**Chapter 8: Dimensionality Reduction**

**@** [**https://medium.com/analytics-vidhya/a-complete-guide-on-dimensionality-reduction-62d9698013d2**](https://medium.com/analytics-vidhya/a-complete-guide-on-dimensionality-reduction-62d9698013d2)

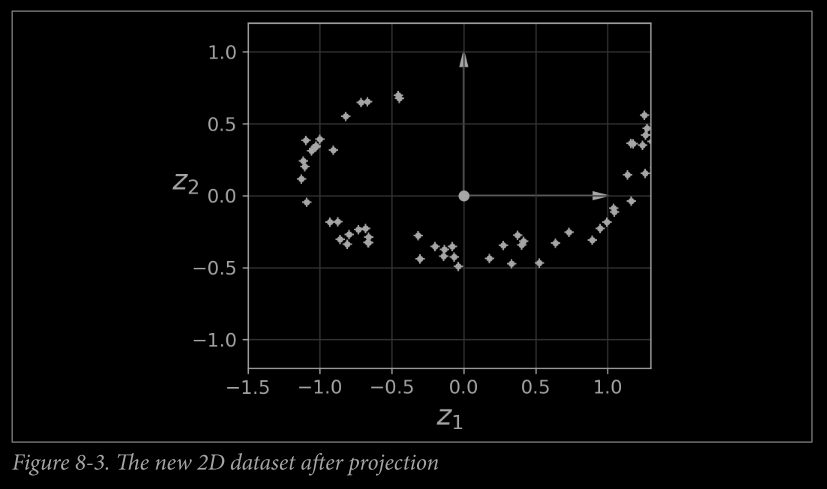
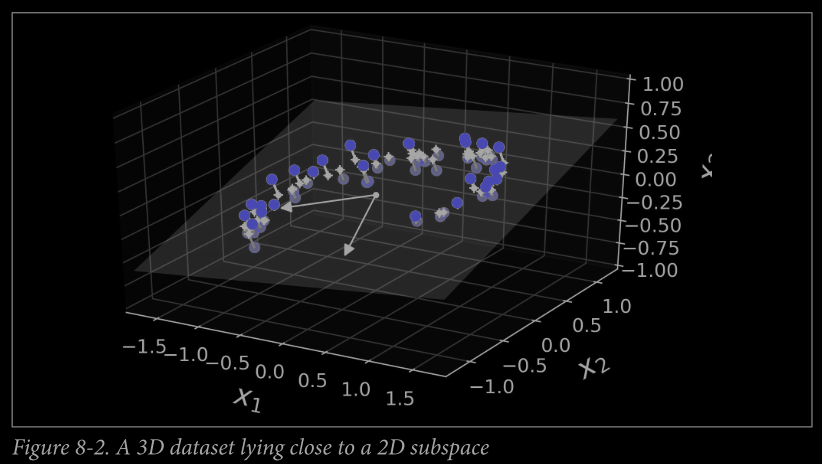
**@** [**https://medium.com/@rinu.gour123/dimensionality-reduction-in-machine-learning-dad03dd46a9e**](https://medium.com/@rinu.gour123/dimensionality-reduction-in-machine-learning-dad03dd46a9e)

**@ https://towardsdatascience.com/11-dimensionality-reduction-techniques-you-should-know-in-2021-dcb9500d388b**

**#** The more dimensions the training set has, the greater is the risk of overfitting. ( # average distance in any hyperplane)

**Main Approaches for Dimensionality Reduction**

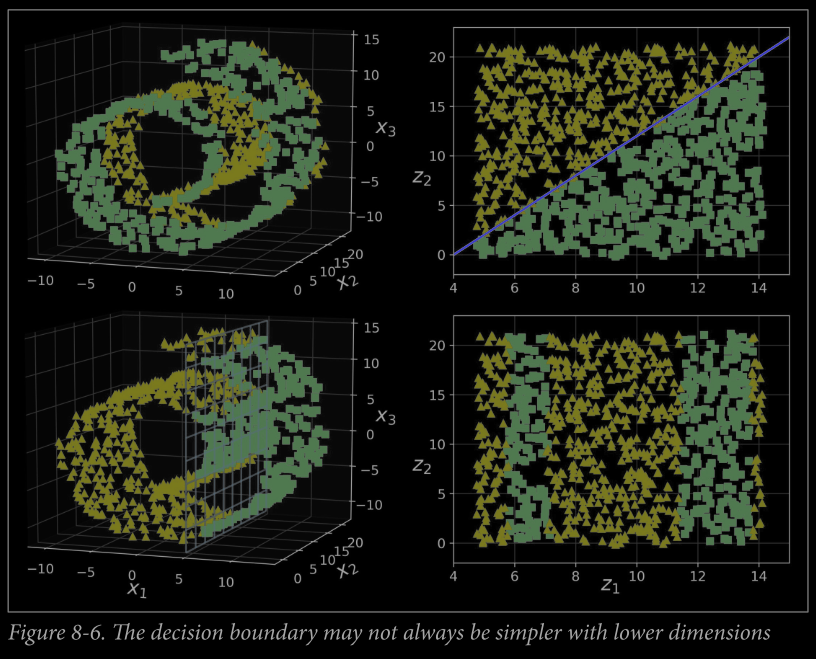
1) **Projection**: In this, all the training instances are projected perpendicularly from higher dimensional space to lower-dimensional space(subspace).



2) **Manifold Learning**: A d-dimensional manifold is part of an n-dimensional space (where d<n) that resembles a d-dimensional hyperplane locally. Many dimensionality reduction algorithms work by modeling the manifold on which the training instances lie; this is called Manifold Learning.

It relies on the **manifold assumption**, also called the **manifold hypothesis**, which holds that most real-world high dimensional datasets lie close to a much lower-dimensional manifold.

# Another Assumption that manifold learning takes is that the task must be simpler in lower-dimensional space.



**PCA: Principal Component Analysis** first identifies the hyperplane that lies closest to the data, and then it projects the data onto it.

@ <https://medium.com/codex/principal-component-analysis-pca-how-it-works-mathematically-d5de4c7138e6>

@ <https://towardsdatascience.com/a-one-stop-shop-for-principal-component-analysis-5582fb7e0a9c>

@ <https://medium.com/analytics-vidhya/principal-component-analysis-pca-558969e63613>

**Preserving the Variance:** The ”right” hyperplane for projection dimensionality reduction is the plane where the training data have **maximum variance.**

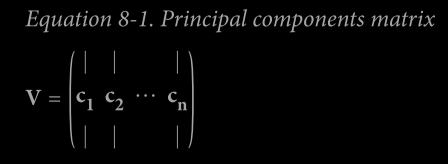
Or, The axis that **minimizes the mean squared distance** between the original dataset and its projection onto that axis.

**Principal Components:**

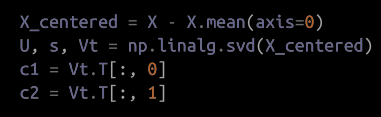
**#** The unit vector that defines the ith axis is called the ith principal component (PC). All the principal components are mutually perpendicular.

# The direction of the principal components is unstable: if you perturb the training set slightly and run PCA again, some of the new PCs may point in the opposite direction of the original PCs. However, they will generally still lie on the same axes. In some cases, a pair of PCs may even rotate or swap, but the plane they define will generally remain the same.

**#** The standard factorization technique for finding Principal Components is **Singular Value Decomposition (SVD)** that can decompose the training set matrix X into the matrix multiplication of three matrices U Σ VT, where V contains all the principal components.



# NumPy’s svd() function is used to obtain all the PC’s

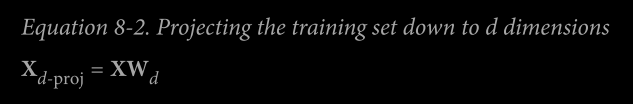


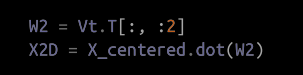
# PCA assumes that the dataset is centered around the origin. Scikit-Learn’s PCA class automatically centers the dataset but for the other libraries, we need to do it separately.

**Projecting Down to d Dimensions**

# The dimensions of the dataset are reduced to d dimensions by projecting onto the hyperplane defined by the **first d principal components**.

# The training set is projected onto the hyperplane by computing the matrix multiplication of the training set matrix X by the matrix Wd, defined as the matrix containing the first d principal components (i.e., the matrix composed of the first d columns of V).



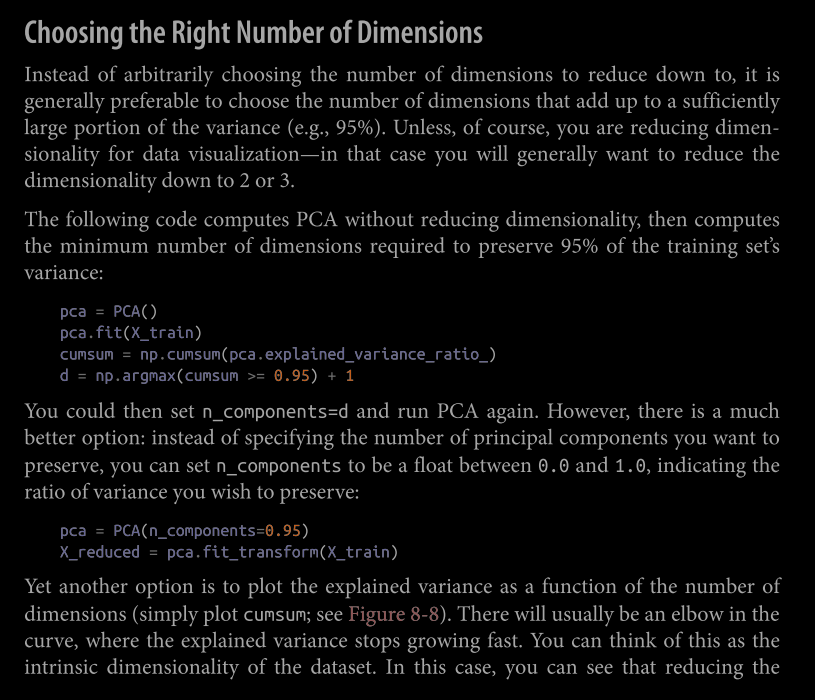


# Using Scikit-Learn: Scikit-Learn’s PCA class implements PCA using SVD decomposition (note that it automatically takes care of centering the data)

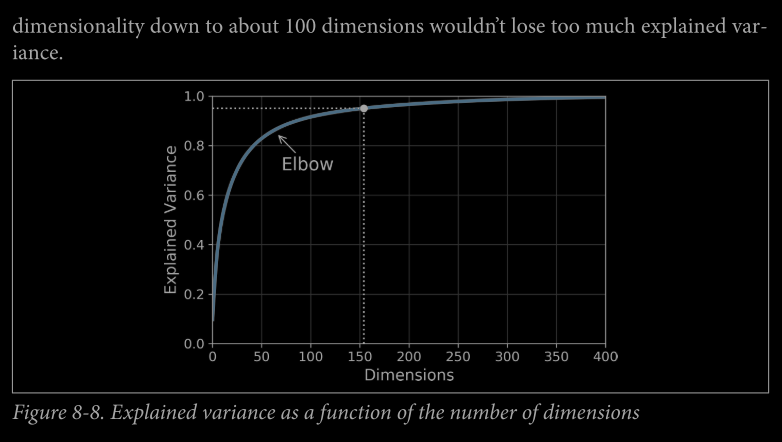


**Explained Variance Ratio:**  It indicates the proportional of the dataset’s variance that lies along the axis of each principal component

This tells that 84.2% of the data lies along with the first principal component and 14.6% along the 2nd PC.



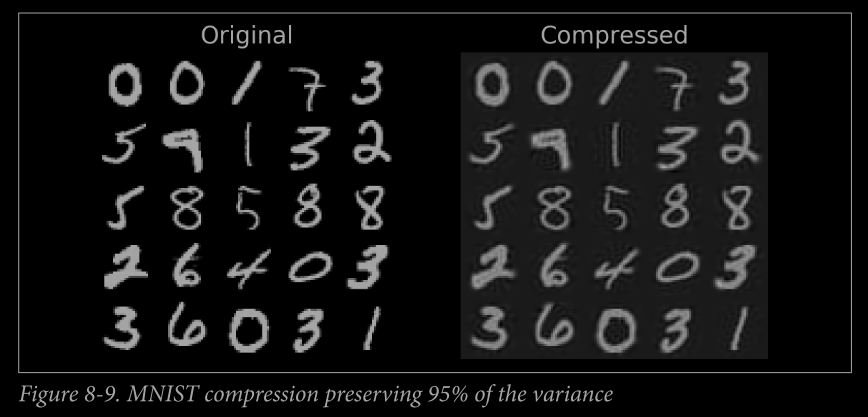
@ <https://towardsdatascience.com/how-to-select-the-best-number-of-principal-components-for-the-dataset-287e64b14c6d>

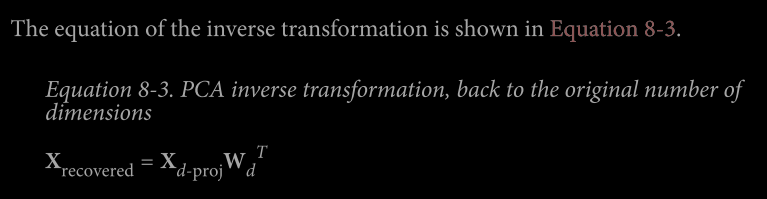


**PCA for Compression:**

# PCA Projections not only compresses the data, but it can also decompress the data by applying inverse transformation of PCA.

The mean squared distance between the original data and the reconstructed data(compressed and decompressed) is called the reconstruction error.





**Curse of Dimensionality:**

**@** [**https://towardsdatascience.com/the-curse-of-dimensionality-50dc6e49aa1e**](https://towardsdatascience.com/the-curse-of-dimensionality-50dc6e49aa1e)

**@** [**https://medium.com/@paritosh\_30025/curse-of-dimensionality-f4edb3efa6ec**](https://medium.com/@paritosh_30025/curse-of-dimensionality-f4edb3efa6ec)

**@** [**https://medium.com/analytics-vidhya/how-to-break-the-curse-of-dimensionality-b366d5d23c86**](https://medium.com/analytics-vidhya/how-to-break-the-curse-of-dimensionality-b366d5d23c86)

**@** [**https://adityaroc.medium.com/curse-of-dimensionality-and-pca-f90f1258a7f2**](https://adityaroc.medium.com/curse-of-dimensionality-and-pca-f90f1258a7f2)

**Randomized PCA and Incremental PCA: READ FROM THE BOOK (Page 227)**

**Kernel PCA**

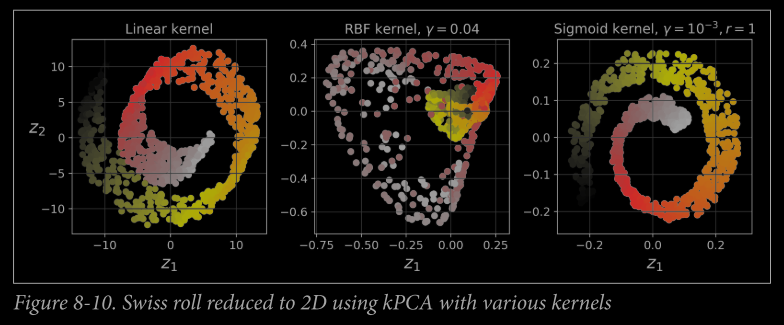
**@** [**https://medium.com/@phsamuel.work/kernel-pca-f7a40f0b4265**](https://medium.com/@phsamuel.work/kernel-pca-f7a40f0b4265)

**@** [**https://medium.com/pursuitnotes/day-47-kernel-pca-edfcedae3b64**](https://medium.com/pursuitnotes/day-47-kernel-pca-edfcedae3b64)

Kernel Method: The mathematical technique that implicitly maps instances into a very high-dimensional space (called the feature space), enabling nonlinear classification and regression with Support Vector Machines. Recall that a linear decision boundary in the high-dimensional feature space corresponds to a complex nonlinear decision boundary in the original space.

**It turns out that the same trick can be applied to PCA, making it possible to perform complex nonlinear projections for dimensionality reduction. This is called Kernel PCA (kPCA).**

# It is often good at preserving clusters of instances after projection, or sometimes even unrolling datasets that lie close to a twisted manifold.



**Selecting a Kernel and Tuning Hyperparameters: READ THE BOOK (Pg. 229)**

**@** [**https://www.analyticsvidhya.com/blog/2022/02/a-comprehensive-guide-on-hyperparameter-tuning-and-its-techniques/**](https://www.analyticsvidhya.com/blog/2022/02/a-comprehensive-guide-on-hyperparameter-tuning-and-its-techniques/)

**@** [**https://sinashariati.medium.com/tune-the-feature-reductions-pca-and-mca-to-build-a-model-on-a-categorical-and-numerical-data-7c27310607b8**](https://sinashariati.medium.com/tune-the-feature-reductions-pca-and-mca-to-build-a-model-on-a-categorical-and-numerical-data-7c27310607b8)

**@** [**https://towardsdatascience.com/machine-learning-step-by-step-6fbde95c455a**](https://towardsdatascience.com/machine-learning-step-by-step-6fbde95c455a)

**LLE (Locally Linear Embedding): READ THE BOOK**