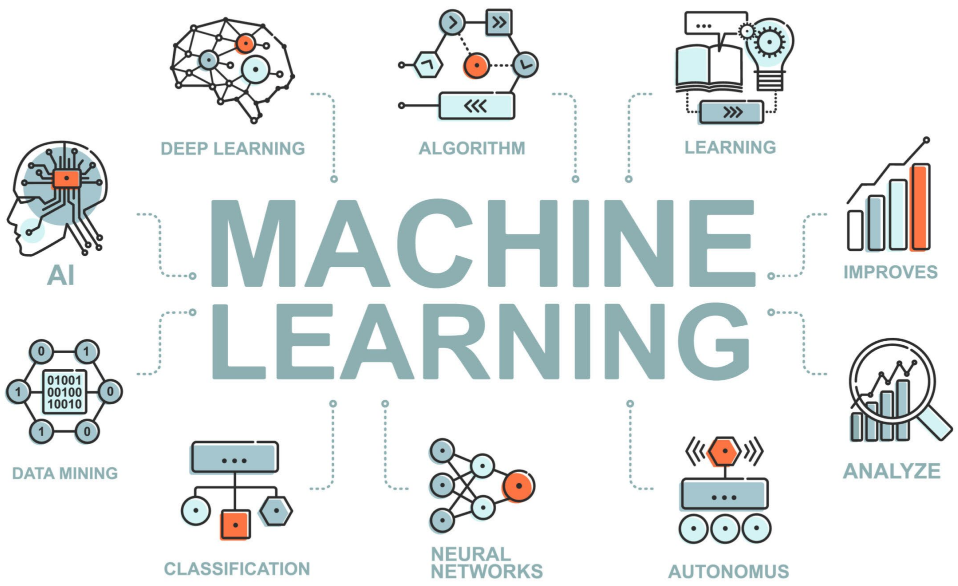


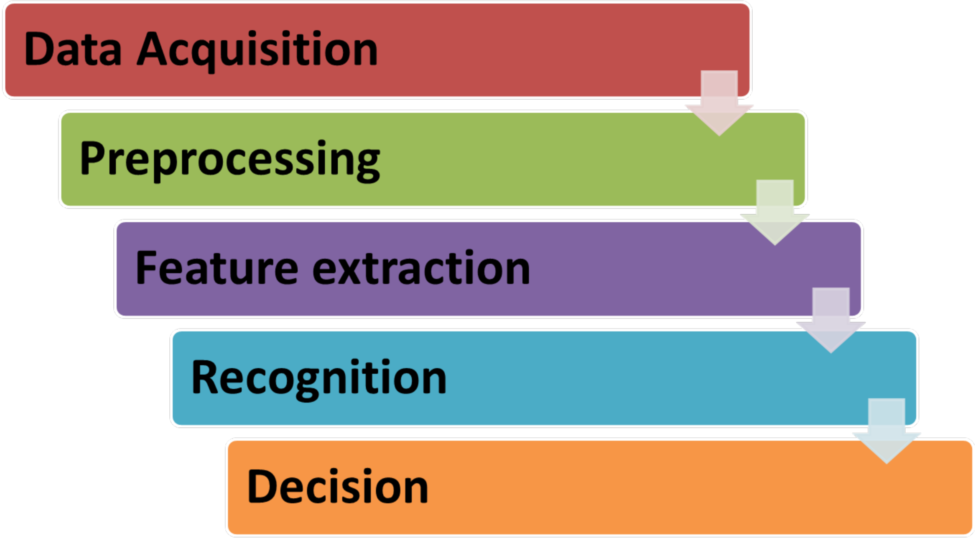
Hes·so VALAIS WALLIS
School of Engineering

ML Data pipeline

Francesco Carrino

Outline

- Learning Process General Schema
- Unbalanced training set
- Feature Normalization
- Diagnosis: bias vs. variance
- Cross-Validation
  - Definition
  - Motivations and goals
  - Procedures and applications
- Performance indicators
  - Confusion matrix
  - Accuracy, Precision, Recall, Specificity, etc.

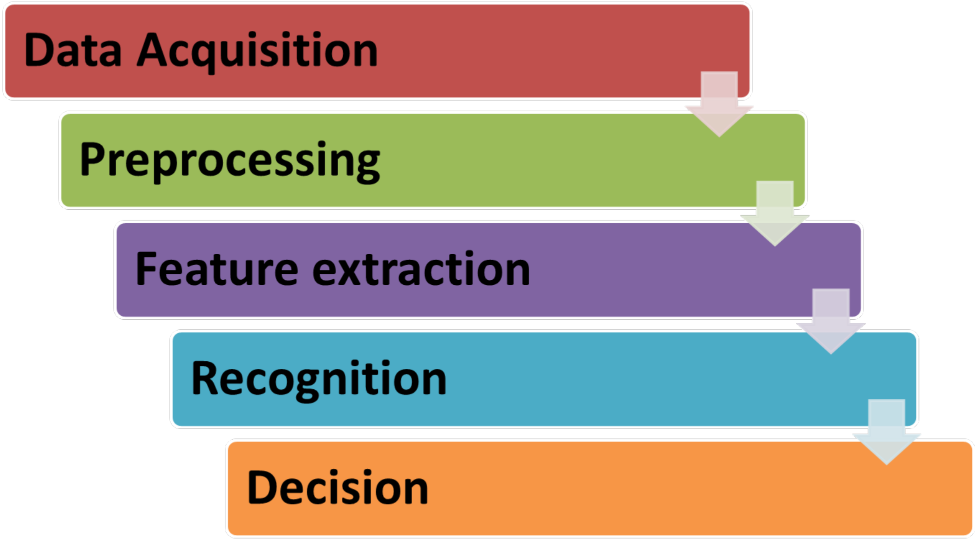


Introduction

- Advice on (*any*) machine learning
  - K-NN, Random Forest, NN, SVM, HMM, etc.

- How to **properly** manage data?
  - Feature selection
  - Normalization
  - Dataset splitting

- How to **properly** evaluate the classification results?



Learning Process – General Schema

Data Acquisition
- Sensors drivers
- Synchronization

Preprocessing
- Filtering, cropping, dropping, denoising, …
- Segmentation

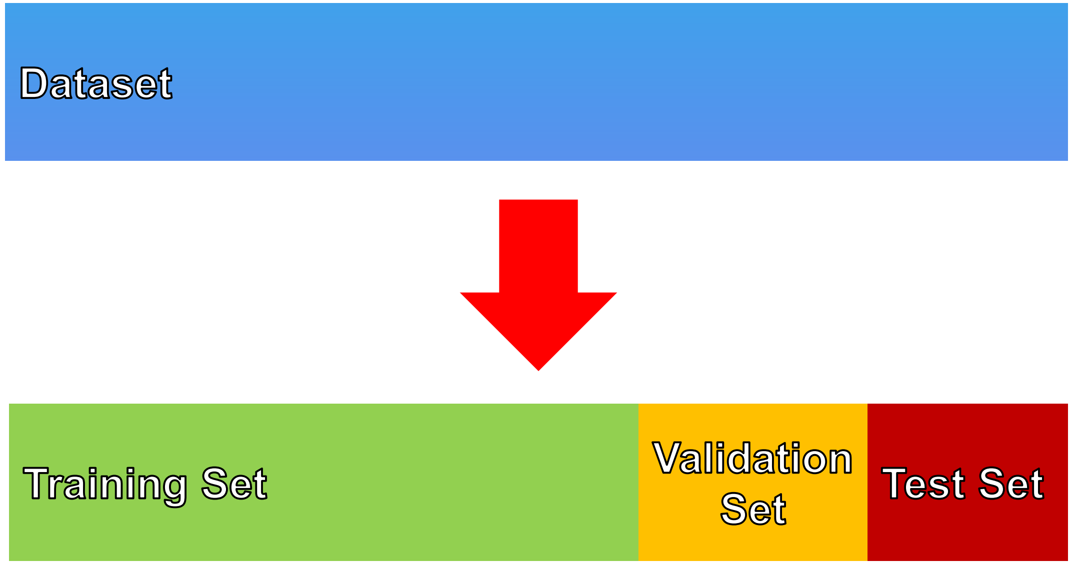
Feature extraction
- Feature construction (selection)
- Feature reduction

Recognition
- Supervised approaches
- Unsupervised approaches

Decision
- Fusion

What and where is the role of an « **infotronics** » in ML?

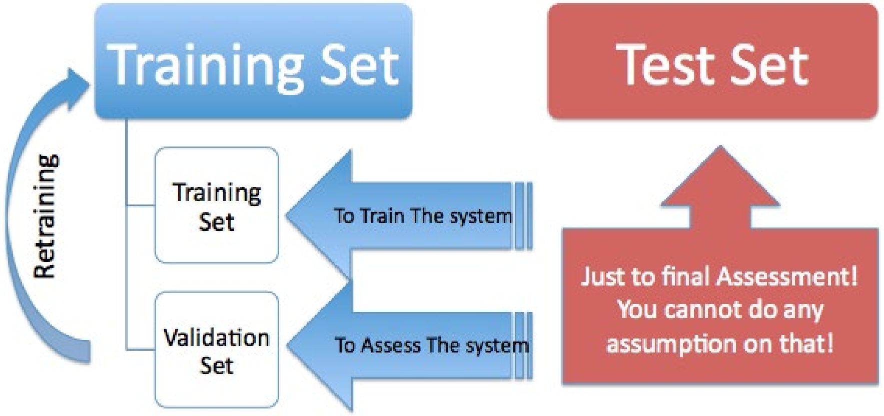
Learning Process – General Schema



Learning Process – General Schema

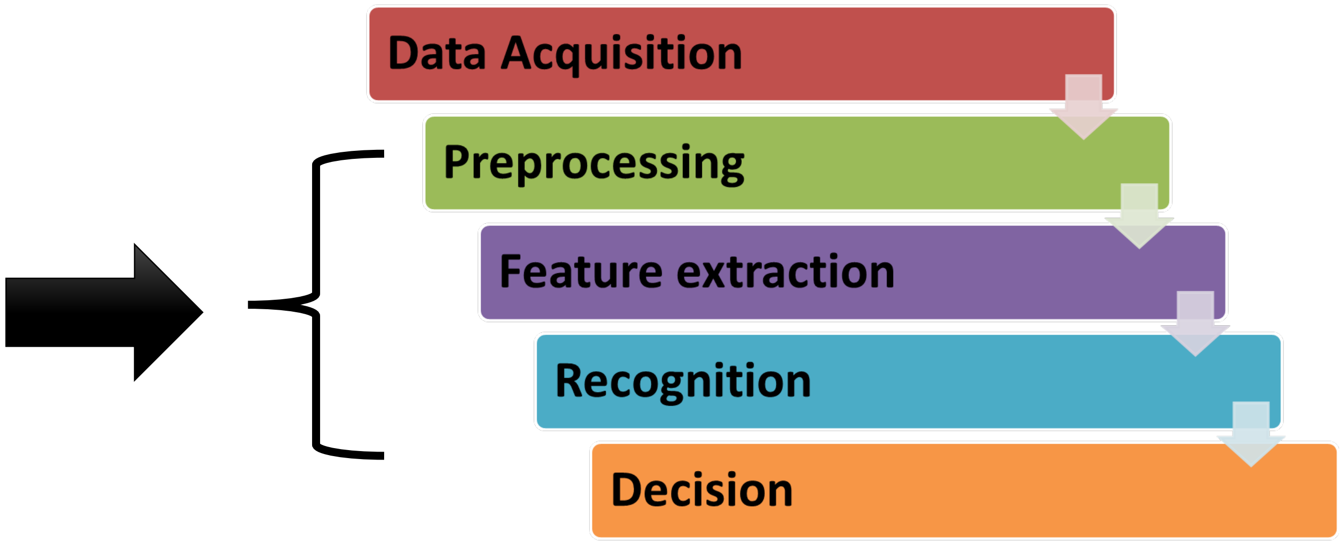
Steps

1. Training Set:
   - Feature extraction
   - Data Modelization
2. Validation Set
   - Optimization of the model
3. Iterate 1 and 2
4. Test Set:
   - Final assessment!
   - No assumption using these data



http://textanddatamining.blogspot.ch/2011/09/how-classifier-accuracy-is-conditioned.html

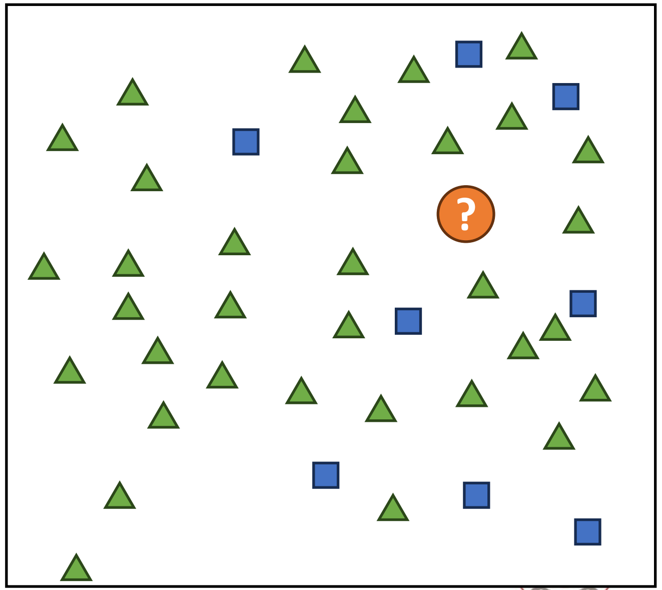
Balancing the Training set



Unbalanced training set

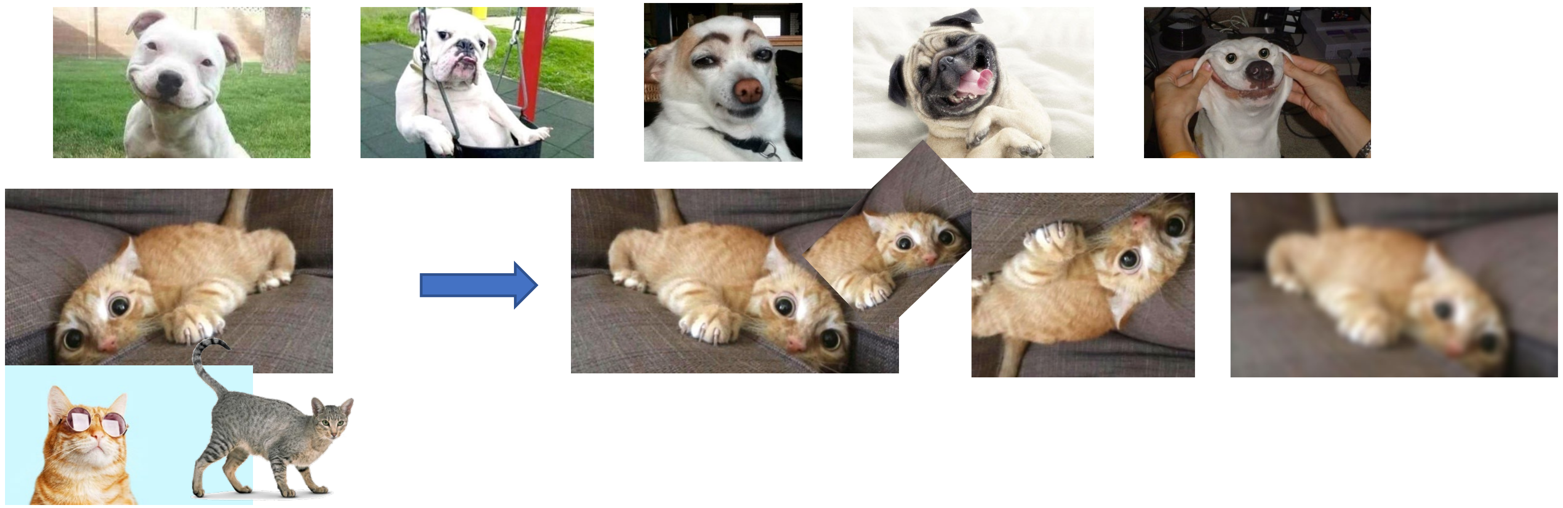
?

In your opinion, K-NN performances are impacted by an unbalanced training set?



- Fish or duck?
  - Training set "unbalanced" (or skewed)
- With an "unbalanced" ***training set*** some classifiers have **bad performance**

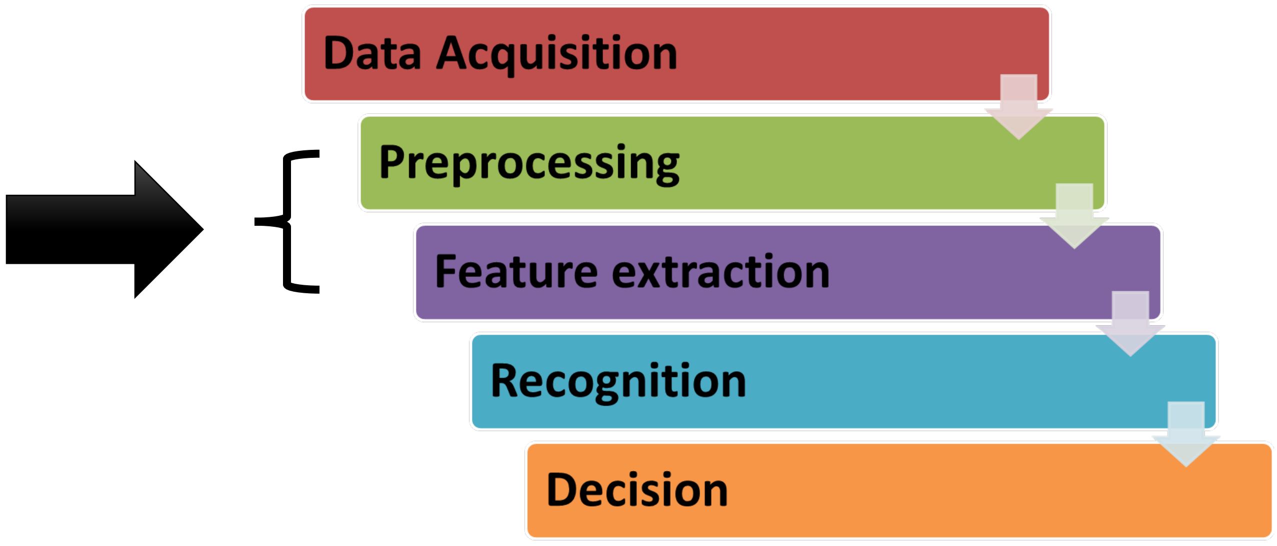
Unbalanced training set



- **Solutions:**
  - make the training set **balanced**
  - samples belonging to the less represented class can be randomly repeated
  - give more importance to errors on the smaller class

Thanks to machine-learning algorithms,
the robot apocalypse was short-lived.

Features scaling (normalization)



Features scaling (normalization)

- How to treat features having different scales?

- Some machine learning algorithms (K-NN, SVM, NN & others) **ignore** features with the smaller scale!

Features scaling (normalization)

- Example: predict flat energy label (A or B?) based on:
  - Feature 1: # of rooms
  - Feature 2: price of the flat

Label	# of rooms	Price
B	4	300’000
A	12	2’000’000
B	6	650’000
B	4.5	400’000
A	4.5	480’000
A	8	1’200’000
…	…	…

Solution 1: Features Rescaling

- Rescaling (or Min-Max)
  - Features are rescaled in the range of **[0,1]** :

$$x' = \frac{x - min(x)}{max(x) - min(x)}$$

Label	# of rooms	Price
B	4	300’000
A	12	2’000’000
B	6	650’000
B	4.5	400’000
A	4.5	480’000
A	8	1’200’000
…	…	…

Sample	# of rooms	Price
1	0	0
2	1	1
3	0.25	0.206
4	0.0625	0.059
5	0.0625	0.106
6	0.5	0.882
…	…	…

Solution 2: Standardization

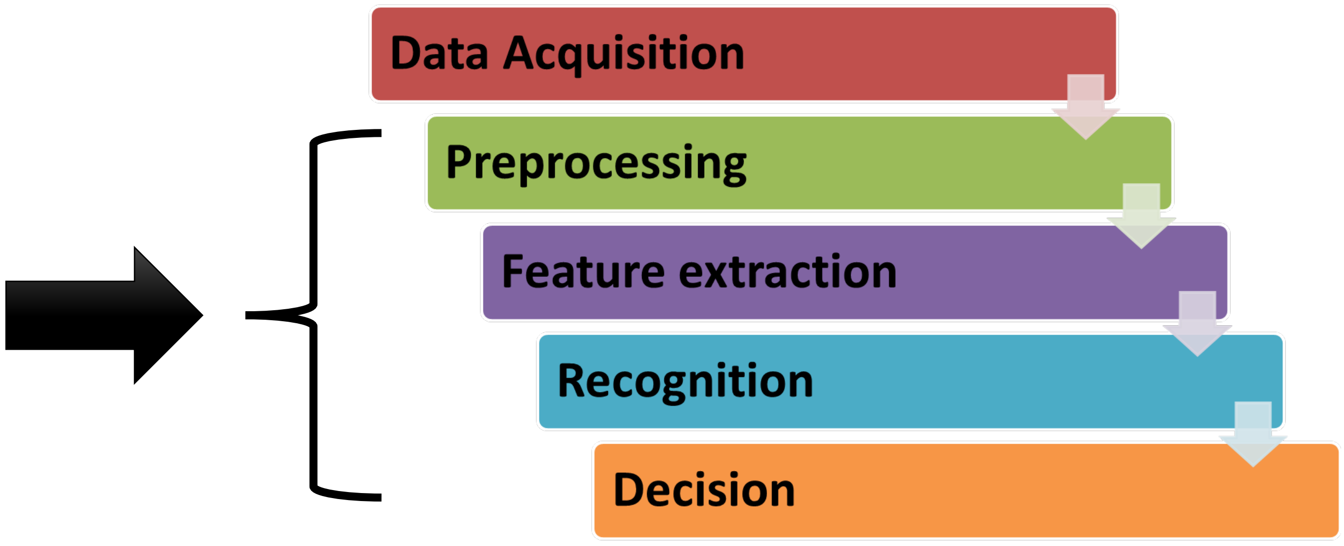
- Standardization
  - Feature standardization makes the values of each feature in the data have **zero-mean** and **unit-variance**

$$x' = \frac{x - mean(x)}{std(x)}$$

Label	# of rooms	Price
B	4	300'000
A	12	2'000'000
B	6	650'000
B	4.5	400'000
A	4.5	480'000
A	8	1'200'000
...	...	...

Sample	# of rooms	Price
1	-0.89324	-0.90435
2	1.965121	1.951493
3	-0.17865	-0.31638
4	-0.71459	-0.73636
5	-0.71459	-0.60197
6	0.535942	0.607567
...	...	...

CROSS-VALIDATION



Cross-Validation

Motivation and Goals

- Reminder: the goal of machine learning is automatically extracting relevant information from data and applying it to analyze new data
  - Regression
  - Classification
- Problem
  - Good prediction capability on the training data
  - **But** might fail to predict future *unseen* data
- **We need a procedure for estimating the generalization performance!**

Cross-Validation

Definition

Cross-Validation

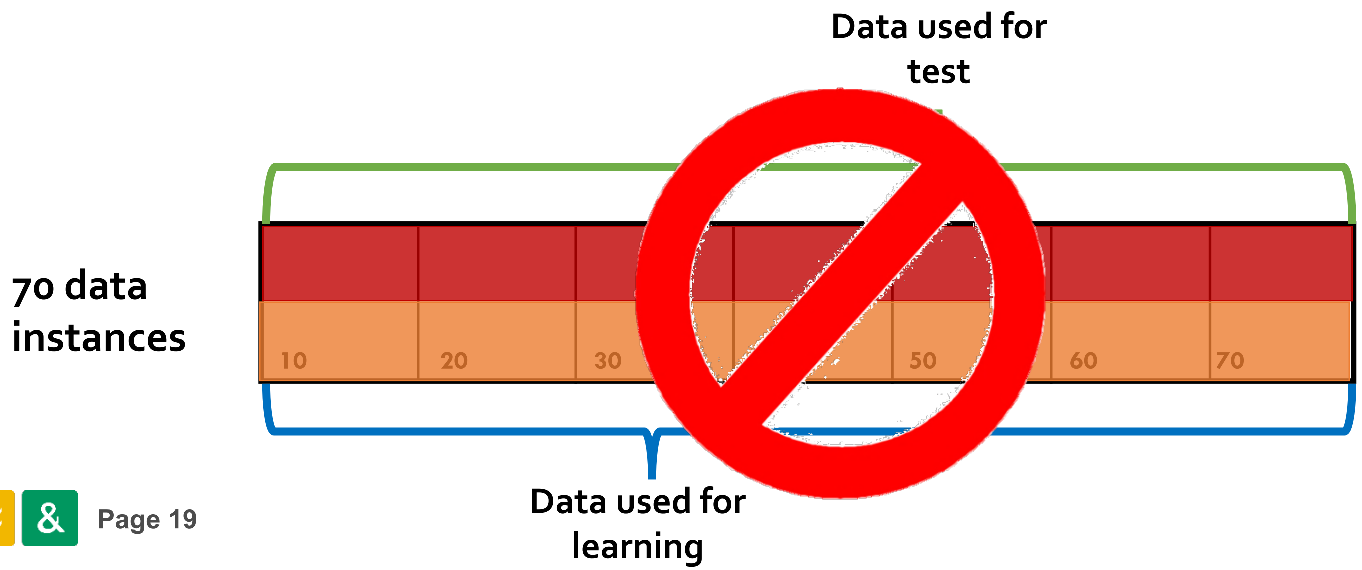
"A statistical method for evaluating and comparing learning algorithms by dividing data into two segments: one used to learn (or train) a model and the other used to validate the model."

«Cross-Validation», Payam Refaeilzadeh, Lei Tang, Huan Liu, Arizona State University[1]

Resubstitution Validation

Types of cross-validation

- **Resubstitution Validation**
  - Learning from all the available data
  - Test on all the available data
    - Pros: it uses all the available data
    - Cons: it suffers **seriously** from overfitting

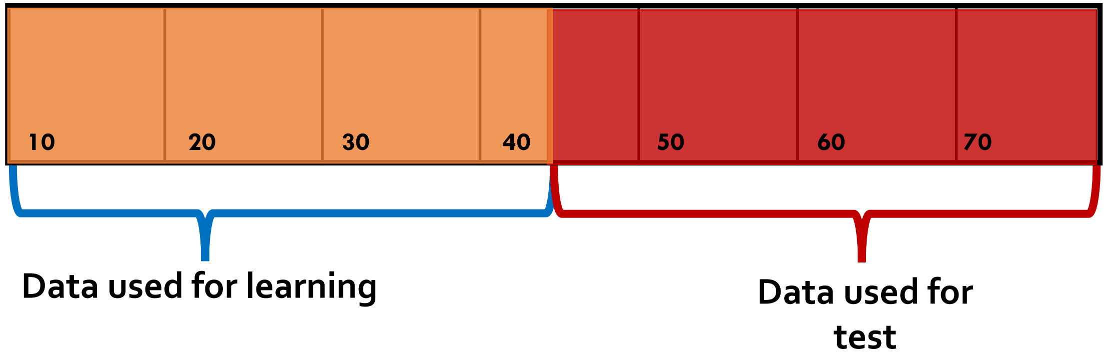


Hold-Out Validation

Types of cross-validation

- **Hold-Out Validation** 50/50
  - Learning from half of the available data
  - Test on the other half of data. The test data is held out and not looked at during training.
    - Pros: it avoids the overlap between training data and test data
    - Cons:
      - Do not use all the available data for the training
      - Results highly dependent on the choice for the training/test split

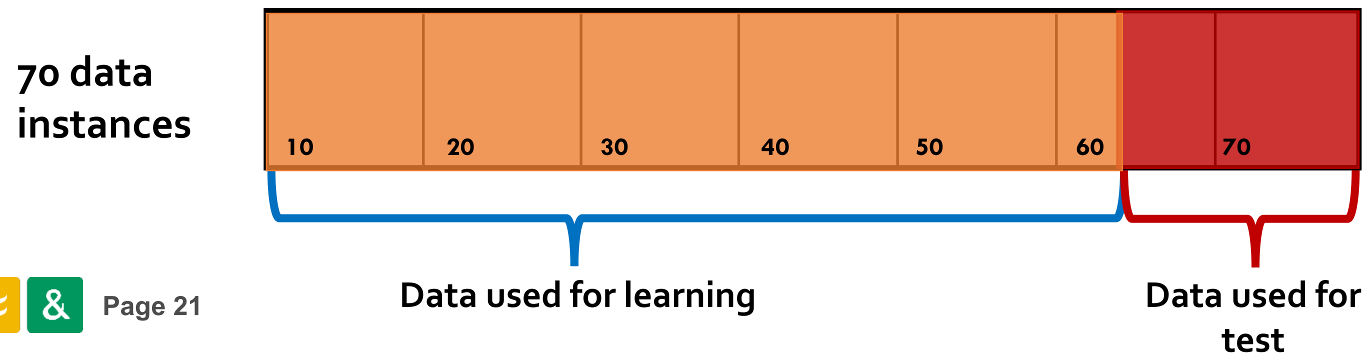
70 data instances



Hold-Out Validation

Types of cross-validation

- **Hold-Out Validation** (80-20)
  - Learning from 75-85% the available data
  - Test on the remaining data. The test data is held out and not looked at during training.
    - Pros: it avoids the overlap between training data and test data
    - Cons:
      - Do not use all the available data for the training
      - Results highly dependent on the choice for the training/test split

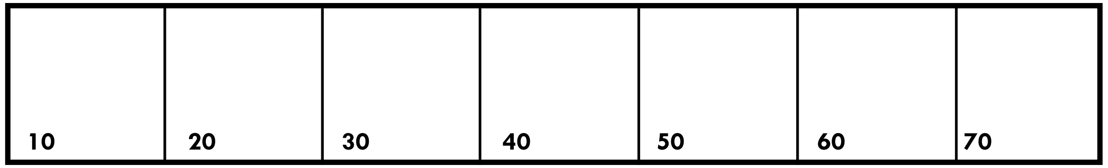


K-fold Cross-validation

Types of cross-validation

- **K-fold Cross-validation**
  - The data is first partitioned into *k* equally sized segments (or folds)
  - K iterations of training and validation, where:
    - Learning on k-1 folds
    - Test on the held-out fold

70 data instances

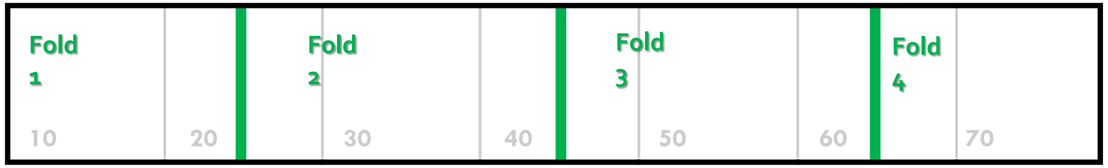


K-fold Cross-validation

Types of cross-validation

- **K-fold Cross-validation**
  - Example: 4-folds

70 data instances, K=4

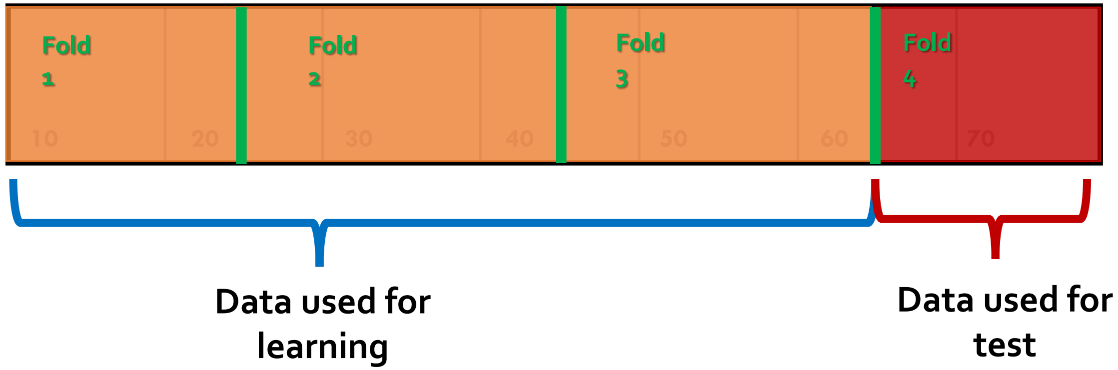


K-fold Cross-validation

Types of cross-validation

- **K-fold Cross-validation**
  - Example: 4-folds

70 data instances, K=4
1st iteration



K-fold Cross-validation

Types of cross-validation

- **K-fold Cross-validation**
  - Example: 4-folds

70 data instances, K=4

1st iteration

2nd iteration

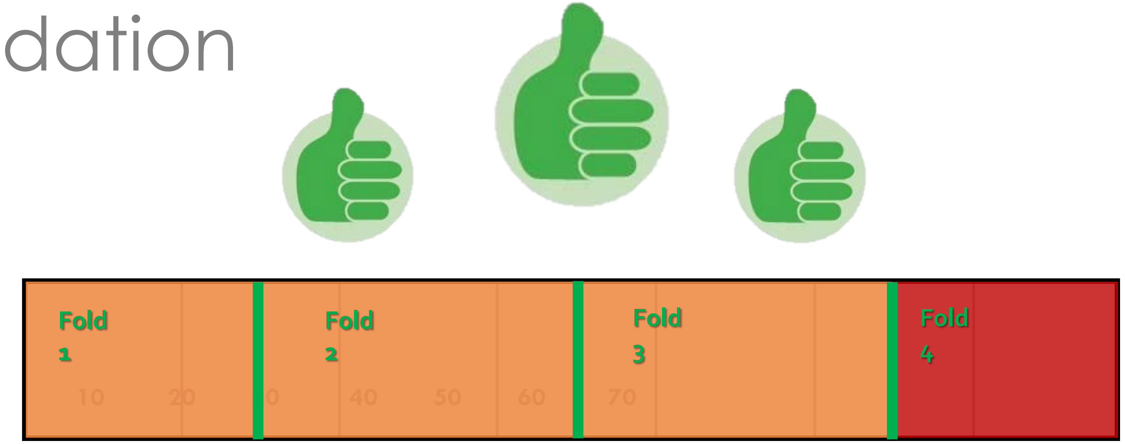
3rd iteration

4th iteration

K-fold Cross-validation

Types of cross-validation

70 data instances,
K=4



- **Note:** data are commonly **stratified**, first
  - Rearranging the data ensuring that each fold is a good representative of the whole (i.e., the training set).
- **Pros**
  - It uses all the available data
  - It avoids the overlap between training data and test data
  - Accurate performance estimation also if few samples are available
- **Cons**
  - Limited samples for performance estimation

K k k?

What is the right number of folds?

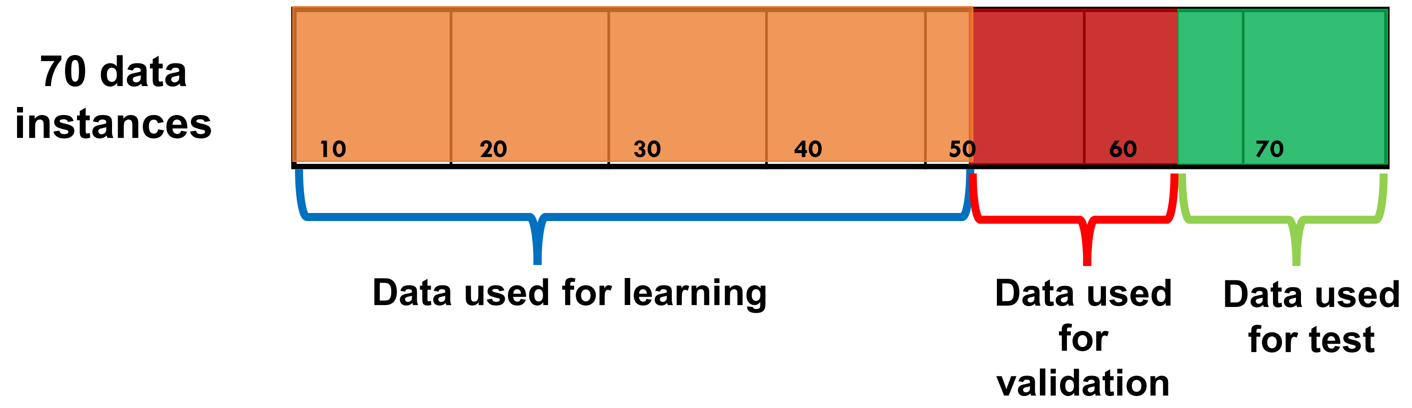
- Larger k…
  - More performance estimations
  - The training set size is closer to the full data size
    - Good generalization
  - The overlap between training sets increases
  - The test set size is very reduced
    - Less precise measurements of the accuracy

- In practice…
  - Bigger the k means longer computation time
  - **K=10** is a good compromise

Full procedure

Model selection with k-fold cross-validation

- **How to chose the optimal hyperparmeters of a model?**
  - Learning from 60% of the available data
  - Validation from 20% of the available data
    - Here we choose the best parameters
  - Test from 20% of the available data



Full procedure

Warning too many “k”

- K-fold (k = number of folds)
- K-NN (k = number of neighbors)

- **So…**
  - In the following example, we will use **n** to indicate the number of **n**eighbors to consider in the k-nn algorithm

Full procedure

Steps 1

- **Exemple: tuning the K-NN (i.e., find the best *hyperparameter* "*n*")**
  - Step 1: put aside the test set (*remember* to stratify the data first)

Full procedure

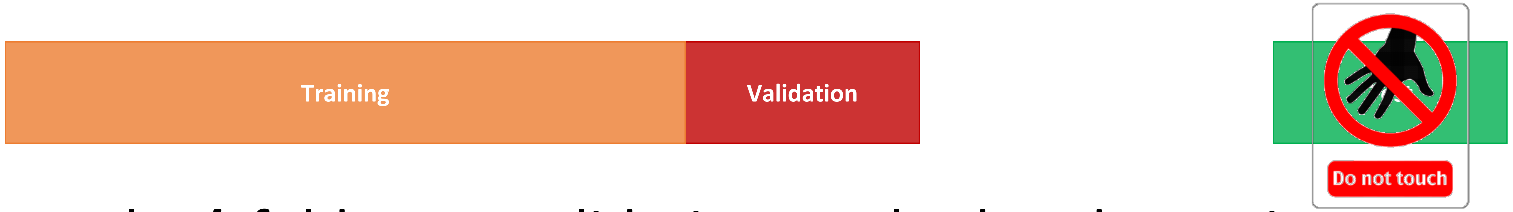
Steps 1

- **Exemple: tuning the K-NN (i.e., find the best *hyperparameter* "*n*")**
  - Step 1: put aside the test set (*remember* to stratify the data first)

Full procedure

Steps 1-2

- **Exemple: tuning the K-NN (i.e., find the best *hyperparameter* "n")**
  - Step 1: put aside the test set (*remember* to stratify the data first)



  - Step 2: use the ***k*** fold cross-validation method to determine the number "n" that optimize the accuracy

…. (repeat K times)

Full procedure

Steps 3-5

- **Exemple: tuning the K-NN (i.e., find the best *hyperparameter "n"*)**
  - Step 3: calculate the mean accuracy as a function of the hyperparameter n: *n*=* **best** *n (i.e., the n* that optimize the accuracy).

  - *Step 4: train your algorithm using n* over the **whole** dataset*

  - *Step 5: evaluate your algorithm on the test set (**unseen** until now!)*

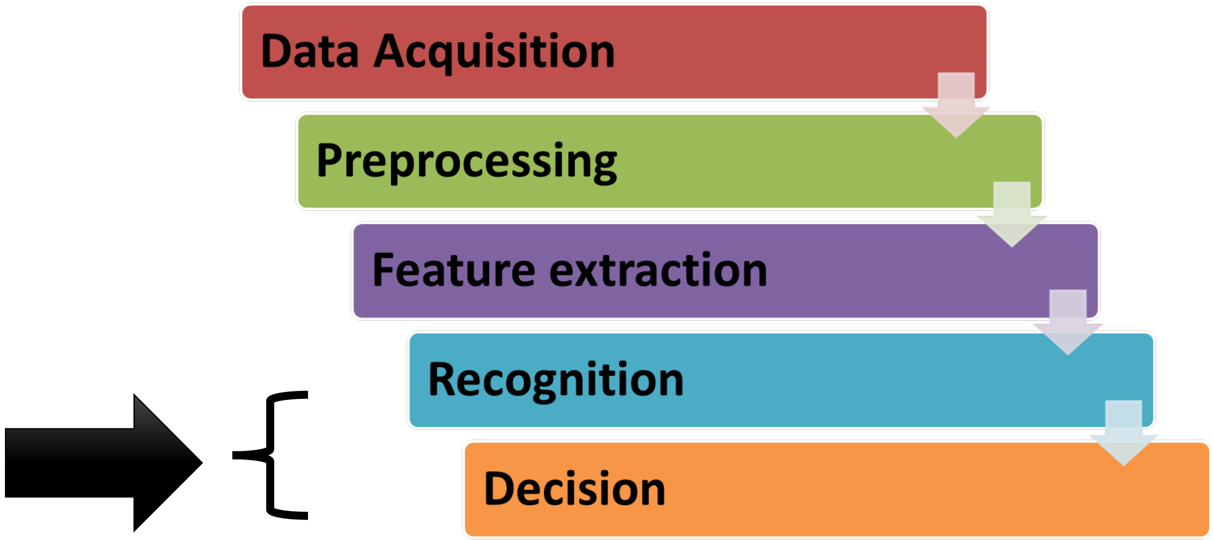
Applications

- **Obtain reliable performances estimation**
  - *Accuracy*
  - *Precision*
  - *Recall*
  - *F-score*
  - …

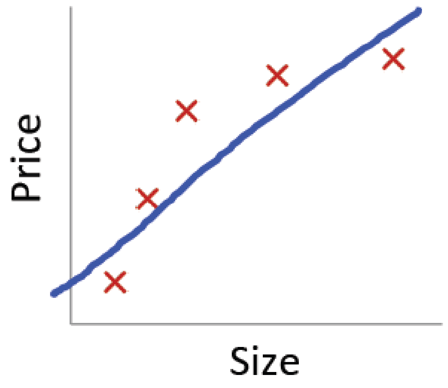
- **Algorithm Tuning**
  - Feature selection to maximize classifiers performances on a particular dataset
  - Find the parameters that optimize the classifiers
    - K for the k-NN
    - Number of tree for Random Forest
    - etc.

Underfitting Vs Overfitting

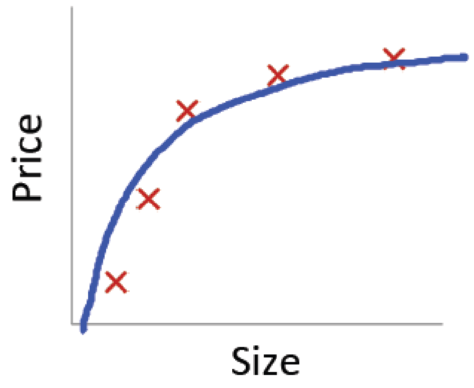
A.k.a., Bias Vs Variance (source: [7])



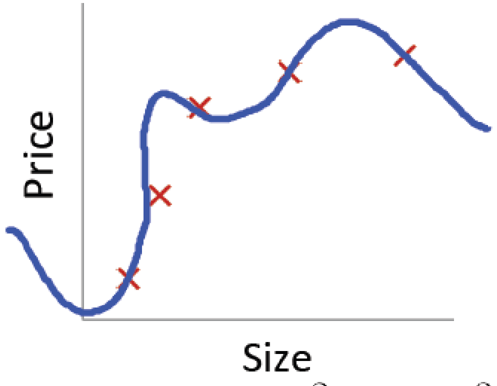
Underfitting Vs Overfitting



$$\theta_0 + \theta_1 x$$



$$\theta_0 + \theta_1 x + \theta_2 x^2$$



$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

Underfitting: the model is too simple to describe the data

“Just right”

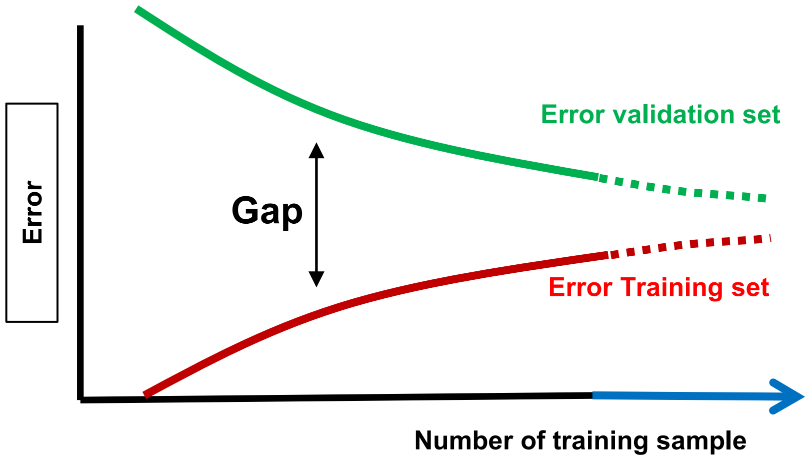
?

Difference between parameters and *hyper*parameters

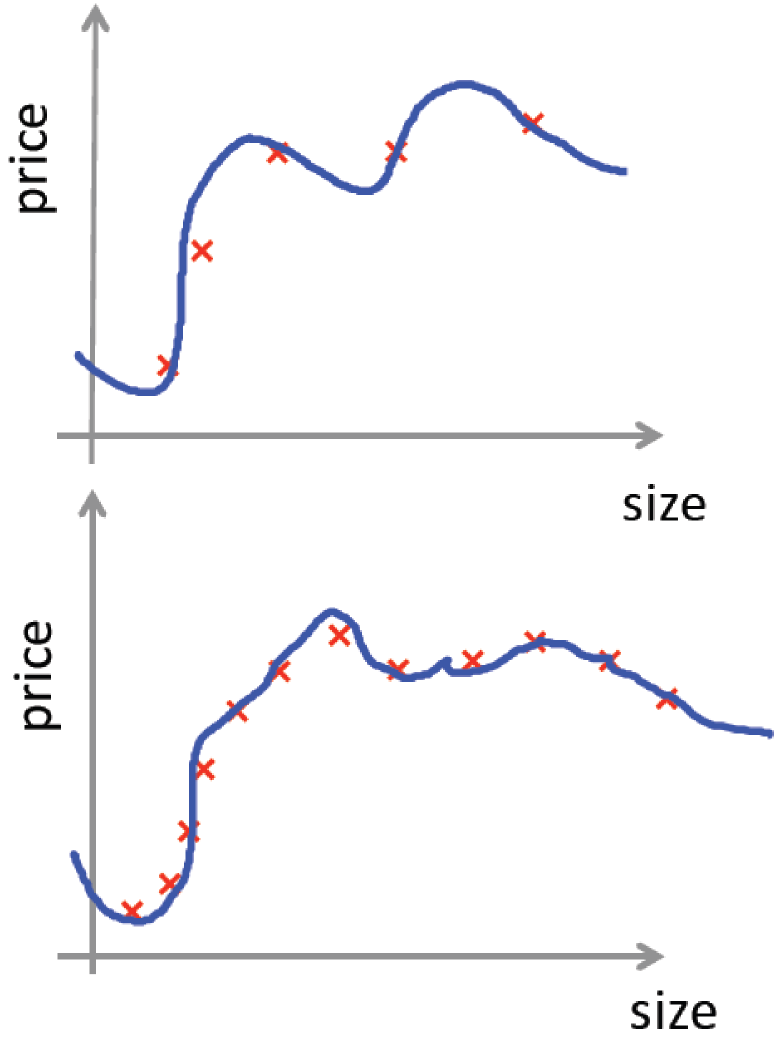
Overfitting: the model is too complex

Overfitting

High Variance

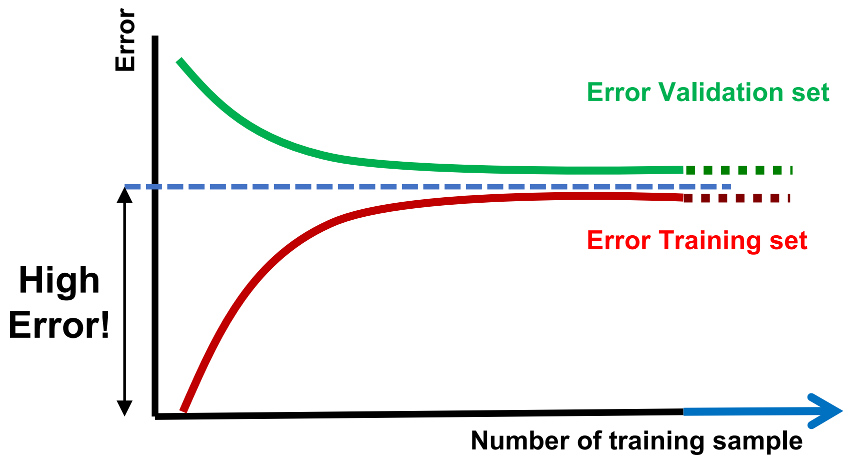


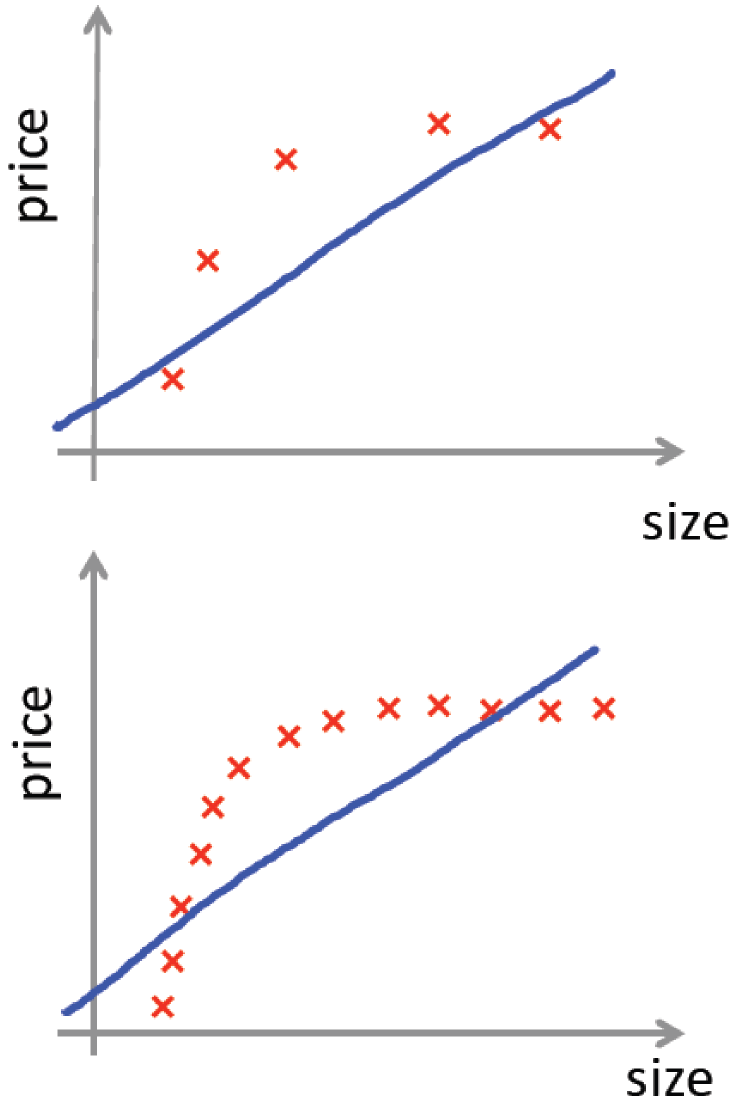
- Symptoms:
  - Larger gap between the two errors
  - Getting more training data is ***likely*** to help!



Underfitting

High bias





- Symptoms:
  - High error in the beginning
  - Getting more training data will **NOT** help!

What to try next

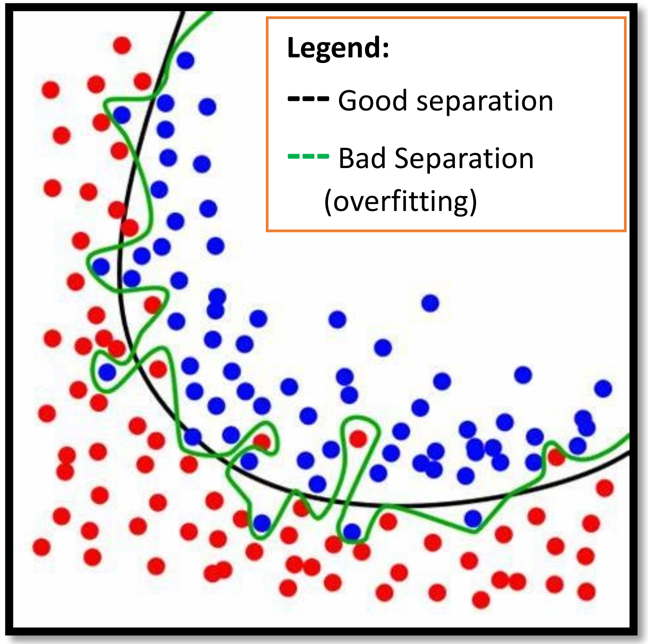
- **High Bias problem (underfitting)**
  - Try getting additional features
- **High Variance problem (overfitting)**
  - Get more training example
  - Try smaller sets of features

SPOILER ALERT!

Random Forest
  - *Small* number of trees
    - Fewer parameters
    - more prone to underfitting
  - *Large* number of trees
    - more parameters
    - more prone to overfitting

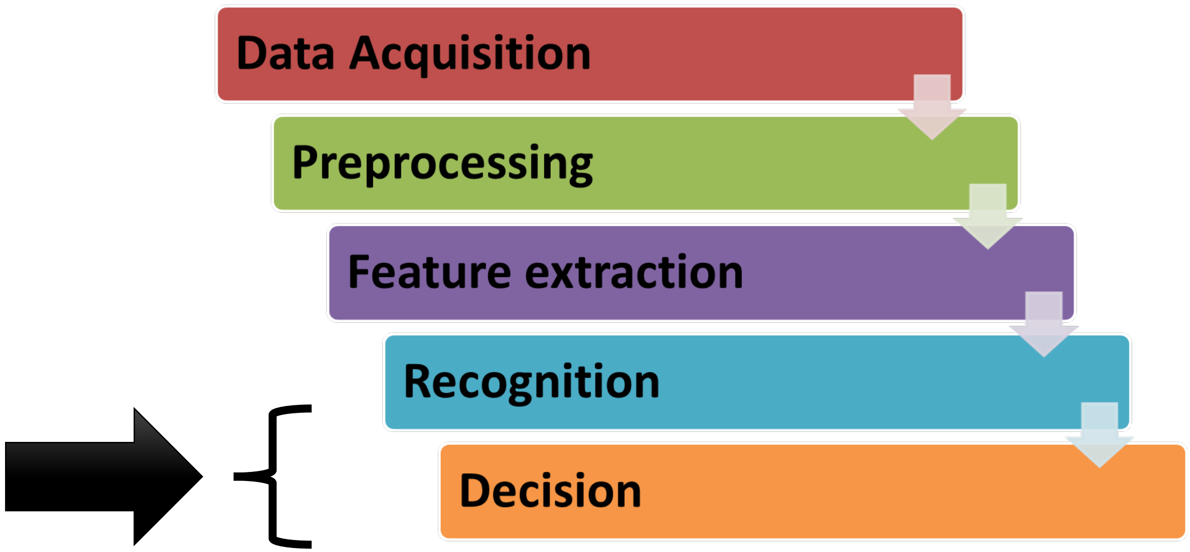
?

Considering K-NN:
k = 1, in general implies overfitting or underfitting?



Source: [Top Data Science Problems and How to Avoid Them - Just Understanding Data]

Performance indicators



Confusion matrix

Definition

Confusion Matrix

*"A confusion matrix is a specific table layout that allows **visualization** of the **performance** of an algorithm"*

https://en.wikipedia.org/wiki/Confusion_matrix

Confusion matrix

example

		Actual Class		
		Tuna	Codfish	Salmon
Predicted Class	Tuna	15	4	7
	Codfish	3	20	4
	Salmon	6	1	15
		24	25	26

Salmon		Actual Class	
		Salmon	Not Salmon
Predicted Class	Salmon	True Positive	False Positive *(Type I error)*
	Not Salmon	False Negative *(Type II error)*	True Negative

Confusion Matrix

True Positives (TP)

		Actual Class		
		Tuna	Codfish	Salmon
Predicted Class	Tuna	15	4	7
	Codfish	3	20	4
	Salmon	6	1	15

Salmon		Actual Class	
		Salmon	Not Salmon
Predicted Class	Salmon	True Positive 15	False Positive 7
	Not Salmon	False Negative 11	True Negative 42

Confusion Matrix

False Positives (FP)

		Actual Class		
		Tuna	Codfish	Salmon
Predicted Class	Tuna	15	4	7
	Codfish	3	20	4
	Salmon	6	1	15

Salmon		Actual Class	
		Salmon	Not Salmon
Predicted Class	Salmon	True Positive 15	False Positive 7
	Not Salmon	False Negative 11	True Negative 42

Confusion Matrix

False Negatives (FN)

		Actual Class		
		Tuna	Codfish	Salmon
Predicted Class	Tuna	15	4	7
	Codfish	3	20	4
	Salmon	6	1	15

Salmon		Actual Class	
		Salmon	Not Salmon
Predicted Class	Salmon	True Positive 15	False Positive 7
	Not Salmon	False Negative 11	True Negative 42

Confusion Matrix

True Negatives (TN)

		Actual Class		
		Tuna	Codfish	Salmon
Predicted Class	Tuna	15	4	7
	Codfish	3	20	4
	Salmon	6	1	15

Salmon		Actual Class	
		Salmon	Not Salmon
Predicted Class	Salmon	True Positive 15	False Positive 7
	Not Salmon	False Negative 11	True Negative 42

Accuracy and Precision

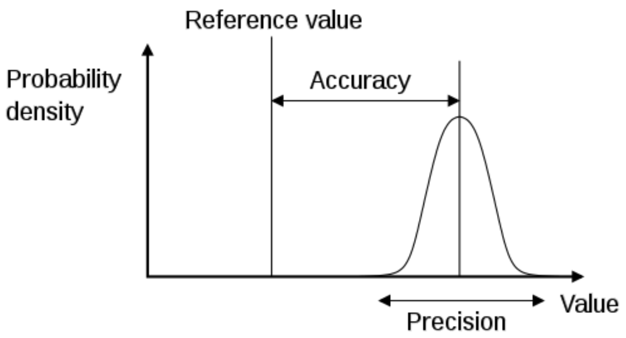
		Actual Class	
		True	False
Predicted Class	Predicted True	True Positive (TP)	False Positive (FP)
	Predicted False	False Negative (FN)	True Negative (TN)

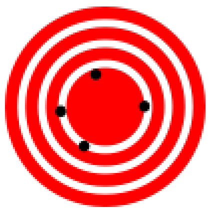
- **Accuracy** = the proportion of true results in the population

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

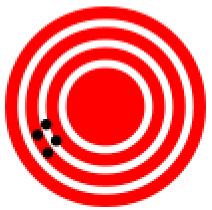
- **Precision** = the proportion of true positive results among what was predicted as positive

$$Precision = \frac{TP}{TP + FP}$$





Low accuracy due to low precision



Low accuracy even with high precision

Source: https://en.wikipedia.org/wiki/Accuracy_and_precision

Accuracy and Precision

Symmetry (binary classification)

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

		Actual Class	
		Men	Wom
Predicted Class	Men	10 TP	3 FP
	Wom	9 FN	9 TN

$$Precision = \frac{TP}{TP + FP}$$

		Actual Class	
		Men	Wom
Predicted Class	Men	10 TN	3 FN
	Wom	9 FP	9 TP

- **Take home message**
  - Accuracy is symmetric: $\text{Accuracy}_{\text{Men}} = \text{Accuracy}_{\text{Women}}$
  - Precision is **not** symmetric: $\text{Precision}_{\text{Men}} \neq \text{Precision}_{\text{Women}}$

Accuracy and Precision

Overall scores

- **Accuracy**
  - Accuracy = Overall Accuracy
  - Accuracy is class *independent*
  - **Attention**: this is true only in the 2 classes case

		Actual Class	
		Men	Wom
Predicted Class	Men	10	3
	Wom	9	9

- **Precision**
  - Precision ≠ Overall Precision
  - Precision is class *dependent*
    - For each class you get a different precision!!
  - **Overall Precision?**

$OverallPrecision = \frac{1}{N}\sum_{i=1}^{N} P_i$, where **N** is the number of classes

$WeightedPrecision = \frac{(P_{c1} * |c1|)+(P_{c2} * |c2|)}{|c1|+|c2|}$, where $|ci|$ is the number of instances in the class **i**

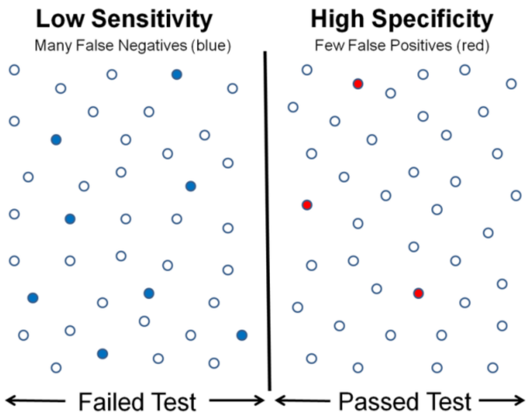
Sensitivity (or recall) and Specificity

Other performance indicators

- Example:
  - True positive: Sick people correctly diagnosed as sick
  - False positive: Healthy people incorrectly identified as sick
  - True negative: Healthy people correctly identified as healthy
  - False negative: Sick people incorrectly identified as healthy.

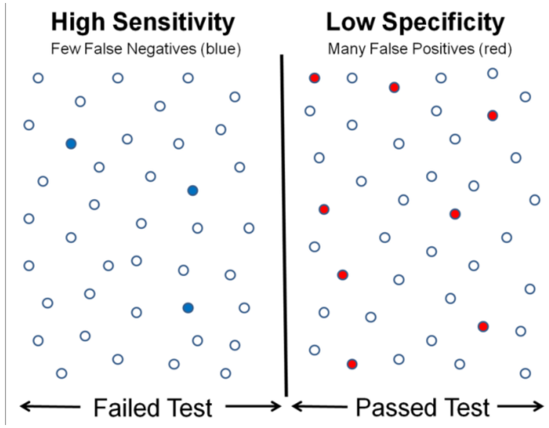
Sensitivity = the probability of a positive test given that the patient is ill

$$Sensitivity = \frac{TP}{TP + FN}$$



Specificity = the probability of a negative test given that the patient is well

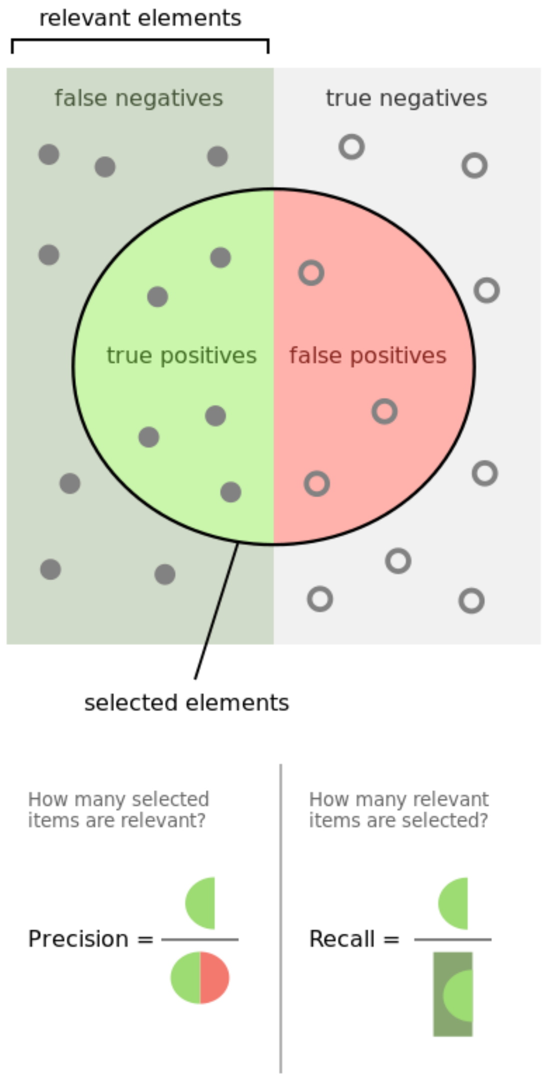
$$Specificity = \frac{TN}{TN + FP}$$



Precision and Recall

Other interpretations

- Information retrieval
  - **Precision** is the fraction of retrieved instances that are relevant
  - **Recall** is the fraction of relevant instances that are retrieved
- Probabilistic interpretation
  - **Precision** is the probability that a (randomly selected) retrieved document is relevant.
  - **Recall** is the probability that a (randomly selected) relevant document is retrieved in a search.



F-Score

Other performance indicators

- **F-score** (or F-measure or F1 score)
  - Combine precision and recall in one metric

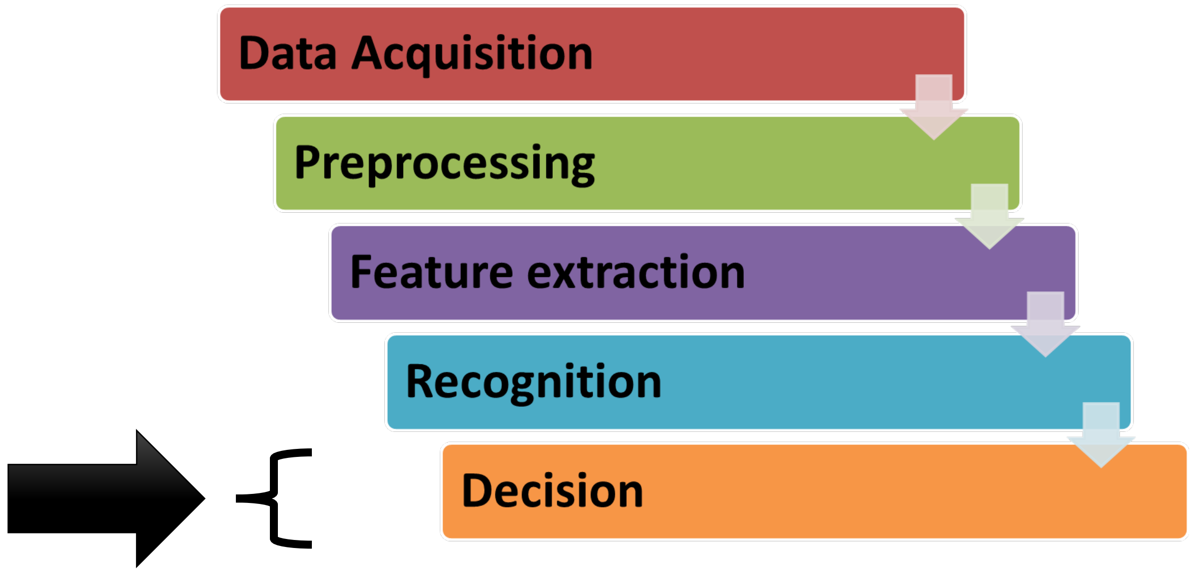
$$F = (1 + \beta) \cdot \frac{precision \cdot recall}{(\beta^2 \cdot precision) + recall}$$

  - The **most used** is the F1 score ($\beta = 1$)

$$F = 2 \cdot \frac{precision \cdot recall}{precision + recall}$$

Confidence Interval

(A gentle introduction)



Confidence Interval

- Motivations
  - A survey samples only a portion of the population.
  - There is **always** uncertainty about the results obtained.
  - This uncertainty is quantified by a **confidence interval**.

- Confidence interval
  - The confidence interval describes the precision of the estimation of a parameter (e.g., mean).
  - Assuming that the parameter to be estimated is in the confidence interval, it is unlikely, but not impossible, that the true value of the parameter is not in the confidence interval.

Confidence Interval

- Definition

$$\left[\bar{X} \pm \underbrace{z_{1-\alpha/2} \cdot \frac{\sigma}{\sqrt{n}}}_{\delta} \right]$$

z_p (quintile table)

$p = 0.90$	$z_p = 1.282$
$p = 0.95$	$z_p = 1.645$
$p = 0.975$	$z_p = 1.960$
$p = 0.99$	$z_p = 2.326$

- **n** is the number of observations
- X the calculated mean (e.g., mean accuracy)
- σ the standard deviation
- z_p the quintile corresponding to the degree of confidence (usually, you find it on a **table**)

- Result (e.g.): 90% ± 2% (i.e.: [88% - 92%]);

Confidence Interval

- Definition

$$\left[\bar{X} \pm \underbrace{z_{1-\alpha/2} \cdot \frac{\sigma}{\sqrt{n}}}_{\delta}\right]$$

z_p (quintile table)

$p = 0.90$	$z_p = 1.282$
$p = 0.95$	$z_p = 1.645$
$p = 0.975$	$z_p = 1.960$
$p = 0.99$	$z_p = 2.326$

- **n** is the number of observations
- X the calculated mean (e.g., mean accuracy)
- σ the standard deviation
- z_p the quintile corresponding to the degree of confidence (usually, you find it on a **table**)

- Result (e.g.): 90% ± 2% (i.e.: [88% - 92%]);

Confidence Interval

Take home message

- The confidence interval indicate the ***reliability*** of an estimate (e.g., the average accuracy, precision, etc. of a classifier).

- A **large** confidence interval is related to an **uncertain** estimate.

- **Increasing** the number of observations (**n**) **reduces the width** of the confidence interval

Conclusions

Your competences after this course

You shou be able to…

- **… discuss** the difference and uses of training set, validation set and test set
- **… explain** the meaning of having unbalanced classes in a training set?
- … **propose** ways to deal with an unbalanced classes in a training set problem?
- **… understand** when and why it is important to use Feature scaling & normalization
- **… implement** (K-fold) Cross-Validation and
  - understand its motivations, goals, mechanics (e.g., stratification) and limitations
- … **explain** and **compute** different performance indicators such as:
  - confusion matrix (True positives, False positives, True negatives, False negatives)
  - accuracy, precision, recall, F1 score
  - confidence interval (idea)

… diagnose Overfitting (High Variance) Vs Underfitting (High Bias)

A couple of questions for you!

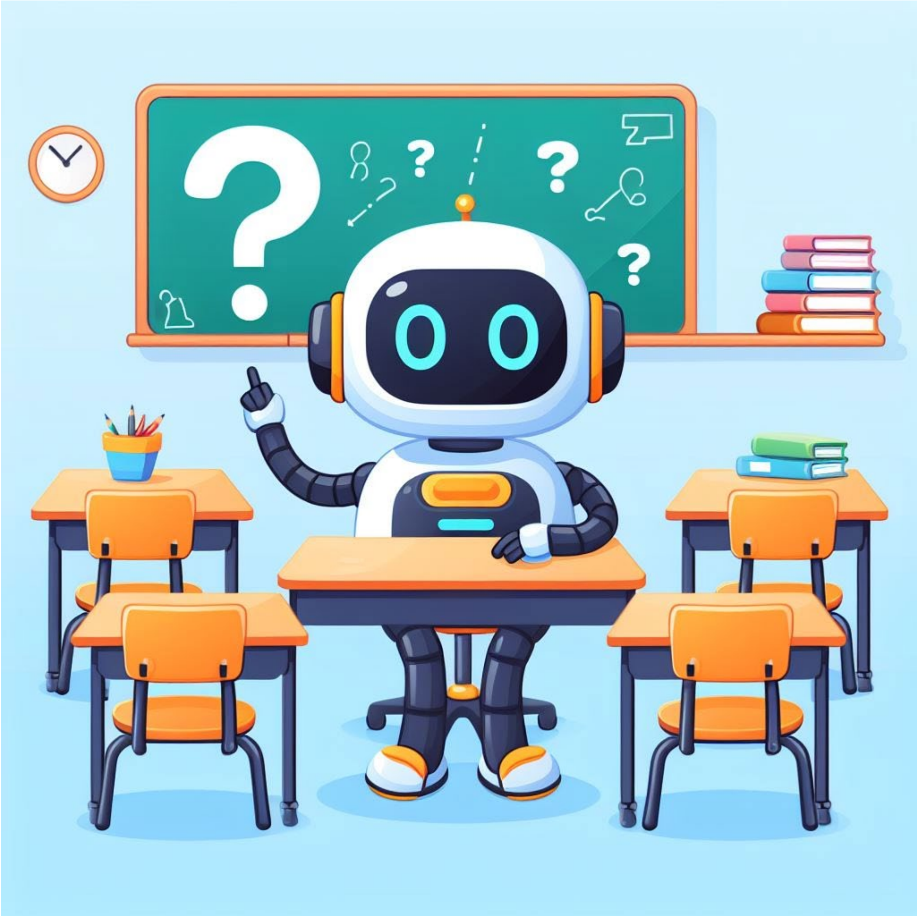
- Confusion matrix, accuracy, precision, etc. are metrics for classification only or they also work for regression?
  - Why and alternatives for regression?

- What are “ROC curve” and “AUC-ROC”?
  - When are these estimators interesting?

- What is the “**curse of dimesionality**” in ML?

Source: https://erikbern.com/2015/10/20/nearest-neighbors-and-vector-models-epilogue-curse-of-dimensionality

Any question?



Prompt: “picture of a cute robot in a class asking questions”

References

- [1] **Cross-Validation**. Payam Refaeilzadeh, Lei Tang, Huan Liu, Arizona State University
- [2] **Estimating the error rate of a prediction rule: improvement on cross-validation**. Efron B. J. Am. Stat. Assoc., 78:316–331,1983.
- [3] Approximate statistical tests for comparing supervised classification learning algorithms. Dietterich T.G. Neural Comput., 10(7):1895–1923
- [4] International vocabulary of metrology — Basic and general concepts and associated terms
- [5] An Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements. John Robert Taylor. University Science Books. pp. 128–129
- [6] Assessing the accuracy of prediction algorithms for classification: an overview. Baldi, P.; Brunak, S.; Chauvin, Y.; Andersen, C. A. F.; Nielsen, H. Bioinformatics 2000 000, 16, 412–424
- [7] Coursera, Machine learning, Andrew Ng, Stanford University, https://www.coursera.org/course/ml
- [8] http://scott.fortmann-roe.com/docs/MeasuringError.html