Demystifying ML, AI & Automation

Spyros Gkezerlis

24-25.10.2017

Intro

Spyros Gkezerlis - @sgez

About
BSc Automation Engineering – Telematic Applications
MSc Information Systems – Reinforcement Learning
Interests – Applied ML, RL especially in time domain

Experience
5 years Software Engineer in Greece
8 years Engineer and Manager in OTE
3 years Program Manager in DTAG

Ariadni Gkezerli
(c) 2017

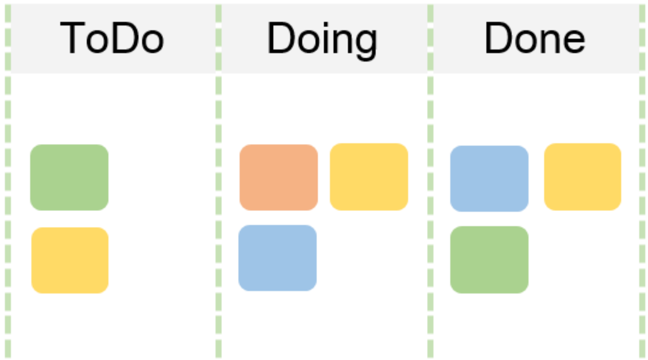
“In theory there is no difference between theory and practice. In practice there is.”

Yoggi Berra

Lecture Guidelines

- Stop for questions at any time! Be curious! Use Laptops for solving problems, exercises during the lectures!
- Lunch all together for questions and NT
- Workflow to be used?
  Kanban board with ToDo, Doing, Done
- First iteration through everything is a must
  if more items: put in ToDo for next time
- Set as ToDo from the beginning

  Basic understanding
  of the landscape



- If noted, slides have links with extra source material
- TLAs can be explained whenever asked :-)

Why Lectures?

- Get a better overview of the current landscape in ML, AI & Automation
- "Set the record straight" sessions
- Create a common understanding of what ML, AI & Automation
- “Start with the problem” philosophy

Pros cons, Tools, etc. should not dictate what we should use!

  - Identify what we want to solve
  - Work to the algorithms & models needed
  - Utilise best approach
- Assess ML potential

> “AI is the New Electricity”
>
> Andrew Ng

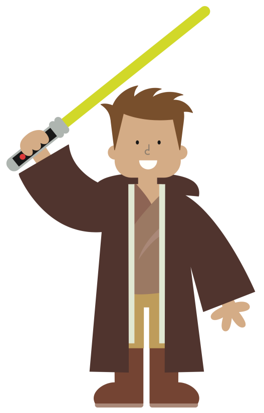
Why ML, AI, Automation?

- Why use Automation, Adv.Stats, Probability Theory, ML, AI?

  Because it can help us:
  - Reduce complexity of network
  - Improve experience by Time-to-market, Time-to-repair
  - Repetitive caused costs can be targeted and reduced

- Also… AI is another fun way to solve problems ☺

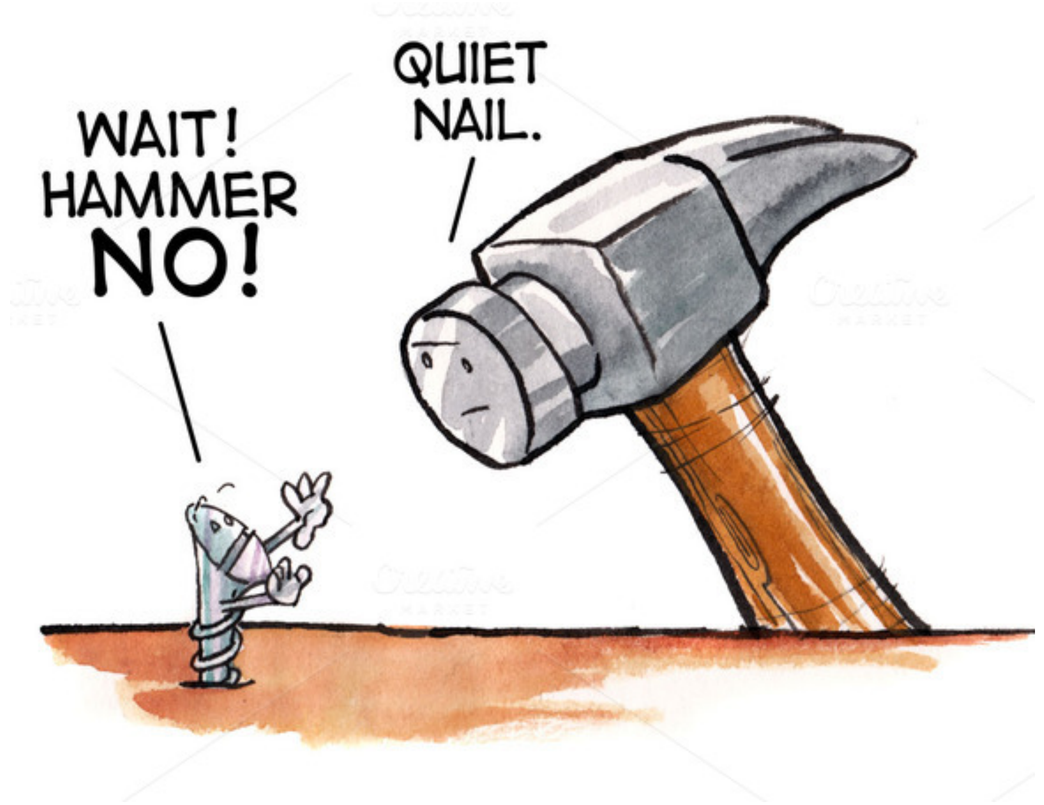
- Keep in mind it can solve particular types of problems!!!



Do you want to become an ML Jedi?

But, be aware of the dark side!

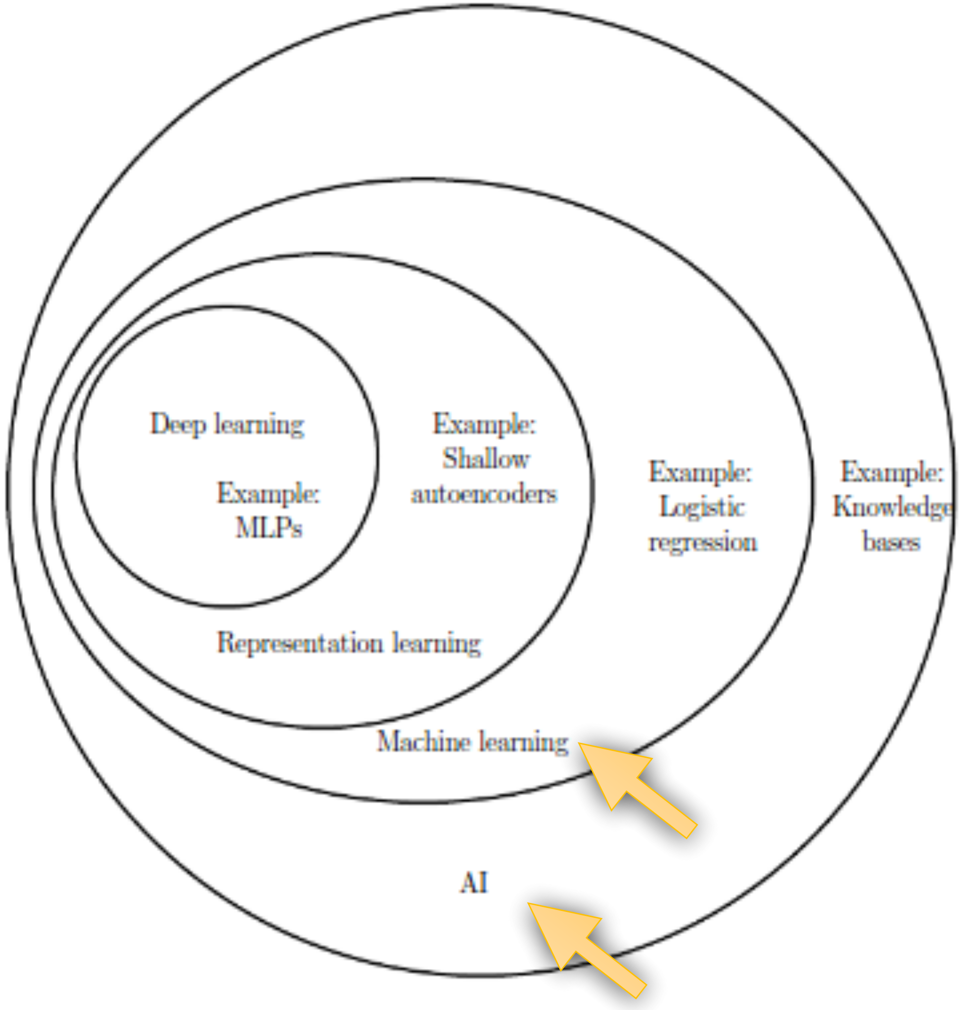
- Keep in mind it can solve particular types of problems!!!
- ML is not a tool to fix everything
- Needs high degree of mathematics understanding and of course, you should know your problem!!!



Landscape overview

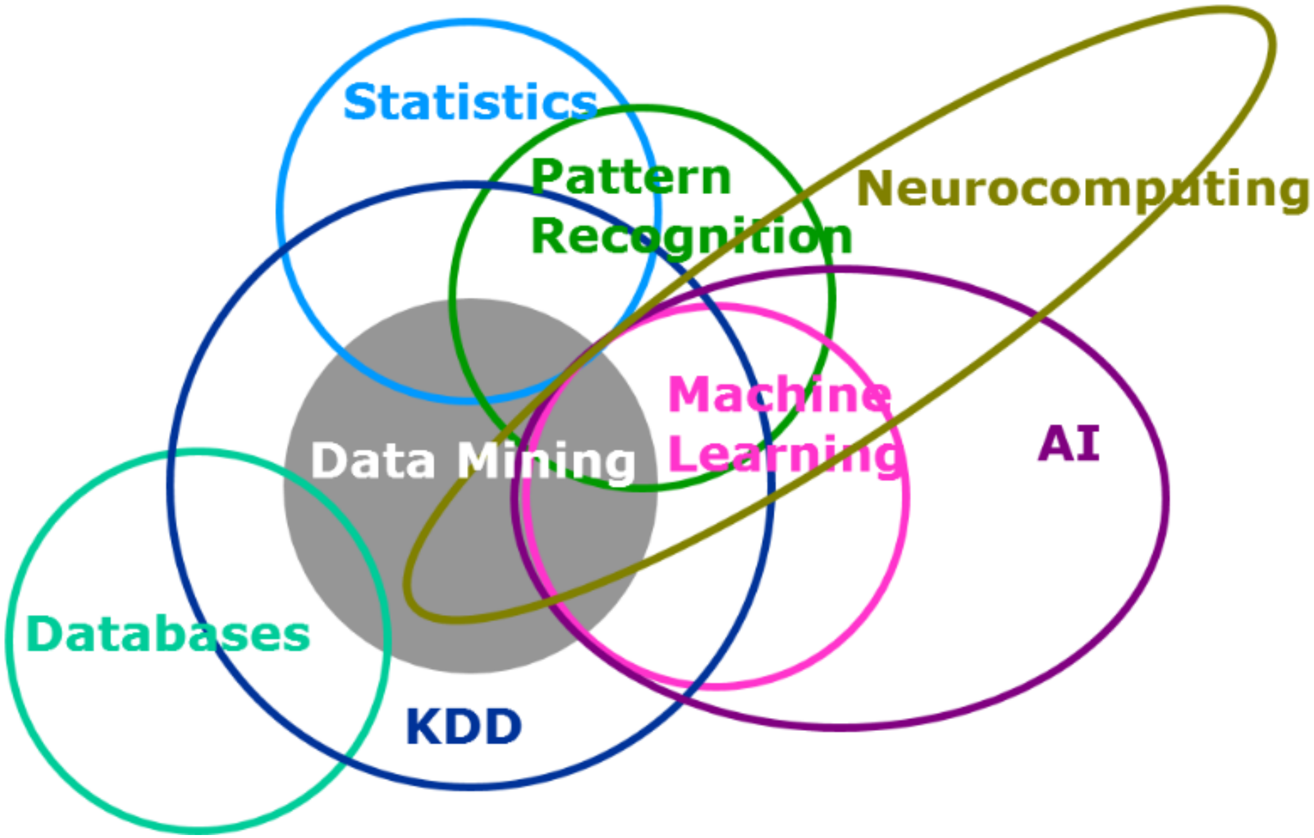
Landscape and focus[1] areas vary:

- Automation
- Statistics & Probability
- Data Mining
- Artificial Intelligence
- Machine Learning
  - Supervised Learning
  - Unsupervised Learning
  - Reinforcement Learning
- Big Data Analytics



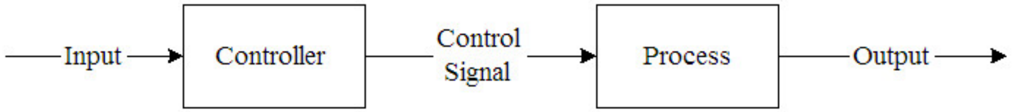
Landscape overview

Landscape and focus[2] areas of ML also could have overlaps with other scientific disciples such as Data Mining, AI, Big Data an other.

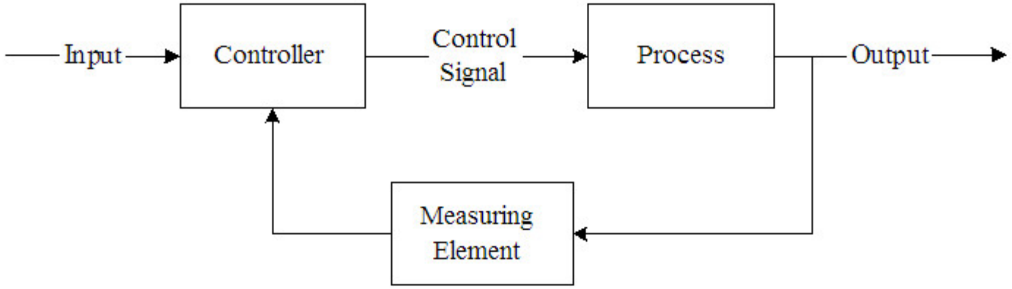


Automation

- Comes from ancient greek word which means the thing that wishes on its own or the has a will or fury by itself
- Describes the tasks that are or can be performed by machines autonomously
- It can be used to improve quality, accuracy, save costs and amplify precision



Open Loop System



Closed Loop System

Bayes Theorem

- P(A|B) ==> Probability of A given that B
    - Exercise on Bayes[3]
      Out of 3000 emails received over a certain period, 2000 are spam and 1000 are not. The word “Rolex” appeared in 250 out of the 2000 which are spam and in 5 out of the 1000. So, if an email is received, and contains the word “Rolex”, what is the possibility that it is a spam?

Let S be the event that the message is spam, and E be the event that the message contains the word w. Under our assumption from before, we have that:

$$p(S|E) = \frac{p(E|S)}{p(E|S) + p(E|\overline{S})}$$

Bayes Theorem exercise

- Example – Solution[3]:
  Out of 3000 emails received over a certain period, 2000 are spam and 1000 are not. The word "Rolex" appeared in 250 out of the 2000 which are spam and in 5 out of the 1000.

  So, if an email is received, and contains the word "Rolex", what is the possibility that it is a spam?

Now p(w) and q(w)
are empirical estimates of
p(E|S) and p(E|S')

$$r(Rolex) = \frac{p(Rolex)}{p(Rolex) + q(Rolex)} =$$

$$\frac{0.125}{0.125 + .005} = \frac{0.125}{0.125 + .005} \approx 0.962$$

Markov Chains

- ## Statistics & Probability - Markov Chains[4]

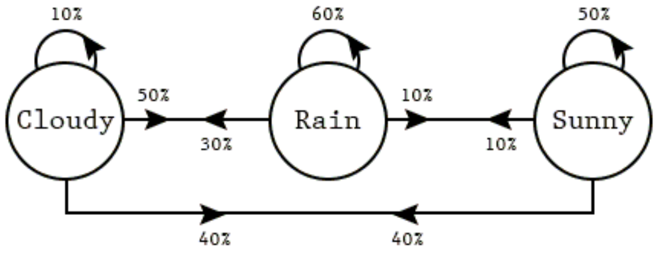
  Markov Chains is a probabilistic process, that relies on the current state to predict the next state. For Markov chains to be effective the current state has to be dependent on the previous state in some way; For instance, from experience we know that if it looks cloudy outside, the next state we expect is rain. We can also say that when the rain starts to subside into cloudiness, the next state will most likely be sunny.

  Not every process has the Markov Property, such as the Lottery, this weeks winning numbers have no dependence to the previous weeks winning numbers.

Markov Table of Probabilities

STATE	NEXT STATE	PROBABILITY	%
CLOUDY	CLOUDY	0.1	10%
CLOUDY	RAIN	0.5	50%
CLOUDY	SUNNY	0.4	40%
RAIN	CLOUDY	0.3	30%
RAIN	RAIN	0.6	60%
RAIN	SUNNY	0.1	10%
SUNNY	CLOUDY	0.4	40%
SUNNY	RAIN	0.1	10%
SUNNY	SUNNY	0.5	50%

Markov State Diagram



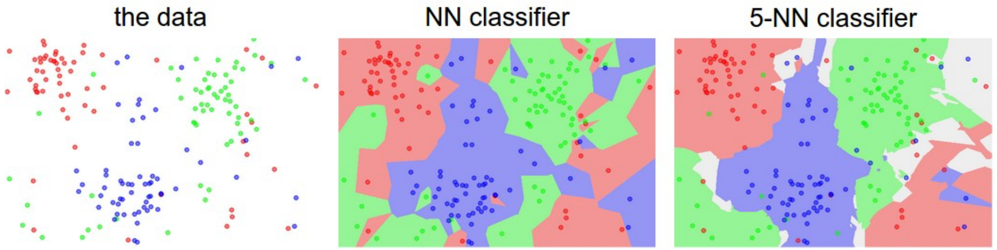
Current State Vector

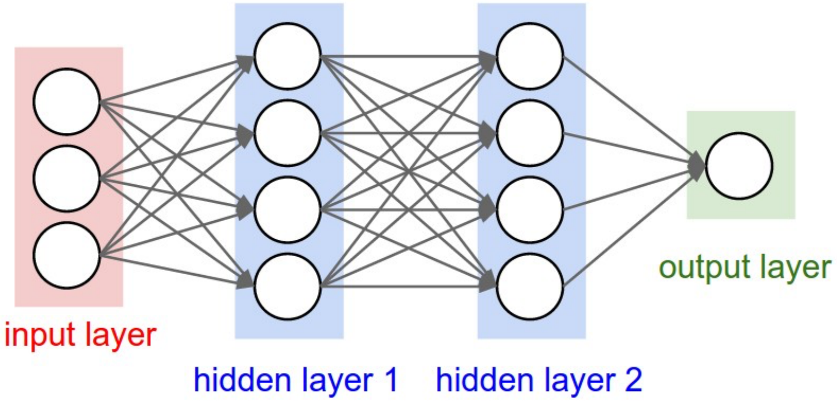
C	R	S
1	0	0

Figure 4

Classifiers and Structures

- Classifiers
  - kNNc[5]
  - Classifiers
- Structures used
  - Trees
  - Graphs
  - Neural Networks[5]

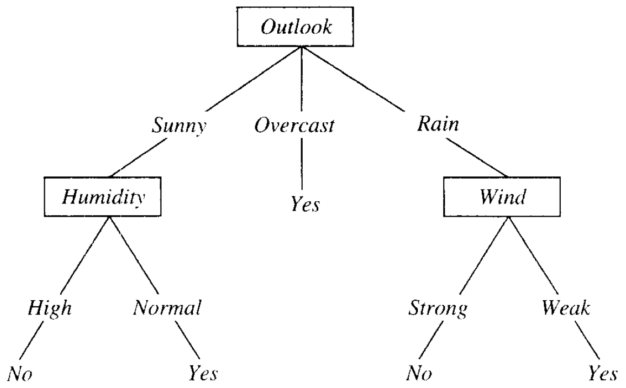




Supervised Learning - DTL

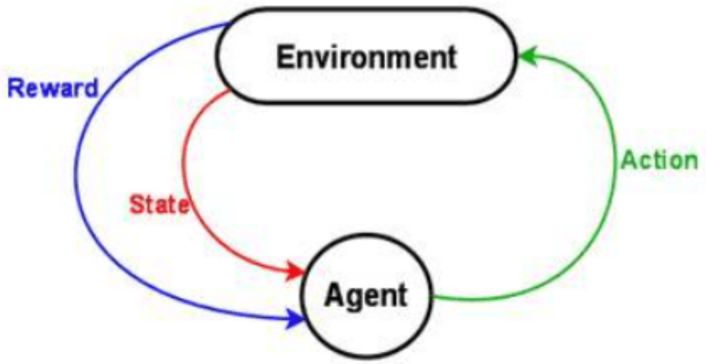
- Decision Tree Learning
  DTL is method for approximating discrete valued target functions, in which the learned function is represented by a decision tree. (Weka example will follow)
- Example[6] dataset converted via algorithm to Decision tree

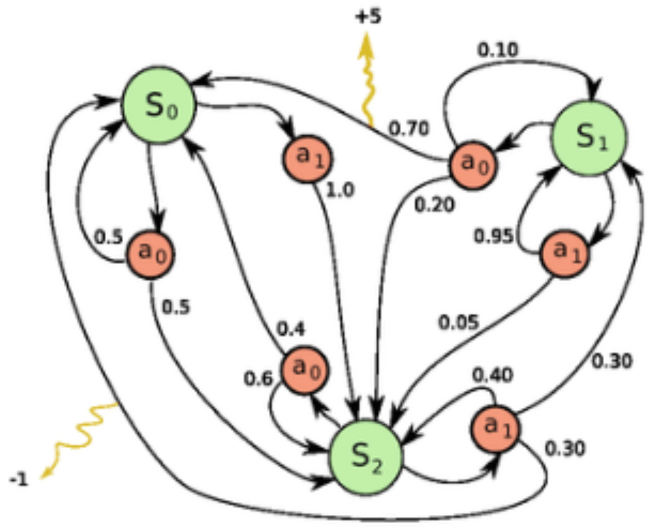
No.	1: outlook Nominal	2: temperature Numeric	3: humidity Numeric	4: windy Nominal	5: **play** Nominal
1	sunny	85.0	85.0	FALSE	no
2	sunny	80.0	90.0	TRUE	no
3	overcast	83.0	86.0	FALSE	yes
4	rainy	70.0	96.0	FALSE	yes
5	rainy	68.0	80.0	FALSE	yes
6	rainy	65.0	70.0	TRUE	no
7	overcast	64.0	65.0	TRUE	yes
8	sunny	72.0	95.0	FALSE	no
9	sunny	69.0	70.0	FALSE	yes
...	rainy	75.0	80.0	FALSE	yes
...	sunny	75.0	70.0	TRUE	yes
...	overcast	72.0	90.0	TRUE	yes
...	overcast	81.0	75.0	FALSE	yes
...	rainy	71.0	91.0	TRUE	no



Reinforcement Learning (RL)

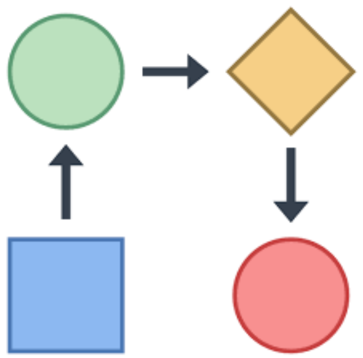
- Reinforcement Learning[7] is learning what to do--how to map situations to actions--so as to maximise a numerical reward signal.
- Reinforcement learning is defined not by characterising learning methods, but by characterising a learning problem.





Workflow – The ML Pipeline[8]

- Start with the question or problem we want to solve
- Find proper data and sources
- Prepare & create a data set
- Choose a model e.g. Decision Tree, J48 Algorithm
- Train system & classify
- Evaluate the system and fine-tune
- Predict / Forecast
- Apply to workflow
- Automate into workflow

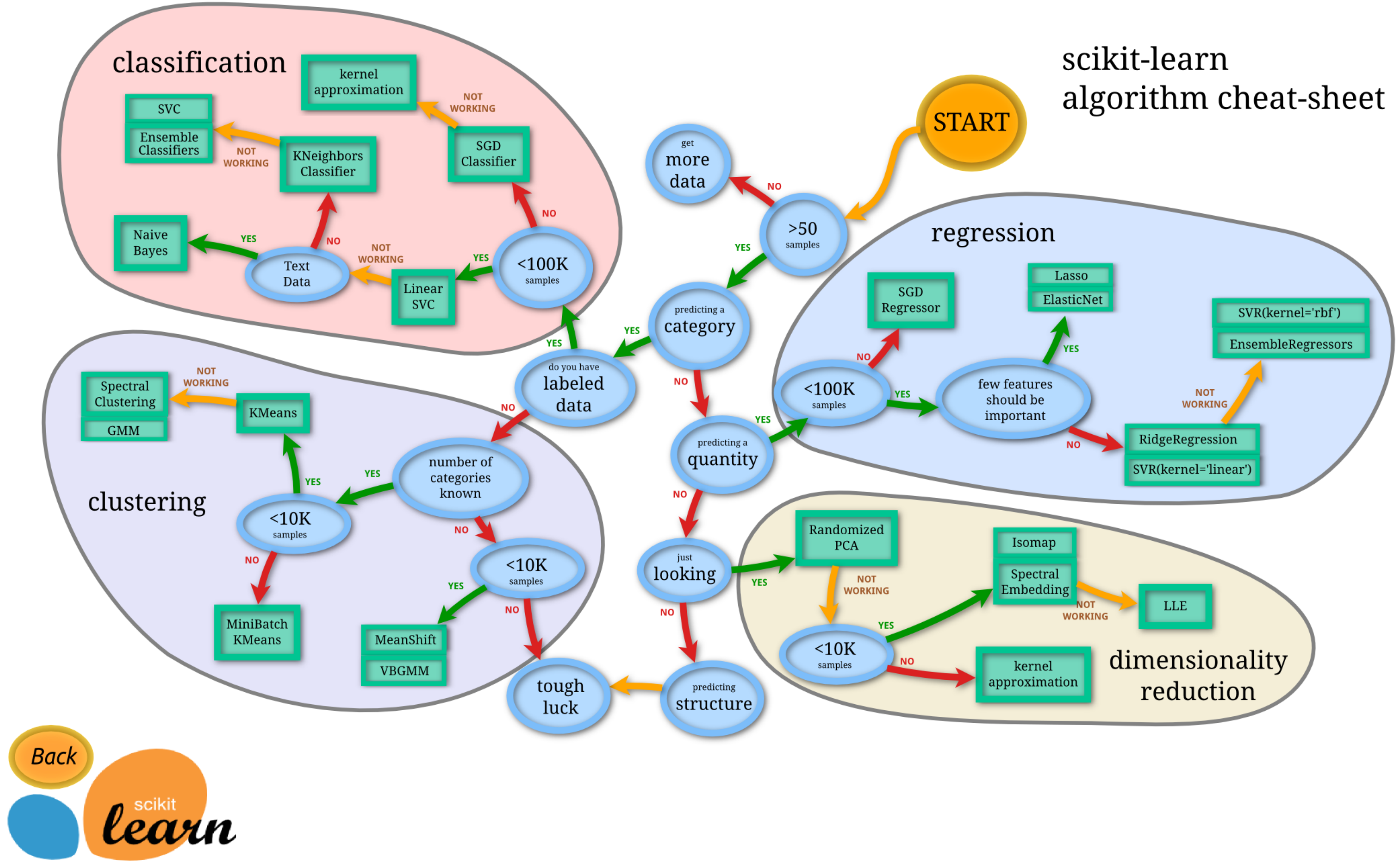


Prepare & create a dataset

- Dataset
  - Features or Attributes
  - Labels
  - Instances

Age	Nationality	Education	Job
25	Greek	MSc	Engineer
38	American	MSc	Analyst
35	Japanese	PhD	Assistant
…	…	…	…

Choose a model[9]



Tools – Landscape[10]

Category	Sub-category	Description	Examples
Packages of ML Implementations	**Statistical Software Packages**	Software toolkits with a large set of implementations of ML algorithms, typically with visualization support	SAS, **R**, Matlab, SPSS
	Data Mining Toolkits	Software toolkits with a relatively limited set of ML algorithms, typically over a data platform, possibly with incremental maintenance	**Weka**, AzureML, ODM, MADlib, Mahout, Hazy-Classify
	Developability-oriented Frameworks	Software frameworks and systems that aim to improve developability, typically from academic research	GraphLab, Bismarck, MLBase
	SRL Frameworks	Implementations of statistical relational learning (SRL)	DeepDive
	Deep Learning Systems	Implementations of deep neural networks	Google Brain, Microsoft Adam
	Bayesian Inference Systems	Systems providing scalable inference for Bayesian ML models	SimSQL, Elementary, Tuffy
Linear Algebra-based Systems	**Statistical Software Packages**	Systems offering an interactive statistical programming environment	SAS, **R**, Matlab
	R-based Analytics Systems	Systems that provide R or an R-like language for analytics, typically over a data platform, possibly with incremental maintenance	RIOT, ORE, SystemML, LINVIEW
Model Management Systems		Systems that provide querying, versioning, and deployment support	SAS, LongView, Velox
Systems for Feature Engineering		Systems that provide abstractions to make feature engineering easier	Columbus , DeepDive
Systems for Algorithm Selection		Systems that provide abstractions to make algorithm selection easier	MLBase, AzureML
Systems for Parameter Tuning		Systems that provide abstractions to make parameter tuning easier	SAS, R, MLBase, AzureML

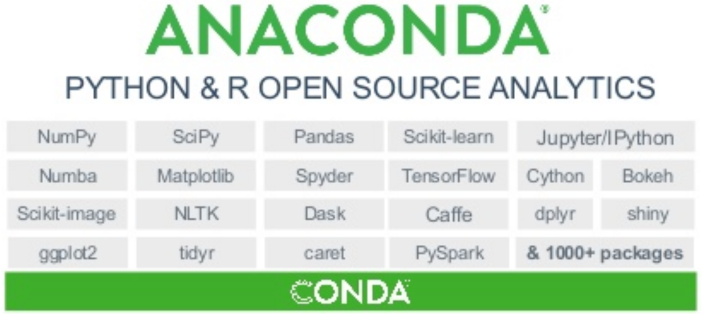
Deep Learning Toolkits comparison[11]

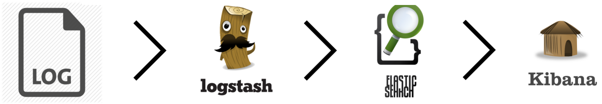
Toolkit	GPU Support	Other
Caffe	Yes	JSON-like text file to describe the network architecture
Deeplearning4j	Yes	Java on Scala API
Tensorflow	Yes	Google backing, high adoption - Python
Theano		Python
Keras		Python - uses Theano or Tensorflow as backend
MXNet	Yes	C++
Lasagne		Python - uses Then
CNTK		VS for ML - developed by Microsoft
DIGITS		Nvidia - web based tool
Torch		Written in C
PyTorch	Yes	Python frontend
Pylearn2		Python
Chainer		

Toolkits for Lectures – Overview

- Weka
    - Data Mining GUI / basic ML
- Anaconda
    - Python
        - Dask
        - Scikit-Learn for ML/DL
        - Tensorflow for DL
- Orange & R Studio
- Elasticsearch - Logstash - Kibana

- Things to consider for a toolkit
    - Environment
    - Dev & Exec speed
    - Training Speed
    - GPU Support
    - Community support & contributors



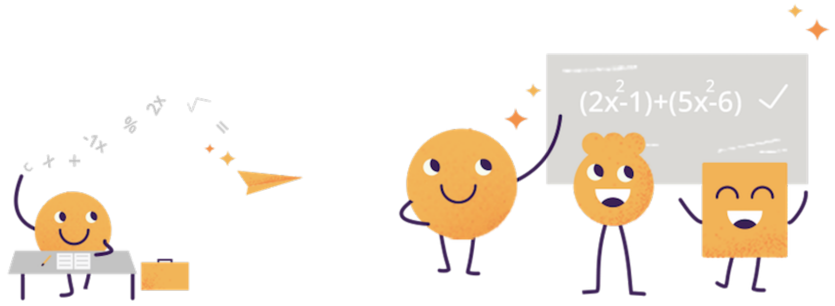


Practice (fun part :-)

- Weka - Statistics - Pipeline - DTL
- R Studio - Statistics
- Anaconda Python - Statistics - basic ML
- Elasticsearch (ELK) - Visualisation & ML

  on:

  - bitcoin
  - cars
  - flights
  - milano_cells



Lectures & Data Sources Page
https://github.com/sgez/MLAI

Special Thanks to Cosmote!

24.10.2017

- Diogenis Georgiadis
    - Zeta Papadopoulou
        - Christos Kavakopoulos

25.10.2017

- Christos Penitas
    - Antonis Galanakis
        - George Georgitselis

References

[1] Deep Learning, Goodfellow; Bengio; Courville, 2016
http://www.deeplearningbook.org

[2] Look at Popular Data mining Algorithms and Deep learning, Sharma, 26 Feb 2015
http://https://www.linkedin.com/pulse/look-popular-data-mining-algorithms-deep-learning-sandeep-sharma-me-

[3] CSE 240 Home Page, Sproull, Spring 2016
https://www.arl.wustl.edu/~todd/cse240/cse240_lecture17_3-24-16.pdf

[4] Markov Chains Explained, Armstrong, 2015
http://techeffigytutorials.blogspot.com/2015/01/markov-chains-explained.html

[5] CS231n: Convolutional Neural Networks for Visual Recognition, Karpathy, 2017
http://cs231n.github.io

[6] Machine Learning, Mitchell, McGraw, 1997.
http://www.cs.princeton.edu/courses/archive/spr07/cos424/papers/mitchell-dectrees.pdf

[7] Reinforcement Learning: An Introduction, Sutton; Barto, 1998
https://mitpress.mit.edu/0262193981

[8] ML Workflow - Loosely based on Yufeng G instructional series
https://medium.com/towards-data-science/the-7-steps-of-machine-learning-2877d7e5548e

[9] Scikit-Learn Algorithm selection Procedure, Scikit-learn website, 23 Oct 2017
http://scikit-learn.org/stable/tutorial/machine_learning_map/index.html

[10] A Survey of the Existing Landscape of ML Systems, Kumar; McCann; Naughton; Patel, 27 Nov 2015
https://minds.wisconsin.edu/handle/1793/73834

[11] Toolkits and Libraries for Deep Learning, Erickson; Korfiatis; Akkus; J Digit Imaging 30: 400., 2017
https://doi.org/10.1007/s10278-017-9965-6

(citation).