Introduction to machine learning concepts

QuakeCoRE Flagship 2

20 November 2018

Xavier Bellagamba

What it is and isn’t

Machine learning is:

- A good way to detect patterns
- A good way to fasten processes
- A set of statistical methods fitted using ‘randomness’

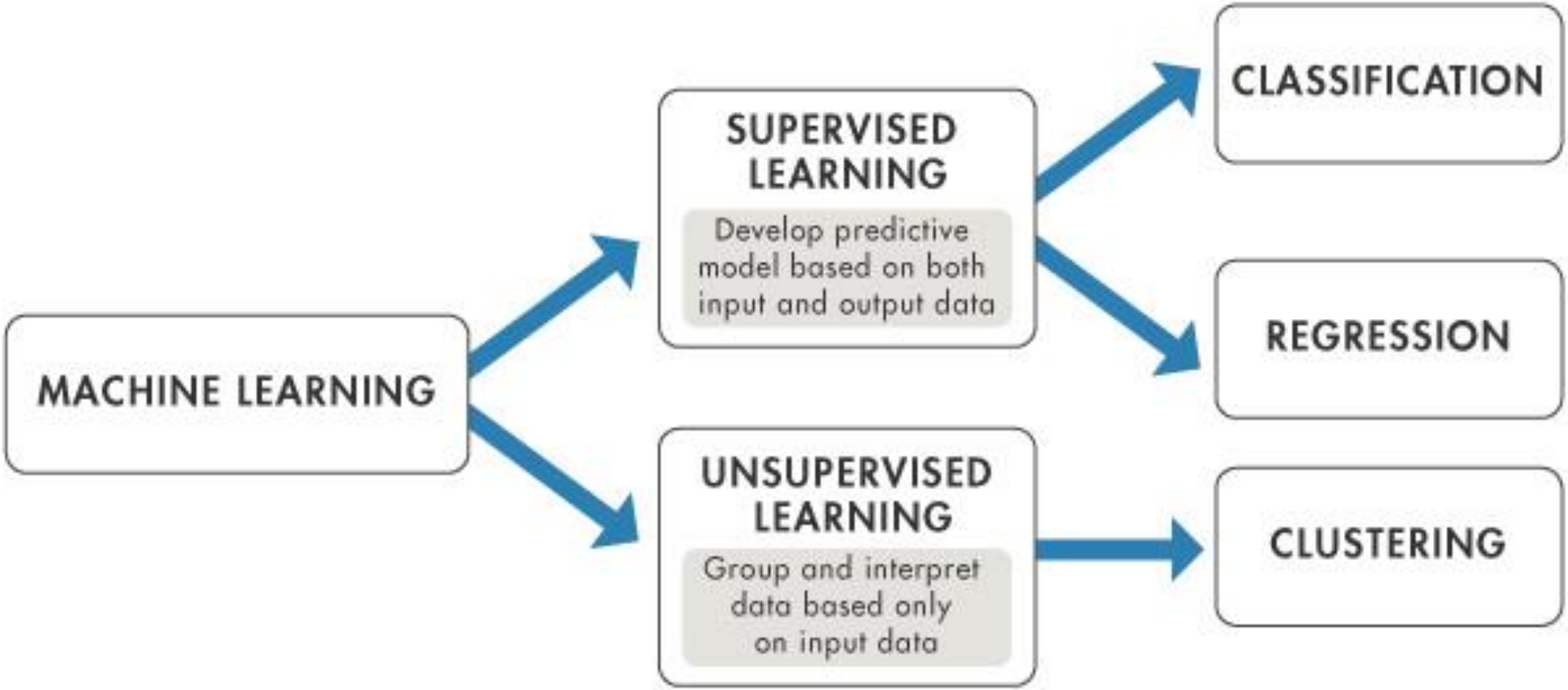
Machine learning is NOT:

- Black magic
- A solution to all problems
- A replacement for physics

When to use it

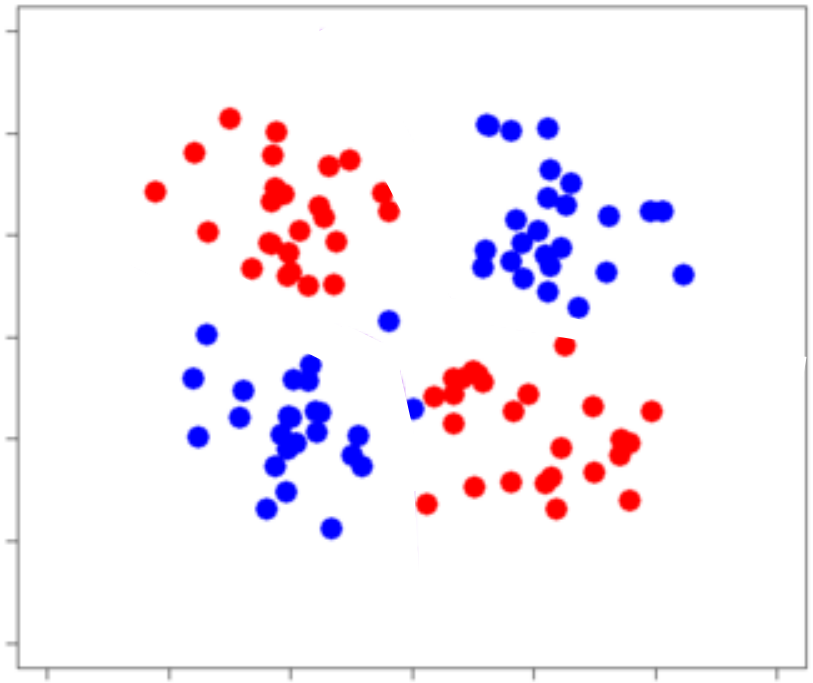
• $\boldsymbol{y} = f(x)$	Forward problem	Ballistic of a basket ball
• $y = f(\boldsymbol{x})$	Inverse problem	Decipher an encrypted email
• $y = \boldsymbol{f}(x)$	Where ML is useful	Would she like movie A

Types of algorithms

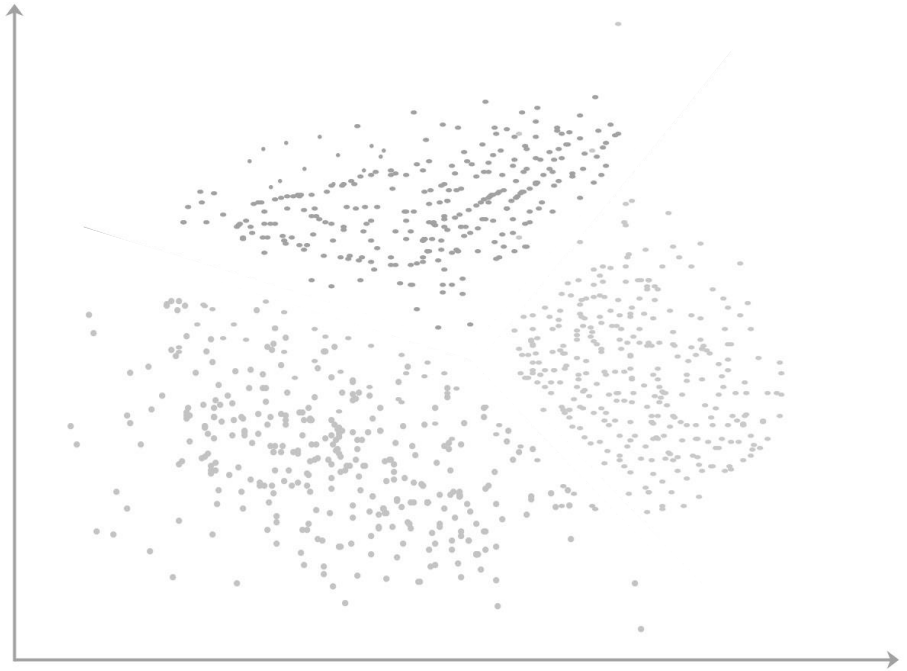


Types of algorithms

Supervised

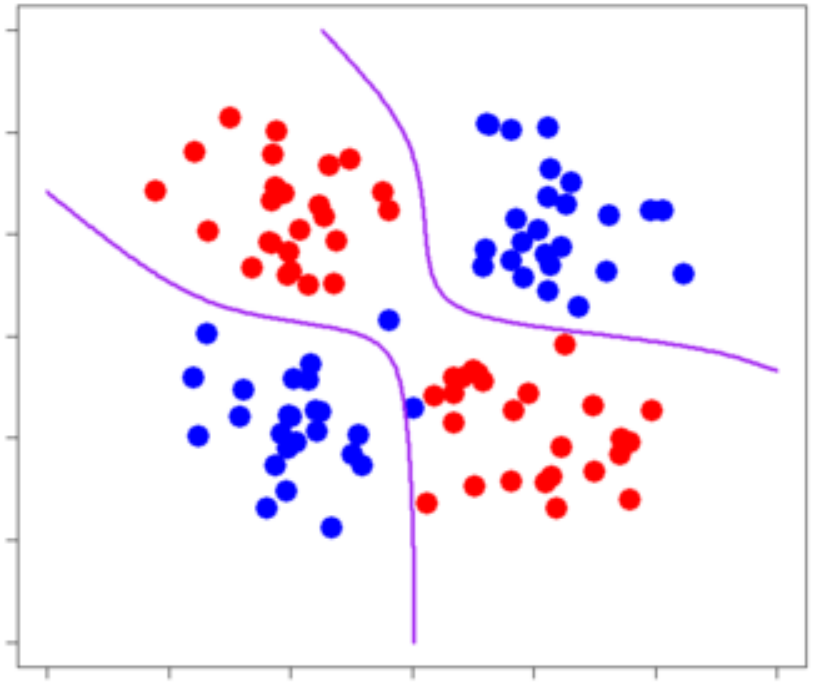


Unsupervised

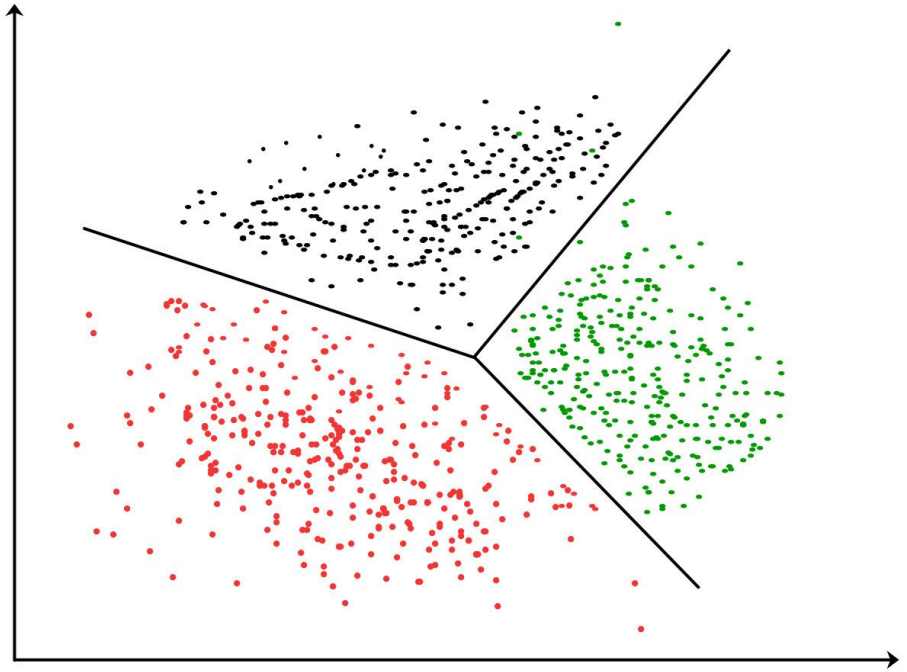


Types of algorithms

Supervised



Unsupervised



Types of algorithms

Supervised

- Logit regression
- Trees / Random forest
- Support-vector machines
- Neural networks

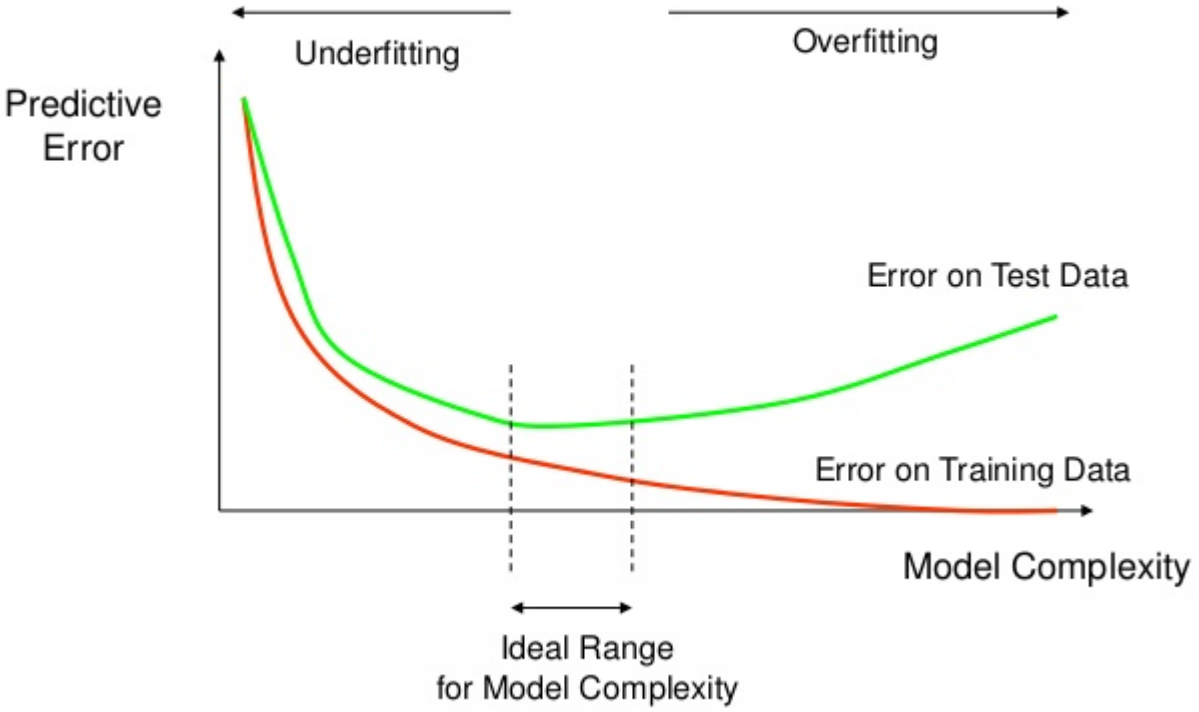
- And many more…

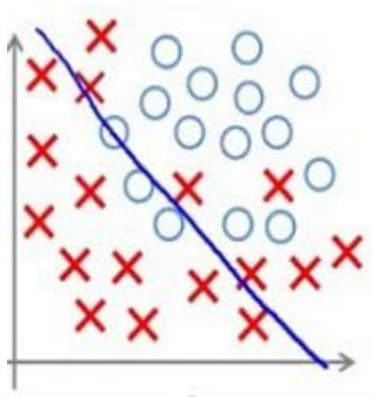
Unsupervised

- K-mean clustering
- Gaussian mixture
- Hierarchical clustering

- And many more…

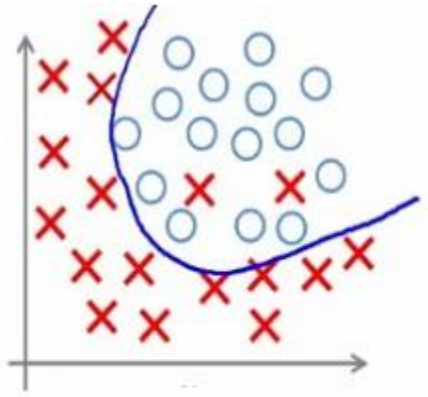
Overfitting



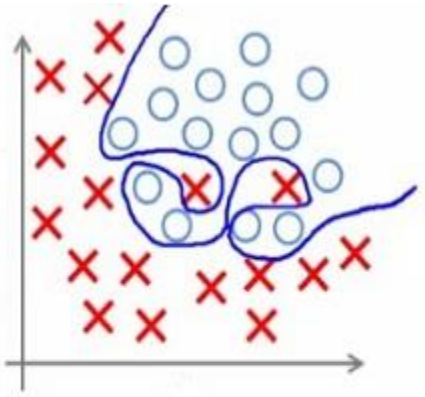


Under-fitting

(too simple to explain the variance)



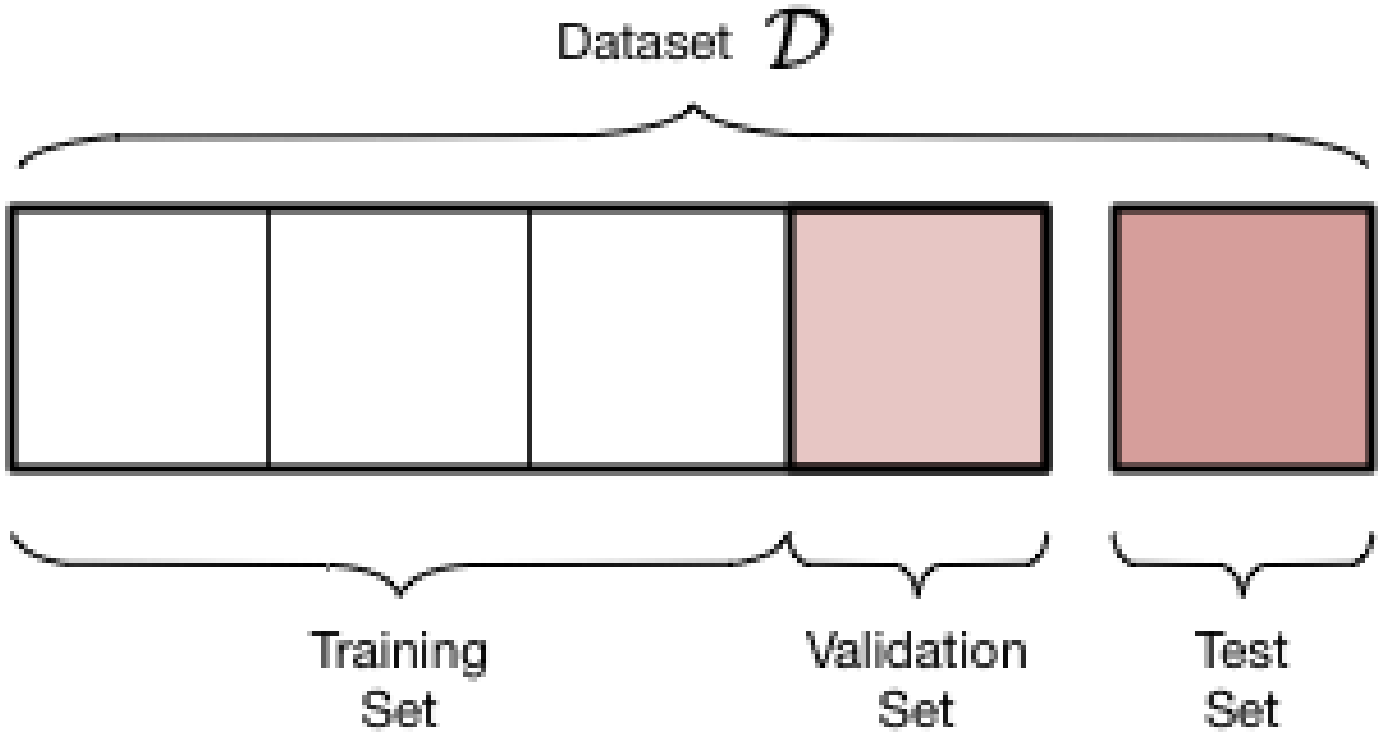
Appropriate-fitting



Over-fitting

(forcefitting -- too good to be true)

Overfitting: the solution



Types of algorithms – Comparison

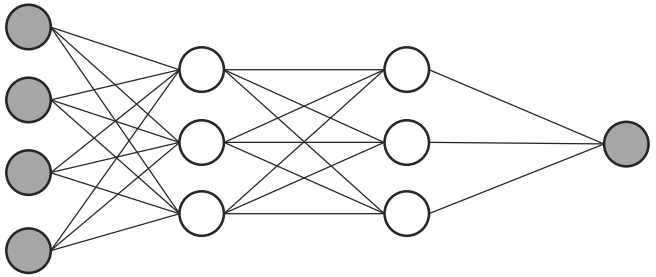
Characteristic	Neural Nets	SVM	Trees	MARS	k-NN, Kernels
Natural handling of data of "mixed" type	▼	▼	▲	▲	▼
Handling of missing values	▼	▼	▲	▲	▲
Robustness to outliers in input space	▼	▼	▲	▼	▲
Insensitive to monotone transformations of inputs	▼	▼	▲	▼	▼
Computational scalability (large N)	▼	▼	▲	▲	▼
Ability to deal with irrelevant inputs	▼	▼	▲	▲	▼
Ability to extract linear combinations of features	▲	▲	▼	▼	◆
Interpretability	▼	▼	◆	▲	▼
Predictive power	▲	▲	▼	◆	▲

Hastie et al. (2008) Elements of statistical learning, Table 10.8

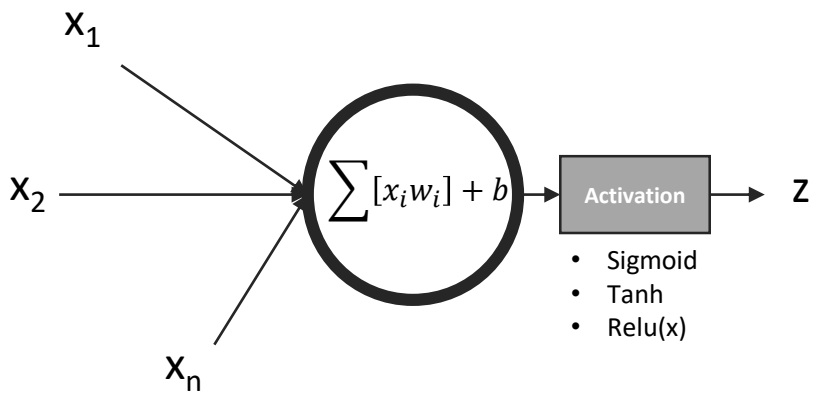
Supervised: Simple neural networks

Architecture:

Input layer Hidden layers Output layer



Anatomy of a neuron:



Training method:

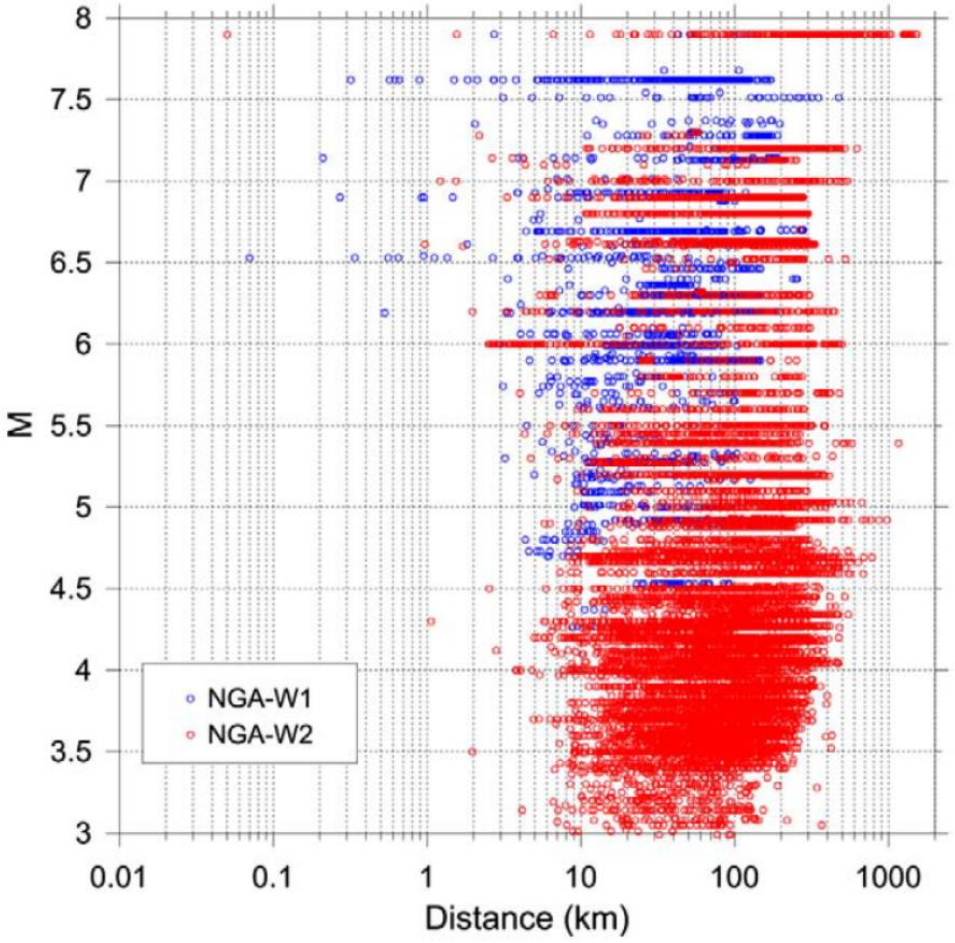
1. Initialize the network

2. Until stopping criteria is reached:

    1. For a randomly selected input, evaluate neural network output (i.e. get $\hat{y}$)
    2. From the predicted $\hat{y}$, evaluate the loss of the model L
    3. Evaluate the gradient of the loss on the output layer
    4. Backpropagate the gradient of the loss across the network
    5. Update the model parameters

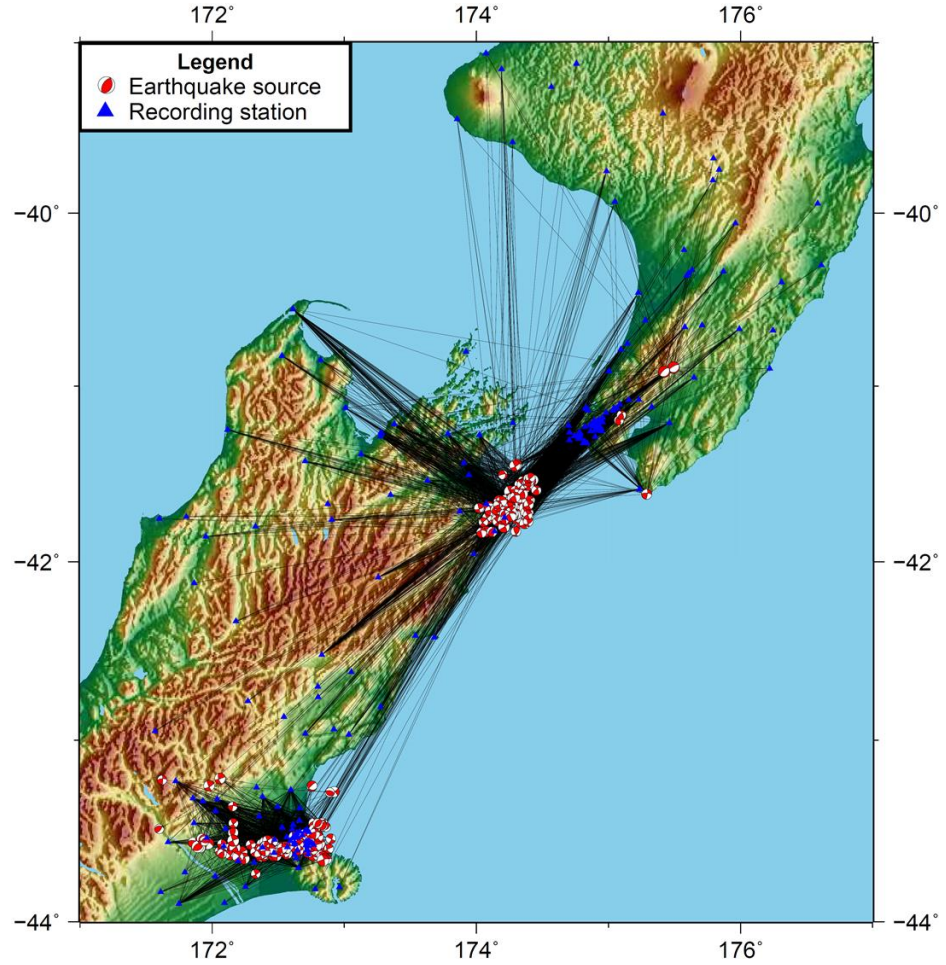
Supervised: Simple neural networks - Example

Automated classification of small-to-moderate GM records – motivations & data:



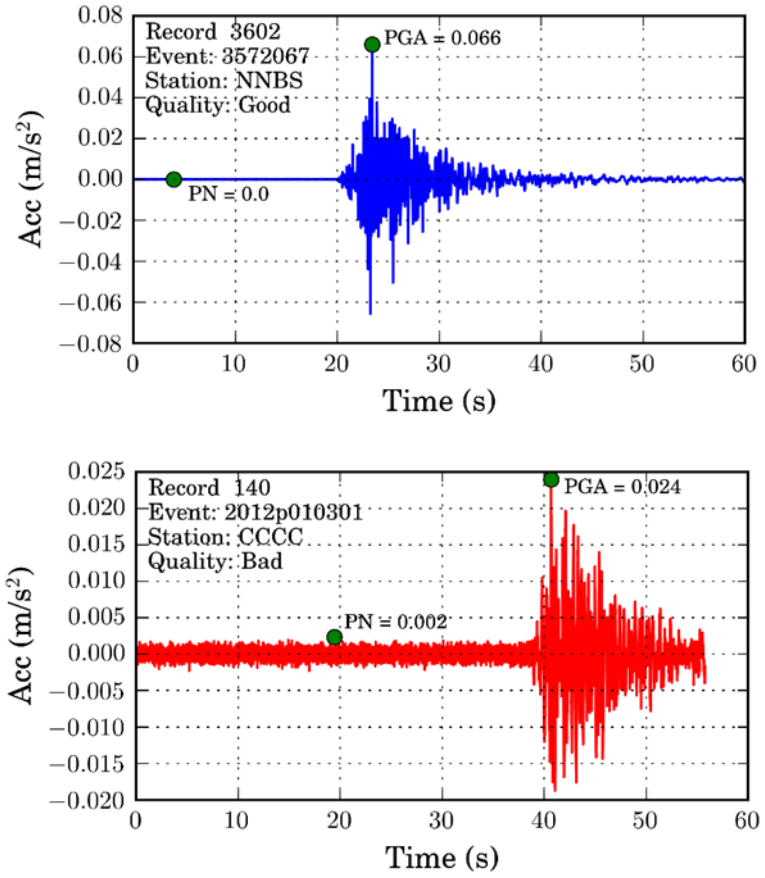
Metrics:

- Peak noise
- Peak noise / peak signal
- Fourier spectra ratio
- Duration
- Tail amplitude
- …



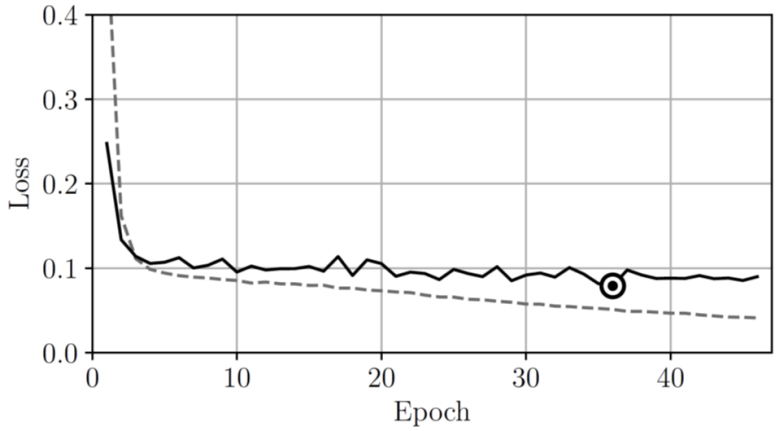
Supervised: Simple neural networks - Example

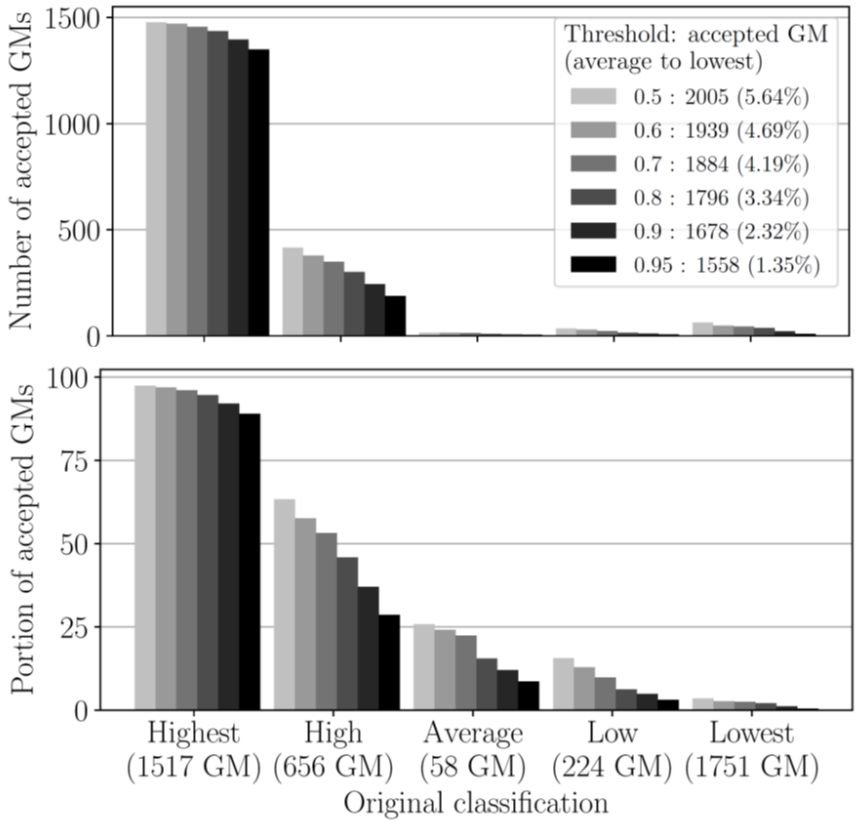
Automated classification of small-to-moderate GM records:



Architecture:

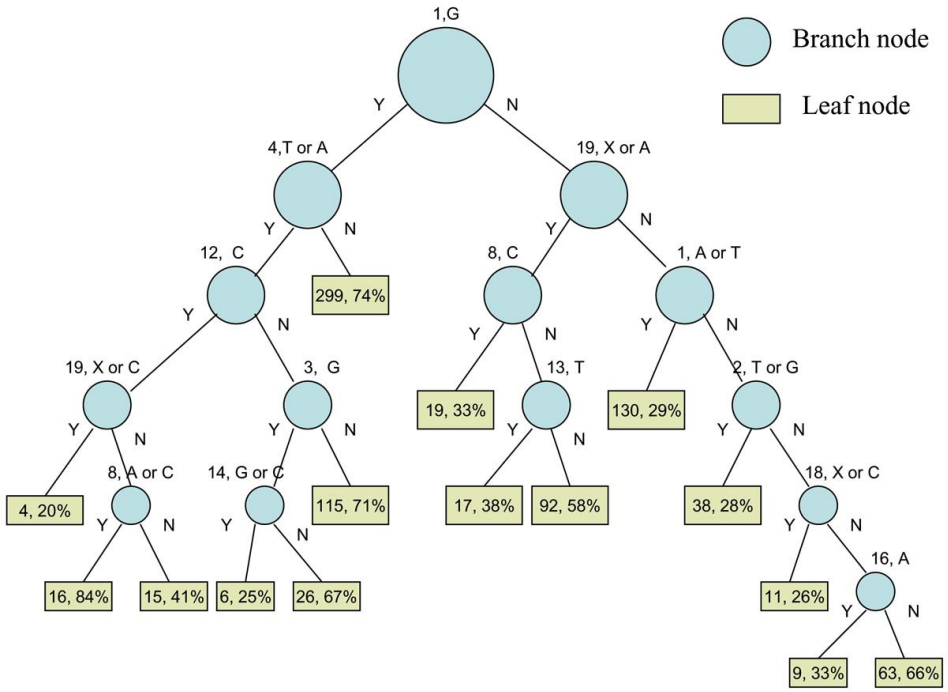
- 2 hidden layers
- N1 = 15
- N2 = 15





Supervised: Trees

Architecture:



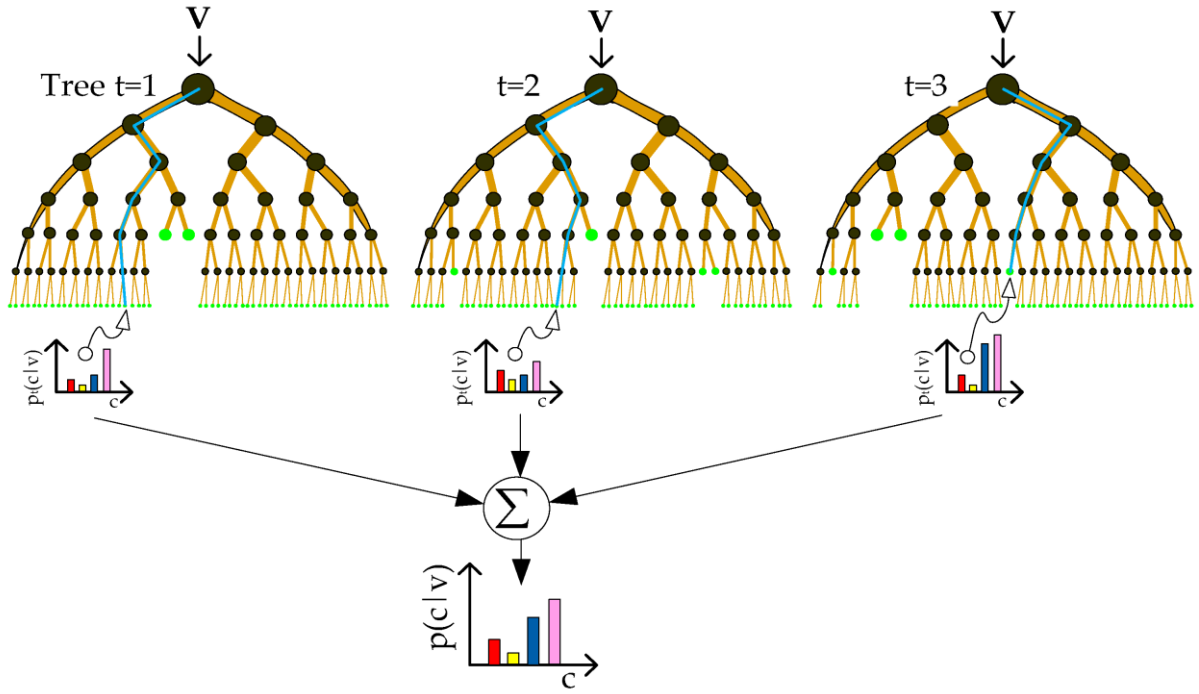
Training method:

As long as labels are not in leaves, construct branches by:

1. Randomly pick N variables
2. Chose the variable that has the best split
3. For the selected variable, determine its optimal value
4. Create branches or leaves

Supervised: Random forests

Architecture:



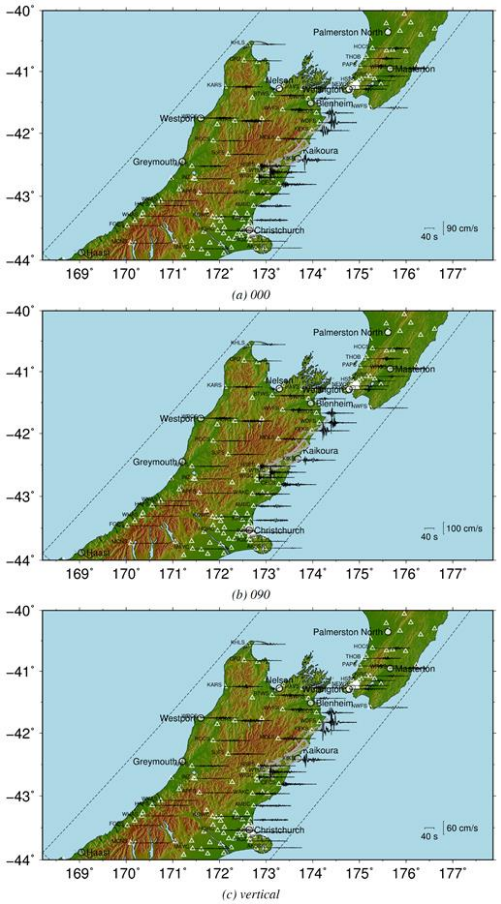
Training method:

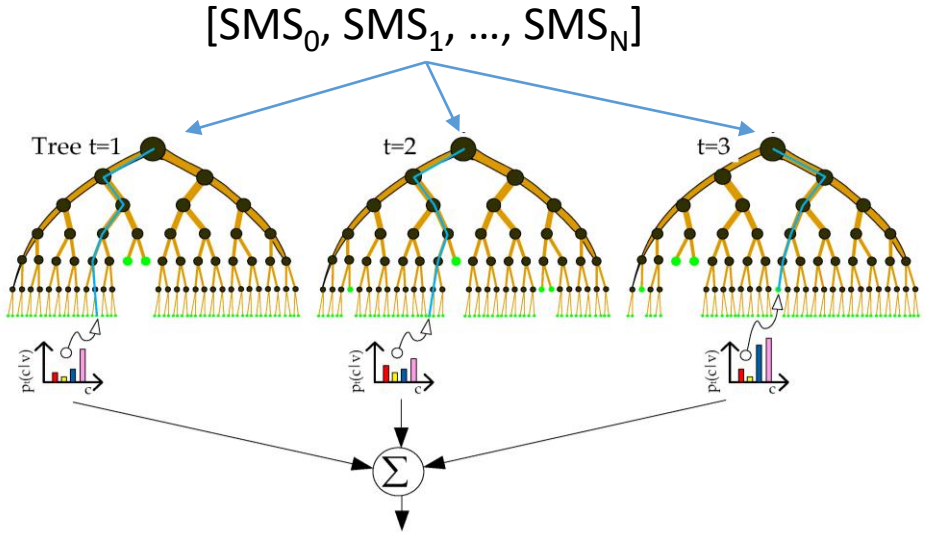
Train multiple trees.

To obtain result: get a *majority vote* among trees

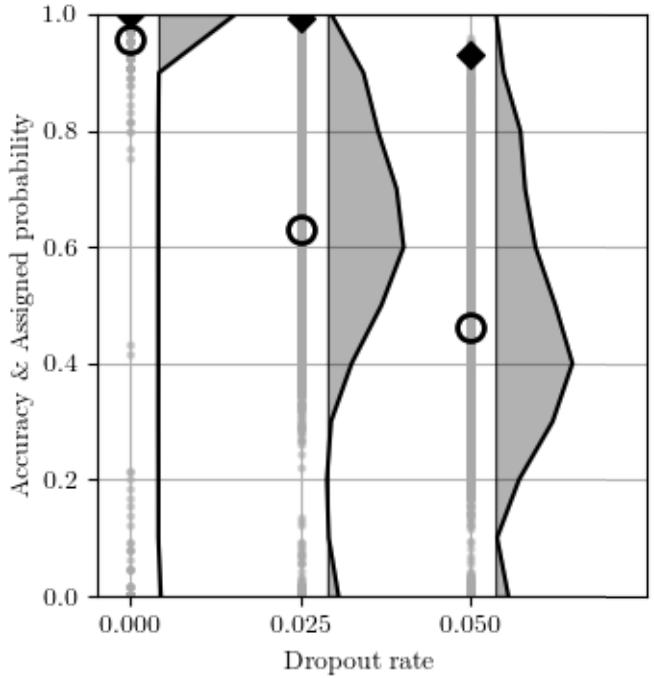
Supervised: Random forests – Example

Determining most likely ruptured fault based on real-time recordings:





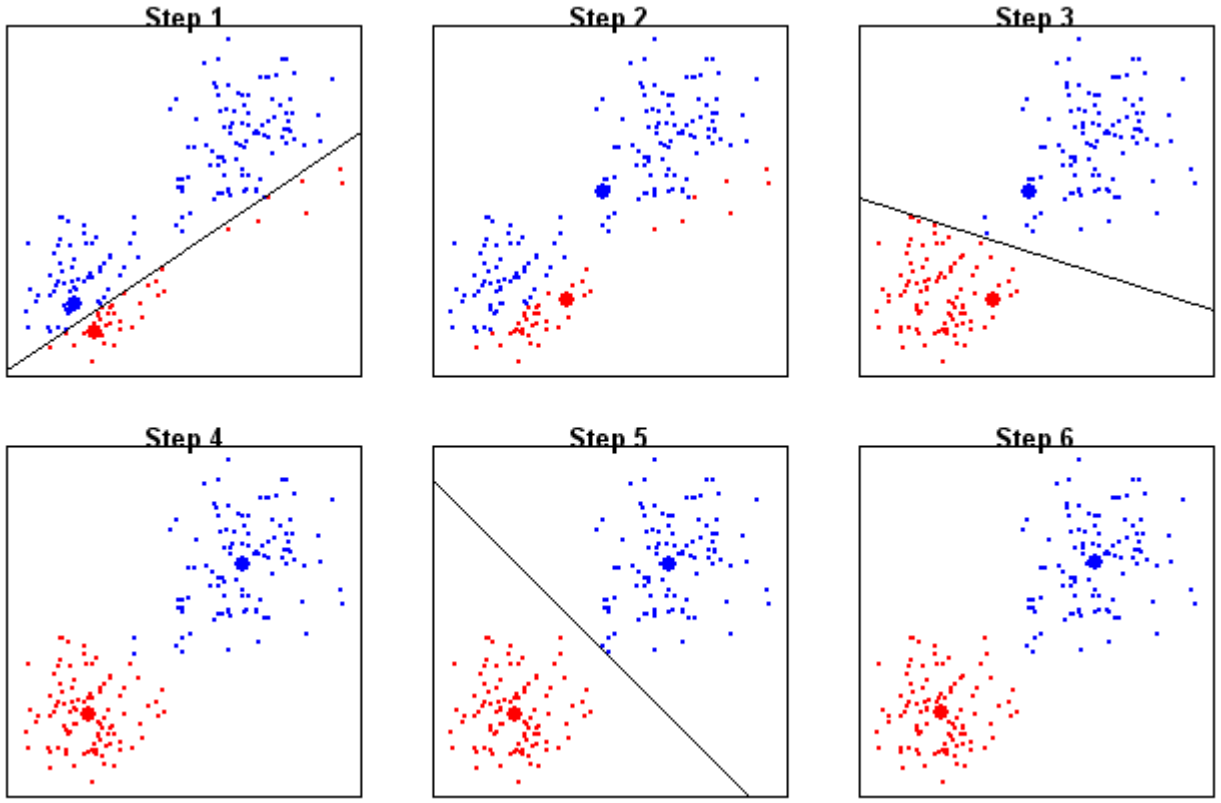
Kekerengu fault (62%)



This example is illustrative only, real results soon!!

Unsupervised: K-mean clustering

Architecture:



Training method:

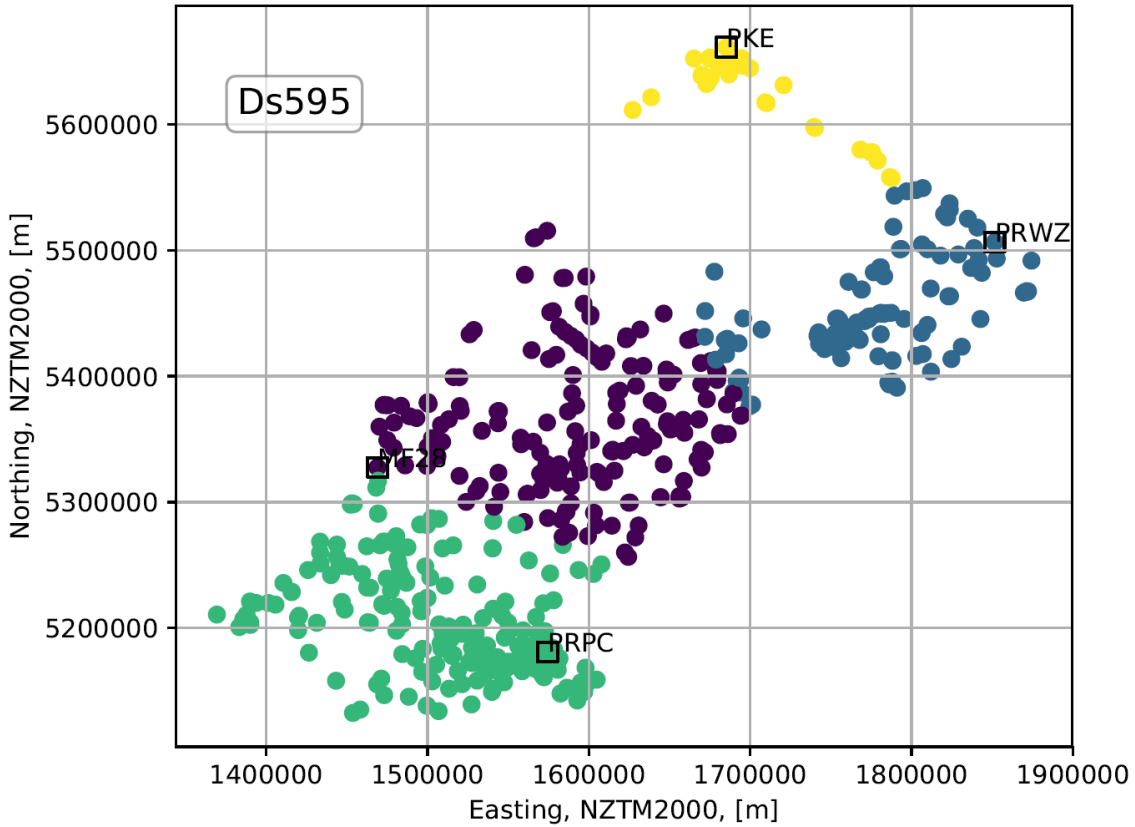
1. Plant N seeds
2. Determine ownership of data points
3. As long as seeds keep moving over iteration:
    1. Re-evaluate centroid of the seeds
    2. Change ownership of data points between cluster s.t. it minimizes *entropy* of the system

Unsupervised: K-mean clustering – Example

Cluster rupture based on ground motion records

From simulated GM in Cybershake:

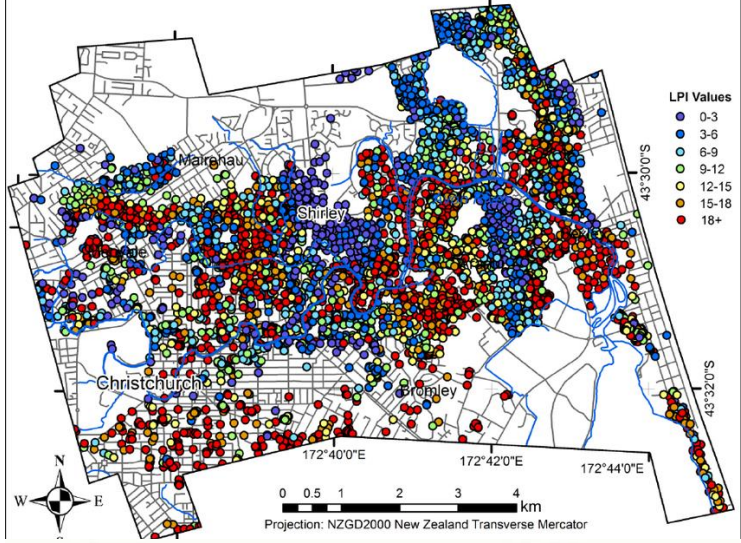
1. Extract results at GM stations
2. Cluster ground motion stations based on location
3. For each cluster: Select station with highest standard deviation
4. Perform K-mean clustering on the selected stations

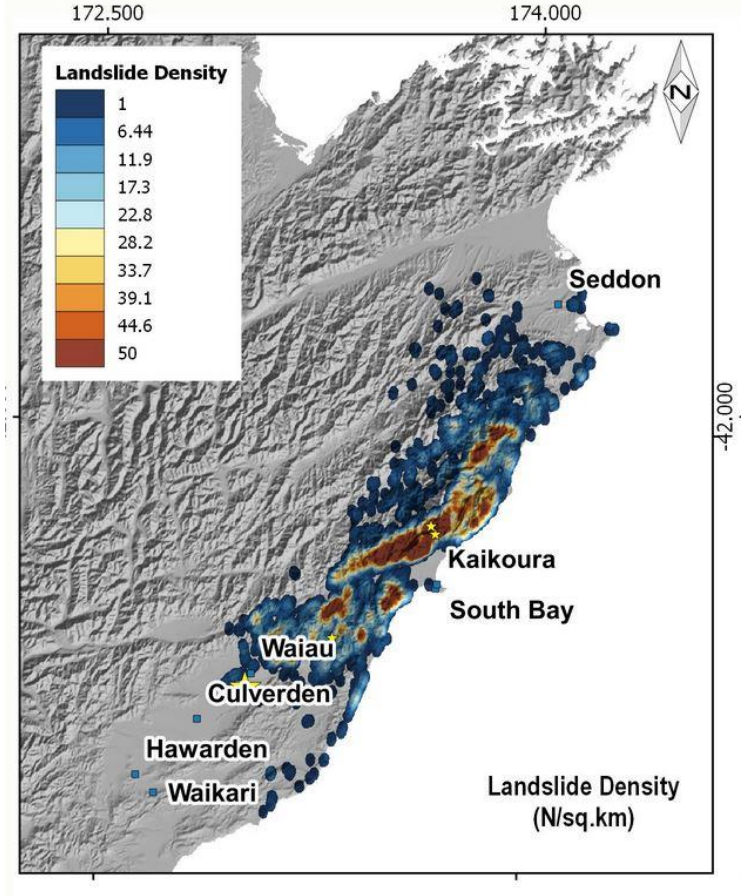


Potential applications in geotech

- Soil profile clustering
- Liquefaction prediction
- Landslide prediction
- Recommendation algorithm for foundation systems
- Surrogate model for soil-structure interaction

- … other ideas?





Recommended literature

Hastie et al. (2008) Elements of statistical learning, Springer Verlag

James et al. (2015) An introduction to statistical learning, Springer Verlag

Goodfellow et al. (2015) Deep learning, MIT press

https://machinelearningmastery.com/start-here/
https://developers.google.com/machine-learning/crash-course/
https://ai.google/education/

And many, many, many, many more

Computational burden

For most large ML – AI applications: **enormous.**

Recommend to:

- Get familiar with parallel coding (hyper-threading and MPI)
- Acquire some GPU capacity
- Work on HPC (combined with the above)

http://www.bluefern.canterbury.ac.nz/courses/

Recommended environment

Python 3.X (or R or Julia, but at all cost, not matlab, way too slow)

Machine learning: sklearn

Deep learning: keras with tensorflow back-end

Also recommend linux rather than Windows (no experience with Mac)

Hardware recommendation: GPU (https://developer.nvidia.com/gpu-grant-application)

Questions?