Introduction to Python for Computer Science

Why Python?

- **Batteries included** Rich collection of already existing bricks of classic numerical methods, plotting or data processing tools. We don't want to re-program the plotting of a curve, a Fourier transform or a fitting algorithm. Don't reinvent the wheel!

- **Easy to learn**

- **Easy to communication** To keep code alive within a lab or a company it should be as readable as a book by collaborators, students, or maybe customers.

- **Efficient code** Python numerical modules are computationally efficient.Python aims for quick development times and quick execution times.

- **Universal Python** is a language used for many different problems.

Scientific Python Ecosystem

- Unlike Matlab, or R, Python does not come with a pre-bundled set of modules for scientific computing.

Core Numeric Libraries and Packages

- **Numpy:** Numerical Computing with Powerful Numerical Arrays objects, and routines to manipulate them

- **Scipy:** High-level numerical routines. Optimization, regression, interpolation etc.

- **Matplotlib:** 2-D visualization, “publication-ready” plots

- **Pandas:** for statistics

- **Scikit-image** for image processing

- **Scikit-learn** for machine learning

Numpy

- The numpy package (module) is used in almost all numerical computation using Python

- provide high-performance vector, matrix and higher-dimensional data structures for Python.

- It is implemented in C and Fortran so when calculations are vectorized (formulated with vectors and matrices), performance is very good.

Numpy

- Why not simply use Python lists for computations instead of creating a new array type?

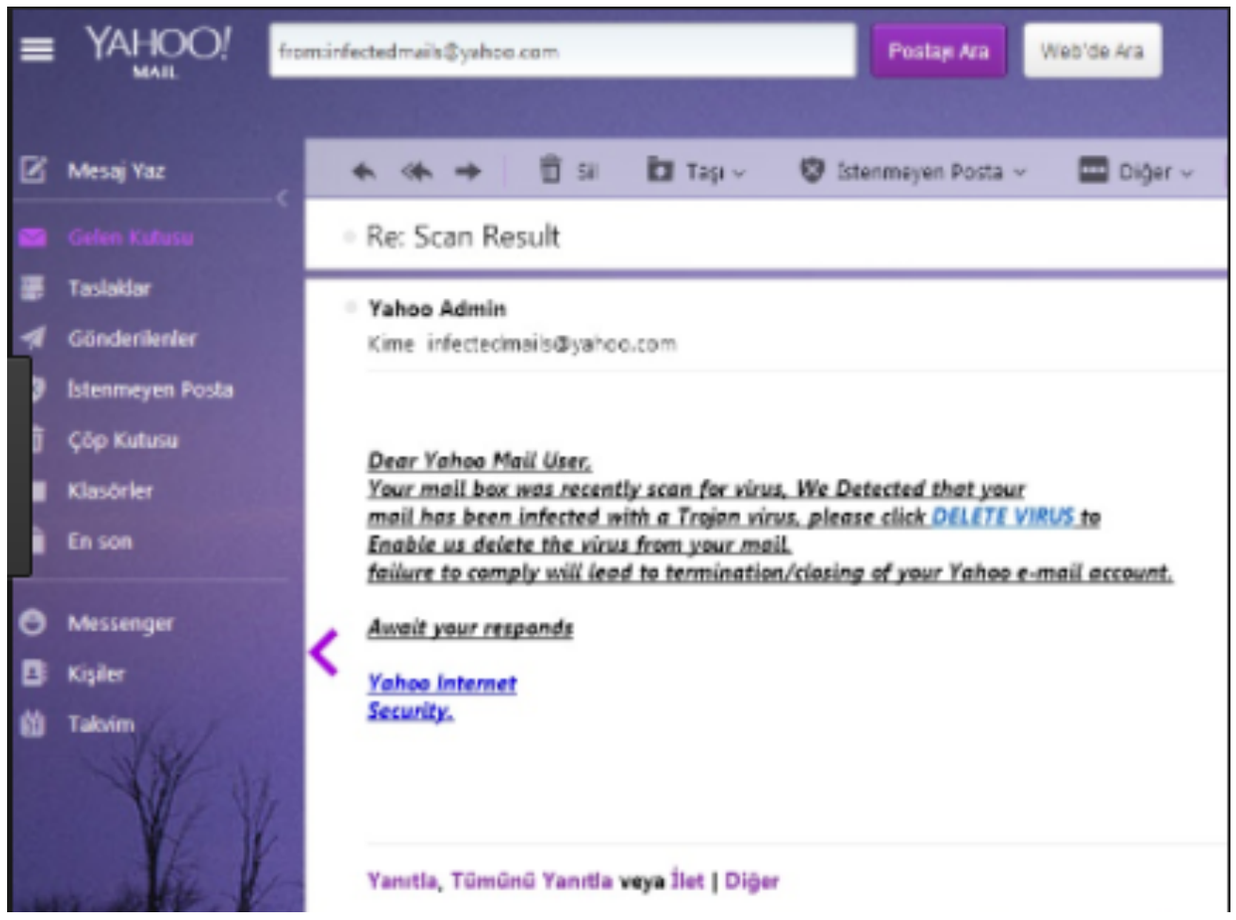
Machine Learning

-Traditional Programming-

-Machine Learning-

Spam Mail Filtering

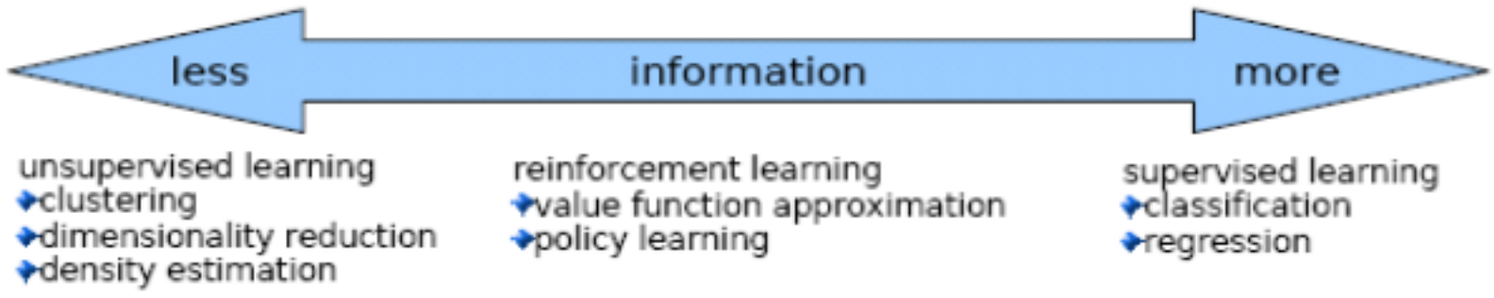
data



prediction

→ Spam vs Not Spam

Learning



- Supervised Learning
- Unsupervised Learning
- Reinforcement Learning

Spam Mail Filtering

- Input: email
- Output: spam/ham
- Setup:
  - Get a large collection of example emails, each labeled “spam” or “ham”
  - Note: someone has to hand label all this data!
  - Want to learn to predict labels of new, future emails

- Features: The attributes used to make the ham / spam decision
  - Words: FREE!
  - Text Patterns: $dd, CAPS
  - Non-text: SenderInContacts
  - …

Dear Sir.

First, I must solicit your confidence in this transaction, this is by virture of its nature as being utterly confidencial and top secret. …

TO BE REMOVED FROM FUTURE MAILINGS, SIMPLY REPLY TO THIS MESSAGE AND PUT "REMOVE" IN THE SUBJECT.

99 MILLION EMAIL ADDRESSES FOR ONLY $99

Ok, Iknow this is blatantly OT but I'm beginning to go insane. Had an old Dell Dimension XPS sitting in the corner and decided to put it to use, I know it was working pre being stuck in the corner, but when I plugged it in, hit the power nothing happened.

Supervised Learning

Data $D=\{d_1, d_2, \ldots, d_n\}$

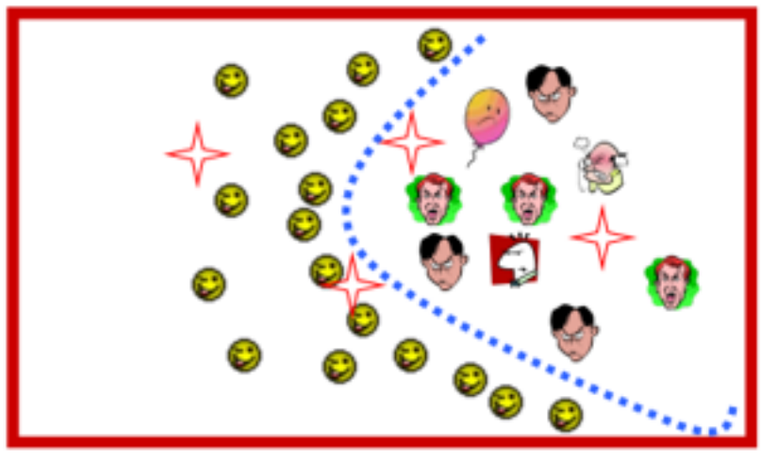
$d_i = < x_i, y_i >$

x_i : Input Vector

y_i : Output

<u>Aim</u> : $y_i \approx f(x_i)$

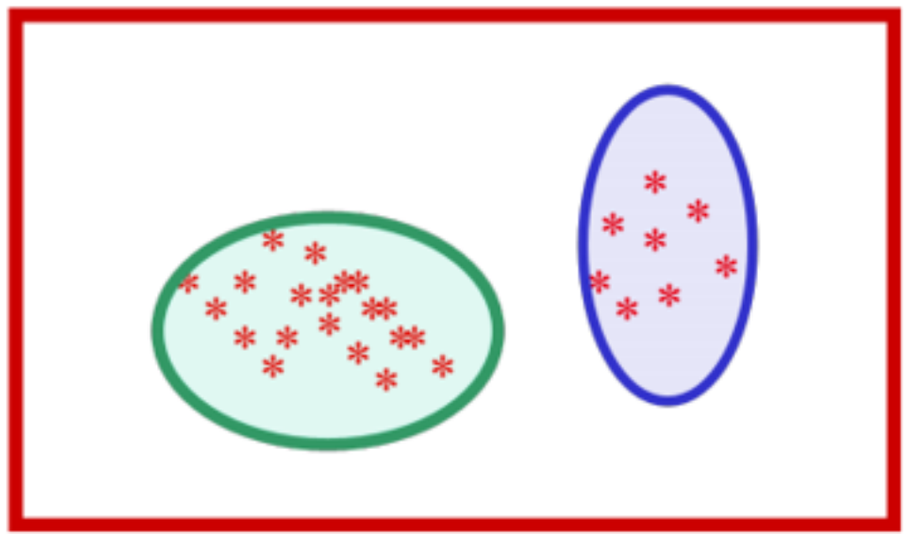
happy = f(x)



High Dimensional Feature (input) Space

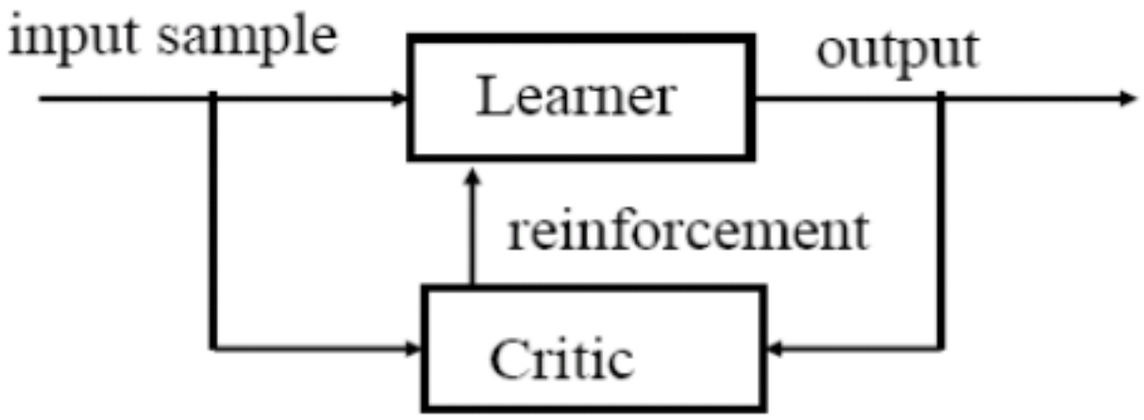
Unsupervised Learning

- What if we don’t know samples’ classes???



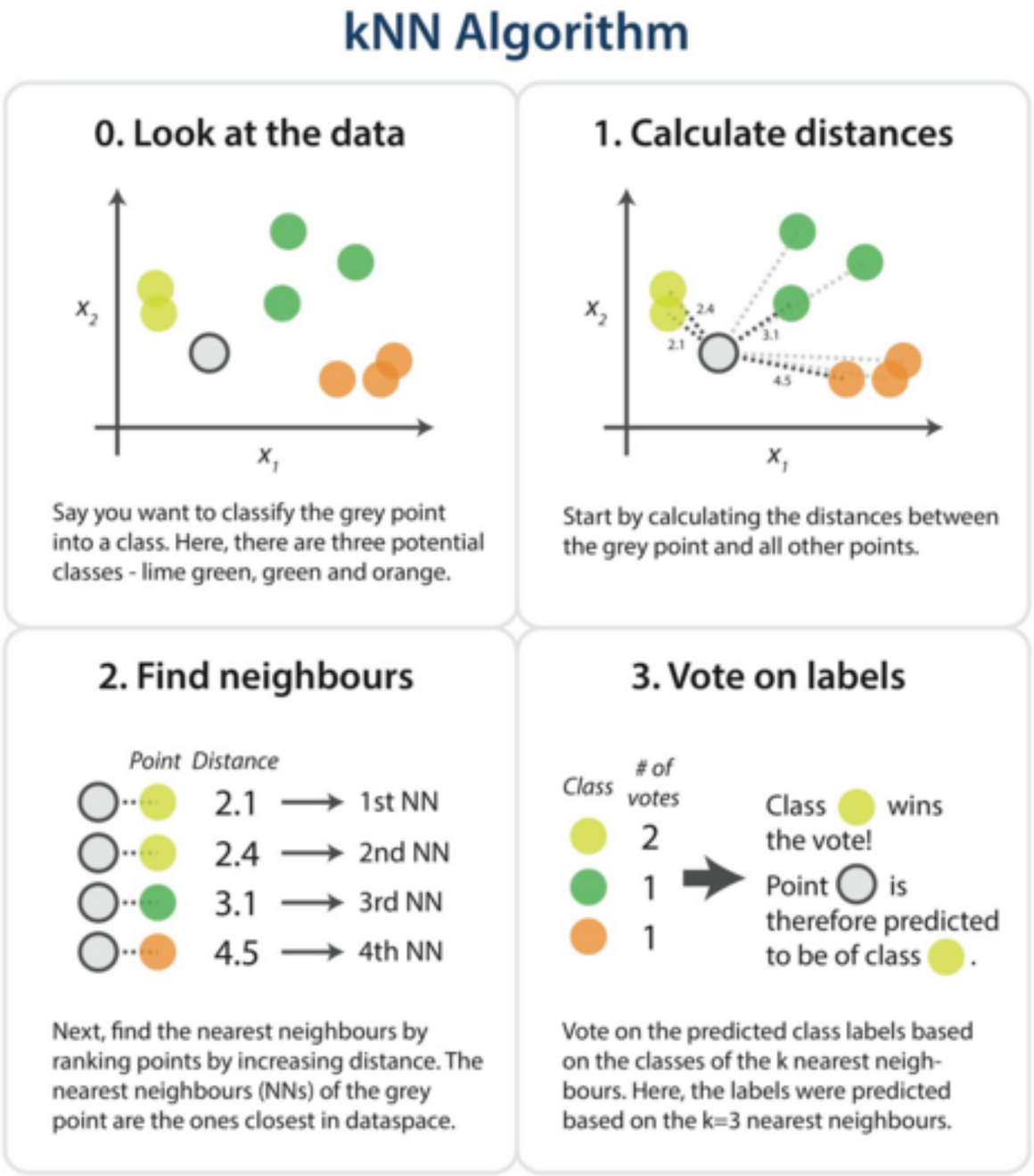
Reinforcement Learning

- We want to learn: $f : X \to Y$
- We see samples of **x** but not y
- Instead of y we get a feedback (reinforcement) from a **critic** about how good our output was

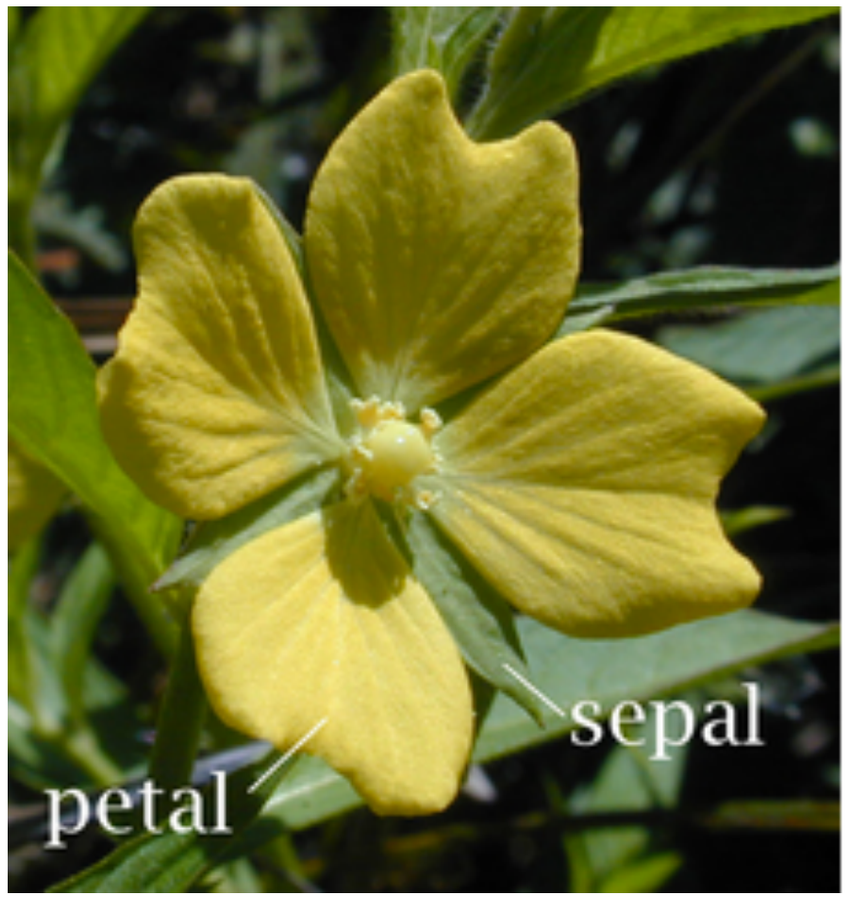


- The goal is to select outputs that lead to the best reinforcement

kNN Algorithm

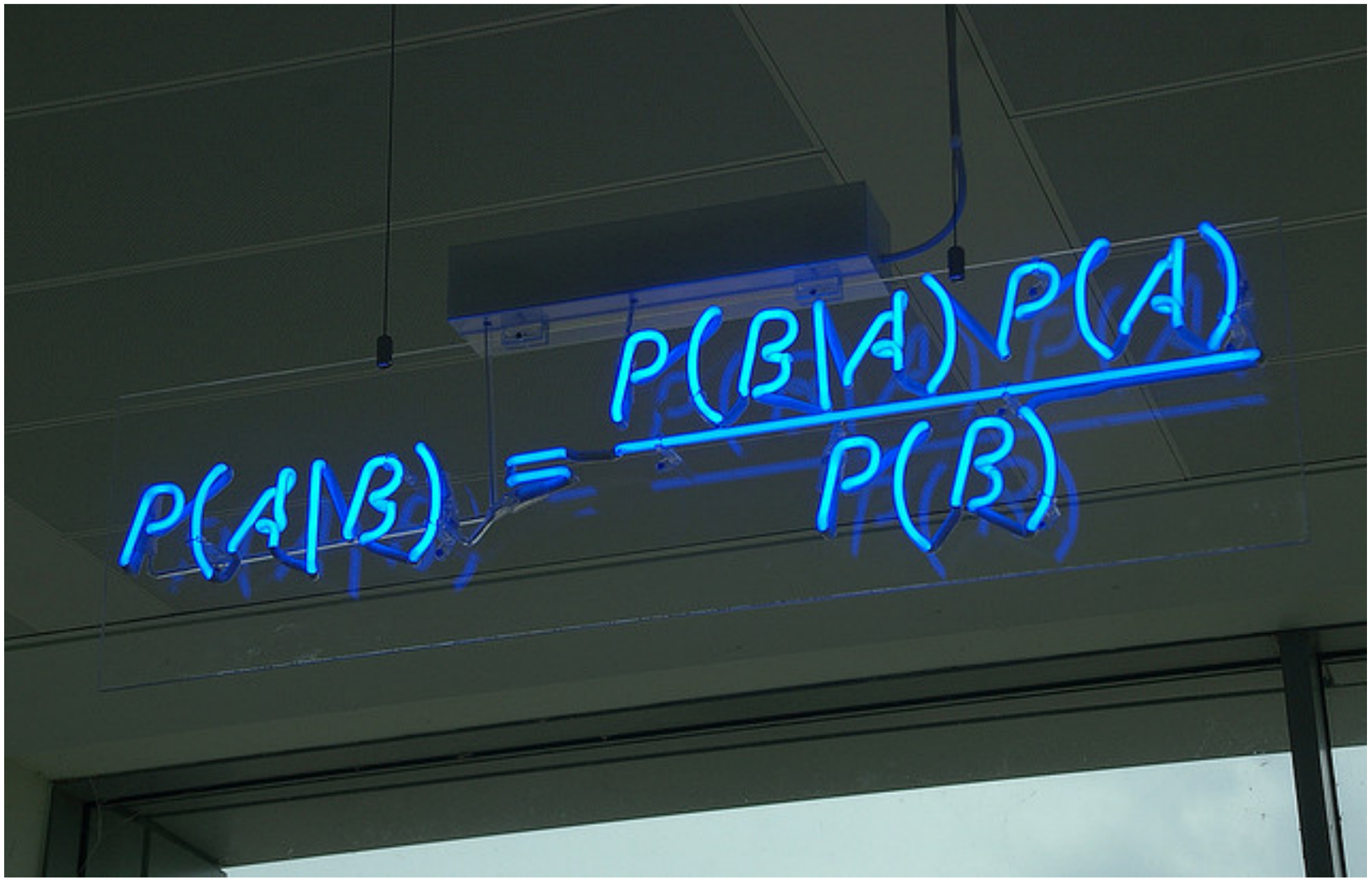


Demo



K Means Clustering

Naive Bayes



Demo

1. Handle the data (%67 - %33)
2. Train the Data
3. Make a Predictions
4. Evaluate Accuracy
5. Tie it Together

SVM

- http://www.scipy-lectures.org/intro/intro.html
- http://www.datasciencecentral.com/profiles/blogs/k-nearest-neighbor-algorithm-using-python