HELLENIC REPUBLIC
UNIVERSITY OF MACEDONIA

MSc in Applied Informatics

Student name: Tserga Georgia

Project 2 (Clustering)

Introduction

The purpose of the report is to evaluate some clustering techniques on images, using different dimensionality reduction models, and determine the best combination based on specific performance metrics.

Clustering is a type of unsupervised learning that involves grouping a set of objects in such a way that objects in the same group (called a cluster) are more similar to each other than to those in other groups (clusters).

The data used is from the fashion-mnist dataset, consisting of 70,000 grayscale images in 10 categories. You can find more information at the following link: https://keras.io/api/datasets/fashion_mnist/

Data Preprocessing

Before applying the clustering algorithms, the images are flattened and normalized. **Principal Component Analysis (PCA)** is used for dimensionality reduction to improve clustering performance and reduce computational cost. This technique transforms a large set of variables into a smaller one that still contains most of the information in the large set. It does this by identifying the principal components, which are the directions in which the data varies the most.

The shapes of the transformed datasets are printed:

- X_train_pca shape: (60000, 187)
- X_test_pca shape: (10000, 187)

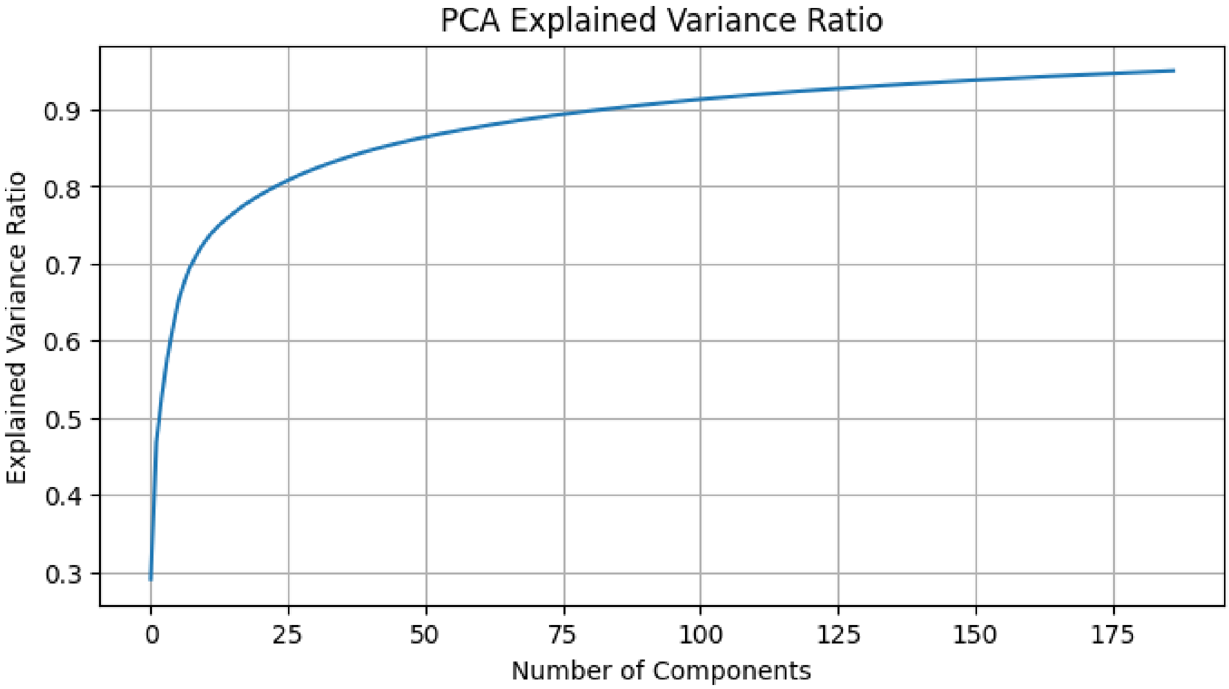
This means that the original 784-dimensional data has been reduced to 187 dimensions while retaining 95% of the variance.

Visualization 1: PCA Explained Variance Ratio

It shows the cumulative explained variance ratio as a function of the number of principal components.

- X-Axis: Number of Components
- Y-Axis: Explained Variance Ratio
- Curve: The plot starts steeply, indicating that the first few components explain a large portion of the variance. As the number of components increases, the curve levels off, indicating diminishing returns in explained variance.

The curve shows that around 187 components are required to explain 95% of the variance in the dataset. This information is used to choose the number of components for PCA to achieve a good balance between dimensionality reduction and retaining variance.



Also, another technique of dimensionality reduction that we used is the **Stacked Autoencoder (SAE)**, which is a type of neural network used for unsupervised learning of efficient codings. It is composed of multiple layers of autoencoders where the output of each layer is wired to the input of the successive layer.

The SAE is constructed with three encoding layers and three decoding layers. The encoding layers reduce the input dimension from 784 to 32 through layers of 128 and 64 neurons. The decoding layers reverse this process.

The autoencoder is trained for 50 epochs with a batch size of 256. The training data is shuffled, and the validation data is also used to monitor the performance during training.

The shapes of the encoded feature sets are printed:

- X_train_sae shape: (60000, 32)
- X_test_sae shape: (10000, 32)

This indicates that the original 784-dimensional data has been reduced to 32 dimensions.

Visualization 2: SAE Training and Validation Loss

It shows the loss values for both training and validation datasets over the course of 50 epochs.

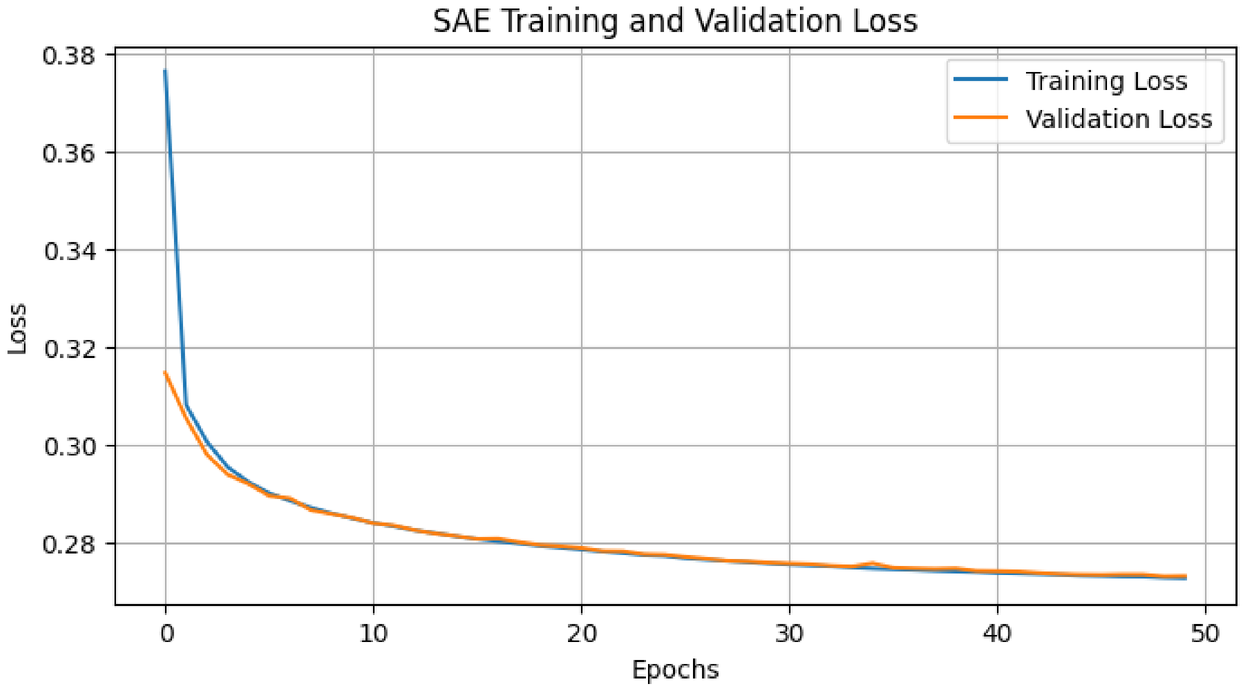
- X-Axis: Epochs
- Y-Axis: Loss

Curves:

- Training Loss: The blue curve represents the training loss.
- Validation Loss: The orange curve represents the validation loss.

Results:

- Initial High Loss: Both training and validation loss start at a relatively high value but quickly decrease.
- Convergence: After a few epochs, both losses converge and continue to decrease gradually, indicating that the model is learning effectively without overfitting.
- Final Loss Values: The final loss values are around 0.273 for both training and validation, showing that the model has achieved a low reconstruction error.

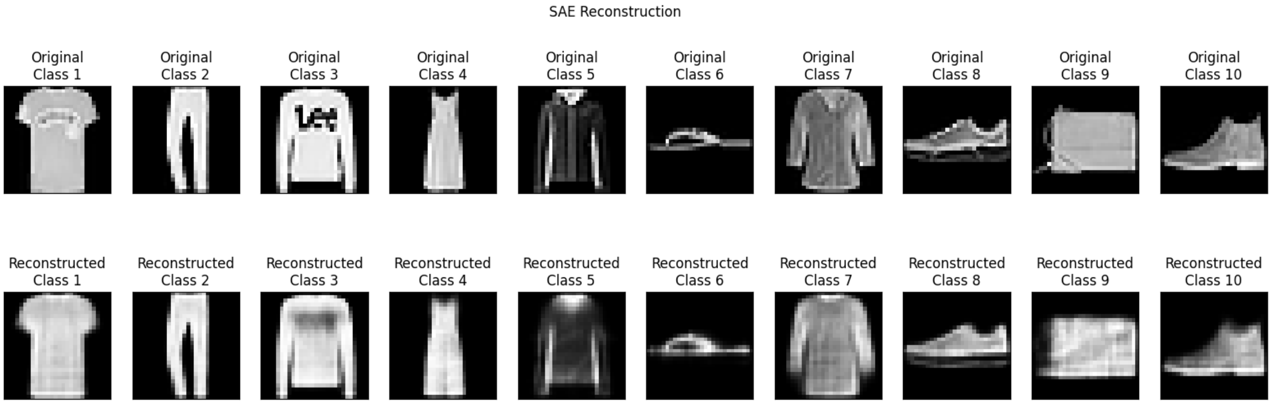


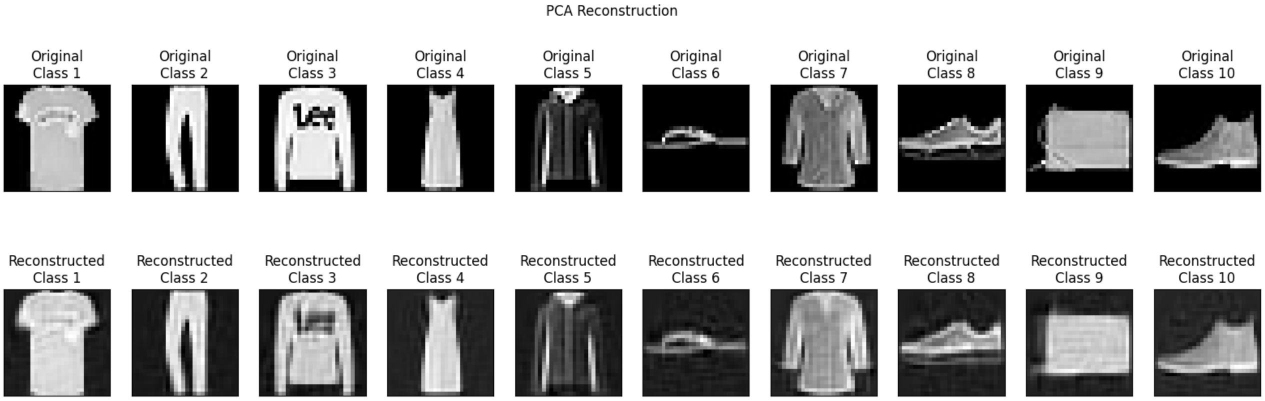
Data Visualization

First, we wanted to visualize one image from each class out of a total of 10, after applying the two dimensionality reduction techniques (SAE & PCA).

- **First Row of Each Section (Original Images)**: Shows the first example of each class from the test dataset. These serve as benchmarks to compare against the reconstructed versions.
- **Second Row of Each Section (Reconstructed Images):** Displays the reconstructed images corresponding to the originals directly above them. This visual layout helps in assessing the quality and accuracy of the reconstruction, as differences and similarities are easily noticeable.

The below images showcase how different each reconstructed image is compared to the original, highlighting the effectiveness and limitations of SAE and PCA in capturing the essential features of the images.





Techniques and Performance metrics Analysis

First of all, let's describe the usability of each Clustering technique and the Performance metrics that we used:

Clustering techniques

1. **Minibatch kmeans** is a variant of the k-means clustering algorithm which uses small, random samples (minibatches) of the dataset to reduce computation time. This makes it more suitable for large-scale data.
2. **K-Means Clustering** is a partitioning method that divides a dataset into K distinct, non-overlapping subsets (clusters). Each cluster is defined by its centroid, which is the mean of the points in that cluster. The algorithm aims to minimize the within-cluster sum of squares (variance).

Performance metrics

1. **Calinski–Harabasz index**: Also known as the Variance Ratio Criterion, is a metric used to evaluate the quality of clustering. It is defined as the ratio of the sum of between-cluster dispersion and within-cluster dispersion for all clusters, with higher values indicating better-defined clusters.
2. **Silhouette Score**: Measures how similar a point is to its own cluster (cohesion) compared to other clusters (separation). It ranges from -1 to 1, where a high value indicates that the object is well-matched to its own cluster and poorly matched to neighboring clusters.
3. **Davies-Bouldin Index**: Measures the average similarity ratio of each cluster with the cluster most similar to it, where similarity is the ratio of within-cluster distances to between-cluster distances. Lower values indicate better clustering performance. Lower values indicate better clustering.

Below is a table with the results after running the code for every Dimensionality Reduction technique, Clustering Algorithm and the corresponding Performance metrics:

Dimensionality Reduction	Clustering Algorithm	Training Time (s)	Execution Time (s)	Number of Clusters	Calinski-Harabasz Index	Davies-Bouldin Index	Silhouette Score
Raw	MiniBatchKMeans	2.241831	0.016117	10	1236.226683	1.988400	0.134635
	KMeans	8.032817	0.029922		1246.284706	2.009225	0.136241
PCA	MiniBatchKMeans	0.211136	0.004002		29.342215	8.456959	-0.065596
	KMeans	4.069616	0.008204		40.156579	7.147615	-0.070387
SAE	MiniBatchKMeans	0.188882	0.001255		2165.710061	1.684853	0.182320
	KMeans	0.281111	0.001494		2250.133470	1.572669	0.200443

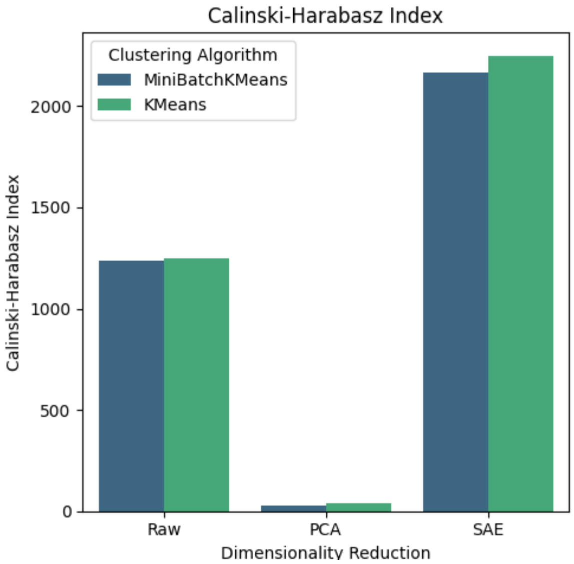
Also, we created the following bar charts to analyze the impact of dimensionality reduction techniques (PCA and SAE) and clustering algorithms (KMeans and MiniBatchKMeans) on clustering performance metrics: training time, execution time, Calinski-Harabasz Index, Davies-Bouldin Index, and Silhouette Score. More specifically:

Training and Execution Time:

- Dimensionality Reduction Impact: Both PCA and SAE significantly reduce the training and execution time compared to using raw data. This is expected as reducing dimensionality generally simplifies the data structure, reducing the computational load.
- Algorithm Efficiency: MiniBatchKMeans consistently shows lower training and execution times across all data types (Raw, PCA, SAE), highlighting its efficiency for larger datasets due to its iterative batch-based approach.

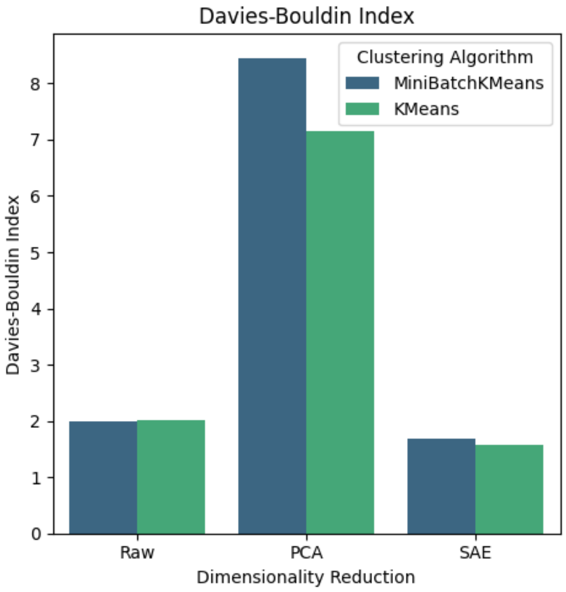
Visualization 1: Calinski-Harabasz Index (CHI)

- This index measures the ratio of the sum of between-clusters dispersion and of within-cluster dispersion for all clusters (higher is better).
- SAE's Superior Performance: SAE with MiniBatchKMeans shows the highest CHI, suggesting that the clusters formed are very compact and well-separated compared to those formed with PCA or raw data.
- PCA's Moderate Performance: Despite PCA's utility in reducing dimensions, its CHI is significantly lower than that of SAE, indicating less distinct clustering.



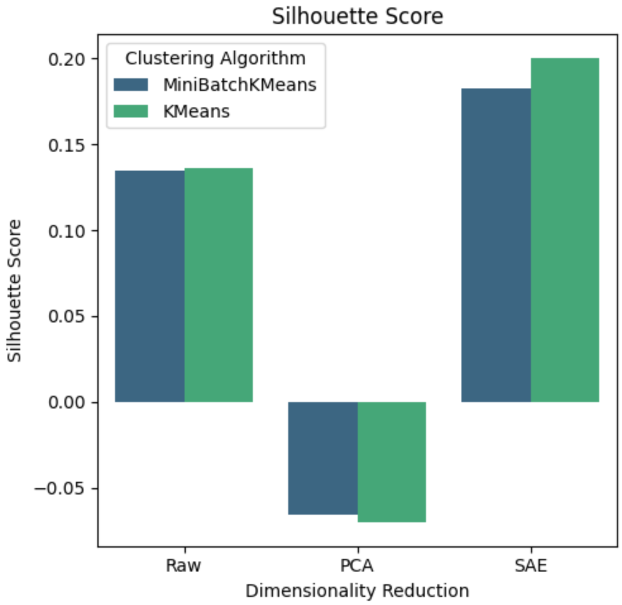
Visualization 2: Davies-Bouldin Index (DBI)

- This index indicates the average 'similarity' between clusters, where lower values signify better clustering.
- SAE's Advantage: SAE results in the lowest DBI, particularly with KMeans, indicating that clusters have less overlap and are more distinct.
- High DBI for PCA: The relatively high DBI for PCA suggests that the clusters are less distinct or have higher variance within them.



Visualization 3: Silhouette Score

- Measures how similar an object is to its own cluster compared to other clusters (range from -1 to 1, higher is better).
- Positive Scores for SAE: Both clustering algorithms yield positive scores with SAE, with KMeans performing slightly better, suggesting good cluster cohesion and separation.
- Negative Scores for PCA: The negative scores for PCA indicate that clusters might be incorrectly assigned or are too overlapping, reducing their distinctiveness.



Summary of Observations:

SAE vs. PCA: SAE consistently outperforms PCA in all metrics, suggesting that for this dataset and the chosen clustering algorithms, SAE provides a better structure and separation of data.

Efficiency of MiniBatchKMeans: MiniBatchKMeans is preferable when considering computational efficiency (time), although KMeans might occasionally provide slightly better cluster quality (as seen in the DBI and Silhouette Score).

Impact of Dimensionality Reduction: Both PCA and SAE improve the performance metrics compared to raw data, emphasizing the utility of dimensionality reduction in clustering tasks.

Comparison and Evaluation of results

Dimensionality Reduction	Clustering Algorithm	Training Time (s)	Execution Time (s)	Number of Clusters	Calinski-Harabasz Index	Davies-Bouldin Index	Silhouette Score
PCA	MiniBatchKMeans	0.211136	0.004002	10	29.342215	8.456959	-0.065596
Raw	KMeans	8.032817	0.029922		1246.284706	2.009225	0.136241
SAE	KMeans	0.281111	0.001494		2250.133470	1.572669	0.200443

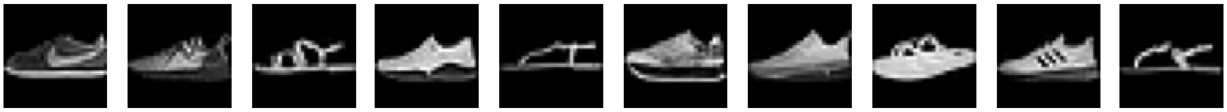
From the table above, it seems that there was a case where one combination achieved the best results in all measurements. More specifically, the combination of **Stacked Autoencoder (SAE)** for dimensionality reduction and **KMeans** for clustering provided the best values in all performance metrics: Calinski-Harabasz Index, Davies-Bouldin Index, and Silhouette Score. This combination achieved:

- The highest **Calinski-Harabasz Index** (2250.13), indicating that the clusters are dense and well separated.
- The lowest **Davies-Bouldin Index** (1.57), showing that the clusters are well separated and the data within each cluster is homogeneous.
- The highest **Silhouette Score** (0.20), indicating that most samples are close to the center of their clusters and the clusters are quite distinct from each other.

This combination proved to be the most efficient for the particular image clustering problem on the fashion-mnist dataset, providing the best quality clusters and the most consistent performances across all metrics.

Finally, we display 10 random images from two random clusters, based on the best clustering and grouping technique. By visually inspecting the images from each selected cluster, we can qualitatively assess the effectiveness of the clustering algorithm. From the following screenshots, it seems that images within the same cluster are similar and share common features.
Also, displaying images helps in understanding what each cluster represents. For example, cluster 4 seems that includes images of shoes and cluster 2 images of t-shirts.

Cluster 4



Cluster 2