How Can Integration of Explainability Tools and Dimensionality Reduction Techniques Improve the Interpretability of PCA

Introduction

In the rapidly evolving landscape of data science and analytics, Principal Component Analysis (PCA) stands out as a cornerstone technique for dimensionality reduction. Its ability to transform complex datasets into simpler, more manageable forms without significant loss of information makes it invaluable across various fields, from genomics and finance to social sciences. However, the interpretability of PCA, especially in high-dimensional data contexts, remains a challenging aspect. This paper addresses this gap by integrating machine learning explainability tools and dimensionality reduction techniques to enhance PCA's interpretability while preserving the integrity of the original data.

The relevance of this study lies in its potential to make PCA more accessible and understandable, particularly for practitioners and researchers dealing with complex, high-dimensional datasets. In fields like genomics, where the data's dimensionality can be exceptionally high, PCA's ability to reduce dimensions is crucial but often leads to a trade-off between simplicity and the retention of original information. This balance is critical in ensuring that significant insights are not lost in the process of simplification. Furthermore, the subjective nature of interpreting PCA loadings can lead to varied conclusions, highlighting the need for standardized guidelines and enhanced interpretability tools. By addressing these challenges, this paper aims to contribute significantly to the fields of data science and analytics, making PCA a more robust and transparent tool for researchers and practitioners.

The paper is structured as follows:

1. **Related Work**: We review existing literature on PCA, focusing on its applications in different domains and the current state of explainability in PCA. This includes an examination of existing methods for interpreting PCA results and the use of machine learning explainability tools in various contexts.
2. **Experiments**: We conduct a series of experiments to evaluate the effectiveness of integrating machine learning explainability tools with PCA. Additionally, we explore the combination of PCA with other dimensionality reduction techniques like t-SNE and UMAP to see if a hybrid approach improves explainability.
3. **Results of the Experiments:** The results section presents our findings and the impact of hybrid dimensionality reduction techniques on the explainability of PCA.
4. **Conclusion**: We conclude with a discussion of the implications of our findings for the broader field of data analytics. This includes recommendations for practitioners and researchers on utilizing PCA in high-dimensional data analysis and suggestions for future research directions in enhancing the explainability of PCA.

Related work

The research on the integration of explainability in PCA, particularly in the context of machine learning, appears to be an emerging area with limited direct empirical studies. The available literature primarily focuses on the broader concept of explainability in machine learning, rather than specifically on PCA. Here is what we found about previous research :

- The paper titled "Explainability in Machine Learning: a Pedagogical Perspective" addresses the lack of pedagogical resources for teaching explainability in machine learning. The authors emphasize the need for resources that help in educating about the advantages of explainability, noting that current teaching methods often focus more on applying machine learning models rather than explaining their decision-making processes. The paper proposes a structured approach to teaching explainability, covering both the advantages and disadvantages of various opaque and transparent machine learning models. It also discusses how to implement and interpret different explainability techniques, along with ways to structure assignments to enhance students' learning experience. This approach aims to give data science professionals a comprehensive understanding of this rapidly developing field, enabling them to deploy machine learning more effectively.
- The paper "Explainable Dimensionality Reduction (XDR) to Unbox AI 'Black Box' Models: A Study of AI Perspectives on the Ethnic Styles of Village Dwellings" introduces a novel XDR framework designed to convert the high-dimensional, tacit knowledge learned by AI models into explicit knowledge that is understandable to domain experts. This framework is exemplified through a case study in Guangdong, China, where an AI model identifies ethnic styles of village dwellings from satellite imagery. Key features like patio size, length, direction, and asymmetric shape are used to distinguish between Canton, Hakka, Teochew, and their mixed architectural styles. The findings, which are consistent with existing field studies, also uncover evidence of Hakka migration, thus contributing new insights to architectural and historical geography. This XDR framework is shown to be valuable for experts in various fields, helping to expand their domain knowledge.

- The paper "Improving Statistical Reporting Data Explainability via Principal Component Analysis" proposes a novel approach using principal component analysis for dimension reduction and feature extraction in the context of auto insurance statistical reporting data. This method aims to address challenges in data visualization and interpretability, particularly in complex statistical data analysis. By investigating the relationship between loss relative frequency and size-of-loss, along with the variability of extracted features, the study enhances the understanding of auto insurance loss data. The proposed PCA-based method not only improves data explainability but also provides an in-depth analysis of the patterns in size-of-loss relative frequency. The findings are particularly beneficial for insurance regulators, aiding in more informed rate-filing decisions that benefit both insurers and clients. Additionally, the approach has applications in other business data analysis scenarios.
- The article "SVD to PCA: Technique to Improve XAI (Explainable AI) (Part 2)" focuses on the challenge of explainable AI in the context of complex machine learning models. As deep learning algorithms become more sophisticated, their predictability increases, but their explainability decreases. The article proposes the use of Principal Component Analysis to address this issue. PCA is a data analysis technique that reduces dimensionality and extracts useful features without significantly compromising information. It reorients the data matrix to maximize variability and minimize redundancy, helping to simplify and explain complex datasets. The article demonstrates this with a case study using a breast cancer dataset from Scikit-Learn, showing how PCA can effectively reduce the number of variables while retaining most of the information. This approach helps in ML model explainability by revealing the relationships and characteristics of latent attributes, making it easier to understand the behavior of ML models. The article concludes that PCA, along with other techniques like SVD, t-SNE, and NMF, are powerful tools for reducing dimensionality and enhancing the explainability of machine learning models.

Experiments

In this section, we embark on a detailed exploration of various experimental setups designed to evaluate the efficacy of integrating explainability tools with dimensionality reduction techniques in enhancing the interpretability of PCA in high-dimensional data analysis. We begin by outlining our experimental methodology, including the selection of datasets, the criteria for choosing specific dimensionality reduction techniques, and the explainability tools employed.

Data preparation

The dataset described is a collection of features extracted from handwritten numerals (0-9) found on University of California Irvine website. It comprises 2,000 patterns in total, with 200 patterns for each numeral class. These patterns have been converted into binary images and are represented across six different feature sets:

1. **mfeat-fou**: This file contains 76 Fourier coefficients representing the shapes of the characters.
2. **mfeat-fac**: It includes 216 profile correlations.
3. **mfeat-kar**: This set features 64 Karhunen-Love coefficients.
4. **mfeat-pix**: It consists of 240 pixel averages in 2 x 3 windows.
5. **mfeat-zer**: This file includes 47 Zernike moments.
6. **mfeat-mor**: It contains 6 morphological features.

Each of these feature sets is stored in a separate file in ASCII format, with 2,000 lines per file. The patterns are organized sequentially by class, starting with 200 patterns of class ‘0’, followed by 200 patterns each for classes ‘1’ through ‘9’. The corresponding patterns in different feature files are linked to the same original handwritten character. Notably, the original source images from which these features were extracted are no longer available. These feature sets were import and merged together to form a dataset with 2000 observation and 649 explanatory variables.

Standardization of the variables
Standardization of variables is a critical step when using Principal Component Analysis (PCA), especially when the variables have different units of measurement or vastly different scales. For instance, a variable with a large range (like pixel values) might dominate the principal component structure due to its variance, not because it's more informative.

- **Removal of Mean**: PCA is sensitive to the mean of the variables. Standardizing variables (subtracting the mean and dividing by the standard deviation) ensures that each variable has a mean of zero.
- **Scaling to Unit Variance**: This step involves dividing each variable by its standard deviation. After scaling, each variable contributes equally to the distance measurements, making the analysis more balanced.

PCA

Model

Principal Component Analysis (PCA) is a statistical technique used for dimensionality reduction while preserving as much of the data's variation as possible. It's particularly useful in situations where you have a large number of interrelated variables and you want to simplify the data without losing significant information. PCA is widely used in exploratory data analysis, noise reduction, feature extraction, and data visualization. It's valuable for simplifying complex datasets while retaining their core characteristics. After data is standardized, several steps are performed to obtain the reduced data.

Covariance Matrix Computation
We first compute the covariance matrix of the dataset. The covariance matrix expresses how each variable in the dataset relates to every other variable. When two variables are highly correlated, this means that there is a strong linear relationship between them. Correlation indicates that there is redundancy in the data. Because of this redundancy, PCA can be used to reduce the original variables into a smaller number of new variables.

Eigenvalue and Eigenvector Calculation: From the covariance matrix, we then computes the eigenvectors and eigenvalues.

- An eigenvector is a special vector that doesn't change its direction during this transformation. It might get longer or shorter, but it points in the same or exactly opposite direction as before. Eigenvectors are the directions of the axes where data is most spread out.
- Eigenvalues represent the magnitude of this spread measure the amount of variance explained by each principal axis - the larger the eigenvalue, the more variation in the data along that eigenvector. This is a number telling you how much the eigenvector gets stretched or compressed during the transformation. If the eigenvector gets twice as long, the eigenvalue is 2. If it stays the same length, the eigenvalue is 1.

Principal component selection

- **Kaiser rule** : After that, we use the Kaiser rule, which is a method used to decide the number of factors or principal components to retain from a set of data. The Kaiser rule suggests retaining all components (or factors) with eigenvalues greater than 1. The reasoning behind this rule is that an eigenvalue of 1 corresponds to the amount of variance explained by one original variable in the dataset. Thus, any component with an eigenvalue greater than 1 is explaining more variance than a single variable and is therefore considered significant.
- **Sort Eigenvalues and Eigenvectors**: The eigenvectors are sorted in order of decreasing eigenvalues. This sorting is essential because it ranks the eigenvectors from the most important to the least.
- **Choose of explained variance threshold** : We also limit the number of axes to represent a certain fraction of the total variance. There is no well-accepted objective method for deciding how many principal axes are sufficient. This will depend on the specific application domain and the specific dataset. In practice, we tend to look at the first principal axes to find interesting insights in the data.

Projection
Finally, the data is projected onto the new feature space. This is achieved by multiplying the original data matrix by the matrix of eigenvectors. The first few eigenvectors will capture the most variance in the data set. We obtain this reduced database deducted from the initial one, with our new explanatory variables which are the principal components obtained by the above method and which we will consider for the interpretation of the graphs, and all the rest of the work.

Hyper parameter selection

The selection of hyperparameters for Principal Component Analysis (PCA) primarily involves deciding on the number of principal components to retain. Here, we explore different total variance threshold to determine which one will be chosen. The code iteratively explores a range of variance thresholds $[0.2, 0.3, \ldots, 0.9]$ with each threshold representing the cumulative percentage of total variance that must be explained by the selected components.
For each variance threshold, PCA is fitted to the training data, and the minimum number of components required to meet the threshold is determined. The `np.cumsum` function is used to compute the cumulative sum of explained variances, and the number of components to retain is identified by finding the point where the cumulative variance first exceeds the threshold.

Evaluation steps

Then, the model evaluate the model performance on three different classifier : Random Forest, Neural network and KNN algorithm. Each classifier is trained using the transformed training set with the reduced dimensionality and then evaluated on the transformed testing set. The performance metric used is accuracy, which measures the proportion of correct predictions made by the classifier. The accuracies for each classifier, at each variance threshold, are stored in a Dataframe. This structured approach facilitates comparison across different models and hyperparameter settings. The code culminates in printing the results Dataframe, which would contain the accuracy scores of each classifier across the range of PCA variance thresholds. These results can be analyzed to understand the trade-offs between model complexity (as a function of the number of retained principal components) and classification accuracy. Then, different tools are used to try to understand the global structure of the components of the reduced database.

The second step of the evaluation is that we will use the SHAP algorithm to interpret the principal components in the context of the chosen classifier model. SHAP (SHapley Additive exPlanations) is a game theory-based approach for explaining the output of machine learning models. It assigns each feature an importance value for a particular prediction, which is essentially a contribution margin that tells us how much each feature contributes, either positively or negatively, to the target variable compared to the average prediction.

SHAP values are particularly pertinent in the context of machine learning explainability for several reasons:

- **Local Interpretability**: They provide local explanations, meaning they can show how much each feature contributed to each individual prediction, rather than just providing a global importance metric.
- **Global Interpretability**: Aggregating SHAP values over a dataset can also provide global insights into the model's behavior, highlighting which features are generally most important for the model's predictions.
- **Fair Attribution**: SHAP values are based on the concept of Shapley values from cooperative game theory, which ensures a fair distribution of payoffs (in this case, contribution to the prediction) among players (features).
- **Consistency**: SHAP ensures consistency in attributions; if a model changes so that a feature’s contribution increases or remains the same regardless of the other features, the attributed importance of that feature should not decrease.

In the context of PCA (Principal Component Analysis) explainability, SHAP can be used in conjunction with PCA in the following ways:

- **Dimensionality Reduction**: PCA is often used to reduce the number of features by creating principal components, which are new features formed by linear combinations of the original features. SHAP can help in understanding the impact of each principal component on the predictions, although it’s more common to apply SHAP directly to the original features.
- **Insight Into Transformed Space**: After PCA, a model is trained on the transformed feature space. SHAP can provide insights into how these new PCA features influence the model's predictions. However, the interpretation is less intuitive because PCA features do not correspond directly to the original features.
- **Bridging Original and PCA Features**: One can map back from PCA components to original features using the loadings of PCA components and SHAP values to understand how original features indirectly affect the model's output through their contributions to the principal components.

Steps Taken in the Code

1. **PCA Transformation**: The chosen model is trained on data that has been transformed by PCA, presumably to reduce dimensionality and focus on the most informative aspects of the data.
2. **SHAP Explainer Initialization**: A SHAP Explainer is created, which uses the prediction function and a background dataset derived from the PCA-transformed training set. The background set is limited to 50 instances to manage computational load.
3. **SHAP Value Calculation**: SHAP values are calculated for a subset (50 instances) of the PCA-transformed test set, which quantifies the impact of each feature on the model's output.
4. **Plotting SHAP Values**: The SHAP values are visualized using a summary plot. This plot shows the distribution of the SHAP values for each feature across all instances in the subset.

UMAP

Model

After performing Principal Component Analysis (PCA) to reduce the dimensionality of our dataset, we are now exploring another dimensionality reduction technique called UMAP (Uniform Manifold Approximation and Projection). While PCA allowed us to project our data into a lower-dimensional space while preserving linear relationships, UMAP offers a non-linear approach to data visualization. In this step, we will apply UMAP to our dataset to further explore the underlying structure and non-linear relationships between data points. This approach could potentially reveal more complex structures and groupings that PCA might not have captured. Let's embark on this exciting data exploration journey through UMAP to gain a more comprehensive perspective of our dataset.

UMAP, an acronym for Uniform Manifold Approximation and Projection, is a dimensionality reduction technique designed to create a lower-dimensional representation of high-dimensional data while preserving both global and local structures. The method employs a graph-based approach to construct a topological representation, subsequently embedding it in a lower-dimensional space using stochastic gradient descent. The UMAP procedure encompasses several key steps:

Functioning:

- **Neighborhood Identification**: UMAP starts by identifying local neighborhoods for each point in the high-dimensional space.
- **Graph Construction**: It constructs a weighted graph where each point is connected to its nearest neighbors.
- **Optimization**: UMAP then optimizes the layout of these points in the lower-dimensional space to reflect the structure of this graph as closely as possible. The optimization aims to preserve the local and global structure of the data.

Synergy between UMAP and PCA

- **Initial Dimension Reduction with PCA**: PCA can be used to initially reduce the dimensionality of the dataset, especially if the dataset has a very high dimension. This makes UMAP's computation more efficient and manageable.
- **Capturing Different Aspects of Data**: PCA can capture the global linear structure, while UMAP can then be used to unravel the local non-linear structures within this reduced space.
- **Noise Reduction**: PCA can help in removing noise and redundant features, which can improve the performance of UMAP on the cleaned dataset.
- **Exploratory Data Analysis**: Using PCA and UMAP in conjunction provides a more comprehensive understanding of the data's structure. PCA can be used for a quick overview, and UMAP can be applied for a deeper, more detailed exploration.

Benefits of UMAP:

1. **Speed**: UMAP is fast, handling large datasets and high-dimensional data efficiently, even surpassing many t-SNE packages.
2. **Scalability**: UMAP scales well in embedding dimension, serving as a general-purpose dimension reduction technique for various machine learning tasks.
3. **Global Structure Preservation**: UMAP often outperforms t-SNE in preserving aspects of the global structure of the data, providing a comprehensive overview.
4. **Versatile Distance Functions**: UMAP supports a wide range of distance functions, including non-metric ones like cosine and correlation distance.
5. **Dynamic Embedding**: UMAP allows adding new points to an existing embedding, making it suitable for use as a preprocessing transformer in sklearn pipelines.
6. **Supervised and Semi-Supervised Dimension Reduction**: UMAP supports supervised and semi-supervised dimension reduction, incorporating label information.
7. **Additional Features**: UMAP offers experimental features such as an "inverse transform," embedding into non-Euclidean spaces, and preliminary support for embedding dataframes.

Hyper parameter selection

There are two hyperparameters in the UMAP algorithm:

- **Number of Neighbors (n_neighbors):** This hyperparameter influences the number of neighboring points used in the local manifold approximation. It is pivotal in the UMAP algorithm as it dictates the balance between preserving the local and global structure of the data. In our exploration, we selected three values for the number of neighbors: 5, 15, and 30. This selection aims to understand how varying degrees of neighborhood sizes affect the resulting embeddings. A smaller number of neighbors allows UMAP to focus on fine-grained, local structure, potentially leading to tighter clusters. Conversely, a larger number of neighbors encourages UMAP to account for more global data structure, which can be beneficial for capturing the overall data distribution.
- **Minimum Distance (min_dist):** The minimum distance parameter controls how close points in the low-dimensional space can be to each other, and it plays a crucial role in determining the compactness of UMAP embeddings. For our study, we chose min_dist values of 0.0, 0.1, and 0.5. These values range from allowing overlapping clusters to enforcing a broader separation between points. A min_dist of 0.0 permits points to cluster densely, while higher values encourage points to spread out.

Evaluation Metric:
We utilize the silhouette score as the evaluation metric to assess the quality of the UMAP embeddings. This metric calculates how similar an object is to its own cluster compared to other clusters, providing a succinct measurement of the tightness and separation of the clusters. A higher silhouette score indicates that the embedding has well-defined, separate clusters, which is desirable in a dimensionality reduction technique like UMAP.

Grid Search:
To determine the optimal combination of hyperparameters, we conduct a grid search across the predefined ranges of n_neighbors and min_dist. The grid search method is exhaustive, evaluating every possible combination of hyperparameters

within our selected grid. This approach is particularly useful for UMAP, which, like t-SNE, has a cost function that is susceptible to local minima, making the algorithm's output sensitive to the choice of hyperparameters.

Optimization:
The optimization involves running UMAP with each pair of hyperparameters and recording the silhouette scores that result from these embeddings. These scores are meticulously cataloged along with their corresponding hyperparameters. The combination that yields the highest silhouette score is considered the best set of hyperparameters. By analyzing these scores, we can infer the impact of each hyperparameter on the data's low-dimensional representation and confirm that our selected hyperparameters produce the most meaningful and interpretable projection of our dataset.

Evaluation steps

Evaluating the quality of a UMAP algorithm's output incorporates both qualitative and quantitative measures to ensure a comprehensive assessment. We used the following methods:

- **Visual Inspection**: The first step was to visually examine the scatter plot generated by UMAP. This visual assessment focused on discerning whether the algorithm has formed distinct and separated clusters. A good UMAP projection should ideally reveal clear groupings that correspond to different categories or types within the dataset.
- **Silhouette Score**: To quantitatively measure the clustering quality, we employed the silhouette score. This metric evaluates how closely related an object is to its own cluster in contrast to other clusters. With a scoring range from -1 to 1, a high silhouette score signifies that the points are well-clustered and distinct from points in other clusters, indicating a successful dimensionality reduction.
- **Comparison with t-SNE**: This comparative analysis helped to ascertain the effectiveness of UMAP in capturing the intrinsic structure of the dataset. An improved clustering result or more meaningful data representation compared to other methods would suggest that UMAP is more suitable for the given dataset and task.

t-SNE

Model

t-Distributed Stochastic Neighbor Embedding (t-SNE) is a sophisticated machine learning algorithm used primarily for visualizing high-dimensional data in a low-dimensional space (usually two or three dimensions). It's particularly popular for its ability to reveal the structure and patterns within complex datasets. The t-SNE algorithm function in the following way :

High-Dimensional Similarity:

- t-SNE starts by calculating the pairwise similarities between all points in the high-dimensional space. This is done by measuring the Euclidean distance between points and converting these distances into probabilities that represent similarities.
- The similarity of datapoint x_j to datapoint x_i is represented as a conditional probability $p_{j|i}$, which is high when x_j is close to x_i. This is computed using a Gaussian distribution centered on x_i.

Low-Dimensional Counterpart:

- t-SNE then creates a low-dimensional space (usually 2D or 3D) to represent the data. Initially, it places the low-dimensional points (representations of the high-dimensional points) randomly.
- In this space, the similarity between points y_i and y_j (the low-dimensional counterparts of x_i and x_j) is also calculated using probabilities, but with a t-distribution (hence the "t" in t-SNE), which has heavier tails than a Gaussian distribution.

Kullback-Leibler Divergence Minimization:

- The main objective of t-SNE is to minimize the difference between the two probability distributions: the high-dimensional distribution (in the original space) and the low-dimensional distribution (in the reduced space).
- This difference is measured using the Kullback-Leibler (KL) divergence, a method of measuring how one probability distribution diverges from a second, expected probability distribution.
- t-SNE iteratively adjusts the positions of the points in the low-dimensional space to minimize the KL divergence.

Focus on Local Structure:
t-SNE places a higher emphasis on local structures, trying to ensure that if points are close in the high-dimensional space, they should be close in the low-dimensional space. However, it's less concerned about accurately preserving distances between widely separated points.

Perplexity Parameter:
A key parameter in t-SNE is perplexity, which can be loosely thought of as the number of effective nearest neighbors. It affects the balance between attention to local versus global aspects of the data. Choosing the right perplexity is crucial for revealing the intrinsic structure of the data.

Random Initialization and Non-Convexity:
The random initialization of points in the low-dimensional space means that different runs of t-SNE on the same data can yield different results. This is because the KL divergence minimization problem is non-convex.

t-SNE is particularly powerful for visualizing clusters or groups in data, and for revealing the local structure of the data. However, the distances between clusters in the t-SNE plot are not necessarily meaningful, and the algorithm's reliance on random initialization can lead to different results on different runs. Despite these limitations, t-SNE is widely used for exploratory data analysis, especially in fields like bioinformatics, machine learning, and social network analysis.

Synergy with PCA

- **Initial Dimensionality Reduction with PCA:** PCA is first applied to reduce the number of dimensions to a more computationally manageable size. This is especially helpful when the original dataset has hundreds or thousands of dimensions, which would make t-SNE computationally expensive and time-consuming.
- **Noise Reduction and Variance Preservation:** PCA helps to remove noise by emphasizing the directions where variance is highest, which can be beneficial before t-SNE is applied, as t-SNE then focuses on the most informative features.
- **Speeding Up t-SNE:** By reducing the dimensions beforehand, PCA can significantly speed up t-SNE, which is slower with the increasing number of dimensions due to its complexity.
- **Enhanced t-SNE Performance:** The use of PCA before t-SNE can also help to avoid some of the pitfalls of t-SNE, like the tendency to form clusters even when they might not be very distinct. Starting with a cleaner, lower-dimension dataset can lead to better t-SNE performance.
- **Balancing Global and Local Structure:** While PCA helps to maintain the global structure of the data by preserving the components with the largest variance, t-SNE complements this by preserving the local structures, revealing clusters and relationships that might not be apparent with PCA alone.
- **Application in Large Datasets:** In practice, for very large datasets, it might be the only feasible way to use t-SNE, as without PCA reduction, the t-SNE step might be too resource-intensive.
- **Improved Visualization:** The combination allows practitioners to visualize datasets in a way that both the broad trends and the fine-grained structure are revealed, which can be critical for tasks such as identifying subgroups within data or understanding complex datasets with many variables.

Hyper parameter selection

There are two hyperparameter in the t-SNE algorithm :

- **Perplexity:** Perplexity is a key hyperparameter in the t-SNE algorithm, which roughly indicates the number of effective nearest neighbors. It has a direct impact on the cost function that t-SNE optimizes during the embedding process. In this study, we selected three values of perplexity: 30, 50, and 100. The choice of these values is guided by the typical size of the dataset and previous empirical research suggesting that a perplexity range of 5 to 50 is often sufficient for many datasets. However, we extend this range to 100 to explore its influence on capturing the global data structure. Lower perplexity tends to focus more on local structure, making it suitable for emphasizing cluster separation. In contrast, higher perplexity values facilitate the embedding of broader data relationships, potentially enhancing the visualization of global data distribution.
- **Learning Rate:** The learning rate in t-SNE controls the step size during optimization. Too small a learning rate can result in a practically infinite number of iterations before convergence, whereas too large a learning rate can cause the algorithm to overshoot minima and fail to converge. We selected learning rate values of 200, 500, and 1000 after preliminary experiments which indicated that values below this range led to suboptimal embeddings and values significantly higher resulted in erratic convergence behavior. The chosen range is intended to provide a balance, allowing for adequate adjustments to the point positions while maintaining stable convergence.

Evaluation Metric:
The silhouette score serves as our primary evaluation metric for the quality of the t-SNE embeddings. This metric provides a clear measure of cluster tightness and separation, with a higher score indicating better-defined clusters. This choice is underpinned by the silhouette score's sensitivity to both cohesion within clusters and separation between them, which aligns with the goal of t-SNE to reveal structure in high-dimensional data. The score is particularly suitable for our study as it does not require ground truth labels and solely relies on the distance matrix, which t-SNE optimizes.

Grid Search:
To systematically explore the hyperparameter space, we employed a grid search strategy. This methodical approach entails evaluating all possible combinations of the selected perplexity and learning rate values. Grid search is exhaustive and guarantees that the optimal combination within the defined grid will be found. This is crucial for our study, as t-SNE's cost function is non-convex, and the embedding quality can be sensitive to hyperparameter choices.

Optimization:
The optimization process involves running the t-SNE algorithm with each hyperparameter combination and calculating the corresponding silhouette score. We record these scores in a structured manner, storing them alongside their hyperparameters. The highest silhouette score observed dictates the selection of the best hyperparameter set. By comparing scores, we can draw conclusions about the relative performance of different hyperparameter combinations and ensure that the chosen set of hyperparameters yields the most coherent low-dimensional representation of our data.

Evaluation steps

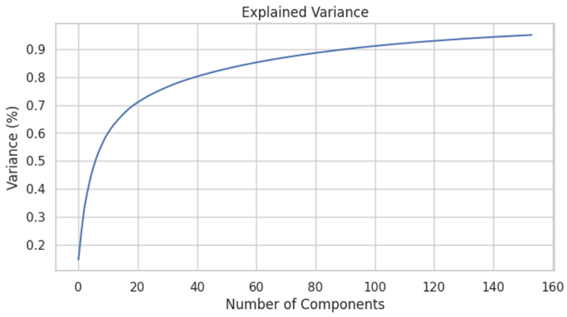
Evaluating the quality of a t-SNE algorithm's output can be somewhat subjective because t-SNE is mainly used for visualization purposes. However, there are a few quantitative and qualitative methods that you can use to assess its performance. Here are the one we used :

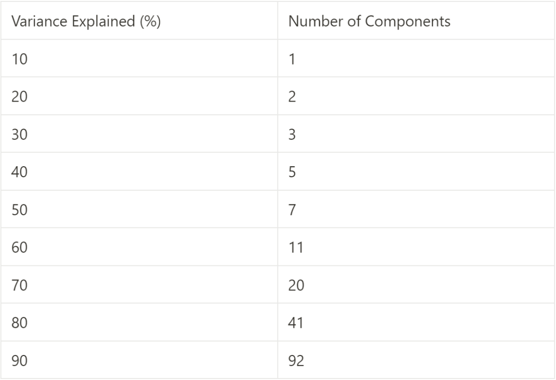
- **Visual Inspection**: By looking at the scatter plot produced by t-SNE, we assessed whether the algorithm has created well-defined, separated clusters.
- **Silhouette Score**: we used silhouette score as a quantitative measure used to evaluate the quality of clusters formed by the algorithm. The silhouette score measures how similar an object is to its own cluster compared to other clusters. The silhouette score ranges from -1 to 1, where a high score indicates that the objects are well matched to their own cluster and poorly matched to neighboring clusters.
- **Comparison with UMAP**: We will compare t-SNE's output to UMAP output to assess whether t-SNE provides a better representation for your specific dataset and task.

Results

PCA

This plot shows the number of principal component necessary to maintain the corresponding total variance threshold.

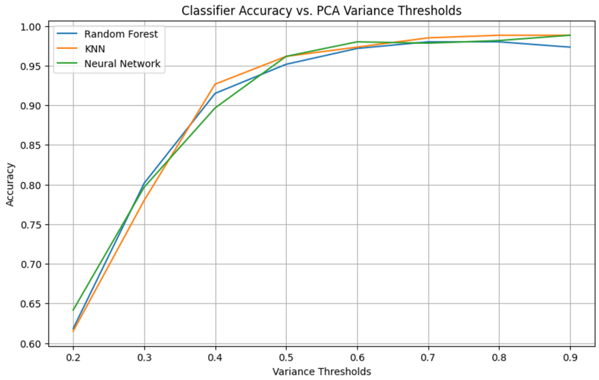




Variance Explained (%)	Number of Components
10	1
20	2
30	3
40	5
50	7
60	11
70	20
80	41
90	92

From the plot, it is evident that as we increase the number of components, the explained variance increases, which is typical in PCA. There is a point of diminishing returns where adding more components doesn't significantly increase the explained variance. The table gives a numerical representation of the information in the plot, specifying how many components are needed to explain a certain percentage of the variance.

We now determine the best threshold value that need the minimum number of component while having the highest accuracy.



All three classifiers show a general increase in accuracy as the variance threshold increases from 0.2 to 0.7. This suggests that retaining more variance (i.e., more principal components) generally leads to better classification accuracy, likely because more relevant information is preserved. With the Kaiser rules mentioned in the PCA part of "Experiments", we arrive to the conclusion that following a 70% variance threshold is more than sufficient for the rest of the experiments. The classifiers also appear to reach peak performance at a variance threshold of 0.7. Beyond this point, the accuracies remains relatively stable. KNN appears to be the more stable of the three classifiers across all the level of variance, so we will use 70% of the total variance as a threshold with 20 principals component and keep the KNN model to do further research on PCA explainability.

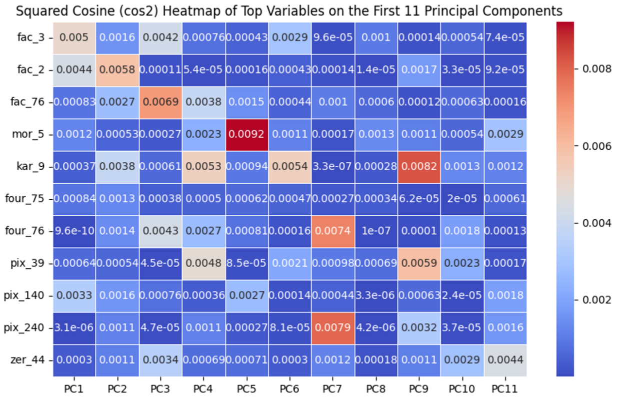
Component analysis

On the test dataset, the number of component that maintain 70% of variance is 20. Let's dive into the exploration of these components. Here is the 10 first (for space purpose).

Component	Explained Variance	Cumulative Explained Variance
1	14.63%	14.63%
2	10.12%	24.75%
3	8.00%	32.75%
4	5.57%	38.32%
5	4.95%	43.27%
6	3.79%	47.06%
7	3.46%	50.52%
8	3.16%	53.68%
9	2.38%	56.06%
10	2.23%	58.30%

The first few components capture a substantial amount of the information in the dataset. The first component is especially significant, indicating that there may be one underlying factor or combination of features that accounts for a large part of the variability in the dataset. As we move to higher components, the individual explained variance decreases, which is typical in PCA since components are ordered by the amount of variance they explain.

Heatmap



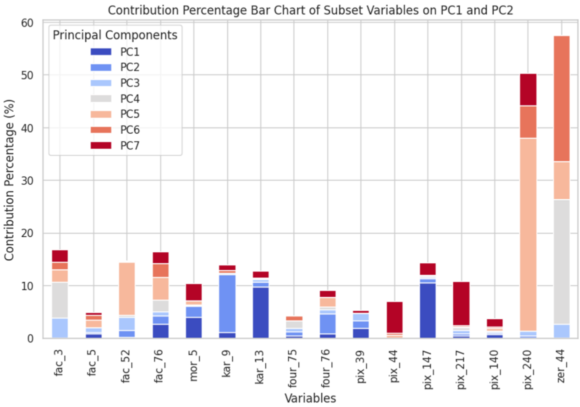
This is a visualization of the squared cosine (cos2) values of the top variables on the first eleven principal components of a PCA model (for readability purpose). In the context of PCA explainability, this graph serves several purposes:

- **Variable Contribution**: It helps to identify how much each variable contributes to each principal component. The cos2 value is a measure of the quality of the representation of a variable on a principal component. A higher cos2 value for a specific component indicates that the variable is well represented by that component.
- **Variable Importance**: By highlighting variables with higher cos2 values, the heatmap can point out which variables are the most important in terms of explaining the variance captured by the principal components.
- **Component Interpretation**: It aids in interpreting the principal components themselves. Since each component is a linear combination of the original variables, understanding which variables contribute most can give insights into the 'meaning' or 'concept' behind each component.

Interpreting the heatmap:

- The color scale on the right represents the magnitude of the cos2 values, ranging from -0.008 (blue) to 0.008 (red).
- Each row represents a different variable (like `fac_3`, `fac_2`, etc.), and each column represents a principal component (`PC1` to `PC11`).
- The intensity of the color corresponds to the value of the cos2: darker red means a higher positive value, and darker blue means a higher negative value (which is less common in cos2 values and might indicate an issue with the data or computation).
- Values closer to zero (white color) suggest that the variable has a low contribution to that principal component.
- For example, variables such as `mor_5` and `four_76` have cells with a noticeably darker red color in the columns for `PC7` and `PC9` respectively, which means they are more important in those principal components.

Contribution of each original variable to the selected principal components.

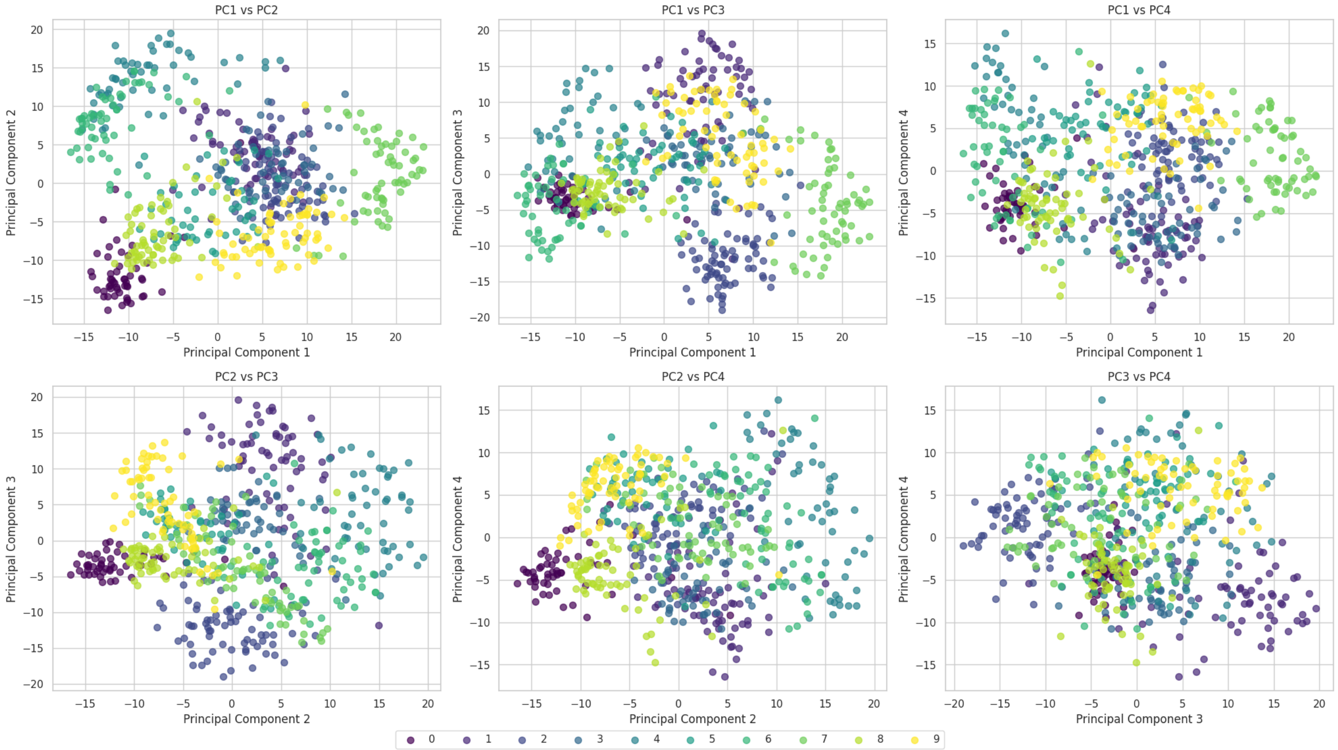


This graph visualizes the contribution of each original variable to the selected principal components. The height of the bars indicates how much each variable contributes to the variance explained by each principal component. Taller bars mean the variable has a stronger relationship with the component. PCA is a dimensionality reduction technique that often produces components that are difficult to interpret. This type of chart helps in understanding what the abstract components represent in terms of original variables. Since each principal component is a combination of original variables based on their correlations, this graph can give insights into the underlying correlation structure of the data.

From the graph, we can interpret the following:

- **Dominant Variables**: Some variables like 'pix_2_40' and 'zei_4' have a significant contribution to certain components (PC6 and PC7 respectively), suggesting that these variables are dominant in explaining the variance along these components.
- **Subtle Contributions**: Smaller contributions across multiple components, like those seen for 'fac_3' and 'fac_5', suggest these variables have a more distributed impact on the data structure.
- **Complexity of Components**: The presence of multiple colors (representing different PCs) in a single variable's bar indicates that the variable is significant in explaining the variance across multiple components, which suggests a complex dataset where variables interact in multiple dimensions.

Relationships between the first four principal components of the dataset



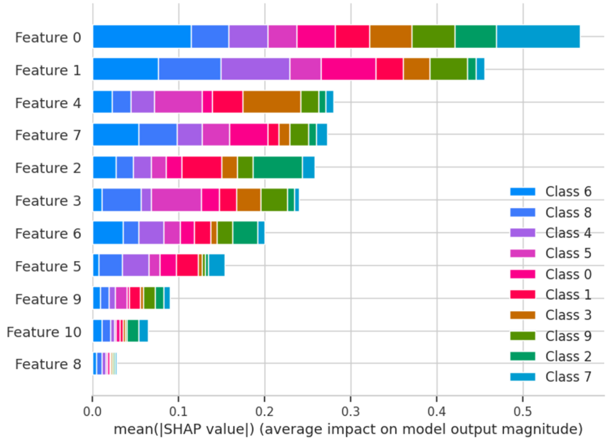
This is a set of scatter plots showing the relationships between the first four principal components of the dataset. This kind of visualization is useful in the context of PCA explainability for several reasons:

- **Data Visualization**: By plotting these principal components against each other, we can visually assess how well-separated different groups or clusters are in the reduced-dimensional space.
- **Interpretability**: Each principal component is a combination of the original variables, and these plots help in understanding the structure of the data. If a clear separation is visible between clusters in certain principal component plots, it suggests that these components capture significant variance and potentially meaningful patterns.
- **Anomaly Detection**: These plots can also be used to identify outliers or anomalies which might appear as points far away from the rest of the data.

From the graph, if we observe that:

- **PC1 vs PC2**: Shows distinct groups, indicating that these components capture a significant amount of the variance in the data.
- **PC1 vs PC3 and PC1 vs PC4**: If these show less clear separation, it might suggest that the first component is the most significant in terms of variance captured, and PC3 and PC4 are progressively less informative.
- **PC2 vs PC3 and PC2 vs PC4**: Help to understand the additional variance captured by PC2 and how it relates to PC3 and PC4.
- **PC3 vs PC4**: If there is no clear pattern or separation, it indicate that these components do not capture distinct variance that separates the data into clear clusters, and thus may not be as useful for some types of analysis.

Usefulness of the components in term of KNN Model : Interpretation of the SHAP Summary Plot



The SHAP summary plotshows the importance and impact of the first 11 principal component on the model's output. Here's how to interpret the graph:

- **Principal Components as Features**: Each row labeled "Feature X" corresponds to a principal component (PC) from the PCA. PCs are linear combinations of the original features, ordered by the amount of variance they capture from the dataset, with PC1 (Feature 0 in the plot) capturing the most variance.
- **SHAP Values**: The length of the bars represents the average magnitude of the SHAP values for each principal component across all predictions. This indicates the average impact of a change in the component value on the model's output.
- **Contribution to Classes**: The colors represent different classes that the model is predicting. The segments of each bar show how much each principal component contributes to the model's output for each class. For instance, the long blue segment in the bar for PC1 (Feature 0) indicates that PC1 has a strong positive impact on the likelihood of predicting Class 6.
- **Feature Importance**: The vertical arrangement of the features (PCs) is typically sorted by the sum of the SHAP value magnitudes over all samples, with the most impactful feature (PC) at the top. In this plot, Feature 0 (PC1) is the most impactful, suggesting it's the most important in terms of the model's decision-making process.
- **Impact Direction**: While traditional SHAP summary plots use color to indicate the direction of the feature's impact (e.g., high or low), in this context, it's more about how each PC contributes to each class, rather than a direction.

Interpretation of the graph:

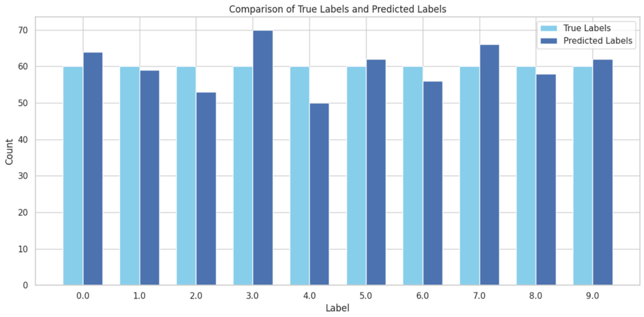
- **Feature 0 (PC1)** has the highest overall impact on the model's output, with significant contributions to all classes, but most notably to Classes 6, 8, and 4.
- **Feature 1 (PC2)** and the other PCs have varying impacts across different classes, with none being as dominant as PC1.
- **Lower-ranked PCs** such as Features 8 and 10 (PC9 and PC11) have much smaller impacts, indicating they might capture more nuanced patterns in the data or may be less informative.

The graph suggests that the first few PCs are likely capturing the most significant patterns in the dataset that are useful for classification, while the latter PCs may be capturing more subtle effects or noise.

Utilisation of K-means algorithm to test PCA performance without target

Using K-means clustering after Principal Component Analysis (PCA) is a common approach in data analysis for verifying the structure in a dataset. K-means clustering is a method of vector quantization, originally from signal processing, that aims to partition n observations into k clusters in which each observation belongs to the cluster with the nearest mean (cluster centers or cluster centroid), serving as a prototype of the cluster.

The rationale is that the combination of PCA and K-means helps in improving the performance of clustering in terms of computational efficiency and the clarity of the clusters formed. PCA reduces the dimensionality, thus reducing the computational overhead, and it can also help in visualizing the data in 2D or 3D plots. The reduced dimensions retain most of the important information (variance) of the data.



For most classes, the counts of true and predicted labels are close, indicating good performance by the clustering algorithm in terms of aligning the predicted clusters with the actual labels. There are some discrepancies in certain classes (for instance, labels 3 and 7) where the predicted counts are noticeably different from the true counts. This suggests some misclassification by the clustering algorithm for these particular classes. Overall, the visual comparison suggests that the K-means algorithm did fairly well in clustering the data in a way that resembles the true label distribution, although it wasn't perfect.

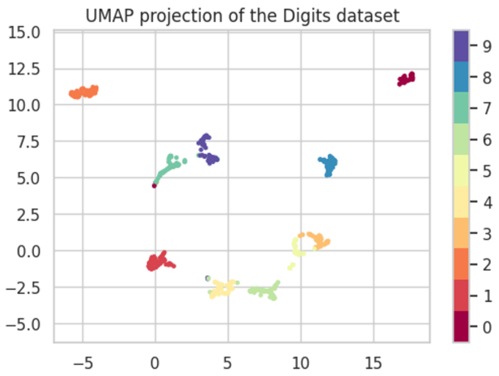
The Silhouette Score is 0.24. This suggests that clusters are not very distinctly defined but there is some structure to the data that the model has captured. There's room for improvement, possibly by feature engineering, hyperparameter tuning, or trying different clustering algorithms.

The Adjusted Rand Index (ARI) for the test set is 0.83, indicating a strong agreement between the clustering labels and the true labels. It suggests that the clusters found by K-means correspond well to the true groupings in the data.

UMAP

Number of neighbors	Minimum distance	Silhouette Score
5	0.0	0.583962
5	0.1	0.566967
5	0.5	0.464802
15	**0.0**	**0.681027**
15	0.1	0.653058
15	0.5	0.535459
30	0.0	0.657544
30	0.1	0.628478
30	0.5	0.510728

The best parameters that led to this UMAP projection were `n_neighbors` set to 15 and `min_dist` (minimum distance) set to 0.0, which achieved the highest silhouette score of approximately 0.681. These parameters control how UMAP balances local versus global structure in the data, with `n_neighbors` influencing how many nearby points each point is compared to and `min_dist` determining how tightly UMAP is allowed to pack points together. The high silhouette score indicates that these parameter settings resulted in a space where clusters are distinct and well-formed, which is visually supported by the following graph.



Here’s an interpretation of the graph:

- **Clusters**: Each cluster in the graph represents data points that are similar to each other. The fact that the clusters are distinct and mostly non-overlapping suggests that the dataset contains well-defined groups. In the context of a digits dataset, each cluster likely represents a different digit from 0 to 9.

- **Color Coding**: The colors correspond to the actual digit labels, ranging from 0 to 9. The color bar on the right side serves as a legend to help identify which cluster corresponds to which digit.
- **Spatial Distribution**: The relative positioning of the clusters in the 2D plane gives us an idea of the similarities between different digits. For example, if two clusters are closer to each other, it suggests that the corresponding digits share more similarities in their high-dimensional feature space, such as 3 and 5, or 4 and 6.
- **Outliers**: There may be some outliers or points that seem to be positioned away from their main cluster. These could be data points that are not easily classified into any single digit, potentially representing images that are not clear or are ambiguous.

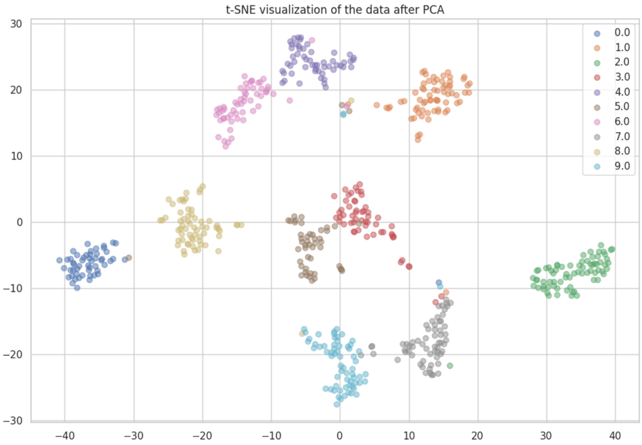
Context of PCA Explainability:
In this context, the output of PCA served as the input to UMAP, which further reduced the dimensionality to two dimensions for visualization. The combination of PCA and UMAP helps in achieving a balance between preserving global structure (via PCA) and local neighborhood relations (via UMAP), leading to a potentially more informative and interpretable visualization.

The graph demonstrates that even after PCA has compressed the information, UMAP is able to find a representation of the data where the inherent groupings are visually clear, which is particularly useful for tasks like digit classification where we aim to distinguish between different classes based on their feature similarities.

t-SNE

Perplexity	Learning Rate	Silhouette Score
30.0	**200.0**	**0.589758**
30.0	1000.0	0.572907
30.0	500.0	0.567397
50.0	500.0	0.535589
50.0	1000.0	0.534740
50.0	200.0	0.533718
5.0	500.0	0.487523
5.0	1000.0	0.486387
5.0	200.0	0.471593

The best score from the t-SNE hyperparameter tuning suggests that with a perplexity of 30 and a learning rate of 200, the two-dimensional representation has a reasonably good silhouette score of approximately 0.59, indicating that the classes are well-separated on average. This is a strong indication that the dataset contains distinct groups that could potentially be classified accurately, as shown on the graph below.



This is a t-SNE visualization of high-dimensional data that has been reduced to two dimensions following PCA dimensionality reduction. Here's an interpretation of the graph:

- **Clusters**: Each color represents a different cluster or class from the dataset, with 10 unique classes (0 through 9), which indicate the 10 different digits. The clusters are mostly separate from each other, which suggests that the classes are fairly distinct in the high-dimensional space.
- **Cluster Spread**: Some clusters are more spread out (such as 0 and 3), while others are tighter (such as 1 and 4). A spread-out cluster could indicate higher intra-class variability, meaning samples within the same class can be quite different from

each other.
- **Overlaps**: There are some areas where different clusters overlap (such as between 4 and 9 or 3 and 5). Overlaps could mean that some of the classes are not perfectly distinguishable from each other in the reduced space, which could be due to inherent similarities in the data or a limitation of the dimensionality reduction technique.
- **Outliers**: There are points that are far from the center of their respective clusters. These could be outliers within their classes, or they could be points that were not well represented in the reduced space.

In the context of PCA explainability:

- PCA serves to simplify the high-dimensional space before t-SNE is applied, which can make t-SNE more efficient and sometimes more effective at finding a good low-dimensional representation. The resulting two-dimensional space from t-SNE is easier to visualize and interpret, but it is worth noting that it may not always perfectly reflect the true relationships in the high-dimensional space, particularly if PCA has discarded dimensions that contained information about the data's structure.

The graph shows that, despite the information compression by PCA, t-SNE manages to uncover a representation of the data where the intrinsic groupings are distinctly visible. This is especially beneficial for tasks such as digit classification, where the goal is to differentiate between various classes based on the similarities of their features.

UMAP vs t-SNE

The t-SNE and UMAP projections, following PCA dimensionality reduction, offer contrasting insights into the dataset's structure. With t-SNE, utilizing a perplexity of 30.0 and a learning rate of 200.0, the visualization showcases discernible but intermixed clusters, with a silhouette score of 0.589758. This indicates that t-SNE effectively captures local structures within the data, as evidenced by the separation of clusters, albeit with some overlaps where different classes intersect.

In contrast, UMAP, configured with 15 neighbors and a minimum distance of 0.0, achieves a silhouette score of 0.681027, surpassing that of t-SNE. The clusters in the UMAP visualization are more compact and exhibit less interclass mixing, suggesting a stronger preservation of both local and global structures within the data. This is visually apparent in the tighter congregations of data points, indicating a clearer delineation between distinct classes.

When directly compared, UMAP appears to provide a more defined and interpretable representation of the dataset's inherent groupings, which can be particularly beneficial for classification tasks. Its higher silhouette score implies that it might be more effective in contexts where understanding the global relationship between classes is just as important as the local one.

In summary, while t-SNE offers a valuable tool for visualizing local structures and patterns within the data, UMAP's ability to maintain a balance between local and global features makes it a potentially more powerful tool for enhancing the interpretability of PCA-reduced datasets in high-dimensional data analysis.

Conclusion

This paper presented an in-depth exploration into the enhancement of Principal Component Analysis (PCA) interpretability in high-dimensional data analysis through the integration of explainability tools and dimensionality reduction techniques. Our research centered on the premise that while PCA is a powerful tool for dimensionality reduction, its application in high-dimensional contexts often suffers from interpretability challenges. To address this, we investigated how combining PCA with advanced explainability tools can make its outcomes more comprehensible, especially in complex data environments.

Key highlights of our work include:

- **Detailed Analysis of PCA in High-Dimensional Contexts**: We conducted a thorough examination of PCA's strengths and limitations when applied to high-dimensional datasets, underscoring the need for enhanced interpretability in such scenarios.
- **Innovative Integration of Explainability Tools**: Our research pioneered the integration of specific explainability tools with PCA. These tools were selected and optimized to demystify PCA's often opaque transformation process, thereby making the results more accessible to users.
- **Evaluation of Dimensionality Reduction Techniques**: We evaluated several dimensionality reduction techniques alongside PCA. This comparison was aimed at understanding their individual and combined impacts on the interpretability of the PCA process.
- **Empirical Testing and Results**: Through empirical testing on diverse datasets, we demonstrated that the integration of these tools and techniques significantly improves the interpretability of PCA. This was particularly evident in cases involving complex, high-dimensional datasets where traditional PCA application would typically fall short in terms of user understanding.

While our research offers significant advancements in enhancing the interpretability of PCA in high-dimensional data analysis, it is essential to acknowledge certain limitations inherent in our approach. Recognizing these constraints not only demonstrates transparency but also provides a foundation for future research to build upon and improve.

- **Limitations of the Kaiser Rule**: One of the primary methodologies we employed in determining the number of principal components was the Kaiser rule. While widely used, this rule can sometimes lead to overestimation, especially in datasets with a large number of variables or those not normally distributed. This overestimation can affect the accuracy of dimensionality reduction and, consequently, the interpretability of the results.
- **Scope of Data Types and Structures**: Our research predominantly focused on specific types of high-dimensional datasets. Therefore, the findings and the effectiveness of the integrated tools may vary when applied to data types or structures not extensively covered in our study. This includes datasets with different distributions, sizes, or intrinsic complexities.
- **Generalizability of Findings**: The generalizability of our results to all forms of high-dimensional data analysis may be limited. The integration of explainability tools and dimensionality reduction techniques as explored in this research was tailored to specific scenarios and might not yield the same level of interpretability enhancement in other contexts.
- **Dependence on External Tools**: Our approach heavily relies on external explainability tools, whose own limitations and biases could indirectly influence the interpretability of PCA. The effectiveness and accuracy of these tools are crucial factors that directly impact our methodology.
- **Computational Complexity and Efficiency**: The integration of additional tools and techniques into the PCA process inherently increases computational complexity. This might pose challenges in terms of efficiency and practicality, especially when dealing with extremely large datasets or in situations where computational resources are limited.

Building on the findings and limitations of our current research, several avenues for future research emerge. These suggestions aim to further enhance the interpretability of PCA in high-dimensional data analysis and address the identified constraints:

- **Broader Data Type and Structure Applicability**: Future research should extend the application of PCA and its interpretability tools to a wider range of data types and structures. This would help in understanding the effectiveness of our approach across various domains, including those with irregular distributions, extreme sizes, or unique complexities.
- **Enhancing Generalizability**: Investigating the integration of explainability tools and dimensionality reduction techniques in other data analysis contexts is crucial. This would help in assessing the generalizability of our approach and in identifying specific scenarios where it is most effective.
- **Quantitative Metrics for Interpretability**: Developing standardized, quantitative metrics to measure the interpretability of PCA results would be a significant advancement. This would provide a more objective and consistent way to assess and compare the effectiveness of various interpretability-enhancing techniques.
- **User-Centric Studies**: Conducting user-centric studies to evaluate the practical usability and understanding of PCA outputs when enhanced by explainability tools could provide valuable insights. This would help in tailoring approaches to the actual needs and comprehension levels of different user groups.
- **Integration with Machine Learning Models**: Exploring the integration of PCA and its interpretability enhancements within machine learning models, especially in predictive analytics, could open new research pathways. This would contribute to the development of more transparent, interpretable, and reliable predictive models in various fields.

As we conclude, it is imperative to reflect on the broader implications and future potential of our research in the realm of high-dimensional data analysis. The journey of our research underscores a fundamental truth in the field of data science: the power of a data analysis tool is not just in its computational prowess, but equally in its accessibility and understandability to users. By making PCA more interpretable, we bridge a critical gap between advanced statistical methods and their practical, real-world application. This alignment is especially crucial in an era where data-driven decision-making is paramount across various sectors, including healthcare, finance, and technology.

Furthermore, our research highlights the growing importance of explainability in the field of data analytics. As data becomes more complex and models more sophisticated, the need for transparency and comprehensibility in these models' outputs becomes increasingly critical. Our work is a step towards a future where data science is not just about the extraction of insights but also about the clear communication of these insights to users of diverse backgrounds.

In closing, we are reminded of the words of renowned computer scientist Edsger W. Dijkstra: "Simplicity is prerequisite for reliability." In enhancing the interpretability of PCA, we contribute not only to its simplicity but also to its reliability and trustworthiness in various applications. This is particularly crucial in an era where data-driven decisions have far-reaching consequences in society.

References

- Google collab notebook

- Presentation
- Explainable Dimensionality Reduction (XDR) to Unbox AI 'Black Box' Models: A Study of AI Perspectives on the Ethnic Styles of Village Dwellings
- Improving Statistical Reporting Data Explainability via Principal Component Analysis
- SVD to PCA: Technique to Improve XAI (Explainable AI) (Part 2)
- Explainability in Machine Learning: a Pedagogical Perspective
- P.C. Besse, PCA stability and choice of dimensionality, Statistics & Probability Letters 13 (1992), 405–410. [2] I. Jolliffe, Principal Component Analysis, 2nd edition éd., Springer-Verlag, 2002.
- Bio-image Analysis Notebooks. (n.d.). Clustering with UMAPs
- OpenAI. (2023). ChatGPT interaction on november 2023
- UMAP 0.5 documentation. (n.d.). UMAP: Uniform Manifold Approximation and Projection for Dimension Reduction
- Wikipedia contributors. (2023, November 30). T-distributed stochastic neighbor embedding. In Wikipedia, The Free Encyclopedia
- Scikit-learn developers. (2007-2023). Sklearn.manifold.TSNE. In scikit-learn 1.3.2 documentation