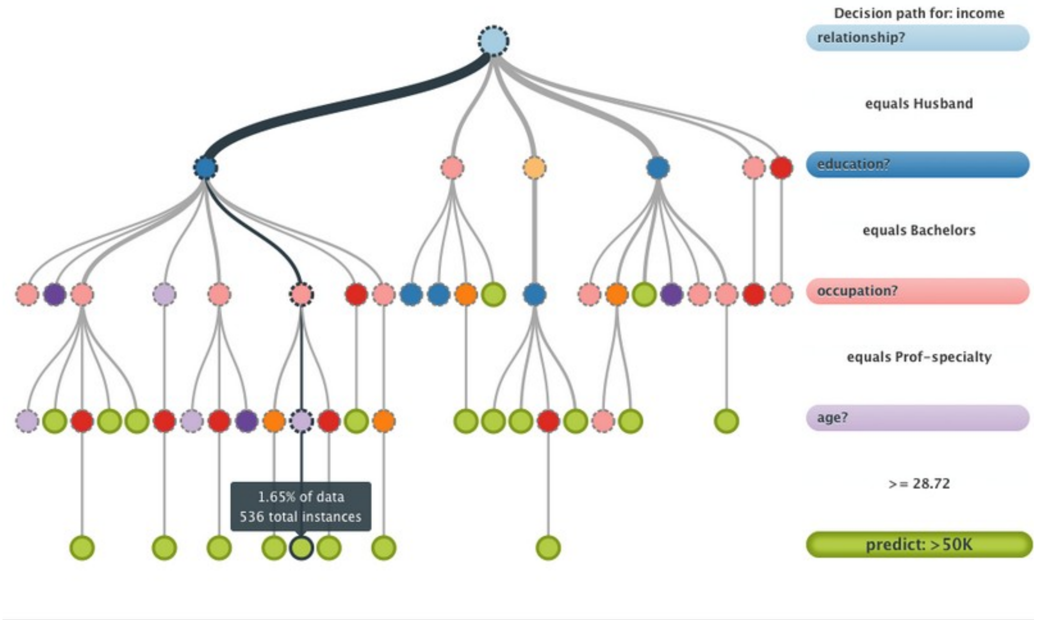
Machine learning
Lecture 3 – a discussion



Marcin Wolter

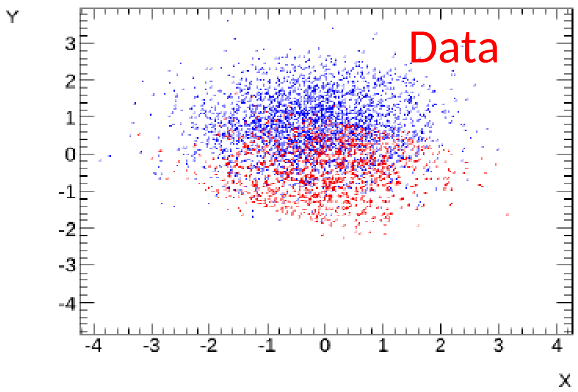
IFJ PAN

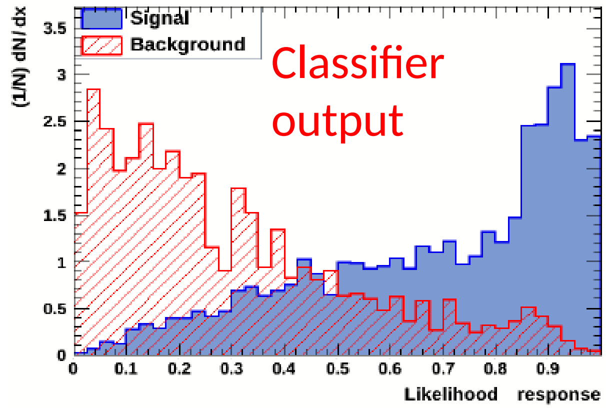
28 October 2020

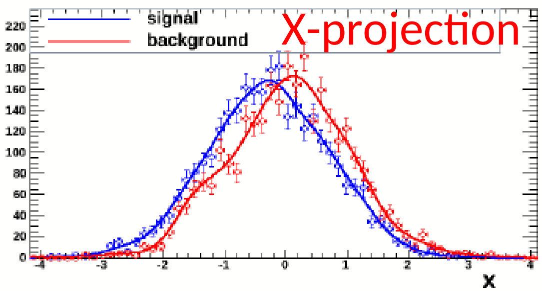
- Discussion and some examples

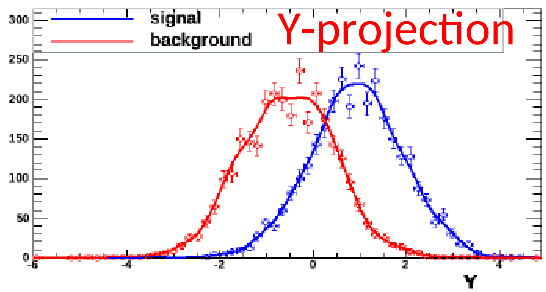
All slides will be here: https://indico.ifj.edu.pl/event/397/

Naive Bayes classifier
(repetition from the previous lecture)









Frequently called **"projected likelihood"** by physicists

- Based on the assumption, that variables are independent (so „naive"):

$$P(y \mid x_1, \dots, x_n) = \frac{P(y)P(x_1, \dots x_n \mid y)}{P(x_1, \dots, x_n)}$$

"Naive" assumption: $P(x_i|y, x_1, \dots, x_{i-1}, x_{i+1}, \dots, x_n) = P(x_i|y),$

Bayes formula

$$P(y \mid x_1, \dots, x_n) = \frac{P(y) \boxed{\prod_{i=1}^{n} P(x_i \mid y)}}{P(x_1, \dots, x_n)}$$

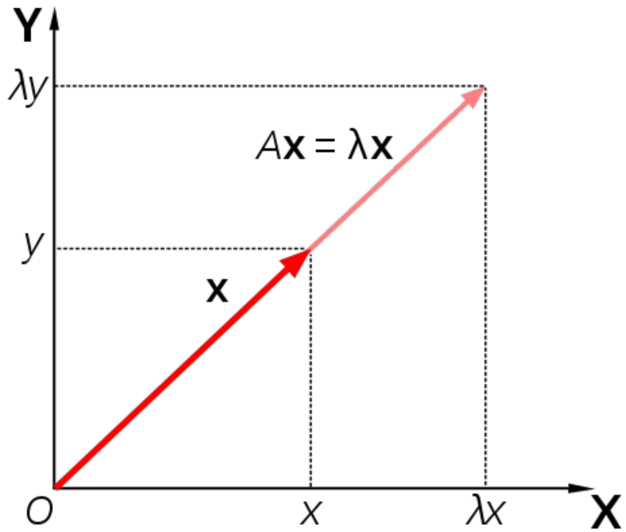
- Output probability is **a product of probabilities for all variables.**
- Fast and stable, not optimal, but in many cases sufficient.

Eigenvalues and eigenvectors

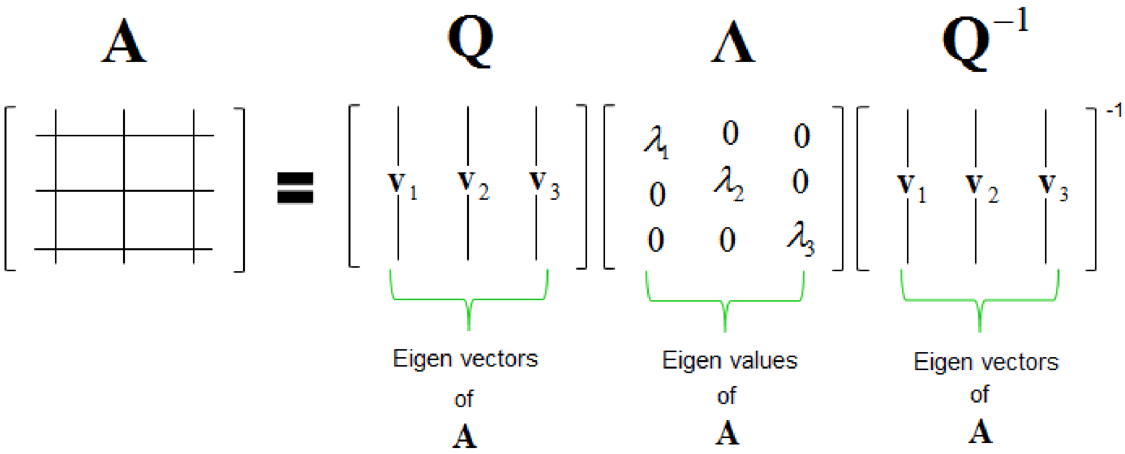
In essence, an eigenvector v of a linear transformation A is a non-zero vector that, when A is applied to it, does not change direction. Applying A to the eigenvector only scales the eigenvector by the scalar value λ, called an eigenvalue. This condition can be written as the equation

$$\mathbf{A}(\mathbf{v}) = \lambda\, \mathbf{v}$$

referred to as the eigenvalue equation or eigenequation. In general, λ may be any scalar. For example, λ may be negative, in which case the eigenvector reverses direction as part of the scaling, or it may be zero or even complex.



Matrix A acts by stretching the vector x, not changing its direction, so x is an eigenvector of A.



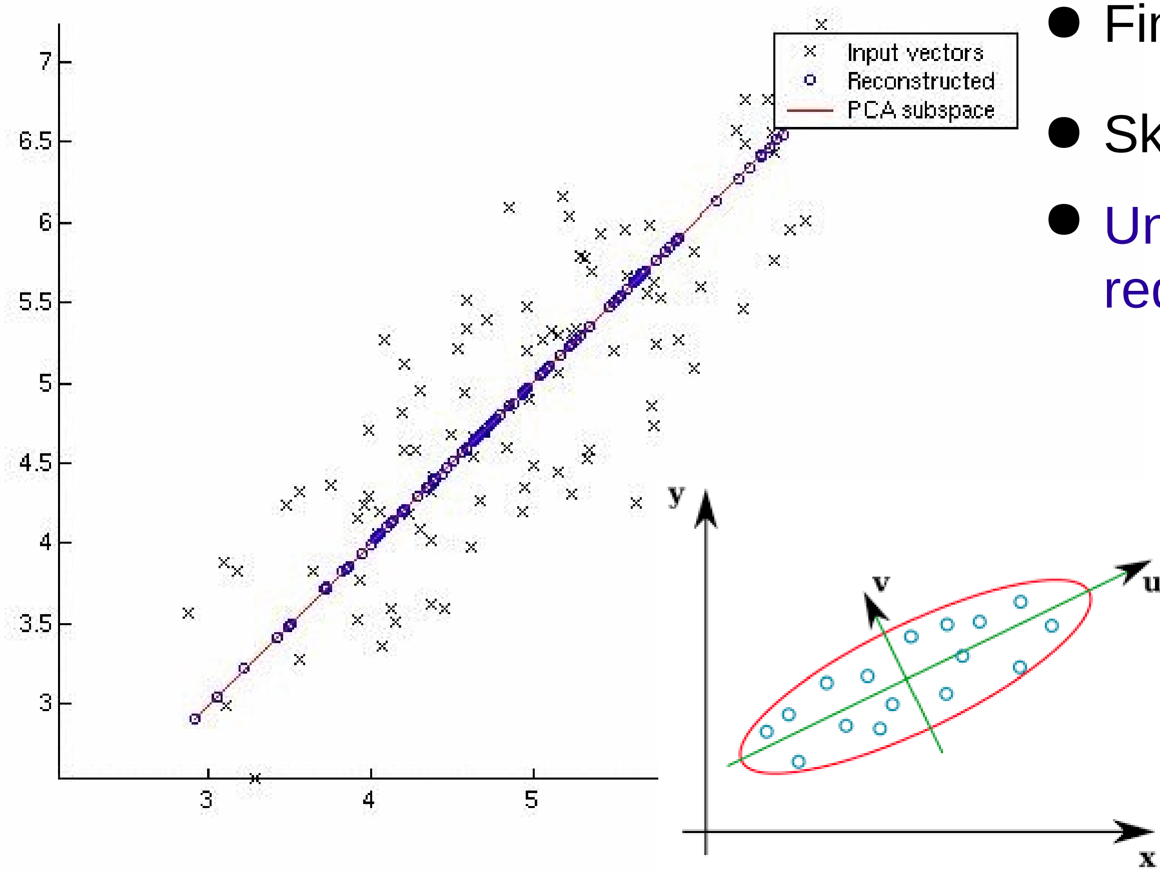
Eigendecomposition of matrix

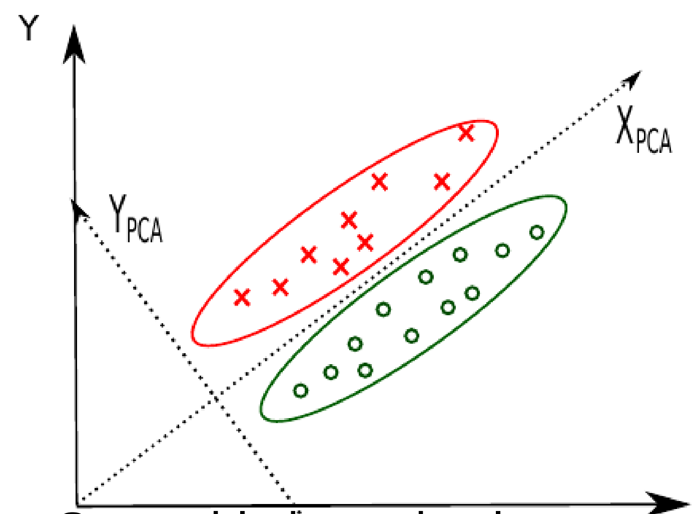
Principal Component Analysis - PCA

- Task: reduce the number of dimensions minimizing the loss of information
- Finds the orthogonal base of the covariance matrix, the eigenvectors with the smallest eigenvalues might be skipped

Procedure:

- Find the covariance matrix Cov(X)
- Find eigenvalues λ_i and eigenvectors v_i
- Skip smallest λ_i
- Unsupervised learning & dimensionality reduction



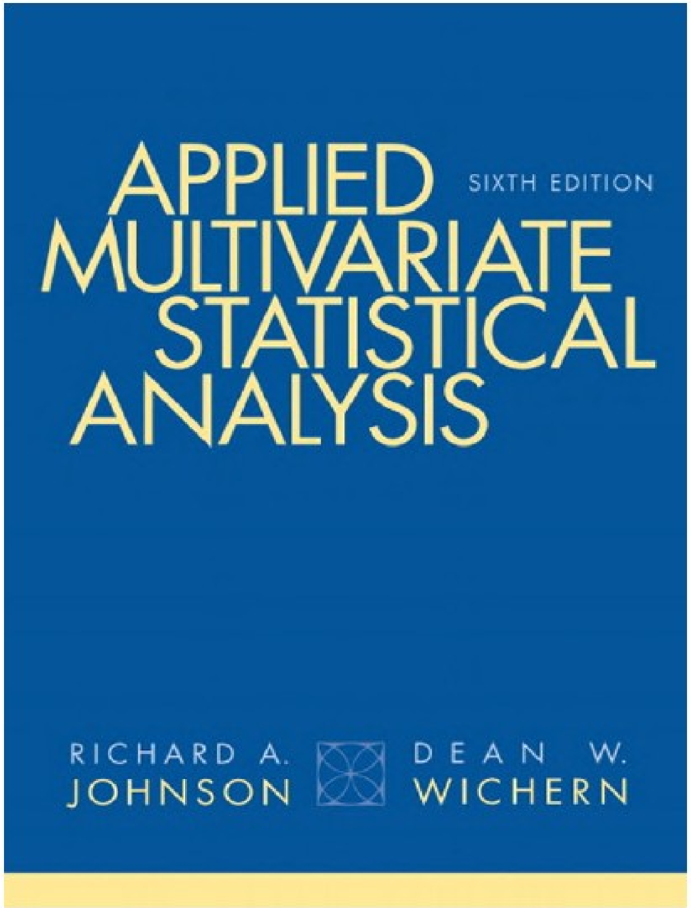


Separable by a simple cut,
but in this case PCA doesn't help...

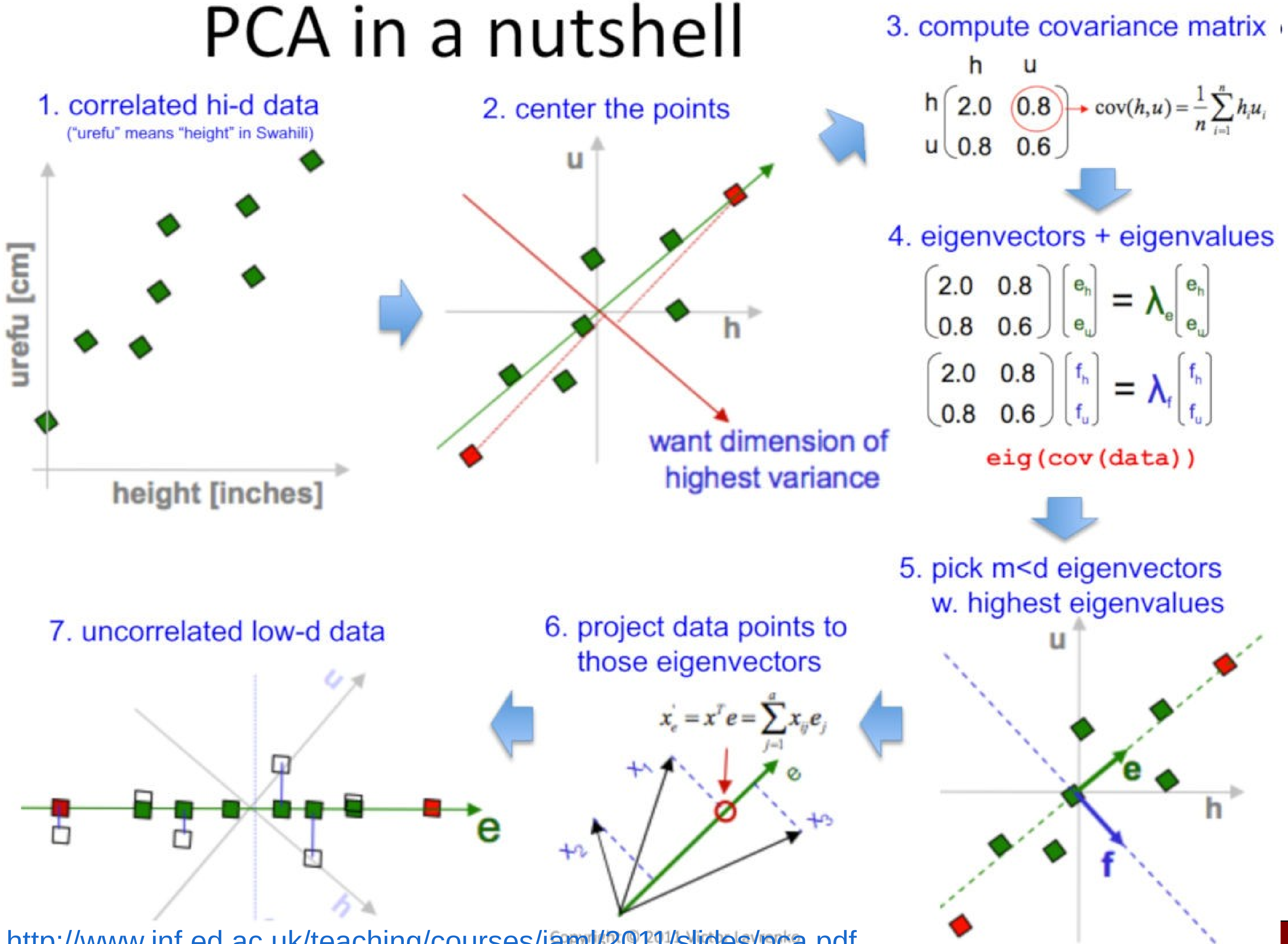
Principal Component Analysis (PCA)

Nicely explained in:

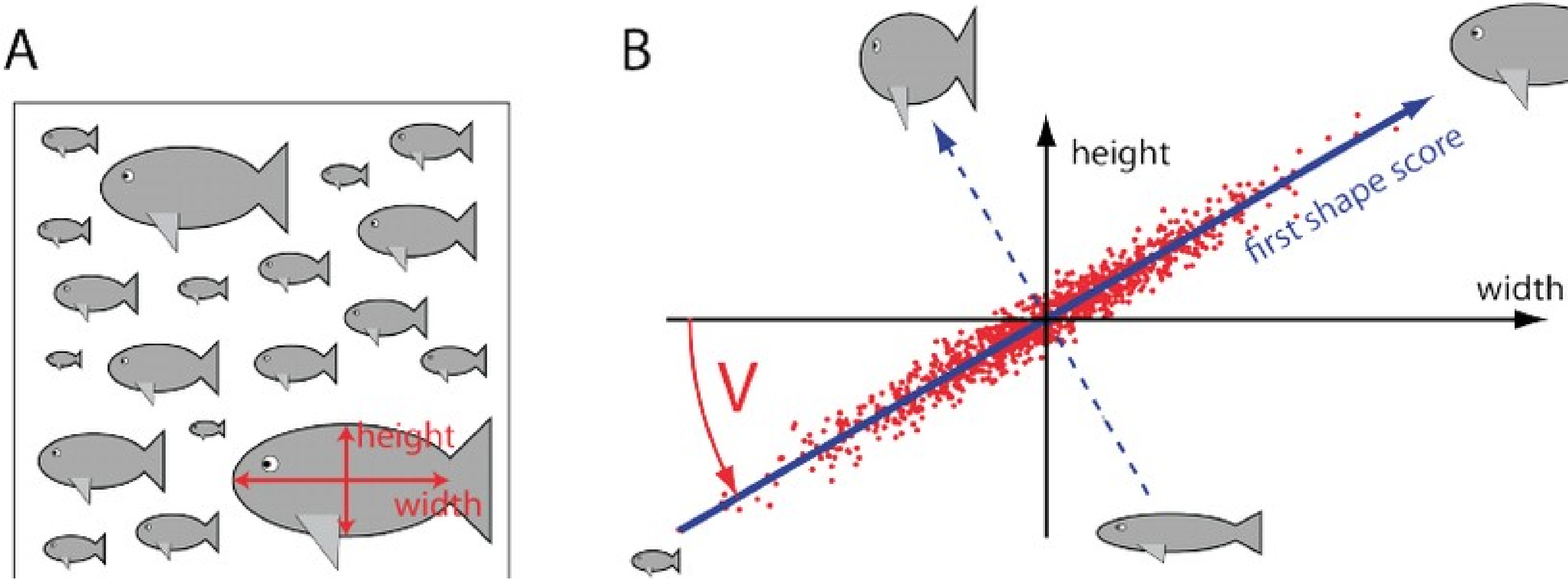
http://docshare04.docshare.tips/files/12598/125983744.pdf



PCA in a nutshell



http://www.inf.ed.ac.uk/teaching/courses/iaml/2011/slides/pca.pdf

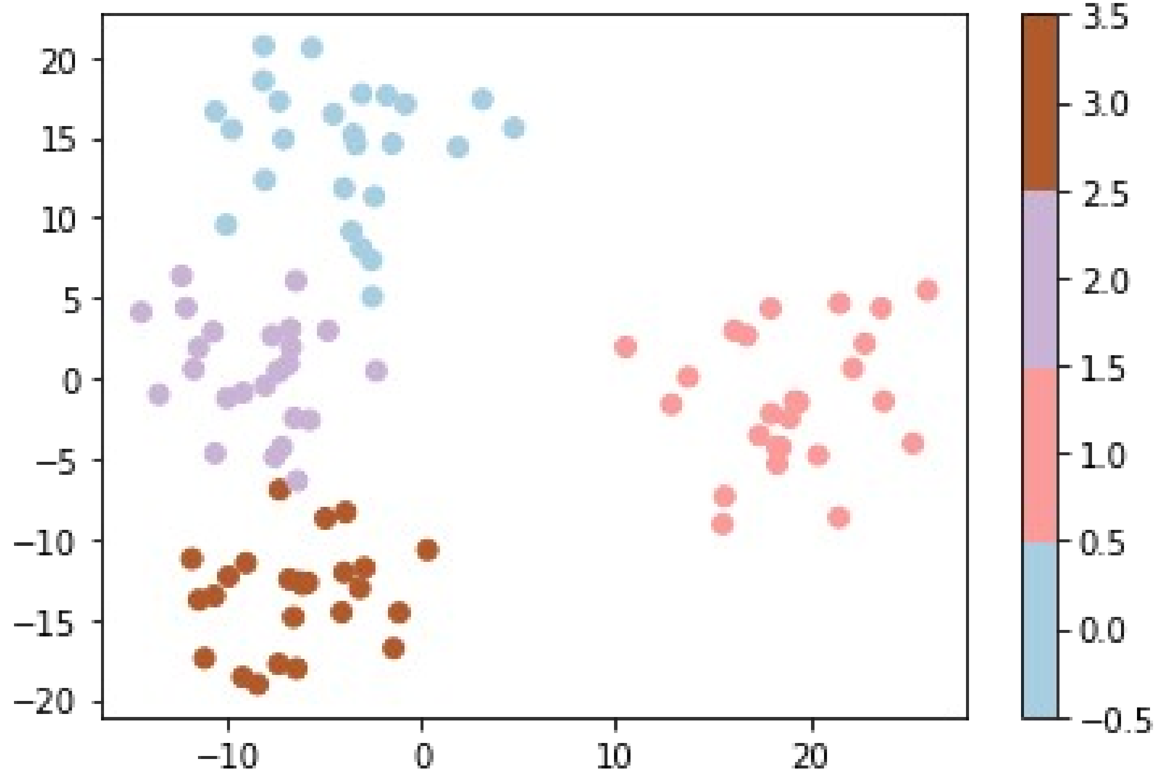


We can describe the shape of a fish with two variables: height and width. However, these two variables are not independent of each other. In fact, they have a strong correlation. Given the height, we can probably estimate the width; and vice versa. Thus, we may say that the shape of a fish can be described with a single component.

This doesn't mean that we simply ignore either height or width. Instead, we transform our two original variables into two orthogonal (independent) components that give a complete alternative description. The first component (blue line) will explain most of the variation in the data. The second component (dotted line) will explain the remaining variation. Note that both components are derived from both height and width.

Very simple PCA example

- https://github.com/marcinwolter/MachineLearning2020/blob/main/PCA_scikit_example.ipynb

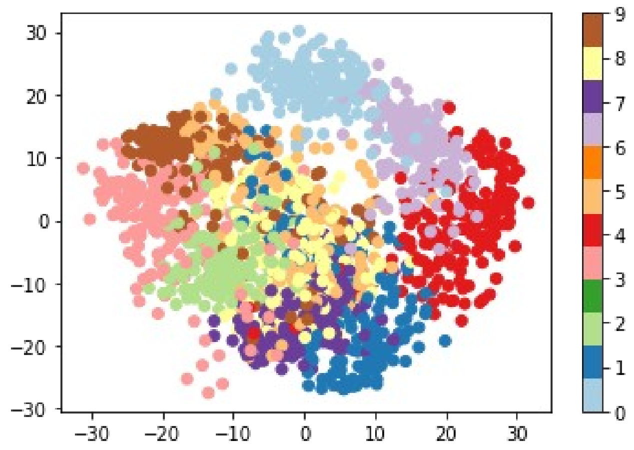


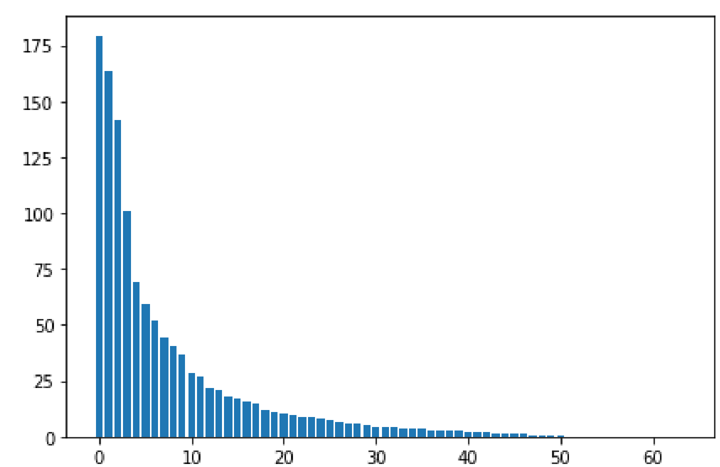
Example

- All examples will be available here:
  https://github.com/marcinwolter/MachineLearning2020

- https://github.com/marcinwolter/MachineLearning2020/blob/main/plot_digits_classif.ipynb
  - iPython notebook prepared to run on Google Colaboratory
  https://colab.research.google.com/

  – Reads handwritten digits
  – Performs PCA
  – Displays two first principal components:

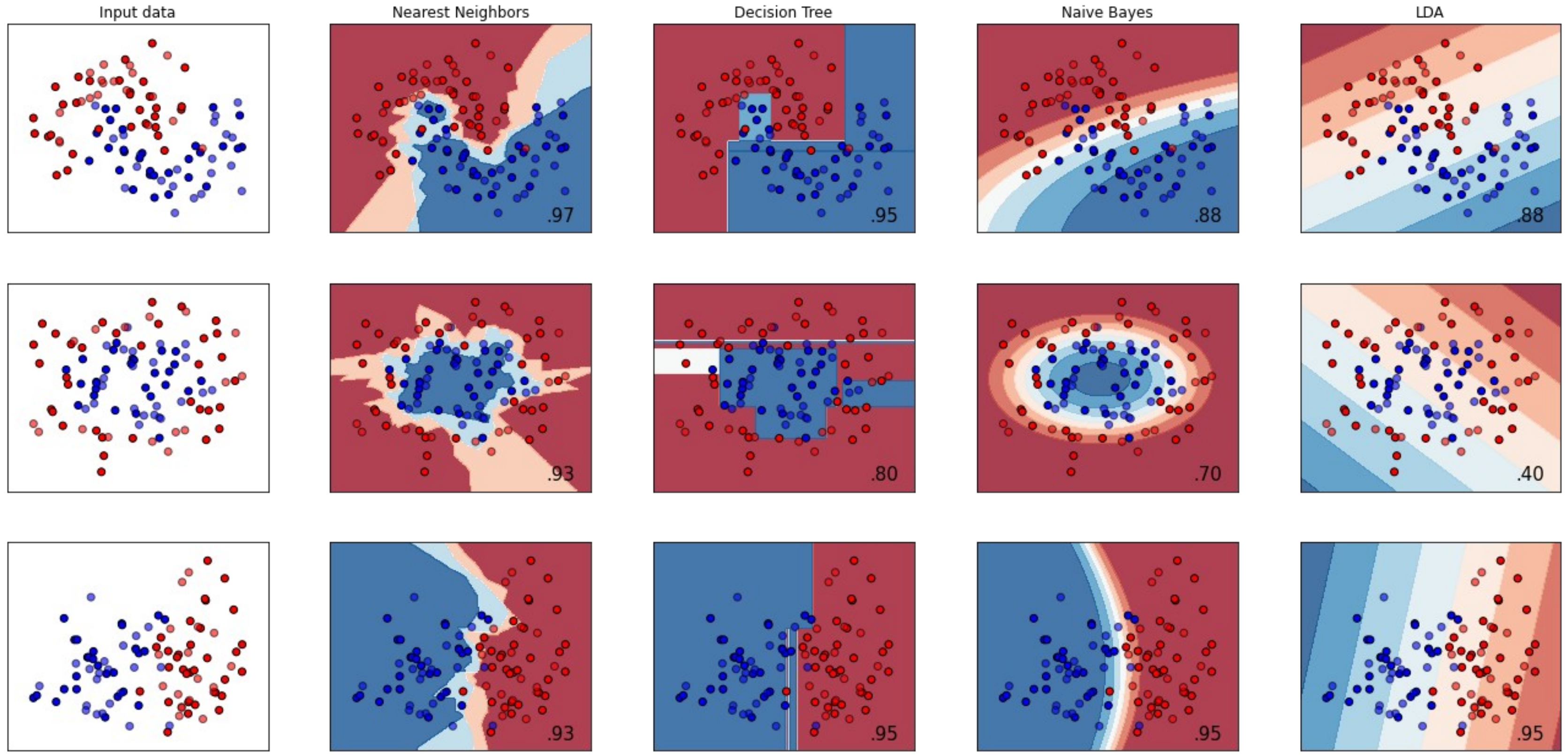




  – Classification using Naive Bayes and LDA

Example of simple classifiers

https://github.com/marcinwolter/MachineLearning2020/blob/main/simple_classifier_comparison.ipynb



Classification of faces

- PCA – each face can be represented as a combination of a limited number of "eigenfaces"
- https://github.com/marcinwolter/MachineLearning2020/blob/main/plot_face_recognition.ipynb