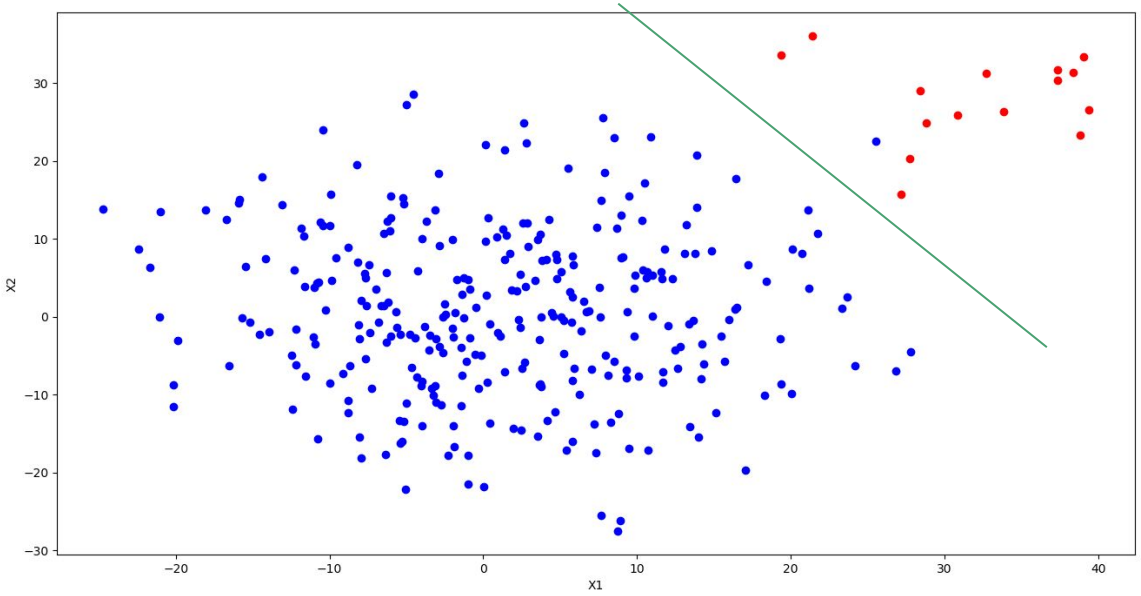
Anomaly Detection

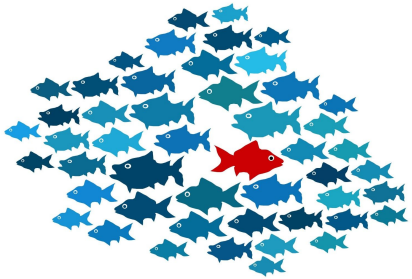
Algorithms in Machine Learning, ISAE-SUPAERO

Jérémy Pirard
Data Scientist
Airbus Commercial Aircraft

Anomaly detection: intuition

Build a model to detect anomalies (labeled in red here)... what do you do ?





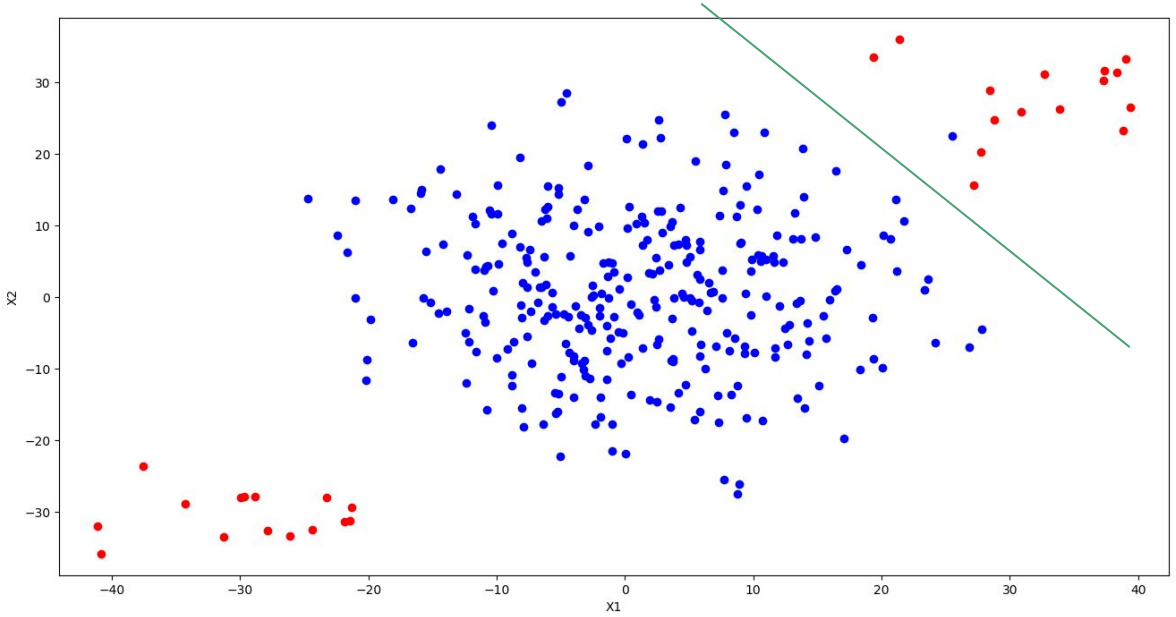
Supervised Learning?

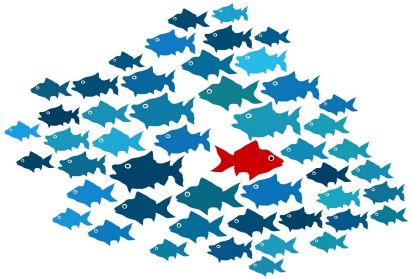
Naive bayes classifier, Random Forest, SVM...

➔ Features = X1, X2
➔ Label = Anomaly or not (0 or 1)

Anomaly detection: intuition

Build a model to detect anomalies (labeled in red here)... what do you do ?

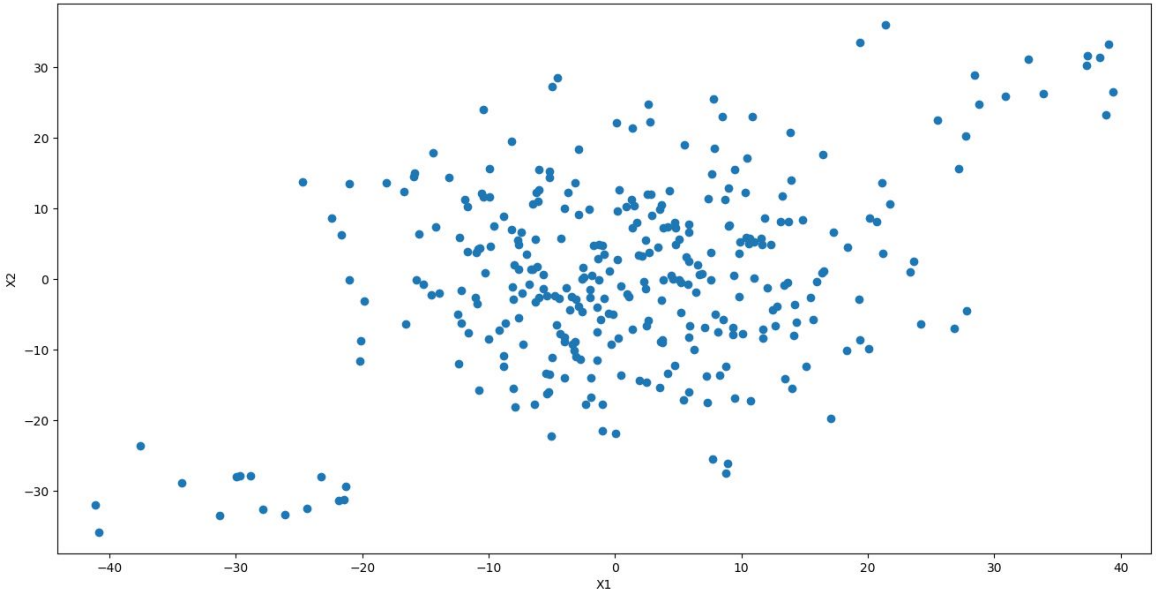


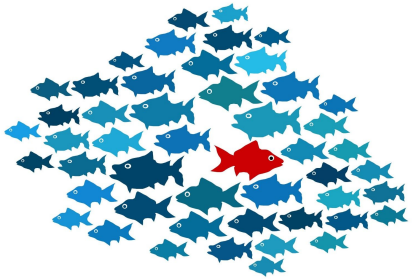


What if new anomalies?

Anomaly detection: intuition

Build a model to detect anomalies... what do you do ?





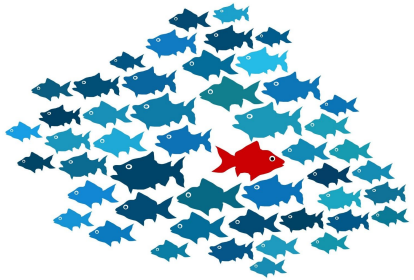
What if no label?

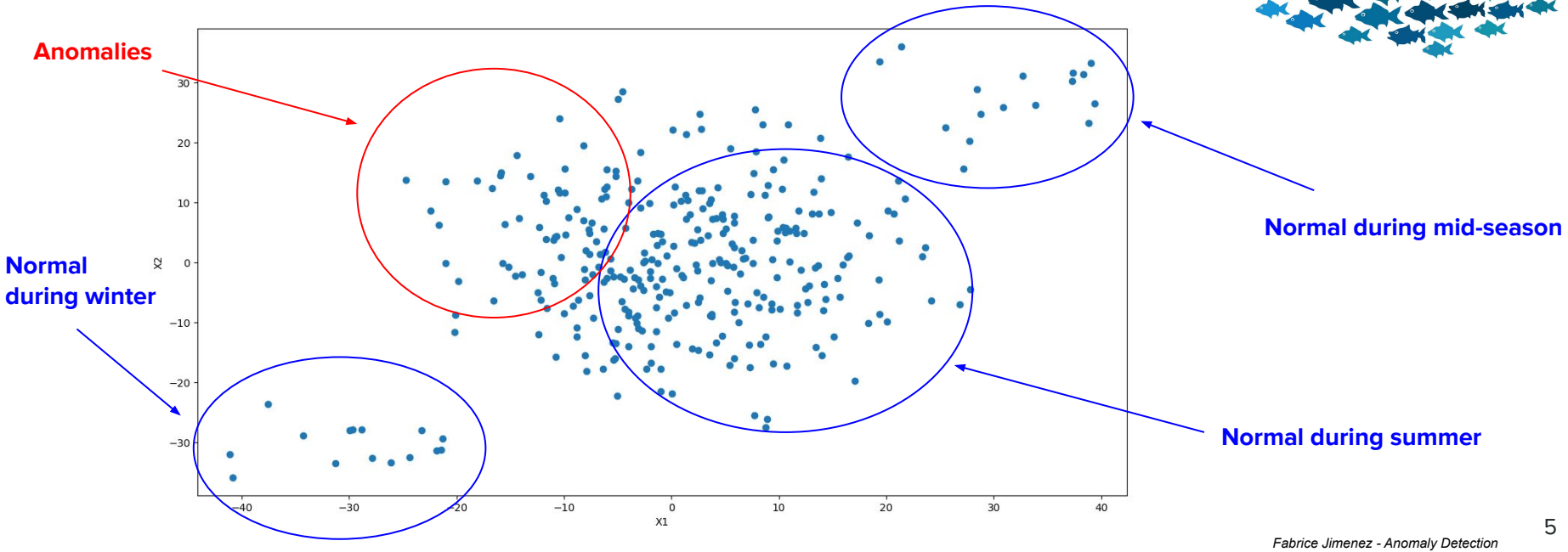
Anomaly detection: definition and scope

What is an anomaly?

1/ Generally: a rare individual (row) in a dataset that differs significantly from the majority of the data

2/ Sometimes: anomalies are not so rare, and may not be so different from the majority of the data...





Anomaly detection: definition and scope

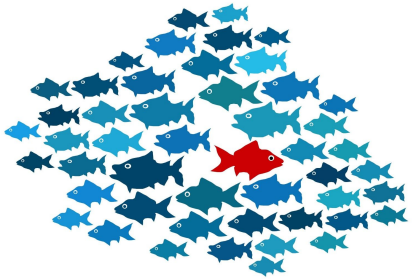
Why not using Supervised Learning with labeled dataset?

Very unbalanced dataset

5 anomalies given 100 000 normal points...

Lack of coverage of all anomaly types

Anomaly = something not expected, what if a new type happens...



We need other approaches...

Outlier detection: the dataset contains anomalies in the sense of statement 1/ (rare + statistically different)

➔ Detect elements in this same dataset which differ from the majority of the data

Novelty detection: you have a clean dataset without anomalies (in the sense of 1/ or 2/)

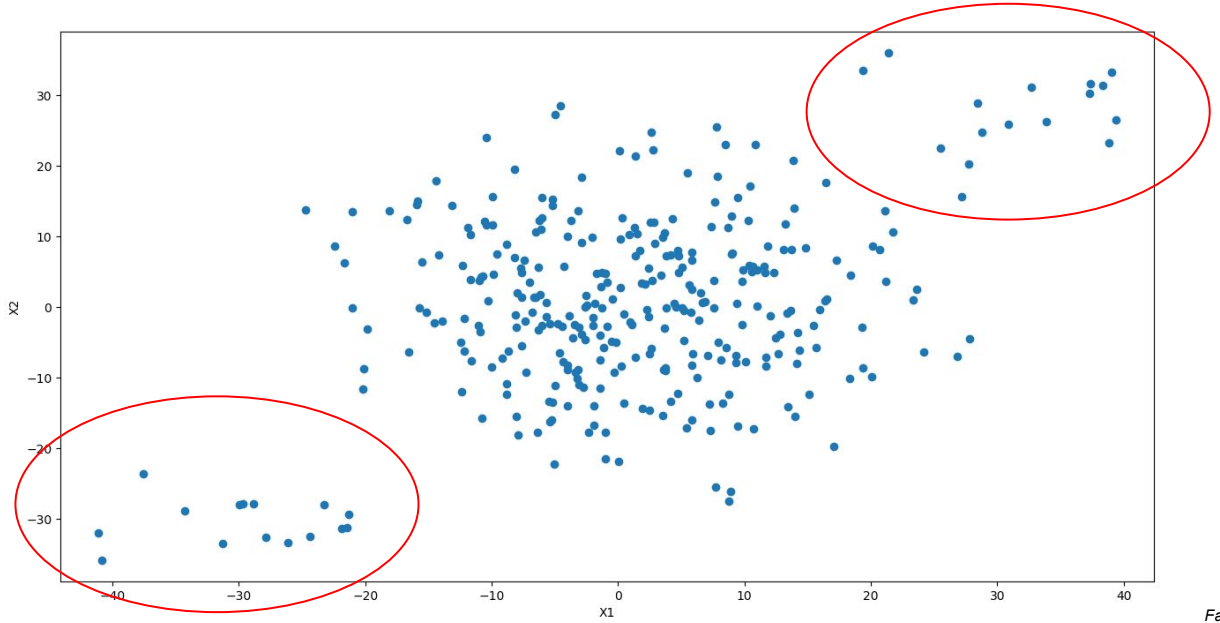
➔ Learn the normal behavior, to be able to check if a new item is normal or an anomaly

➔ Some techniques can be used for both, but **be aware** of the approach you are using, and why...

Outlier detection

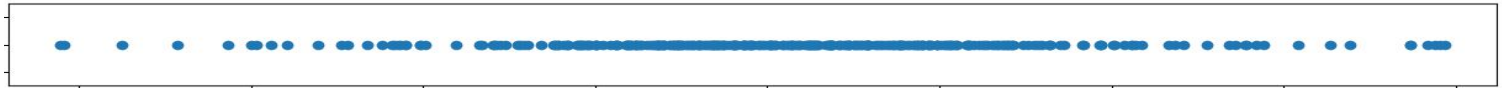
1/ Anomaly = a rare individual (row) in a dataset that differs significantly from the majority of the data

Outlier detection: the dataset contains anomalies in the sense of statement 1/
➜ Detect elements in this same dataset which differ from the majority of the data

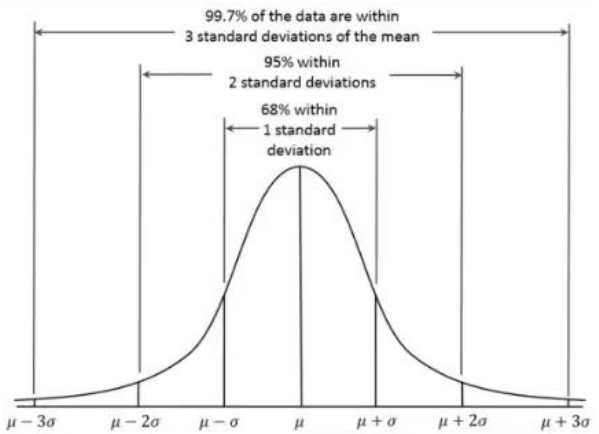


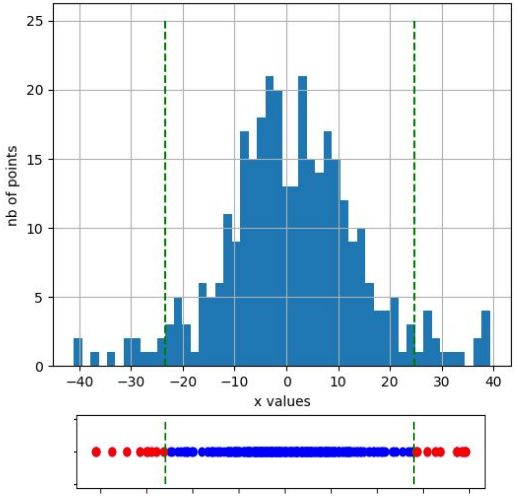
Outlier detection: 1D

Example



Univariate case: in 1 dimension (1 variable), how would you detect anomalies?





Remember your normal distribution!

➔ Mean and Std help quantify density of data

➔ outliers = points outside [mean - 2xstd, mean + 2xstd]

Outlier detection: 1D

Are mean and std always reliable?

They quantify the data density in the case of **normal distribution**... It is not always the case!

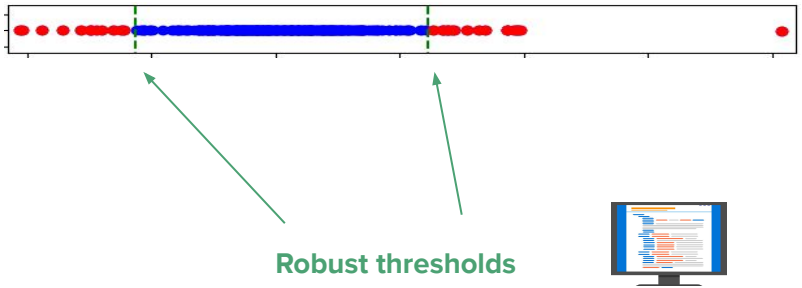
Sensitive to outliers!
If too far outliers or many outliers → **distorts estimation**!

What is the alternative? Let's go MAD!

Robustify mean? → median

Robustify standard deviation? → ...

MAD = Median Absolute Deviation = median(| x - median(x) |)



Example:
→ outliers = points outside [median - 3xMAD, median + 3xMAD]

What threshold to use? Why 3 x MAD?
→ there are relationships to quantify percentiles with median and MAD
→ they depend on the type of distribution...

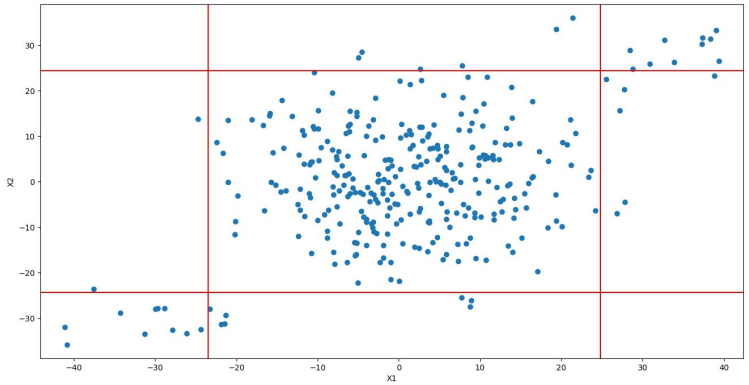
→ **Thresholds always need human intervention / fine-tuning!**

Outlier detection: nD

Multivariate case: generalize what we saw in 1D ?

➔ 1st approach: median and MAD on each of the variables (still univariate...)

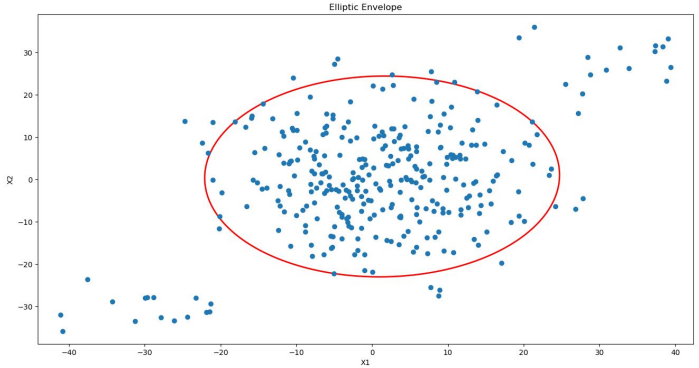
It does not take at all into account **relationship** between x1 and x2..



➔ We can use covariance matrix!

$$\sum = \frac{1}{n} \times \left(M - \bar{M}\right)^T . \left(M - \bar{M}\right)$$

$$M = \begin{bmatrix} X_{1\,1} & \dots & X_{1\,p} \\ \dots & \dots & \dots \\ X_{n\,1} & \dots & X_{n\,p} \end{bmatrix}$$



Elliptic envelope
➔ Threshold to define!

Mahalanobis distance of a point to the distribution:

$$X = (X_1, X_2, ..., X_n)$$

$$D\left(X, M\right) = \sqrt{\left(X - \mu\right)^T \Sigma^{-1} \left(X - \mu\right)}$$

If scaled distribution, Euclidean distance to the center!

For more details on robust covariance estimator (FastMCD algorithm):
A Fast Algorithm for the Minimum Covariance Determinant Estimator
Peter J. Rousseeuw and Katrien Van Driessen

Outlier detection: nD

Minimum Covariance Determinant (MCD)

➙ 1/ Randomly select a subset of datapoint

➙ 2/ Calculate the covariance matrix, its determinant and mean on the subset

➙ 3/ Repeat 1 and 2 several times and keep the matrix with smallest determinant

The determinant of the covariance matrix "measures" how broad a distribution is

➙ 4/ Compute the Mahalanobis distance for each observation based on previous estimation.

... Again, threshold to be defined ...

For more details on robust covariance estimator (FastMCD algorithm):
A Fast Algorithm for the Minimum Covariance Determinant Estimator
Peter J. Rousseeuw and Katrien Van Driessen

Outlier detection: nD

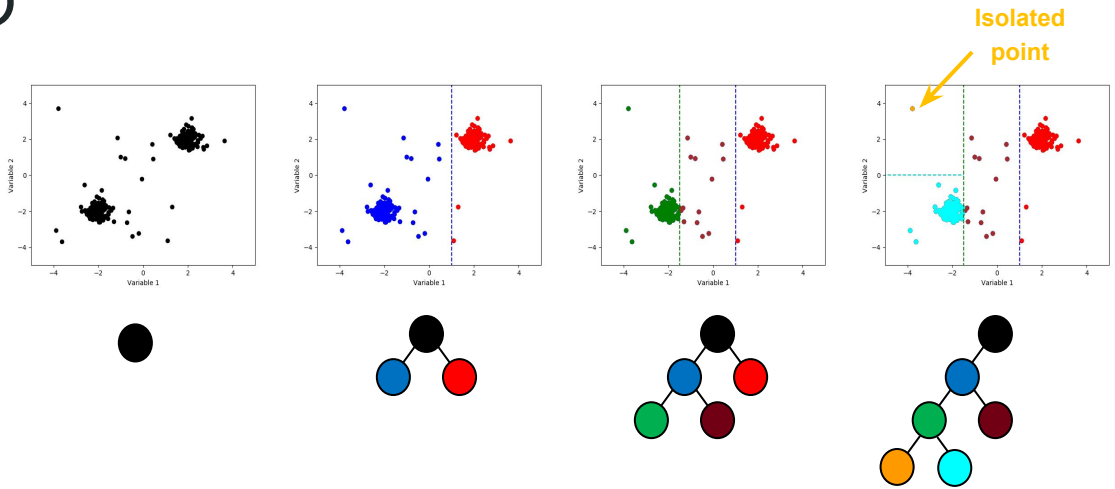
Other methods - Isolation Forest

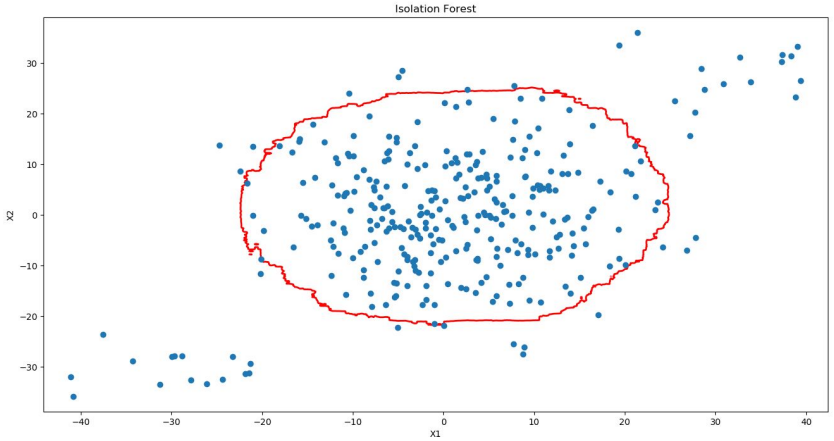
➔ 1/ Build Isolation Tree:
Split entire dataset with random variables and random thresholds

➔ 2/ Repeat with 100, 1000 trees...

➔ 3/ Average depth of a point in the forest = anomaly score*

Low depth = high anomaly score
High depth = low anomaly score





Once again, threshold to define!

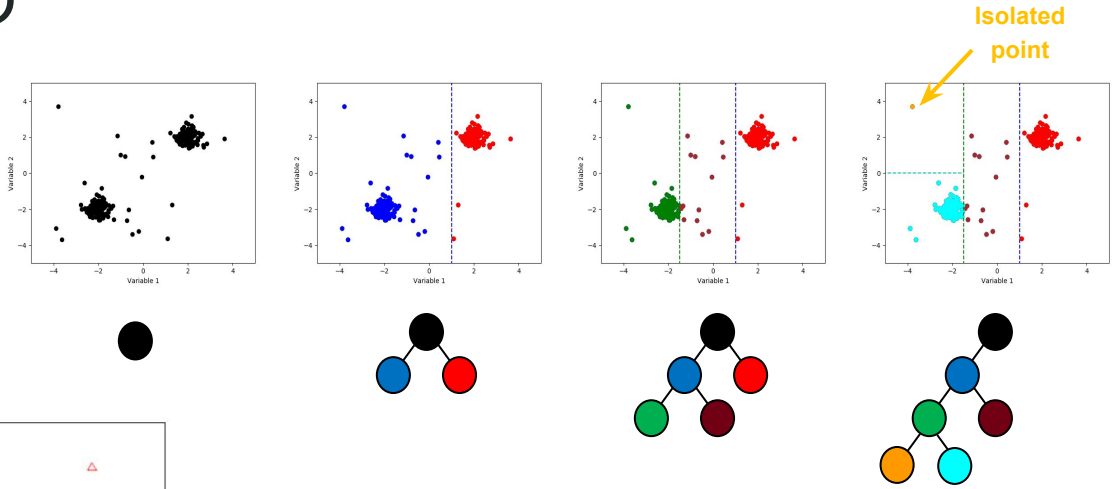
Advantages:
➔ Few hyperparameters to tune
➔ Linear complexity: time does not explode with volume!

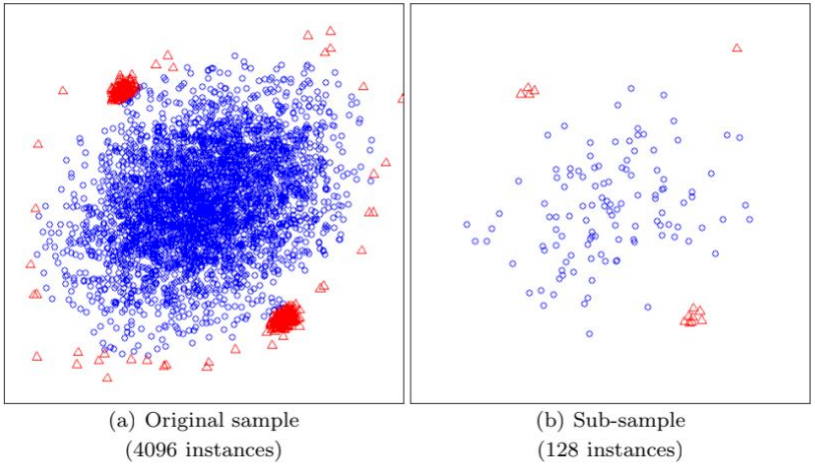
* Anomaly score = average depth normalized with average depth of unsuccessful searches in a binary search tree. **For more details:** *Isolation-based Anomaly Detection,* Fei Tony Liu and Kai Ming Ting

Outlier detection: nD

Other methods - Isolation Forest

Not exactly true... Each tree splits a subset of the data (max 256 points) to avoid **swamping** and **masking**





(a) Original sample (4096 instances)

(b) Sub-sample (128 instances)

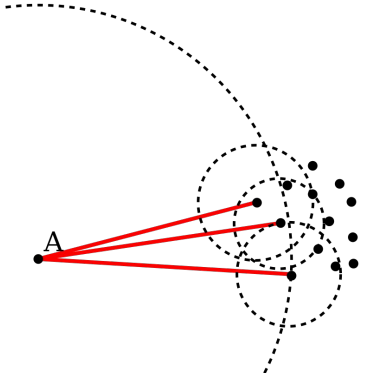
Swamping: predicting normal points as anomalies, because local density is lower

Masking: locally dense anomaly clusters, therefore predicting these anomalies as normal points

Subsampling reduces these 2 effects

Image taken from:
Isolation-based Anomaly Detection, Fei Tony Liu and Kai Ming Ting

Outlier detection: nD



Other methods - Local Outlier Factor (LOF)

➔ 1/ For each point A, the k-distance to all the other points → K-distance(A,B) is the distance of to its k-th neighbour

➔ 2/ Compute the Reachability Distance(LR) of A →

$$\text{reachability-distance}_k(A,B) = \max\{k\text{-distance}(B), \text{d}(A,B)\}$$

➔ 3/ Compute the inverse of the average RD of A to its k-neighbours: Local Reachability Density →

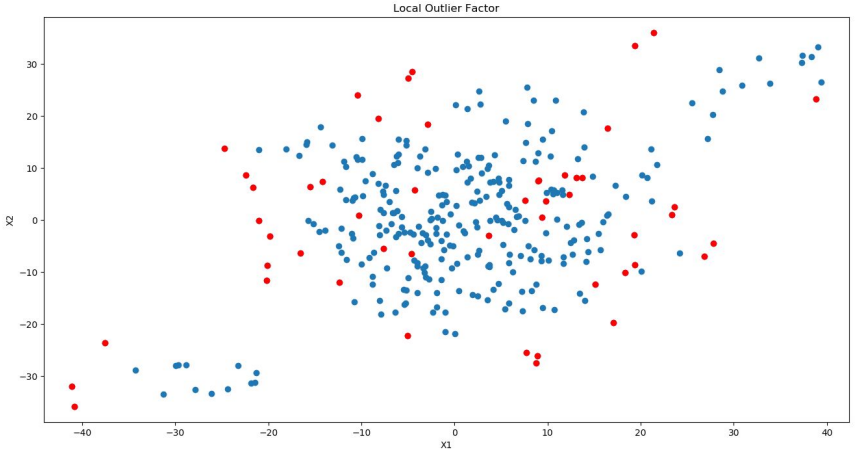
$$LRD_k(A) = \frac{1}{\sum_{Xj \in N_k(A)} \frac{\text{RD}(A,Xj)}{||N_k(A)||}}$$

Low LRD values means that closest cluster of points from A are "far"

➔ 4/ Local Outlier Factor

$$LOF_k(A) = \frac{\sum_{Xj \in N_k(A)} LRD_k(Xj)}{||N_k(A)||} \times \frac{1}{LRD_k(A)}$$

LOF <= 1: similar or higher density than neighbors = low anomaly score
LOF > 1: lower density than neighbors = high anomaly score



Once again, threshold to define!

Advantage:

➔ Locality aspect: points close to very dense cluster can still be anomalies, compared to "border"-based methods

Inconvenient:

➔ Anomaly score (ratio) is hard to interpret

For more details:

LOF: Identifying Density-Based Local Outliers
Markus M. Breunig, Hans-Peter Kriegel, Raymond T. Ng, Jörg Sander
LoOP: local outlier probabilities
H. Kriegel, Peer Kröger, Erich Schubert, Arthur Zimek

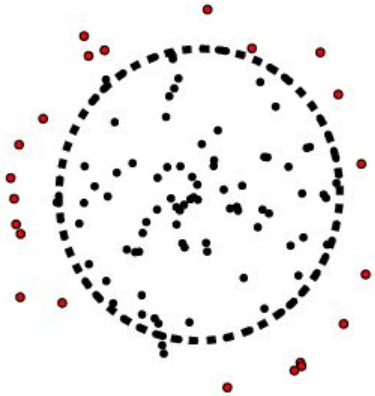
Outlier detection: nD

Other methods - One Class SVM

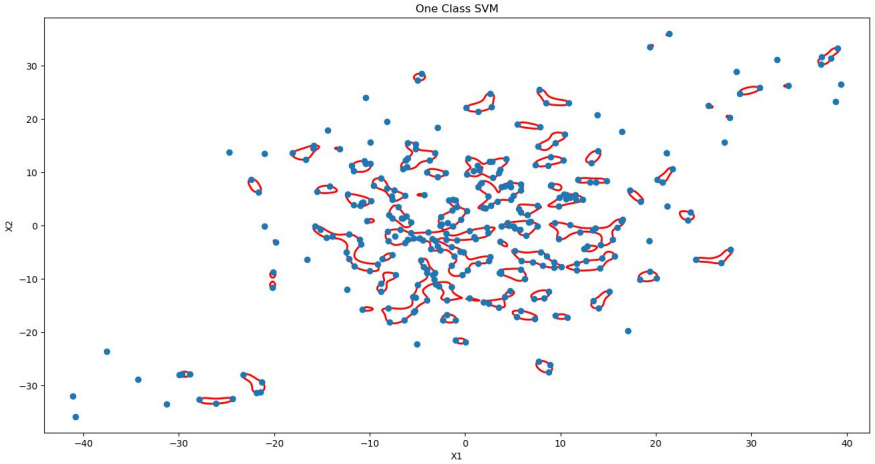
➔ Simple idea: draw a circle around your data points!

➔ You allow a “soft-margin”, tuned with parameter nu (contamination rate), because you have outliers in your dataset

➔ With a kernel: projection of dataset in higher dimension, compute the circle, translate into a non-linear boundary in initial space!



Outside circle: outlier
Inside circle: normal point



Kernel trick used as regular SVM: no need to know the projection, just the dot product...

Once again, threshold to define!

Advantage:
➔ Complex boundary definition

Inconvenient:
➔ Very sensitive to threshold and choice of kernel...
See https://scikit-learn.org/stable/modules/generated/sklearn.svm.OneClassSVM.html

For more details:
Estimating the Support of a High-Dimensional Distribution
Bernhard Scholkopf et al.

Outlier detection: nD

It’s time to play with these methods with sklearn...

Main interest = play with parameters to see the impact on detection boundaries, and explain it through theory

Outlier detection: score VS decision

Be careful!

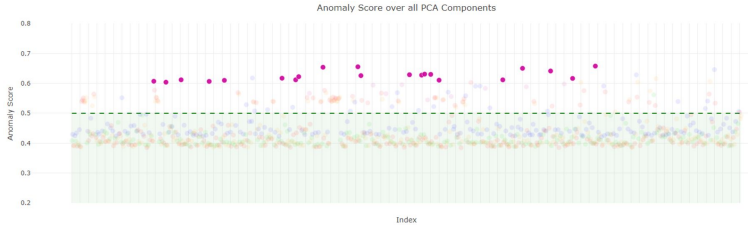
- These unsupervised methods give only a **relative measure of abnormality**
  - Elliptic envelope: mahalanobis distance
  - iForest: average depth
  - LOF: density ratio with neighbors
  - ...

Continuous scores

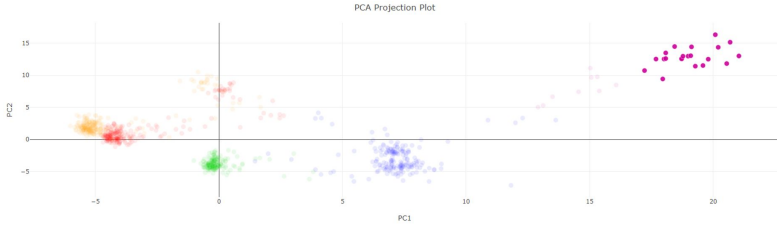
- The **decision itself (outlier or not)** is proposed by default in those methods, but it **always requires threshold tuning**!
  - Always need for human intervention, especially with complex interdependent systems!
  - For example: cross the anomaly scores with manual cluster analysis with PCA, geometrical interpretation...

NOT because technology is not mature enough...

BUT **because the problem is badly formulated!**
"Anomaly" is not clearly defined a priori, and statistics will never tell you what it is!



Human intervention for threshold and / or decision!

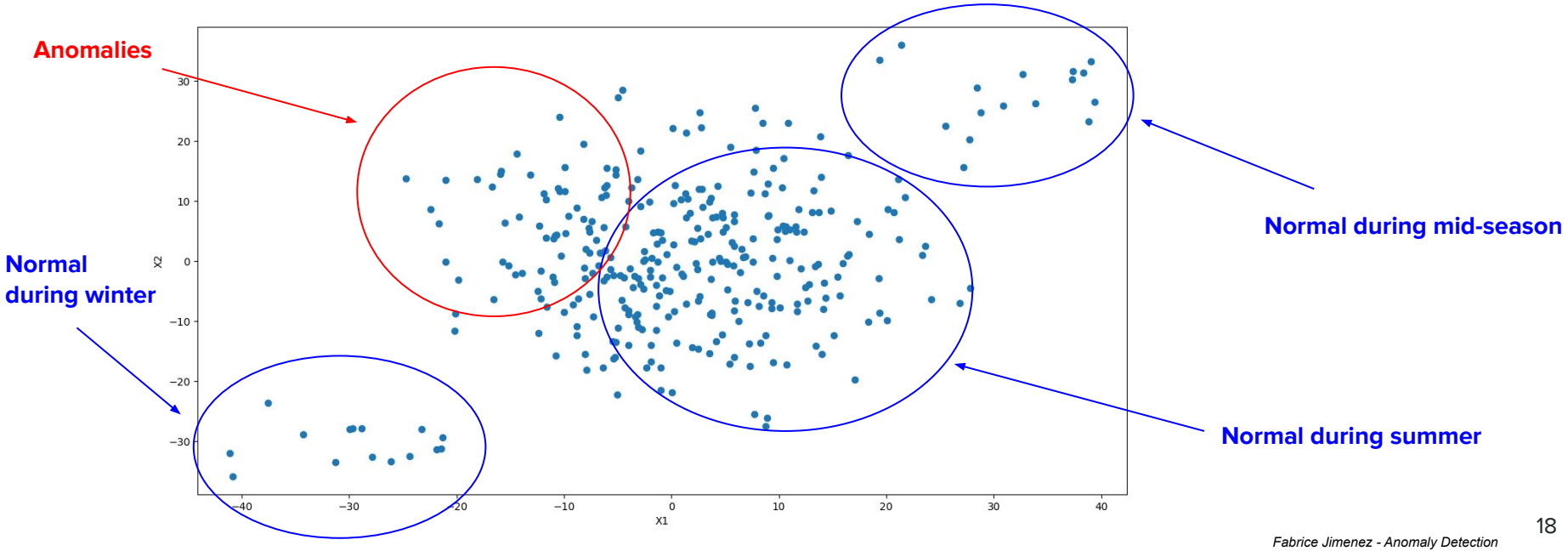


Novelty detection

2/ Anomalies = not so rare, and may not be so different from the majority of the data...

Novelty detection: you have a clean dataset without anomalies (in the sense of 1/ or 2/)

➔ Learn the normal behavior, to be able to check if a new item is normal or an anomaly



Novelty detection

Basic principle

Clean dataset without anomalies: **“learn” the normal behavior.**
Predict the value / score of new points to find out if they match the normal behavior or not

➙ Unsupervised methods we have seen can be used in this case (One Class SVM is even better at this than outlier detection!)

New possibilities

Why not using supervised learning to learn the normal behavior?

v1	v2	v3
8.4	15	2.2
9.1	10	5.1
...	...	...

Model 1: v1 = f(v2,v3)
Model 2: v2 = f(v1,v3)
Model 3: v3 = f(v1,v2)

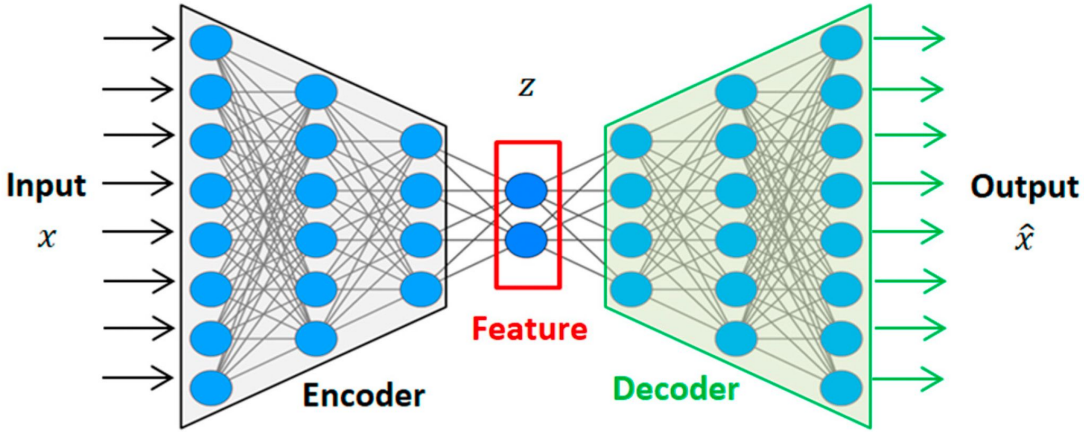
Predict each variable by using the others as features:
➙ Linear regression
➙ Random Forest
➙ SVM...

➙ A new point comes in: (x1,x2,x3)
➙ Compute the predictions [x1] = f(x2,x3), [x2] = f(x1,x3), [x3] = f(x1,x2)
➙ Compute the errors [xi] - xi : squared error, absolute error...

High error = does not fit the “normal” model = high anomaly score

Novelty detection

The rise of deep learning and neural network gives new possibilities in anomaly detection
Example of AutoEncoders

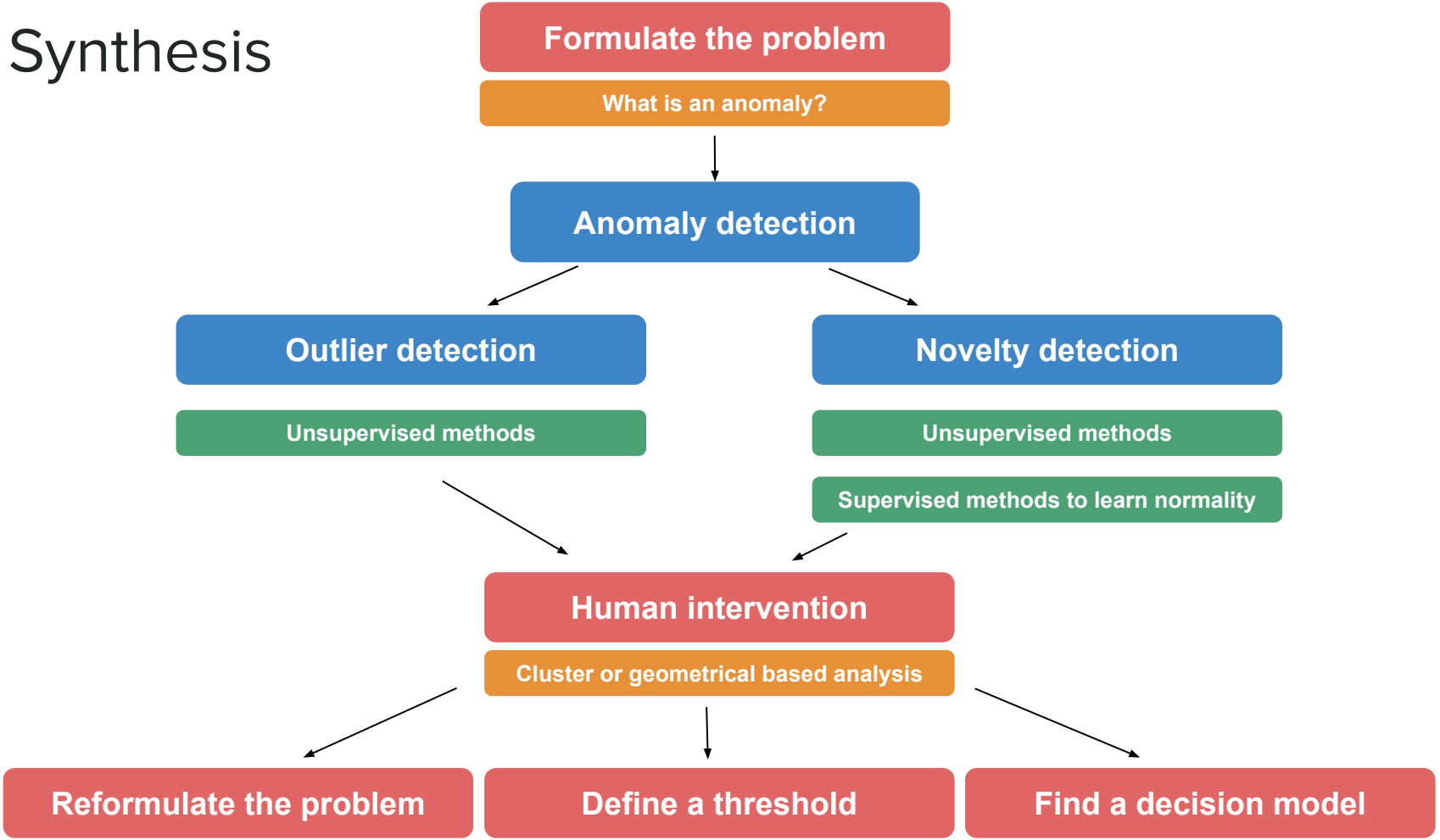


➡ **Use error of reconstruction as a score of Anomaly**
➡ **Architecture choice, loss is problem dependant and requires lots of iterations**

Going further:

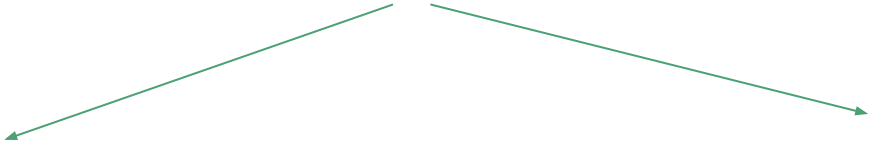
- *Variational autoencoder https://github.com/Michedev/VAE_anomaly_detection*

Synthesis


Formulate the problem
What is an anomaly?
Anomaly detection
Outlier detection
Novelty detection
Unsupervised methods
Unsupervised methods
Supervised methods to learn normality
Human intervention
Cluster or geometrical based analysis
Reformulate the problem
Define a threshold
Find a decision model

What’s next ?

Not this end of the story … Monitoring the performance of the deployed algorithm is key



Data drift monitoring

Is my input data still have the same characteristics? Sensors issues ?

Concept drift

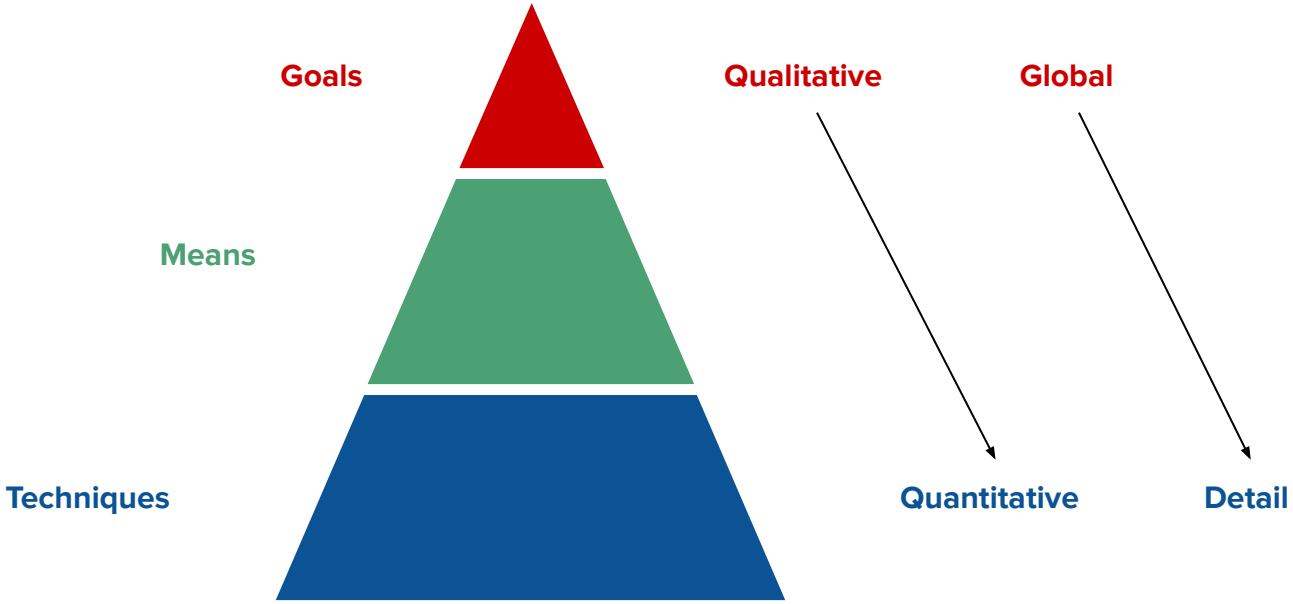
Is my understanding of the anomaly still relevant ? Explainability ?

➔ ***Data collection, retraining strategy …***

Quick piece of advice…

Machine Learning = complex field → a lot of: models, ideas, approaches, theories… every day!

How to keep up the rhythm? → Build your own understanding, <u>from global to detail</u>



Example:

Based on historical data, detect when behavior is changing

Novelty detection: learn normal past behavior, use prediction error as anomaly score

Random Forest regression to predict each feature in function of others, use mean squared error

Questions?