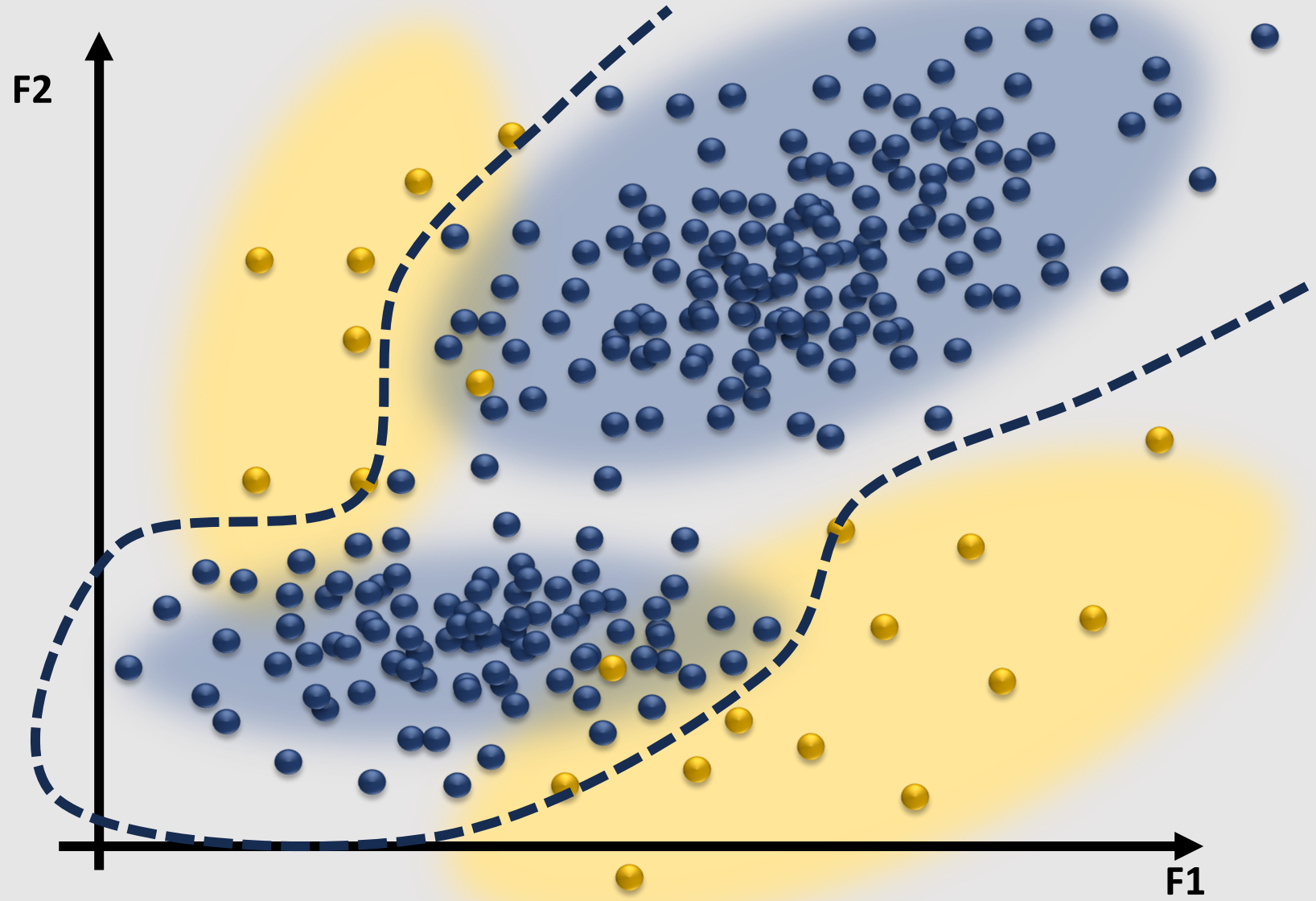
Mechatronic Engineering program:
Python for machine learning and data science

Data challenges

Ziemowit Dworakowski

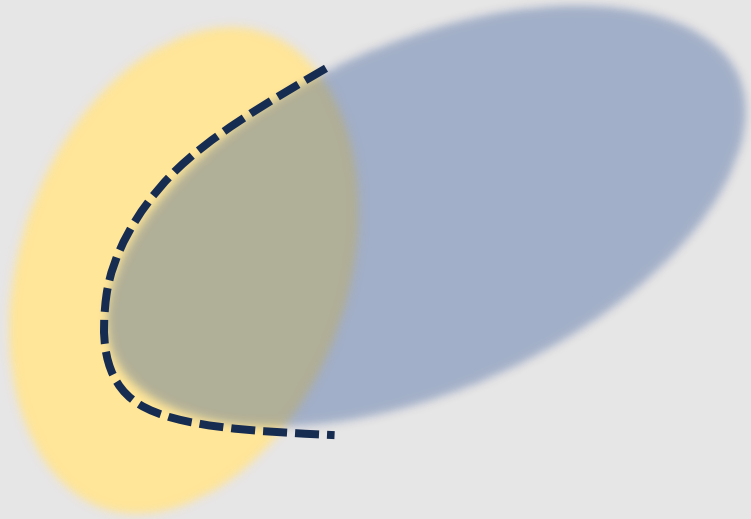
AGH University of Krakow

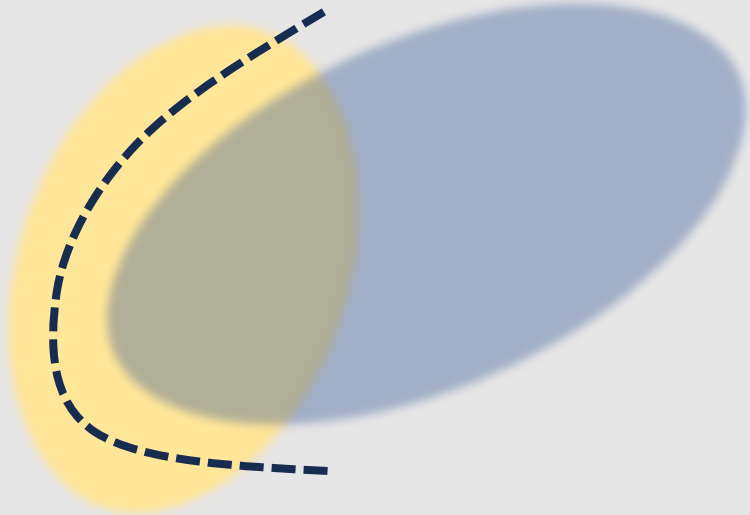
What to do if classess are under-represented?



What to do if classess are under-represented?

If we just consider overall error, the training algorithm will mostly ignore under-represented class wherever there is contest





Often, the classifier might even artificially extend the area of the over-represented class

However, we are rarely interested in raw average accuracy (which would be minimized in such a scenario). Usually, in-class accuracy can be just as important

What to do if classess are under-represented?

Use classifier that ignores class density – e.g. SVM

Assign weights to data, low-density classes get more important

Augment dataset: Generate artificial data for low-density classes



This is often used also if ALL the classes are under-represented (if we just don't have enough data)

If we want to do data augmentation, we do it AFTER division of data into training/testing datasets – so data for testing would not be taken from augmented training samples!

Data augmentation

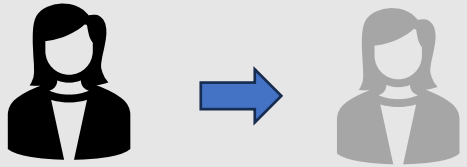
Random noise addition / regeneration of data from distribution

Data transformations

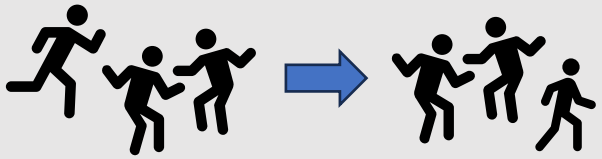
Flipping, rotating, etc.



Global transformations



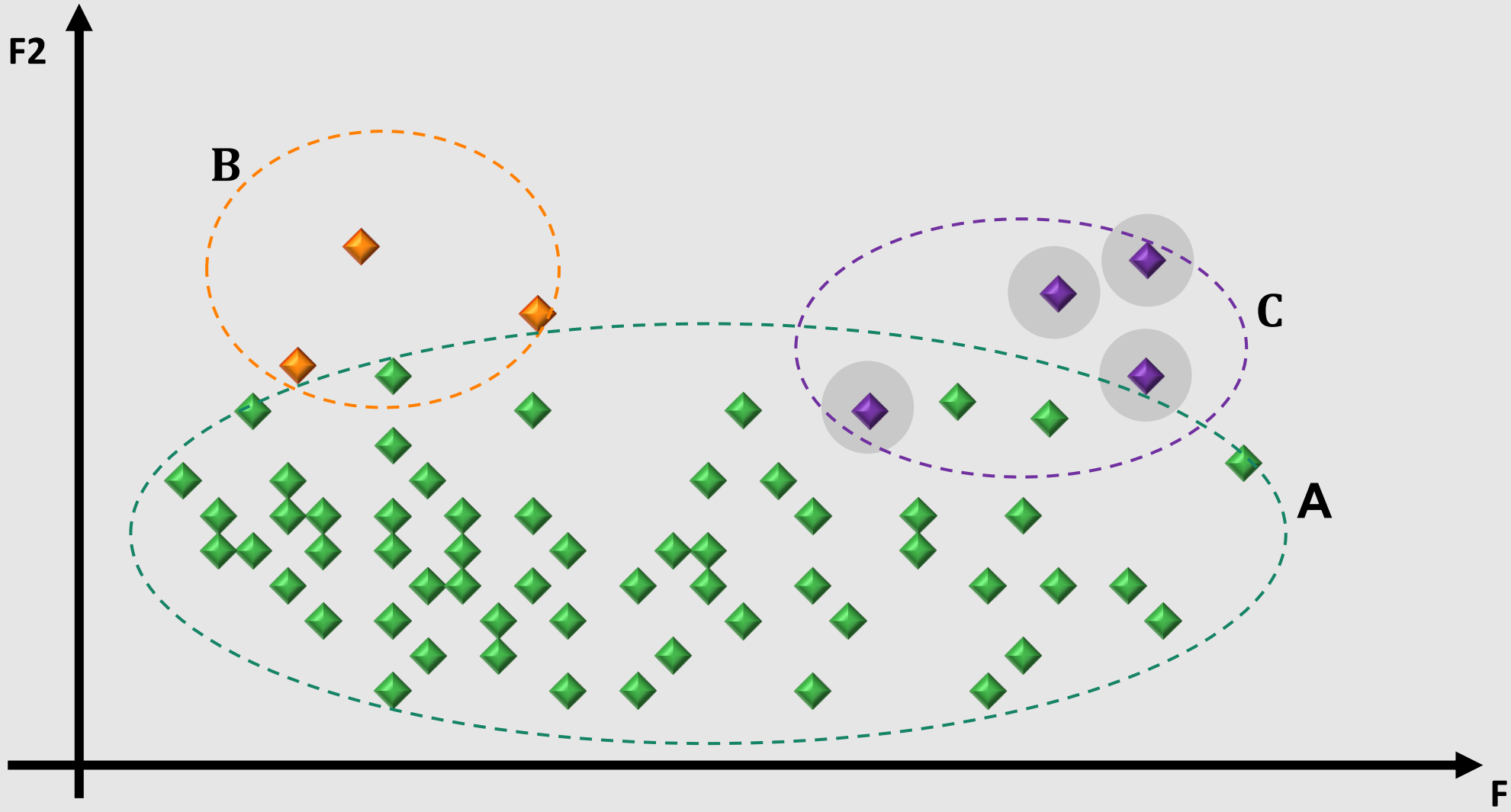
Rearrange components of different samples



Works great for images.
A bit tricky for other data

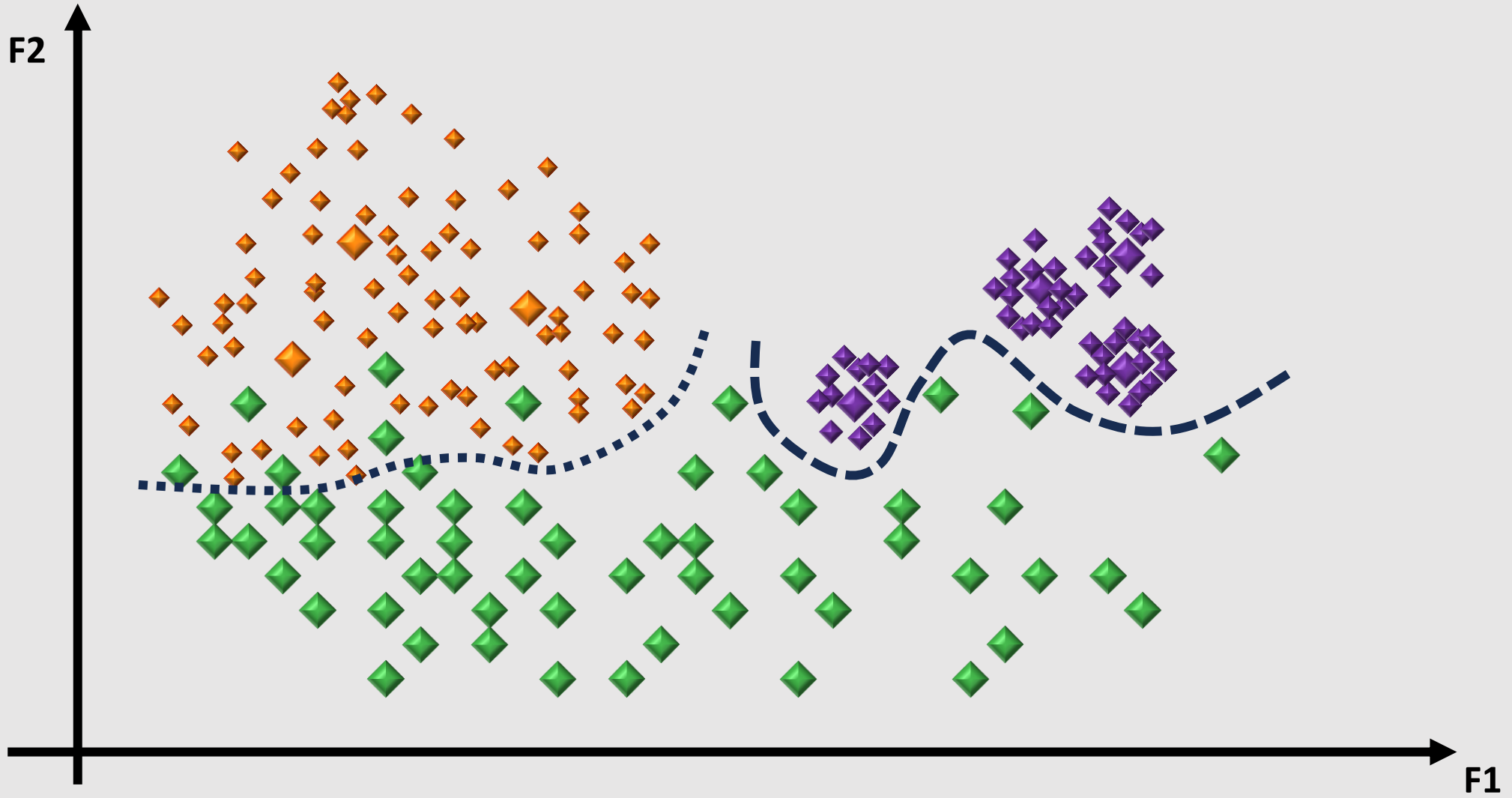
Random noise addition / regeneration of data from distribution

1) Generate new data around old data (by adding random noise)



Random noise addition / regeneration of data from distribution

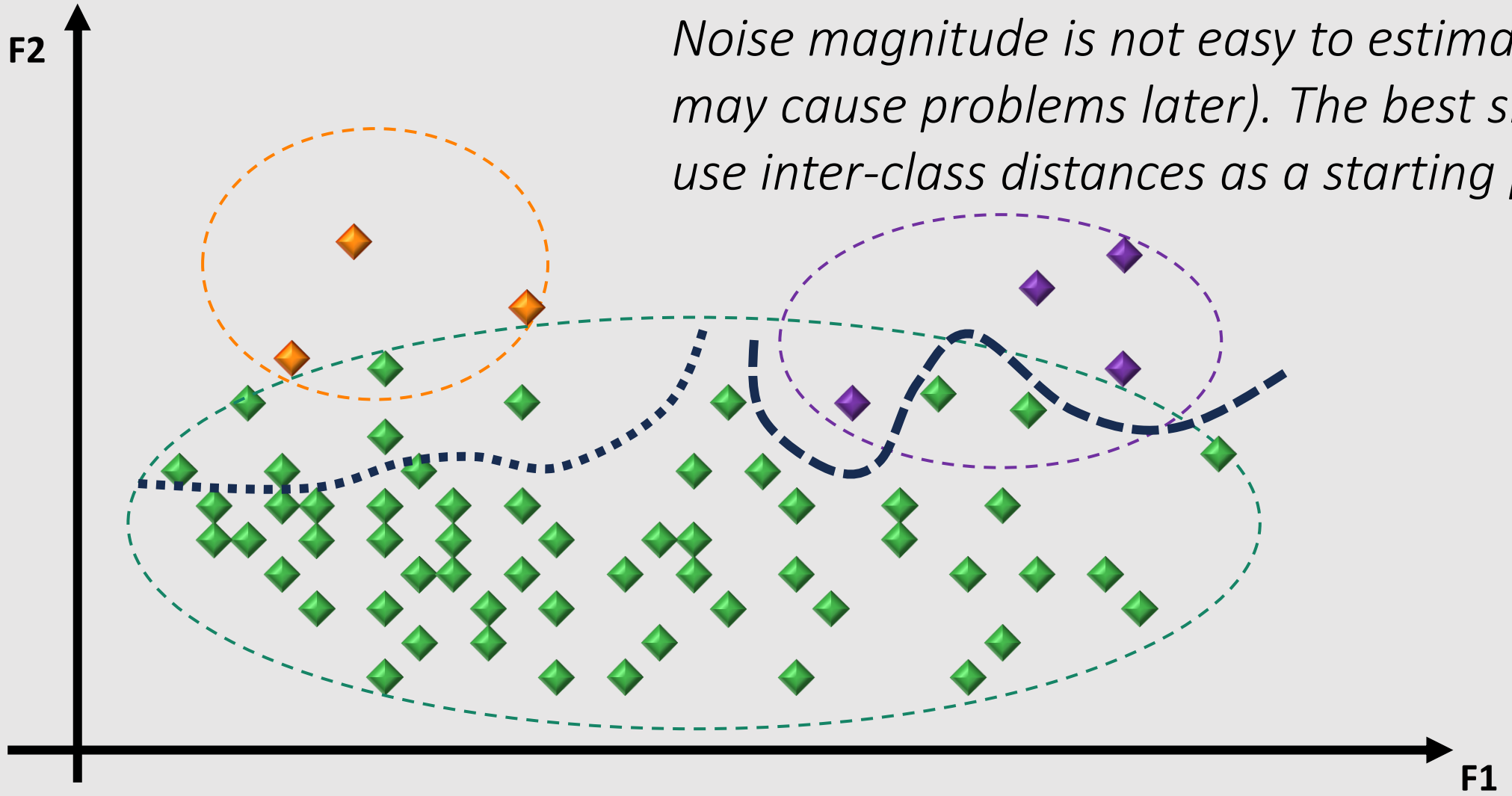
1) Generate new data around old data (by adding random noise)



Random noise addition / regeneration of data from distribution

1) Generate new data around old data (by adding random noise)

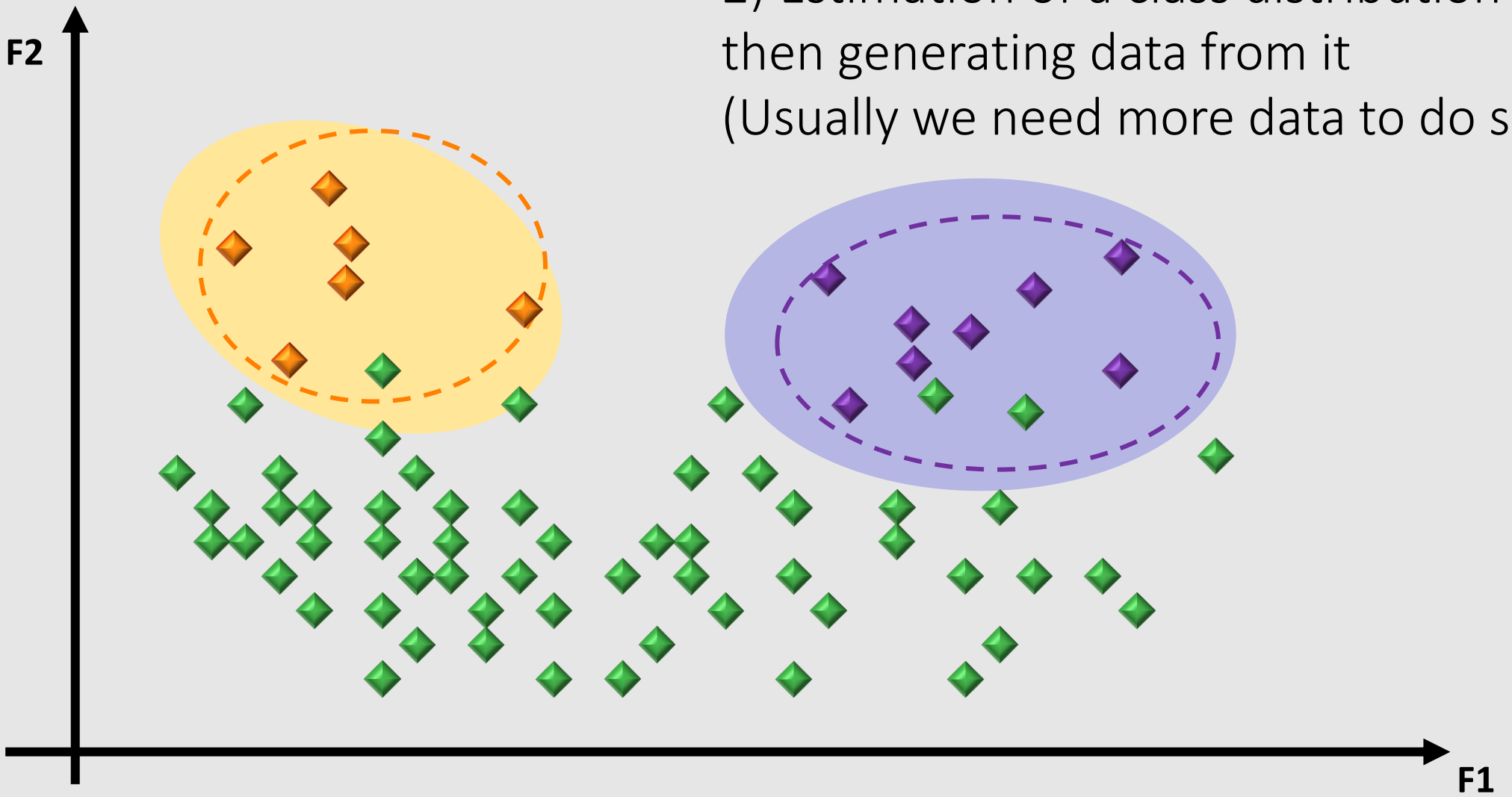
Noise magnitude is not easy to estimate (and may cause problems later). The best shot is to use inter-class distances as a starting point



Random noise addition / regeneration of data from distribution

1) Generate new data around old data (by adding random noise)

2) Estimation of a class distribution – and then generating data from it
(Usually we need more data to do so…)



Random noise addition / regeneration of data from distribution

1) Generate new data around old data (by adding random noise)

2) Estimation of a class distribution – and then generating data from it
(Usually we need more data to do so…)



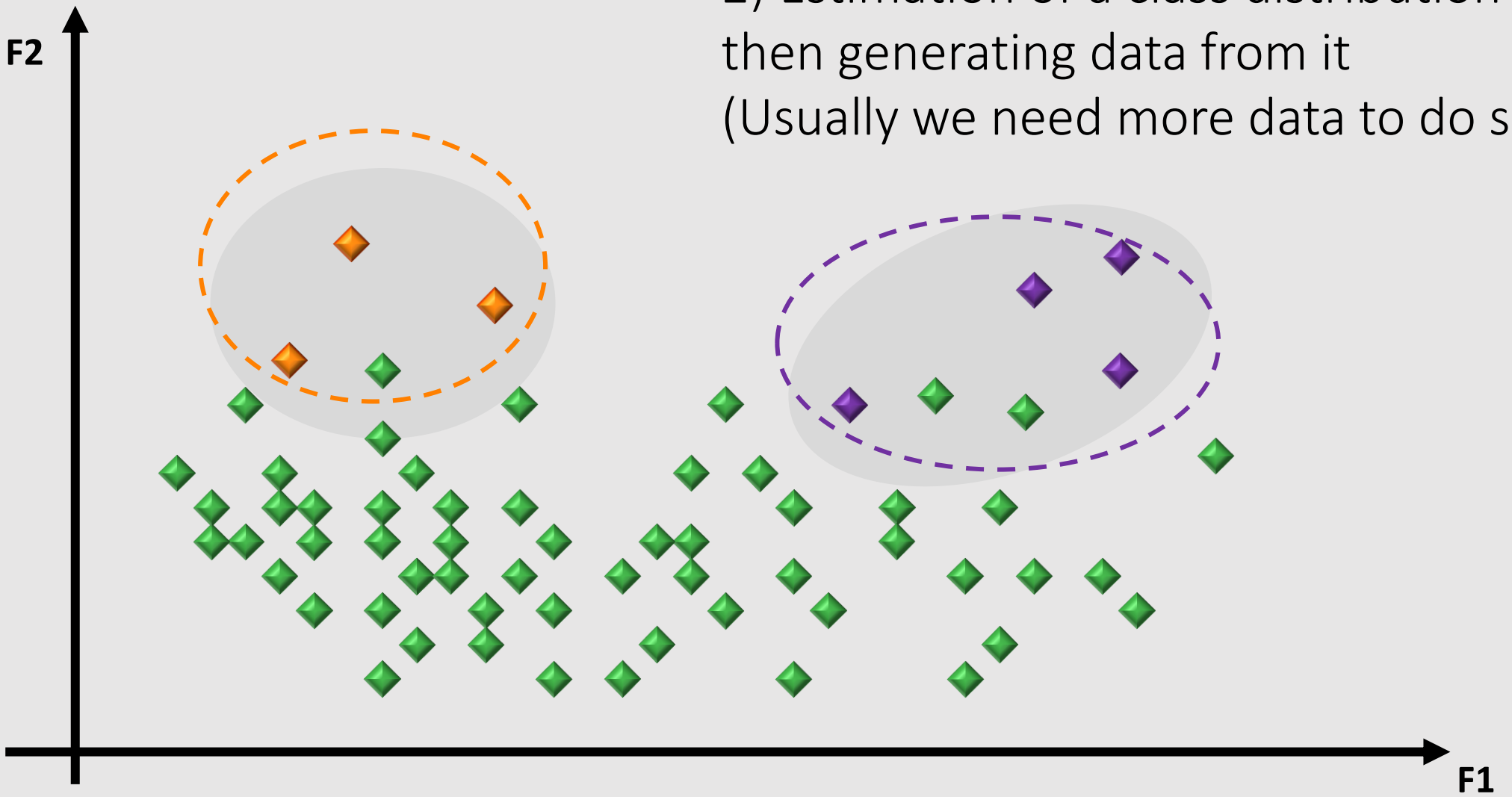
If you have enough data to correctly derive its distribution – you usually have also enough data to do classification…

So it works mostly if you know additional context (what distribution can be assumed a priori)

Random noise addition / regeneration of data from distribution

1) Generate new data around old data (by adding random noise)

2) Estimation of a class distribution – and then generating data from it
(Usually we need more data to do so…)



What if we have entire class missing?

One-class classification (novelty detection)



One-class classification (novelty detection)



One-class classification (novelty detection)

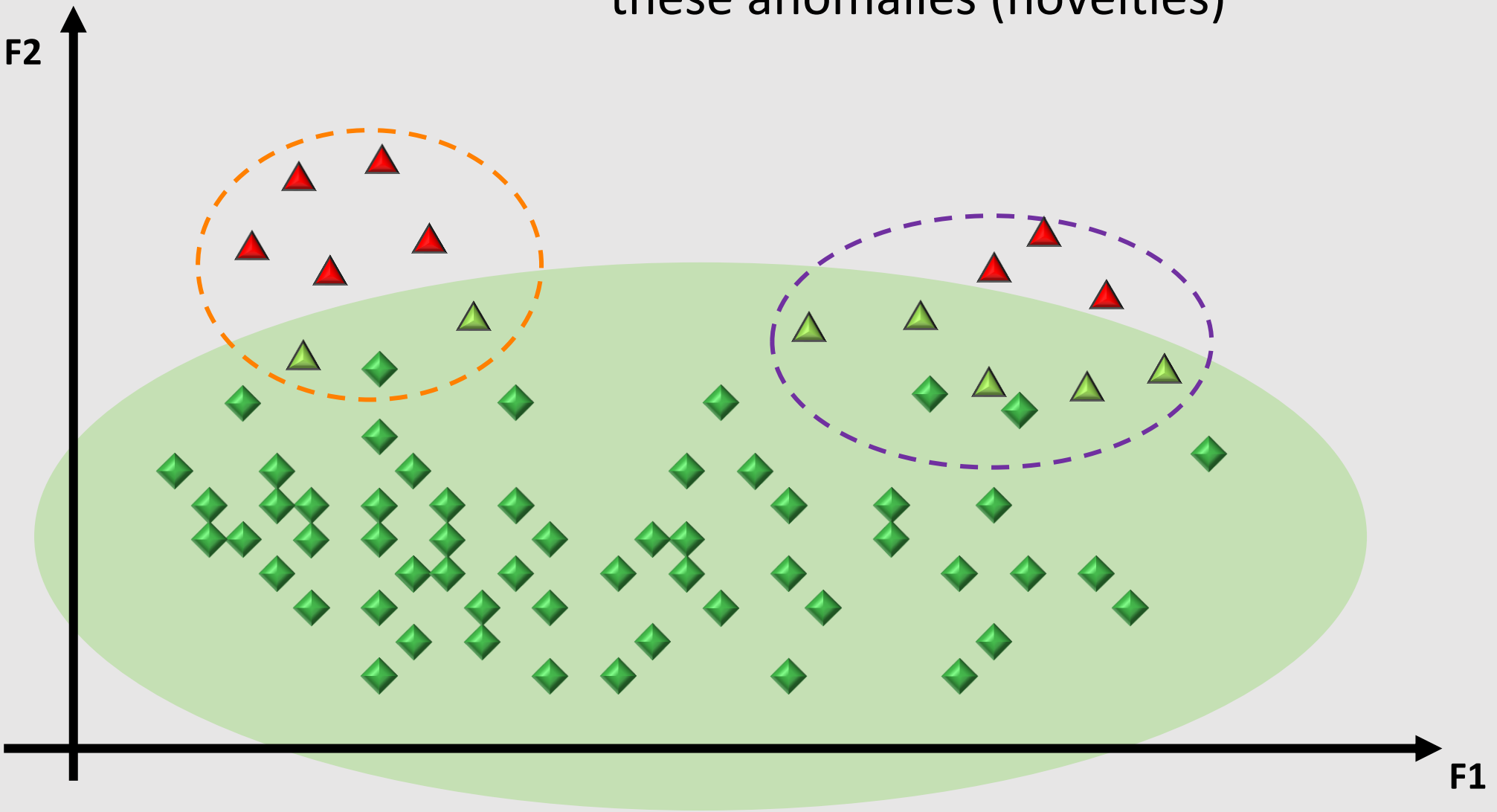
Sometimes we have a dataset with lots of examples belonging to just one class and none or almost none to the other classes:

- „**Healthy** people”
- „**Normal** behavior in subway”
- „**Normal** operation state of an assembly line
- „**Typical** weather conditions in September”

Then, our goal might be to learn this **normal** range of data, to detect any **anomalies (outliers, novelties)**

In Novelty Detection we usually don’t know what to expect (there is possibly Infinite set of „norm breaches”)

1. Learn the normal range
2. Detect what breaches normal range
3. If you have other class data, use it to classify only these anomalies (novelties)



1. Learn the normal range
2. Detect what breaches normal range
3. If you have other class data, use it to classify only these anomalies (novelties)



One-class classification (novelty detection)

How to learn „normal range”?

1. Estimate typical distances between samples within training set, detect anything that breaches it

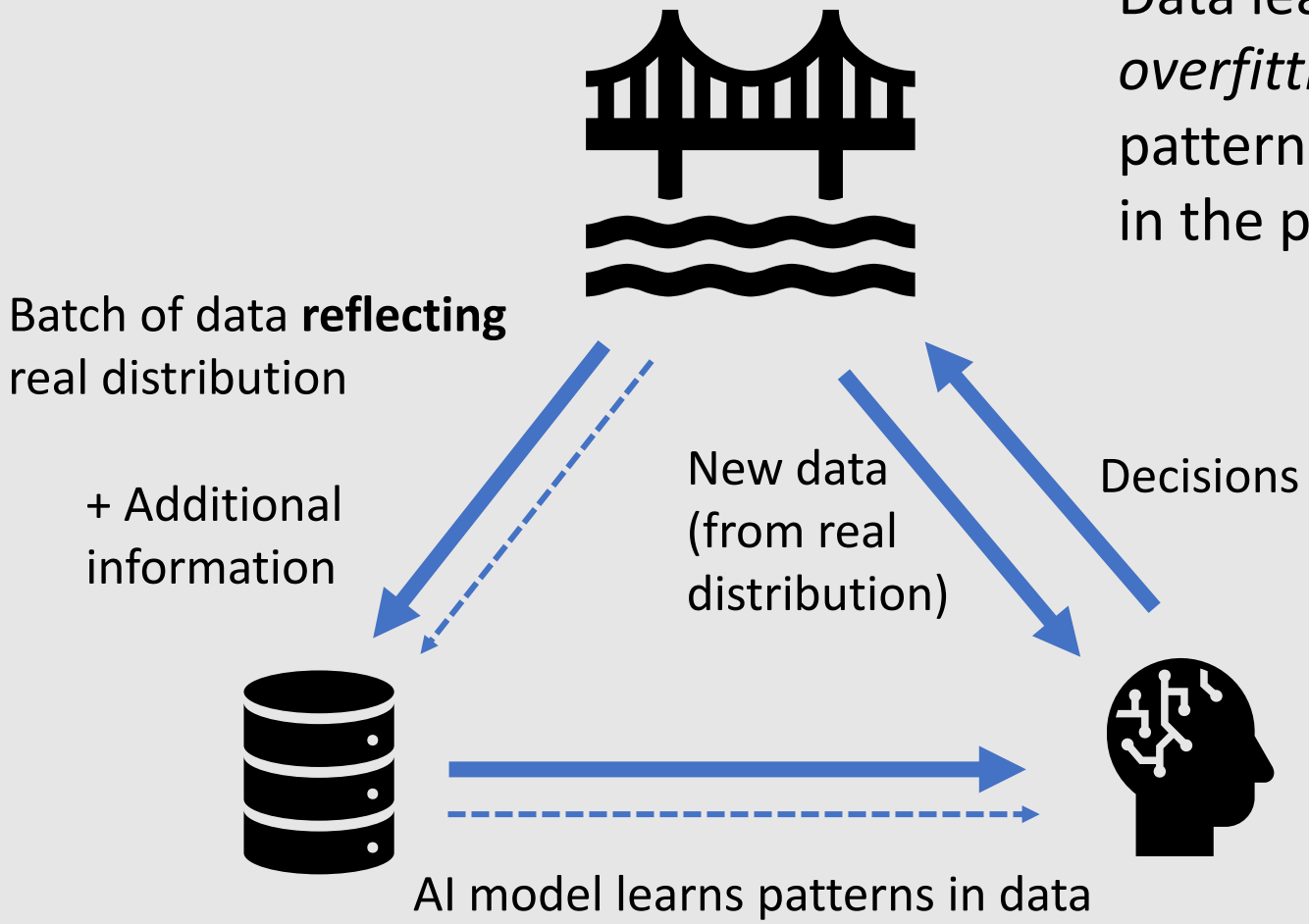
2. Estimate underlying probability density for data – e.g. by fitting multidimensional gaussians into training data

3. Train a regression system to derive some of the features from others – in normal range (for known data) it should work very good, for novel samples it should produce large errors

Data leakage

Data leakage occurs, when the decision system (*e.g. a classifier*) uses information during training and initial testing that will not be present in-operation.

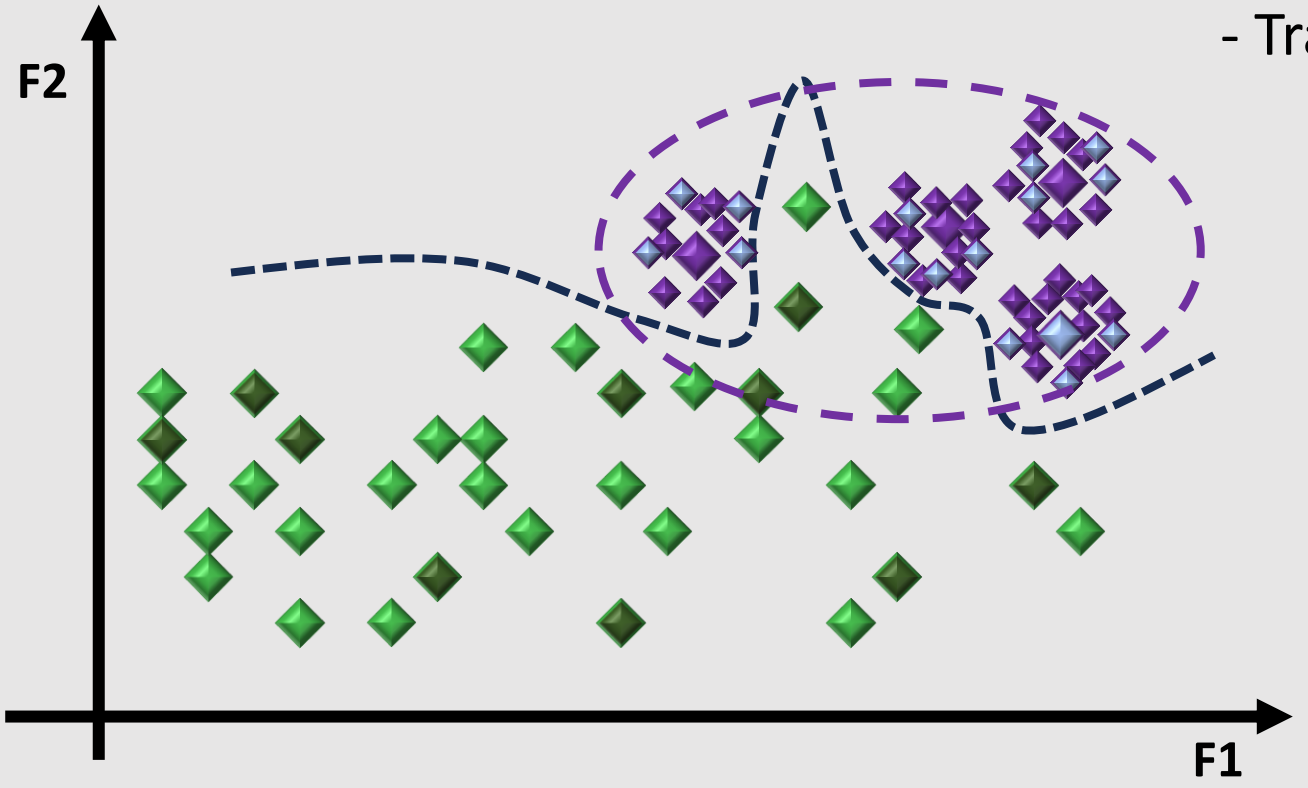
Data leakage can be viewed also as a „*hidden overfitting*" phenomenon: A model learns patterns that are not general but present only in the particular setup of training data.



Data leakage

Data leakage can happen due to:

Training example leakage (row-wise leakage) – training samples influencing directly testing samples (poor augmentation or data acquisition)



- Train/test split after augmentation of data

- Training data

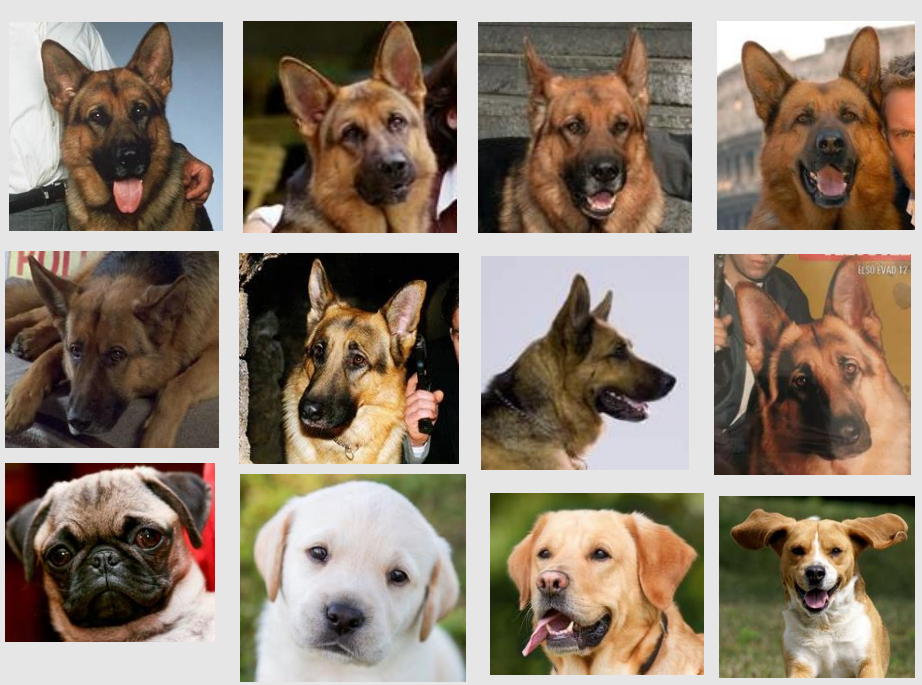
- Testing data

Data leakage

Data leakage can happen due to:

Training example leakage (row-wise leakage) – training samples influencing directly testing samples (poor augmentation or data acquisition)

- Train/test split after augmentation of data
- Samples being „copies” of each other



These are all pictures of the same dog!

Even if we add different dogs to the dataset, pictures of the same one should not be split into training and testing at the same time!

Data leakage

Data leakage can happen due to:

Training example leakage (row-wise leakage) – training samples influencing directly testing samples (poor augmentation or data acquisition)

Feature leakage (column-wise leakage) – inclusion of columns directly related to targets



Data leakage

Data leakage can happen due to:

Training example leakage (row-wise leakage) – training samples influencing directly testing samples (poor augmentation or data acquisition)

Feature leakage (column-wise leakage) – inclusion of columns directly related to targets

Data acquisition procedures that differ for training data and in-operation data

Cat images were acquired indoors



Dog images were acquired outdoors



A classifier recognizing cats from dogs

Is required to recognized new data in all surroundings





Data leakage

Data leakage can happen due to:

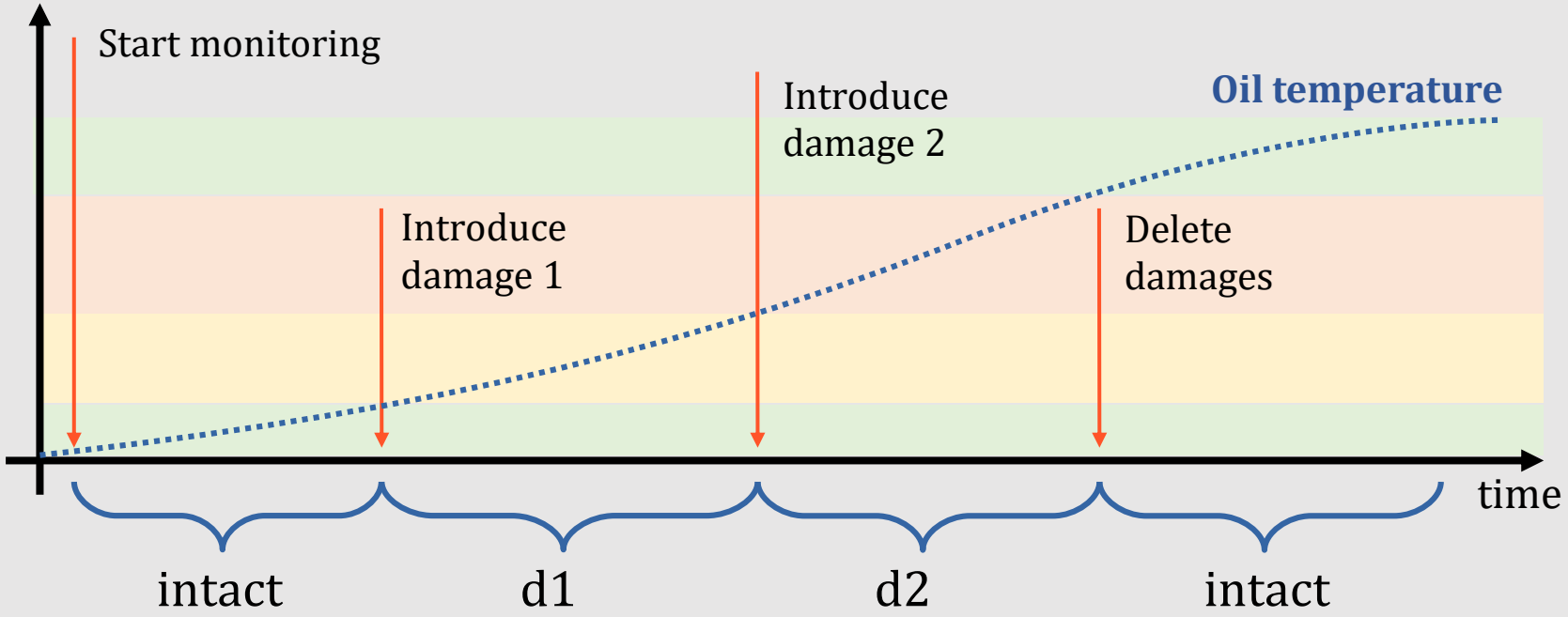
Training example leakage (row-wise leakage) – training samples influencing directly testing samples (poor augmentation or data acquisition)

Feature leakage (column-wise leakage) – inclusion of columns directly related to targets

Data acquisition procedures that differ for training data and in-operation data

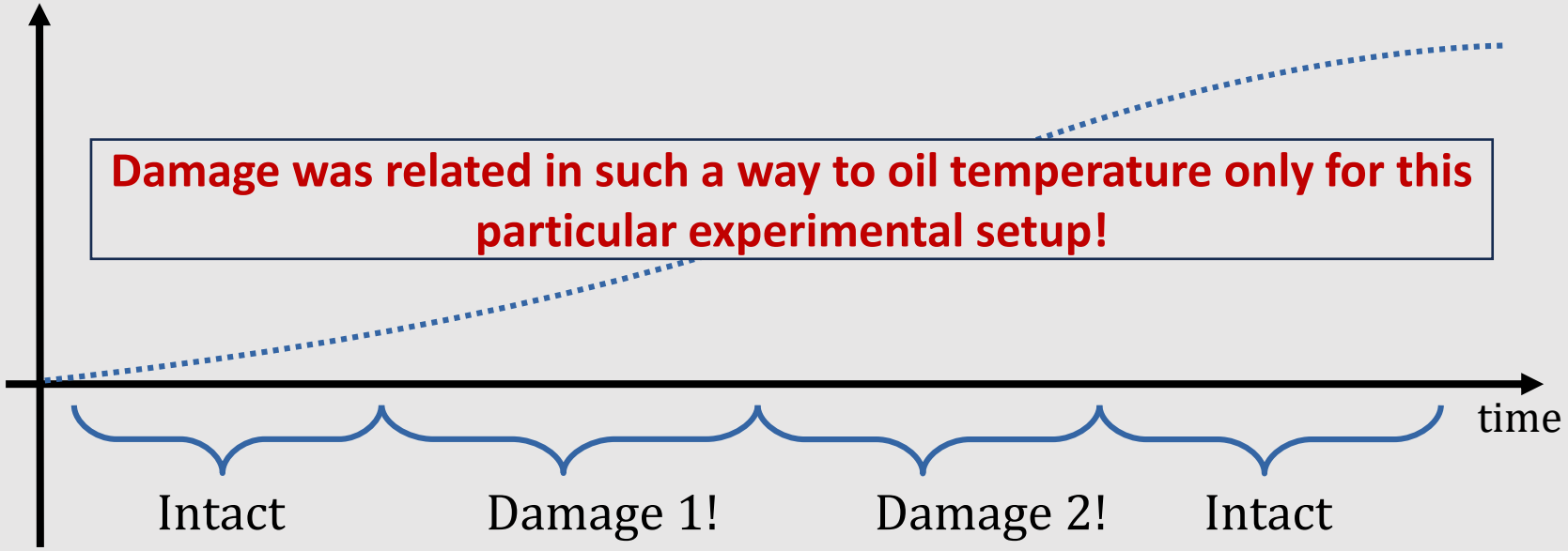
Lack of independence of samples – e.g. by indirect inclusion of time-related information

An example: A fan monitoring software



A lot of features are acquired: vibro features, **oil quality and temperature**, bearings temperatures, electrical features…

The experiment was repeated in this configuration several times…



Then a standard random division into training/validation/testing subsets. Classifier obtains very high accuracy…

Then a CEO is invited for a demonstration…

Data diversity

So how to check if our dataset may contain/cause data leakage?

Unfortunately, there are no straightforward methods for that… ☹
But we can:

Check features and think if they can be measured with no prior knowledge of targets (if no – possible leakage)

Think about independence of data samples. If they are dependant, testing set should consider a separately acquired batch of data (random split should not be used)

Check data distribution and think if it is close to one expected in-operation

Consider order of samples' acquisiton. If they were acquired in particular order, consider using the newest samples for testing purpose only

Check for experimental conditions and think if they look plausible (if all the expected in-operation conditions are covered)

Check features and think if some of them would require exactly similar knowledge as targets (e.g. month salary vs year salary)

If there are different procedures required for acquisition of various classes data, double-check if they influence data features by themselves

Wherever possible, refrain from random train/test splits in favor of informed splits ensuring independence of samples (mimicking possible future experiment setting)

Things to remember:

1. Risk of having under-represented classes and three solutions to this problem
2. When can we do data augmentation (in relation to division of data into subsets)?
3. How can we augment data? Explain two methods to this end
4. What is a novelty detection problem? Explain three approaches to solve it
5. Data leakage: explain the problem, four main sources for it, provide at least two different practical examples