# Human Factors (HF) in Artificial Intelligence (AI)

Which role for machine learning (ML)?

# From AI to HF applications

# Artificial intelligence aims at automatic…

- …representation of knowledge and reasoning

"Given the history of patients how can I determine if a patient is at risk of a disease ?"

- …planning and decision making

"Given a state of the world which actions should a robot perform to reach its goals ?"

- …perception

"Can I detect an given object in an image ?"

- …learning of new knowledge and rules

"What features of an image defines what is a cat ?"

# Machine learning (ML)

ML is learning from observations / data (to perceive, plan, take decisions, etc.)

Three major types of machine learning:

- Supervised machine learning (data and some feedback about success and error)
- Unsupervised machine learning (data but no feedback)
- Reinforcement learning (discovery of data and some feedback about success and error)

# Why using AI and machine learning in HF ?

Cope with human limitations (e.g. perception, repetitive tasks, etc.)

- Perception example: some sensors can give information that our senses do not provide

Autonomous machine also mean machines which can behave WITH humans and IN a human world

- Machine must understand and display social cues (entropomorphism ?, bonding ?)

HF aims at a better understanding of human in its interaction with a (already) complex system.

- The interaction is complex, AI helps to cope with this complexity

Ideally, most of HF work is done before the development of systems but AI allows to monitor humans in real-time.

# Brain storming (5 groups of 3)

- Pick a human “limitation” seen in the previous lecture

Workload

Cognitive bias

Conflict

Stress

Memory

Lack of systematicity

Disability

Lack of power (Health)

- Propose how AI could solve such a limitation by focusing on:
  - Which states / events / behaviors you would like to detect
  - Which input data you could use to detect it
  - How the AI output can change the interaction / solve the issue

# The AI / HF loop

With the development of AI there is much more interaction between humans and machines

- proposition of decisions from AI
- humans information is used by AI to learn

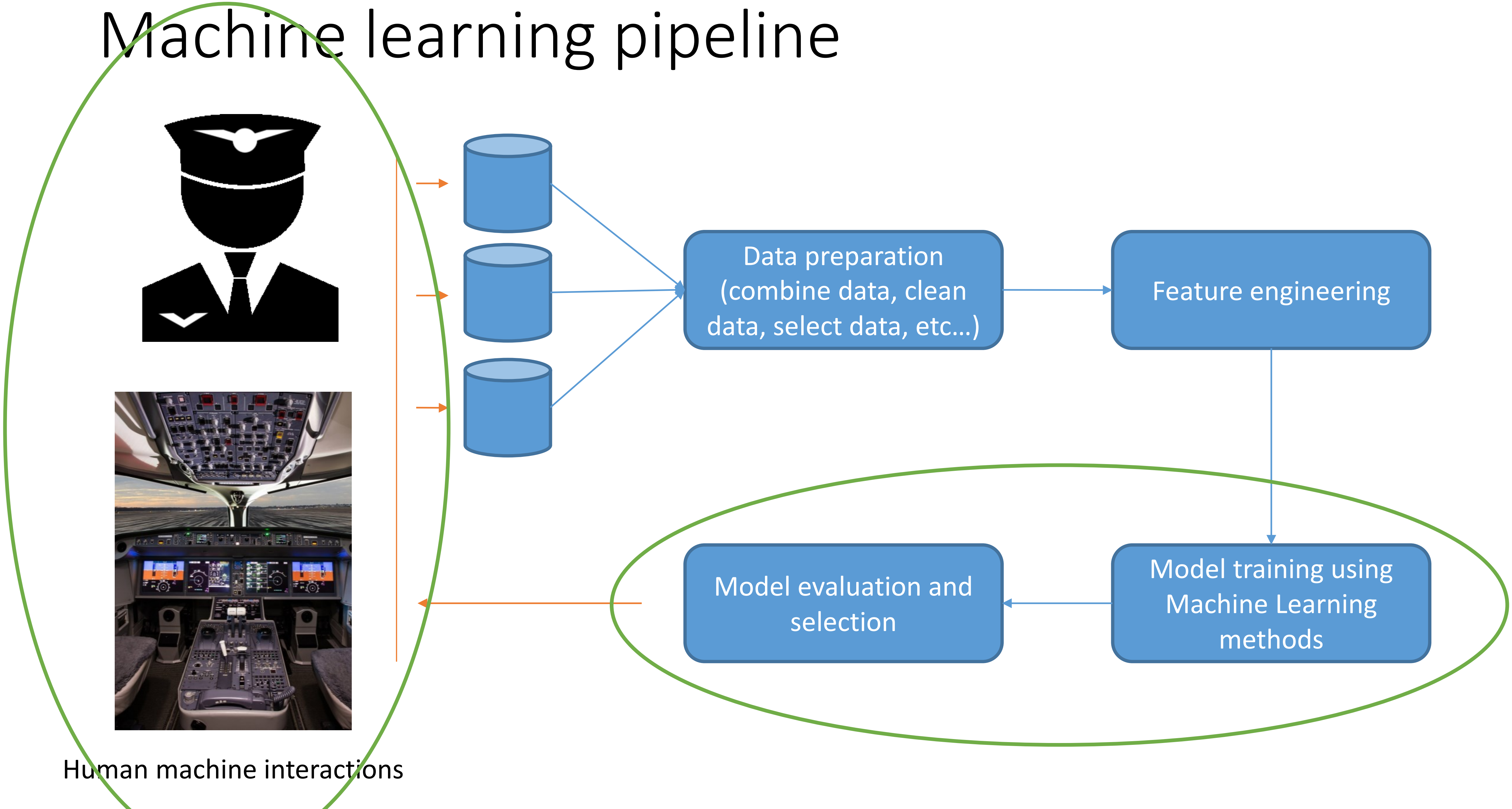
→ augmentation of the complexity of the interaction

Because of this increase of interaction machines need to take into account human factors such as **moods, emotions, satisfaction (usability), fatigue, cognitive load, perception bias, etc.**

How to detect/predict that ? Use AI as well !!!

- collecting data related to the concepts above
- training on this data

# Machine learning pipeline



Data preparation (combine data, clean data, select data, etc…)

Feature engineering

Model training using Machine Learning methods

Model evaluation and selection

Human machine interactions

# Experimental design for data acquisition and testing

# Conceptualization

Concepts are created from conceptions, the process by which we use a term to refer to some common observations

e.g. “after doing some sport many feel their muscles are sour and difficult to control“ -> the concept of fatigue.

Conceptualization is the process which aims at specifying the meaning of a concept:

- This is done by finding indicators of a concept, the presence or absence of those indicators specifies the concept;
- Concepts can be decomposed in dimensions (in which case indicators are given for each dimension)

# Conceptualization – Fatigue example

Fatigue is defined as tiredness resulting from motor or cognitive efforts (at least 2 dimensions).

We will focus on visual fatigue. When watching a screen for a long time people often report the following markers of visual fatigue:

- Eye-strain
- Focusing difficulty
- Headache
- Nausea

# Operationalization

Defines how the previously selected indicators will be measured.

| Indicator | Operationalization |
|---|---|
| Eye-strain | • Questionnaire items on likert scales (my eyes are heavy, my eyes are dry)<br>• Measure number of eye-blinks using an eye-tracker |
| Nausea | Measure salivation |
| Focusing difficulty | Measure pupil size |
| Headache | • Questionnaire (I have pain in the temples, I have pain in the forehead,…)<br>• Measure brain activity |

# Questionnaires

Pre-test questionnaires:

- Identify potential covariates and confounding variables (see after)
- Can be used to determine the evolution of the concept

Reporting during the test:

- short reports to have a regular measure of the concept evolution
- reports on stimuli basis (i.e. for each given event)

Post-test questionnaires:

- Used to determine the evolution of a concept before/after experiment

# Variables in an acquisition protocol

- Manipulated variables (independent variables)
  - What you manipulate (i.e. the cause)
  - Strong ties to how you will stimulate participants
  - Very often decomposed in several conditions, including a control condition.
  - In term of ML this is very often the target you want to predict

- Measured variables (dependent variables)
  - What you measure (i.e. the effect)
  - Strong ties to how the concept is operationalized
  - In term of ML this is very often the features, but can also be the target (e.g. questionnaire answers about fatigue)

# Acquisition of data for screen visual fatigue

- Manipulated variables

  Participants will be distributed in three groups:

  - The control group will play a board game for 1h;
  - The VF1 group will play the equivalent video game for 1h;
  - The VF2 group will play a video game for 2h.

  All participants will be recorded while watching a movie right after the preparation phase.

- Measured variables
  - Eye-movements and pupil size
  - Salivation every 5 minutes
  - Answer questionnaires every 5 minutes

ML to predict 3 classes

# Acquisition of data for screen visual fatigue

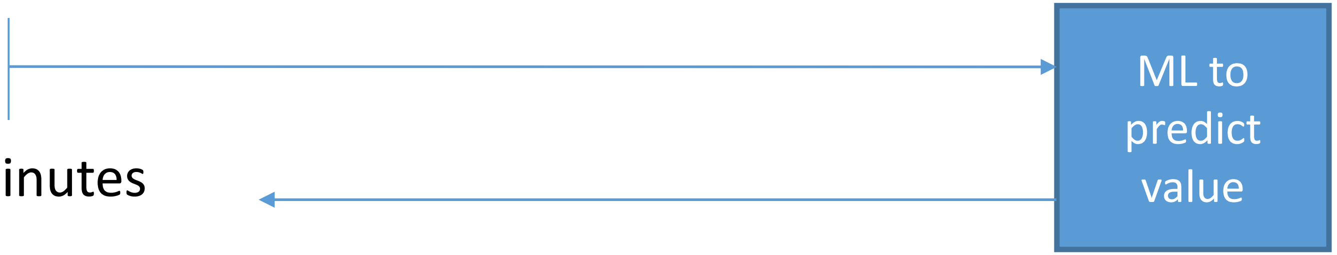
- Manipulated variables

  Participants will be distributed in three groups:

  - The control group will play a board game for 1h;
  - The VF1 group will play the equivalent video game for 1h;
  - The VF2 group will play a video game for 2h.

  All participants will be recorded while watching a movie right after the preparation phase.

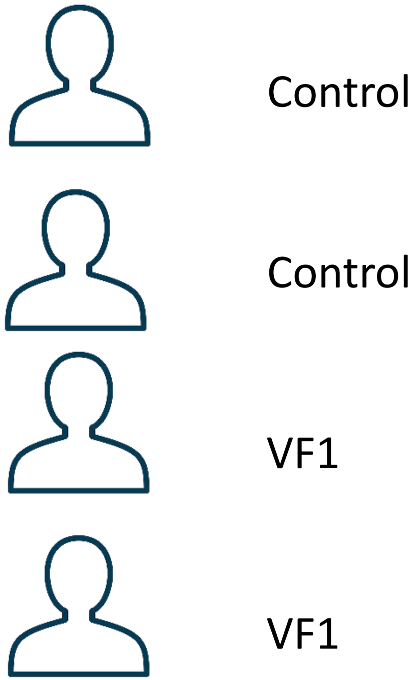
- Measured variables
  - Eye-movements and pupil size
  - Salivation every 5 minutes
  - Answer questionnaires every 5 minutes



# Between vs. within participant designs

The values of an independent variable (conditions) can be distributed with:

One value per participant



Control

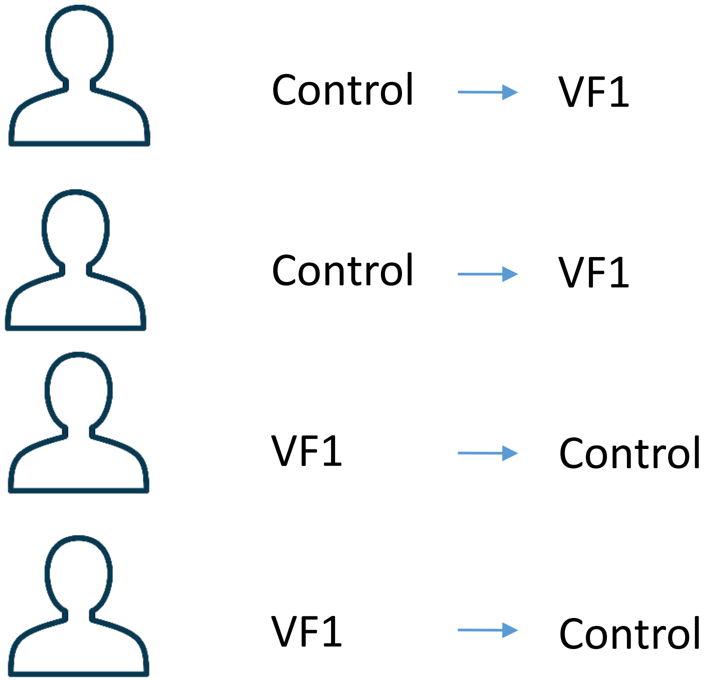
Control

VF1

VF1

Between participant design

All values per participant



Control → VF1

Control → VF1

VF1 → Control

VF1 → Control

Within participant design

# Within vs. between participant designs

In term of machine learning:

Within participant designs allow building models:

- Which are unique to a given participant

  This is useful when the features have a high inter-participant variability

  $\Rightarrow$ Higher performance

- Which are general to all participants

  This is useful as it does not necessitate to train a model for each new user

Between participant design only allow to build a general model.

# Confounding variables

Confounding variables are those which can influence the measured variables but which are not part of the concept we aim to measure. They can:

- Introduce noise in your models;
- Lead to over- or under-estimation of your performance.

These variables should be controlled for when building models of the concept.

Visual fatigue example:

- Some people can be affected by dry eyes more than others without it being related to visual fatigue;
- The time of the recording can influence fatigue (e.g. after lunch / at night)

# Machine learning challenges for human factors

# Examples of challenges

- Multimodal learning
- Multi-target learning
- Noisy targets
- Imbalanced classes
- Inter-participant variability

# Machine learning challenges for human factors

Unbalanced classes

# Unbalanced classes: the problem

When using experimental protocols the classes are generally balanced due to the protocol design (i.e. independent variable).

However:

- When a measured variable is used as target nothing guaranties balanced classes;
- When collecting data in the field (i.e. out of the lab) it is often difficult to use a well defined protocol;
- Many phenomena important for HF occurs rarely (e.g. loss of consciousness, operation errors, etc.).

# Supervised ML reminder

- We need some data

$$D = \{(\boldsymbol{x}_1, y_1), \quad \dots \quad , (\boldsymbol{x}_n, y_n)\}$$

is the training sample  and can be a vector of features
is the associated target value (i.e. the class for classification)

- to train a model

$$\tilde{y}_i = F_\theta(\boldsymbol{x}_i)$$

is a set of parameters to obtain by training

- which aims at minimizing

$$L_{total}(D, \theta) = \frac{1}{n} \sum_{i=1}^{n} L(\tilde{y}_i, y_i)$$

is a cost / loss function

# Binary logistic regression

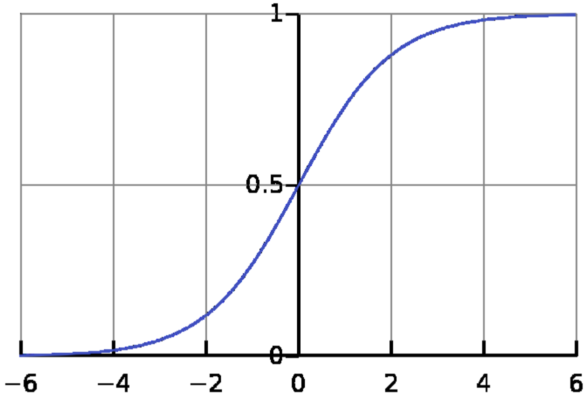
- to train a model

  {0,1}

  $\tilde{y}_i = F_\theta(\boldsymbol{x}_i)$

  $\tilde{y}_i = \sigma(\boldsymbol{\theta}^T \boldsymbol{x}_i + \theta_0)$

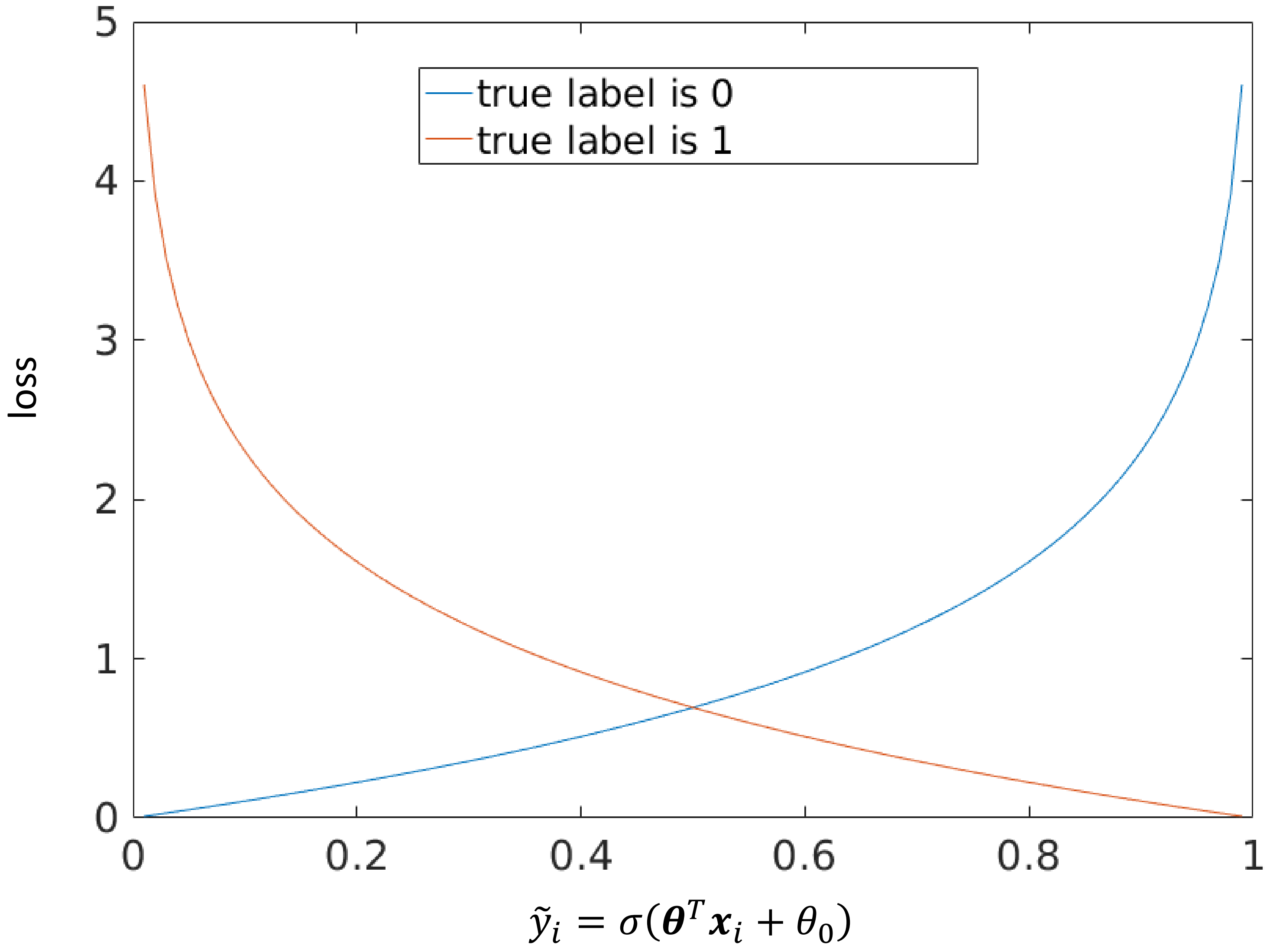
  is the sigmoid function



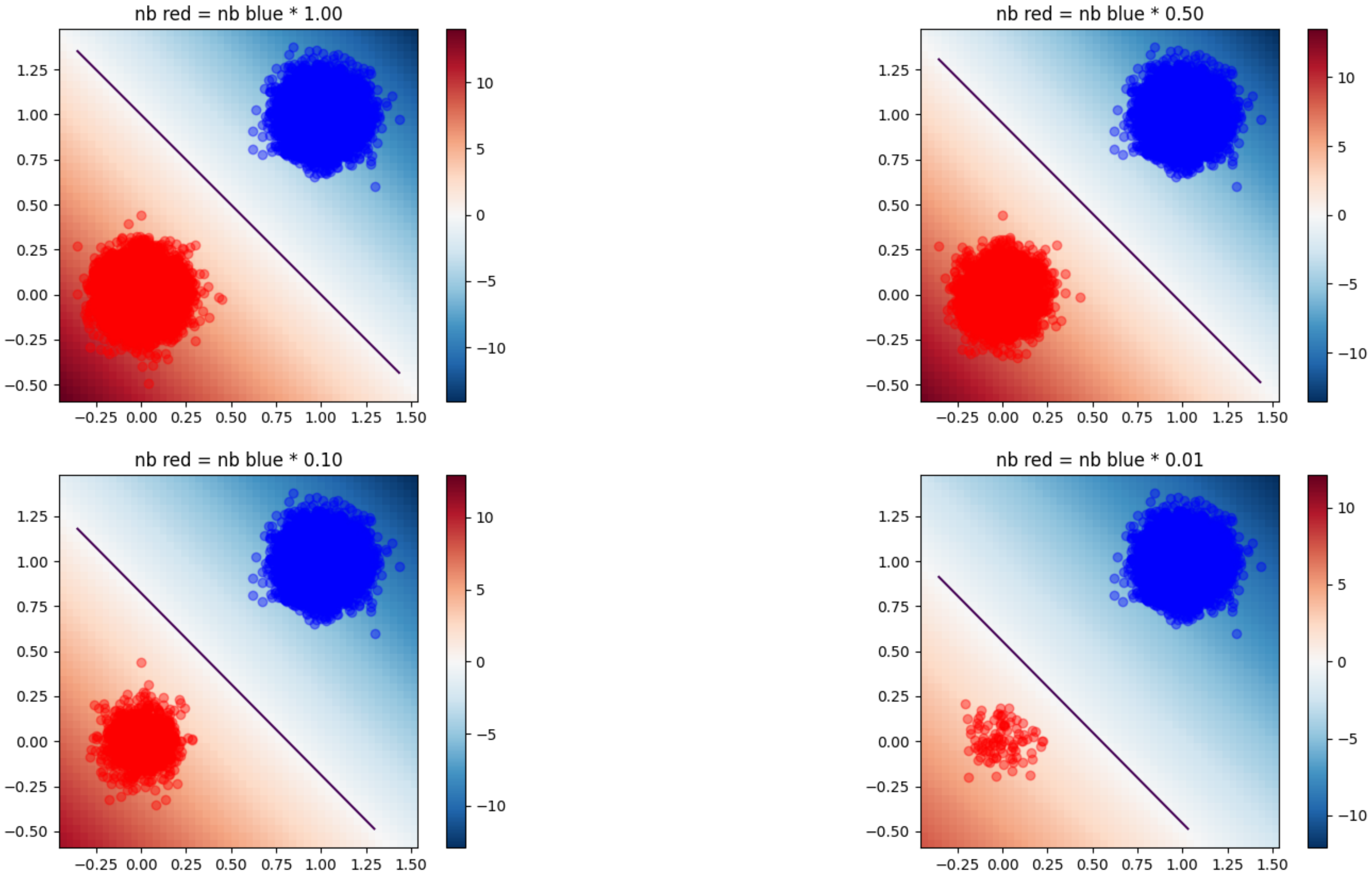
- which aims at minimizing

$$L_{total}(D,\theta) = \frac{1}{n}\sum_{i=1}^{n} -y_i log(\tilde{y}_i) - (1-y_i) log(1-\tilde{y}_i)$$

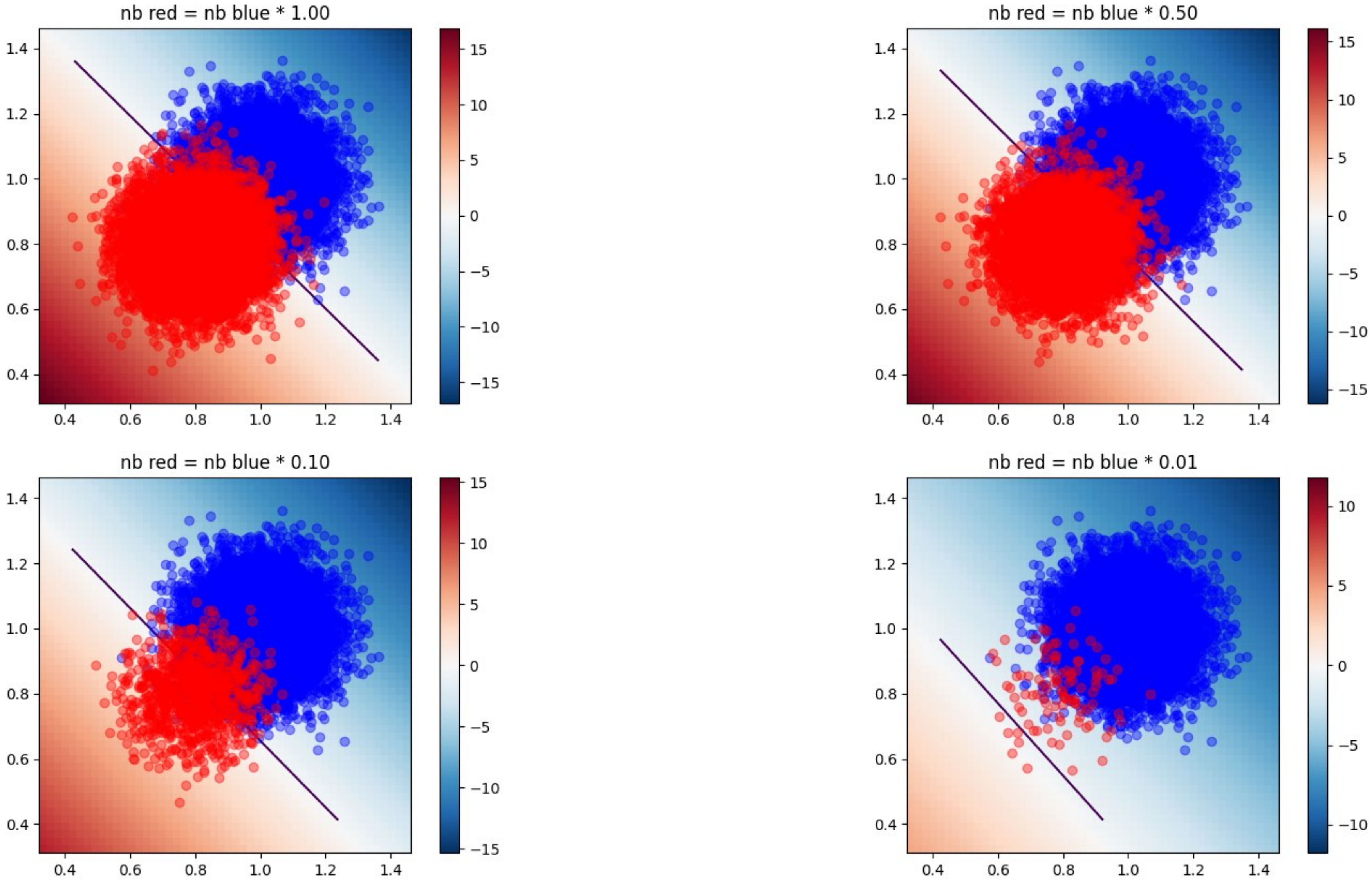
# Binary logistic regression: loss representation



# B. Logistic regression: effect of class imbalance



# B. Logistic regression: effect of class imbalance



# How to learn with class imbalance?

- Under-sampling of the majority class
    - Randomly select a subset of data
    - Regenerate samples according to the current distribution

- Over-sampling of the minority class
    - Pick data at random with replacement and duplicate
    - SMOTE (synthetic minority over-sampling technique)

Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, *16*, 321-357.

- Combination of both

# Learning with class imbalance: sample weighting

Samples which are less represented should be weighted more in the loss function.

$$L_{total}(D,\theta) = \frac{1}{n}\sum_{i=1}^{n} w_i . L(\tilde{y}_i, y_i)$$

For instance the weights can be inversely proportional to the frequency of samples in the corresponding class.

$$w_i = \begin{cases} \frac{n}{2n_0} if y_i = 0 \\ \frac{n}{2n_1} if y_i = 1 \end{cases}$$

is the number of samples belonging to class 0

is the number of samples belonging to class 1

# Classification evaluation

How to measure the performance of a classifier ?

| Ground truth | Estimated |
|---|---|
| 0 | 1 |
| 1 | 1 |
| 0 | 0 |
| 0 | 0 |
| 1 | 0 |
| 1 | 1 |
| 0 | 0 |
| 0 | 0 |
| 1 | 1 |
| 1 | 0 |

Accuracy:
Compute the percentage of correct answers

$$A = \frac{1}{N}\sum_{i=1}^{N} \delta(\tilde{y}_i - y_i)$$

$$A = \frac{7}{10} = 70\%$$

The percentage of errors is the inverse measure = 30%

Random classification should be around 50% for two classes

# Classification evaluation

## Did this classifier learn something ?

| Ground truth | Estimated |
|---|---|
| 1 | 1 |
| 1 | 1 |
| 1 | 1 |
| 1 | 1 |
| 1 | 1 |
| 1 | 1 |
| 1 | 1 |
| 0 | 1 |
| 0 | 1 |
| 0 | 1 |

$$A = \frac{7}{10} = 70\%$$

- Compare accuracy against a « random » / dummy classifier
- Are there measures which are not sensitive to unbalanced classes ?

# Classification evaluation

Let’s have a look at the confusion matrix

| | Estimated 0 | Estimated 1 | Sum |
|---|---|---|---|
| GT 0 | **4** | 1 | 5 |
| GT 1 | 2 | **3** | 5 |
| Sum | 6 | 4 | 10 |

Ground truth: 0 1 0 0 1 1 0 0 1 1

Estimated: 1 1 0 0 0 1 0 0 1 0

$$A = \frac{4+3}{10} = 70\%$$

# Classification evaluation

## Computation of precision

|  | Estimated 0 | Estimated 1 | Sum |
|---|---|---|---|
| GT 0 | **4** | 1 | 5 |
| GT 1 | 2 | **3** | 5 |
| Sum | 6 | 4 | 10 |

Ground truth: 0 1 0 0 1 1 0 0 1 1

Estimated: 1 1 0 0 0 1 0 0 1 0

$$P_0 = \frac{4}{6} = 66\% \qquad P_1 = \frac{3}{4} = 75\%$$

- “Among the 1 estimated as class 1 how many are actually from class 1?”
- Can be estimated for class 0 (false discovery rate)

# Classification evaluation

## Computation of precision

| | Estimated 0 | Estimated 1 | Sum |
|---|---|---|---|
| GT 0 | 0 | 3 | 3 |
| GT 1 | 0 | 7 | 7 |
| Sum | 0 | 10 | 10 |

Ground truth: 1 1 1 1 1 1 1 0 0 0

Estimated: 1 1 1 1 1 1 1 1 1 1

Not computable

$$P_1 = \frac{3}{4} = 75\%$$

- If not computable -> potential problem of imbalance
- Note that precision can be high even if classes are imbalanced (imagine the case where the classifier would have estimated one 0 correctly in the example above.

# Classification evaluation

## Computation of recall

Ground truth: 0 1 0 0 1 1 0 0 1 1

Estimated: 1 1 0 0 0 1 0 0 1 0

| | Estimated 0 | Estimated 1 | Sum |
|---|---|---|---|
| GT 0 | **4** | 1 | 5 |
| GT 1 | 2 | **3** | 5 |
| Sum | 6 | 4 | 10 |

$$R_0 = \frac{4}{5} = 80\%$$

$$R_1 = \frac{3}{5} = 60\%$$

- “Among all those which belong to class 1 how many were estimated as class 1”
- Also called sensitivity

- Can be computed for class 0 (specificity)

# Classification evaluation

## Computation of recall

Ground truth: 1 1 1 1 1 1 1 0 0 0

Estimated: 1 1 1 1 1 1 1 1 1 1

|  | Estimated 0 | Estimated 1 | Sum |
|---|---|---|---|
| GT 0 | **0** | 3 | 3 |
| GT 1 | 0 | **7** | 7 |
| Sum | 0 | 10 | 10 |

$$R_0 = \frac{0}{3} = 0\%$$

$$R_1 = \frac{7}{7} = 100\%$$

- Average recall is defined as:

$$R_{avg} = \frac{1}{C}\sum_{i=1}^{C} R_i$$

is the number of classes

- It is also called balanced accuracy and can be used as a performance indicator for imbalanced classes.

# Classification evaluation

## Computation of Cohen’s Kappa

Ground truth: 1 1 1 1 1 1 1 0 0 0

Estimated: 1 1 1 1 1 1 1 1 1 1

| | Estimated 0 | Estimated 1 | Sum |
|---|---|---|---|
| GT 0 | **0** | 3 | 3 |
| GT 1 | 0 | **7** | 7 |
| Sum | 0 | 10 | 10 |

$$P_E(0) = \frac{0}{10} = 0\%$$

$$P_E(1) = \frac{10}{10} = 100\%$$

$$P_{GT}(0) = \frac{3}{10} = 30\%$$

$$P_{GT}(1) = \frac{7}{10} = 70\%$$

$$Random\ agreement = R = P_{GT}(0).\ P_E(0) + P_{GT}(1).\ P_E(1) = 70\%$$

# Classification evaluation

## Computation of Cohen’s Kappa

$$K = \frac{A - R}{1 - R}$$

For imbalanced classes and a dummy classifier

$$K = \frac{70\% - 70\%}{100\% - 70\%} = 0$$

For balanced classes

$$K = \frac{70\% - 50\%}{100\% - 50\%} = 0.4$$

- Cohen’s Kappa account for ground truth and estimator unbalance
- The K value can be interpreted as follows:
  - 0: at chance level
  - < 0: worse than chance level
  - 1: perfect classifier

# Classification evaluation

For multiple classes…

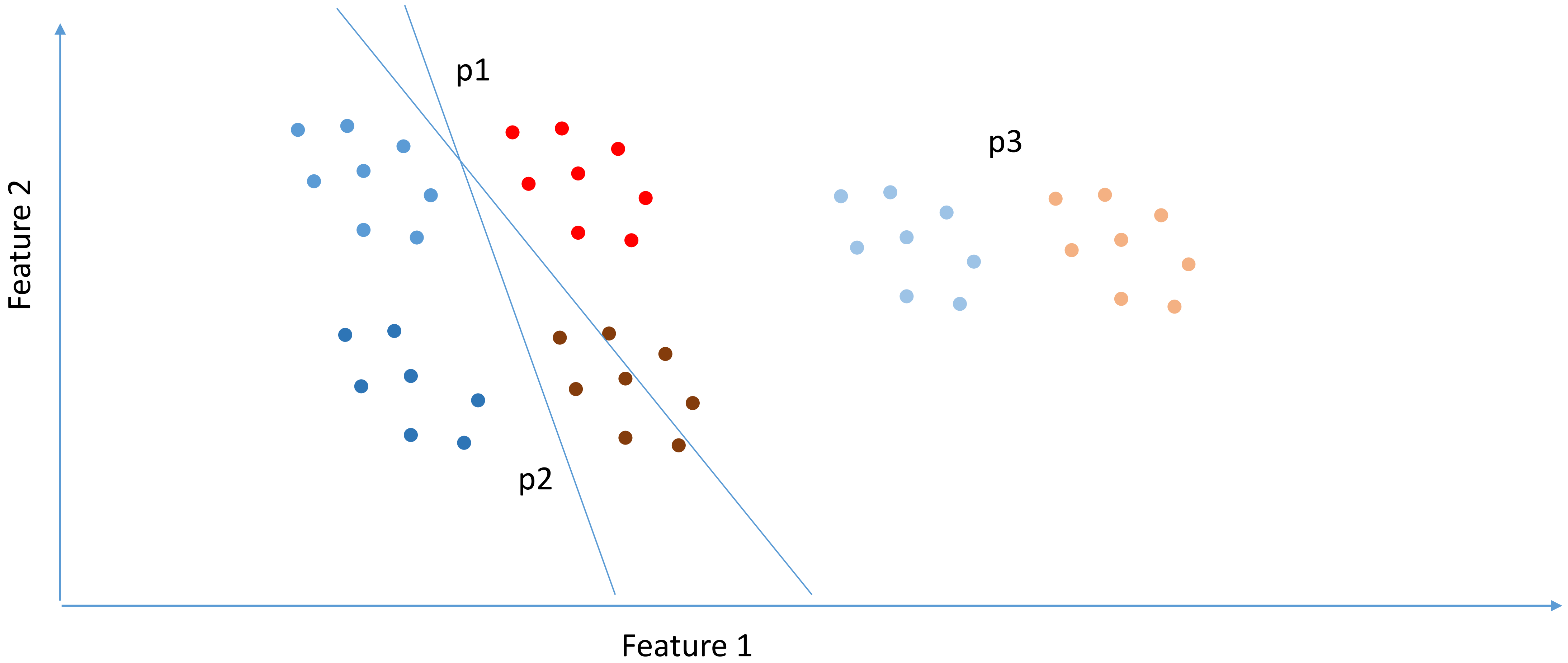
- Accuracy can be directly computed
- Cohen’s Kappa can be adapted to multiple classes
- Recall and  precision are computed for each class

# Machine learning challenges for human factors

Inter-participant variability

# Inter-participant variability: the problem

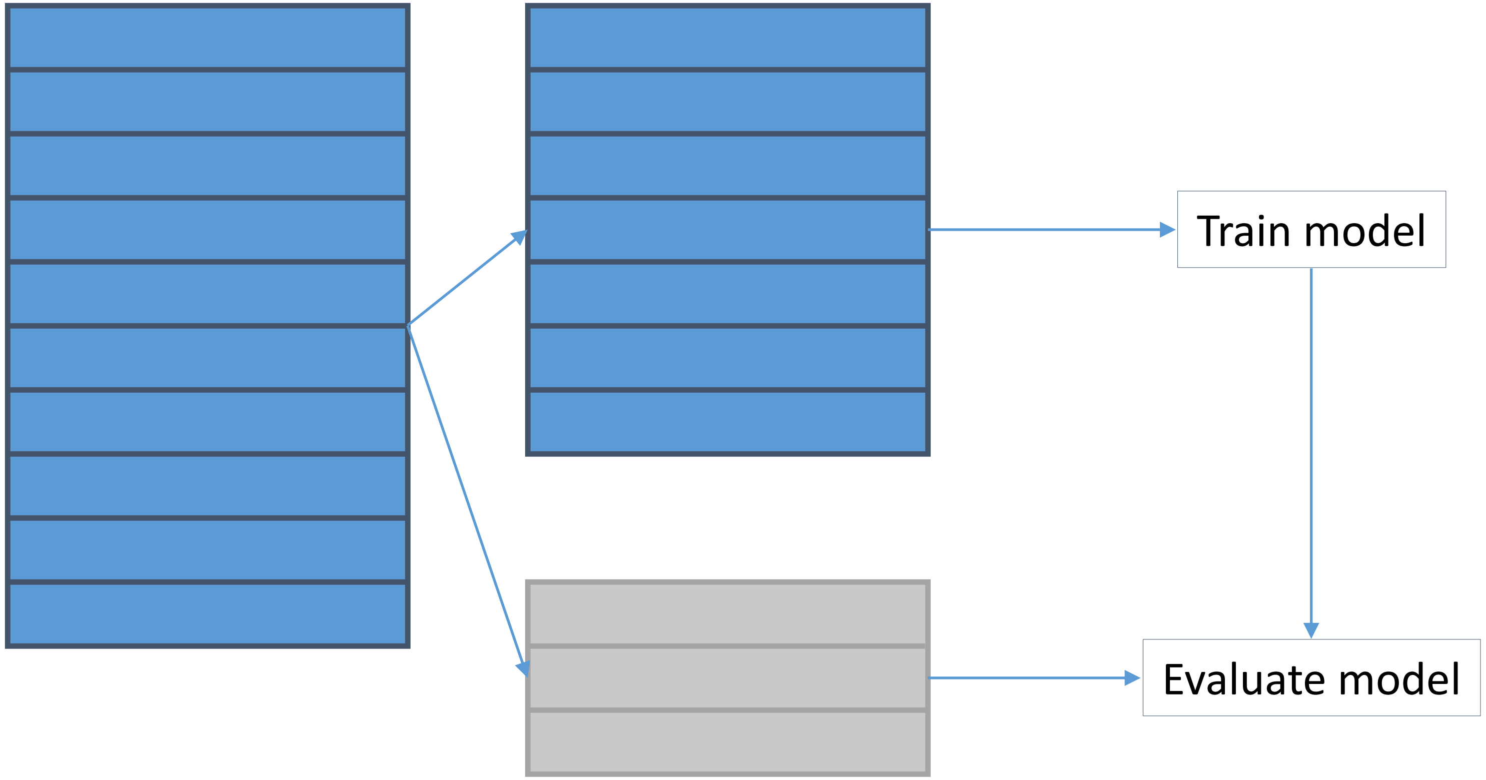
As previously seen human behaviors are highly variable from a person to another (although there might be some common patterns)



# Training and test sets: reminder

We divide the data in two sets:

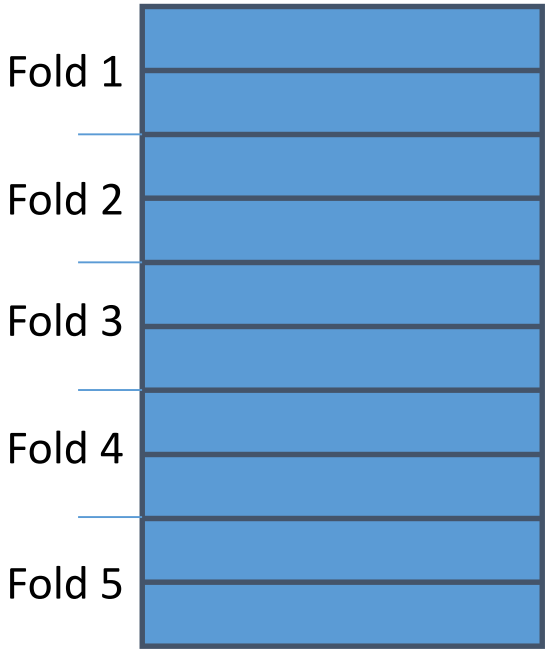
- The training set is used for training the classifier / the model
- The test set is used for evaluating the classifier



# Cross-validation (CV): reminder

When we do not have enough data:

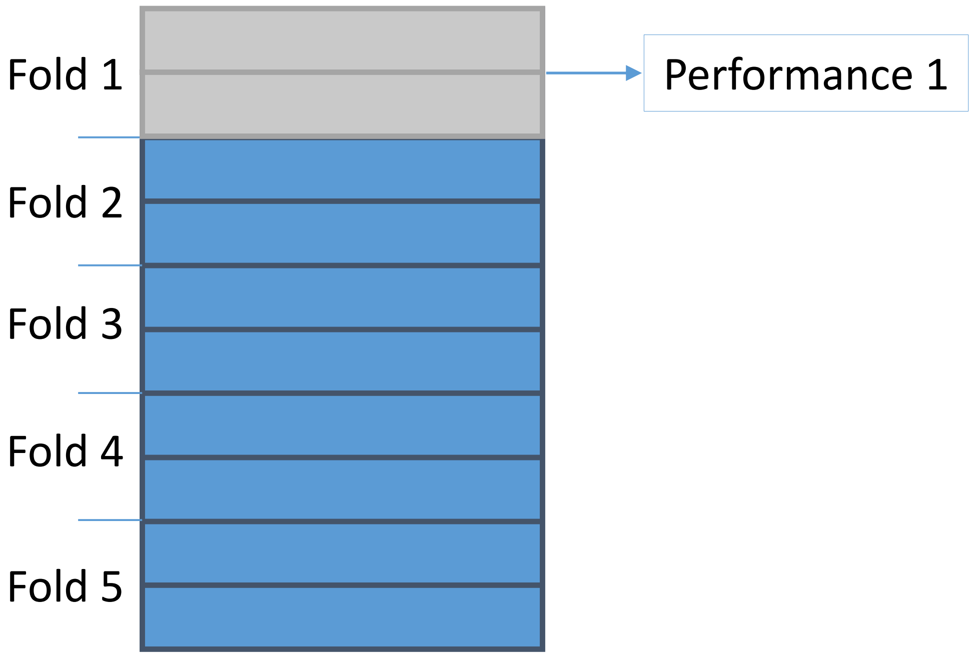
- Split the data in k-folds (k-fold cross validation)
- Use each fold in turn as the test set and the other folds as the training set



# Cross-validation (CV): reminder

When we do not have enough data:

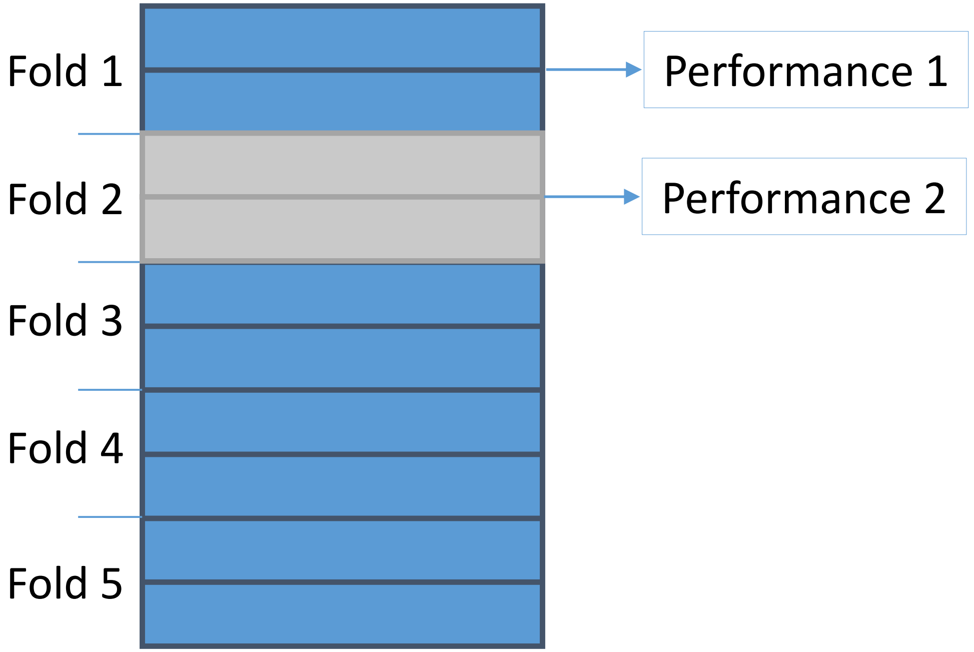
- Split the data in k-folds (k-fold cross validation)
- Use each fold in turn as the test set and the other folds as the training set



# Cross-validation (CV): reminder

When we do not have enough data:

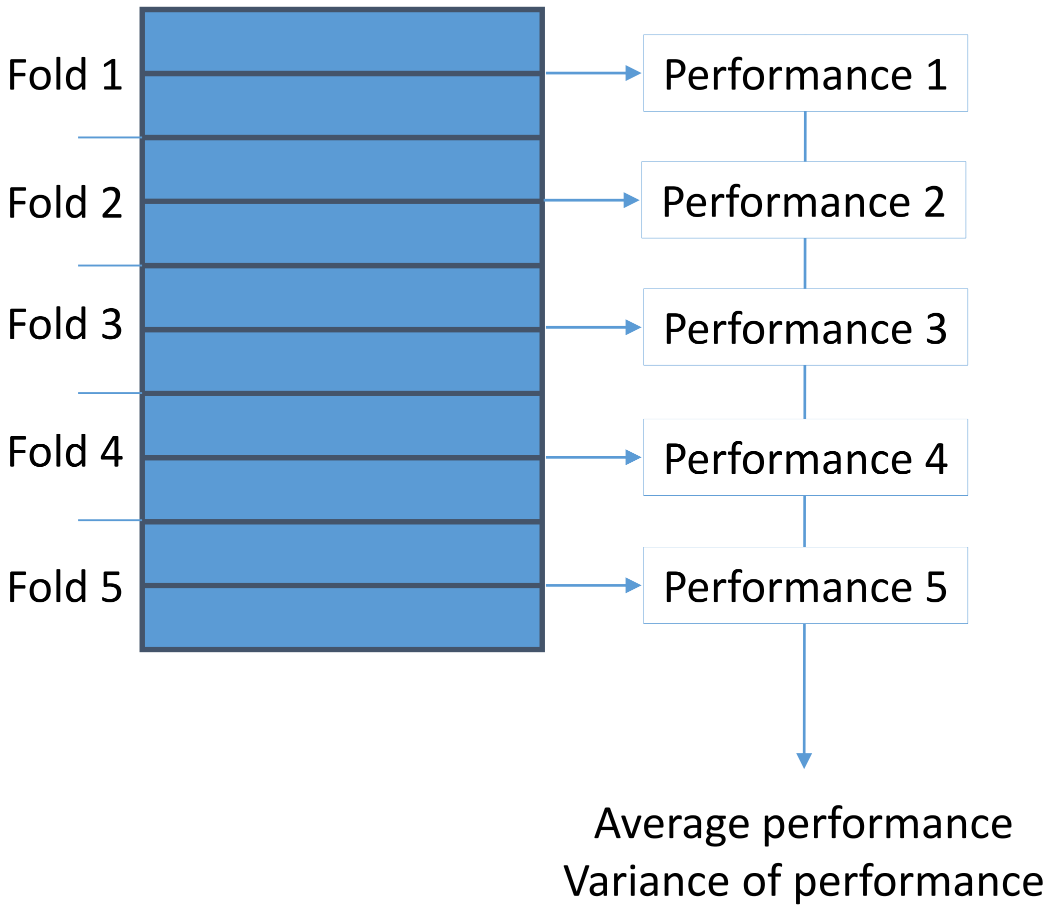
- Split the data in k-folds (k-fold cross validation)
- Use each fold in turn as the test set and the other folds as the training set



# Cross-validation (CV): reminder

When we do not have enough data:

- Split the data in k-folds (k-fold cross validation)
- Use each fold in turn as the test set and the other folds as the training set



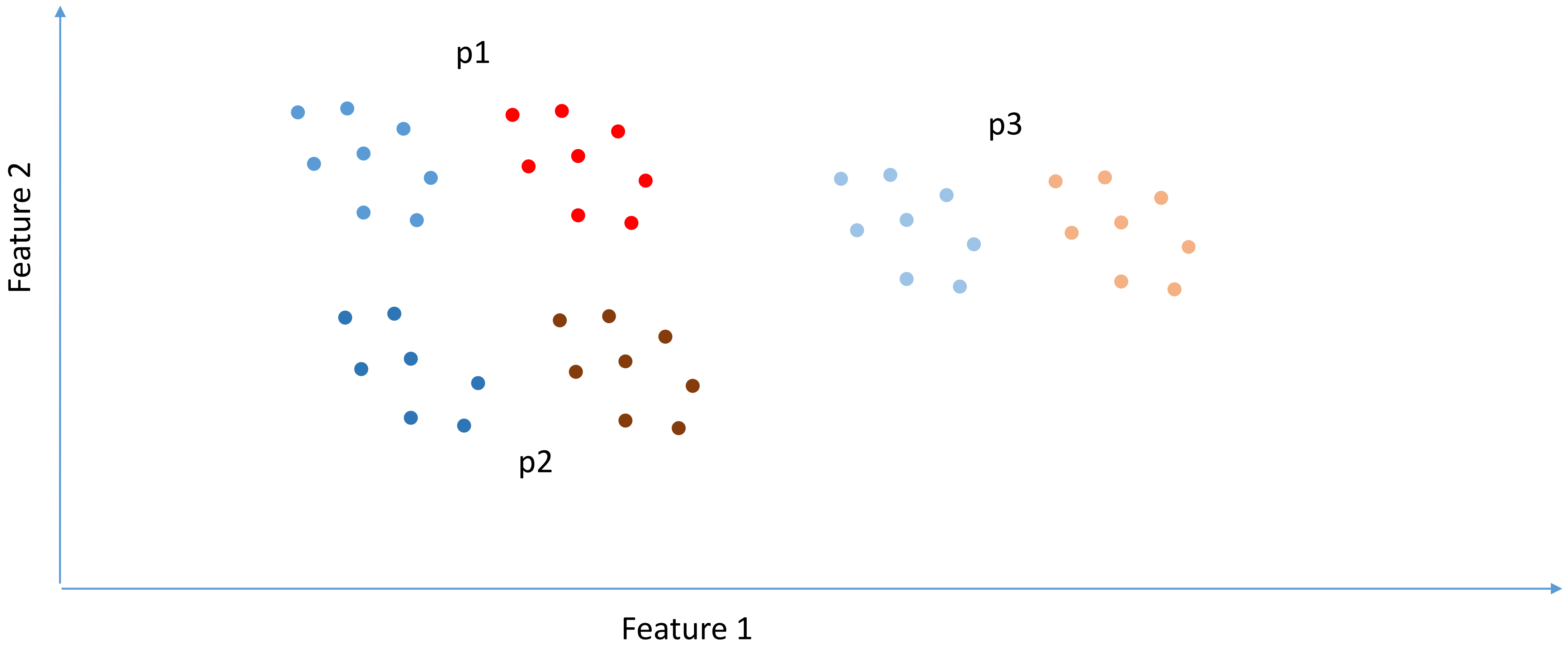
# Classifiers  hyper-parameters: reminder

When you need to set some hyper-parameters of classifiers the value should be chosen by:

- Using a validation set (i.e. training + validation + testing)

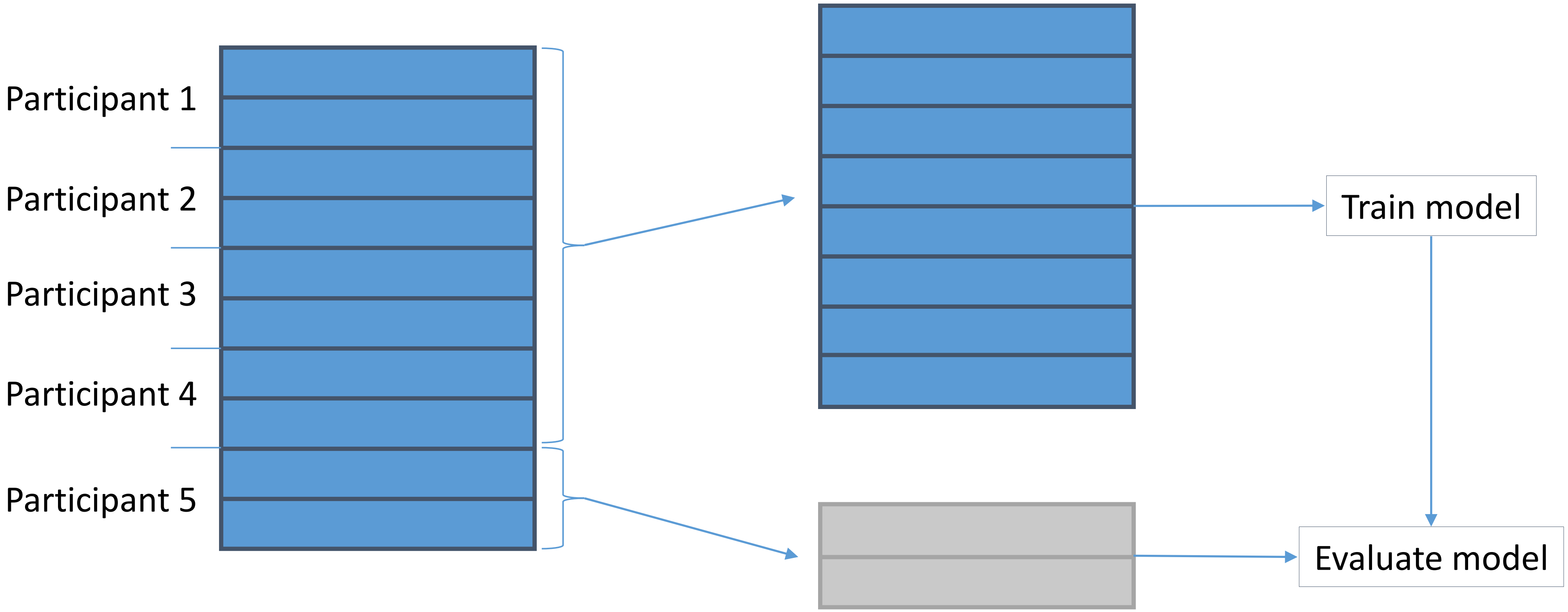
- Using a nested cross-validation loop
  - A CV loop for classifier accuracy
  - A nested CV loop for testing different values of the parameters and choosing the best

# Inter-participant variability: training set vs. test set



# Interparticipant variability: group CV

To ensure generalization across participants, one should ensure that test and validation sets do not include data from the participants in the training set.



# Interparticipant variability: potential solutions

- Features engineering: favor relative features rather than absolute features
  - Use derivative of signals
  - Changes with respect to a baseline

- Normalize data per participant (requires to have enough testing data, application dependent)

- The field of transfer learning aims at tackling variability among various domains