Dimensionality Reduction

Dimensionality reduction, or dimension reduction, is the transformation of data from a high-dimensional space into a low-dimensional space so that the low-dimensional representation retains some meaningful properties of the original data, ideally close to its intrinsic dimension. Working in high-dimensional spaces can be undesirable for many reasons; raw data are often sparse as a consequence of the curse of dimensionality, and analyzing the data is usually computationally intractable (hard to control or deal with). Dimensionality reduction is common in fields that deal with large numbers of observations and/or large numbers of variables, such as signal processing, speech recognition, neuroinformatics, and bioinformatics.

Methods are commonly divided into linear and nonlinear approaches. Approaches can also be divided into feature selection and feature extraction. Dimensionality reduction can be used for noise reduction, data visualization, cluster analysis, or as an intermediate step to facilitate other analyses.

A gentle introduction to dimensionality reduction for machine learning

- Large numbers of input features can cause poor performance for machine learning algorithms.
- Dimensionality reduction is a general field of study concerned with reducing the number of input features.
- Dimensionality reduction methods include feature selection, linear algebra methods, projection methods, and autoencoders.

There are two types of Dimensionality Reduction techniques:

1. Feature Selection
2. Feature Extraction

Feature Selection techniques are Backward Elimination, Forward Selection, Bidirectional Elimination, Score Comparison and more.

The following Feature Extraction techniques:

1. Principal Component Analysis (PCA)
2. Linear Discriminant Analysis (LDA)
3. Kernel PCA

Principal component analysis (PCA)

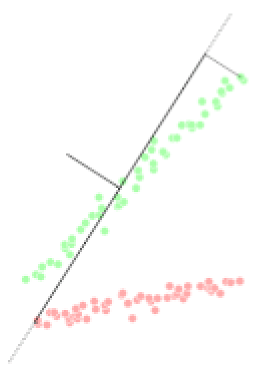
The main linear technique for dimensionality reduction, principal component analysis, performs a linear mapping of the data to a lower-dimensional space in such a way that the variance of the data in the low-dimensional representation is maximized.

Principal component analysis (**PCA**) is a linear dimensionality reduction technique with applications in exploratory data analysis, visualization and data preprocessing.

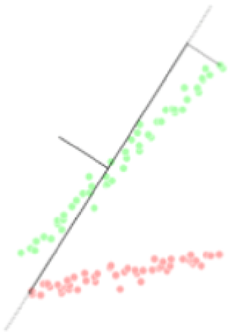
The data is linearly transformed onto a new coordinate system such that the directions (principal components) capturing the largest variation in the data can be easily identified.

The **principal components** of a collection of points in a real coordinate space are a sequence of p unit vectors, where the i-th vector is the direction of a line that best fits the data while being orthogonal to the first i-1 vectors. Here, a best-fitting line is defined as one that minimizes the average squared perpendicular distance from the points to the line. These directions (i.e., principal components) constitute an orthonormal basis in which different individual dimensions of the data are linearly uncorrelated. Many studies use the first two principal components in order to plot the data in two dimensions and to visually identify clusters of closely related data points.

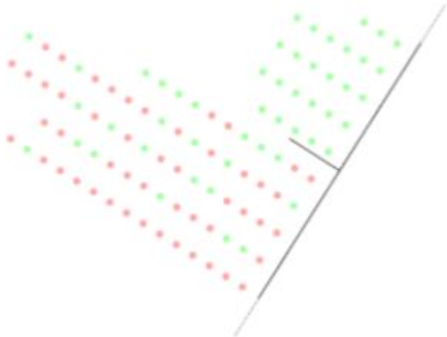
Principal component analysis has applications in many fields such as population genetics, microbiome studies, and atmospheric science.



A visual depiction of the resulting PCA projection for a set of 2D points.



Actual Dataset



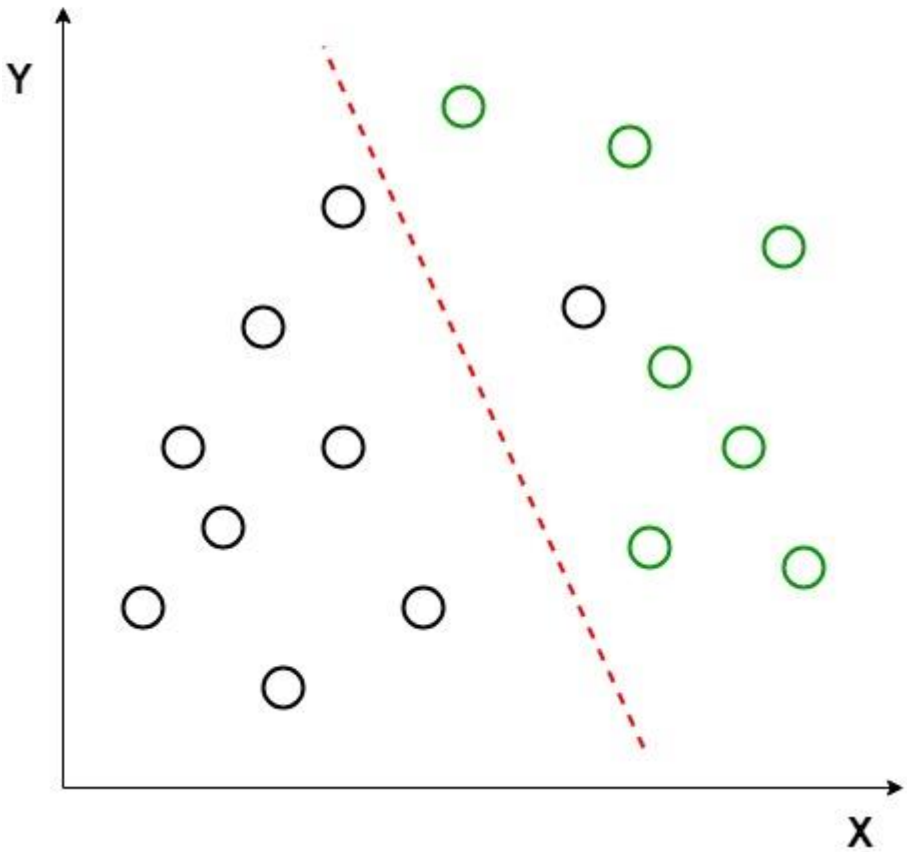
Final Result use PCA

Linear Discriminant Analysis (LDA)

Linear Discriminant analysis is one of the most popular dimensionality reduction techniques used for supervised classification problems in machine learning. It is also considered a pre-processing step for modeling differences in ML and applications of pattern classification.

LDA assumes that the data has a Gaussian distribution and that the covariance matrices of the different classes are equal. It also assumes that the data is linearly separable, meaning that a linear decision boundary can accurately classify the different classes.

Suppose we have two sets of data points belonging to two different classes that we want to classify. As shown in the given 2D graph, when the data points are plotted on the 2D plane, there's no straight line that can separate the two classes of data points completely. Hence, in this case, LDA (Linear Discriminant Analysis) is used which reduces the 2D graph into a 1D graph in order to maximize the separability between the two classes.

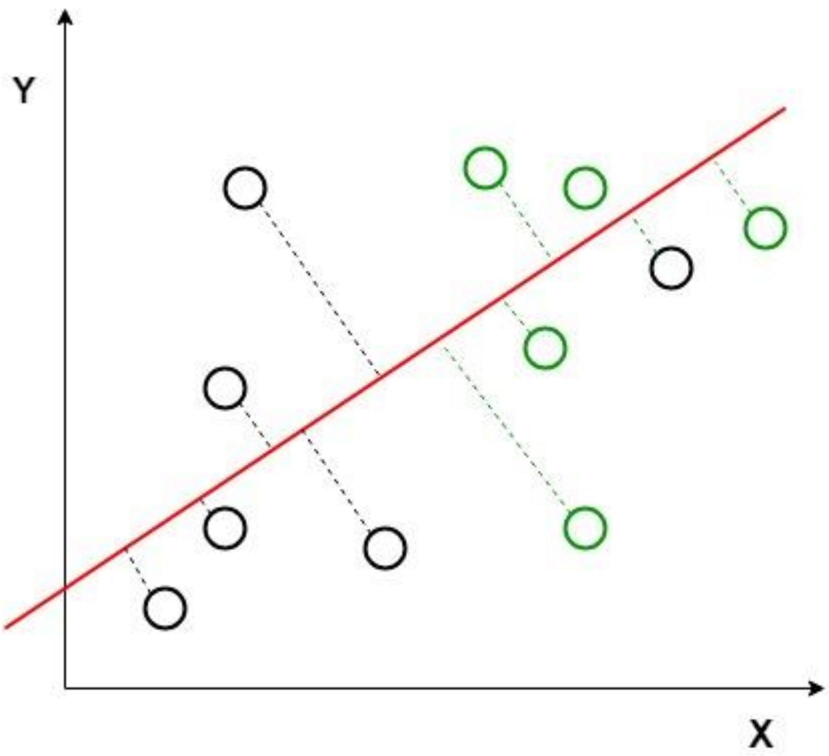


Linearly Separable Dataset

Here, Linear Discriminant Analysis uses both axes (X and Y) to create a new axis and projects data onto a new axis in a way to maximize the separation of the two categories and hence, reduces the 2D graph into a 1D graph.

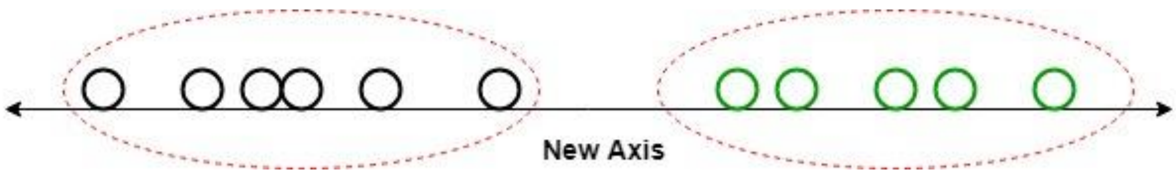
Two criteria are used by LDA to create a new axis:

1. Maximize the distance between the means of the two classes.
2. Minimize the variation within each class.



The perpendicular distance between the line and points

In the above graph, it can be seen that a new axis (in red) is generated and plotted in the 2D graph such that it maximizes the distance between the means of the two classes and minimizes the variation within each class. In simple terms, this newly generated axis increases the separation between the data points of the two classes. After generating this new axis using the above-mentioned criteria, all the data points of the classes are plotted on this new axis and are shown in the figure given below.



But Linear Discriminant Analysis fails when the mean of the distributions are shared, as it becomes impossible for LDA to find a new axis that makes both classes linearly separable. In such cases, we use non-linear discriminant analysis.

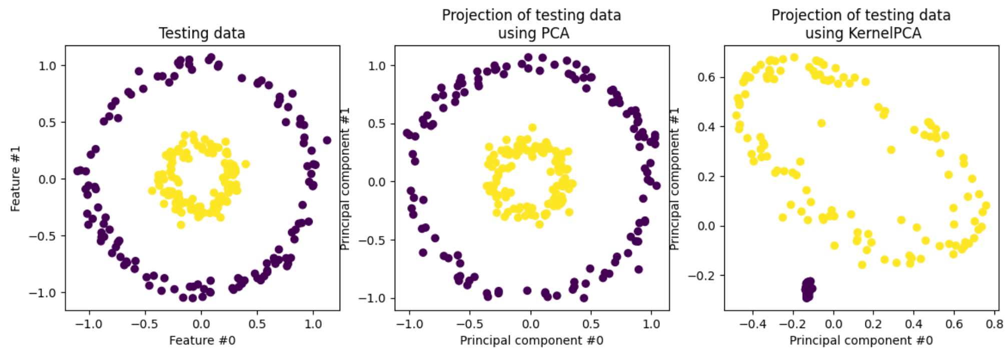
Kernel Principal Component Analysis (KPCA)

Kernel Principal Component Analysis (KPCA) is a technique used in machine learning for nonlinear dimensionality reduction. It is an extension of the classical Principal Component Analysis (PCA) algorithm, which is a linear method that identifies the most significant features or components of a dataset. KPCA applies a nonlinear mapping function to the data before applying PCA, allowing it to capture more complex and nonlinear relationships between the data points.

In KPCA, a kernel function is used to map the input data to a high-dimensional feature space, where the nonlinear relationships between the data points can be more easily captured by linear methods such as PCA. The principal components of the transformed data are then computed, which can be used for tasks such as data visualization, clustering, or classification.

Kernel PCA

Published on: January 2, 2022



Kernel PCA is an extension of PCA that allows for the separability of nonlinear data by making use of kernels. The basic idea behind it is to project the linearly inseparable data onto a higher dimensional space where it becomes linearly separable.

Kernel PCA can be summarized as a 4 step process:

1. Construct the kernel matrix K from the training dataset

$$K_{i,j} = \kappa(\mathbf{x_i}, \mathbf{x_j})$$

2. If the projected dataset $\{\phi(\mathbf{x}_i)\}$ doesn't have zero mean use the Gram matrix $\widetilde{K}$ to substitute the kernel matrix K.

$$\widetilde{K} = K - \mathbf{1_N}K - K\mathbf{1_N} + \mathbf{1_N}K\mathbf{1_N}$$

3. Use $K_{a_k} = \lambda_k N_{a_k}$ to solve for the vector a_i.

4. Compute the kernel principal components $y_k(x)$

$$y_k(\mathbf{x}) = \phi(\mathbf{x})^T \mathbf{v}_k = \sum_{i=1}^{N} a_{ki}\kappa(\mathbf{x_i}, \mathbf{x_j})$$