UNIVERSITAT DE BARCELONA

Facultat de Matemàtiques
i Informàtica

Introduction to Machine Learning

Dimensionality Reduction

Eva Veli
Andras Kasa
Niklas Long Schiefelbein

Contents

1 Introduction

Dimensionality reduction is a technique used to handle high-dimensional data by mapping it onto a lower-dimensional space through linear or non-linear transformations. It reduces the number of features in the dataset while retaining most of the variability and essential patterns within the data [3].

This work involves implementing Principal Component Analysis (PCA) from scratch using Python 3.9 and the PyCharm IDE [6]. We analyze its performance on various datasets and compare the results to library-based solutions. Additionally, clustering methods such as the K-Means++ and OPTICS are integrated to evaluate how dimensionality reduction impacts clustering outcomes [1]. Three-dimensional visualizations provide insights into the transformations, helping to uncover patterns that are otherwise hidden in high-dimensional spaces. To visualize data in a low-dimensional space (3D), both PCA and Uniform Manifold Approximation and Projection (UMAP) are employed [4].

For the analysis, we utilize datasets from the UCI Machine Learning Repository [2], specifically the Contraceptive Method Choice (CMC) and Pen-Based Recognition of Handwritten Digits (Pen-Based) datasets.

2 Principal Component Analysis (PCA)

Principal Component Analysis (PCA) [3] is a widely used dimensionality reduction technique in data analysis and machine learning. It transforms high-dimensional data into a lower-dimensional space while preserving as much variability as possible. By identifying the directions (principal components) that maximize the variance in the data, PCA simplifies the complexity of data without significantly losing important information. Below you will find the pseudocode of PCA Algorithm.

```
Input: Dataset X ∈ ℝ^{n×d}, number of principal components k
Output: Transformed dataset X_pca ∈ ℝ^{n×k}

1. Standardize X to have zero mean
2. Compute covariance matrix C = 1/(n−1) X^T X
3. Perform eigen decomposition on C to obtain eigenvectors
4. Select top k eigenvectors V_k
5. Transform data: X_pca = X V_k
6. Return X_pca
```

2.1 Implementation of PCA

In our implementation of PCA, we calculate the mean of each feature and center the data by subtracting these means, ensuring that the dataset has a mean of zero for each feature. This centering is crucial as it aligns the data around the origin, allowing PCA to effectively identify the directions of maximum variance. Following this, we compute the covariance matrix, which quantifies the variance and covariance between different features, providing a foundation for understanding the data's structure. To extract the principal components, we perform eigen decomposition on the covariance matrix, obtaining eigenvalues and eigenvectors that represent the magnitude of variance and the directions of the principal components, respectively. These eigenvalues and eigenvectors are then sorted in descending order based on the eigenvalues to prioritize components that capture the most significant variance. The number of principal components is determined dynamically by either selecting a predefined threshold for the cumulative explained variance ratio or by specifying the exact number of components desired. By selecting the top eigenvectors corresponding to the largest eigenvalues, we project the original centered data into a reduced-dimensional space, thereby retaining the essential patterns and variability while mitigating the complexity of high-dimensional data. Additionally, our implementation ensures numerical stability by converting any complex numbers resulting from the eigen decomposition to their real counterparts.

As the last step in our PCA implementation we reconstruct the original data by calculating the dot product of the transpose of the eigenvector matrix with the before transformed data. With this, we transform the principal component scores back into the original feature space. Finally, we restore the data to its original scale by reintroducing the mean values that were subtracted during the initial centering step. The reconstructed data is then plotted and saved to a file called `reconstructed_data_plot.png`.

2.2 Effects of PCA in our datasets

cmc Figure 1 compares the representation of the CMC dataset using the three most descriptive features (left) and the first three principal components obtained through PCA (right).

On the left, the scatter plot visualizes the dataset based on the three features that provide the highest discriminative power. The points are color-coded to represent the three classes (Class 1, Class 2, and

Class 3). While the three features provide some class separation, the clustering structure remains fairly constrained, suggesting the need for additional dimensions to fully capture the data's variability.

On the right, the dataset is visualized in the space of the first three principal components, derived from PCA. These components are linear combinations of the original features and capture the maximum variance in the data. The PCA-transformed plot shows a more spread-out distribution of points, with better class overlap visualization, as PCA reduces redundancy among features and projects the data into an optimal low-dimensional space.This comparison highlights how PCA can effectively summarize high-dimensional data while preserving essential structure, making it suitable for clustering and visualization tasks.

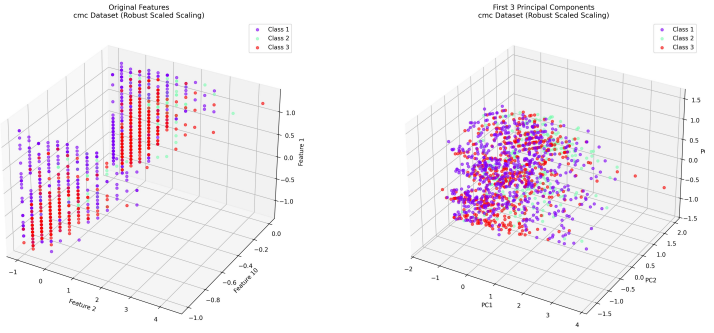

Figure 1: Original Features vs PCA for cmc dataset

Figure 2 illustrates the explained variance analysis for PCA on robustly scaled data, with the cumulative variance ratio showing a steep increase for the first few components before flattening, and the individual variance contributions highlighting that the first two components capture most of the variance. This analysis helps to select the optimal number of components for dimensionality reduction. As summarized in Figure 3, achieving 80% variance retention requires 9 components, reducing dimensionality by 57.1%. Higher thresholds, such as 90% and 95%, require 11 and 13 components, with reductions of 47.6% and 38.1%, respectively. For 99% retention, 16 components are needed, reducing dimensionality by 23.8%. These results demonstrate the trade-off between variance retention and dimensionality reduction.

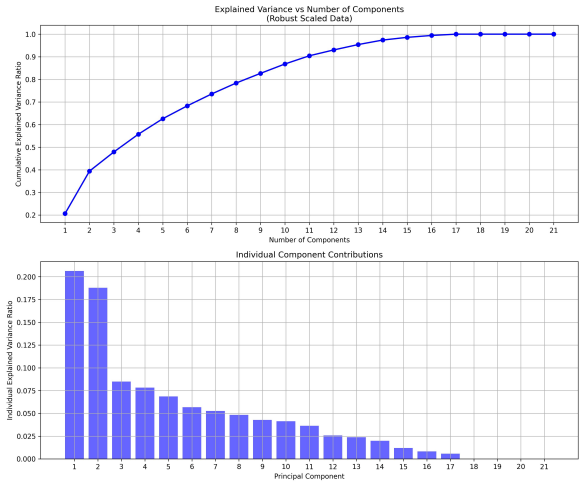

Figure 2: Explained Variance vs Number of Components for cmc dataset

Threshold	Components	Reduction (%)
0.80	9	57.1
0.85	10	52.4
0.90	11	47.6
0.95	13	38.1
0.99	16	23.8

Figure 3: PCA Variance Analysis for Robust Scaled cmc Dataset

pen-based Figure 4 shows again that only slicing the three most discriminative features results in a complex and densely packed structure, making it challenging to identify clear boundaries or patterns. Despite scaling, the slice of the high-dimensional feature space still results in overlapping distributions, with classes appearing interwoven rather than neatly separated.

The PCA-transformed plot reveals more distinguishable structures and patterns, with certain classes forming tighter clusters and less visual overlap. By reducing the data to its three most variant components and therefore focusing on the most informative directions, PCA makes the underlying class structure easier to understand. Subsequently the interpretation and understanding of the clusters is facilitated.

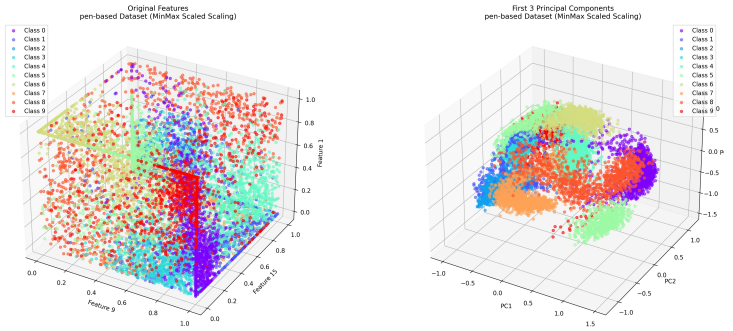

Figure 4: Orginal Features vs PCA for pen-based dataset

Figures 5 and 2 show again the explained variance analysis after applying PCA, but for pen-based dataset.

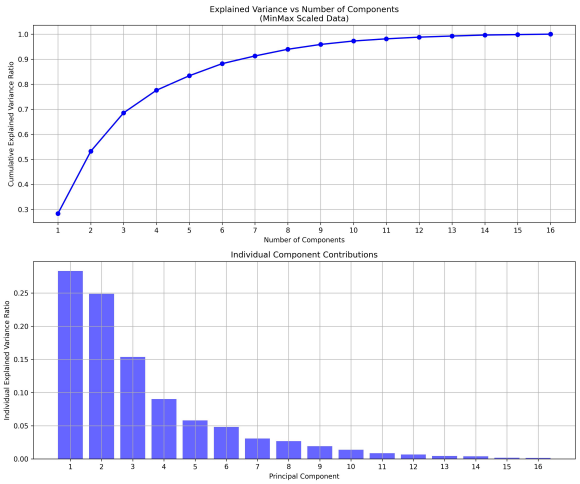

Figure 5: Explained Variance vs Number of Components for pen-based dataset

Threshold	Components	Reduction (%)
0.80	5	68.8
0.85	6	62.5
0.90	7	56.2
0.95	9	43.8
0.99	13	18.8

Figure 6: PCA Variance Analysis for MinMax Scaled pen-based Dataset

2.3 Validity of our PCA Implementation

In this section, we compare the results of our custom PCA implementation with two library-based implementations: `sklearn.decomposition.PCA` and `sklearn.decomposition.IncrementalPCA` from the scikit-learn library [5]. The `PCA` implementation in scikit-learn uses eigen decomposition and is designed for datasets that can fit entirely in memory. On the other hand, `IncrementalPCA` processes data in mini-batches, making it well-suited for handling large-scale datasets that exceed memory limitations. Our custom PCA implementation also relies on eigen decomposition, similar to `PCA`, and serves as a baseline for comparison.

Our PCA	sklearn PCA	sklearn IPCA	Diff 1-2	Diff 2-3
-0.0797	-0.0797	-0.0846	0	0.0049
-1.1753	-1.1753	-1.1913	0	0.0159
0.4807	0.4807	0.9321	0	0.4514
2.5974	2.5974	2.6092	0	0.0118
-0.1093	-0.1093	-0.0920	0	0.0173
-0.0679	-0.0679	-0.7347	0	0.6668
1.6672	1.6672	1.6622	0	0.0050
-0.3867	-0.3867	-0.3976	0	0.0109
0.1431	0.1431	-0.8173	0	0.9604
2.1252	2.1252	2.1488	0	0.0235

Our PCA	sklearn PCA	sklearn IPCA	Diff 1-2	Diff 2-3
0.6131	0.6131	0.6162	0	0.0031
0.6183	0.6183	0.6297	0	0.0114
-1.3193	-1.3193	1.3863	0	2.7056
0.5693	0.5693	0.5779	0	0.0085
0.8040	0.8040	0.8150	0	0.0110
-1.1101	-1.1101	1.1263	0	2.2364
0.8805	0.8805	0.8896	0	0.0091
0.7515	0.7515	0.7526	0	0.0011
-0.9661	-0.9661	1.0012	0	1.9673
1.0125	1.0125	1.0246	0	0.0121

Table 1: Comparing first 10 flattened data points of our own PCA implementation, sklearn's PCA, and sklearn's IPCA for *cmc* (left) using RobustScaler and $pen-based$ (right) using MinMaxScaler

Table 1 shows that our implementation of PCA does not differ from the `sklearn.decomposition.PCA`. For this we applied both of the methods on the same datasets and calculated the difference of the obtained projected datapoints that have been flattened before. This shows how consistent the coordinate values are for each data sample across the different PCA implementations.

When comparing our PCA implementation with sklearn's `PCA`, we find that the displayed values are generally consistent. However, it's important to note that sometimes our implementation produces values

with opposite signs. Specifically, some of our results are multiplied by -1 compared to sklearn's values. This sign difference does not affect the validity of the PCA, as the principal components are unique only up to their sign.The last column illustrates this behavior by showing differences that are approximately twice the absolute values in our PCA implementation. Additionally, it highlights that the iterative process of IPCA produces slightly different results. The continuous batch updates of the algorithm create an overall similar trend in the data but may lead to marginally less accurate components due to its iterative nature and reliance on batch-wise processing. We chose to compare these results using a table with numerical data instead of a visualization because visualizations make it challenging to observe detailed comparisons and exact differences between data points.

3 Clustering

Cluster analysis divides data into groups to understand and analyze its structure based only on the descriptions of the observations and their relationships, so that instances within the same group are similar to one another and different from those in other groups. In this section, we investigate the impact of performing or not performing dimensionality reduction prior to clustering using K-Means++ and OPTICS clustering. To better understand the results and the clustering structure, we visualize the original datasets and the clustering outcomes in 3D spaces.

3.1 PCA: OPTICS

Since the previous work involved conducting a grid search to determine the optimal parameters and the best number of clusters for OPTICS, we extended the same methodology here. An additional grid search was performed, incorporating the different values of explained variance thresholds from Figure 3 and Figure 6. The results of this process are summarized in Table 2 and Table 3. These tables present the best clustering configurations achieved for the reduced datasets. Based on these results, we will analyze and select the optimal value of k to ensure the most effective clustering.

cmc For cmc dataset, we chose $\mathbf{k = 3}$ as the most appropriate number of clusters because it provided a strong balance between internal consistency and external quality measures. While the silhouette score and some external indices like ARI and FMI were slightly lower than with k = 2, the Calinski-Harabasz (CH) index for k = 3 was significantly higher, indicating a much more pronounced cluster structure with tighter, more distinct groupings. In addition, the purity measure increased when moving from two to three clusters, suggesting that the discovered partition better separated the data into meaningful subsets. Thus, despite a trade-off in certain metrics, the combination of a significantly improved CH index, a higher purity value, and stable clustering parameters led us to determine that $k = 3$ yielded the most coherent and interpretable clustering solution. Choosing $k = 3$ retains 90% of the variance, and the number of components is reduced from 21 to 11, as shown in Figure 3.

k	threshold	min_samples	max_eps	metric	algorithm	sil	db	ch	ari	fmi	purity
2	0.95	45	1.500	cosine	auto	0.524	0.728	152.366	0.014	0.428	0.462
3	0.90	35	1.500	euclidean	auto	0.474	0.814	264.875	0.046	0.368	0.474
4	0.99	25	1.500	euclidean	auto	0.503	0.735	177.446	0.021	0.309	0.431
5	0.90	35	3.000	manhattan	auto	0.365	1.277	193.597	0.044	0.306	0.502
6	0.80	25	1.500	cosine	auto	0.450	0.806	156.102	0.007	0.246	0.415
7	0.90	25	1.500	euclidean	auto	0.359	1.174	187.956	0.019	0.295	0.469

Table 2: Best cmc OPTICS Clustering Results per k

pen-based For the pen-based dataset, while k = 9 and k = 10 showed exceptionally high external validity yielding nearly perfect purity scores and very strong Adjusted Rand Index (ARI) and Fowlkes-Mallows Index (FMI) values $\mathbf{k = 11}$ stood out with the highest Calinski-Harabasz (CH) index (1380.005) and the best silhouette score (0.713). These strong internal metrics indicate a more distinct and robust cluster structure, suggesting that the clusters formed at k = 11 are both cohesive and well-separated. Although there is a slight decrease in external measures compared to k = 9 and k = 10, the improvement in internal measures at k = 11 implies a more interpretable and reliable clustering solution overall. Therefore, despite a trade-off in some metrics, the combination of superior internal coherence and near-optimal external validity supports choosing k = 11 as the best number of clusters for the pen-based dataset. Choosing $k = 11$ retains 85% of the variance, and the number of components is reduced to 6, as shown in Figure 6.

k	threshold	min_samples	max_eps	metric	algorithm	sil	db	ch	ari	fmi	purity
8	0.85	15	1.000	euclidean	auto	0.707	0.442	619.521	0.798	0.845	0.995
9	0.80	15	1.000	euclidean	auto	0.690	0.335	763.911	0.977	0.991	0.996
10	0.80	15	1.000	euclidean	brute	0.689	0.345	746.043	0.976	0.990	0.996
11	0.85	23	1.000	cosine	auto	0.713	0.461	1380.005	0.911	0.943	0.995
12	0.80	23	1.000	cosine	auto	0.461	0.520	792.106	0.926	0.953	0.986
14	0.99	15	1.000	manhattan	auto	0.249	0.936	1273.676	0.485	0.596	0.646

Table 3: Best pen-based OPTICS Clustering Results per k

3.2 PCA: K-Means++

cmc On evaluation of clustering metrics on the CMC dataset, we identified $k = 2$ as the optimal number of clusters for K-Means++ analysis. This choice is supported by the highest Silhouette Score (0.306) and Calinski-Harabasz Index (319.337), indicating well-defined, compact clusters with strong separation. Additionally, $k = 2$ achieved the highest Adjusted Rand Index (0.033) and Fowlkes-Mallows Index (0.440), reflecting better alignment with potential ground truth labels. While Purity improves slightly with higher k, it comes at the expense of the Silhouette Score and Calinski-Harabasz Index. The Davies-Bouldin Index also provides no significant advantage for higher k. Choosing $k = 2$ retains 80% of the variance and reduces the number of components from 21 to 9, as shown in Figure 3.

k	threshold	sil	db	ch	ari	fmi	purity
2	0.80	0.306	2.079	319.337	0.033	0.440	0.427
3	0.80	0.300	1.799	319.236	0.028	0.365	0.443
4	0.80	0.303	1.816	274.908	0.020	0.316	0.440
5	0.80	0.300	1.805	247.868	0.012	0.282	0.443
6	0.80	0.293	1.795	228.570	0.014	0.259	0.449
7	0.80	0.267	1.793	209.527	0.018	0.244	0.458

Table 4: Best cmc K-Means++ Clustering Results per k

pen-based While evaluating various values of k, we observed that increasing k enhances external metrics. For instance, $k = 12$ achieves the highest Adjusted Rand Index, Fowlkes–Mallows Index, and Purity Score, indicating its potential as the best choice. However, the low Silhouette Score for $k = 12$ suggests diminishing returns, resulting in poorly separated clusters. To balance internal and external metrics, we selected $k = 9$ for our pen-based K-Means++ analysis. With $k = 9$, we obtained the highest Silhouette Score (0.563) and the lowest Davies–Bouldin Index (0.949), reflecting well-separated, cohesive clusters and strong internal consistency. Additionally, the Adjusted Rand Index (0.515) and Fowlkes–Mallows Index (0.575) are higher than those for $k = 8$, indicating better alignment with the true labels. Choosing $k = 9$ retains 80% of the variance and reduces the number of components from 16 to 5, as shown in Figure 6.

k	threshold	sil	db	ch	ari	fmi	purity
8	0.80	0.561	0.992	5012.216	0.502	0.567	0.646
9	0.80	0.563	0.949	4958.496	0.515	0.575	0.682
10	0.85	0.547	1.075	3974.102	0.556	0.607	0.706
11	0.85	0.347	1.151	3917.323	0.605	0.645	0.798
12	0.85	0.352	1.156	3866.193	0.627	0.664	0.809

Table 5: Best pen-based K-Means++ Clustering Results per k

3.3 Comparison of results

In this section, we evaluate the performance of the K-Means++ and OPTICS algorithms on the CMC and pen-based datasets, both with and without Principal Component Analysis (PCA) for dimensionality reduction. Note that the information for clustering without PCA is obtained in *W3 - Clustering*, while the results with PCA are discussed in Sections 3.1 and 3.2. To compare the results with the previous work, we use exactly the same preprocessing techniques.

For the cmc dataset, K-Means++ without PCA performed best with K = 5, yielding a Silhouette score of 0.2400 and a Davies-Bouldin index of 2.3400, indicating poor cluster separation. Applying PCA reduced the number of clusters to K = 2, improving the Silhouette score to 0.3060 and lowering the Davies-Bouldin index to 2.0790, while the Adjusted Rand Index (ARI) and Fowlkes-Mallows Index (FMI) slightly increased, though Purity decreased marginally. On the other hand, Optics performed better without PCA on the cmc dataset, achieving a high Silhouette score of 0.6610 and a low Davies-Bouldin index of 0.6260 with K = 2 clusters. With PCA, the best performer for Optics was the configuration with K = 3, which slightly reduced internal metrics but improved ARI and Purity, enhancing alignment with true labels.

For the pen-based dataset, we identified K-Means++ without PCA as the best performer with K = 12, achieving a Silhouette score of 0.4900 and a Davies-Bouldin index of 1.2700. Applying PCA, we selected

K = 9, which significantly improved the Silhouette score to 0.5630 and lowered the Davies-Bouldin index to 0.9490, though ARI and Purity declined. Optics excelled on the pen-based dataset without PCA, where we determined K = 14 clusters to be the best performer with an excellent ARI of 0.9900 and near-perfect Purity of 0.9980, alongside strong internal metrics. Applying PCA reduced the clusters to K = 11 in our evaluation, further enhancing internal metrics but causing a slight decrease in ARI and Purity.

Dataset	Clustering Algorithm	PCA	Best K	Silhouette	DB	CH	ARI	FMI	Purity
cmc	Optics	No	2	0.6610	0.6260	452.2230	-0.0050	0.4180	0.4120
	Optics	Yes	3	0.4740	0.8140	264.8750	0.0460	0.3680	0.4740
	K-Means++	No	5	0.2400	2.3400	147.8500	0.0100	0.3000	0.4500
	K-Means++	Yes	2	0.3060	2.0790	319.3370	0.0330	0.4400	0.4270
pen-based	Optics	No	14	0.4210	0.7160	737.3410	0.9900	0.9930	0.9980
	Optics	Yes	11	0.4610	0.5200	792.1060	0.9260	0.9530	0.9860
	K-Means++	No	12	0.4900	1.2700	2639.1900	0.5700	0.6100	0.7600
	K-Means++	Yes	9	0.5630	0.9490	4958.4960	0.5150	0.5750	0.6820

Table 6: Clustering results for cmc and pen-based datasets using K-Means++ and OPTICS with and without PCA.

3.4 Visualization

In this task, we aim to visualize clustering results in a 3-dimensional space to better understand their structure and separability using PCA and UMAP. PCA is a linear method that preserves global structure by maximizing variance, while UMAP is a non-linear technique that excels at maintaining local neighborhood relationships, making it more effective for revealing complex patterns. By applying these techniques, we will analyze and visualize the results of the best-improved versions of the K-Means and OPTICS algorithms, both with and without dimensionality reduction.

3.4.1 Orginal dataset

To compare the effect of reducing the dimensionality of the data by clustering, we first show the clusters obtained with the original data. So, from the preprocessed data, we perform clustering with K-Means++ and OPTICS for each dataset, obtaining the numeric results shown in Table 6.

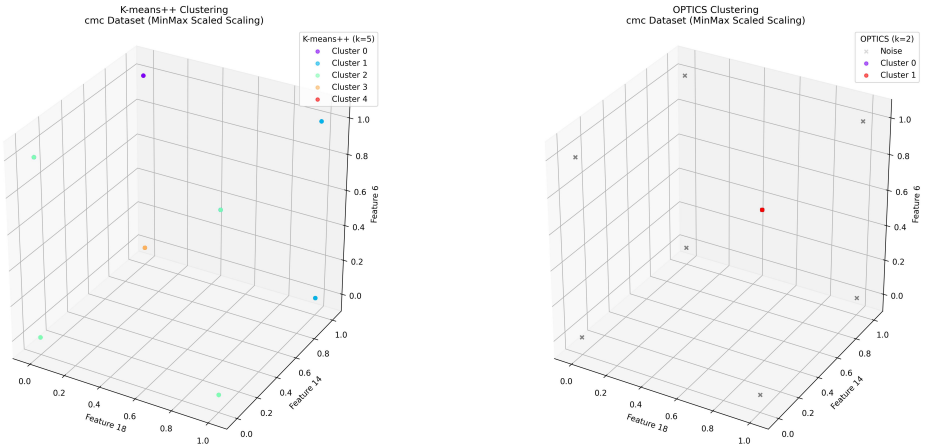

Figure 7: Clustering on cmc dataset without dimensionality reduction

The scattered point distribution on Figure 7 observed in the corners of the feature space is a direct result of the data encoding scheme, likely stemming from categorical variables that were one-hot encoded. This encoding constrains the data points to discrete locations, typically 0 or 1, creating a distinctive corner-based pattern where points are restricted to the vertices of a cube rather than being continuously distributed. This highly discretized space highlights a limitation of applying traditional distance-based clustering algorithms to one-hot encoded data. For instance, K-means++ attempts to group these discrete corner positions into clusters, assigning the points to 5 clusters, while OPTICS, as a density-based algorithm, struggles with this distribution, classifying most points as noise (gray x's) and identifying only 2 meaningful clusters. This geometric structure challenges the extraction of meaningful patterns, as the distances between points in this transformed space may not accurately represent the true relationships between the original categorical variables.

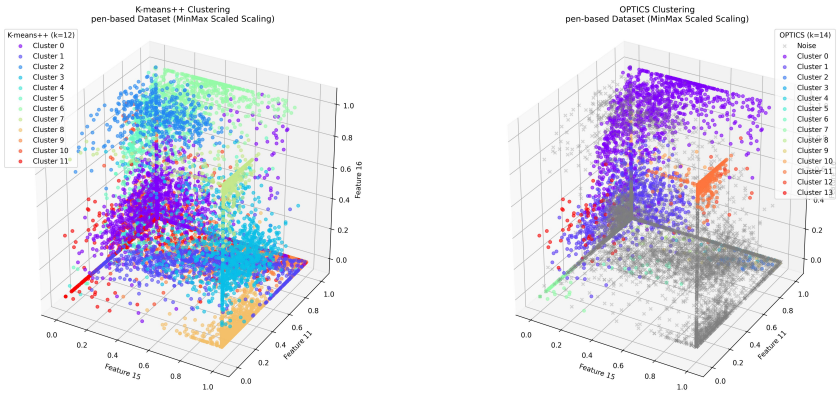

Figure 8: Clustering on pen-based dataset without dimensionality reduction

Looking at the pen-based visualization on Figure 8 , we can observe a meaningful structure in how the points are distributed, unlike the earlier scenario where they were constrained to corners due to encoding. Both K-means++ and OPTICS are uncovering real patterns in the data, though they approach clustering differently. K-means++ organizes the space into 12 well-distributed clusters, with each cluster occupying a distinct region that aligns with the flow of the data points, making the clustering appear reasonable and balanced. On the other hand, OPTICS takes a more selective approach, identifying 14 clusters while classifying a significant number of points as noise (represented by gray x's). This behavior reflects OPTICS' focus on densely packed regions, which aligns well with the nature of handwritten numbers, as certain writing styles tend to be more frequent. The clusters identified by OPTICS are particularly distinct, with strong separation in areas such as the purple and orange clusters, emphasizing its sensitivity to density-based patterns.

3.4.2 PCA

cmc Figure 9 shows the Orginal PCA(left), PCA+K-Means++ (middle), and PCA+OPTICS (right) visualization of our best performers with dimensionality reduction. When visualizing the data with their corresponding ground truth labels (Figure 9 left), we end up with an unexpectedly intertwined cloud that appears more or less chaotic, which can be viewed as a difficult problem to clusterize and generate groups. From our experiments, we determined that the optimal number of clusters for the K-Means++ algorithm is $k = 2$. However, as shown in Figure 9 (middle), the resulting clusters are not well-separated and exhibit significant overlap. This overlap makes it challenging to clearly distinguish between the groups.

Meanwhile, OPTICS visualization highlights the considerable impact of the numerous noise points. In *Work 3 - Clustering*, we evaluated the clusters using various internal and external evaluation metrics, while excluding the noise points from the evaluation, following standard techniques for OPTICS clustering, as noise points are not considered part of any cluster. However, since we did not visualize the clusters at that stage, we did not notice the significant number of noise points. Instead, we selected the best parameters purely based on the evaluation metrics. To investigate this further, we revisited our previous work to analyze the behavior of the noise points. From this analysis, we concluded that when evaluating and selecting the best parameters for OPTICS, it is crucial to consider not only the evaluation metrics but also the number of noise points. If you are interested in reviewing the parameters that resulted in the smallest number of noise points along with their corresponding evaluation metric scores, please refer to Appendix A.

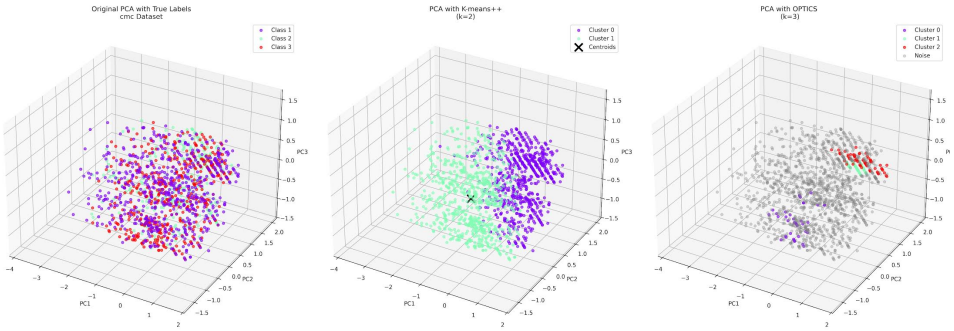

Figure 9: Dimensionality reduction with PCA and clustering methods for the cmc dataset

pen-based In Figure 10 (left), the ten ground truth labels form clearly separable clusters, indicating that the data has a high degree of class distinctiveness. Although a few outliers are visible, such as some purple points that appear to the top-right of the main cluster, the majority of points for each class remain tightly grouped. One can also spot that certain classes, for example the red cluster in the upper region, are particularly cohesive, suggesting that these labels represent well-defined patterns in the dataset.

The second subplot (Figure 10, middle) represents the clustering result obtained by applying K-Means++ (with $k = 9$) to the PCA-reduced data. While the clustering algorithm captures some structure in the data, the clusters exhibit considerable overlap, particularly between adjacent groups, and lack clear boundaries. This indicates that the clustering process struggles to fully capture the underlying separation of classes.

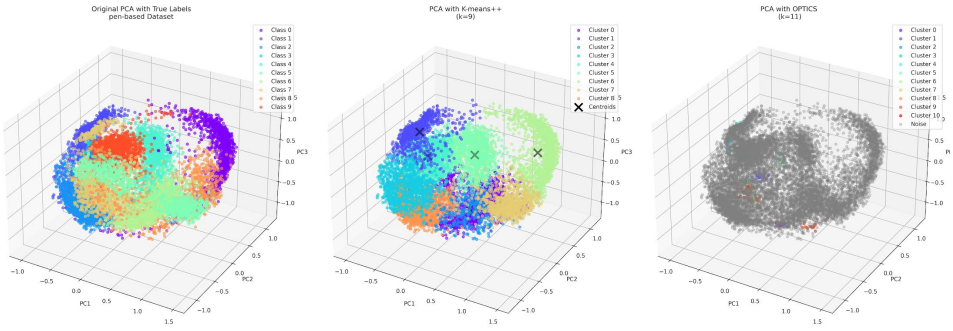

Figure 10: Dimensionality reduction with PCA and clustering methods for the pen-based dataset

The third plot (Figure 10, right) ,shows the clustering result obtained using OPTICS (with $k = 11$) on the same PCA-reduced data. OPTICS results, highlights once more the significant effect of noise points in this method. A large fraction of the dataset has been assigned as noise (shown in grey), with only a few smaller clusters emerging in the background. The abundance of noise points suggests that the distribution of the dataset does not perfectly suit the underlying assumptions of the algorithm.

3.4.3 UMAP

cmc Figure 11 presents the application of UMAP for dimensionality reduction on the CMC dataset, visualized in three subplots. The first subplot (left) shows the UMAP-reduced data with true labels, offering a baseline view of the dataset's structure. As we can see, it exists a notable overlap between certain groups.

The second subplot (Figure 11, middle) depicts the clustering result using K-Means++ (with $k = 2$) on the UMAP-reduced data. Cluster 0 appears fragmented, forming multiple small, localized groups scattered throughout the UMAP-reduced space. In contrast, Cluster 1 covers a larger, denser region, exhibiting a more continuous and elongated shape around its centroid.The centroids reflect the geometric centers of their clusters, with Cluster 1's centroid situated near the densest area, emphasizing its dominant structure, while Cluster 0's centroid lies within the scattered groups, capturing their spatial dispersion. Although the two clusters are separated, some transitional points near the cluster boundaries highlight overlaps.

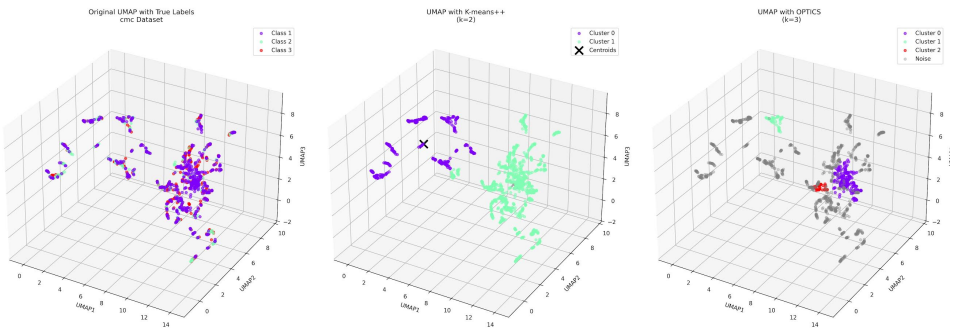

Figure 11: Dimensionality reduction with UMAP and clustering methods for the cmc dataset

The third subplot (Figure 11, right) shows the clustering result obtained using OPTICS (with $k = 3$). Cluster 0 is the largest and most prominent, with a dense grouping of points located near the center of the UMAP-reduced space. Cluster 1 is smaller and more isolated, appearing in a distinct region far from

the other clusters, indicating good separation. Cluster 2 is compact and relatively small, tightly grouped in a localized area, showing well-defined boundaries.The noise points, represented in gray, are scattered throughout the space, highlighting areas where the algorithm could not assign points to any specific cluster due to insufficient density or ambiguous proximity. The presence of these noise points emphasizes the complexity of the dataset and the limitations of OPTICS in fully classifying all data points.

pen-based Figure 12 illustrates again the application of UMAP for dimensionality reduction on the pen-based dataset. The first subplot (Figure 12 left) shows the UMAP-reduced data with true labels, highlighting the inherent structure of the dataset. While some classes are distinctly separated, there is still noticeable overlap between certain groups.

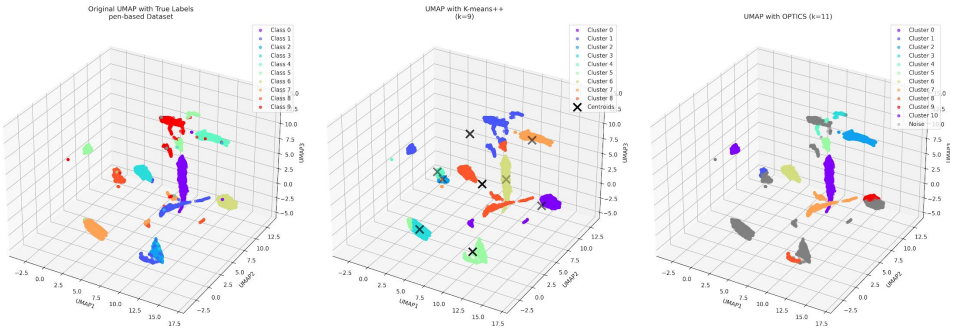

Figure 12: Dimensionality reduction with UMAP and clustering methods for the penbased dataset

The second subplot (Figure 12 middle) depicts the clustering result obtained using K-Means++ (with $k = 9$) on the UMAP-reduced data. The clusters are generally well-separated in the UMAP space, indicating that the dimensionality reduction effectively preserved the inherent structure of the high-dimensional data. Some clusters, such as those in orange, green, and blue, are tightly grouped, demonstrating strong clustering performance in these areas. However, in certain cases, such as cluster 2 and cluster 3, some overlap is observed, which may indicate challenges in differentiating similar data points.The third subplot (Figure 12, right) presents the clustering result using OPTICS (with $k = 11$). The identified clusters are well-separated in many areas, appearing compact and distinct. However, the Noise category (gray points) suggests that some data points did not meet the density criteria, which may indicate sparse or ambiguous regions in the dataset.Clusters like red and green display irregular and stretched shapes, a common characteristic of density-based methods that adapt to the underlying distribution of the data.

We also generated visualizations by creating clusters that match the true number of classes to analyze the clustering behavior. For detailed results, please refer to Appendix B and Appendix C.

3.5 Execution Time

Table 7 shows a comparison of the execution times for the clustering both with the data reduction and without. We check the computational time of each configuration, each one with the best parameters determined previously.

Dataset	Reduction dimensionality technique	Method	Time (s)
cmc	None	OPTICS	0.105
	None	K-Means++	0.081
	PCA (ours)	OPTICS	0.073
	PCA (ours)	K-Means++	0.066
	PCA (sklearn)	OPTICS	0.070
	PCA (sklearn)	K-Means++	0.077
	Incremental PCA	OPTICS	0.085
	Incremental PCA	K-Means++	0.067
	UMAP	OPTICS	2.032
	UMAP	K-Means++	1.088
pen-based	None	OPTICS	3.588
	None	K-Means++	0.473
	PCA (ours)	OPTICS	3.320
	PCA (ours)	K-Means++	0.120
	PCA (sklearn)	OPTICS	2.980
	PCA (sklearn)	K-Means++	0.167
	Incremental PCA	OPTICS	2.8
	Incremental PCA	K-Means++	0.153
	UMAP	OPTICS	5.214
	UMAP	K-Means++	3.813

Table 7: Comparison of clustering methods and dimensionality reduction techniques across datasets.

4 Analysis and Conclusions

To conclude, in this work we successfully implemented PCA from scratch and validated its effectiveness against established library-based methods, demonstrating its accuracy in reducing the dimensionality of our datasets. Importantly, the results from our PCA matched those obtained using scikit-learn's PCA. Additionally, Incremental PCA (IPCA) offered advantages in handling larger datasets by mitigating memory constraints. PCA provides valuable insights into the underlying structure of datasets by reducing dimensionality and highlighting the principal components that capture the most variance, facilitating the identification of key features and patterns within the data. This makes it easier to understand the intrinsic properties of complex datasets. However, dimensionality reduction through PCA can sometimes result in the loss of critical information, which may impact optimal cluster alignment. Consequently, while PCA proves beneficial for uncovering underlying patterns in larger and more complex datasets like the pen-based dataset, its advantages are less pronounced for relatively smaller and more mixed datasets like the CMC dataset.

Regarding visualization, our analysis revealed that for the CMC dataset, PCA attempts to separate different classes by grouping similar ones and distancing them from others. However, the data's inherent mixing makes it almost impossible to discern the different true labels clearly, thereby making clustering very challenging. In contrast, the pen-based dataset yielded superior results when applying PCA, as it effectively separated the data by clustering similar points along the principal components, maximizing variance and revealing clearer group structures.

On the other hand, OPTICS was not particularly effective in providing insights into the underlying information in our datasets. As discussed in Section 3.4, OPTICS struggled with a significant number of noise points. Despite extensive experimentation with various parameter settings to reduce these noise points, it became clear for us that the dataset's distribution does not align well with the algorithm's assumptions. Given this limitation, Spectral Clustering would be a better alternative to use in this work, as it excels at capturing the underlying structures of the datasets and provides more meaningful insights compared to OPTICS. While evaluating various clustering algorithms on the reduced datasets offered mentioned insights, we cannot definitively conclude whether these results align with those from the original, non-reduced data. This limitation arises from the nature of our analysis, which relies on clustering evaluation metrics rather than direct pairwise comparisons between reduced and non-reduced outcomes. Moreover, we selected our best-performing clusterings based on different values of k (see Table 6), making a direct comparison across reduced and non-reduced scenarios unfeasible.

When comparing the visualizations obtained using PCA and UMAP, both preserve the overall structure of the original data, facilitating the analysis of high-dimensional datasets. However, they exhibit distinct differences in their behavior and outcomes. On the pen-based dataset, PCA organizes data points into overlapping clouds, making it challenging to discern clear cluster boundaries. UMAP, on the other hand, provides better separation of clusters by preserving local structures and neighborhood relationships. For instance, UMAP can organize data into distinct patches even in 3D space, behaving somewhat like a clustering algorithm, though its primary goal is dimensionality reduction. This distinct separation benefits clustering algorithms like K-Means++ and OPTICS. However, UMAP also reflects natural ambiguities in the dataset, such as the similarity between the digits '2' and '7.' In contrast, PCA, being a linear technique, struggles to separate data points as well. While K-Means++ performs reasonably in the PCA space, it fails to distinguish certain digits, such as '8' and '6,' due to overlapping clusters. Additionally, OPTICS struggles more with dense and overlapping clusters in PCA but performs better on UMAP-reduced data due to UMAP's superior separation. When applied to the CMC dataset, which is predominantly categorical with few continuous variables, both PCA and UMAP face challenges in effectively capturing the data's natural structure. One-hot encoding of categorical variables creates a subspace where data points are saturated at the corners of the [0, 1] space, limiting the effectiveness of dimensionality reduction techniques like PCA and UMAP.

It is also important to note that during the visualization of OPTICS using PCA and UMAP, we had to adjust the parameters. This adjustment was necessary because the same parameters resulted in different numbers of clusters due to variations in the density of the data points.

Regarding the clustering execution time discussed in Section 3.5, applying dimensionality reduction consistently reduces clustering time across all cases, which aligns with our initial expectations since fewer features result in a simpler grouping process. It is evident that UMAP required more time compared to other methods. Additionally, it is important to note that the grid search for identifying the optimal parameters for K-Means++ and OPTICS took slightly longer than in previous work due to the introduction of grid search on the percentage of variance explained.

References

[1] Charu C. Aggarwal and Chandan K. Reddy. *Clustering Algorithms and Applications*. CRC Press, 2013.

[2] Dheeru Dua and Casey Graff. "UCI Machine Learning Repository". In: *University of California, Irvine, School of Information and Computer Sciences* (2019). URL: `http://archive.ics.uci.edu/ml`.

[3] Ian T. Jolliffe and Jorge Cadima. *Principal Component Analysis*. Springer, 2002.

[4] Leland McInnes, John Healy, and James Melville. "UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction". In: *arXiv preprint arXiv:1802.03426* (2018). URL: `https://arxiv.org/abs/1802.03426`.

[5] Fabian Pedregosa et al. "Scikit-learn: Machine Learning in Python". In: *Journal of Machine Learning Research* 12 (2011), pp. 2825–2830.

[6] John Zelle. "Python programming: An Introduction to Computer Science". In: (2010).

A Minimum Number of Noise Points Achieved

Dataset	min_samples	algorithm	max_epsilon	Silhouette	DB	CH	ARI	Purity	FMI	k	Noise Pts
pen-based	20	auto	1	0.3384	0.9859	2420.600	0.6246	0.7334	0.6912	10	2195
cmc	20	auto	3	0.4567	0.9662	177.667	0.0084	0.4596	0.1505	20	620

Table 8: Clustering Evaluation Metrics

B Visualization of Clustering Results with True Class Labels and PCA

The clustering results for the CMC and Pen-Based datasets reveal distinct patterns in the effectiveness of PCA, K-Means++, and OPTICS. For the CMC dataset, PCA applied to the original data with true labels demonstrates significant overlapping clusters, as the data is inherently mixed and lacks clear separability. When PCA is combined with K-Means++, the algorithm forms three clusters corresponding to the true classes; however, these clusters remain poorly separated, with substantial overlap, reflecting the difficulty of partitioning the data due to its intrinsic complexity. OPTICS further highlights the challenges of clustering the CMC dataset, identifying a large number of noise points and forming clusters that are not well-defined. This demonstrates that the dataset's high degree of overlap significantly limits the performance of clustering algorithms, even after dimensionality reduction.

In contrast, the Pen-Based dataset exhibits well-separated clusters after applying PCA, especially in the original data visualization with true labels, where distinct groups are evident with minimal overlap. When PCA is combined with K-Means++, the algorithm successfully forms ten clusters corresponding to the true classes, with relatively well-separated regions in the reduced-dimensional space. However, some overlapping clusters are still present, particularly in areas of natural ambiguity. OPTICS also performs better on the Pen-Based dataset than on the CMC dataset, forming tighter, more distinct clusters while identifying fewer noise points. Despite this improvement, some overlapping clusters and noise points remain, indicating the inherent limitations of OPTICS in handling denser regions of the data.

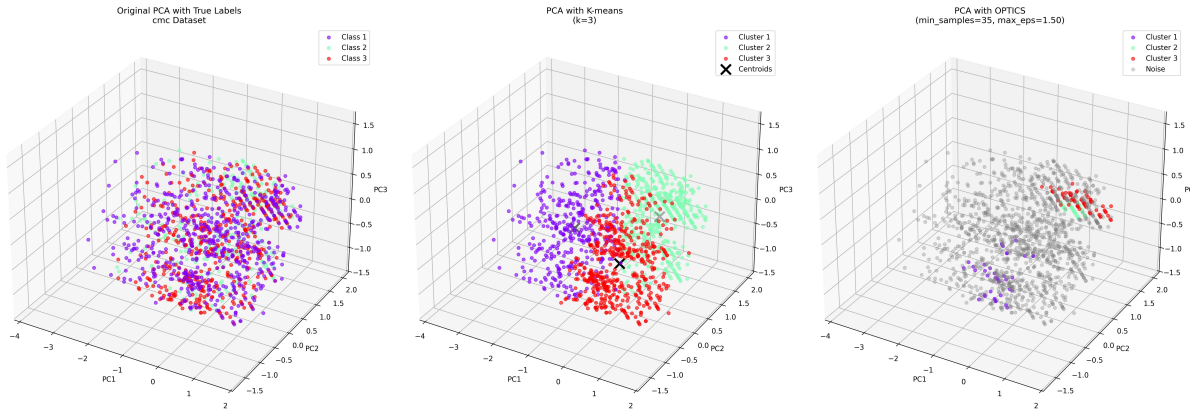

Figure 13: cmc dataset with 3 clusters

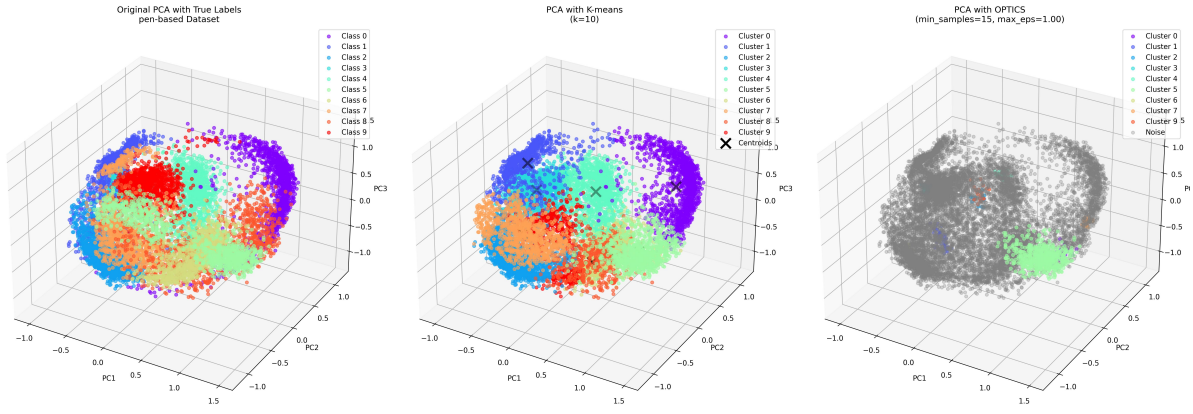

Figure 14: pen-based dataset with 10 cluster

C Visualization of Clustering Results with True Class Labels and UMAP

The clustering results using UMAP, K-Means++, and OPTICS for the CMC and Pen-Based datasets reveal significant differences in performance based on the dataset characteristics. For the CMC dataset, UMAP applied to the original data with true labels shows overlapping clusters with minimal separation, indicating the intrinsic difficulty of clustering this dataset. When combined with K-Means++, UMAP produces three clusters corresponding to the true classes; however, these clusters remain poorly separated with significant overlap, reflecting the challenges of clustering highly mixed data. OPTICS applied to the UMAP-reduced CMC dataset identifies three clusters but also designates a substantial number of points as noise, highlighting the dataset's inherent complexity and the limitations of OPTICS in such cases. On the other hand, the Pen-Based dataset demonstrates much better clustering outcomes with UMAP. The true labels indicate that UMAP effectively separates the ten classes into distinct, well-separated clusters, significantly reducing overlap. When combined with K-Means++, UMAP produces well-defined clusters that closely align with the true classes, showcasing its ability to enhance clustering performance. OPTICS also benefits from UMAP's dimensionality reduction on the Pen-Based dataset, forming tight and distinct clusters with fewer noise points compared to the CMC dataset. However, minor overlap and noise remain in denser regions. Overall, UMAP proves to be a powerful tool for dimensionality reduction, particularly for datasets like Pen-Based that exhibit clear intrinsic structures, while the challenges with the CMC dataset highlight the limitations of clustering algorithms in managing overlapping clusters and poorly separated data.

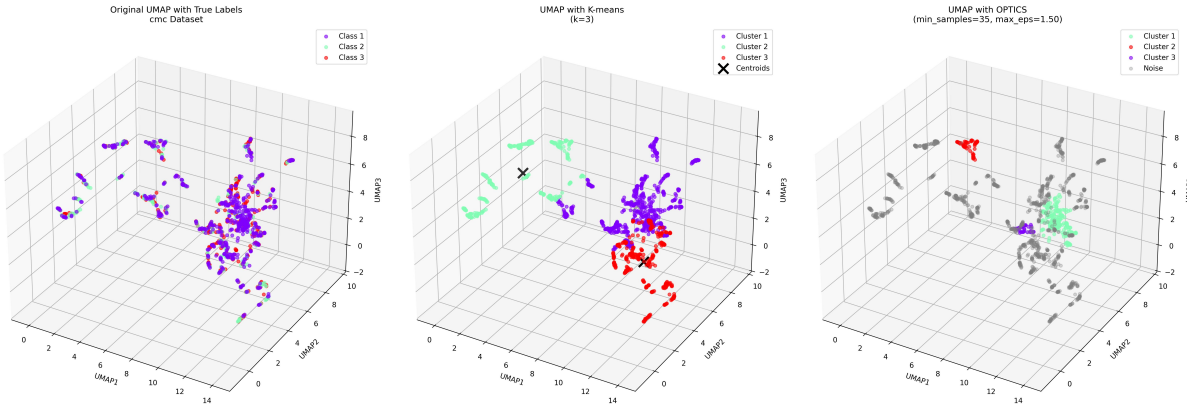

Figure 15: cmc dataset with 3 clusters

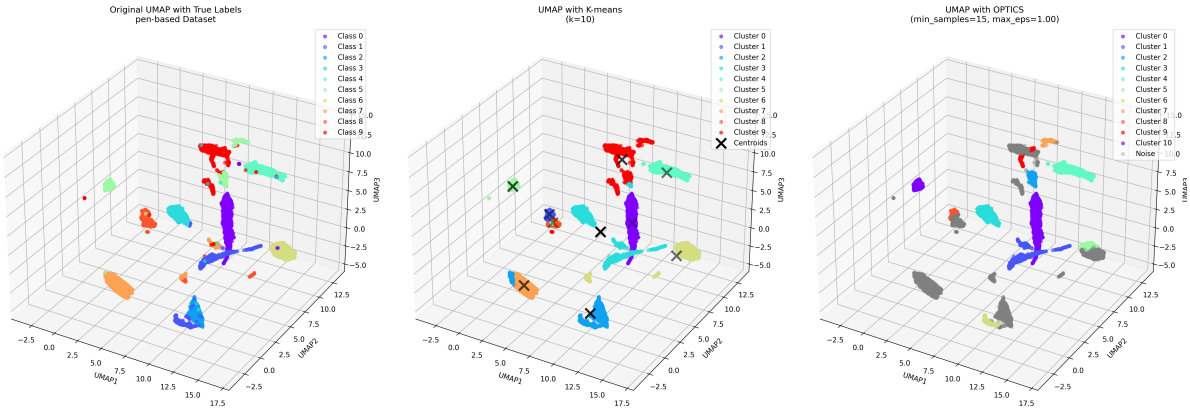

Figure 16: pen-based dataset with 10 cluster