



Machine Learning for Data Science

Philippe Besse, Sébastien Gerchinovitz, Béatrice Laurent

Machine Learning for Data Science
CERFACS – May 2019

Credits: Aurélien Garivier

Outline

Machine Learning: when Artificial Intelligence meets Big Data

The Learning Models

Machine Learning Methodology

Artificial Intelligence (AI): Definition

Intelligence exhibited by machines

- emulate cognitive capabilities of humans (big data: humans learn from abundant and diverse sources of data).
- a machine mimics "cognitive" functions that humans associate with other human minds, such as "learning" and "problem solving".

Ideal "intelligent" machine =

flexible rational agent that perceives its environment and takes actions that maximize its chance of success at some goal.

Founded on the claim that human intelligence

"can be so precisely described that a machine can be made to simulate it."

Artificial Intelligence: Tension

Operational goals

- Autonomous robots for not-too-specialized tasks
- In particular, vision + understand and produce language

Tension between operational and philosophical goals

- As machines become increasingly capable, facilities once thought to require intelligence are removed from the definition. For example, optical character recognition is no longer perceived as an exemplar of "artificial intelligence"; having become a routine technology.
- Capabilities still classified as AI include advanced Chess and Go systems and self-driving cars.

Machine Learning (ML): Definition

Arthur Samuel (1959)

Field of study that gives computers the ability to learn without being explicitly programmed

Tom M. Mitchell (1997)

A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T , as measured by P , improves with experience E .

ML: Learn from and make predictions on **data**

- Algorithms operate by building a model from **example** inputs in order to make data-driven **predictions or decisions**...
- ...rather than following strictly static program instructions: useful when designing and programming explicit algorithms is unfeasible or poorly efficient.

Within Data Analytics

- Machine Learning used to devise complex models and algorithms that lend themselves to **prediction** - in commercial use, this is known as *predictive analytics*.
- www.sas.com: "Produce reliable, repeatable decisions and results" and uncover "hidden insights" through learning from historical **relationships and trends** in the data.
- evolved from the study of pattern recognition and computational learning theory in artificial intelligence.

Machine Learning: Typical Problems

- spam filtering, text classification
- optical character recognition (OCR)
- search engines
- recommendation platforms
- speech recognition software
- computer vision
- bio-informatics, DNA analysis, medicine
- etc.

For each of these tasks, it is possible but very inefficient to write an explicit program reaching the prescribed goal.

It proves much more succesful to have a machine infer what the good decision rules are.

Related Fields

- **Computational Statistics:** focuses in prediction-making through the use of computers together with statistical models (ex: Bayesian methods).
- **Statistical Learning:** ML by statistical methods, with statistical point of view (probabilistic guarantees: consistency, oracle inequalities, minimax)
→ more focused on *correlation*, less on *causality*
- **Data Mining** (unsupervised learning) focuses more on exploratory data analysis: discovery of (previously) unknown properties in the data. This is the analysis step of Knowledge Discovery in Databases.
- Importance of **probability**- and **statistics**-based methods → **Data Science** (Michael Jordan)
- Strong ties to **Mathematical Optimization**, which delivers methods, theory and application domains to the field

[Src: Bouzeghoub, Mastodons: *Une approche interdisciplinaire des Big Data*]

Qu'est-ce qu'une (très grande) masse de données ?



VLDB

Big Data

Very Big Data

Data Deluge

Massive Data

Data Masses

Data inflation

2

Unit	Size	What it means
Bit (b)	1 or 0	Short for "binary digit", after the binary code (1 or 0) computers use to store and process data
Byte (B)	8 bits	Enough information to create an English letter or number in computer code. It is the basic unit of computing
Kilobyte (KB)	1,000, or 2^{10} , bytes	From "thousand" in Greek. One page of typed text is 2KB
Megabyte (MB)	1,000KB; 2^{20} bytes	From "large" in Greek. The complete works of Shakespeare total 5MB. A typical pop song is about 4MB
Gigabyte (GB)	1,000MB; 2^{30} bytes	From "giant" in Greek. A two-hour film can be compressed into 1-2GB
Terabyte (TB)	1,000GB; 2^{40} bytes	From "monster" in Greek. All the catalogued books in America's Library of Congress total 15TB
Petabyte (PB)	1,000TB; 2^{50} bytes	All letters delivered by America's postal service this year will amount to around 5PB. Google processes around 1PB every hour
Exabyte (EB)	1,000PB; 2^{60} bytes	Equivalent to 10 billion copies of <i>The Economist</i>
Zettabyte (ZB)	1,000EB; 2^{70} bytes	The total amount of information in existence this year is forecast to be around 1.2ZB
Yottabyte (YB)	1,000ZB; 2^{80} bytes	Currently too big to imagine

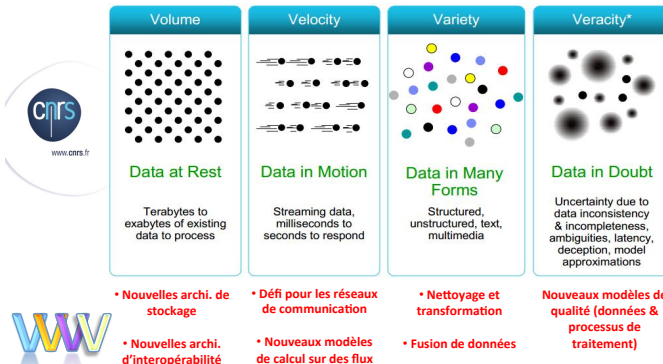
The prefixes are set by an international group, the International Bureau of Weights and Measures. Yotta and Zetta were added in 1991; terms for larger amounts have yet to be established.

Source: *The Economist*

Grandes Conf du domaine: VLDB, XLDB, ICDE, EDBT, ...

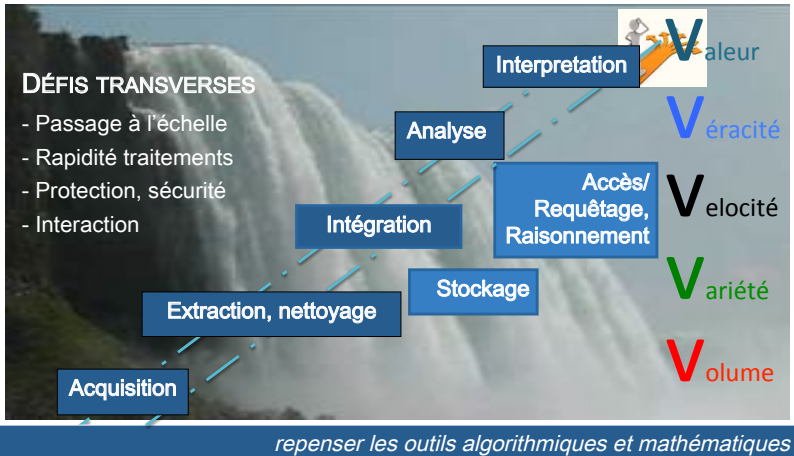
[Src: Bouzeghoub, Mastodons: *Une approche interdisciplinaire des Big Data*]

Complexité multidimensionnelle des Big Data



<http://www.datasciencecentral.com/profiles/blogs/data-veracity>

Défis accompagnant les chgts

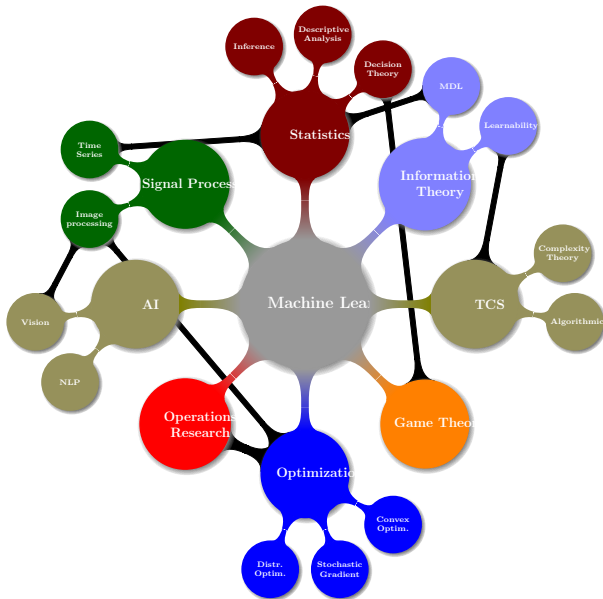


inspired by "Big Data and Its Technical Challenges, Communications of the ACM, July 2014, vol 57, n°7", © H.V. Jagadish et al.

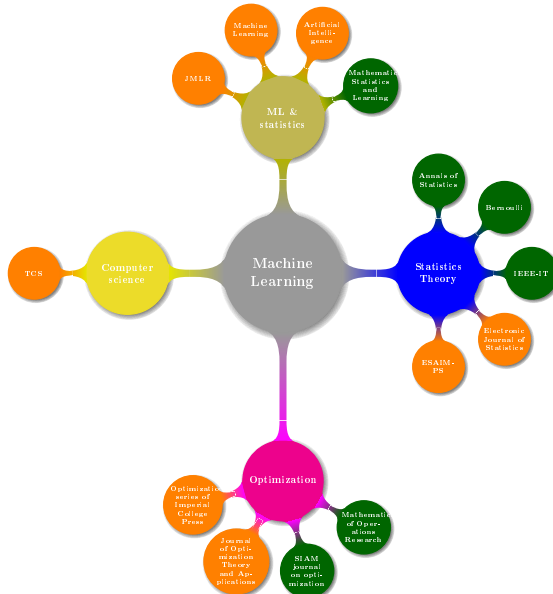
Machine Learning and Statistics

- Data analysis (inference, description) is the goal of statistics for long.
- Machine Learning has more **operational** goals (ex: consistency is important in the statistics literature, but often makes little sense in ML).
Models (if any) are *instrumental*.
Ex: linear model (nice mathematical theory) vs Random Forests.
- Machine Learning/big data: no separation between statistical modelling and optimization (in contrast to the statistics tradition).
- In ML, data is often here before (unfortunately)
- No clear separation (statistics evolves as well).

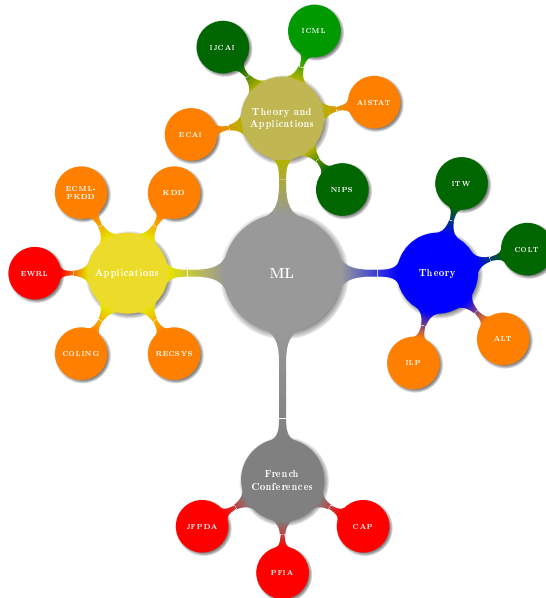
ML and its neighbors



ML journals



ML conferences



Outline

Machine Learning: when Artificial Intelligence meets Big Data

The Learning Models

Machine Learning Methodology

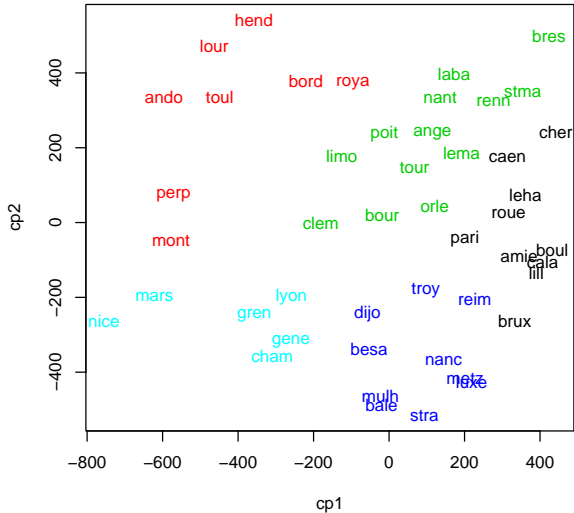
What ML is composed of



Unsupervised Learning

- (many) observations on (many) individuals
- need to have a simplified, structured overview of the data
- *taxonomy*: untargeted search for *homogeneous clusters* emerging from the data
- Examples:
 - customer segmentation
 - image analysis (recognizing different zones)
 - exploration of data

Example



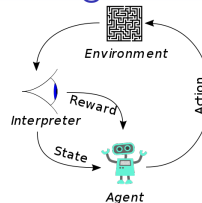
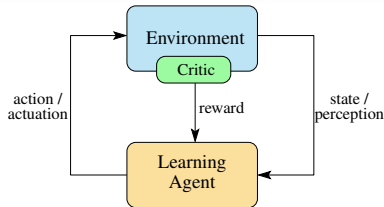
Supervised Learning

- observations = pairs (X_i, Y_i)
- goal = learn to *predict* Y_i given X_i
- regression (when Y is continuous)
- classification (when Y is discrete)
- statistical technique: linear models, and much more!

Example: Character Recognition

Input space \mathcal{X} Output space \mathcal{Y} Joint distribution $P(x, y)$	64×64 images $\{0, 1, \dots, 9\}$?
Prediction function $h \in \mathcal{H}$ Risk $R(h) = P(h(X) \neq Y)$	
Sample $\{(x_i, y_i)\}_{i=1}^n$ Empirical risk $\hat{R}_n(h) = \frac{1}{n} \sum_{i=1}^n \mathbb{1}\{h(x_i) \neq y_i\}$	MNIST dataset
Learning algorithm $\phi_n : (\mathcal{X} \times \mathcal{Y})^n \rightarrow \mathcal{H}$ Expected risk $R_n(\phi) = \mathbb{E}_n[R(\phi_n)]$	NN,boosting...
Empirical risk minimizer $\hat{h}_n = \operatorname{argmin}_{h \in \mathcal{H}} \hat{R}_n(h)$ Regularized empirical risk minimizer $\hat{h}_n = \operatorname{argmin}_{h \in \mathcal{H}} \hat{R}_n(h) + \lambda C(h)$	

Reinforcement Learning



[Src: https://en.wikipedia.org/wiki/Reinforcement_learning]

- area of machine learning inspired by behaviourist psychology
- how software agents ought to take actions in an environment so as to maximize some notion of cumulative reward.
- Model: random system (typically : Markov Decision Process)
 - agent
 - state
 - actions
 - rewards
- sometimes called approximate dynamic programming, or neuro-dynamic programming

Markov decision process

A **Markov Decision Process** is defined as a tuple $M = (X, A, p, r)$:

- X is the **state** space,
- A is the **action** space,
- $p(y|x, a)$ is the **transition probability** with

$$p(y|x, a) = \mathbb{P}(x_{t+1} = y | x_t = x, a_t = a),$$

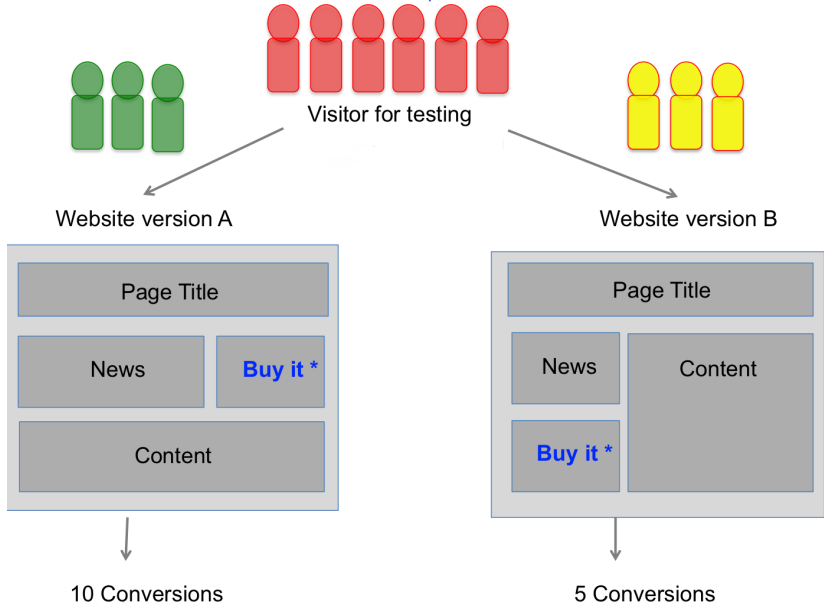
- $r(x, a, y)$ is the **reward** of transition (x, a, y) .

Example: the Retail Store Management Problem

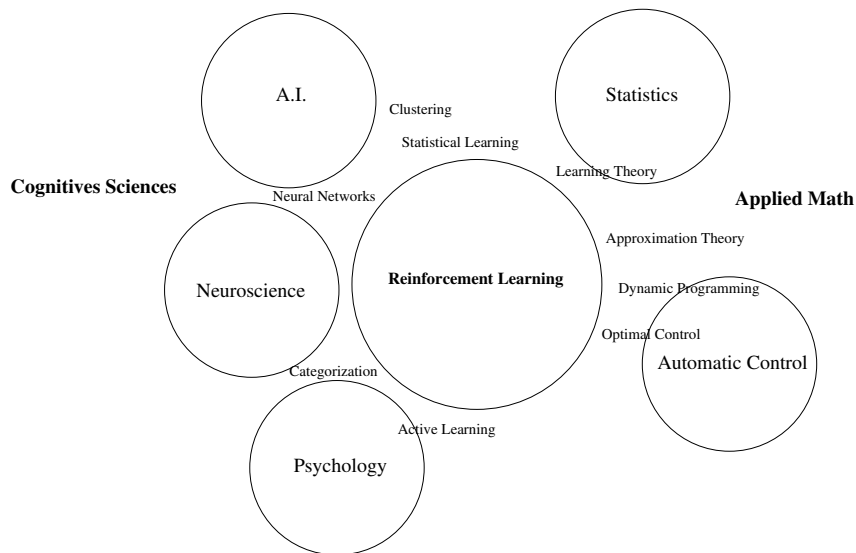
At each month t , a store contains x_t items of a specific goods and the demand for that goods is D_t . At the end of each month the manager of the store can order a_t more items from his supplier. Furthermore we know that:

- The cost of maintaining an inventory of x is $h(x)$.
- The cost to order a items is $C(a)$.
- The income for selling q items is $f(q)$.
- If the demand D is bigger than the available inventory x , customers that cannot be served leave.
- The value of the remaining inventory at the end of the year is $g(x)$.
- **Constraint:** the store has a maximum capacity M .

Example: A/B testing



Reinforcement Learning and its neighbors



Outline of the training program

Day 1 Panorama of machine learning.

Unsupervised learning:

- Principal Component Analysis
- Agglomerative Hierarchical Clustering
- k-means, k-medoids, and variants
- Overview of other methods: spectral clustering, Affinity Propagation, dbscan

Day 2 Supervised learning 1/2:

- Gaussian linear model, logistic regression, model selection
- LASSO and variants
- Support Vector Machines

Outline of the training program

Day 3 Supervised learning 2/2:

- Decision trees
- Bagging, Random Forests, Boosting
- Neural networks, introduction to deep learning

Day 4 Other learning paradigms:

- Sequential learning, multi-armed bandit problems
- Super-learning and expert aggregation
- Reinforcement learning (introduction)

Outline

Machine Learning: when Artificial Intelligence meets Big Data

The Learning Models

Machine Learning Methodology

ML Data

n -by- p matrix X

- n examples = observation points
- p features = characteristics measured for each example

Questions to consider:

- Are the features centered?
- Are the features normalized? bounded?

In `scikitlearn`, all methods expect a 2D array of shape (n, p) often called

X `(n_samples, n_features)`

Data repositories

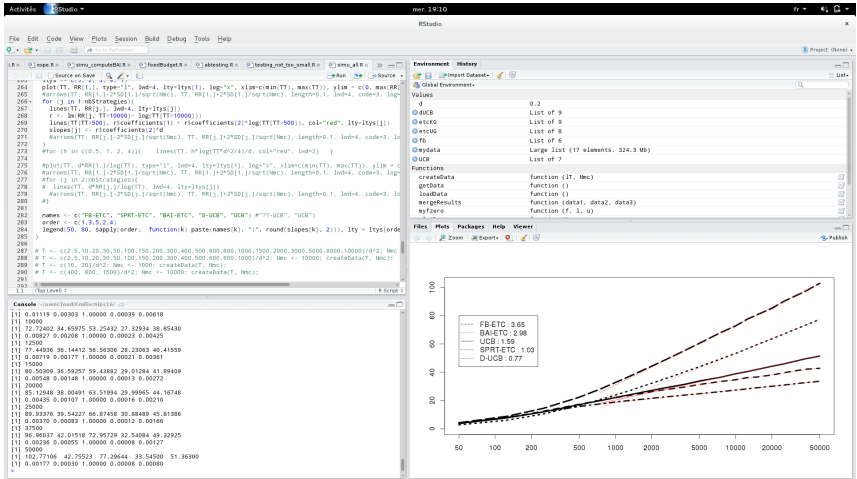
- Inside R: package datasets
- Inside scikitlearn: package sklearn.datasets
- UCI Machine Learning Repository
- Challenges: Kaggle, etc.



The big steps of data analysis

- ① Extracting the data to expected format
- ② Exploring the data
 - detection of outliers, of inconsistencies
 - descriptive exploration of the distributions, of correlations
 - data transformations
- ③ Random partitioning of the data: (see also: cross-validation)
 - learning sample
 - validation sample
 - test sample
- ④ For each algorithm: parameter estimation using training and validation samples
- ⑤ Choice of final algorithm using testing sample, risk estimation

Machine Learning tools: R



Machine Learning tools: python

Activités mer 19:17 Spyder (Python 2.7)

Fichier Édition Recherche Source Exécution Débugger Consoles Outils Affichage Aide

Éditeur : /home/agarvin/ownCloud/prop/python/HC/deprecated/kaplan_meier.py

payment_renewal_study_mysmootherhazard.py survival.py kaplan_meier.py

```

1 # -*- coding: utf-8 -*-
2 """
3 Created on Tue Aug 16 03:05:03 2016
4
5 @author: agarvin
6 """
7
8 import numpy as np
9 from random import random
10 import matplotlib.pyplot as plt
11
12
13 def kaplan_meier(Delta, censorDate):
14     N = len(Delta)-1
15     atRisk = np.zeros(N) # alive after Delta[i]
16     survived = np.zeros(N) # dead between Delta[i] and Delta[i+1]
17     for i in range(N):
18         l = 0
19         while l < N and x[i][0] <= censorDate and x[i][0] < Delta[i+1]:
20             atRisk[i] += 1
21             if x[i][0] < Delta[i+1] < x[i][1]:
22                 survived[i] += 1
23             l += 1
24     S = np.concatenate((l[1], np.cumprod(survived/atRisk)))
25     return S
26 # arbitrary index shift of 17 see kaplan_meier_2 which conforms to package
27
28 plt.close("all")
29 plt.clf()
30 n = 10000
31 x = [np.cumsum(random(), random()) for k in range(n)]
32 x = [x[i][0], x[i][1]-x[i][0]*0.001 for k in range(n)]
33 N = n
34 Delta = np.array([float(i)/N for i in range(N+1)])
35 S = kaplan_meier(Delta, n*n)
36 print(S)
37
38 # plot step(Delta, S)
39 # plot hold-on?
40 # plot hold(Delta, 1-Delta)
41 # plot hold("off")
42
43 # plot.mpl(-0.05, 1, 0.1)
44
45
46 def km_surv(T, E):
47     T = np.argsort(T)
48     t = T[1:] for k in range(T)
49     n = len(t)
50     n = len(t)
51     S = np.zeros(n)
52     for k in range(n-1):
53         S[k+1] = S[k] * (n-k-1)/(n-k)
54         H[k+1] = H[k] + x[k]/(n-k+0.5)

```

Console Python

```

Python 2.7.10 [Anaconda 2.3.0 (64-bit)] (default, Sep 15 2015, 14:50:01)
Type "copyright", "credits" or "license()" for more information.

Python 4.0.0 -- An enhanced Interactive Python.
> Introduction and overview of IPython's features.
> Quick reference.
> Python's own help system.
> Details about 'object' type, use 'object?' for extra details.
> A brief reference about the graphical user interface.

In [1]: runfile('/home/agarvin/ownCloud/prop/python/HC/payment_renewal_study_mysmootherhazard.py',
Traceback (most recent call last):

File "<ipython-input-1-9b3c8426932>", line 1, in <module>
    runfile('/home/agarvin/ownCloud/prop/python/HC/payment_renewal_study_mysmootherhazard.py',
    wdir='/home/agarvin/ownCloud/prop/python/HC')

File "/home/agarvin/anaconda/lib/python2.7/site-packages/ipython/lib/clipboard/external/lib2to3/astrewrite.py", line 885, in runfile
    execute(file_name, namespace)

File "/home/agarvin/anaconda/lib/python2.7/site-packages/ipython/lib/clipboard/external/lib2to3/astrewrite.py", line 78, in execute
    builtins.execute(file_name, "where")

File "/home/agarvin/ownCloud/prop/python/HC/payment_renewal_study_mysmootherhazard.py", line 69, in <module>
    (t, S) = km(T, E)

TypeError: name 'km' is not defined

In [2]: runfile('/home/agarvin/ownCloud/prop/python/HC/deprecated/kaplan_meier.py',
Traceback (most recent call last):

File "/home/agarvin/ownCloud/prop/python/HC/deprecated/kaplan_meier.py", line 53, in km_surv
    S[k+1] = S[k] * (n-k-1)/(n-k)
    ~~~~~
ValueError: operands could not be broadcast together with shapes (10000,) (9999,)

```

Figure 2

console Historique Console Python

ins de ligne : LF Encodage : UTF-8

Ligne : 1 Colonne : 1 Mémoire : 21 %

scikitlearn:

<http://scikit-learn.org/stable/index.html>



The screenshot shows the scikit-learn website homepage. At the top, there's a navigation bar with links for Home, Installation, Documentation, and Examples. Below this is a large blue banner with the scikit-learn logo and the text "Machine Learning in Python". To the left of the banner is a grid of 12 small images showing various data visualizations. To the right of the banner is a list of bullet points highlighting the library's features. Below the banner, the page is organized into a grid of six sections: Classification, Regression, Clustering, Dimensionality reduction, Model selection, and Preprocessing. Each section contains a brief description, applications, and algorithms. At the bottom, there are three columns: News, Community, and Who uses scikit-learn?, each with a brief description and a link to more information.

scikit-learn
Machine Learning in Python

- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

Classification

Identifying to which category an object belongs to.

Applications: Spam detection, Image recognition.

Algorithms: SVM, nearest neighbors, random forest, ... — Examples

Regression

Predicting a continuous-valued attribute associated with an object.

Applications: Drug response, Stock prices.

Algorithms: SVR, ridge regression, Lasso, ... — Examples

Clustering

Automatic grouping of similar objects into sets.

Applications: Customer segmentation, Grouping experiment outcomes

Algorithms: k-Means, spectral clustering, mean-shift, ... — Examples

Dimensionality reduction

Reducing the number of random variables to consider.

Applications: Visualization, Increased efficiency

Algorithms: PCA, feature selection, non-negative matrix factorization. — Examples

Model selection

Comparing, validating and choosing parameters and models.

Goal: Improved accuracy via parameter tuning

Modules: grid search, cross validation, metrics. — Examples

Preprocessing

Feature extraction and normalization.

Application: Transforming input data such as text for use with machine learning algorithms.

Modules: preprocessing, feature extraction. — Examples

News

On-going development: What's new (Changelog)

Community

About us See authors and contributing More Machine Learning Find related

Who uses scikit-learn?

AWeber COMMUNICATIONS

Knime, Weka and co: integrated environments

The screenshot displays the Weka Explorer application window, which is used for data mining and machine learning. The interface includes several tabs: Preprocess, Classify (selected), Cluster, Associate, Select attributes, and Visualize.

Classifier: J48 -C 0.25 -M 2

Test options:

- ☒ Use training set
- ☐ Supplied test set (Set...)
- ☐ Cross-validation (Folds: 10)
- ☐ Percentage split (%: 66)

Classifier output:

```
=== Stratified cross-validation ===
=== Summary ===
Correctly Classified Instances      144      96 %
Incorrectly Classified Instances     6       4 %
Kappa statistic                    0.94
Mean absolute error                 0.035
Root mean square error              0.035
```

Tree View:

```
graph TD
    A[petalwidth] -- "<= 0.6" --> B[Iris-setosa 50.0]
    A -- "> 0.6" --> C[petalwidth]
    C -- "<= 1.7" --> D[petallength]
    C -- "> 1.7" --> E[Iris-virginica 46.0/1.0]
    D -- "<= 4.9" --> F[Iris-versicolor 48.0/1.0]
    D -- "> 4.9" --> G[petalwidth]
    G -- "<= 1.5" --> H[Iris-virginica 3.0]
    G -- "> 1.5" --> I[Iris-versicolor 3.0/1.0]
```

Visualize: Y: petalwidth (Num)

Result list (right-click for...):

- 1 View in main window
- View in separate window
- Save result buffer
- Load model
- Save model
- Re-evaluate model
- Visualize classification
- Visualize tree
- Visualize margin
- Visualize threshold

Visualize: Iris-versicolor (red X), Iris-virginica (green X)