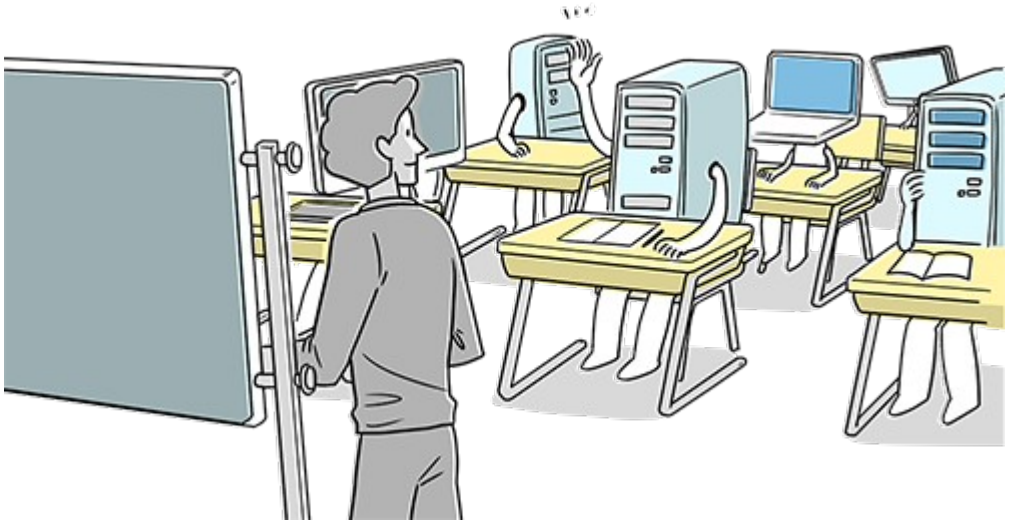
Machine learning
Lecture 1



Marcin Wolter

IFJ PAN

14 October 2020

- Machine learning: what does it mean?
- Software and literature.
- A little bit of mathematics and examples of simple linear classifiers.
- Some examples

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

Outline of the course

- Probability – an introduction
- Machine learning: what does it mean? Software to work with and literature. A little bit of mathematics and examples of simple linear classifiers.
- Simple non-linear methods like k-nearest neighbors, Parzen kernel methods, Independent Component Analysis ICA
- Ensemble learning – Boosted Decision Trees BDT
- Cluster analysis
- Shallow neural networks
- Bayesian Neural Networks, Deep Neural Networsk
- How to build a Deep Neural Network – Keras tutorial, Convolutional Deep Neural Network
- Training - cross-validation. Optimization of hyperparameters.
- Mixed Density Network
- Generative Adversarial Networks (GANs)

Recommended books

- In Polish: M. Krzyśko, Systemy uczące się: rozpoznawanie wzorców, analiza skupień i redukcja wymiarowości. WNT, 2008.
- C. Bishop, Pattern recognition and machine learning. Springer, 2009.
- Internet information about Deep Neural Networks – many things have changed in the recent years. I will point to some manuals during the lecture.

Programs

- **Python** programming language (there is python course as well)

- http://scikit-learn.org
  scikit-learn - Machine Learning in Python
  Simple and efficient tools for data mining and data analysis
  Built on NumPy, SciPy, and matplotlib

- https://keras.io/
  Keras: The Python Deep Learning library
  Emulates Deep Neural Network, uses google TensorFlow software

- **Google Colaboratory** as a working platform – you need a google ID
  https://colab.research.google.com/
  Platform to run python notebooks, gives free GPU

- **Github** to store the code
  https://github.com

We will say few more words about today

About the course

- We will have 20 hours of lectures in total.
- I propose to do some exercises, run some python programs on *google colaboratory*. So please use your laptops!
- We should make some type of exam. I propose, that instead of a regular exam you will prepare small projects using machine learning. This gives a chance to learn how to use machine learning tools.

Statistics

- **Statistics –science of collecting and analyzing numerical data.**
- **First works** –أبو يـوسفيـعقوببـنإسحاقالـصبّاح الـكندي **Al-Kindi (801-873)** - used statistical methods to break the Caesar cipher by analyzing the frequency of letters in the text.

Caesar cipher

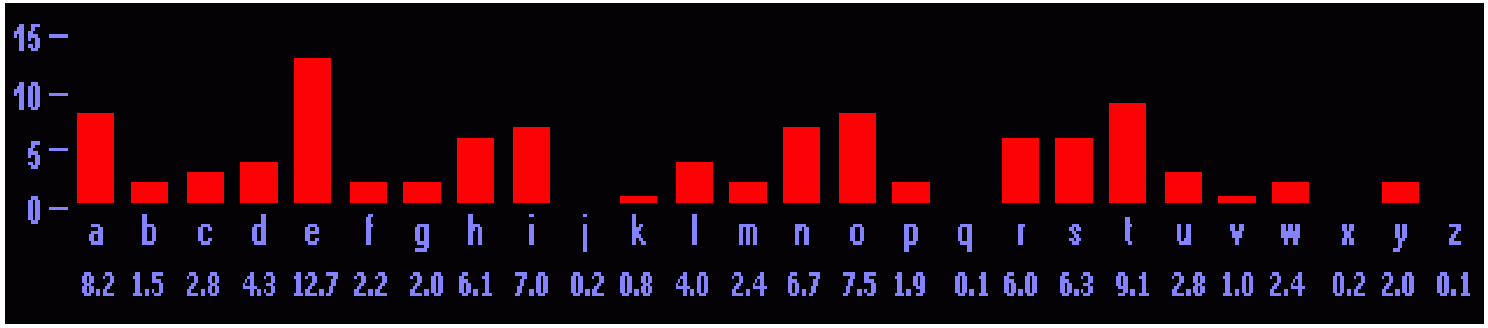
- Each letter in the text is replaced by another, but shifted by ***n*** letters:

A	B	C	D	E
C	D	E	F	G

- In general – replacement cipher (each letter is replaced by another, but always the same letter)

A	B	C	D	E
Z	D	P	G	T

- Frequency analysis – investigation, how frequently each symbols appears in text.



- Frequency of appearance of different letters in English.

What is an “event”?
A basic concept

- Elementary event is each result of an experiment (like throwing a dice), which result is random. It contains only a single outcome in the sample space.

The numerical value of a probability may sometimes be obtained from its "classical" definition: The probability is equal to the quotient of the number of cases "favoring" a certain event to the total number of "equally possible" cases. (Laplace 1812).

- Let’s denote a set of all possible events as Ω. Elements of a set Ω are elementary events ω, so Ω is a set of elementary events. The set of events favoring A is a subset of Ω and:

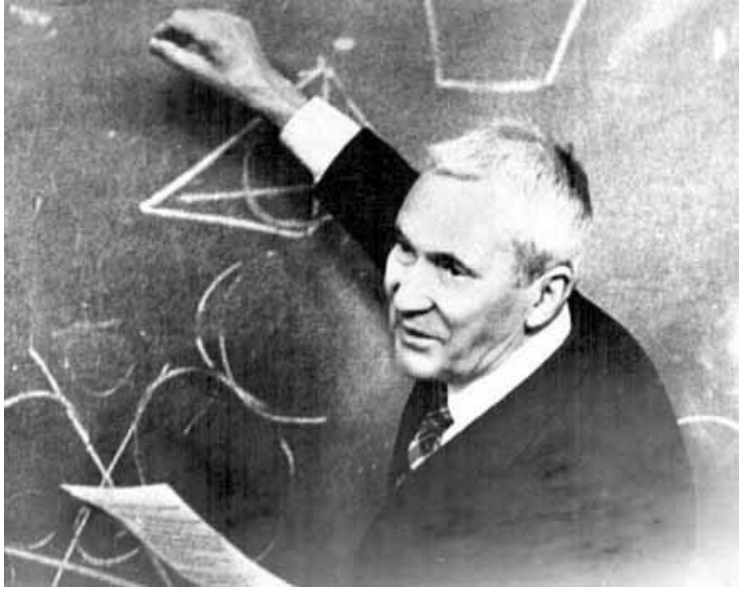
$$P(A) = \frac{|A|}{|\Omega|}$$

where |A| is a number of elements of a set A, and |Ω| a number of elements of a set Ω.

Pierre-Simon Laplace (1749–1827)

Example: probability of getting 6 while throwing a dice. Set of elementary events Ω = {1, 2, 3, 4, 5, 6}, so the number of elementary events is |Ω| = 6. Set of events favoring A = {6}, their number is |A| = 1. So P(6)=1/6

Axiomatic definition of probability



Andrey Kolmogorov
Андре́й Никола́евич Колмого́ров (1903-1987)

1) **For any event A, P(A) ≥ 0.**

   In English, that's "For any event A, the probability of A is greater or equal to 0".

2) **When S is the sample space of an experiment; i.e., the set of all possible outcomes, P(S) = 1.**

   In English, that's "The probability of any of the outcomes happening is one hundred percent", or—paraphrasing— "anytime this experiment is performed, something happens".

3) **If A and B are mutually exclusive outcomes, P(A ∪ B) = P(A) + P(B).**

   Here ∪ stands for 'union'. We can read this by saying "If A and B are mutually exclusive outcomes, the probability of either A or B happening is the probability of A happening plus the probability of B happening"

All other properties derive from the above.

Some properties of probability

Some properties, which can be derived from the Kolmogorov definition:

- $P(\bar{A}) = 1 - P(A)$, where $\bar{A}$ is a complement of A,
- $P(A \cup \bar{A}) = 1$,
- $0 \leq P(A) \leq 1$,
- $P(\varnothing) = 0$,
- if $A \subset B$, then $P(A) \leq P(B)$,
- $P(A \cup B) = P(A) + P(B) - P(A \cap B)$.

We leave them without proof.

Frequentist definition of probability

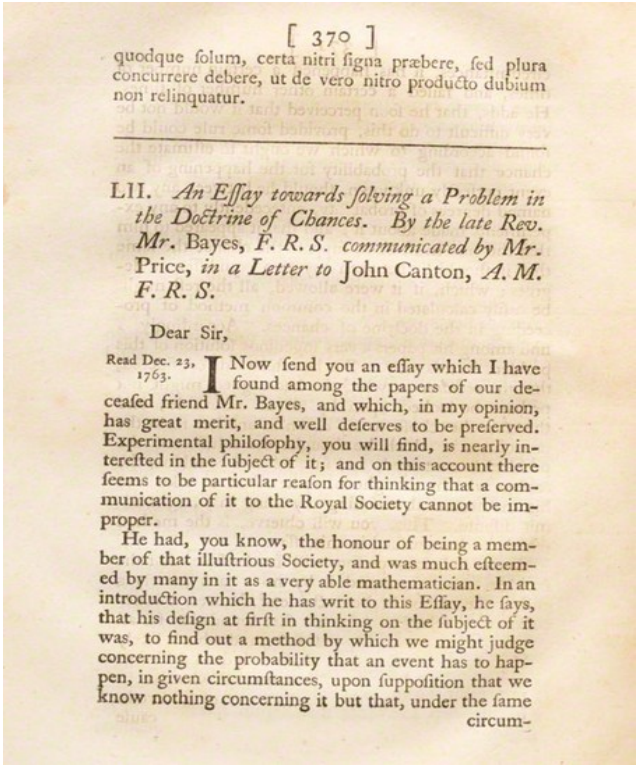
Probability (frequentist definition) of an event A is a limit (N approaching infinity) of the ratio of n events when the event A occurred to the total number of trials N:

$$P(A) = \lim_{N \to \infty} \frac{n}{N}$$

What is a probability of getting “6” while throwing a dice? It is the relative ratio of obtaining “6” in an infinite series of throws.

Bayesian definition

- Probability “a priori”, called unconditional, is a measure of belief, based on rational premises, that a given event will occur.
- In the next step we perform an experiment, called observation, and their results allow to modify the probability. We get the probability “a posteriori”, which is again a measure of belief, but modified by the observation.



[370]

quodque ſolum, certa nitri ſigna præbere, ſed plura concurrere debere, ut de vero nitro producto dubium non relinquatur.

LII. *An Eſſay towards ſolving a Problem in the Doctrine of Chances. By the late Rev. Mr.* Bayes, *F. R. S. communicated by Mr.* Price, *in a Letter to* John Canton, *A. M. F. R. S.*

Dear Sir,

Read Dec. 23, 1763. I Now ſend you an eſſay which I have found among the papers of our deceaſed friend Mr. Bayes, and which, in my opinion, has great merit, and well deſerves to be preſerved. Experimental philoſophy, you will find, is nearly intereſted in the ſubject of it; and on this account there ſeems to be particular reaſon for thinking that a communication of it to the Royal Society cannot be improper.

He had, you know, the honour of being a member of that illuſtrious Society, and was much eſteemed by many in it as a very able mathematician. In an introduction which he has writ to this Eſſay, he ſays, that his deſign at firſt in thinking on the ſubject of it was, to find out a method by which we might judge concerning the probability that an event has to happen, in given circumſtances, upon ſuppoſition that we know nothing concerning it but that, under the ſame circum-

Thomas Bayes (1702 - 1761) was an English statistician, philosopher and Presbyterian minister.The most important work: „Essay Towards Solving a Problem in the Doctrine of Chances”.

Summarizing probability interpretations

I. **Relative frequency**

A, *B*, ... are outcomes of a repeatable experiment

$$P(A) = \lim_{n \to \infty} \frac{\text{times outcome is } A}{n}$$

quantum mechanics, particle scattering, radioactive decay...

II. **Subjective probability**

A, *B*, ... are hypotheses (statements that are true or false)

$$P(A) = \text{degree of belief that } A \text{ is true}$$

The fundamental notion of probability is how well a proposition is supported by the evidence for it.

Both interpretations are consistent with Kolmogorov axioms.

Conditional probability, independence

Conditional probability of *A* given *B* (with *P(B) ≠ 0*):

$$P(A|B) = \frac{P(A \cap B)}{P(B)}$$

E.g. rolling dice: $P(n < 3 \,|\, n \text{ even}) = \frac{P((n<3) \cap n \text{ even})}{P(\text{even})} = \frac{1/6}{3/6} = \frac{1}{3}$

Subsets *A*, *B* independent if: $P(A \cap B) = P(A)P(B)$

If *A*, *B* independent, $P(A|B) = \frac{P(A)P(B)}{P(B)} = P(A)$

Bayes theorem

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)},$$

where **A** and **B** are events, **P(A|B)** is the conditional probability that event **A** occurs given that event **B** has already occurred

P(B|A) *has the same meaning and* ***P(A)*** *and* ***P(B)*** *are the marginal probabilities of event* ***A*** *and event* ***B*** *occurring respectively.*

Proof: from the definition of conditional probability we have,

$$P(A|B) = \frac{P(A \cap B)}{P(B)} \quad \text{and} \quad P(B|A) = \frac{P(B \cap A)}{P(A)}$$

but $P(A \cap B) = P(B \cap A)$, so

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)},$$

First published (posthumously) by the **Reverend Thomas Bayes (1702−1761)**

An example using Bayes' theorem

Suppose the probability (for anyone) to have a disease D is:

$$P(\mathrm{D}) = 0.001$$

$$P(\text{no D}) = 0.999$$

← prior probabilities, i.e., before any test carried out

Consider a test for the disease: result is + or -

$$P(+|\mathrm{D}) = 0.98$$

$$P(-|\mathrm{D}) = 0.02$$

← probabilities to (in)correctly identify a person with the disease

$$P(+|\text{no D}) = 0.03$$

$$P(-|\text{no D}) = 0.97$$

← probabilities to (in)correctly identify a healthy person

Suppose your result is +. How worried should you be?

Bayes' theorem example (cont.)

The probability to have the disease given a + test result is:

$$p(\mathrm{D}|+) = \frac{P(+|\mathrm{D})P(\mathrm{D})}{P(+|\mathrm{D})P(\mathrm{D}) + P(+|\text{no D})P(\text{no D})}$$

$$= \frac{0.98 \times 0.001}{0.98 \times 0.001 + 0.03 \times 0.999}$$

$$= 0.032 \qquad \leftarrow \text{posterior probability}$$

i.e. you're probably OK!

Frequentist Statistics – general philosophy

In frequentist statistics, probabilities are associated only with the data, i.e., outcomes of repeatable observations (shorthand: $\vec{x}$).

Probability = limiting frequency

Probabilities such as:

P (Higgs boson exists),
P (0.117 < α_s < 0.121),

etc. are either 0 or 1, but we don't know which.

The tools of frequentist statistics tell us what to expect, under the assumption of certain probabilities, about hypothetical repeated observations.

The preferred theories (models, hypotheses, ...) are those for which our observations would be considered 'usual'.

Bayesian Statistics – general philosophy

In Bayesian statistics, use subjective probability for hypotheses:

probability of the data assuming hypothesis *H* (the likelihood)

prior probability, i.e., before seeing the data

$$P(H|\vec{x}) = \frac{P(\vec{x}|H)\pi(H)}{\int P(\vec{x}|H)\pi(H)\,dH}$$

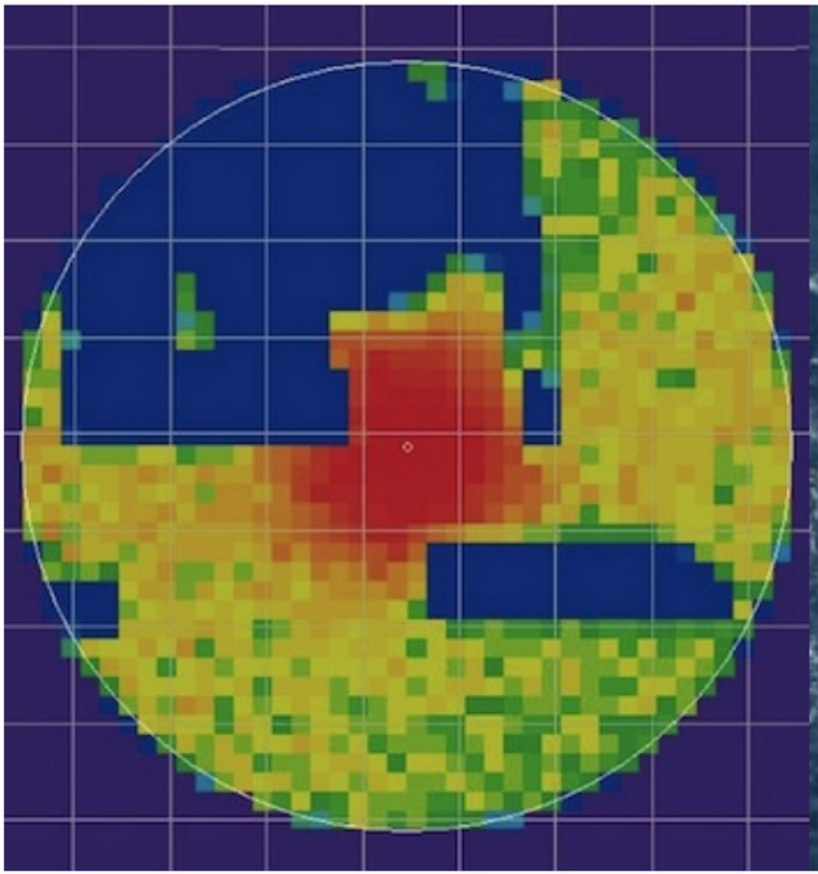
posterior probability, i.e., after seeing the data

normalization involves sum over all possible hypotheses

Bayes' theorem has an "if-then" character: If your prior probabilities were $\pi(H)$, then it says how these probabilities should change in the light of the data.

No unique prescription for priors (subjective!)

Here's *exactly* the same idea, in practice;

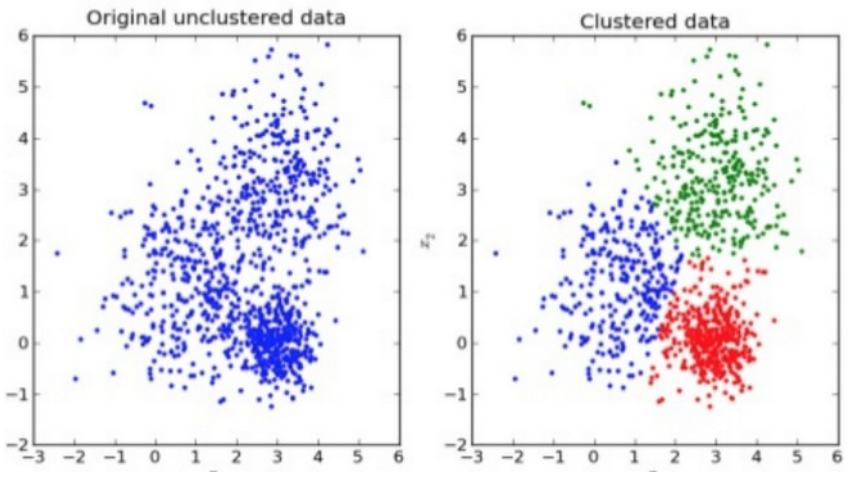


- During the search for Air France 447, from 2009-2011, knowledge about the black box location was described via probability – i.e. using Bayesian inference
- Eventually, the black box was found in the red area

http://faculty.washington.edu/kenrice/BayesIntroClassEpi2018.pdf

What does “machine learning” mean?

- **Machine learning** is a field of computer science that gives computer systems the ability to "learn" (i.e. progressively improve performance on a specific task) with data, without being explicitly programmed.
- Problems:
  - Supervised learning (classification & regression)
  - Clustering (unsupervised learning)
  - Dimensionality reduction
  - Reinforcement learning
  - Many others….



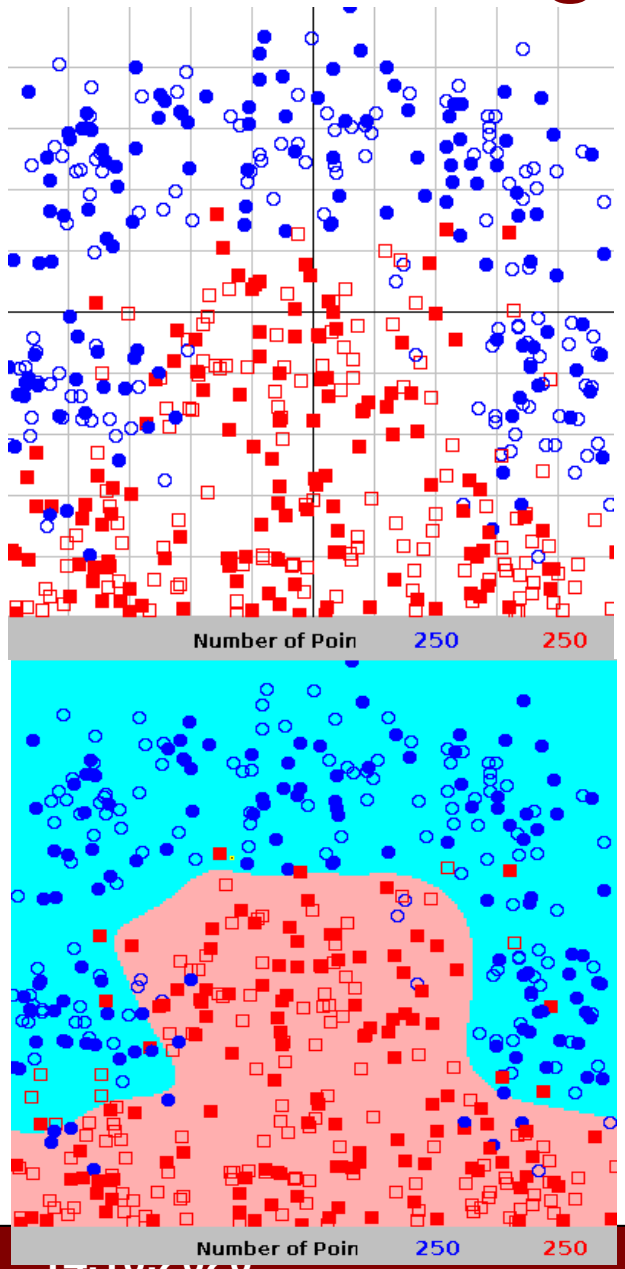
➢Unsupervised Learning

❑Technique of trying to find hidden structure in unlabeled data

➢Supervise Learning

❑Technique for creating a function from training data. The training data consist of pairs of input objects (typically vectors), and desired outputs.

How do the (supervised) machine learning algorithms work?

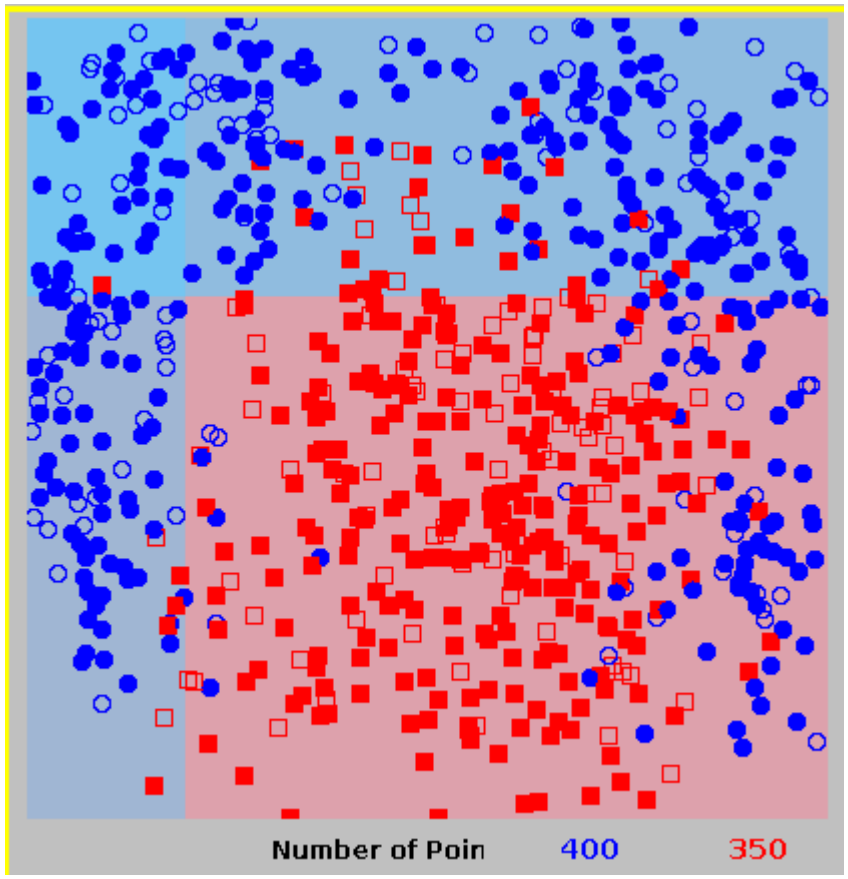


- We need **training data**, for which we know the correct answer, whether it's a signa or background. We divide the data into two samples: training and test.
- We find the best function ***f(x)*** which describes the probability, that a given event belongs to the class "signal". This is done by minimizing the loss function (for example χ^2).
- Different algorithms differ by: the class of function used as ***f(x)*** (linear, non-linear etc), loss function and the way it's minimized.
- All these algorithms try to approximate the unknown *Bayessian Decisive Function* (BDF) relying on the finit training sample.

BDF -an ideal classification function given by the unknown probability densities of signal and background.

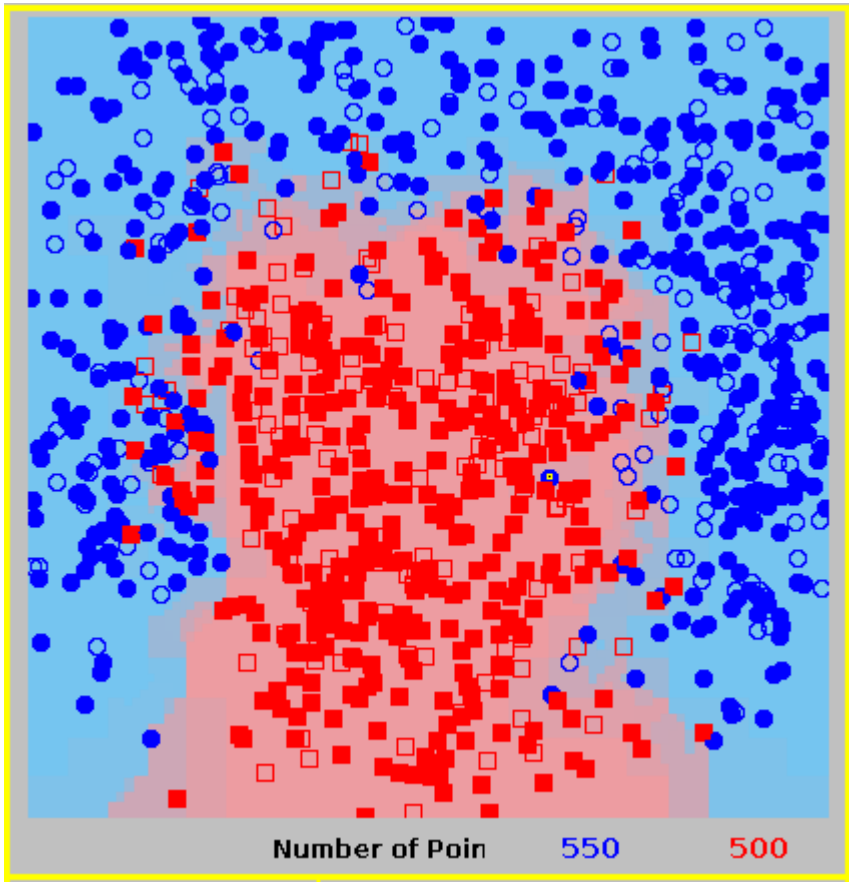
Cuts vs non-linear separation

Cuts



?

Non-linear separation



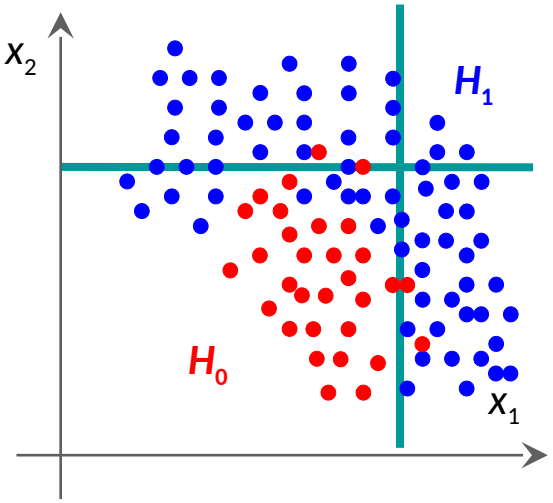
Neural Networks, boosted decision trees, and so on….

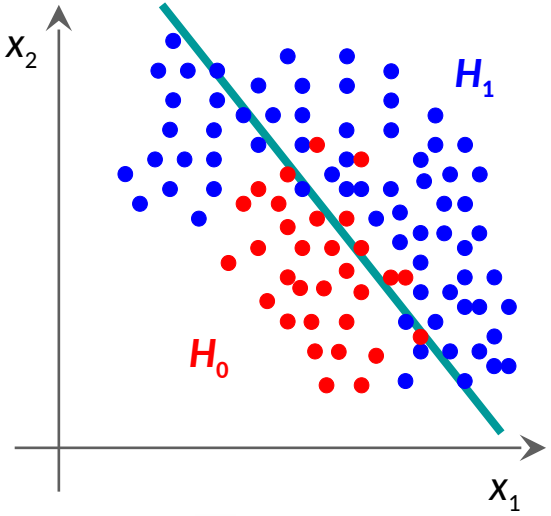
Types of algorithms

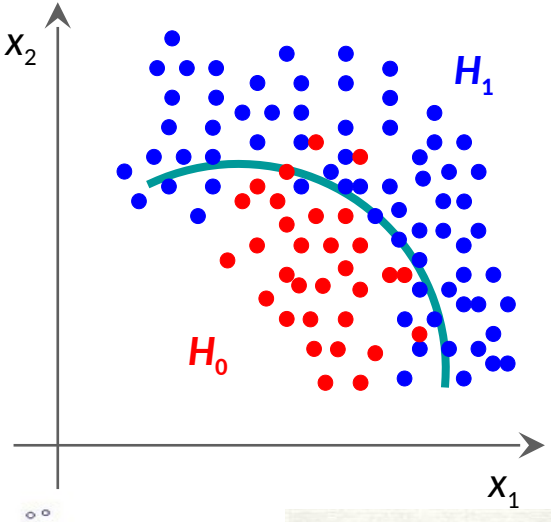
How to use the information available

Classification: find a function ***f(x1,x2)*** giving the probability, that a given data point belongs to a given class (signal vs background).

Simple cuts (easy and intuitive)	Linear (fast and stable)	Non-linear (most effective)

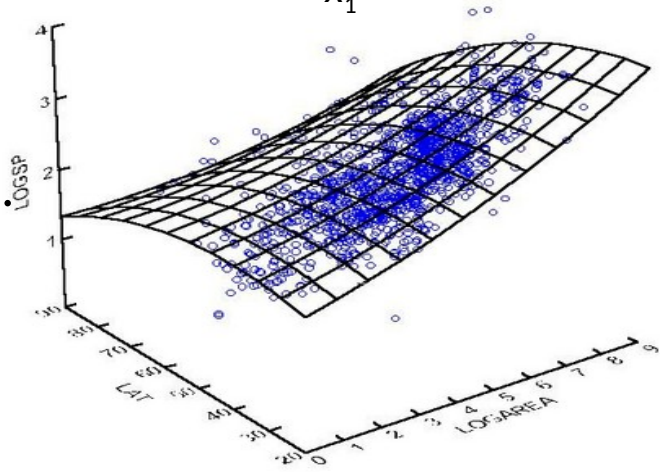


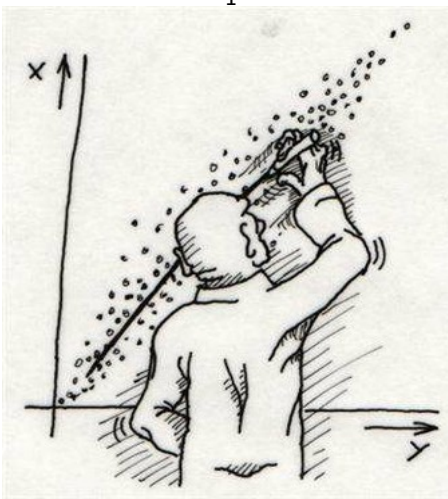




Regression: fit a continuous function

(find particle energy from calo readouts).





Classification

A Bayes classifier (optimal classifier):

$$p(S|x) = \frac{p(x|S)\,p(S)}{p(x|S)\,p(S) + p(x|B)\,p(B)}$$

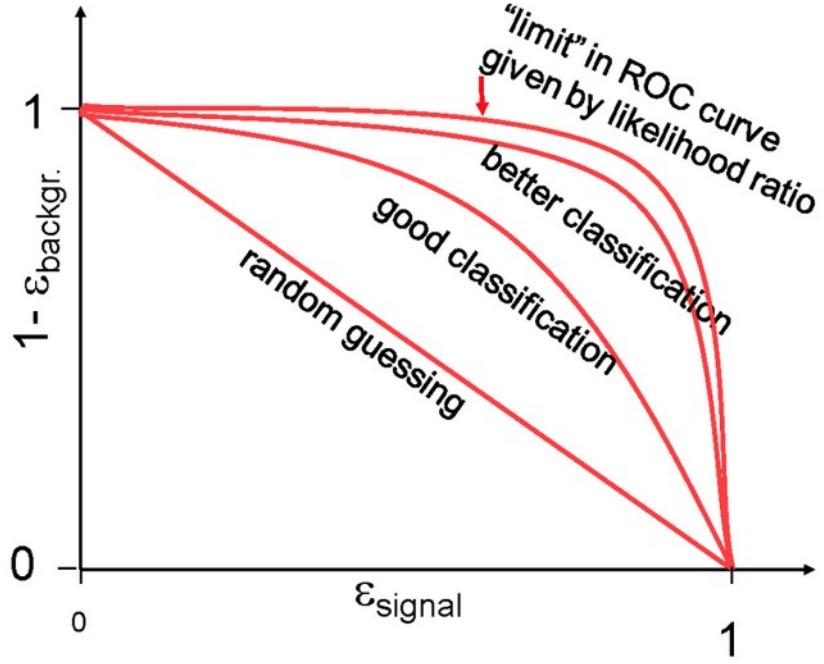
where ***S*** is associated with *y* = **1** and ***B*** with *y* = **0**. **Bayes classifier** *accepts events x if p(**S**|x) > **cut** as belonging to **S**.*

We need to approximate probability distributions *P(**x**|**S**)* and *P(**x**|**B**).*

- *If your goal is to **classify objects** with the fewest errors, then the **Bayes classifier** is the **optimal** solution.*
- Consequently, if you have a classifier known to be ***close*** to the **Bayes limit**, then *any* other classifier, *however sophisticated*, can ***at best*** be only marginally better than the one you have.
  - =>If your problem is **linear** you don't gain anything by using sophisticated **Neural Network**
- *All* classification methods, such as all we will be talking about, are different numerical approximations of the Bayes classifier.

ROC curve

- ROC (Receiver Operation Characteristic) curve was first used to calibrate radars.
- Shows the background rejection ($1-\varepsilon_B$) vs signal efficiency ε_B. Shows how good the classifier is.
- The integral of ROC could be a measure of the classifier quality:



Integral(ROC) = ½ – random

Integral(ROC) = 1 - ideal

Practical applications

A Short List of Multivariate Methods

- Cuts
- Linear Discriminants (like Fisher)
- Naive Bayes (Likelihood Discriminant)
- Kernel Density Estimation
- Decision Trees
- Neural Networks
- Bayesian Neural Networks
- Genetic Algorithms
- Support Vector Machines

- And many, many others..... I want to present briefly just few of them.

We will talk in the first lectures about:

- Simple ML linear methods:
  - Cuts
  - Fisher linear discriminant
  - Naive Bayes
  - Principal Component Analysis, PCA
  - Independent Component Analysis, ICA

Google Colab & Github
short tutorial

Both will be explained in detail on the course of the python programming language.

Here only a very brief introduction is given.

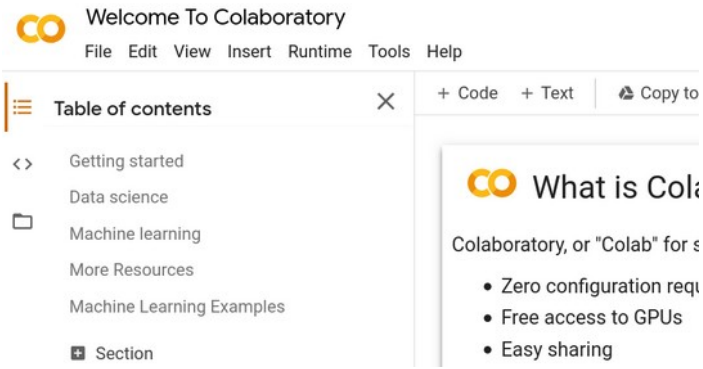
How to run python?

- You can install python on your laptop…
- … but you can also use FREE framework over internet to run your notebooks:
  - You do not need to install anything
  - You can share your code
  - You get free CPU or even GPU (Graphic Processor Units), which make the code running much faster.

- Notebooks available on WEB:
  - **Google Colab (I use it)** **https://colab.research.google.com**
  - And some others, see: https://analyticsindiamag.com/5-alternatives-to-google-colab-for-data-scientists/

Let’s play with Google Colaboratory

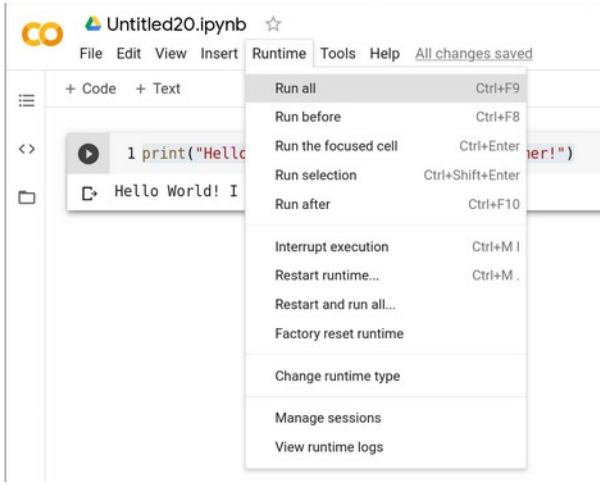
- Open: https://colab.research.google.com
- Sign in with your google account
- Click: File → New Notebook



- Paste the following code:

```
print("Hello World! I am an IPython programmer!")
```

- Run the code:

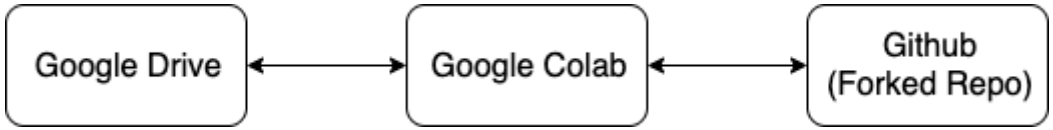


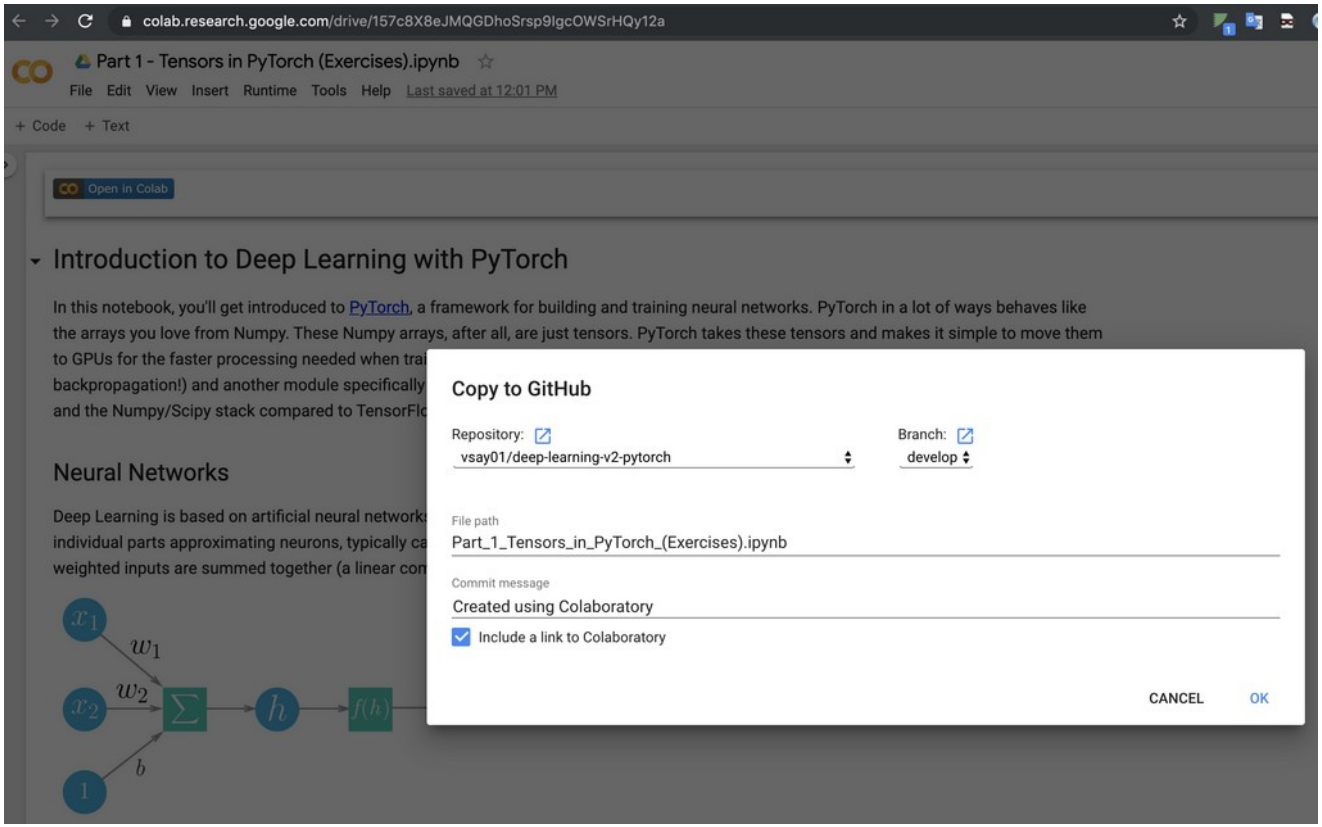
Easy???

How to use Google Colab + Github

- Very nice tutorial:

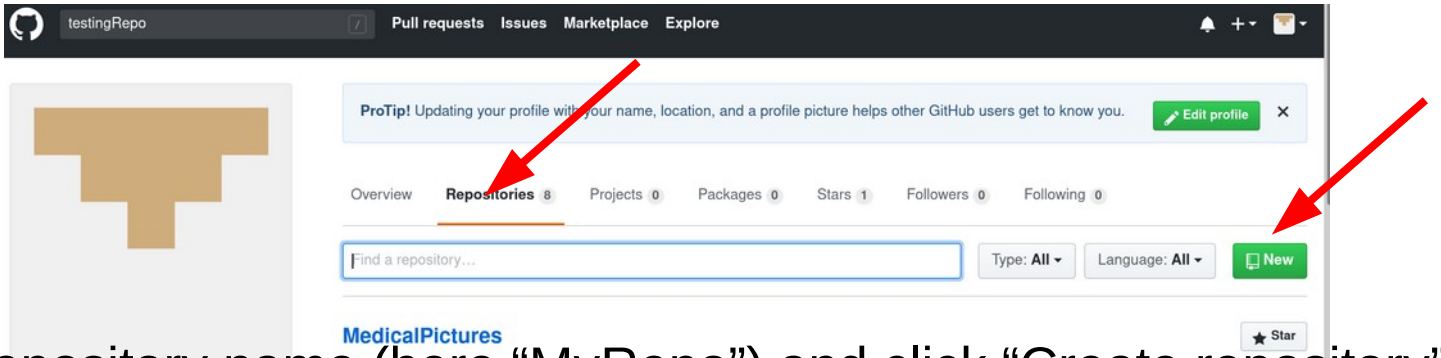
  https://towardsdatascience.com/google-drive-google-colab-github-dont-just-read-do-it-5554d5824228



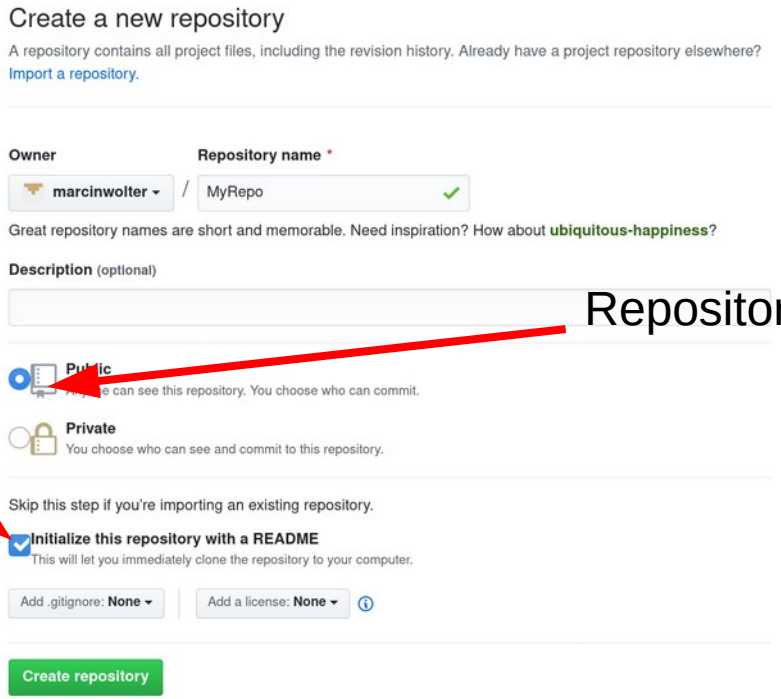


Save a copy of your iPython notebook to Github

- Go to https://github.com/ and **sign up**
- Create new repository by clicking “Repositories” and then “New”:

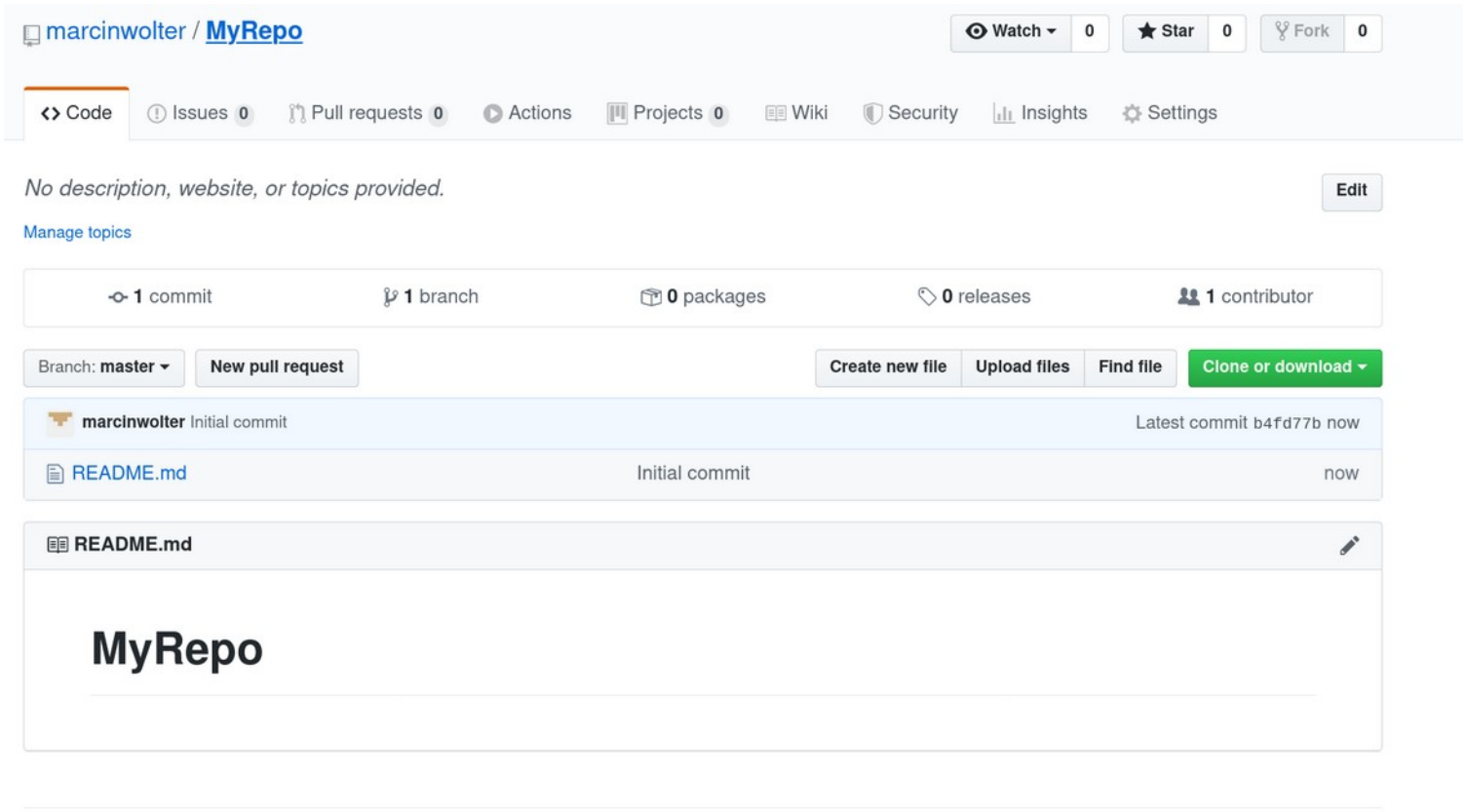


- Type the repository name (here “MyRepo”) and click “Create repository”:



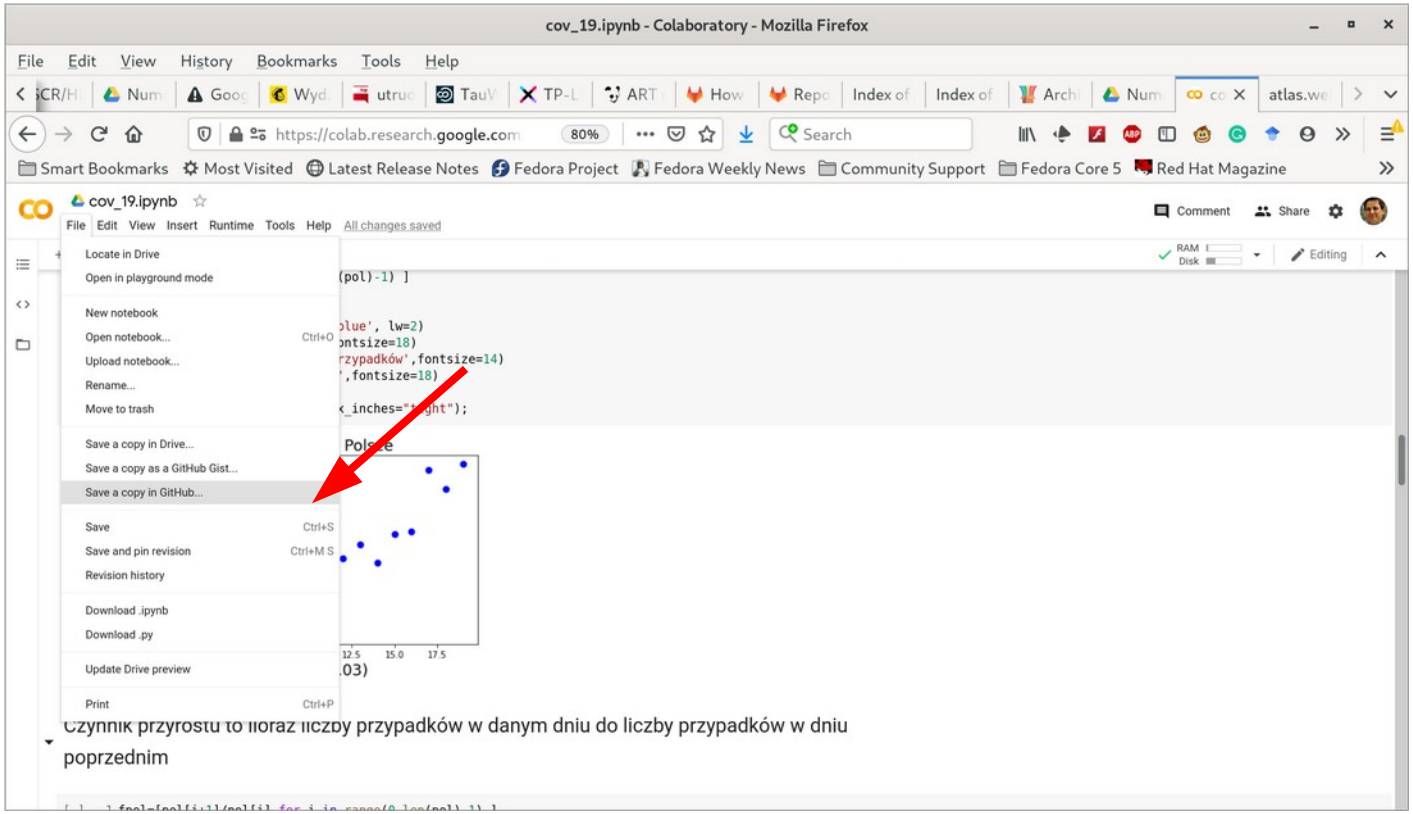
Repo initialized with a README file

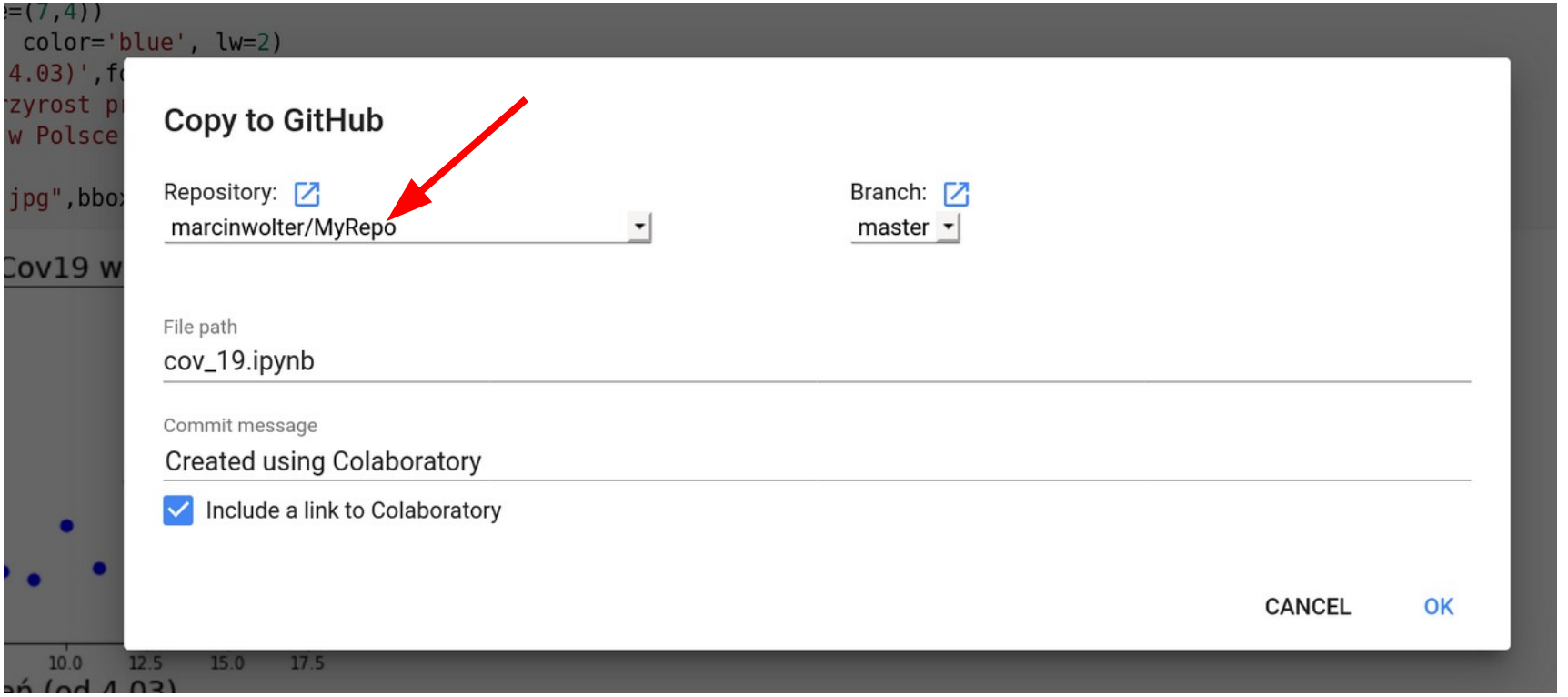
Repository is public



You have created a new repository with a README.md file inside

Now go to Google Colab and click: File → Save a copy in Github





By clicking choose the proper repository (here “MyRepo”)
And click OK to create a copy.

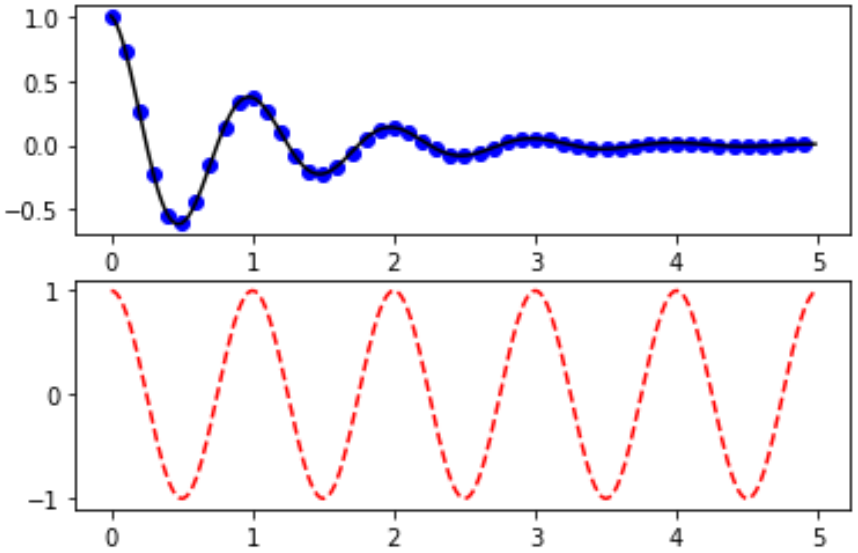
GOOD LUCK!!!!

Run example from Github

- Open a github repository:

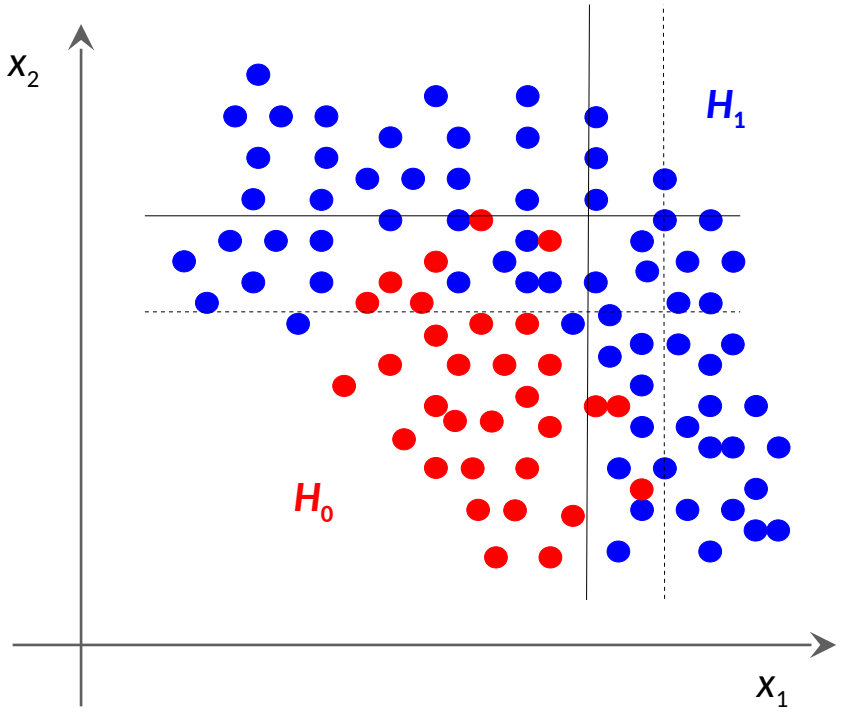
  https://github.com/marcinwolter/MachineLearning2020
- Open a python notebook: VisualizationCode.ipynb
- Open it in Google Colab: 

- Run it!!!! There are examples how to use graphics
- Try to modify and play with the code. It shows how to use graphics.



Back to Machine Learning

Cuts



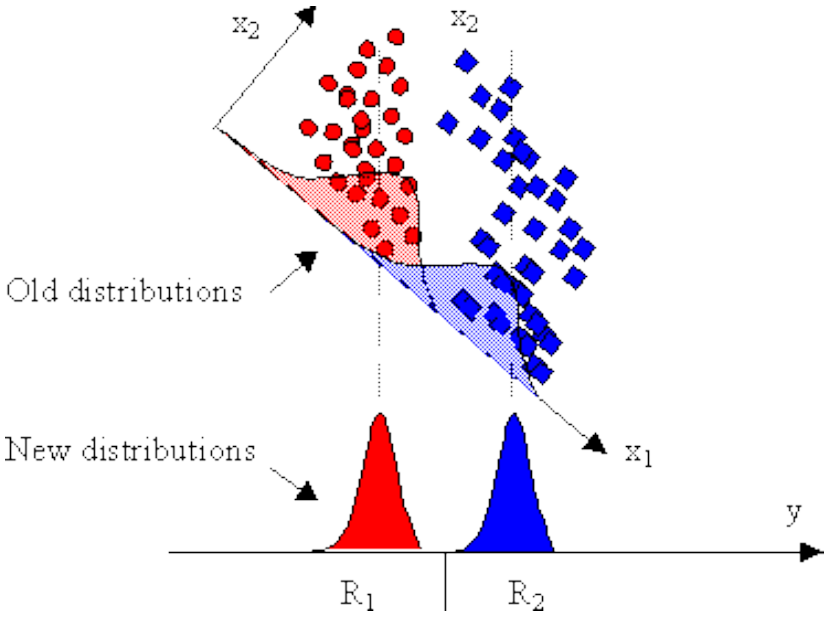
Optimization of cuts:

- Move cuts as long as we get the optimal signal vs. background selection. For a given signal efficiency we find the best background rejection → we get the entire ROC curve.
- Optimization methods:
  - Brute force
  - Genetic algorithms
  - Many others...

Fisher discriminants

LDA, Linear Discriminat Analysis

Projection to one dimension, than discrimination



Equivalent to linear separation

We choose a projection vector in such a way, that the separation is maximized.

Assumptions for new basis:

- Maximize distance between projected class means
- Minimize projected class variance

Method introduced by Fisher in 1936.
Optimal separation for Gaussian distributions.

Fisher Linear Discriminant Analysis

Variance: $$\Sigma_{XX} = Var(X) = E\{(X-\mu)^T(X-\mu)\}$$

Objective

$$argmax_w J(w) = \frac{w^T S_B w}{w^T S_W w}$$

$$m_i = \frac{1}{n_i}\sum_{x \in C_i} x$$

$$S_B = (m_2 - m_1)(m_2 - m_1)^T$$ Variance Between classes

$$S_W = \sum_j^2 \sum_{x \in C_j} (x - m_j)(x - m_j)^T$$ Variance Within class

Algorithm

1. Compute class means
2. Compute $w = S_W^{-1}(m_2 - m_1)$
3. Project data $y = w^T x$

Fisher's linear discriminant

The terms *Fisher's linear discriminant* and *LDA* are often used interchangeably, although Fisher's original article *The Use of Multiple Measures in Taxonomic Problems* (1936) actually describes a slightly different discriminant, which does not make some of the assumptions of LDA such as normally distributed classes or equal class covariances.

Suppose two classes of observations have means $\vec{\mu}_{y=0}, \vec{\mu}_{y=1}$ and covariances $\Sigma_{y=0}, \Sigma_{y=1}$. Then the linear combination of features $\vec{w} \cdot \vec{x}$ will have means $\vec{w} \cdot \vec{\mu}_{y=i}$ and variances $\vec{w}^T \Sigma_{y=i} \vec{w}$ for $i = 0,1$. Fisher defined the separation between these two distributions to be the ratio of the variance between the classes to the variance within the classes:

$$S = \frac{\sigma^2_{between}}{\sigma^2_{within}} = \frac{(\vec{w} \cdot \vec{\mu}_{y=1} - \vec{w} \cdot \vec{\mu}_{y=0})^2}{\vec{w}^T \Sigma_{y=1} \vec{w} + \vec{w}^T \Sigma_{y=0} \vec{w}} = \frac{(\vec{w} \cdot (\vec{\mu}_{y=1} - \vec{\mu}_{y=0}))^2}{\vec{w}^T (\Sigma_{y=0} + \Sigma_{y=1}) \vec{w}}$$

This measure is, in some sense, a measure of the signal-to-noise ratio for the class labelling. It can be shown that the maximum separation occurs when

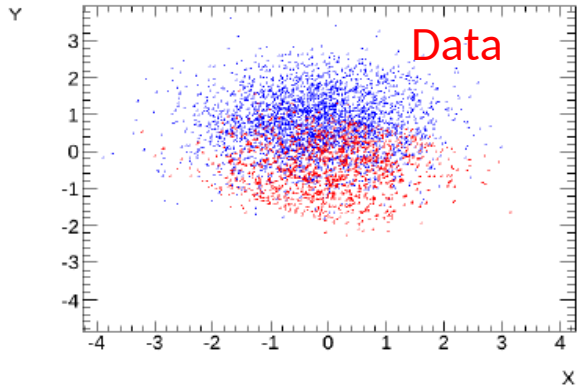
$$\vec{w} = (\Sigma_{y=0} + \Sigma_{y=1})^{-1} (\vec{\mu}_{y=1} - \vec{\mu}_{y=0})$$

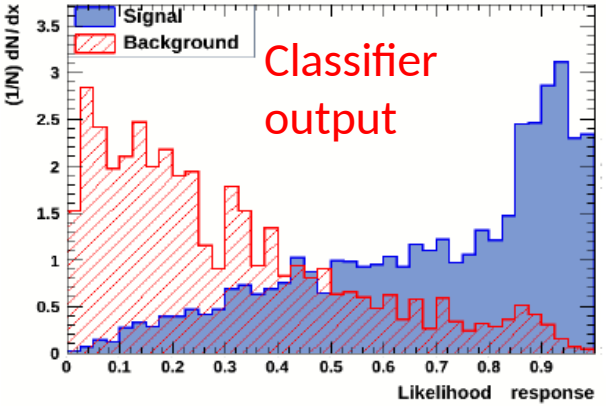
When the assumptions of LDA are satisfied, the above equation is equivalent to LDA.

Be sure to note that the vector $\vec{w}$ is the normal to the discriminant hyperplane. As an example, in a two dimensional problem, the line that best divides the two groups is perpendicular to $\vec{w}$.

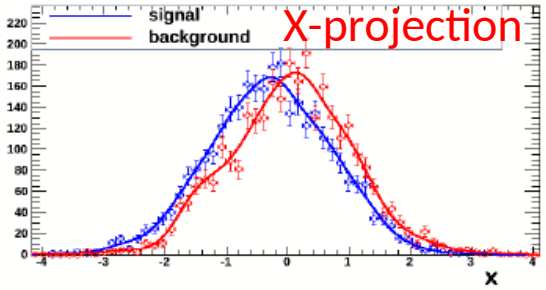
Generally, the data points are projected onto $\vec{w}$. However, to find the actual plane that best separates the data, one must solve for the bias term b in $w^T \mu_1 + b = -(w^T \mu_2 + b)$.

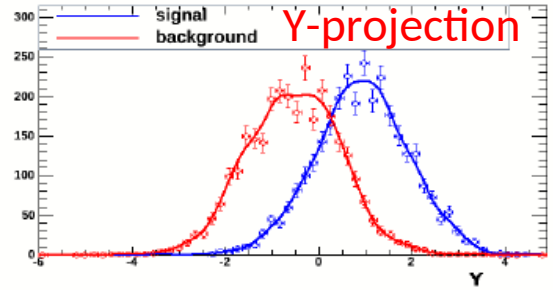
Naive Bayes classifier





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





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

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

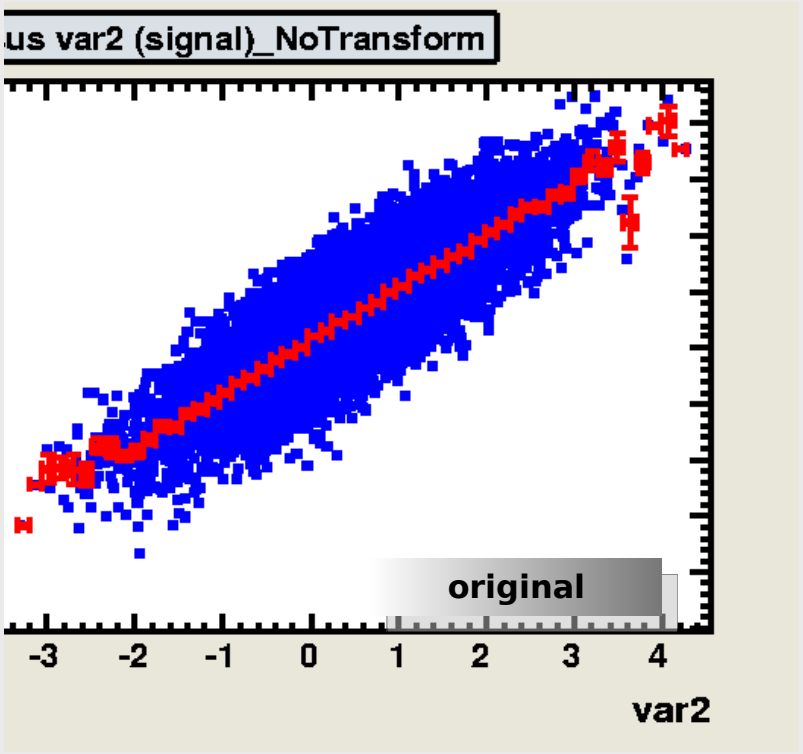
“Naive” assumption: $P(x_i|y, x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n) = P(x_i|y),$

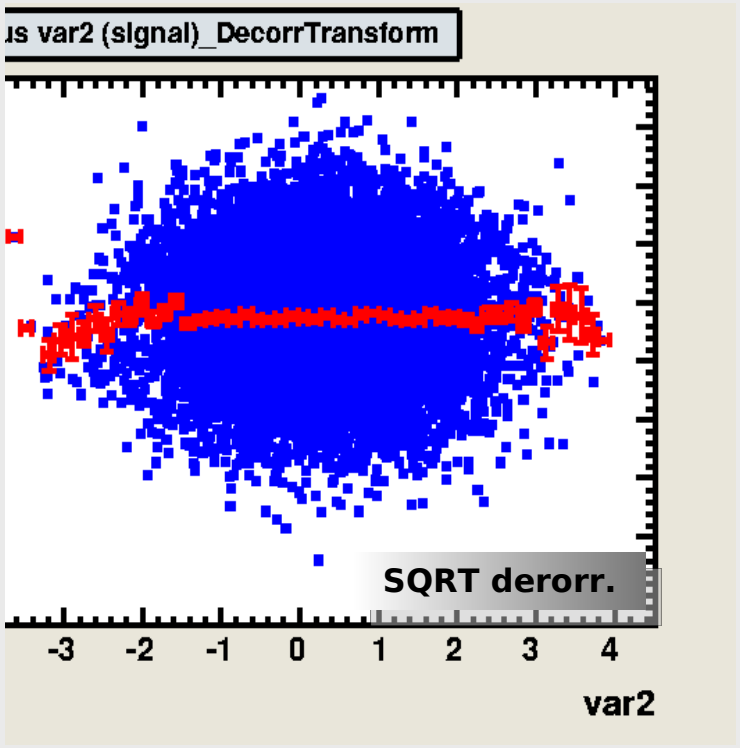
$$P(y \mid x_1, \ldots, x_n) = \frac{P(y)\prod_{i=1}^{n} P(x_i \mid y)}{P(x_1, \ldots, x_n)}$$

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

Decorrelation

- **Removes correlation between variables by a rotation in the space of variables**





Correlation matrix

The correlation matrix refers to the symmetric array of numbers

$$\mathbf{R} = \begin{pmatrix} 1 & r_{12} & r_{13} & \cdots & r_{1p} \\ r_{21} & 1 & r_{23} & \cdots & r_{2p} \\ r_{31} & r_{32} & 1 & \cdots & r_{3p} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ r_{p1} & r_{p2} & r_{p3} & \cdots & 1 \end{pmatrix}$$

where

$$r_{jk} = \frac{s_{jk}}{s_j s_k} = \frac{\sum_{i=1}^{n}(x_{ij} - \bar{x}_j)(x_{ik} - \bar{x}_k)}{\sqrt{\sum_{i=1}^{n}(x_{ij} - \bar{x}_j)^2}\sqrt{\sum_{i=1}^{n}(x_{ik} - \bar{x}_k)^2}}$$

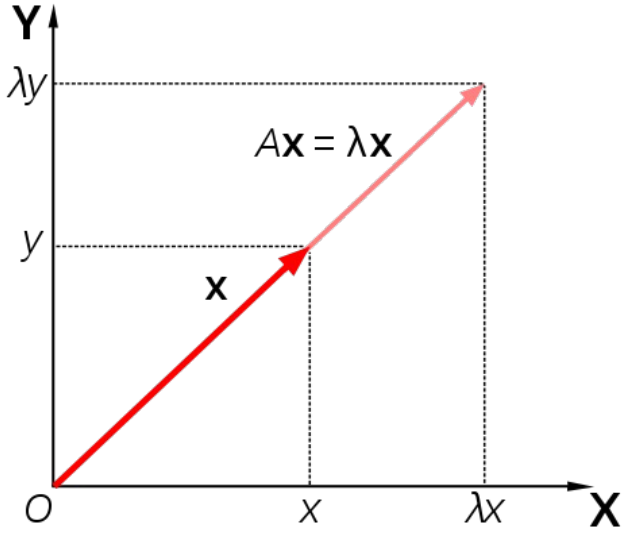
is the Pearson correlation coefficient between variables $\mathbf{x}_j$ and $\mathbf{x}_k$.

Eigenvalues and eigenvectors

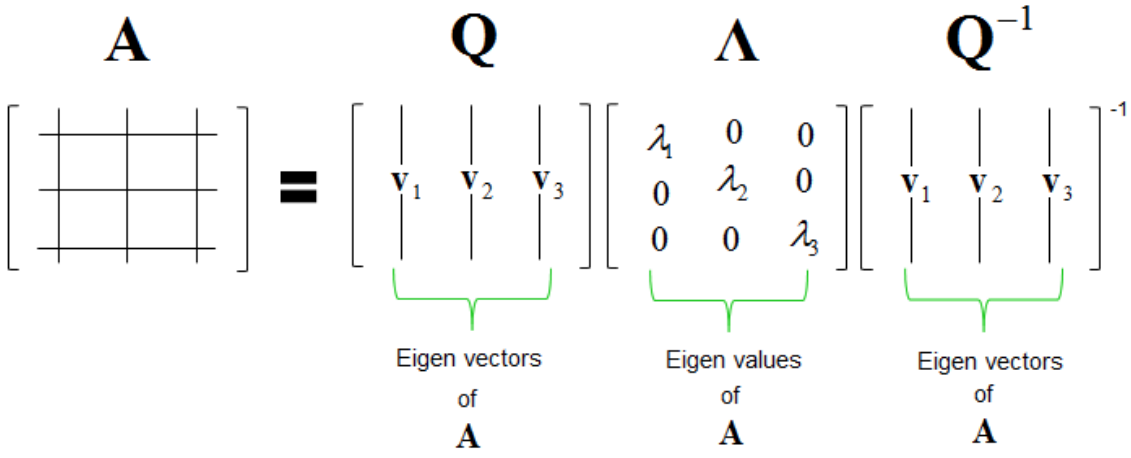
In essence, an eigenvector v of a linear transformation T is a non-zero vector that, when T is applied to it, does not change direction. Applying T to the eigenvector only scales the eigenvector by the scalar value λ, called an eigenvalue. This condition can be written as the equation

$$T(\mathbf{v}) = \lambda \mathbf{v}$$

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 complex.



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



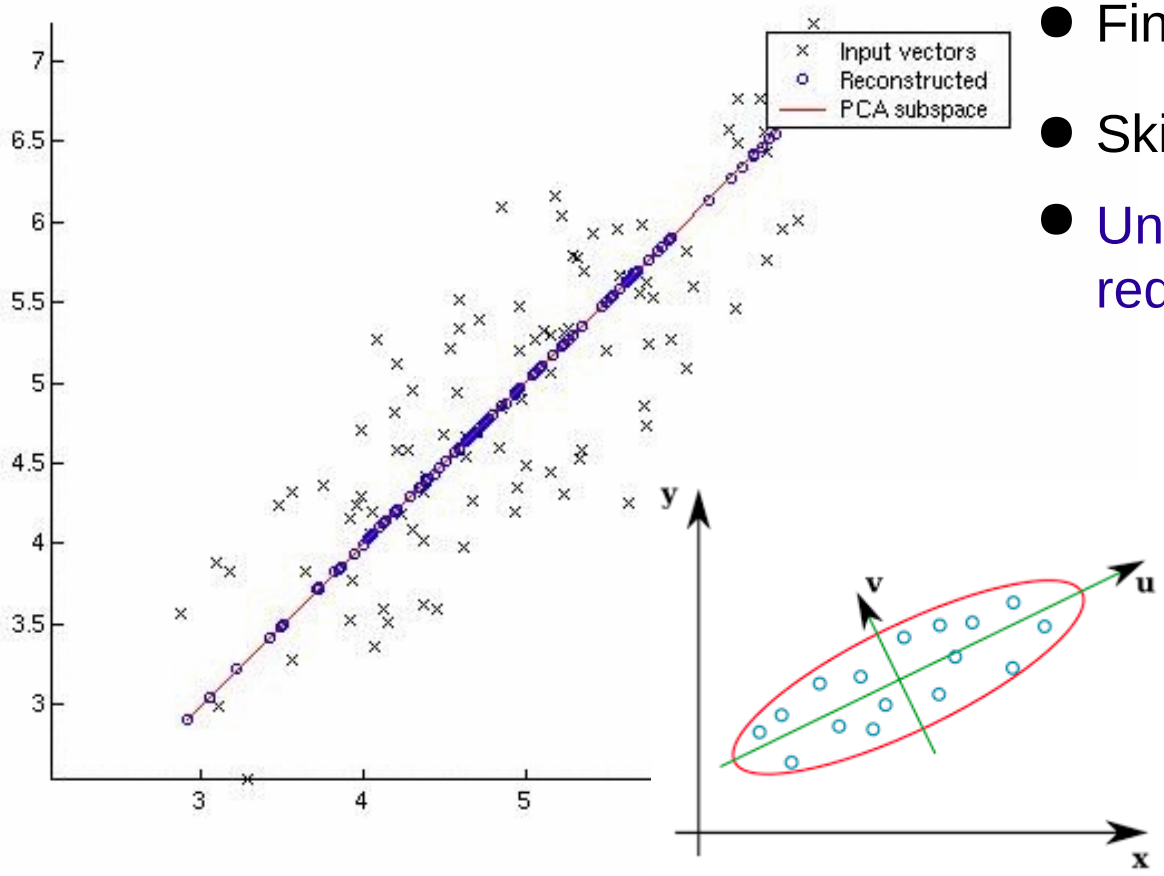
Eigendecomposition of a matrix

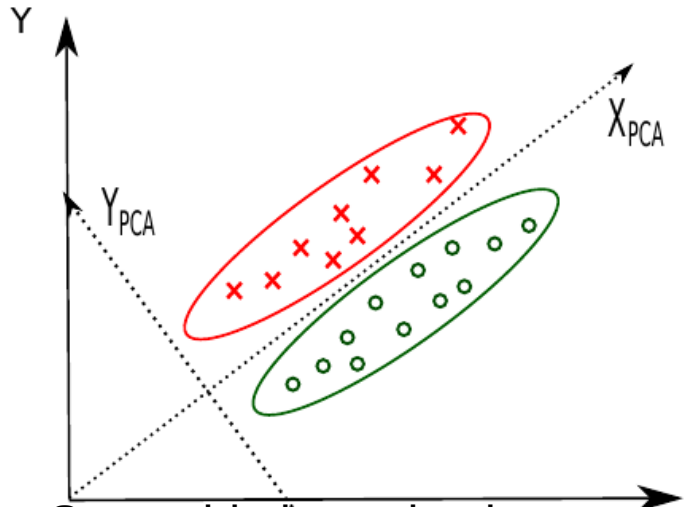
Principal Component Analysis - PCA

- 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 PCA doesn't help...

Covariance Matrix

- Let X be a p-variate random vector. The covariance matrix of X is defined as:

$$\Sigma_{XX} = Var(X) = E\{(X-\mu)^T(X-\mu)\}$$

$$= \begin{pmatrix} Var(X_1) & Cov(X_1,X_2) & \dots & Cov(X_1,X_p) \\ Cov(X_2,X_1) & Var(X_2) & \dots & Cov(X_2,X_p) \\ \vdots & \vdots & \ddots & \vdots \\ Cov(X_p,X_1) & Cov(X_p,X_2) & \dots & Var(X_p) \end{pmatrix}.$$

Where:

$$Var(X_i) = E\{(X_i - \mu(X_i)) \cdot (X_i - \mu(X_i))\}$$
$$Cov(X_i,Y_j) = E\{(X_i - \mu(X_i)) \cdot (Y_j - \mu(Y_j))\}$$

Correlation or covariance matrix?

- Mean-centering is unnecessary if performing a principal components analysis on a correlation matrix, as the data are already centered after calculating correlations.
- We tend to use the covariance matrix when the variable scales are similar and the correlation matrix when variables are on different scales.

$$\mathrm{Corr}(X,Y) = \frac{\mathrm{Cov}(X,Y)}{\sigma_x \sigma_y}$$

Covarianced normalized by Standard Deviation

Correlation between X and Y

Standard deviation of X

Standard deviation of Y

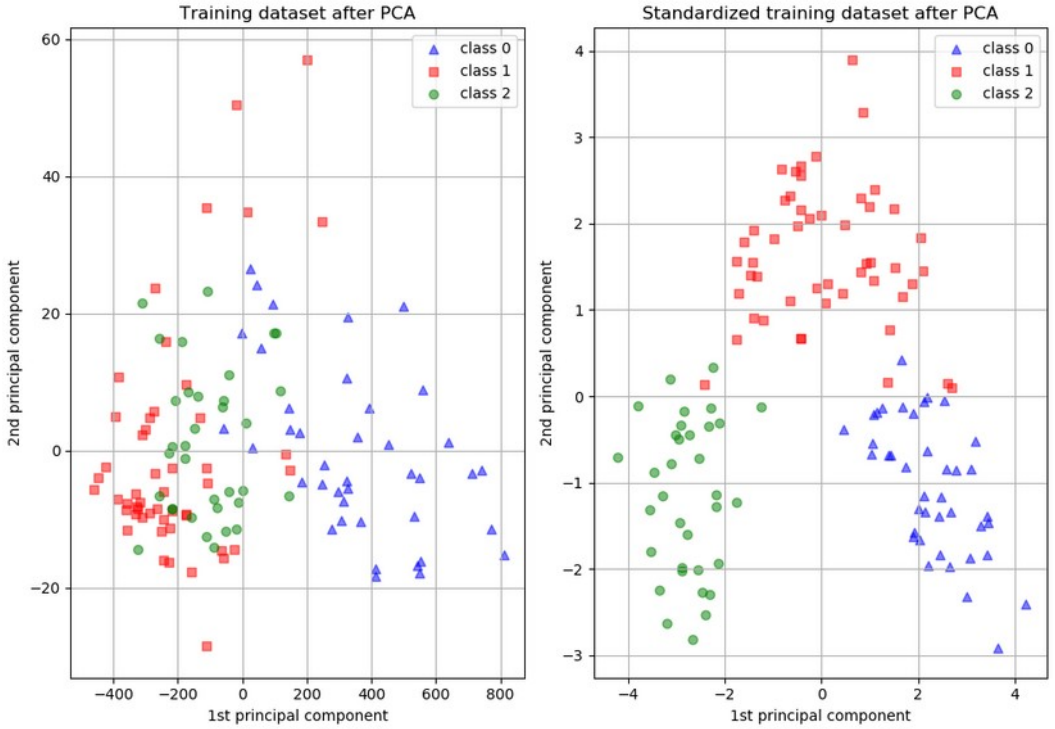
Applications

- **Uses:**
  - Data Visualization
  - Data Reduction
  - Data Classification
  - Noise Reduction

- **Examples**:
  - How many unique “sub-sets” are in the sample?
  - How are they similar / different?
  - Which measurements are needed to differentiate?
  - How to best present what is “interesting”?
  - Which “sub-set” does this new sample rightfully belong?

What happens if we perform PCA without normalization? Why do we normalize data?

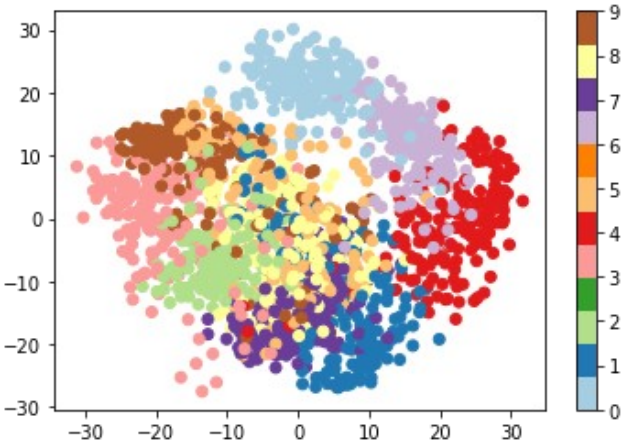
In PCA we are interested in the components that maximize the variance. If one component (e.g. human height) varies less than another (e.g. weight) because of their respective scales (meters vs. kilos), PCA might determine that the direction of maximal variance more closely corresponds with the 'weight' axis, if those features are not scaled. As a change in height of one meter can be considered much more important than the change in weight of one kilogram, this is clearly incorrect.

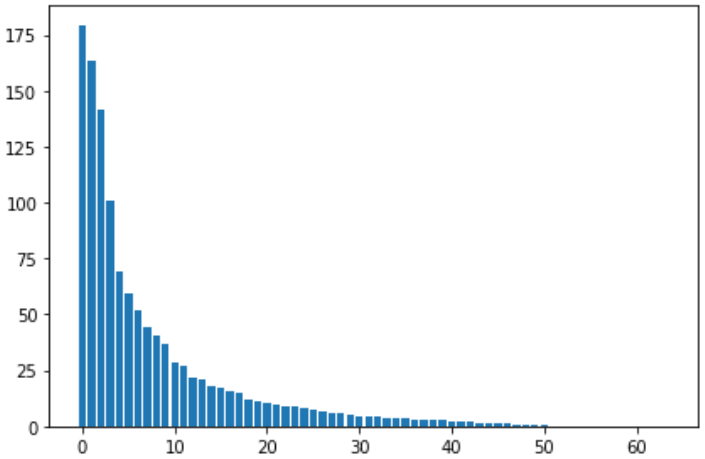


The dataset used is the Wine Dataset available at UCI. This dataset has continuous features that are heterogeneous in scale due to differing properties that they measure (i.e alcohol content, and malic acid).

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

Summary

- What is machine learning
- Classifiers:
  - Cuts
  - Naive Bayes
  - Fisher Linear Discriminants
- Simple unsupervised methods – PCA and decorellation.