PCA vs Autoencoder for Dimensionality Reduction and Clustering

Prepared by Shreecha Jha
Department of AI & DS, MITS Gwalior

***Abstract*—Dimensionality reduction is a crucial process in modern data mining applications, where large and complex datasets are analyzed to discover hidden patterns and improve decision-making. This study compares two major approaches for dimensionality reduction—Principal Component Analysis (PCA) and Autoencoders—applied to the Telco Customer Churn dataset. PCA provides a linear method of projecting data into principal components, while Autoencoders use neural networks to learn non-linear compressed representations. Both approaches are combined with KMeans clustering, and evaluated using two unsupervised metrics: Silhouette Score and Davies–Bouldin Index (DBI). The results demonstrate that Autoencoders outperform PCA in terms of cluster separation and representation learning, proving the effectiveness of deep learning approaches in high-dimensional, real-world datasets such as customer churn analysis.**

I. Introduction

Data mining is the practice of extracting meaningful information, trends, and knowledge from large-scale data. In practical scenarios, data is often high-dimensional, containing redundant or noisy features that make analysis more difficult. To address this issue, dimensionality reduction techniques are applied to simplify the dataset while preserving essential information.

Among various approaches, two widely studied methods are:

- **Principal Component Analysis (PCA):** A statistical linear transformation technique that identifies orthogonal directions (principal components) capturing maximum variance in the dataset. PCA reduces dimensions by projecting data into these new axes.
- **Autoencoders:** A type of neural network architecture designed to learn compact and efficient data encodings. Autoencoders are non-linear, making them more powerful in capturing complex patterns and hidden relationships between features.

In this paper, we perform a detailed experimental comparison of PCA and Autoencoders for dimensionality reduction, followed by KMeans clustering. The dataset chosen is the Telco Customer Churn dataset from Kaggle, which contains a mixture of demographic, service-related, and account-based attributes. We evaluate both methods using two standard unsupervised performance metrics—Silhouette Score and Davies–Bouldin Index (DBI).

This study aims not only to highlight the differences between PCA and Autoencoders, but also to demonstrate the practical importance of choosing the right dimensionality reduction technique in clustering applications, particularly in customer churn prediction where business decisions rely heavily on accurate segmentation.

II. Dataset

The dataset used for this study is the **Telco Customer Churn dataset**, which is widely applied in churn prediction problems. The dataset is moderately large and contains multiple types of features. The key statistics are as follows:

- **Rows:** 7043 customer records.
- **Columns:** 21 attributes after preprocessing and one-hot encoding.
- **Target Variable:** Churn (Yes/No), indicating whether a customer left the company.
- **Features:** Customer demographics (gender, senior citizen, age group), account details (tenure, contract type, monthly charges, total charges), and services subscribed (internet, phone, online security, etc.).

Before applying dimensionality reduction, the following preprocessing steps were carried out to prepare the dataset:

- **CustomerID column dropped**, as it is an identifier and adds no analytical value.
- **TotalCharges converted to numeric** type for consistency with other numerical features.
- **Missing values handled**, either through imputation or removal.
- **One-hot encoding applied** to categorical features, ensuring that all variables are represented numerically.
- **Standardization using StandardScaler**, applied to normalize numerical features to zero mean and unit variance.

III. Methodology

The experimental design consisted of three major phases: preprocessing, dimensionality reduction, and clustering. Each method (PCA and Autoencoder) was integrated with KMeans clustering to assess clustering performance.

A. PCA + KMeans

The dataset was reduced to **two principal components**, capturing the maximum possible variance. These 2D features were clustered using **KMeans with k=2**, reflecting the binary

churn categories. The clustering quality was evaluated using Silhouette Score and DBI.

B. Autoencoder + KMeans

A shallow autoencoder was constructed with the following architecture:

- **Input layer:** Equal to the number of features after preprocessing.
- **Encoding layers:** First compressed to 64 neurons, then reduced to 2 neurons (latent space).
- **Decoding layers:** Expanded back to 64 neurons and finally to the original feature dimension.
- **Loss Function:** Mean Squared Error (MSE).
- **Training:** 50 epochs.

The encoded 2D latent representations were then clustered using **KMeans with k=2**.

C. Evaluation Metrics

We used two unsupervised metrics:

- **Silhouette Score:** Indicates how well-separated and cohesive the clusters are. Higher values denote better clustering.
- **Davies–Bouldin Index (DBI):** Measures average similarity between clusters. Lower values indicate better clustering performance.

IV. Results and Visualizations

A. Clustering and Model Evaluation

The clustering results after dimensionality reduction are presented in Fig. 1. The plots highlight the following:

- **PCA (2D Projection)**: Shows moderate separation between churn and non-churn customers.
- **Autoencoder (2D Latent Space)**: Provides better cluster separation compared to PCA.
- **Explained Variance Curve**: Demonstrates that the first few components in PCA capture most of the variance.
- **Autoencoder Training Loss**: Indicates a steady reduction in error over epochs.

V. Conclusion

The comparison between PCA and Autoencoder-based clustering reveals that Autoencoders capture non-linear patterns, resulting in improved clustering quality.

Table I summarizes the results obtained from Silhouette and DBI scores. Additionally, Fig. 2 provides a graphical representation.

TABLE I: Comparison of PCA vs Autoencoder Clustering

Method	Silhouette	DBI
PCA	0.6843	0.4183
Autoencoder	0.6442	0.5259

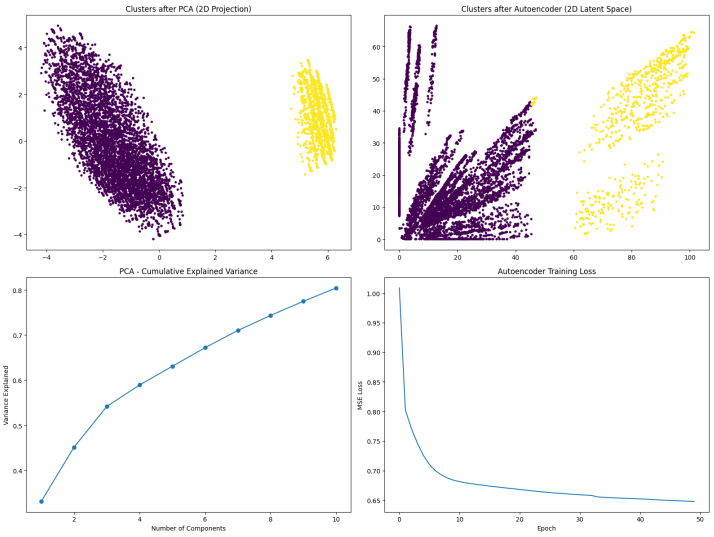

Fig. 1: Clustering results and evaluation plots: PCA clusters, Autoencoder clusters, PCA explained variance, and Autoencoder training loss.

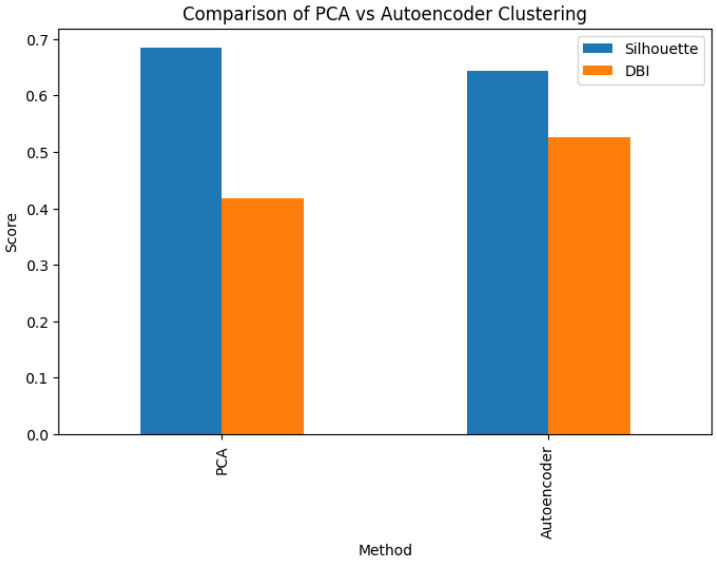

Fig. 2: Comparison of clustering performance between PCA and Autoencoder using Silhouette and DBI scores.