Introduction to Machine Learning tools with Python

Stanisław Robak

Agenda

- Why machine learning?
- Why Python?
- Introduction to Python
- Libraries
- Examples: regression, classification

Why Machine Learning?

AI (Artificial Intelligence) - the science of making machines that:

Think like humans	Think rationally
Act like humans	Act rationally

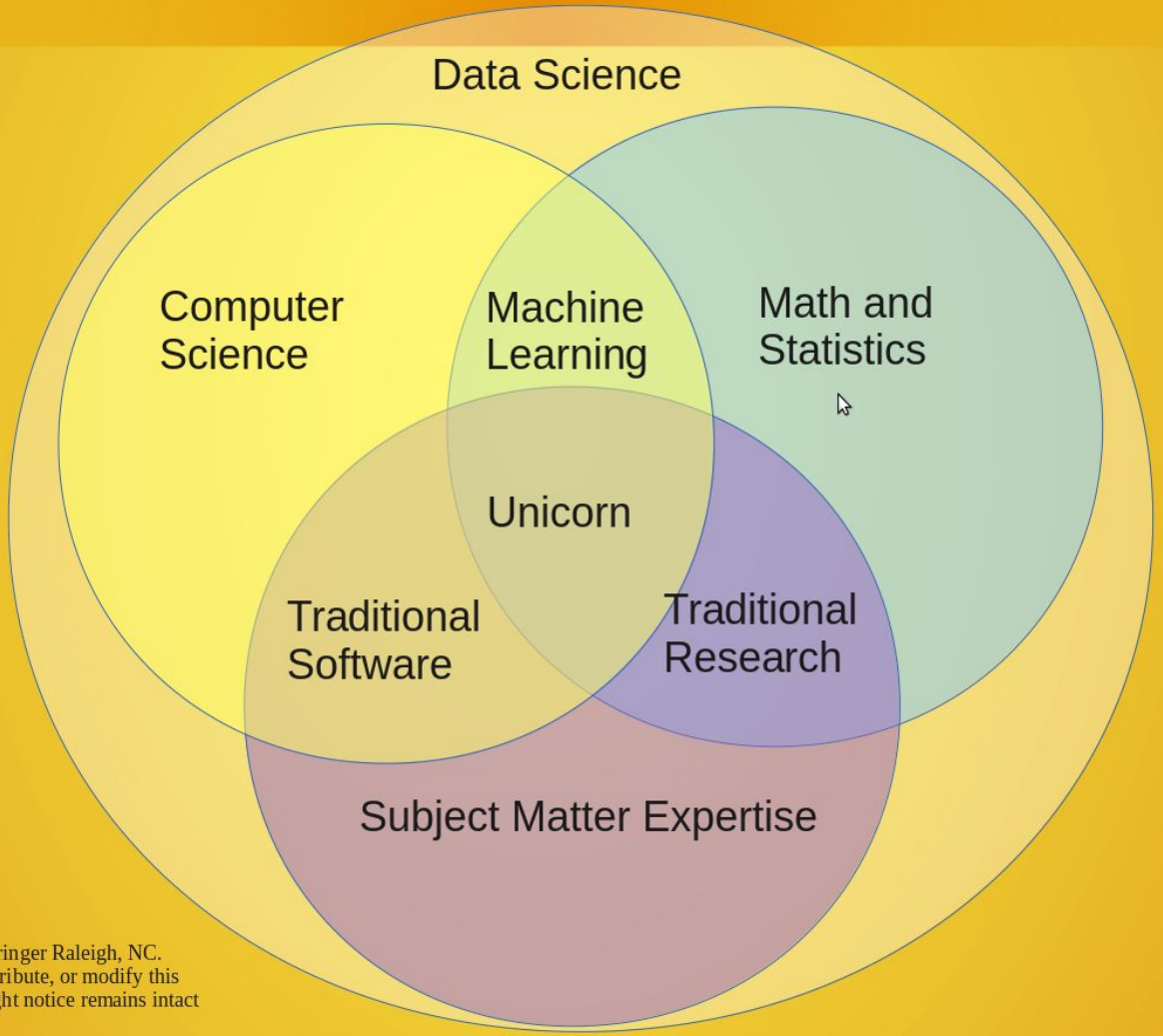
```
<!-- Synonimy -->
<category>
<pattern>Czesc</pattern>
<template>Witaj.</template>
</category>
<category>
<pattern>Dzien dobry</pattern>
<template><srai>Czesc</srai></template>
</category>
<category>
<pattern>Witaj</pattern>
<template><srai>Czesc</srai></template>
</category>
<category>
<pattern>Dobry wieczor</pattern>
<template><srai>Czesc</srai></template>
</category>

<category>
<pattern>Z jakiej jestes firmy</pattern>
<template>Jestem botem firmy y-kom</template>
</category>
<category>
<pattern>Jaka firme reprezentujesz</pattern>
<template><srai>Z jakiej jestes firmy</srai></template>
</category>

```

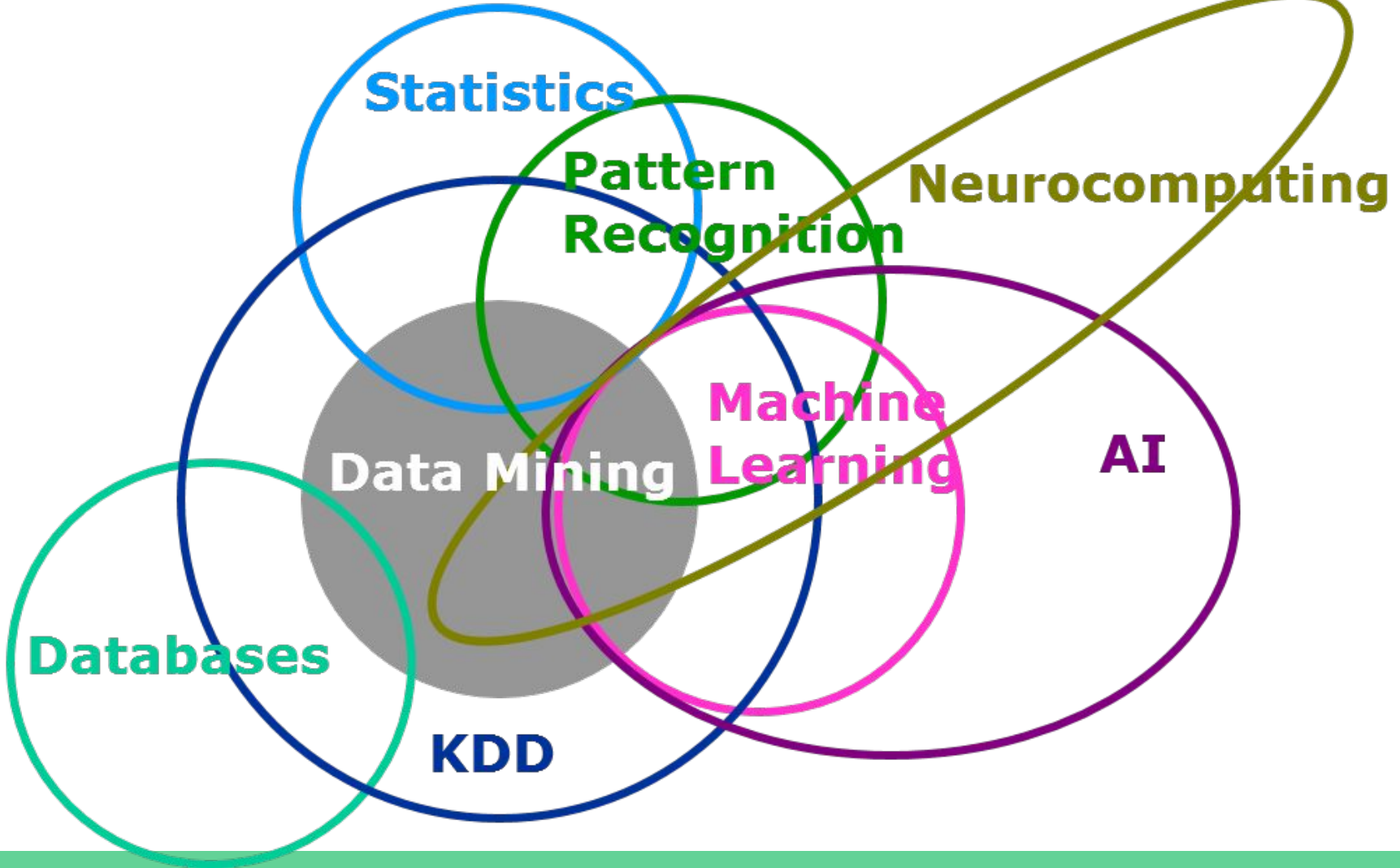
chatbot

Machine learning is a type of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed.

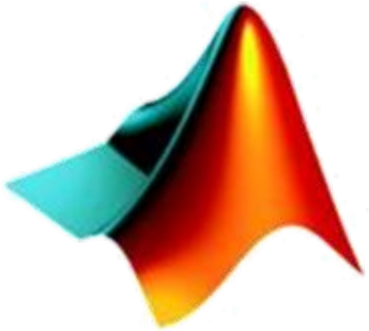
Machine learning focuses on the development of computer programs that can teach themselves to grow and change when exposed to new data.

“A Few Useful Things to Know about Machine Learning”, Pedro Domingos




Statistics
Pattern Recognition
Neurocomputing
Machine Learning
AI
Data Mining
Databases
KDD

Machine learning tools

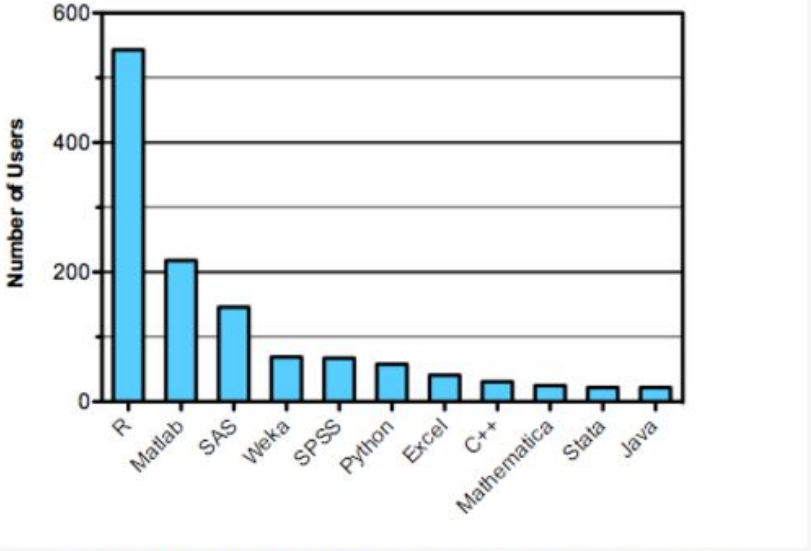


MATLAB

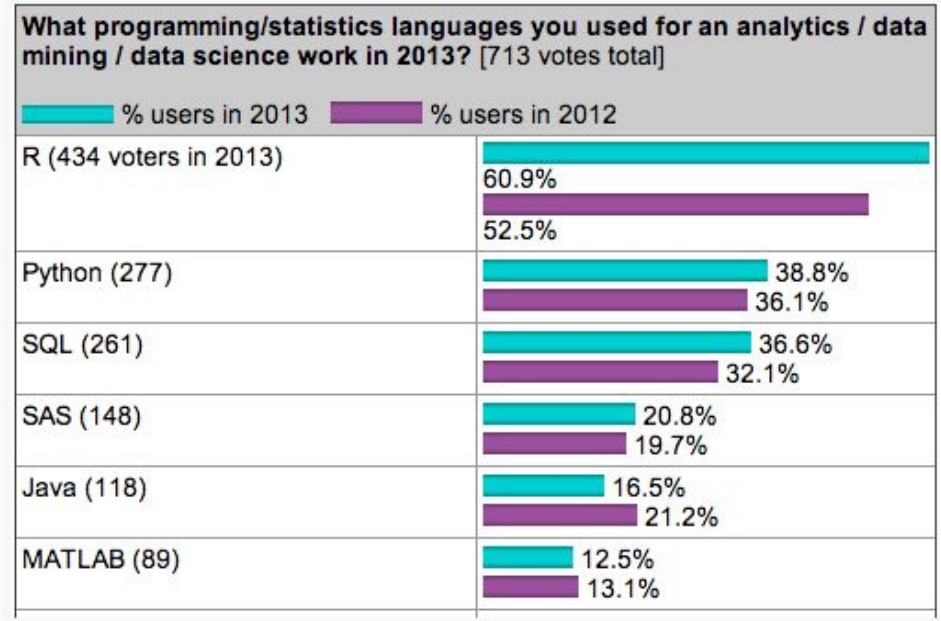
julia

For more precise comparison …

http://machinelearningmastery.com/best-programming-language-for-machine-learning/



The most popular tools used on Kaggle, the machine learning competition website.



The most popular platforms for machine learning, taken from the KDnuggets 2013 poll.

Why Python?

General Characteristic:

- **clean and simple language**: easy-to-read and intuitive code, easy-to-learn minimalistic syntax, maintainability scales well with size of projects
- **expressive language:** fewer lines of code, fewer bugs, easier to maintain
- **broad variety of libraries**

Introduction to Python

>> ipython notebook

For more python

Check: **notebook gallery** in Anaconda and

Introduction to Python

scikit-learn http://scikit-learn.org

scikit-learn

Machine Learning in Python

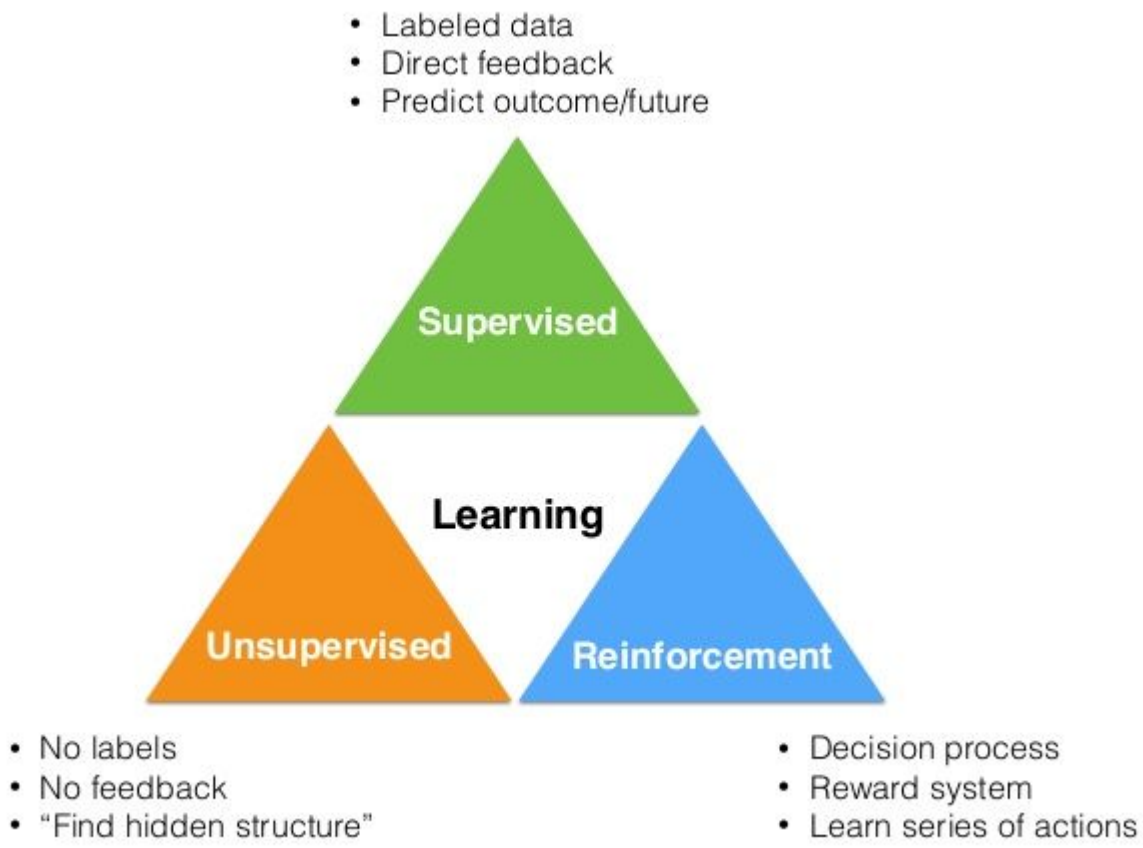
- Simple and efficient tools for data mining and data analysis
- Accessible to everybody, and reusable in various contexts
- Built on NumPy, SciPy, and matplotlib
- Open source, commercially usable - BSD license

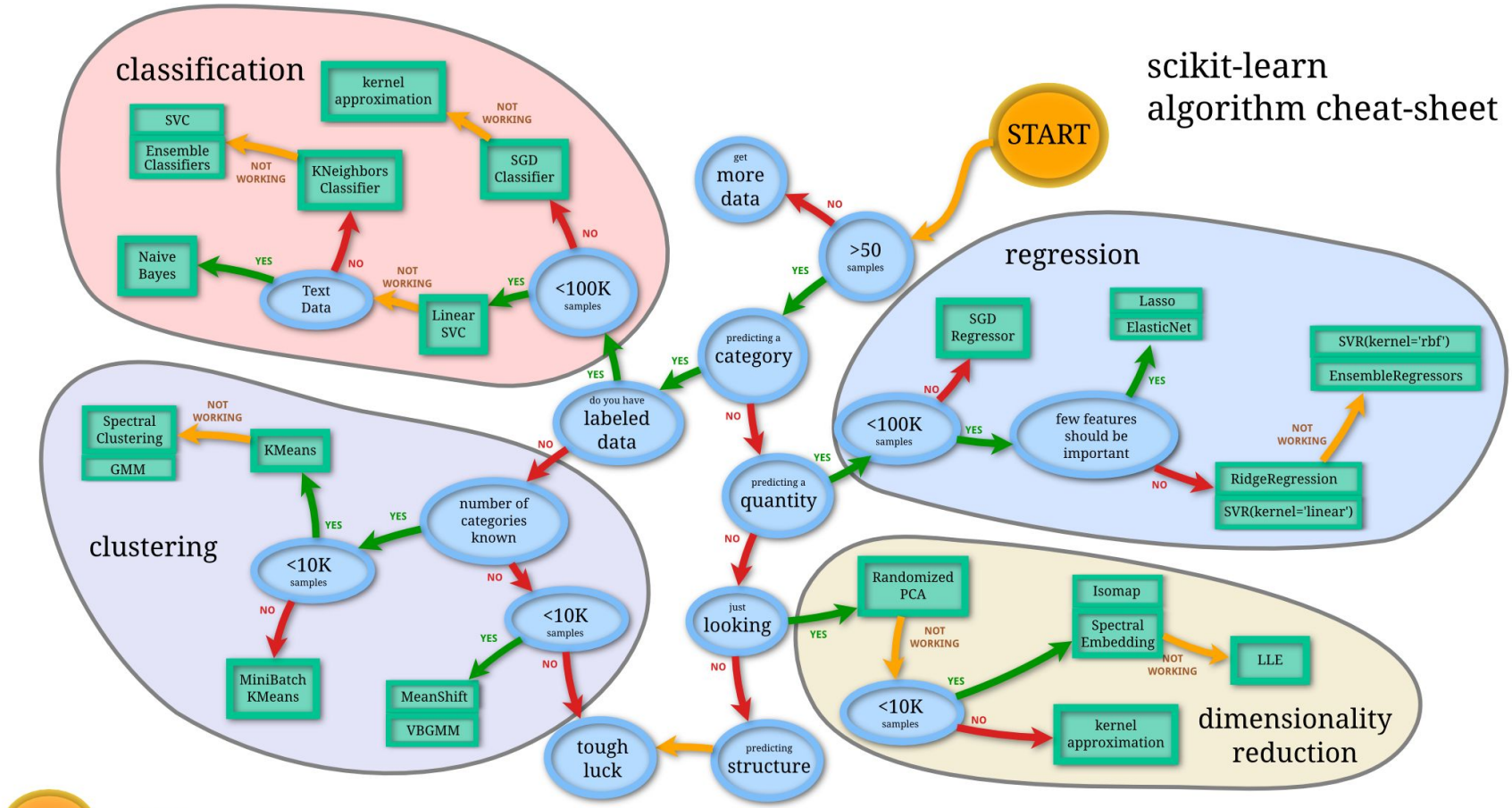
machine learning in Python

theano - http://deeplearning.net/software/theano/

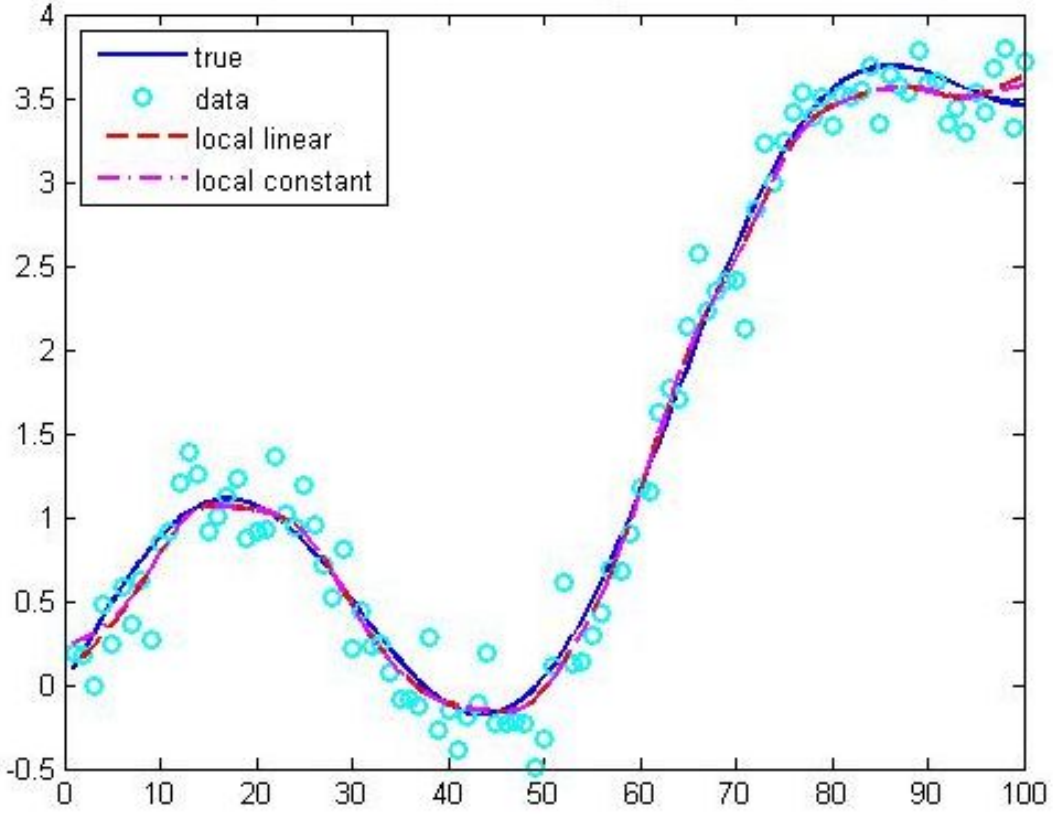
- define, optimize and evaluate mathematical expressions involvinf multi-dimensional arrays efficiently
- transparent use of GPU
- speed and stability optimizations
- tight integration with NumPy

Examples


• Labeled data
• Direct feedback
• Predict outcome/future
Supervised
Learning
Unsupervised
Reinforcement
• No labels
• No feedback
• "Find hidden structure"
• Decision process
• Reward system
• Learn series of actions


scikit-learn
algorithm cheat-sheet
START
get more data
NO
>50 samples
YES
predicting a category
YES
NO
do you have labeled data
YES
NO
predicting a quantity
YES
NO
just looking
NO
predicting structure
tough luck
classification
kernel approximation
NOT WORKING
SVC
Ensemble Classifiers
NOT WORKING
KNeighbors Classifier
SGD Classifier
NO
Naive Bayes
YES
Text Data
NO
NOT WORKING
Linear SVC
YES
<100K samples
YES
clustering
Spectral Clustering
GMM
NOT WORKING
KMeans
YES
number of categories known
YES
<10K samples
NO
MiniBatch KMeans
NO
<10K samples
YES
MeanShift
VBGMM
NO
regression
SGD Regressor
NO
<100K samples
YES
few features should be important
YES
Lasso
ElasticNet
NO
RidgeRegression
SVR(kernel='linear')
NOT WORKING
SVR(kernel='rbf')
EnsembleRegressors
dimensionality reduction
Randomized PCA
YES
NOT WORKING
<10K samples
YES
Isomap
Spectral Embedding
NOT WORKING
LLE
NO
kernel approximation
Back
scikit learn

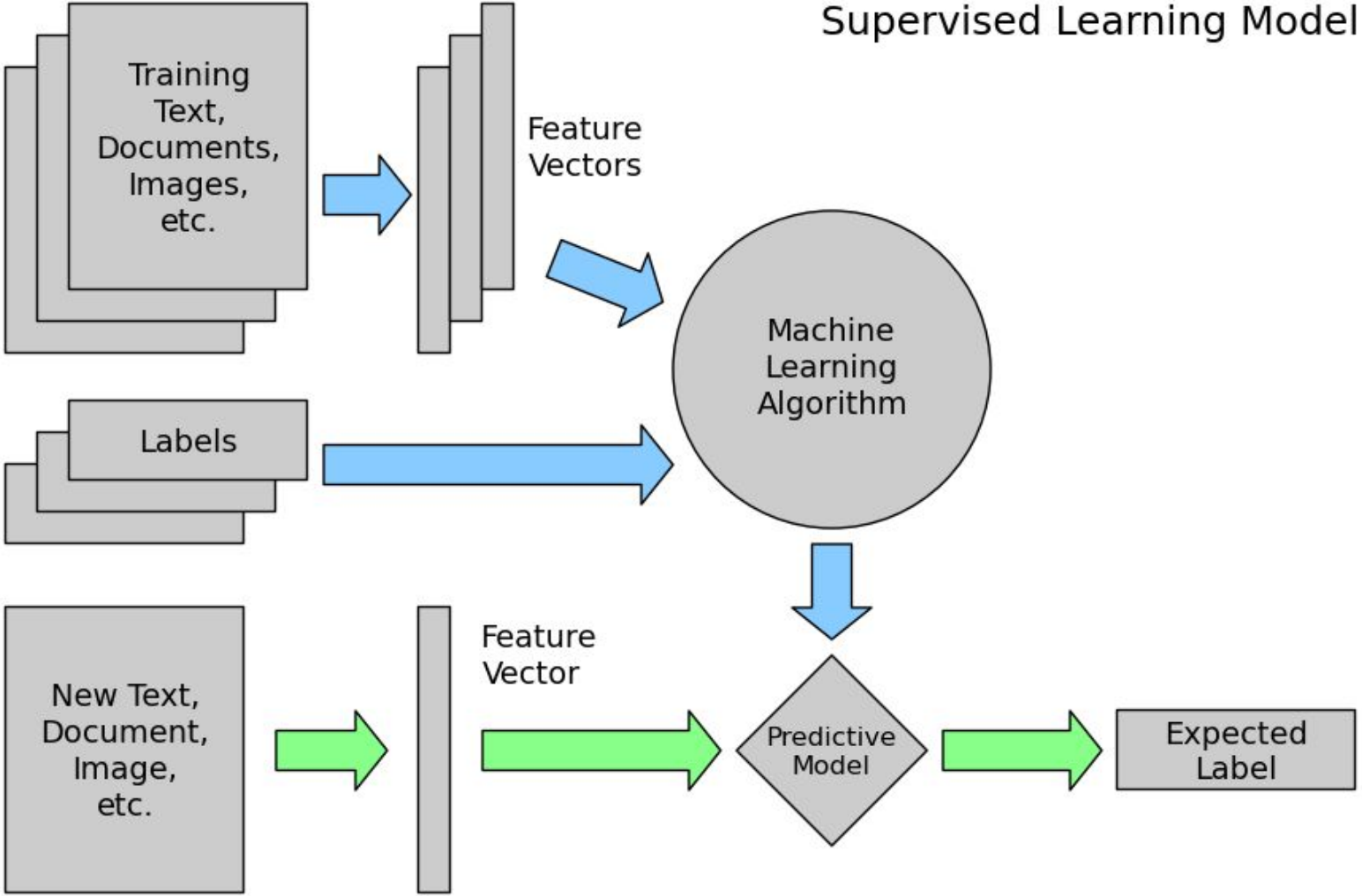
Regression

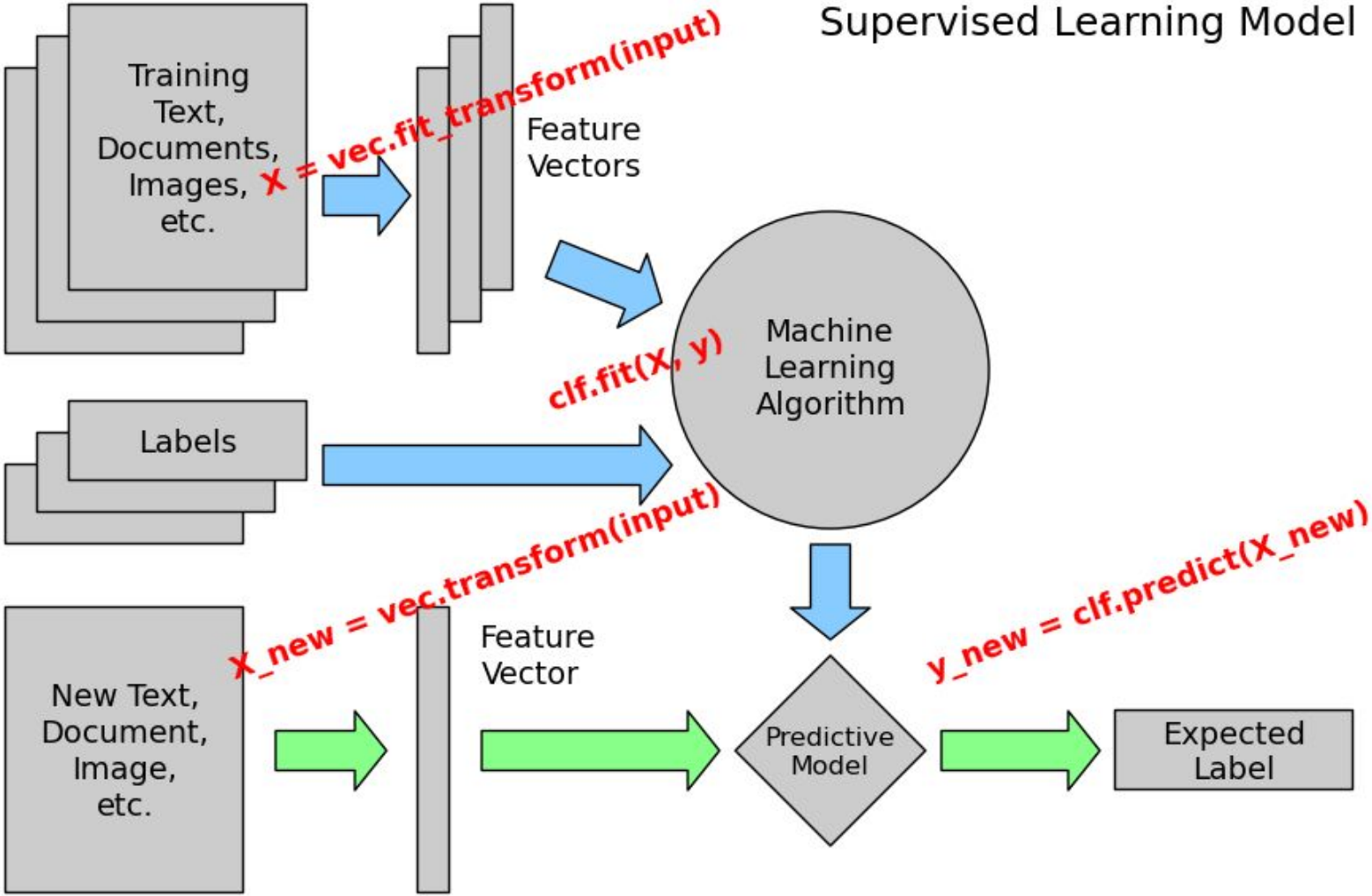

true
data
local linear
local constant
4
3.5
3
2.5
2
1.5
1
0.5
0
-0.5
0
10
20
30
40
50
60
70
80
90
100

>> ipython notebook

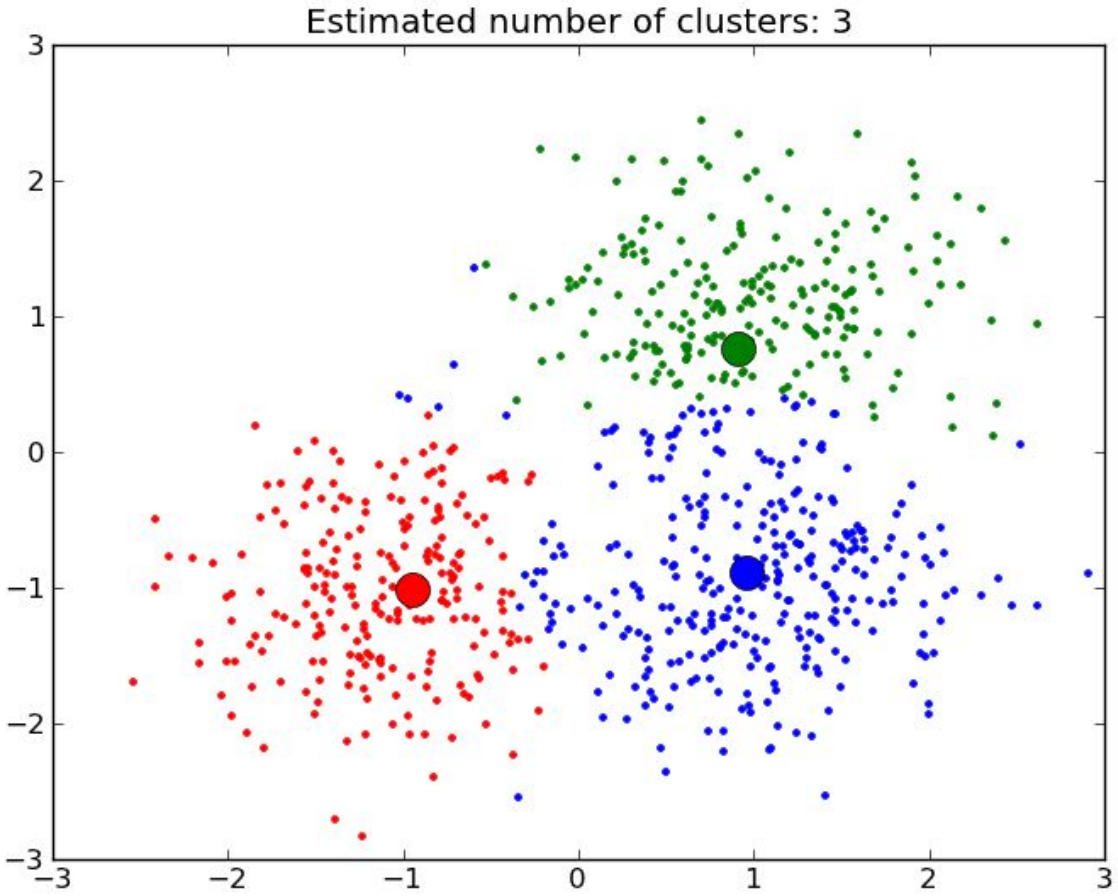
>Regression

Supervised Learning Model




Supervised Learning Model
Training Text, Documents, Images, etc.
X = vec.fit_transform(input)
Feature Vectors
clf.fit(X, y)
Machine Learning Algorithm
Labels
X_new = vec.transform(input)
Feature Vector
y_new = clf.predict(X_new)
New Text, Document, Image, etc.
Predictive Model
Expected Label

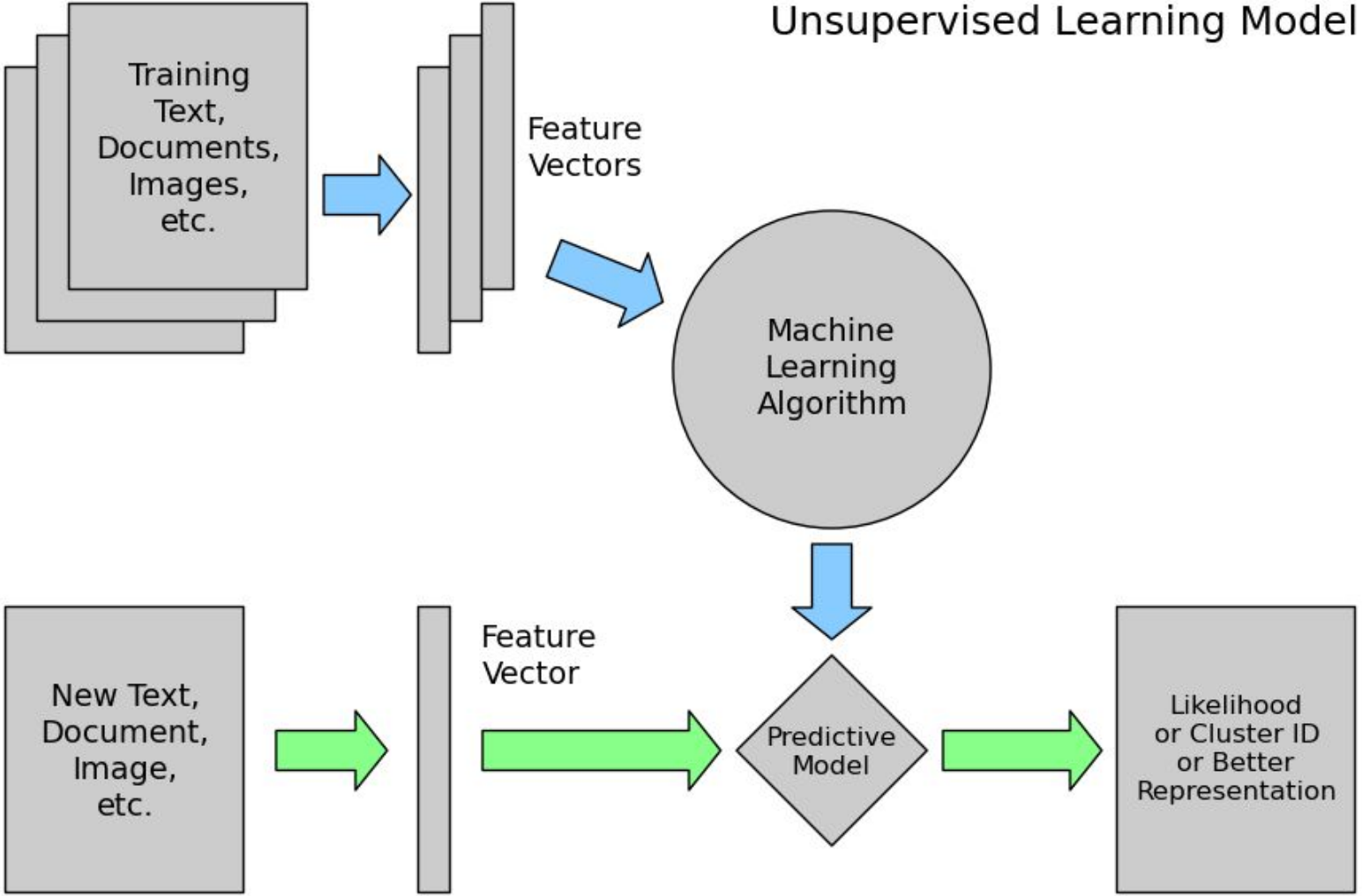
Clustering



>> ipython notebook

>Clustering

Unsupervised Learning Model



Want to learn more?

- https://www.coursera.org/learn/ml-foundations/ - Coursera specialization
- https://www.coursera.org/learn/machine-learning - Andrew Ng's, has its own history in BIT AI
- https://github.com/jakevdp/sklearn_pycon2015 - Sklearn tutorial