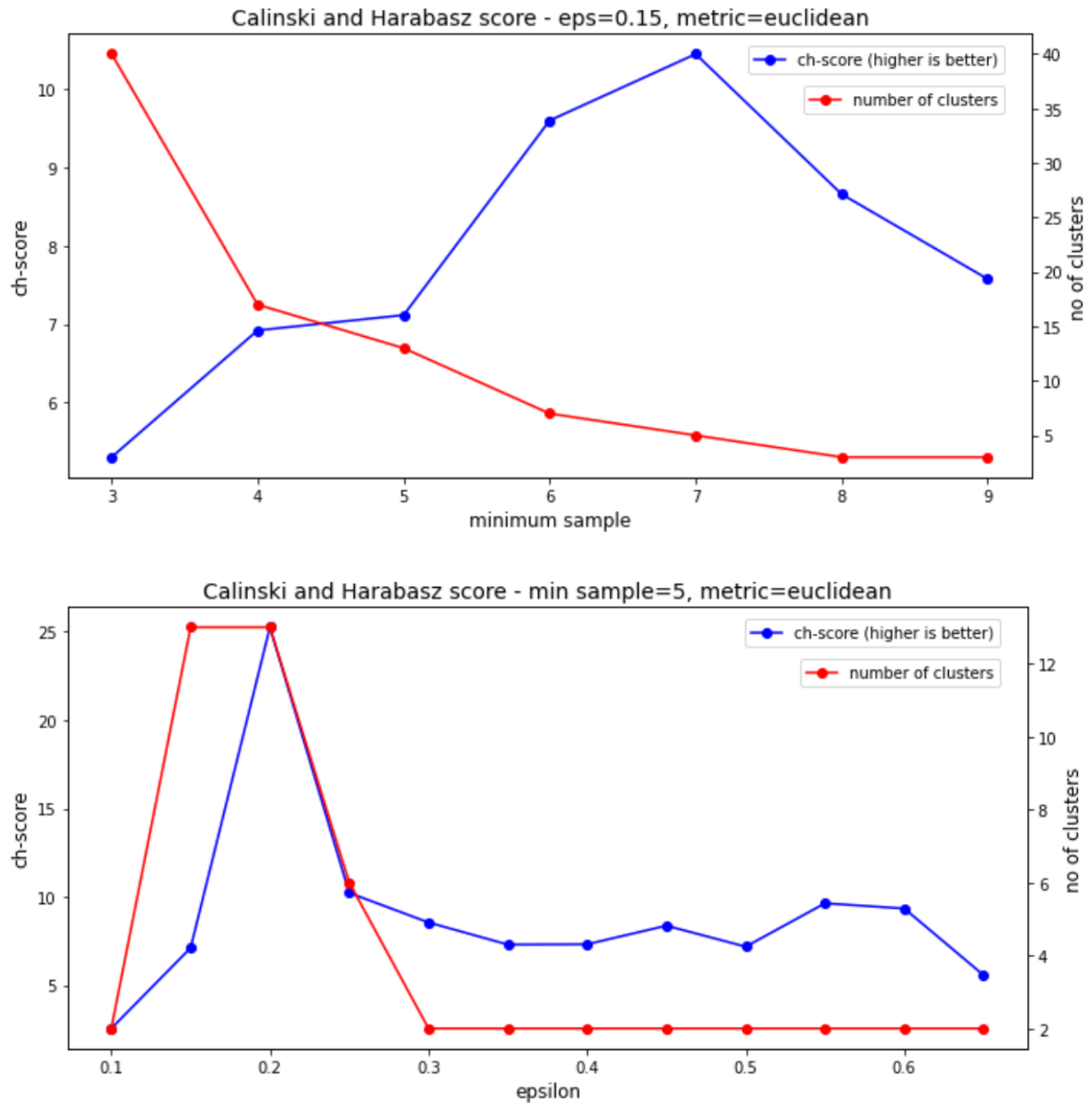


Figure 5.1.3e. Calinski-Harabasz scores: Euclidean distance

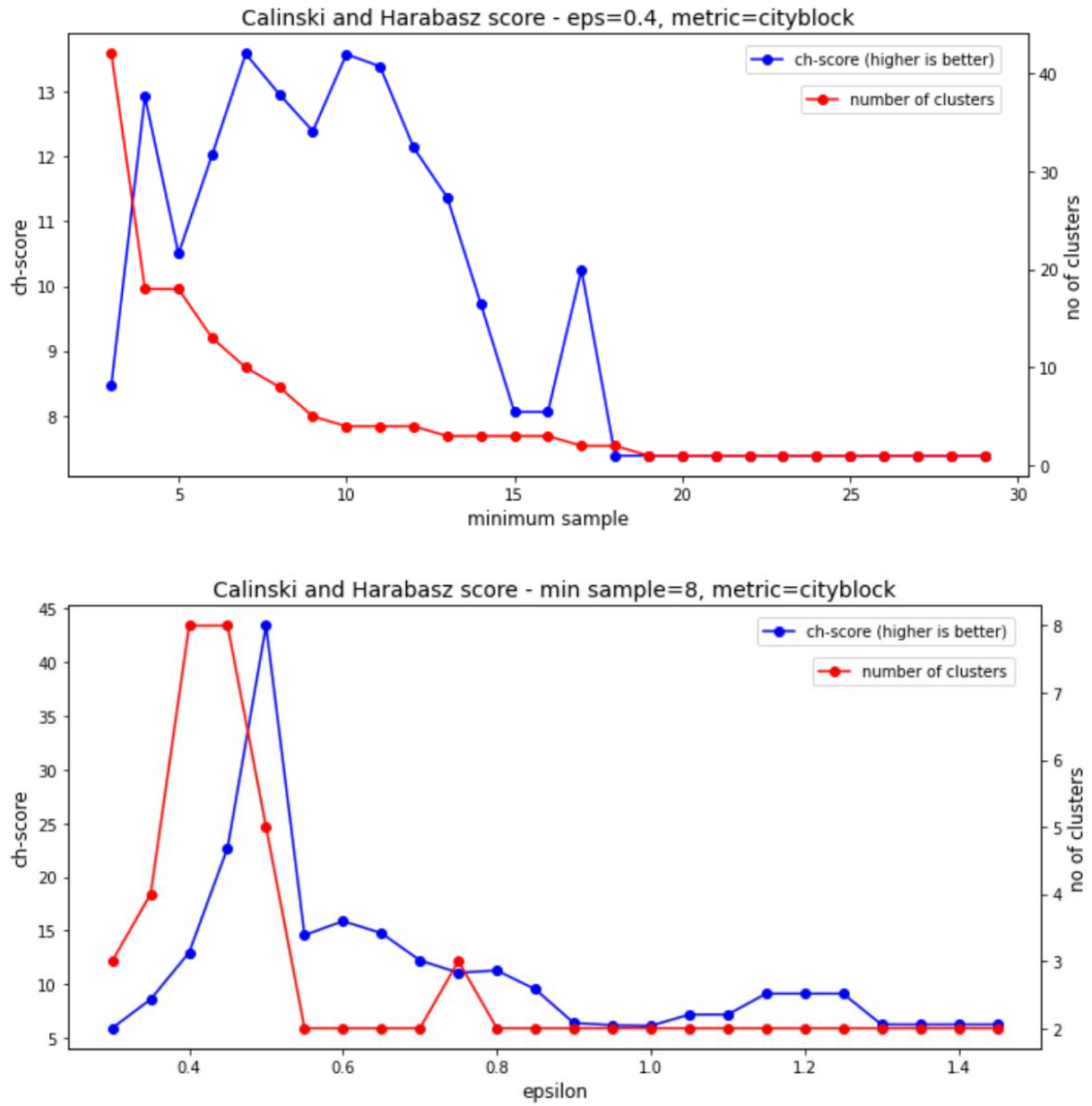


Figure 5.1.3f. Calinski-Harabasz scores: Cityblock distance

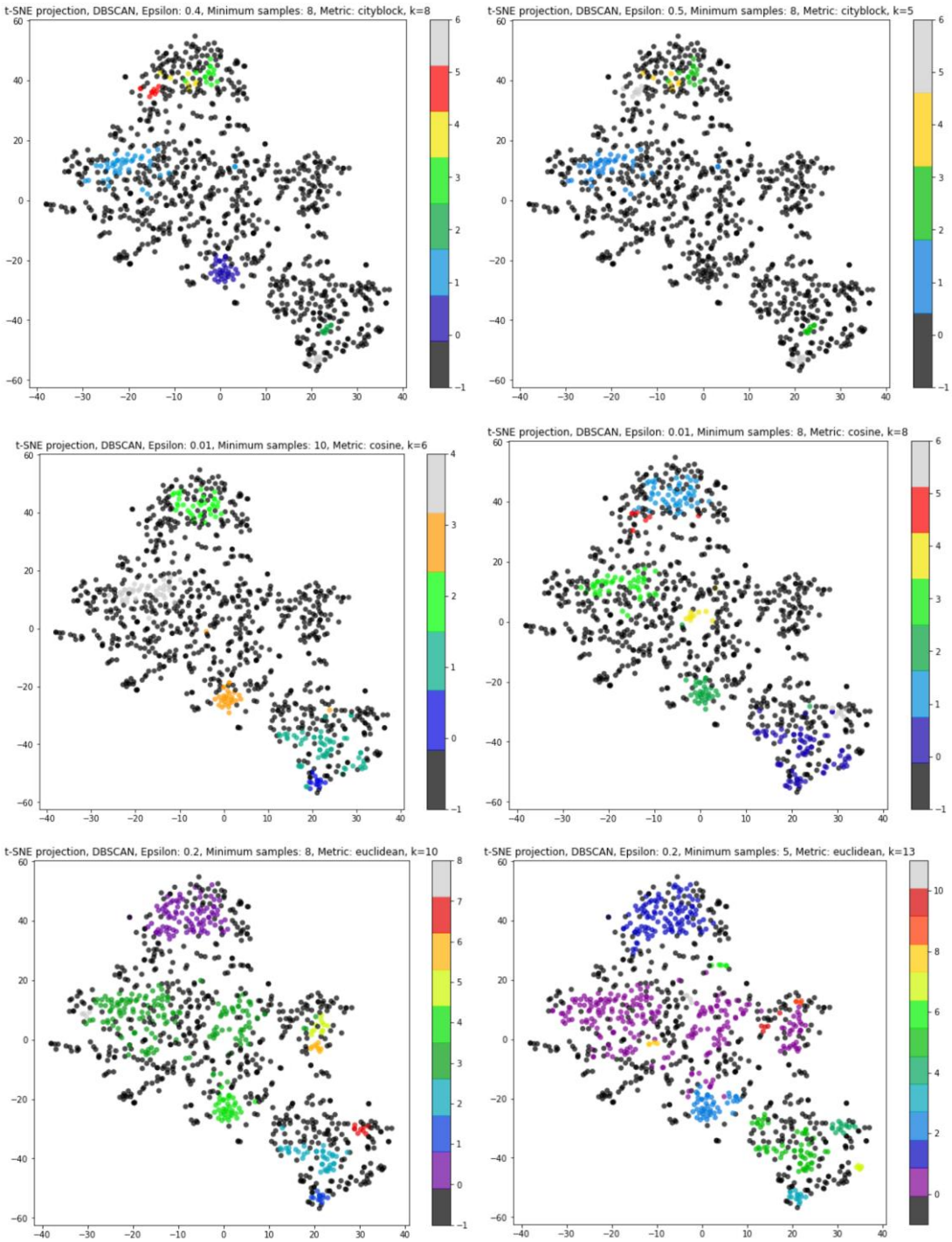


Figure 5.1.3g. t-SNE projections: Cityblock (top), Cosine (middle) and Euclidean (bottom) with different combinations of epsilons and minimum samples

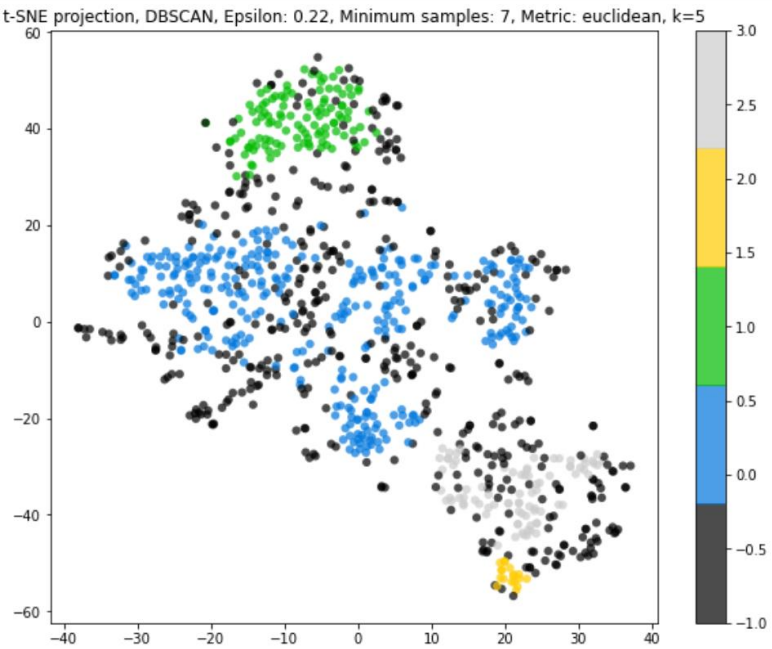


Figure 5.1.3h. t-SNE projection: Euclidean with eps=0.22 and min samples=7

5.2. ICMLA 2014 Accepted Papers Dataset

5.2.1. K-Means Clustering

	k	completeness_euclidean	completeness_cosines	homogeneity_euclideans	homogeneity_cosines
0	2	0.585970	0.306342	0.090153	0.066738
1	4	0.421106	0.357319	0.151228	0.152740
2	6	0.505761	0.434965	0.260263	0.241011
3	8	0.510553	0.516255	0.317248	0.325441
4	10	0.525276	0.481175	0.365347	0.334768
5	12	0.542419	0.523547	0.419864	0.398371
6	14	0.518510	0.545460	0.420758	0.446957
7	16	0.538494	0.552170	0.465632	0.476606
8	18	0.570144	0.565922	0.505076	0.500566
9	20	0.560604	0.561677	0.523101	0.516710
10	22	0.581348	0.560059	0.558515	0.531264
11	24	0.605560	0.562309	0.597427	0.543523
12	26	0.597457	0.568320	0.600817	0.567968
13	28	0.603318	0.570369	0.622020	0.585900
14	30	0.612626	0.573218	0.646794	0.598361
15	32	0.613495	0.581640	0.658287	0.619375
16	34	0.612101	0.586500	0.667599	0.631890
17	36	0.630284	0.591631	0.697218	0.645428
18	38	0.616527	0.596263	0.689426	0.657943
19	40	0.619899	0.606986	0.699344	0.681607
20	42	0.622651	0.611578	0.714457	0.694883
21	44	0.626393	0.613308	0.725949	0.706584
22	46	0.626823	0.619410	0.734293	0.725054
23	48	0.630276	0.621747	0.745234	0.739143

Table 5.2.1a. Completeness and Homogeneity Scores for k-means Euclidean and Cosine

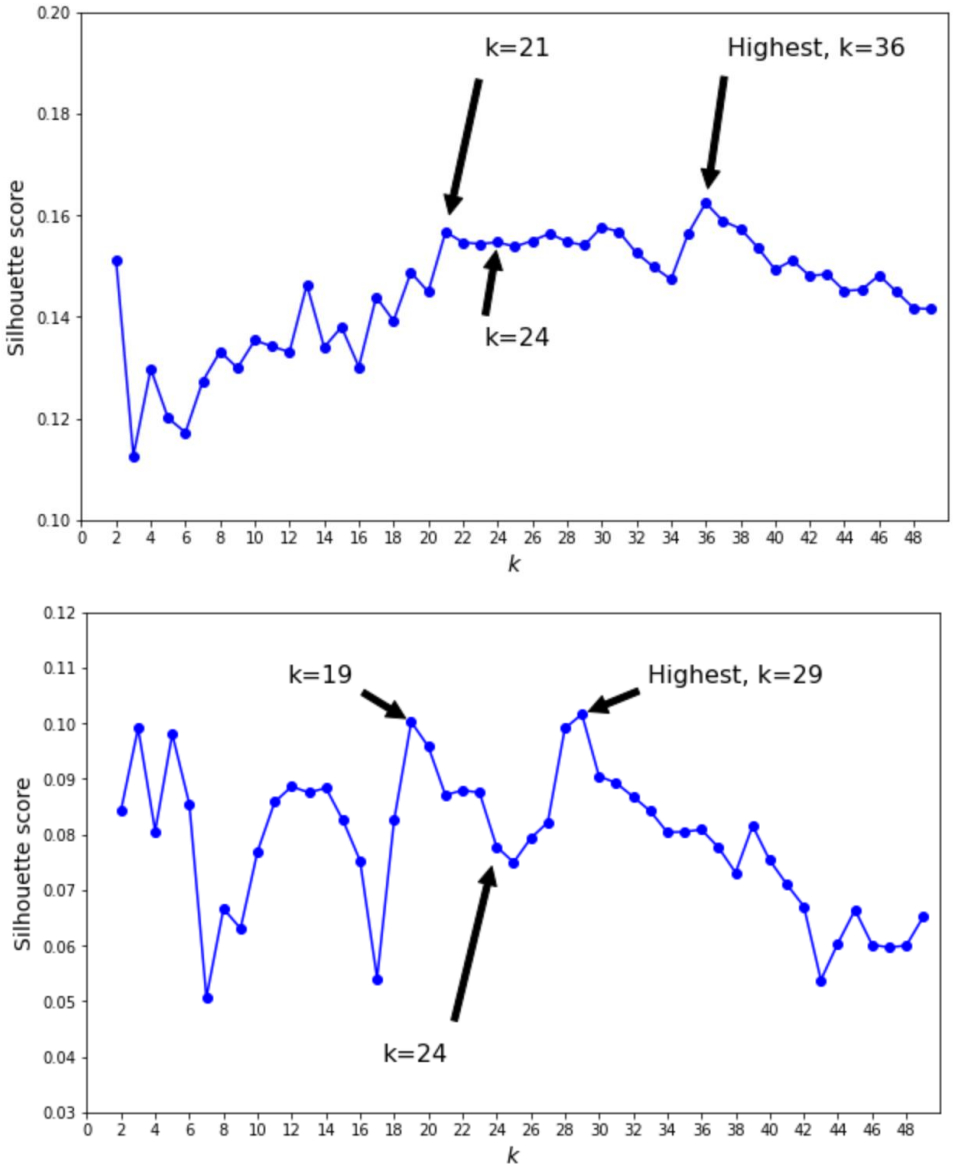


Figure 5.2.1a. Silhouette Scores: k-means Euclidean (top) vs k-means Cosine (bottom)

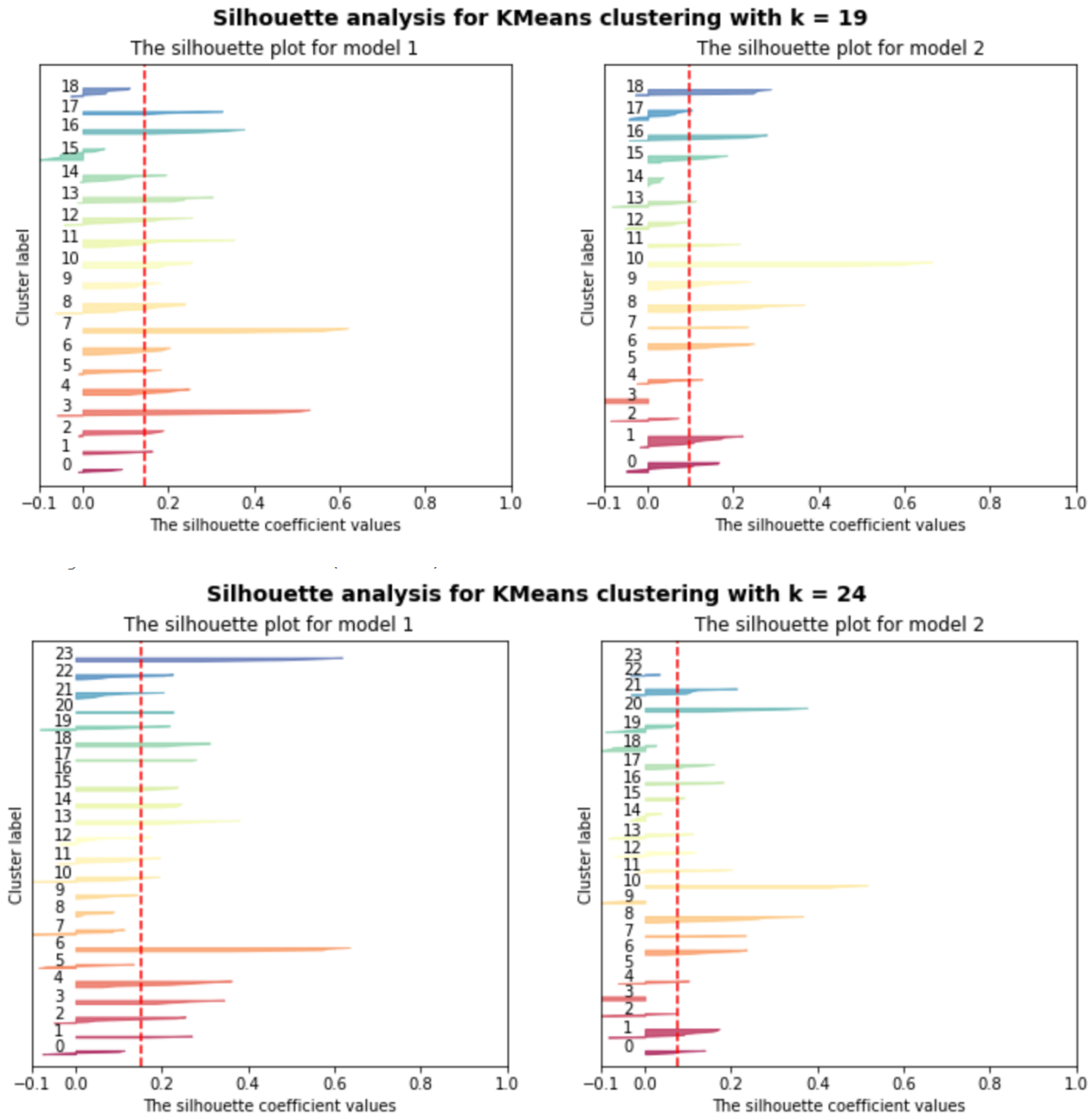


Figure 5.2.1b. Silhouette Analysis: K-Means Euclideans (left) vs Cosines (right)

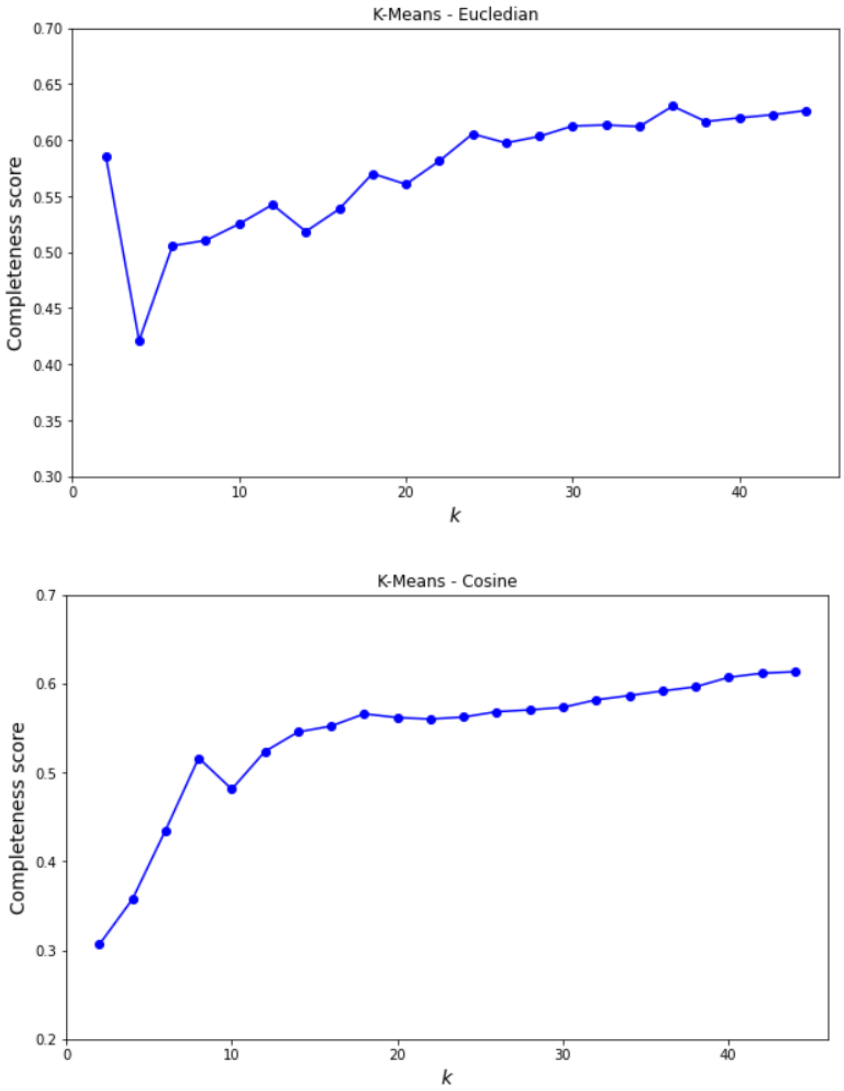


Figure 5.2.1c. Completeness Scores: k-means Euclidean (top) vs k-means Cosine (bottom)

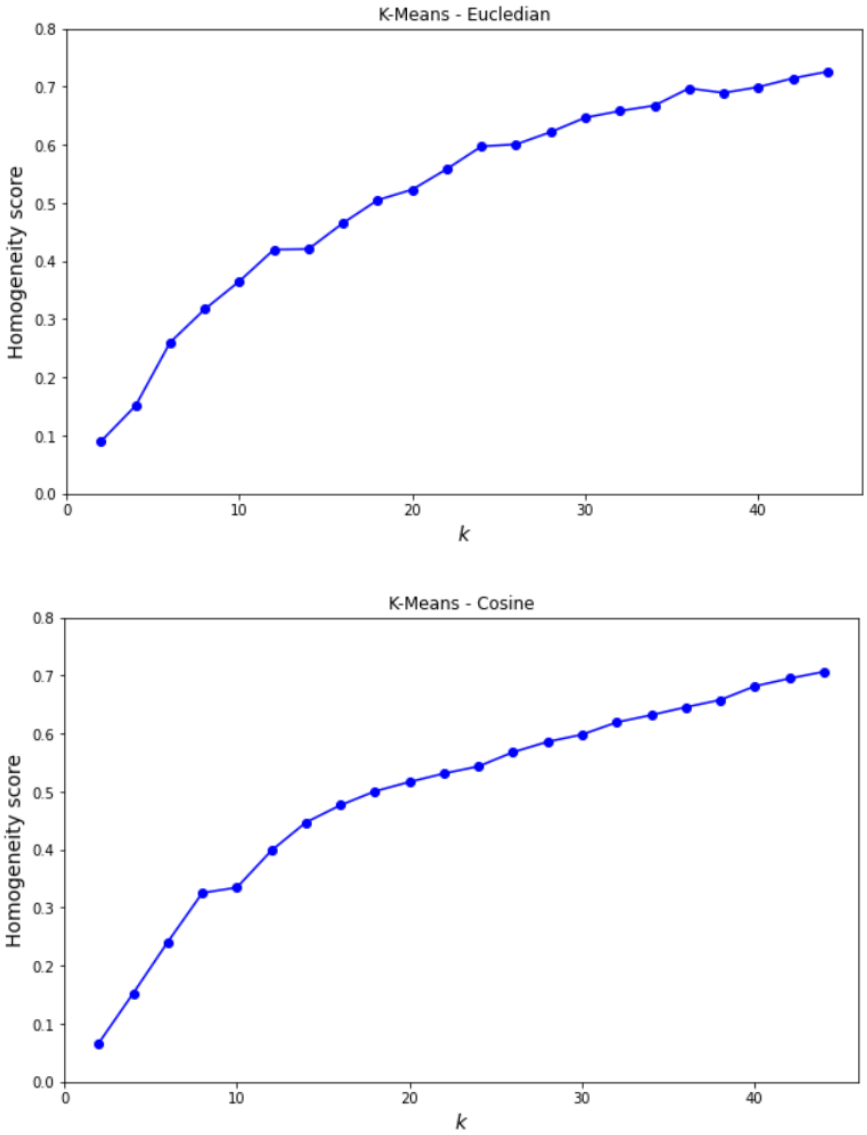


Figure 5.2.1d. Homogeneity Scores: k-means Euclidean (top) vs Cosine (bottom)

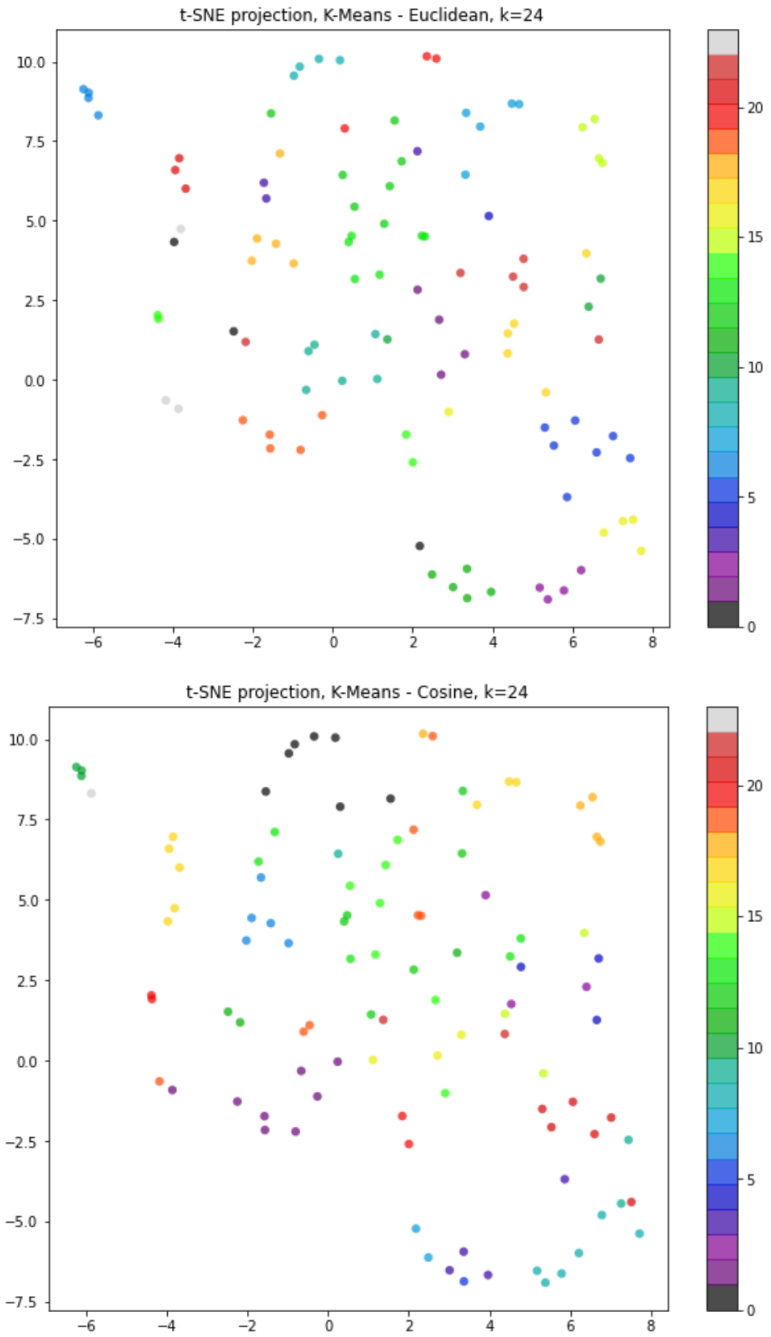


Figure 5.2.1e. t-SNE projections (k=24): k-means Euclidean (top) vs Cosine (bottom)

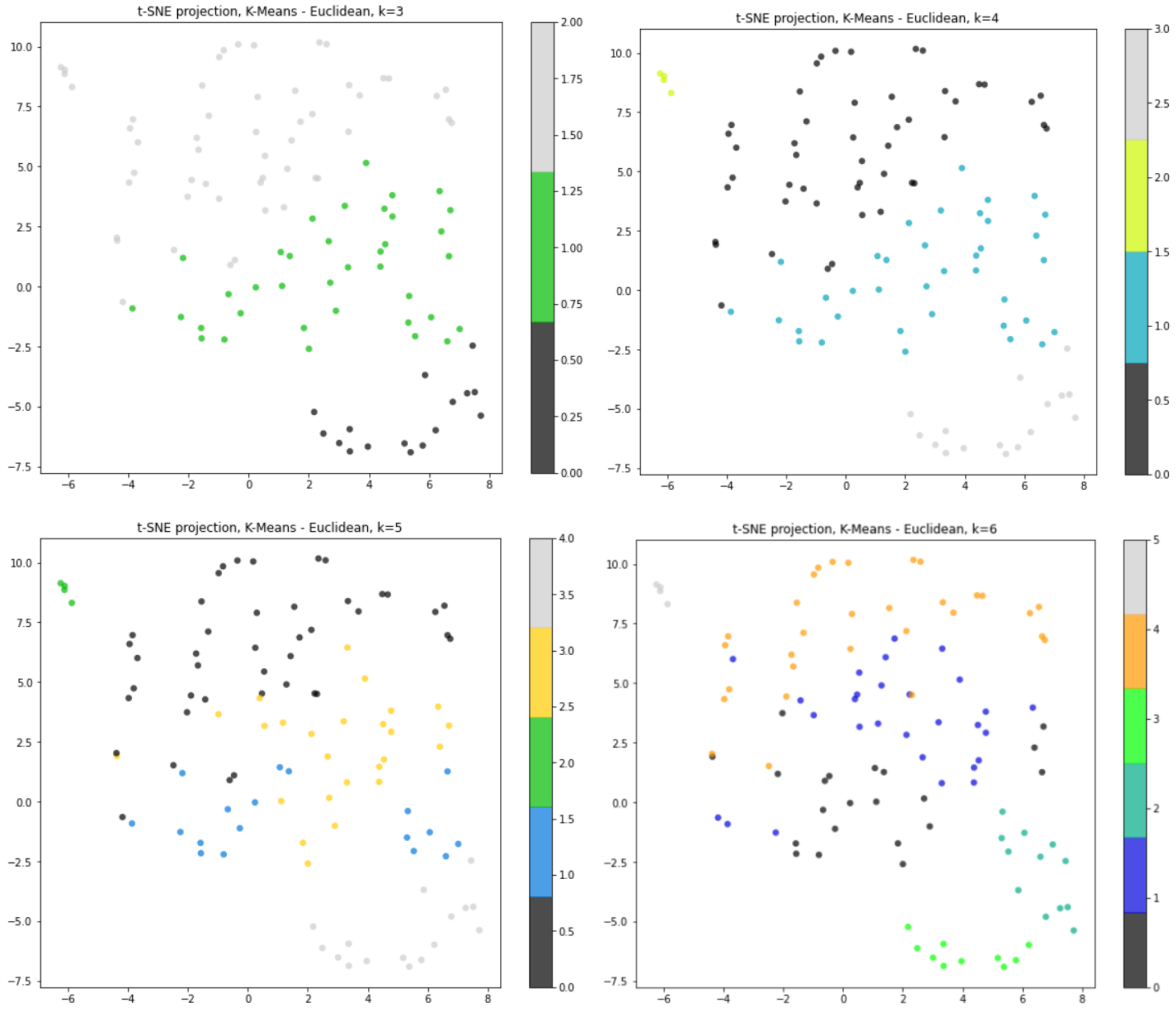


Figure 5.2.1f. t-SNE projections: k-means Euclidean as k was increased from 3-6

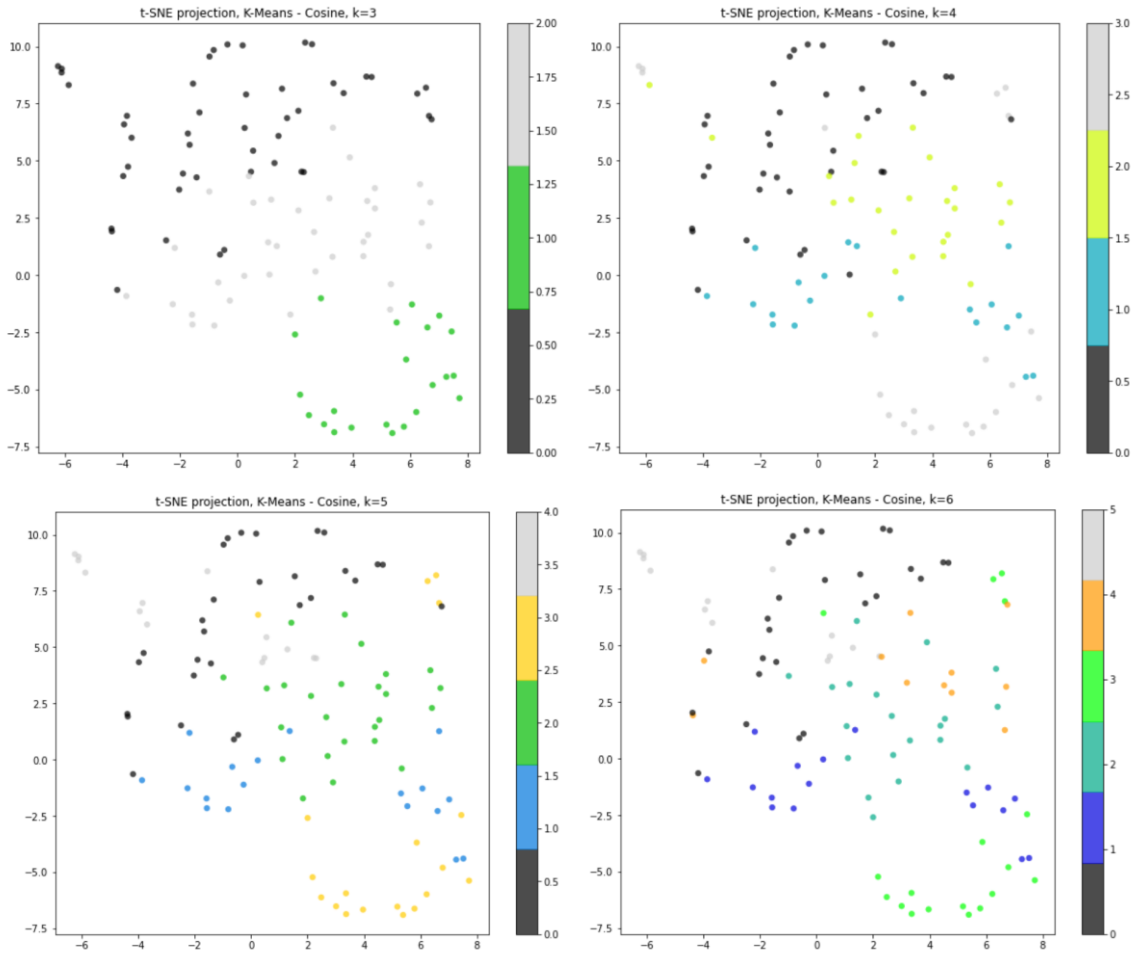


Figure 5.2.1g. t-SNE projections: k-means Cosine as k was increased from 3-6

Table 5.2.1b. k-means Euclidean: Labelling table

	label	session		label	session
94	0	Adaptive Data-Driven Modeling in Dynamic Envir...	**103**	12	Machine learning of graphical models in static...
38	0	Medicine and Bioinformatics	**35**	12	Medicine and Bioinformatics
46	0	Information Retrieval I	**54**	12	Science and Industry
0	1	Ensemble Methods	**45**	12	Information Retrieval I
102	1	Machine learning of graphical models in static...	**59**	13	Machine Learning II
3	1	Ensemble Methods	**91**	13	Machine Learning Algorithms, Systems and Appli...
70	1	Machine Learning for Predictive Models II	**51**	13	Information Retrieval II
25	2	Neural Network II	**44**	13	Real-time Systems and Industry
29	2	Neural Network II	**41**	13	Real-time Systems and Industry
20	2	Neural Networks I	**40**	13	Real-time Systems and Industry
22	2	Neural Networks I	**58**	14	Machine Learning II
39	3	Medicine and Bioinformatics	**18**	14	Machine Learning I
86	3	Machine Learning in Energy Applications	**23**	14	Neural Networks I
62	3	Medicine, Science and Music	**96**	14	Adaptive Data-Driven Modeling in Dynamic Envir...
4	4	Ensemble Methods	**82**	15	Machine Learning in Information and System Sec...
43	5	Real-time Systems and Industry	**83**	15	Machine Learning in Information and System Sec...
75	5	Machine Learning Applications in Education I	**95**	15	Adaptive Data-Driven Modeling in Dynamic Envir...
37	5	Medicine and Bioinformatics	**33**	15	Semi-Supervised Learning
104	5	Machine learning of graphical models in static...	**42**	16	Real-time Systems and Industry
11	5	Feature Extraction and Selection	**100**	16	Machine Learning in Visual Information Processing
12	5	Feature Extraction and Selection	**26**	16	Neural Network II
13	5	Feature Extraction and Selection	**63**	16	Medicine, Science and Music
89	6	Machine Learning Algorithms, Systems and Appli...	**10**	16	Feature Extraction and Selection
92	6	Machine Learning Algorithms, Systems and Appli...	**21**	17	Neural Networks I
56	6	Machine Learning II	**31**	17	Semi-Supervised Learning
90	6	Machine Learning Algorithms, Systems and Appli...	**36**	17	Medicine and Bioinformatics
49	7	Information Retrieval II	**61**	17	Medicine, Science and Music
15	7	Machine Learning I	**50**	17	Information Retrieval II
47	7	Information Retrieval I	**68**	18	Machine Learning for Predictive Models I
9	7	Applications in Security	**2**	18	Ensemble Methods
81	7	Machine Learning in Information and System Sec...	**5**	18	Applications in Security
79	8	Machine Learning Applications in Education II	**64**	18	Medicine, Science and Music
80	8	Machine Learning Applications in Education II	**69**	18	Machine Learning for Predictive Models II
93	8	Machine Learning Algorithms, Systems and Appli...	**53**	19	Science and Industry
84	8	Machine Learning in Information and System Sec...	**74**	19	Machine Learning Applications in Education I
76	9	Machine Learning Applications in Education I	**72**	19	Machine Learning for Predictive Models II
32	9	Semi-Supervised Learning	**71**	19	Machine Learning for Predictive Models II
65	9	Machine Learning for Predictive Models I	**48**	19	Information Retrieval I
66	9	Machine Learning for Predictive Models I	**98**	20	Machine Learning in Visual Information Processing
8	9	Applications in Security	**101**	20	Machine Learning in Visual Information Processing
57	9	Machine Learning II	**52**	20	Science and Industry
77	10	Machine Learning Applications in Education II	**67**	21	Machine Learning for Predictive Models I

6	10	Applications in Security	**60**	21	Medicine, Science and Music
7	10	Applications in Security	**30**	21	Semi-Supervised Learning
24	11	Neural Networks I	**97**	22	Machine Learning in Visual Information Processing
27	11	Neural Network II	**14**	22	Feature Extraction and Selection
87	11	Machine Learning in Energy Applications	**16**	22	Machine Learning I
99	11	Machine Learning in Visual Information Processing	**17**	22	Machine Learning I
28	11	Neural Network II	**1**	22	Ensemble Methods
85	12	Machine Learning in Energy Applications	**19**	22	Machine Learning I
78	12	Machine Learning Applications in Education II	**55**	23	Science and Industry
73	12	Machine Learning Applications in Education I	**34**	23	Semi-Supervised Learning
			88	23	Machine Learning in Energy Applications

Table 5.1.2c. k-means Cosine: Labelling table

	label	session		label	session
52	0	Science and Industry	**81**	12	Machine Learning in Information and System Sec...
73	0	Machine Learning Applications in Education I	**102**	12	Machine learning of graphical models in static...
79	0	Machine Learning Applications in Education II	**40**	13	Real-time Systems and Industry
80	0	Machine Learning Applications in Education II	**14**	13	Feature Extraction and Selection
84	0	Machine Learning in Information and System Sec...	**5**	13	Applications in Security
35	0	Medicine and Bioinformatics	**1**	13	Ensemble Methods
93	0	Machine Learning Algorithms, Systems and Appli...	**39**	13	Medicine and Bioinformatics
66	1	Machine Learning for Predictive Models I	**85**	14	Machine Learning in Energy Applications
53	1	Science and Industry	**78**	14	Machine Learning Applications in Education II
71	1	Machine Learning for Predictive Models II	**91**	14	Machine Learning Algorithms, Systems and Appli...
72	1	Machine Learning for Predictive Models II	**0**	14	Ensemble Methods
48	1	Information Retrieval I	**45**	14	Information Retrieval I
74	1	Machine Learning Applications in Education I	**103**	14	Machine learning of graphical models in static...
8	1	Applications in Security	**42**	14	Real-time Systems and Industry
88	1	Machine Learning in Energy Applications	**31**	15	Semi-Supervised Learning
61	2	Medicine, Science and Music	**50**	15	Information Retrieval II
4	2	Ensemble Methods	**21**	15	Neural Networks I
7	2	Applications in Security	**3**	16	Ensemble Methods
99	3	Machine Learning in Visual Information Processing	**57**	16	Machine Learning II
27	3	Neural Network II	**70**	16	Machine Learning for Predictive Models II
12	3	Feature Extraction and Selection	**60**	17	Medicine, Science and Music
87	3	Machine Learning in Energy Applications	**30**	17	Semi-Supervised Learning
16	4	Machine Learning I	**49**	17	Information Retrieval II
17	4	Machine Learning I	**47**	17	Information Retrieval I
6	4	Applications in Security	**55**	17	Science and Industry
24	5	Neural Networks I	**67**	17	Machine Learning for Predictive Models I
64	6	Medicine, Science and Music	**9**	17	Applications in Security
68	6	Machine Learning for Predictive Models I	**46**	17	Information Retrieval I
2	6	Ensemble Methods	**98**	18	Machine Learning in Visual Information Processing
69	6	Machine Learning for Predictive Models II	**95**	18	Adaptive Data-Driven Modeling in Dynamic Envir...
86	6	Machine Learning in Energy Applications	**33**	18	Semi-Supervised Learning
38	7	Medicine and Bioinformatics	**83**	18	Machine Learning in Information and System Sec...
28	7	Neural Network II	**82**	18	Machine Learning in Information and System Sec...
25	8	Neural Network II	**59**	19	Machine Learning II
29	8	Neural Network II	**34**	19	Semi-Supervised Learning
20	8	Neural Networks I	**32**	19	Semi-Supervised Learning
22	8	Neural Networks I	**101**	19	Machine Learning in Visual Information Processing
100	8	Machine Learning in Visual Information Processing	**44**	19	Real-time Systems and Industry
26	8	Neural Network II	**62**	19	Medicine, Science and Music
54	9	Science and Industry	**76**	19	Machine Learning Applications in Education I
13	9	Feature Extraction and Selection	**18**	20	Machine Learning I
63	9	Medicine, Science and Music	**58**	20	Machine Learning II

56	10	Machine Learning II	**23**	20	Neural Networks I
90	10	Machine Learning Algorithms, Systems and Appli...	**96**	20	Adaptive Data-Driven Modeling in Dynamic Envir...
92	10	Machine Learning Algorithms, Systems and Appli...	**10**	21	Feature Extraction and Selection
94	11	Adaptive Data-Driven Modeling in Dynamic Envir...	**75**	21	Machine Learning Applications in Education I
19	11	Machine Learning I	**37**	21	Medicine and Bioinformatics
97	11	Machine Learning in Visual Information Processing	**43**	21	Real-time Systems and Industry
41	12	Real-time Systems and Industry	**11**	21	Feature Extraction and Selection
65	12	Machine Learning for Predictive Models I	**104**	21	Machine learning of graphical models in static...
51	12	Information Retrieval II	**77**	22	Machine Learning Applications in Education II
15	12	Machine Learning I	**36**	22	Medicine and Bioinformatics
			89	23	Machine Learning Algorithms, Systems and Appli...

5.2.2. Hierarchical Clustering

	k	cosine-single	cosine-average	cosine-complete	euclidean-single	euclidean-average	euclidean-complete	cityblock-single	cityblock-average	cityblock-complete
0	2	0.601829	0.540787	0.314942	0.669615	0.669615	0.602672	0.669615	0.520184	0.327161
1	3	0.549282	0.437335	0.379742	0.652103	0.622332	0.622332	0.652103	0.462619	0.382375
2	4	0.550750	0.461752	0.381278	0.632470	0.609145	0.390259	0.653476	0.480907	0.449613
3	5	0.560877	0.468268	0.424226	0.581485	0.562680	0.414303	0.592560	0.507973	0.446936
4	6	0.532538	0.499711	0.436176	0.578208	0.550648	0.455200	0.588248	0.529484	0.464956
5	7	0.480619	0.522066	0.462697	0.584603	0.468129	0.468003	0.615533	0.557622	0.479665
6	8	0.487390	0.532924	0.479177	0.585074	0.489725	0.500475	0.613601	0.567556	0.487499
7	9	0.490479	0.547647	0.505898	0.593008	0.493800	0.510189	0.608493	0.549186	0.486495
8	10	0.495397	0.557242	0.515105	0.572367	0.517146	0.516874	0.607621	0.558417	0.500508
9	11	0.504567	0.558648	0.520747	0.579144	0.520818	0.528245	0.619001	0.566025	0.510529
10	12	0.529402	0.571616	0.530916	0.586414	0.526758	0.529906	0.596030	0.569741	0.522670
11	13	0.531820	0.582755	0.532892	0.595336	0.537483	0.549762	0.595477	0.564990	0.541628
12	14	0.549430	0.580053	0.545892	0.597762	0.549007	0.562095	0.598019	0.569367	0.541857
13	15	0.566135	0.580774	0.553666	0.599168	0.553182	0.562855	0.604810	0.576678	0.553116
14	16	0.567098	0.585873	0.554475	0.609430	0.556276	0.567527	0.612959	0.588747	0.566817
15	17	0.572724	0.591511	0.562950	0.608120	0.558460	0.570997	0.610919	0.588885	0.580978
16	18	0.578763	0.591174	0.566410	0.606459	0.561326	0.564918	0.609510	0.594467	0.583056
17	19	0.580541	0.593971	0.569761	0.599757	0.569633	0.573737	0.612755	0.603937	0.590701
18	20	0.580169	0.594522	0.571300	0.598660	0.576145	0.583698	0.611310	0.604365	0.598072
19	21	0.583573	0.598922	0.575109	0.589767	0.570128	0.584634	0.614787	0.612445	0.602417
20	22	0.585019	0.603473	0.581146	0.588849	0.578224	0.589040	0.618895	0.610315	0.606668
21	23	0.584474	0.605874	0.583063	0.591565	0.584174	0.590836	0.608067	0.607829	0.607206
22	24	0.583925	0.611040	0.587383	0.592178	0.584799	0.586379	0.605473	0.611861	0.611294
23	25	0.585144	0.609728	0.585982	0.586505	0.585996	0.592348	0.605806	0.614278	0.612717
24	26	0.586266	0.608167	0.590761	0.589243	0.589377	0.592857	0.603320	0.618783	0.614099
25	27	0.594142	0.611475	0.596591	0.591833	0.593510	0.596198	0.601406	0.619775	0.618431
26	28	0.593307	0.614728	0.595799	0.595543	0.592741	0.595415	0.600295	0.622929	0.622167
27	29	0.597047	0.620210	0.596468	0.603125	0.595742	0.600329	0.602725	0.623183	0.621189
28	30	0.602357	0.622357	0.599697	0.601525	0.600237	0.604192	0.602848	0.625940	0.621362
29	31	0.611190	0.621541	0.602876	0.610698	0.601421	0.608486	0.608581	0.628657	0.621896

Table 5.2.2a. Completeness scores for different combinations of Hierarchical clustering models

	k	cosine-single	cosine-average	cosine-complete	euclidean-single	euclidean-average	euclidean-complete	cityblock-single	cityblock-average	cityblock-complete
0	2	0.010231	0.091853	0.055698	0.034243	0.034243	0.059888	0.034243	0.015501	0.066461
1	3	0.025674	0.145212	0.114170	0.044356	0.092902	0.092902	0.044356	0.091631	0.130819
2	4	0.035057	0.191606	0.157488	0.053680	0.108555	0.130876	0.055463	0.164212	0.176023
3	5	0.045171	0.209099	0.199839	0.093608	0.162940	0.148708	0.095391	0.189984	0.213099
4	6	0.083535	0.256978	0.236069	0.102679	0.208492	0.202222	0.104462	0.211777	0.255902
5	7	0.114834	0.300979	0.266376	0.113499	0.241010	0.239988	0.117064	0.237218	0.282055
6	8	0.124425	0.332491	0.299005	0.123264	0.270806	0.294221	0.126862	0.266824	0.305943
7	9	0.133221	0.352798	0.341293	0.190979	0.285930	0.330886	0.135867	0.317825	0.323706
8	10	0.153797	0.386735	0.365724	0.211322	0.330704	0.347103	0.153168	0.340473	0.350734
9	11	0.159543	0.402795	0.385971	0.223009	0.361698	0.374395	0.160697	0.369539	0.365080
10	12	0.176230	0.433654	0.413264	0.229777	0.392494	0.386305	0.178139	0.386226	0.394129
11	13	0.185638	0.453907	0.427872	0.238417	0.409772	0.423549	0.187695	0.419782	0.425317
12	14	0.226890	0.462594	0.450858	0.252020	0.434332	0.445247	0.198238	0.432850	0.435579
13	15	0.242790	0.477707	0.465243	0.262090	0.448838	0.466793	0.210318	0.451663	0.455831
14	16	0.258799	0.487833	0.477152	0.273582	0.459725	0.481226	0.278965	0.474649	0.481897
15	17	0.270400	0.499326	0.493839	0.282586	0.474367	0.495733	0.294625	0.491679	0.510628
16	18	0.277169	0.514451	0.508346	0.298335	0.485254	0.501896	0.303450	0.503172	0.520842
17	19	0.287152	0.526549	0.520861	0.331497	0.501941	0.520278	0.307622	0.529771	0.537529
18	20	0.296068	0.533870	0.531802	0.340083	0.515479	0.541976	0.316401	0.537092	0.554216
19	21	0.300239	0.549951	0.540145	0.361113	0.521642	0.552157	0.327730	0.555621	0.564342
20	22	0.310133	0.569235	0.553683	0.369440	0.547083	0.562283	0.333476	0.557196	0.574469
21	23	0.318958	0.574982	0.563897	0.373612	0.560621	0.572497	0.363897	0.572390	0.584649
22	24	0.327737	0.587584	0.574023	0.382906	0.567942	0.576116	0.371517	0.582173	0.601336
23	25	0.337492	0.593331	0.576620	0.404404	0.578882	0.590502	0.380867	0.597285	0.611550
24	26	0.347198	0.595928	0.588113	0.414984	0.587018	0.597823	0.388378	0.608778	0.621763
25	27	0.358691	0.604271	0.602498	0.425502	0.597144	0.606166	0.402627	0.618144	0.633256
26	28	0.367327	0.612615	0.606670	0.452844	0.601316	0.610338	0.417197	0.626487	0.643383
27	29	0.373073	0.627000	0.616850	0.499069	0.608845	0.622940	0.427777	0.636668	0.647554
28	30	0.381417	0.632746	0.625194	0.505965	0.620338	0.633067	0.436720	0.644197	0.654875
29	31	0.395802	0.641673	0.633537	0.532328	0.629704	0.644560	0.447329	0.651726	0.665816

Table 5.2.2b. Homogeneity scores for different combinations of Hierarchical clustering models

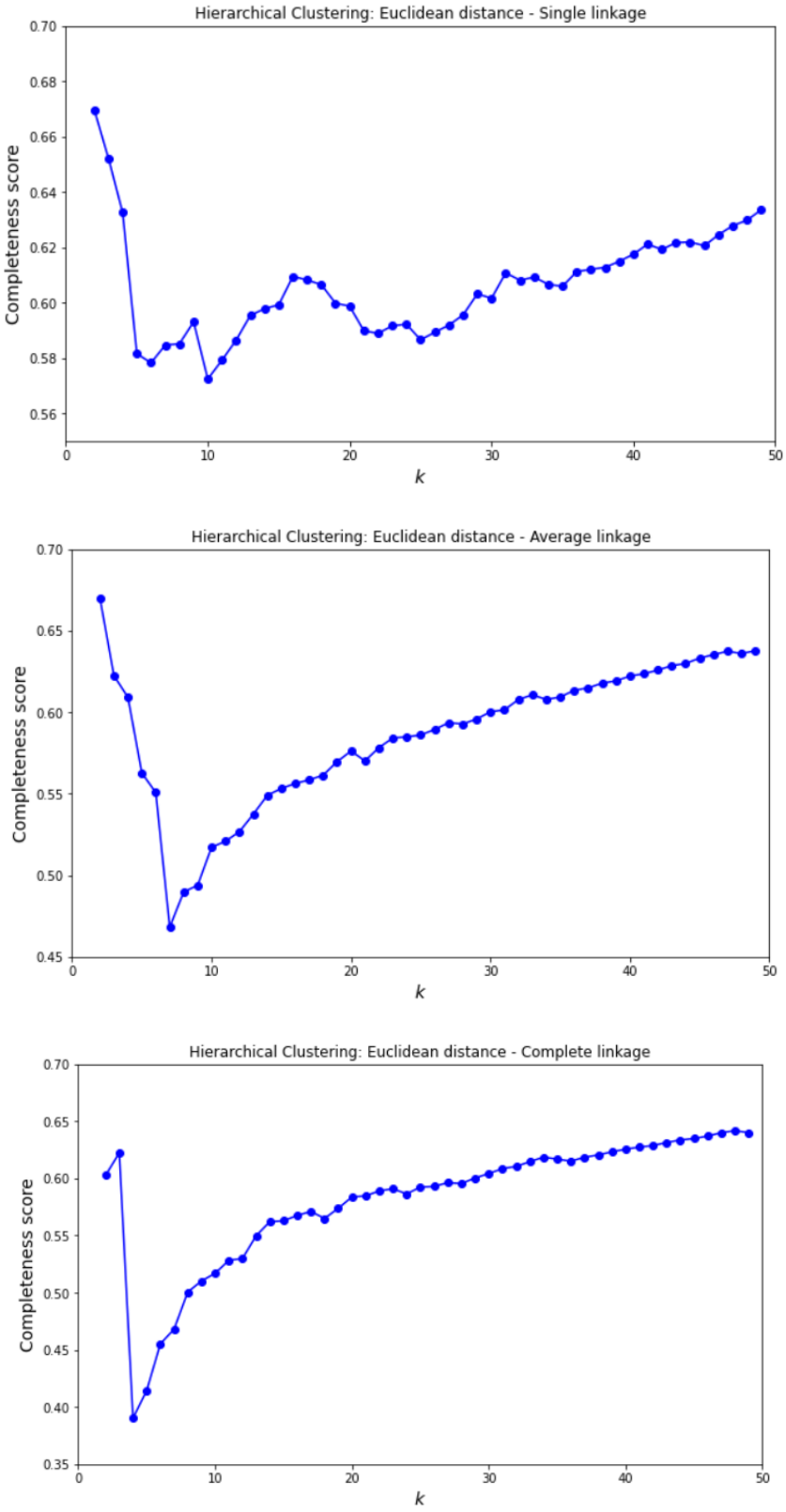


Figure 5.2.2a. Completeness scores: Hierarchical Euclidean models

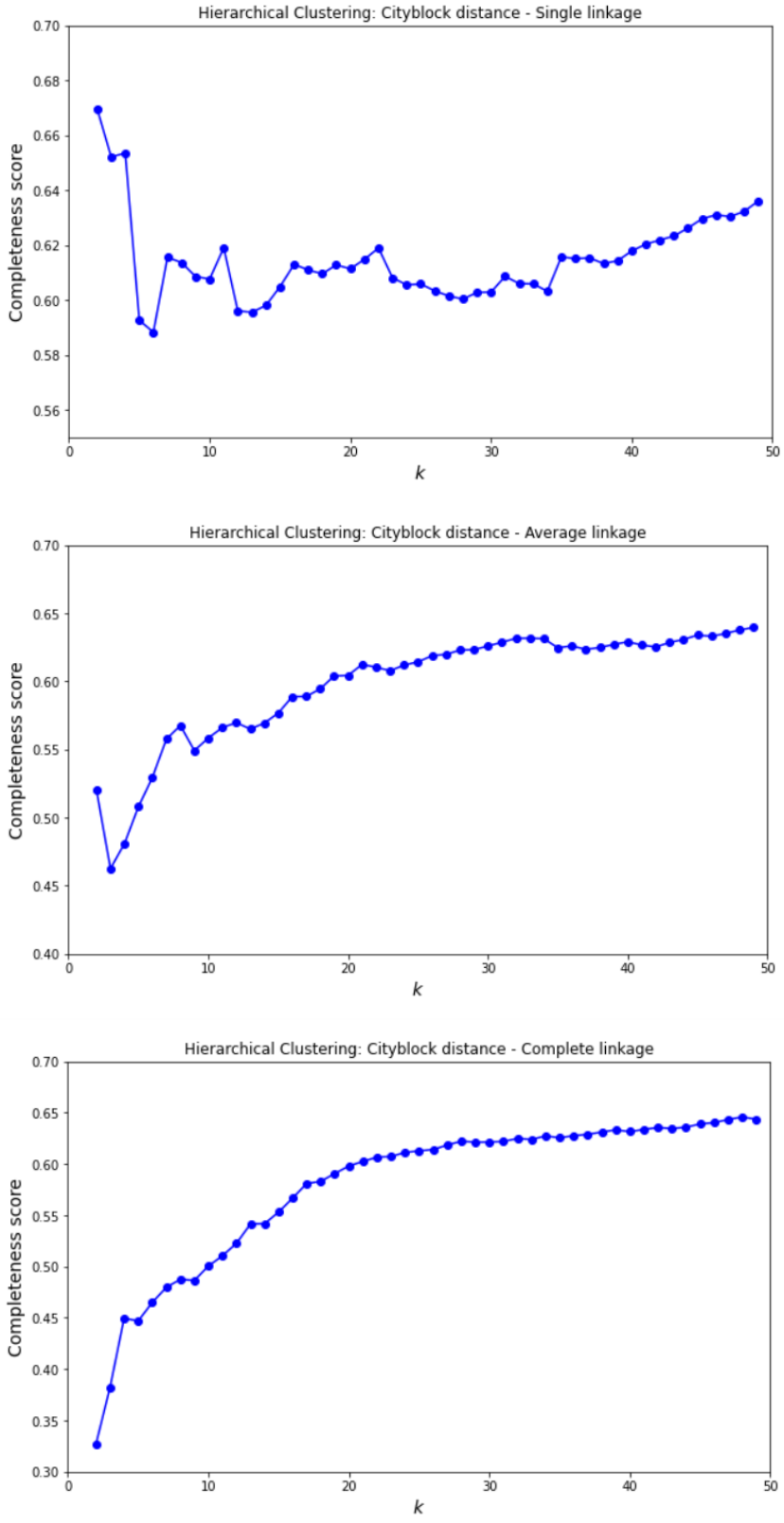


Figure 5.2.2b. Completeness scores: Hierarchical Cityblock models

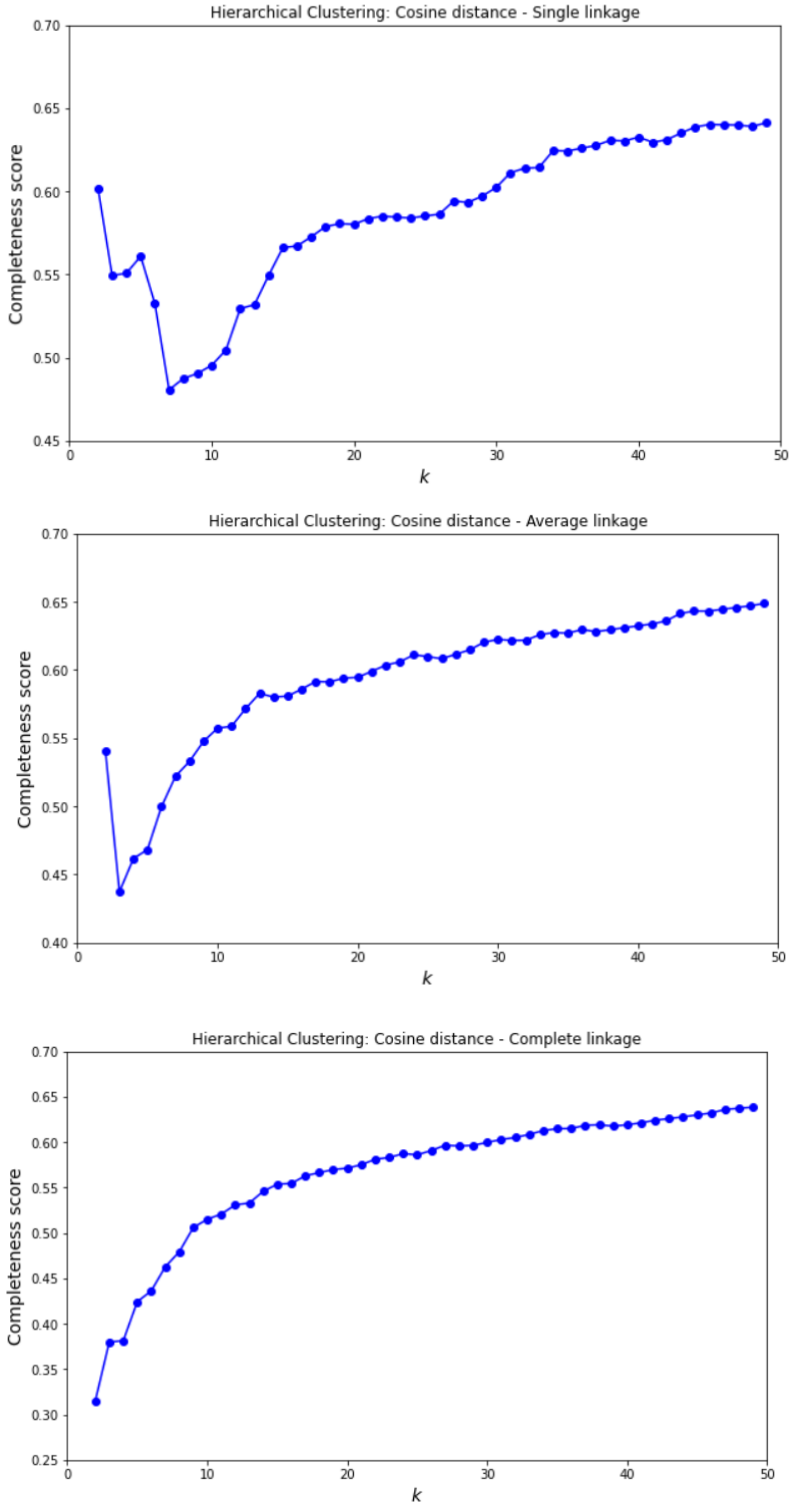


Figure 5.2.2c. Completeness scores: Hierarchical Cosine models

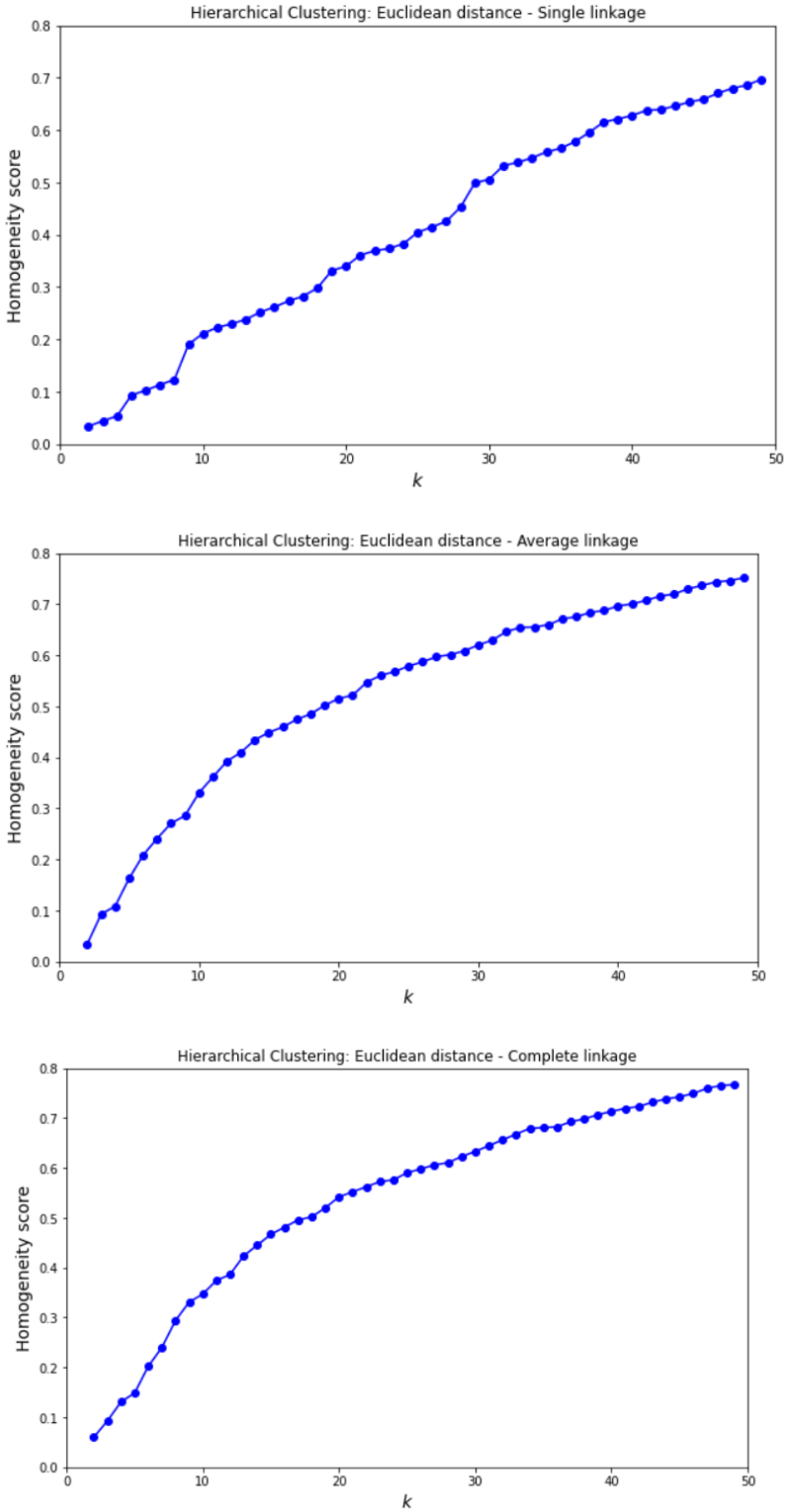


Figure 5.2.2d. Homogeneity scores: Hierarchical Euclidean models

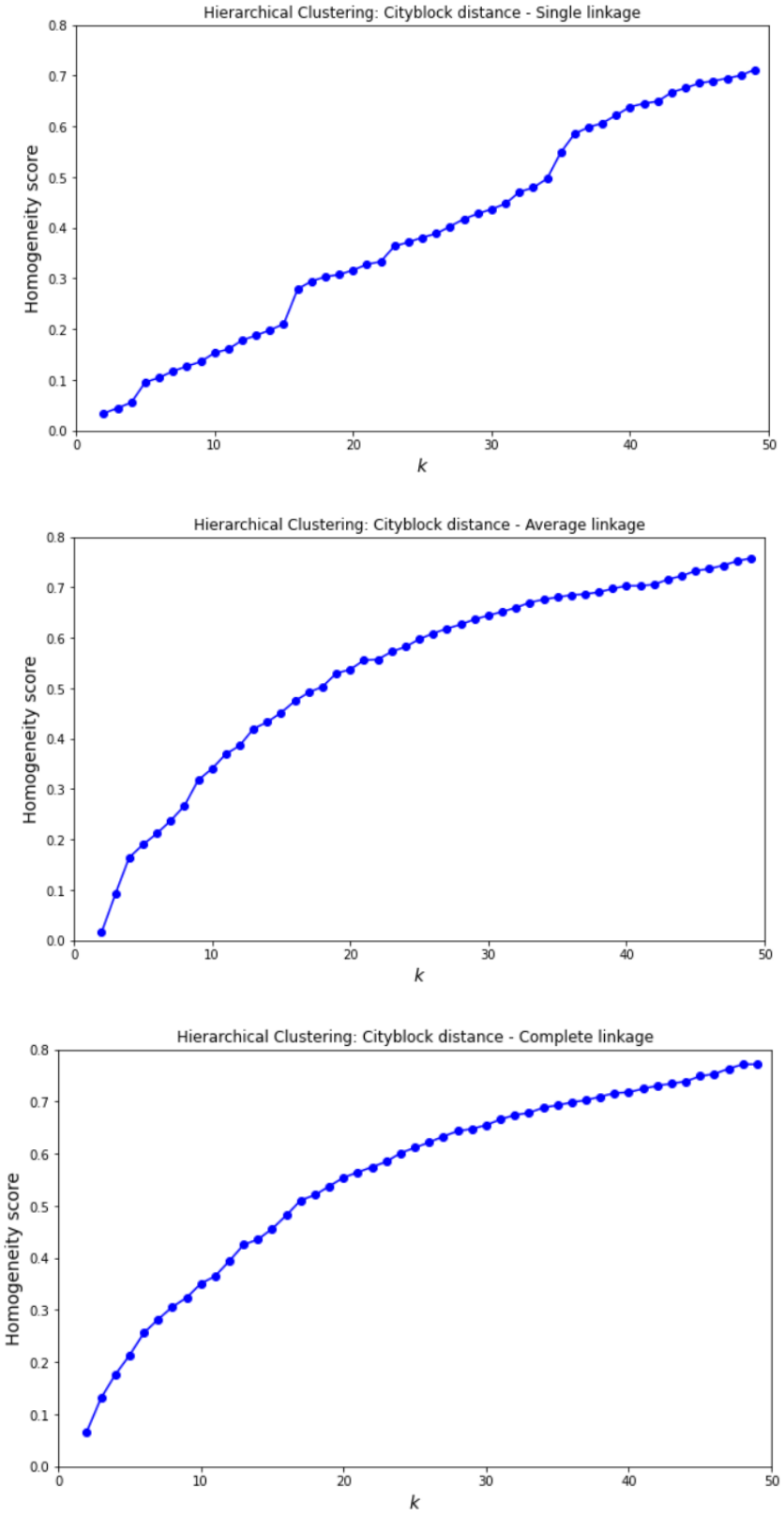


Figure 5.2.2e. Homogeneity scores for Hierarchical Cityblock models

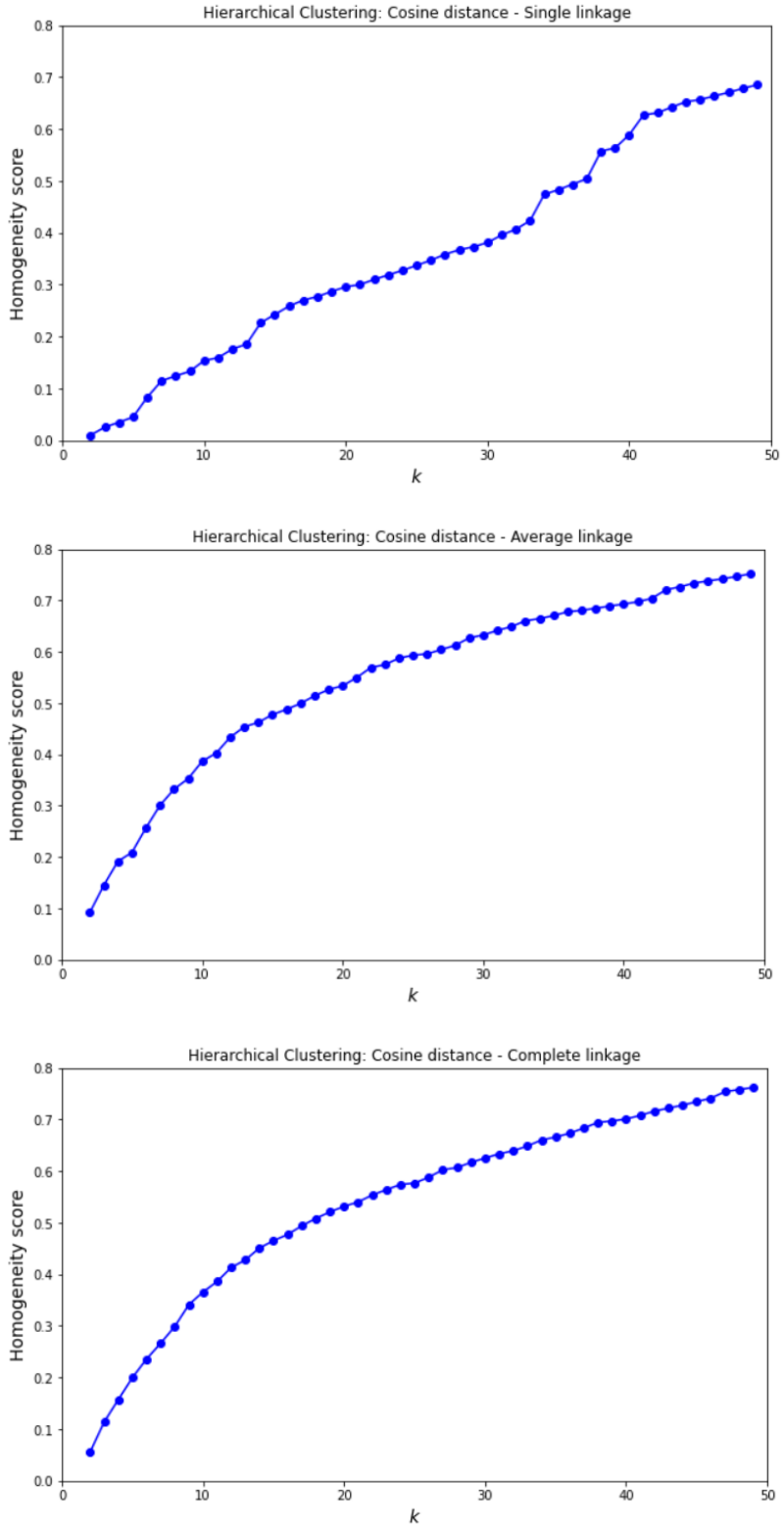


Figure 5.2.2f. Homogeneity scores: Hierarchical Cosine models

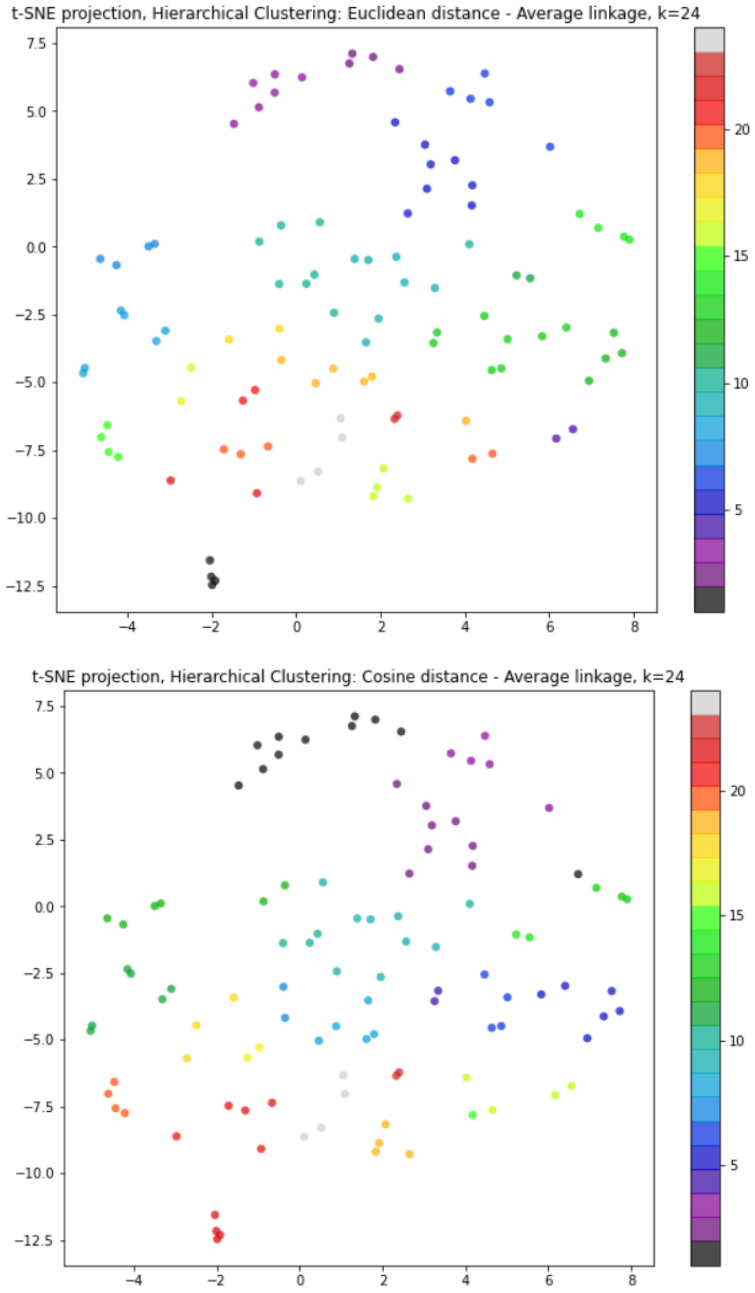


Figure 5.2.2g. t-SNE projections (k=24): Euclidean-average (top) vs Cosine-average (bottom)

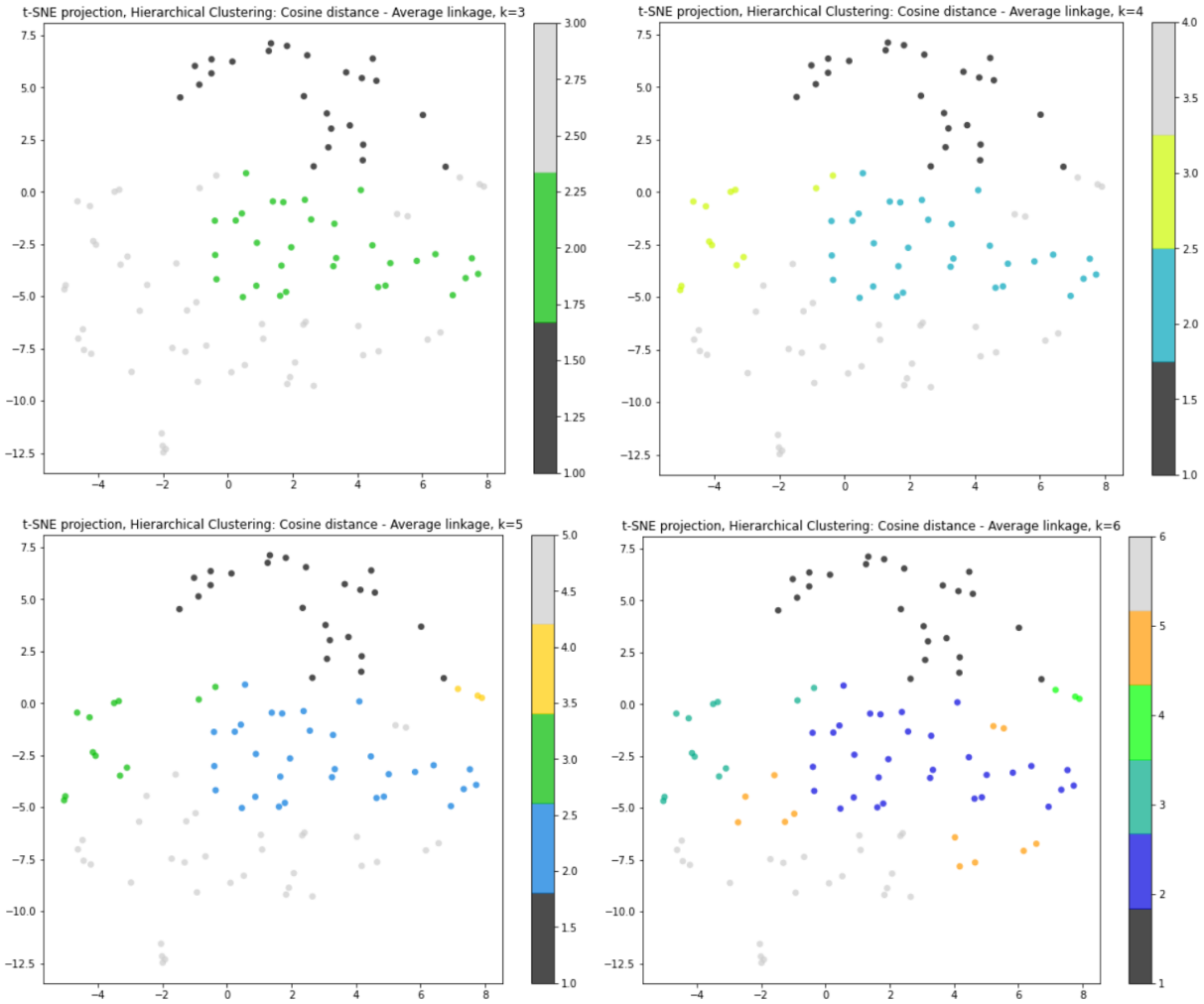


Figure 5.2.2h. t-SNE projections: Cosine-average as k was increased from 3-6

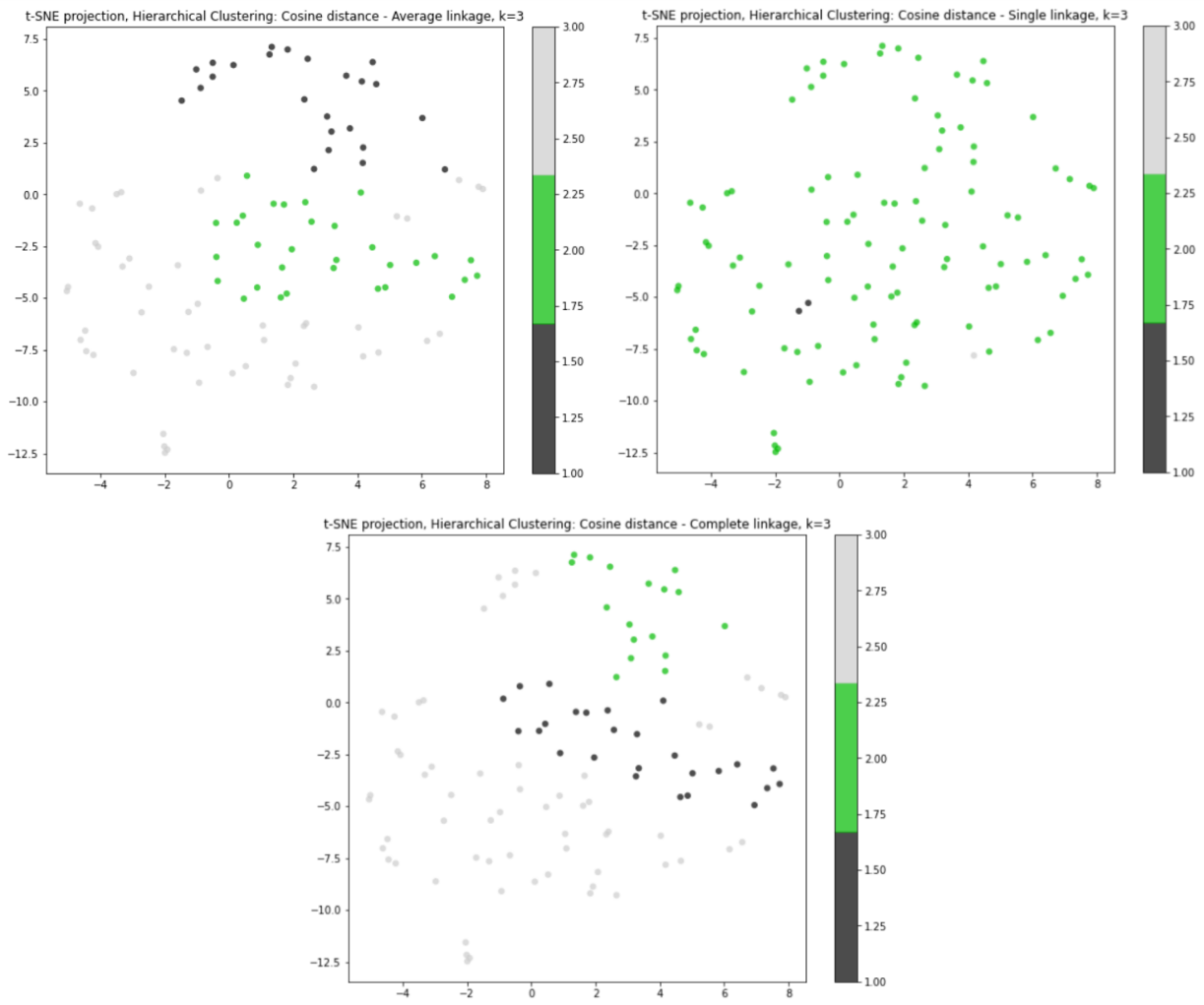


Figure 5.2.2i. t-SNE projections (k=3): Cosine-Single vs Cosine-average vs Cosine-complete

Figure 5.2.2j. Dendogram: Hierarchical Cosine-average-link clustering

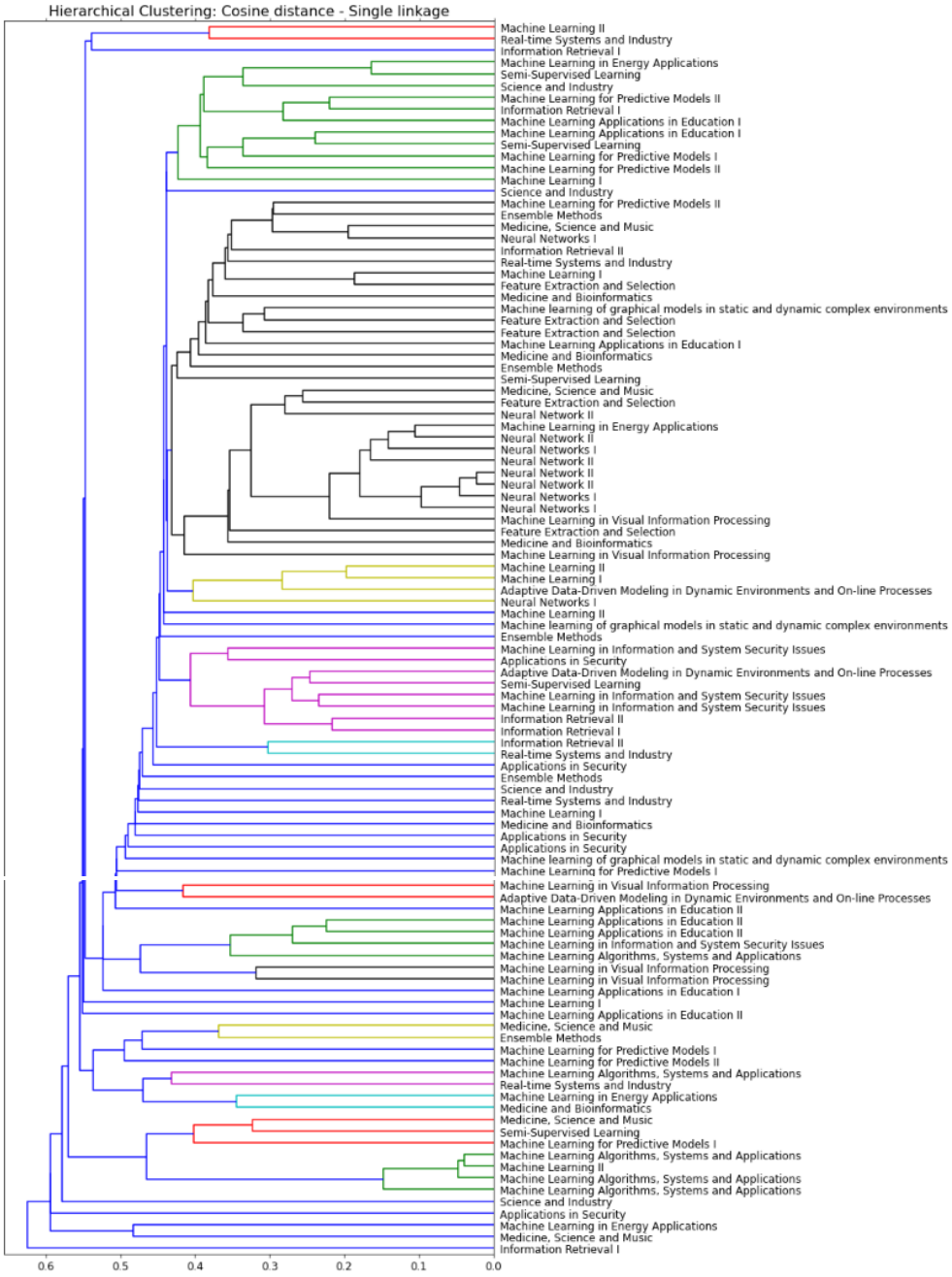


Figure 5.2.2k. Dendogram: Hierarchical Cosine-single-link clustering

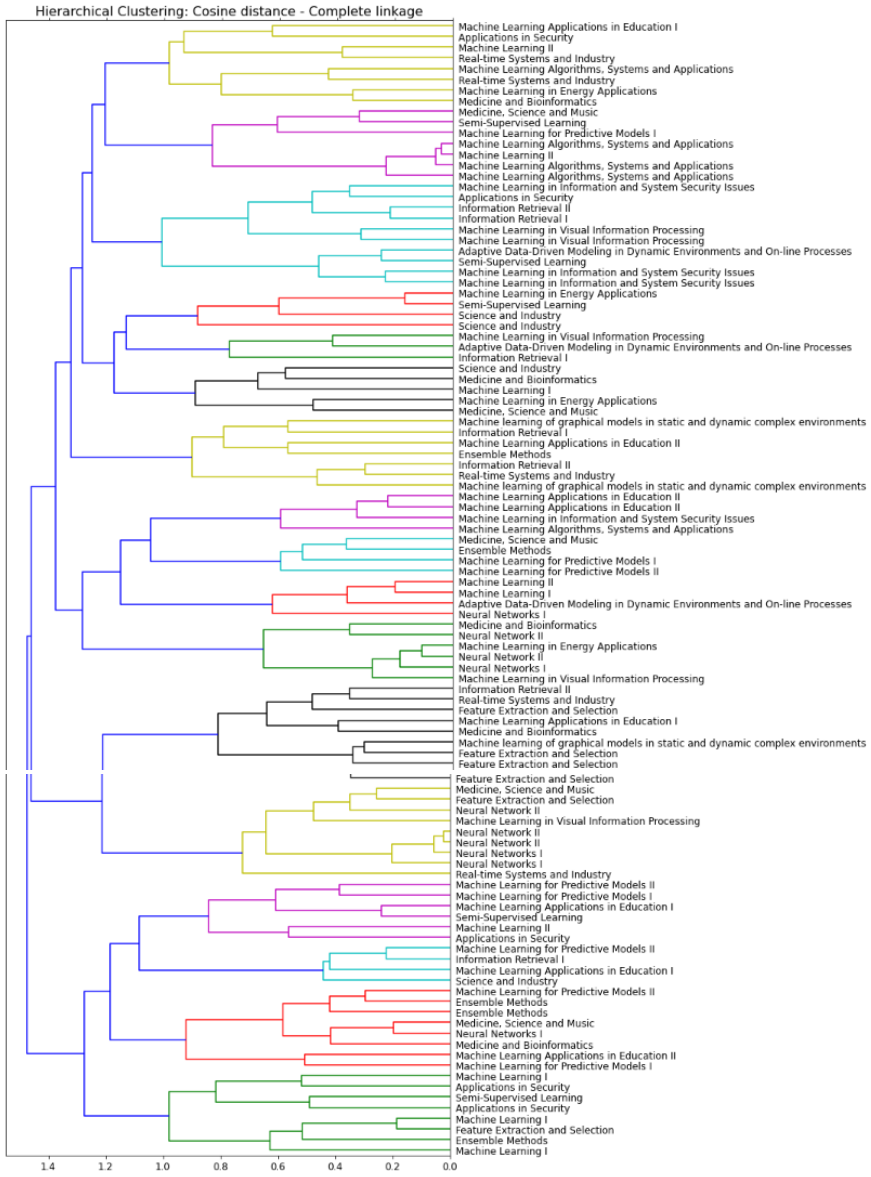


Figure 5.2.2l. Dendogram: Hierarchical Cosine-complete-link clustering

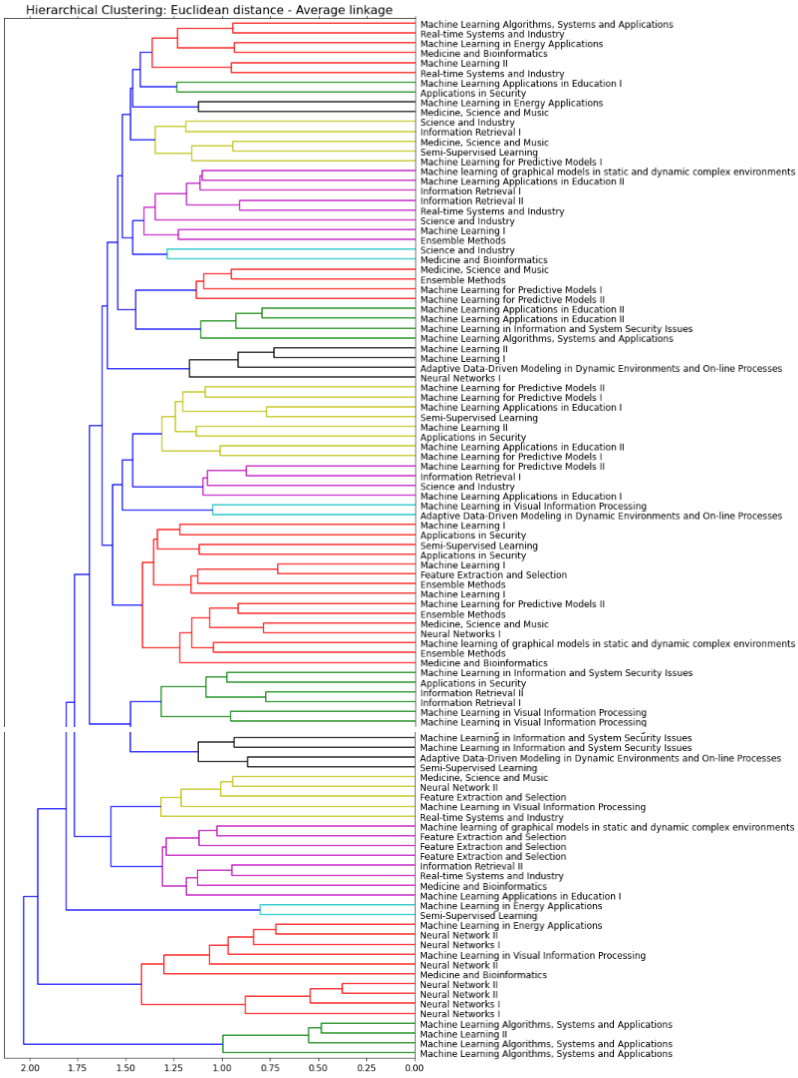


Figure 5.2.2m. Dendogram: Hierarchical Euclidean-average-link clustering

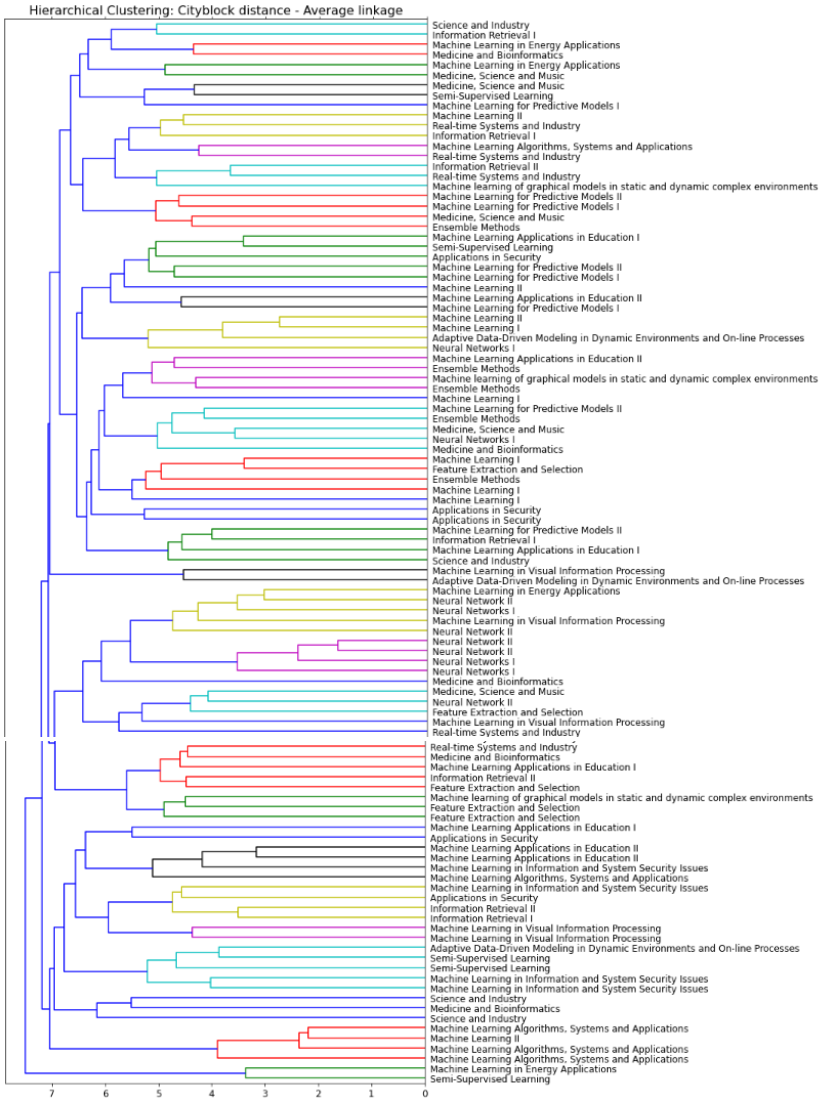


Figure 5.2.2n. Dendogram: Hierarchical Cityblock-average-link clustering

*Table 5.2.2c. Hierarchical clustering: labelling table (*Cosine-average*)*

	label	session		label	session
22	1	Neural Networks I	**17**	10	Machine Learning I
38	1	Medicine and Bioinformatics	**1**	10	Ensemble Methods
99	1	Machine Learning in Visual Information Processing	**47**	11	Information Retrieval I
87	1	Machine Learning in Energy Applications	**9**	11	Applications in Security
29	1	Neural Network II	**98**	11	Machine Learning in Visual Information Processing
28	1	Neural Network II	**81**	11	Machine Learning in Information and System Sec...
27	1	Neural Network II	**101**	11	Machine Learning in Visual Information Processing
25	1	Neural Network II	**49**	11	Information Retrieval II
24	1	Neural Networks I	**82**	12	Machine Learning in Information and System Sec...
23	1	Neural Networks I	**95**	12	Adaptive Data-Driven Modeling in Dynamic Envir...
20	1	Neural Networks I	**33**	12	Semi-Supervised Learning
50	2	Information Retrieval II	**83**	12	Machine Learning in Information and System Sec...
37	2	Medicine and Bioinformatics	**31**	12	Semi-Supervised Learning
43	2	Real-time Systems and Industry	**6**	12	Applications in Security
104	2	Machine learning of graphical models in static...	**58**	13	Machine Learning II
11	2	Feature Extraction and Selection	**18**	13	Machine Learning I
75	2	Machine Learning Applications in Education I	**96**	13	Adaptive Data-Driven Modeling in Dynamic Envir...
13	2	Feature Extraction and Selection	**94**	14	Adaptive Data-Driven Modeling in Dynamic Envir...
12	2	Feature Extraction and Selection	**97**	14	Machine Learning in Visual Information Processing
63	3	Medicine, Science and Music	**46**	15	Information Retrieval I
42	3	Real-time Systems and Industry	**88**	16	Machine Learning in Energy Applications
10	3	Feature Extraction and Selection	**34**	16	Semi-Supervised Learning
100	3	Machine Learning in Visual Information Processing	**54**	16	Science and Industry
26	3	Neural Network II	**55**	16	Science and Industry
77	4	Machine Learning Applications in Education II	**85**	17	Machine Learning in Energy Applications
65	4	Machine Learning for Predictive Models I	**62**	17	Medicine, Science and Music
53	5	Science and Industry	**35**	18	Medicine and Bioinformatics
74	5	Machine Learning Applications in Education I	**52**	18	Science and Industry
66	5	Machine Learning for Predictive Models I	**15**	18	Machine Learning I
71	5	Machine Learning for Predictive Models II	**2**	19	Ensemble Methods
72	5	Machine Learning for Predictive Models II	**64**	19	Medicine, Science and Music
48	5	Information Retrieval I	**69**	19	Machine Learning for Predictive Models II
57	6	Machine Learning II	**68**	19	Machine Learning for Predictive Models I
76	6	Machine Learning Applications in Education I	**79**	20	Machine Learning Applications in Education II
8	6	Applications in Security	**84**	20	Machine Learning in Information and System Sec...
32	6	Semi-Supervised Learning	**80**	20	Machine Learning Applications in Education II
4	7	Ensemble Methods	**93**	20	Machine Learning Algorithms, Systems and Appli...
78	7	Machine Learning Applications in Education II	**73**	21	Machine Learning Applications in Education I
103	8	Machine learning of graphical models in static...	**5**	21	Applications in Security
102	8	Machine learning of graphical models in static...	**67**	22	Machine Learning for Predictive Models I
41	8	Real-time Systems and Industry	**56**	22	Machine Learning II
51	8	Information Retrieval II	**30**	22	Semi-Supervised Learning

45	8	Information Retrieval I	**90**	22	Machine Learning Algorithms, Systems and Appli...
61	9	Medicine, Science and Music	**89**	22	Machine Learning Algorithms, Systems and Appli...
70	9	Machine Learning for Predictive Models II	**60**	22	Medicine, Science and Music
0	9	Ensemble Methods	**92**	22	Machine Learning Algorithms, Systems and Appli...
21	9	Neural Networks I	**44**	23	Real-time Systems and Industry
36	9	Medicine and Bioinformatics	**59**	23	Machine Learning II
3	9	Ensemble Methods	**91**	24	Machine Learning Algorithms, Systems and Appli...
14	10	Feature Extraction and Selection	**86**	24	Machine Learning in Energy Applications
16	10	Machine Learning I	**39**	24	Medicine and Bioinformatics
7	10	Applications in Security	**40**	24	Real-time Systems and Industry
19	10	Machine Learning I			

5.2.3. DBSCAN Clustering

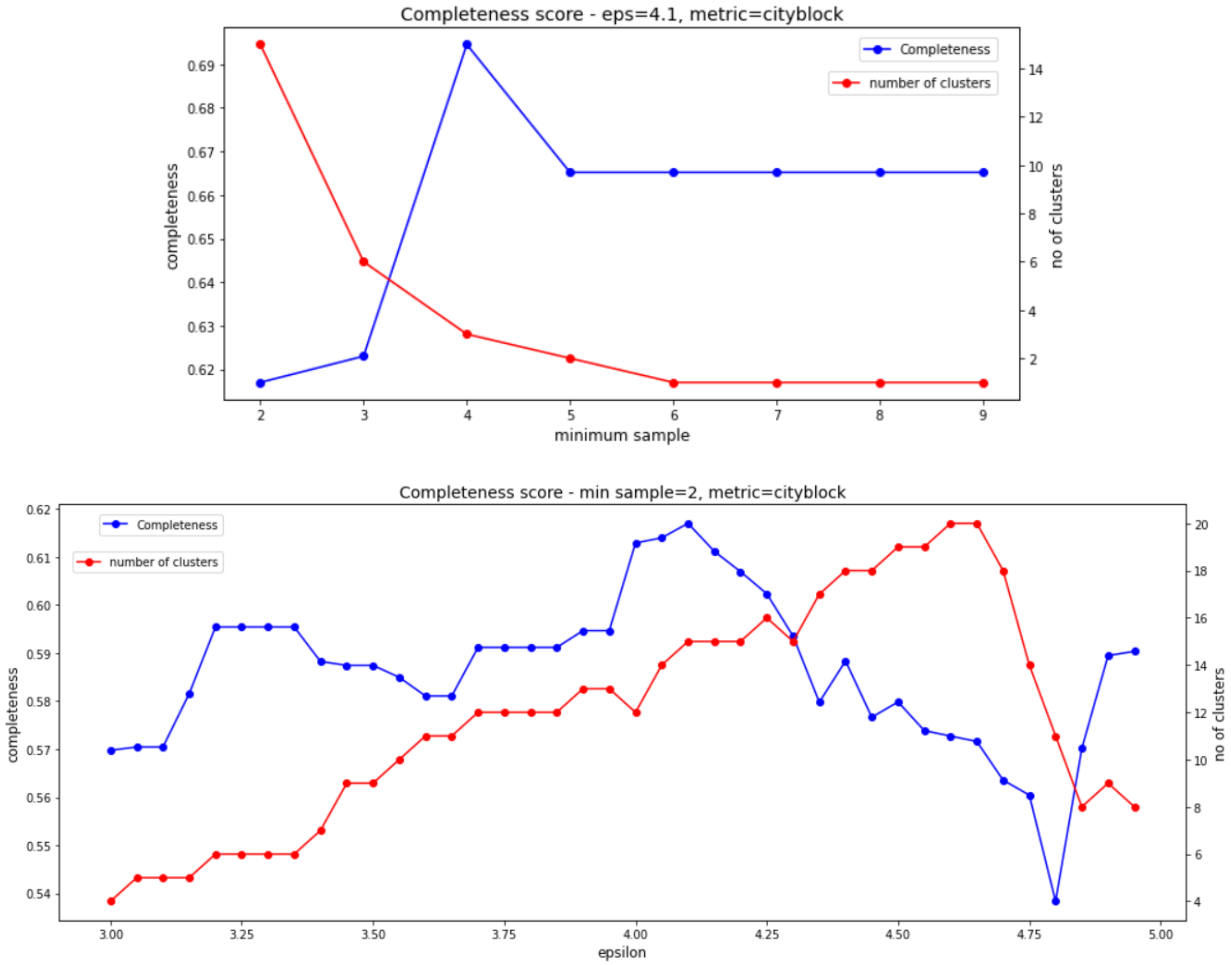


Figure 5.2.3a. Completeness scores: DBSCAN Cityblock models

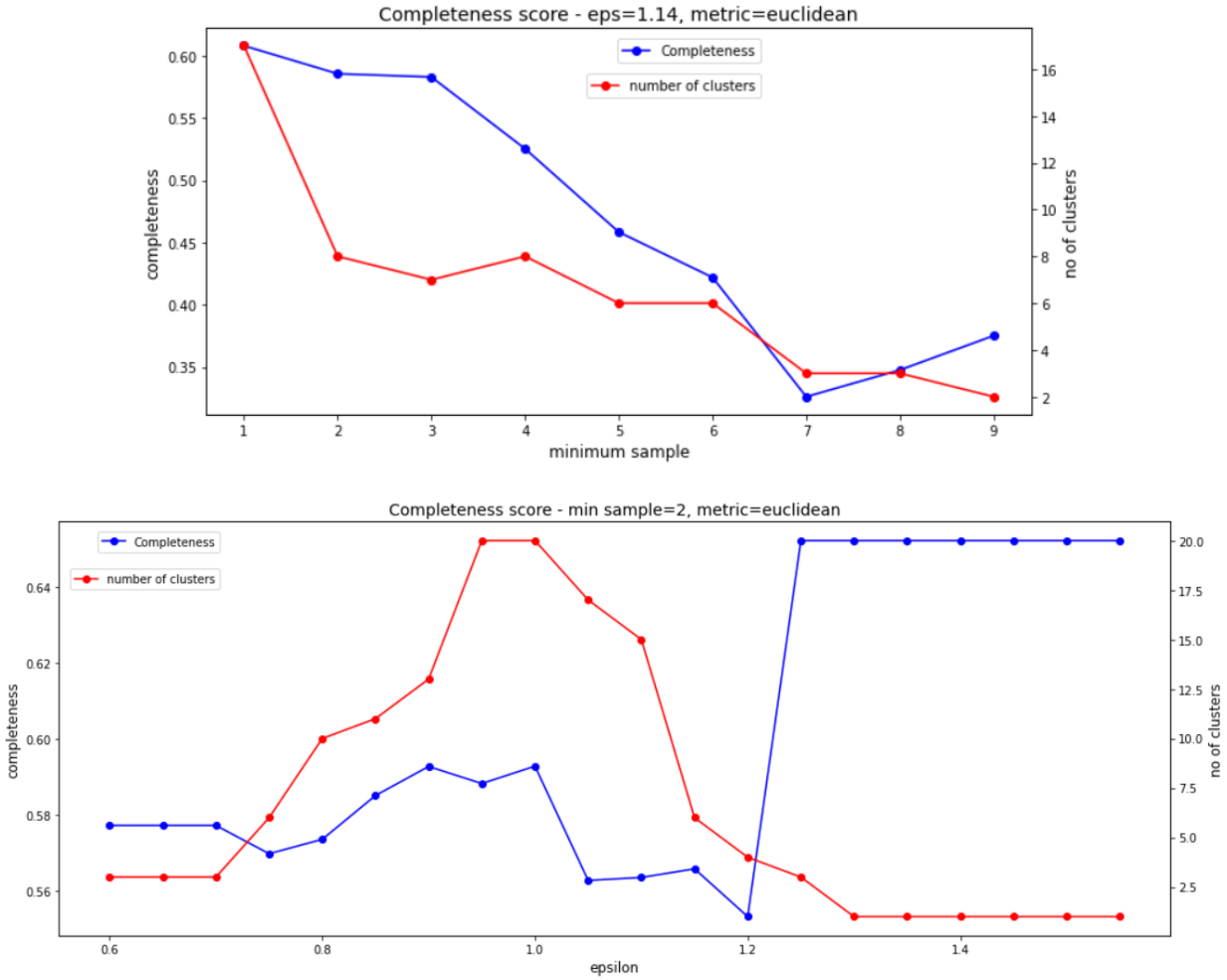


Figure 5.2.3b. Completeness scores: DBSCAN Euclidean models

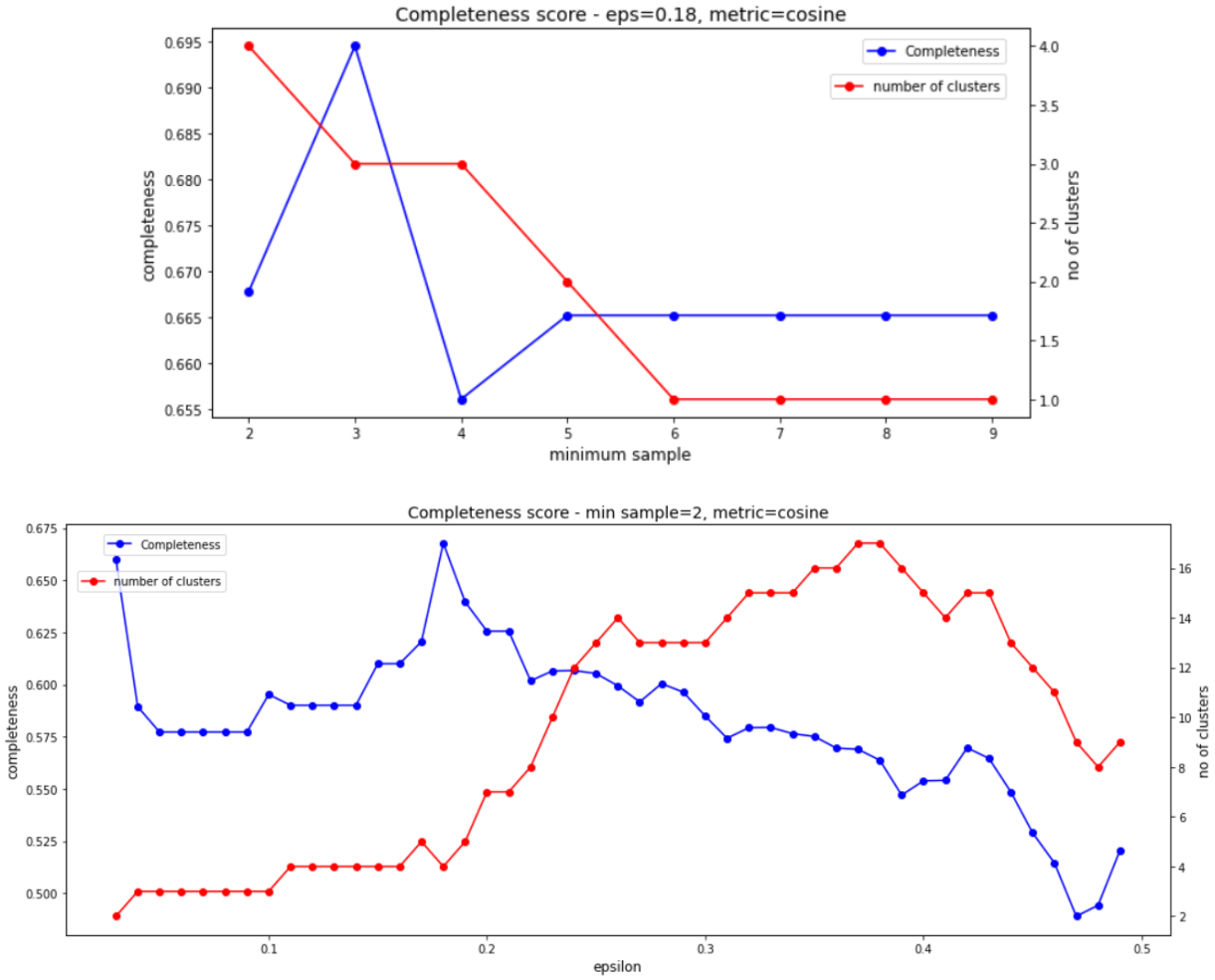


Figure 5.2.3c. Completeness scores: DBSCAN Cosine models

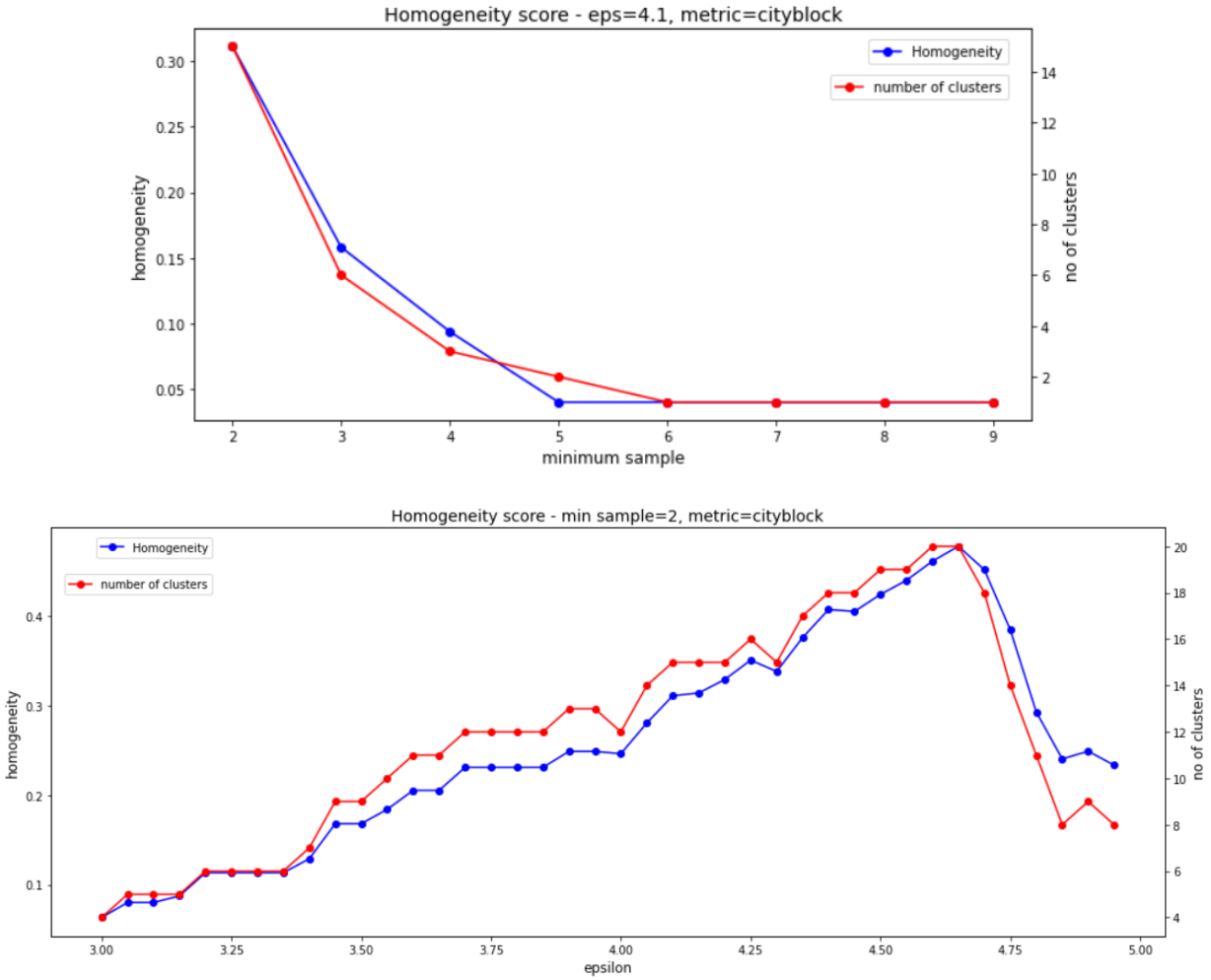


Figure 5.2.3d. Homogeneity scores: DBSCAN Cityblock models

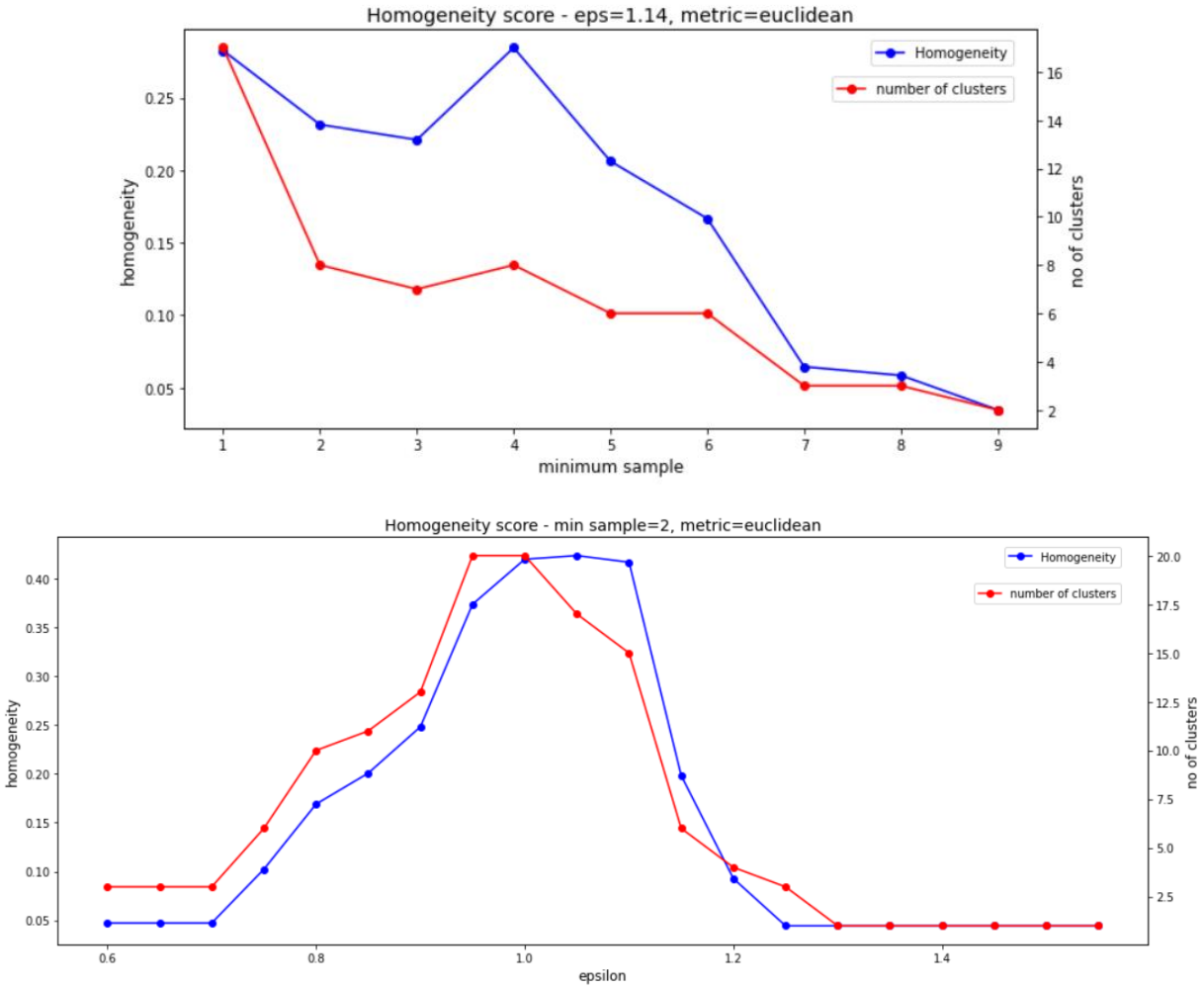


Figure 5.2.3e. Homogeneity scores: DBSCAN Euclidean models

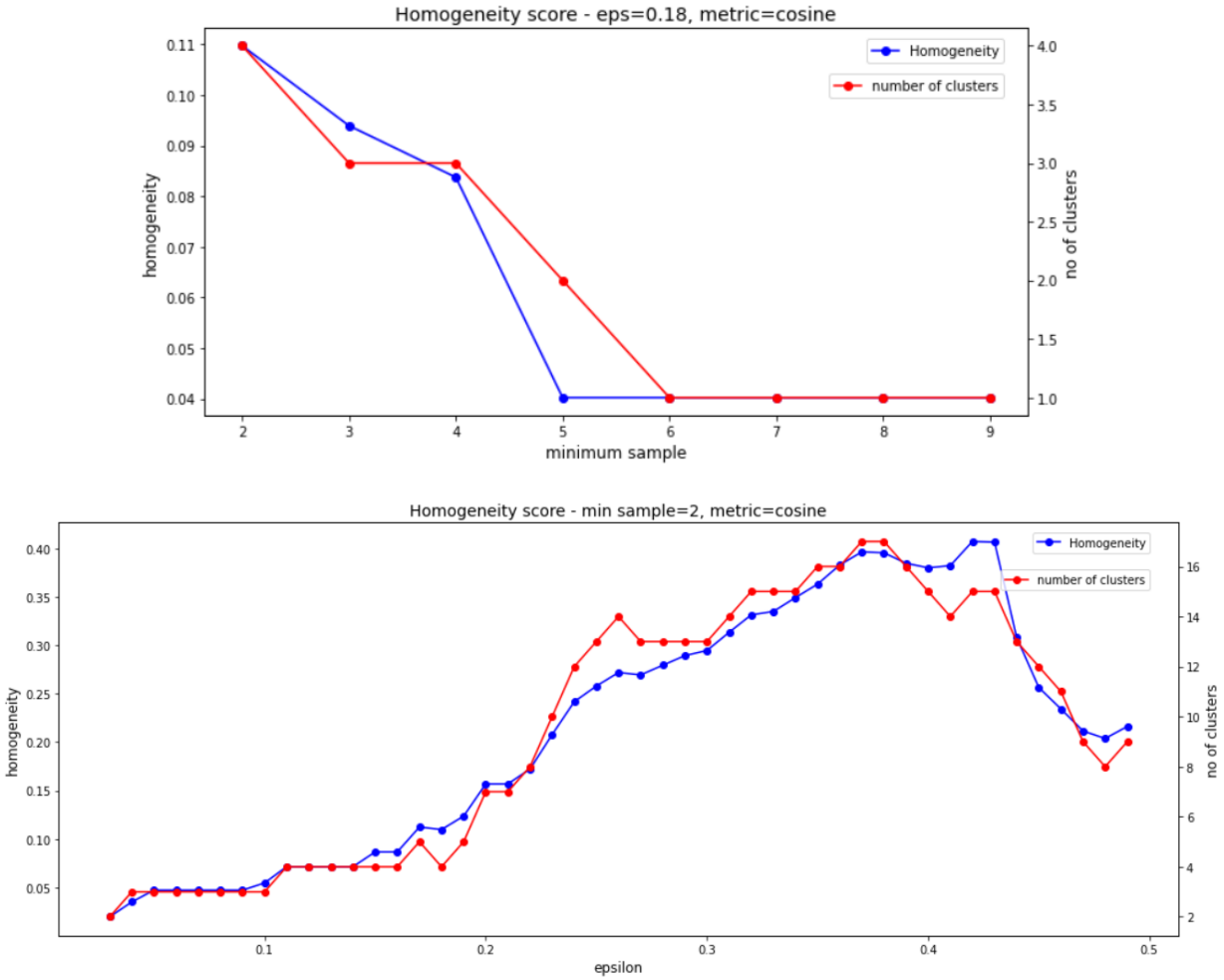


Figure 5.2.3f. Homogeneity scores: DBSCAN Cosine models

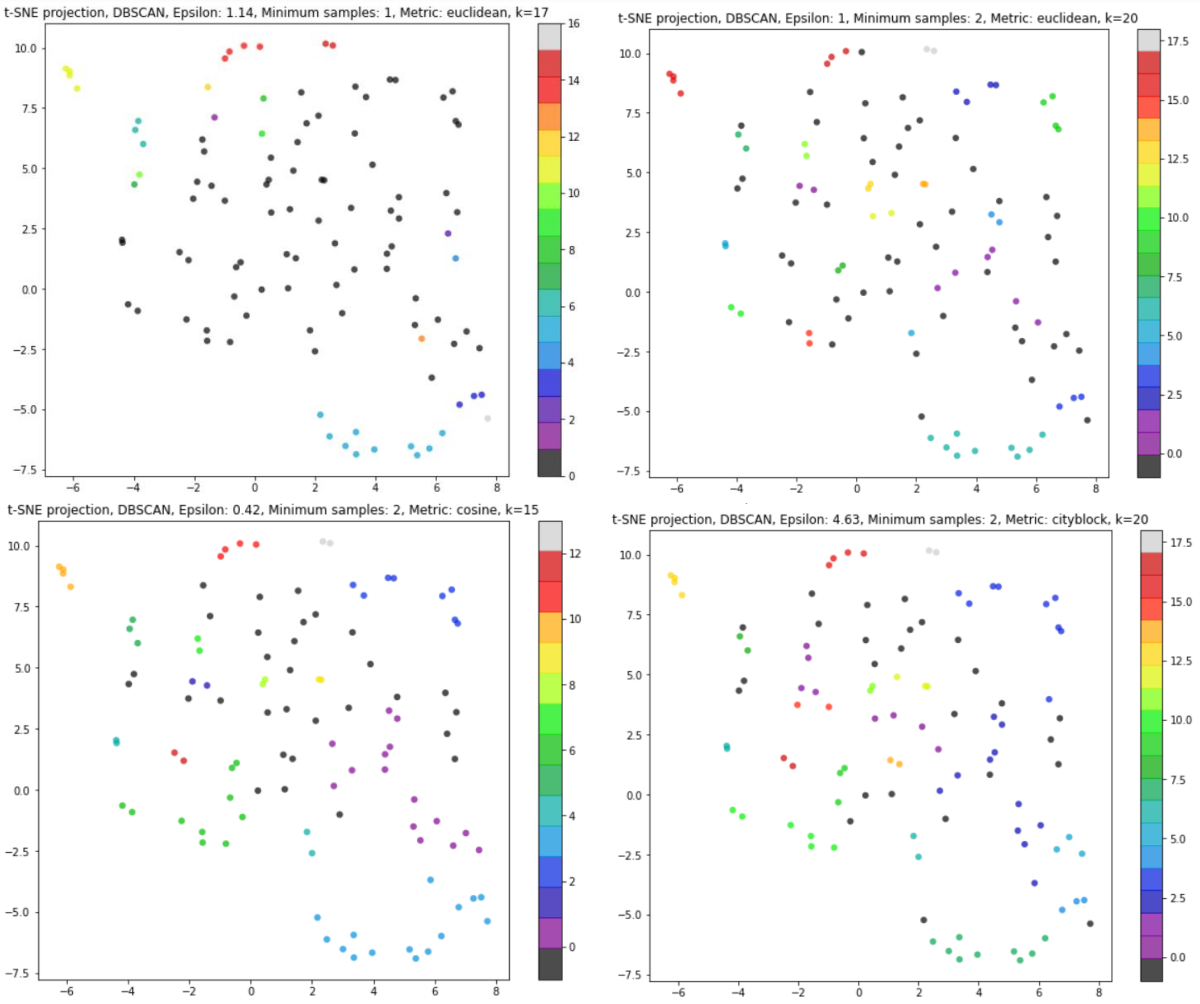


Figure 5.2.3g. t-SNE projections: DBSCAN Euclidean (top), Cosine and Cityblock (bottom)

Table 5.2.3a. DBSCAN: labelling table

	label	session		label	session
52	-1	Science and Industry	**81**	3	Machine Learning in Information and System Sec...
38	-1	Medicine and Bioinformatics	**95**	3	Adaptive Data-Driven Modeling in Dynamic Envir...
42	-1	Real-time Systems and Industry	**26**	4	Neural Network II
46	-1	Information Retrieval I	**63**	4	Medicine, Science and Music
103	-1	Machine learning of graphical models in static...	**10**	4	Feature Extraction and Selection
54	-1	Science and Industry	**13**	5	Feature Extraction and Selection
55	-1	Science and Industry	**11**	5	Feature Extraction and Selection
57	-1	Machine Learning II	**104**	5	Machine learning of graphical models in static...
19	-1	Machine Learning I	**18**	6	Machine Learning I
62	-1	Medicine, Science and Music	**23**	6	Neural Networks I
17	-1	Machine Learning I	**96**	6	Adaptive Data-Driven Modeling in Dynamic Envir...
67	-1	Machine Learning for Predictive Models I	**58**	6	Machine Learning II
36	-1	Medicine and Bioinformatics	**20**	7	Neural Networks I
72	-1	Machine Learning for Predictive Models II	**22**	7	Neural Networks I
15	-1	Machine Learning I	**24**	7	Neural Networks I
78	-1	Machine Learning Applications in Education II	**99**	7	Machine Learning in Visual Information Processing
85	-1	Machine Learning in Energy Applications	**25**	7	Neural Network II
8	-1	Applications in Security	**27**	7	Neural Network II
7	-1	Applications in Security	**28**	7	Neural Network II
6	-1	Applications in Security	**29**	7	Neural Network II
5	-1	Applications in Security	**87**	7	Machine Learning in Energy Applications
4	-1	Ensemble Methods	**60**	8	Medicine, Science and Music
100	-1	Machine Learning in Visual Information Processing	**30**	8	Semi-Supervised Learning
1	-1	Ensemble Methods	**76**	9	Machine Learning Applications in Education I
73	-1	Machine Learning Applications in Education I	**32**	9	Semi-Supervised Learning
35	-1	Medicine and Bioinformatics	**66**	9	Machine Learning for Predictive Models I
86	0	Machine Learning in Energy Applications	**71**	10	Machine Learning for Predictive Models II
91	0	Machine Learning Algorithms, Systems and Appli...	**34**	10	Semi-Supervised Learning
102	0	Machine learning of graphical models in static...	**48**	10	Information Retrieval I
40	0	Real-time Systems and Industry	**53**	10	Science and Industry
39	0	Medicine and Bioinformatics	**88**	10	Machine Learning in Energy Applications
0	0	Ensemble Methods	**74**	10	Machine Learning Applications in Education I
2	1	Ensemble Methods	**41**	11	Real-time Systems and Industry
64	1	Medicine, Science and Music	**51**	11	Information Retrieval II
75	2	Machine Learning Applications in Education I	**44**	12	Real-time Systems and Industry
21	2	Neural Networks I	**45**	12	Information Retrieval I
50	2	Information Retrieval II	**59**	12	Machine Learning II
61	2	Medicine, Science and Music	**90**	13	Machine Learning Algorithms, Systems and Appli...
43	2	Real-time Systems and Industry	**56**	13	Machine Learning II
3	2	Ensemble Methods	**89**	13	Machine Learning Algorithms, Systems and Appli...
16	2	Machine Learning I	**92**	13	Machine Learning Algorithms, Systems and Appli...
70	2	Machine Learning for Predictive Models II	**77**	14	Machine Learning Applications in Education II

12	2	Feature Extraction and Selection	**65**	14	Machine Learning for Predictive Models I
37	2	Medicine and Bioinformatics	**69**	15	Machine Learning for Predictive Models II
14	2	Feature Extraction and Selection	**68**	15	Machine Learning for Predictive Models I
33	3	Semi-Supervised Learning	**84**	16	Machine Learning in Information and System Sec...
83	3	Machine Learning in Information and System Sec...	**80**	16	Machine Learning Applications in Education II
9	3	Applications in Security	**79**	16	Machine Learning Applications in Education II
49	3	Information Retrieval II	**93**	16	Machine Learning Algorithms, Systems and Appli...
47	3	Information Retrieval I	**94**	17	Adaptive Data-Driven Modeling in Dynamic Envir...
82	3	Machine Learning in Information and System Sec...	**97**	17	Machine Learning in Visual Information Processing
31	3	Semi-Supervised Learning	**98**	18	Machine Learning in Visual Information Processing
			101	18	Machine Learning in Visual Information Processing

6. References

1. SciKit Learn documentation: MinMaxScaler, viewed on 21 May 2020, https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.MinMaxScaler.html
2. Caren Han, Lab09, COMP5318: Machine Learning and Data Mining, 27 April 2020
3. SciKit Learn documentation: Principal component analysis (PCA), viewed on 21 May 2020, https://scikit-learn.org/stable/modules/generated/sklearn.decomposition.PCA.html?highlight=pca#sklearn.decomposition.PCA
4. SciKit Learn documentation: LabelEncoder, viewed on 21 May 2020, https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.LabelEncoder.html?highlight=labelencoder#sklearn.preprocessing.LabelEncoder
5. SciKit Learn documentation: KMeans, viewed on 21 May 2020, https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html?highlight=kmeans#sklearn.cluster.KMeans
6. Han, Jiawei, et al. Data Mining: Concepts and Techniques, Elsevier Science & Technology, 2011, ProQuest E-book Central, downloaded 4 April 2020
7. Elbow method (clustering), Wikipedia, viewed on 21 May 2020, https://en.wikipedia.org/wiki/Elbow_method_(clustering)
8. SciKit Learn documentation: Davies-Bouldin score, viewed on 21 May 2020, https://scikit-learn.org/stable/modules/generated/sklearn.metrics.davies_bouldin_score.html
9. SciKit Learn documentation: Calinski and Harabasz score, viewed on 21 May 2020, https://scikit-learn.org/stable/modules/generated/sklearn.metrics.calinski_harabasz_score.html
10. Caren Han, Lecture 9, COMP5318: Machine Learning and Data Mining, 27 April 2020
11. Caren Han, Lab11, COMP5318: Machine Learning and Data Mining, 11 May 2020
12. SciKit Learn documentation: Completeness Score, viewed on 21 May 2020, https://scikit-learn.org/stable/modules/generated/sklearn.metrics.completeness_score.html?highlight=completeness#sklearn.metrics.completeness_score
13. SciKit Learn documentation: Homogeneity Score, viewed on 21 May 2020, https://scikit-learn.org/stable/modules/generated/sklearn.metrics.homogeneity_score.html?highlight=homogeneity#sklearn.metrics.homogeneity_score
14. SciKit Learn documentation: Agglomerative Clustering, viewed on 21 May 2020, https://scikit-learn.org/stable/modules/generated/sklearn.cluster.AgglomerativeClustering.html?highlight=agglomerative#sklearn.cluster.AgglomerativeClustering
15. SciPy documentation: Hierarchical clustering, viewed on 21 May 2020, https://docs.scipy.org/doc/scipy/reference/cluster.hierarchy.html
16. Tan, Steinbach, Karpatne, Kumar 2020, Introduction to Data Mining (Global Edition), Pearson NY
17. SciPy documentation: Dendrogram, viewed on 21 May 2020, https://docs.scipy.org/doc/scipy/reference/generated/scipy.cluster.hierarchy.dendrogram.html#scipy.cluster.hierarchy.dendrogram
18. SciKit Learn documentation: DBSCAN, viewed on 21 May 2020, https://scikit-learn.org/stable/modules/generated/sklearn.cluster.DBSCAN.html
19. Caren Han, Lecture 10, COMP5318: Machine Learning and Data Mining, 4 May 2020
20. Irena Koprinska, Lecture 2, COMP5318: Machine Learning and Data Mining, 2 March 2020