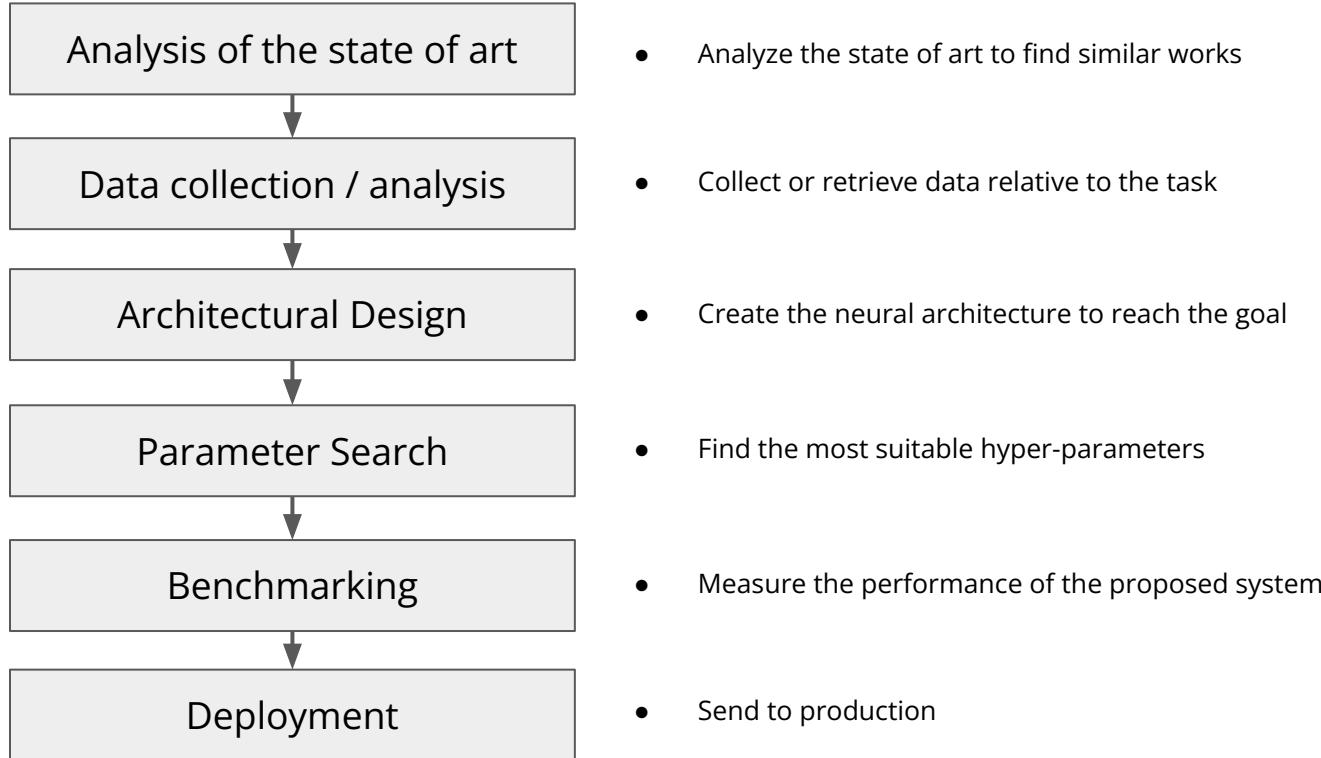


Università degli
Studi di Milano-Bicocca

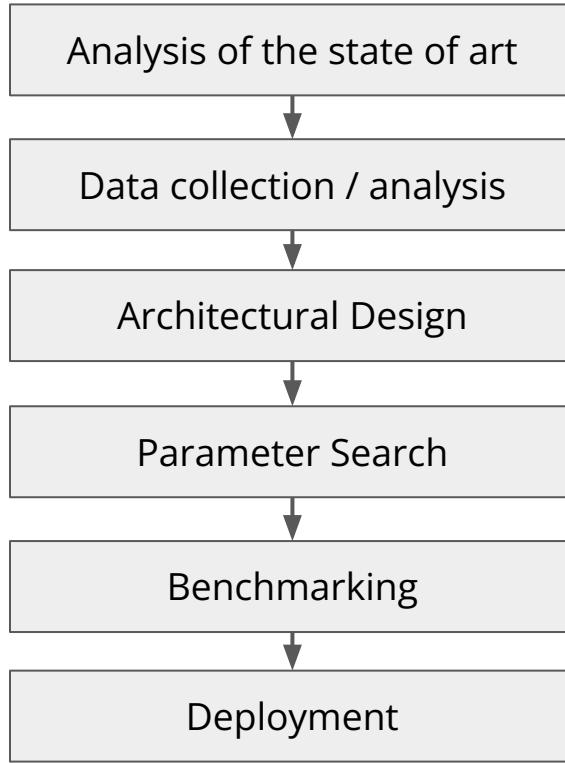
Data collection & analysis

Prof. Flavio Piccoli, Dr. Mirko Paolo Barbato

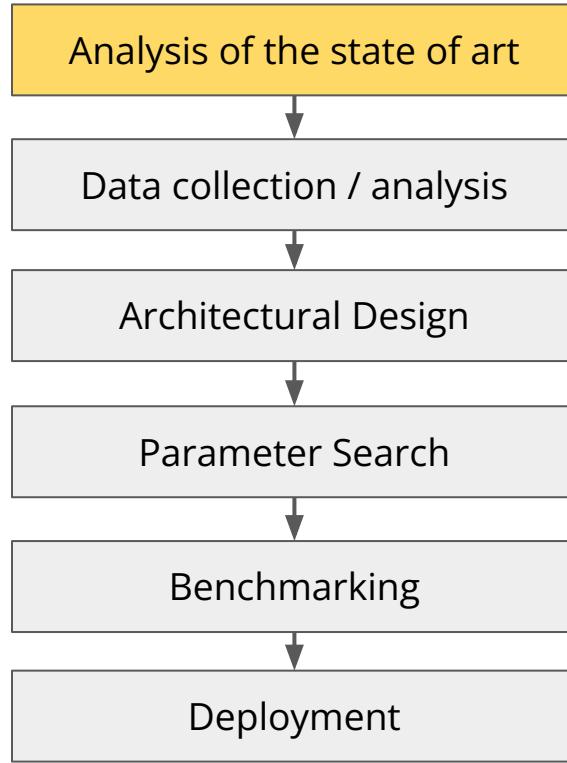
R&D process



R&D process

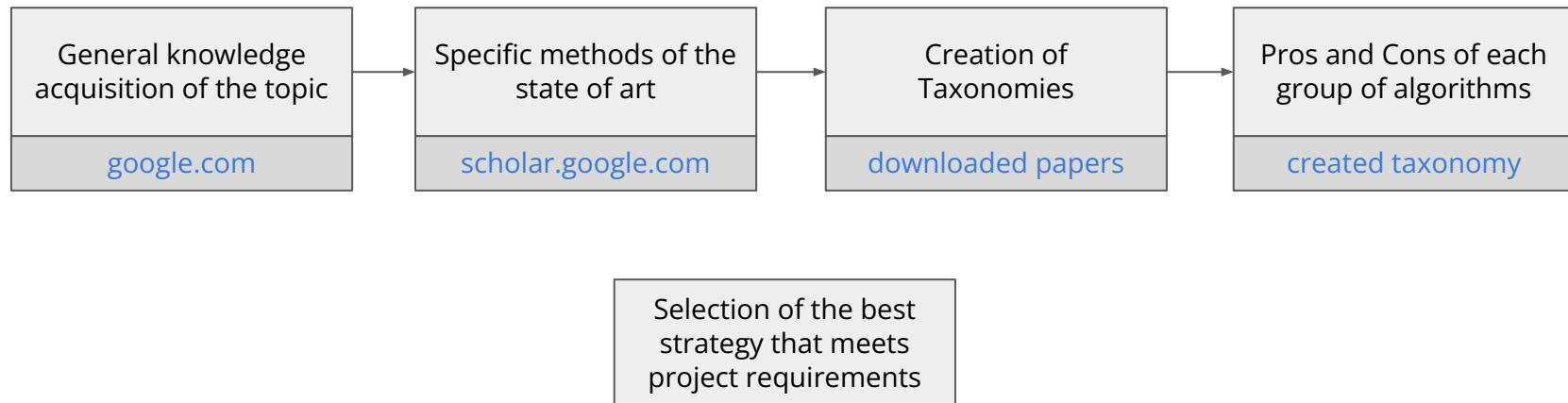


R&D process



Analysis of the state of art

- Before starting to design a method, it is very important to analyze the state of art to find:
 - **methods** doing the same task
 - **benchmarks** to compare your method with the state of art
 - **metrics** for assessing your method
- Two important tools:
 - Google and
 - Google Scholar <https://scholar.google.it>



Google Scholar

- Search engine for scientific papers

The screenshot shows a Google Scholar search results page for the query "object detection". The search bar at the top contains the query. Below it, a sidebar on the left provides filtering options: "In qualsiasi momento" (set to "Dal 2019"), "Ordina per pertinenza", "Ordina per data", "Qualsiasi lingua" (set to "Pagine in Italiano"), and "Qualsiasi tipo" (set to "Articoli scientifici"). The main content area displays three search results:

- Object detection**
Y Amit, P Felzenszwalb, R Girshick - Computer Vision: A Reference Guide, 2020 - Springer
... **object detection** is to detect all instances of **objects** from one or several known classes, such as people, cars, or faces in an image. Typically only a small number of **objects** ... **detection** is ...
☆ Salva 99 Cita Citato da 58 Articoli correlati Tutte e 7 le versioni
- [HTML] Salient object detection: A survey**
A Borji, MM Cheng, Q Hou, H Wang, J Li - Computational visual media, 2019 - Springer
... **object detection** and situate this field among other closely related areas such as generic scene segmentation, **object** ... and evaluation metrics for salient **object detection**. We also discuss ...
☆ Salva 99 Cita Citato da 778 Articoli correlati Tutte e 14 le versioni Web of Science: 246
- Object detection in 20 years: A survey**
Z Zou, Z Shi, Y Guo, J Ye - arXiv preprint arXiv:1905.05055, 2019 - arxiv.org
... Abstract—**Object detection**, as of one the most fundamental ... If we think of today's **object detection** as a technical aesthetics ... reviews 400+ papers of **object detection** in the light of its ...
☆ Salva 99 Cita Citato da 1038 Articoli correlati Tutte e 3 le versioni

On the right side of the results, there are links to "[PDF] researchgate.net", "[HTML] springer.com", and "[PDF] arxiv.org". At the bottom right of the sidebar, there is a link to "Full Text Bicocca".

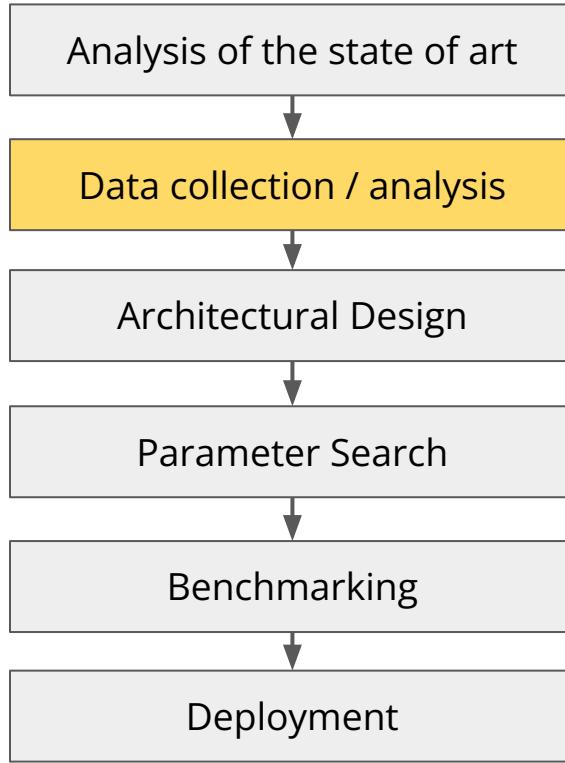
It is useful to select newest methods



once you find an interesting paper, read also citing literature



R&D process



Data collection and analysis

- Data is the most important resource for a data scientist
- Two types of datasets:
 - benchmarks in the state of art
 - custom datasets

Dataset type	Pros	Cons
Sota benchmark	<ol style="list-style-type: none">1. comparison with sota methods2. reliable results3. huge amount of data available	<ol style="list-style-type: none">1. data may not be replicable in real life2. model may not work
Custom benchmark	<ol style="list-style-type: none">1. same data of the final application2. full knowledge of the dataset	<ol style="list-style-type: none">1. comparison with sota methods may not be replicable2. creation is extremely hard and expensive

Types of data

A generic input sample can be:

- **adimensional**: the quantity is a single number
- **monodimensional**: the quantity is a signal
- **bidimensional-tridimensional**: the quantity is an image
- **quadri-dimensional**: the quantity is a sequence of images



Adimensional data

It is a scalar representing a measure of a phenomenon or a class of belonging of the sample

Label

- indicates that the sample belongs to a specific class
- e.g. if associated to audio, can indicate the sex of the speaker

Measure

- it's a value that measures a quantity
 - can belong to a specific interval
 - can be an unlimited value

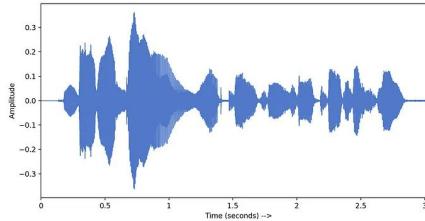


Digital signal data

In data science, it's the discrete representation of a wave carrying information.

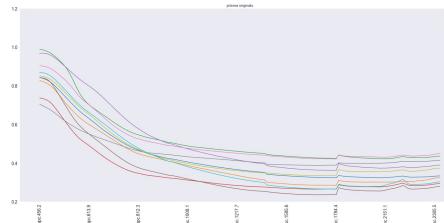
It can be:

- **time dependent**
 - the signal represents a quantity varying in the time



Audio = amplitude varying along time

- **time independent**
 - the signal represents several quantities dependent from each other, measured in a specific time point



hyperspectral signal = wide spectrum of light



Images

Single feature image

Single-channel image depicting one feature



Graylevel image



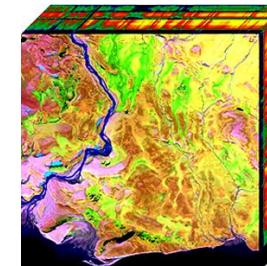
Thermal photo

Multiple feature image

Multiple channel image depicting multiple features with spatial relationship



RGB image



Hyperspectral image

Types of ground-truths

- Categorical label

cat	dog	frog
-----	-----	------

- Scalar feature

5.3 2 False

- Image level feature



Existing tools & frameworks for data collection

- If your project requires custom data, you have several options:
 - collect unlabelled data and manually tag data
 - collect unlabelled data and outsource the tagging phase (expensive)



Comparison in terms of **content visibility**: LEFT is better No difference RIGHT is better

Comparison in terms of **image artifacts**: LEFT is better No difference RIGHT is better

Comparison in terms of **image color**: LEFT is better No difference RIGHT is better

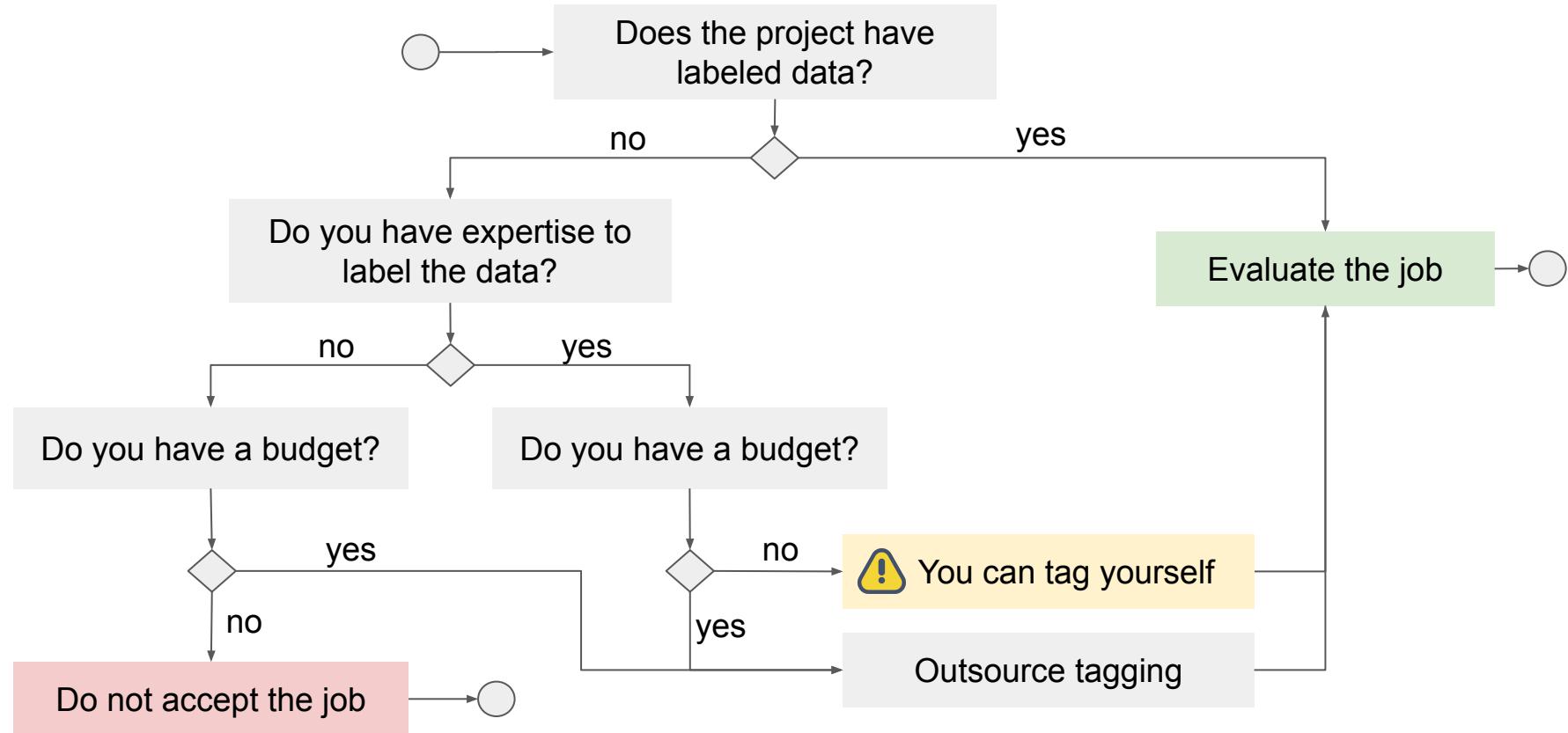
Comparison in terms of **image contrast**: LEFT is better No difference RIGHT is better

Comparison in terms of **overall quality**: LEFT is better No difference RIGHT is better

Free comment (optional):

Submit

Feasibility evaluation



Data preprocessing

Cleaning



Deal with missing data

Encoding



Convert data in a machine-learning suitable format

Normalizing



Normalize the variables

Splitting



Split in train / validation / test



Cleaning

What do we do with missing data?

Three possible strategies:

- discard feature having missing data
- discard samples having missing data
- substitute missing data with plausible content
 - **booleans / categorical**: replace with mode
 - **integers**: replace with median
 - **floats**: interpolate



Encoding

- Machine learning models can only work with numerical values
- It is necessary to transform the categorical values of the relevant features into numerical ones

One-hot encoding (for input variables)

brand
Fiat
BMW
Lamborghini
Fiat

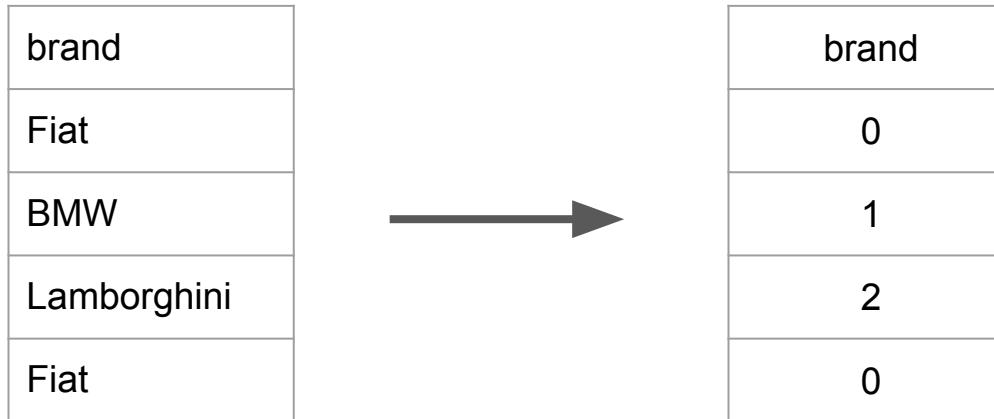


brand_fiat	brand_bmw	brand_lambo
1	0	0
0	1	0
0	0	1
1	0	0

Encoding

- Machine learning models can only work with numerical values
- It is necessary to transform the categorical values of the relevant features into numerical ones

Labeling (for estimated variables)



Normalizing

- The range of the variables affect their importance
- We need to normalize them so that each variable resides in the same range
 - **min - max normalization**
 - if the variable under analysis has a specific range, it's possible to use this normalization
 - `from sklearn.preprocessing import MinMaxScaler`
 - **standardization**
 - if the range is unknown a priori
 - sets the mean to 0 and the variance to 1
 - `sklearn.preprocessing import StandardScaler`



Outlier detection - elliptic envelope

- assumes that all points follow a Gaussian distribution
- outlier = point that is not well fit by the Gaussians

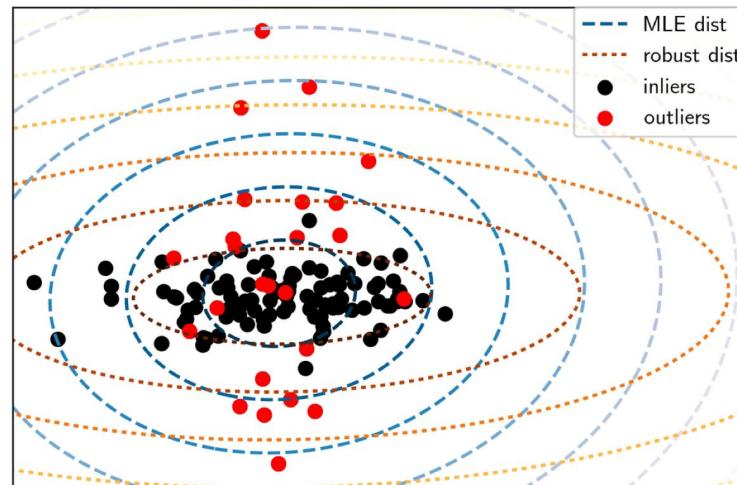
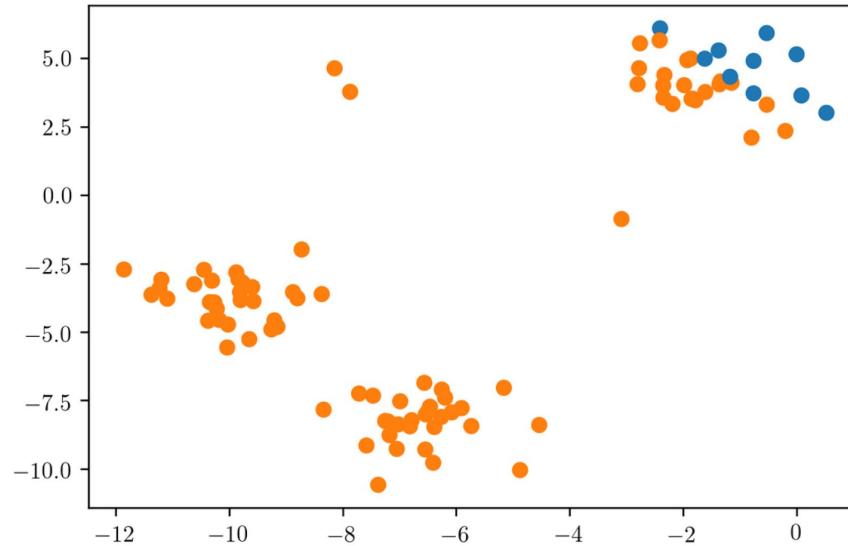


image courtesy: <https://amueller.github.io/aml/03-unsupervised-learning/03-outlier-detection.html>

Outlier detection - elliptic envelope

- What if the samples do not follow a gaussian distribution?



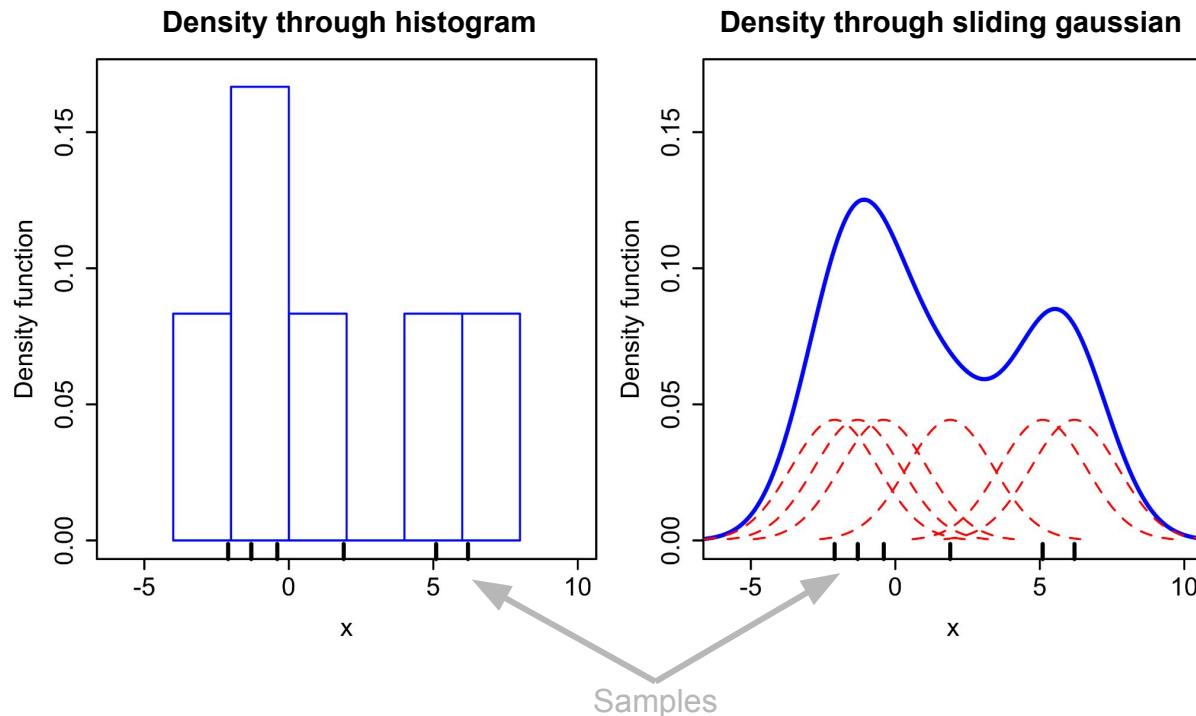
It fails!

- What about gaussian mixture models? → no, because they will try to fit all of the data



Outlier detection - Kernel density estimation

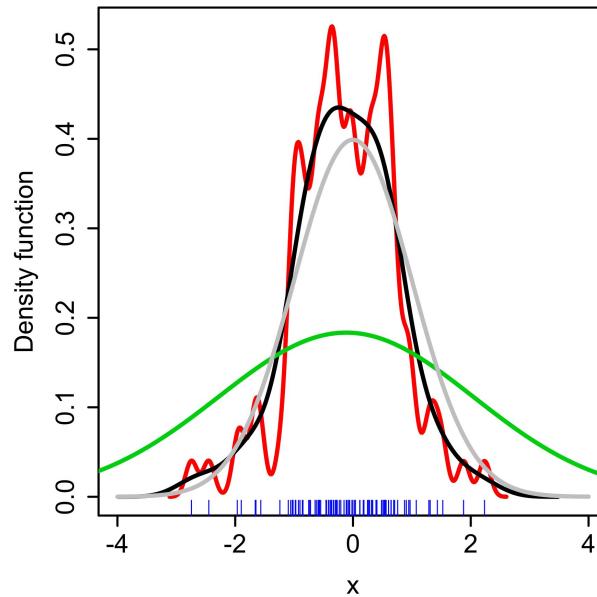
- Another approach is to study the density of the parameters and mark as outliers the ones with lower repr.



Outlier detection - Kernel density estimation

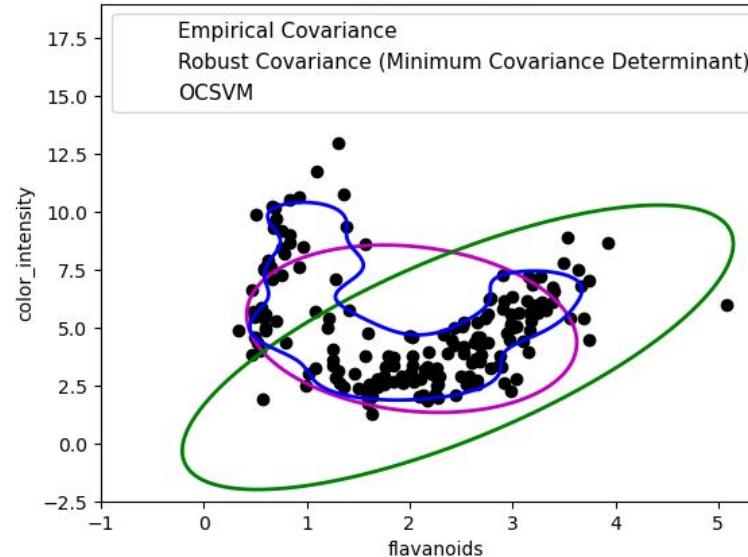
Problems:

- hard to find the correct size of the windowing function (a.k.a. bandwidth)
- outliers affect the estimation
- does not scale well on higher dimensional data



Outlier detection - one class SVM

- uses gaussian kernel to cover data
- makes use of support points (so it does not use all of them)
- Specify outlier ratio through parameter *nu* (a.k.a. *contamination*)

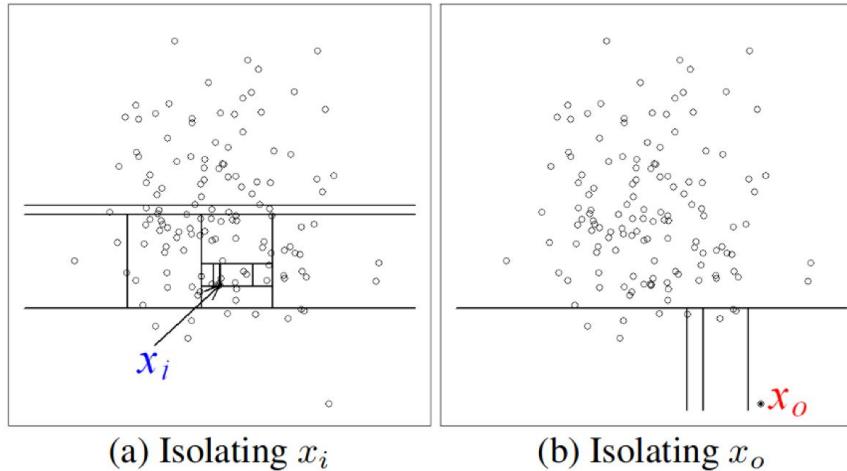


- here again there is a parameter..

Outlier detection - isolation forests

idea:

- build complete random trees
- for each tree, randomly select a feature and randomly select a split value between min and max
- **idea:** the deeper we go in the tree, the more dense is that point
- **robustness:** average over a forest



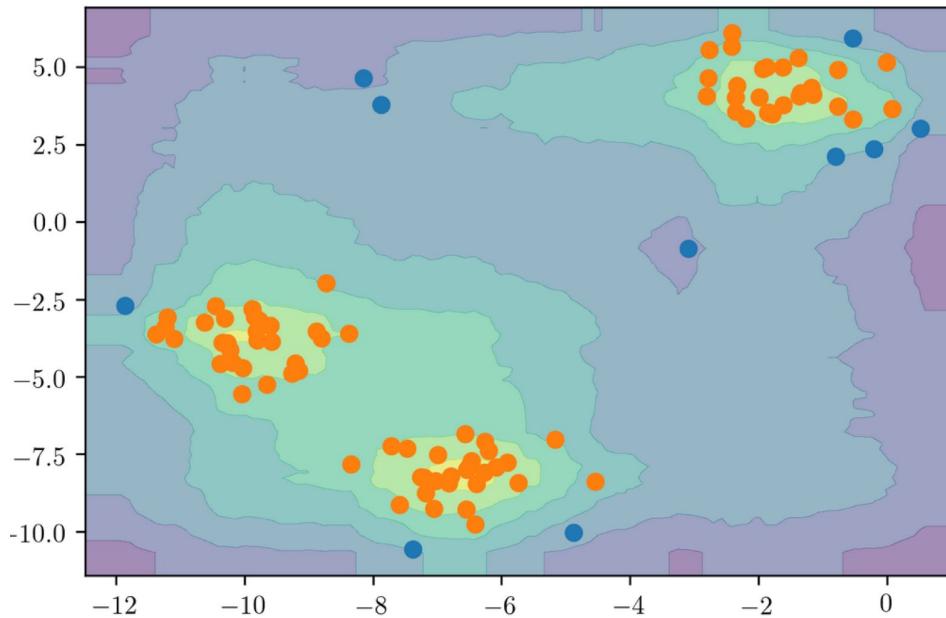
x_i is a normal sample
it is not easy to isolate it

x_o is a normal sample
it is very easy to isolate it

- this method does not require any parameter!

Outlier detection - isolation forests

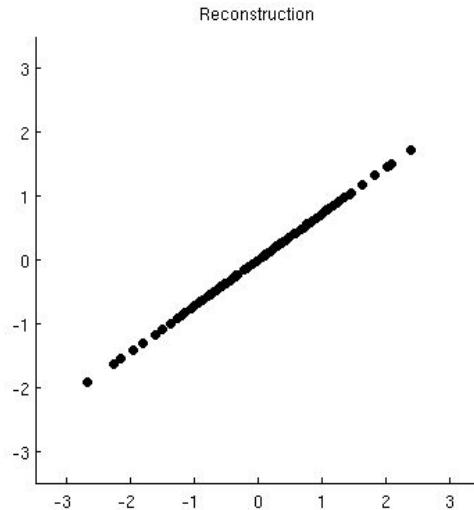
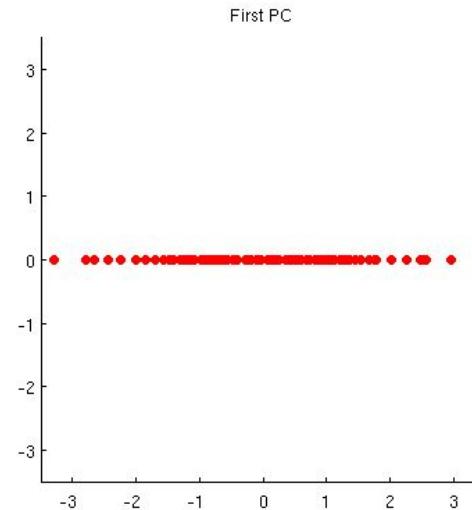
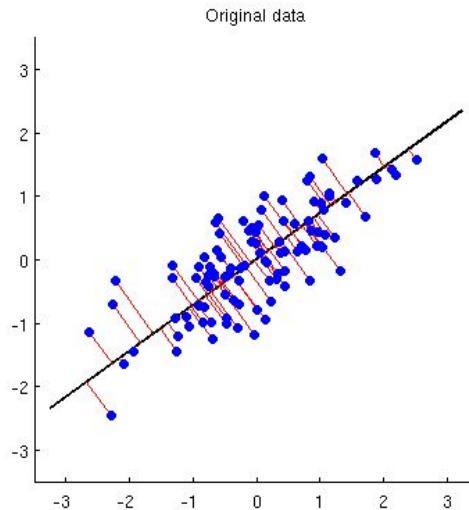
- Here there is a visual representation of the isolation through iso-curves on the two features



Outlier detection - PCA

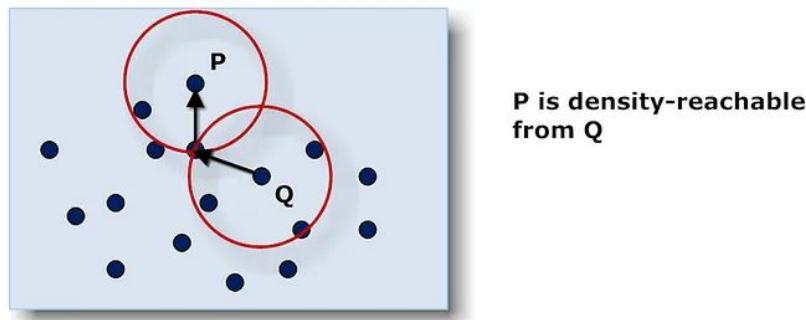
idea:

- compute PCA
- drop less important components
- reconstruct the samples
- outliers = points having the highest reconstruction error (because they do not follow the distribution)



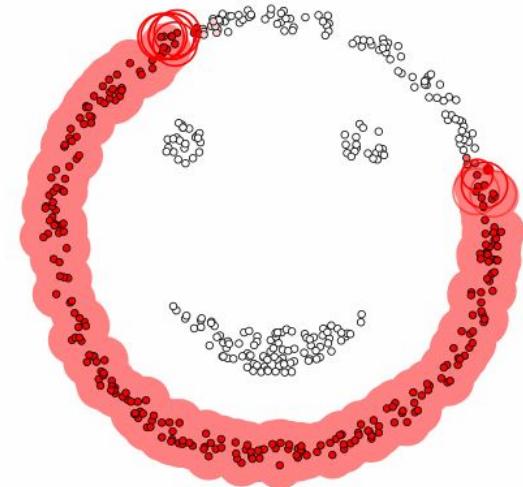
DBSCAN - Density-Based Spatial Clustering of Applications with Noise

- identifies distinctive clusters in the data
- based on the key idea that a cluster is a group of high data point density
- Two parameters:
 - **minPts**: the minimum number of points clustered together for a region to be considered dense
 - **eps**: a distance measure
- two concepts:
 - **density reachability**: a point to be reachable from another if it lies within a particular distance from it
 - **density connectivity**: transitivity in connection: if $a \rightarrow b$ and $b \rightarrow c$, then $a \rightarrow c$



DBSCAN - Density-Based Spatial Clustering of Applications with Noise

1. initially selects random data point
2. all points within distance **eps** are considered neighbors
3. if the number of neighbors is less than **minPts**, these points are considered noise
4. else, all points within eps distance are added until it is not possible to do it
5. When finished, restart from point 1



Case study on outlier detection

The dataset used for this experiment

THFOOD50

It is a dataset for Thai food classification.

Class 1 - BitterMelonSoup



Class 2 - BooPadPongali



- 50 classes
- 15679 images
 - Train 14086
 - Test / val 1593

Which outliers do we want to find?

Mislabel



Image belonging to the dataset but having a wrong ground truth (e.g. a pear marked as an apple)

Domain error



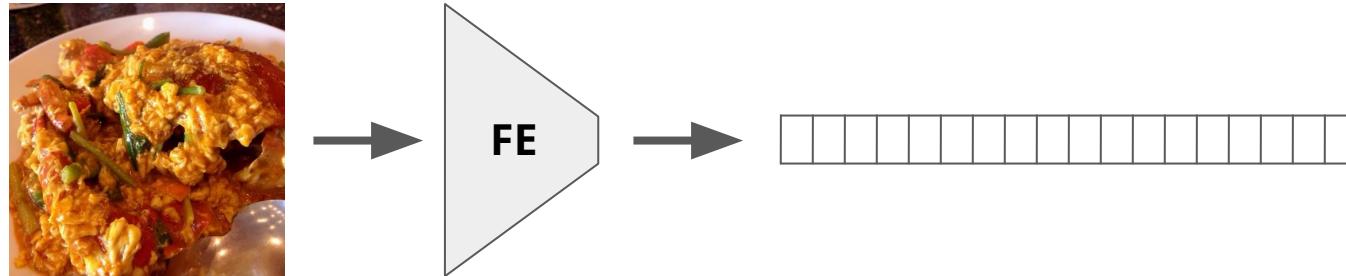
Image that does not belong to the domain under consideration (in this case, non-food image)

In this investigation, outliers have been artificially created



Experimental setup

Suppose we have already a feature extractor (FE) that converts an image in a meaningful feature vector



Convert all samples of the dataset in vectors. Inject outliers and perform outlier detection, class by class.

Samples of class 1

sample #1

sample #1																
sample #2																
sample #3																
sample #4																

sample #2

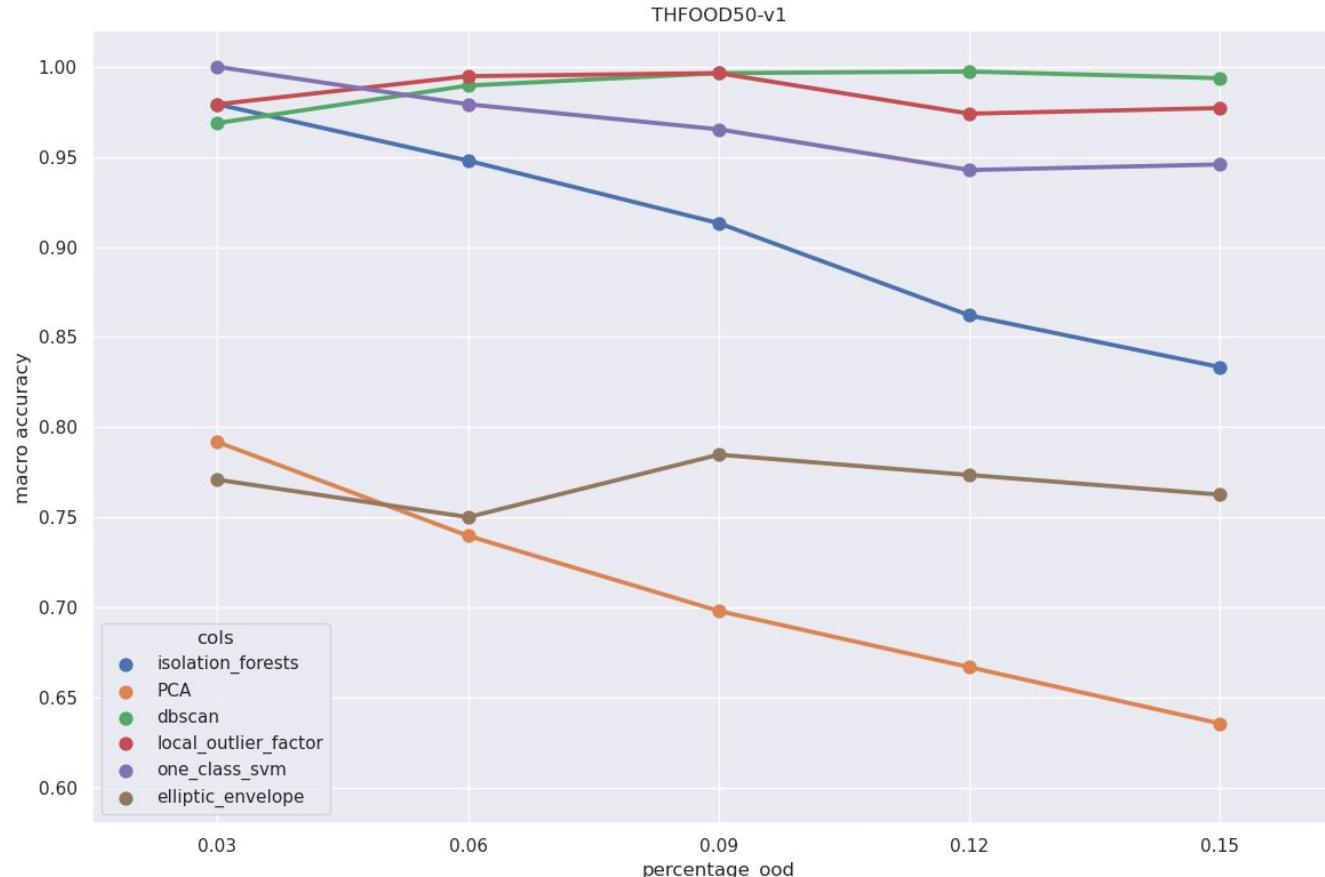
sample #3

sample #4

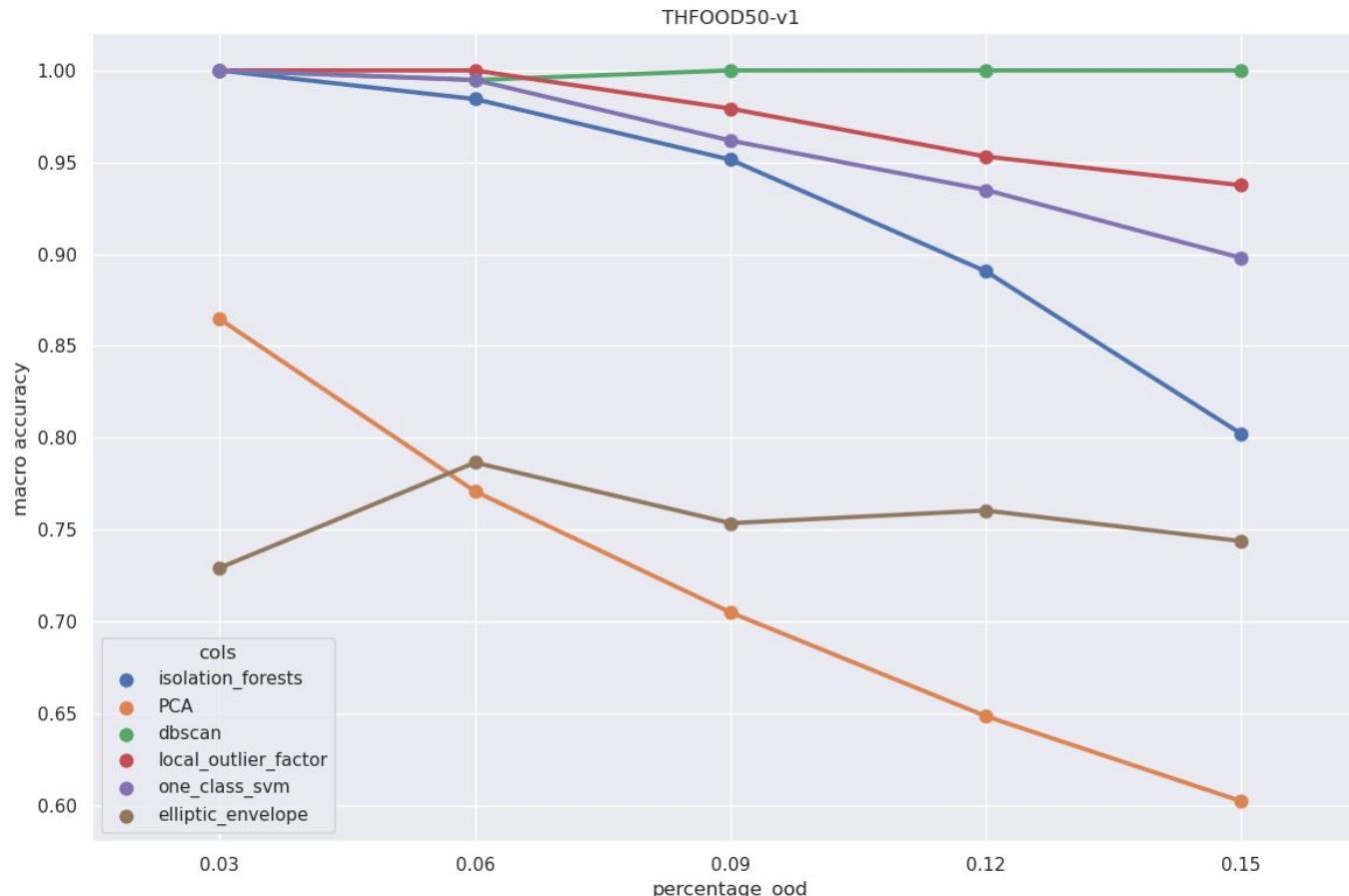
gt for mislabels

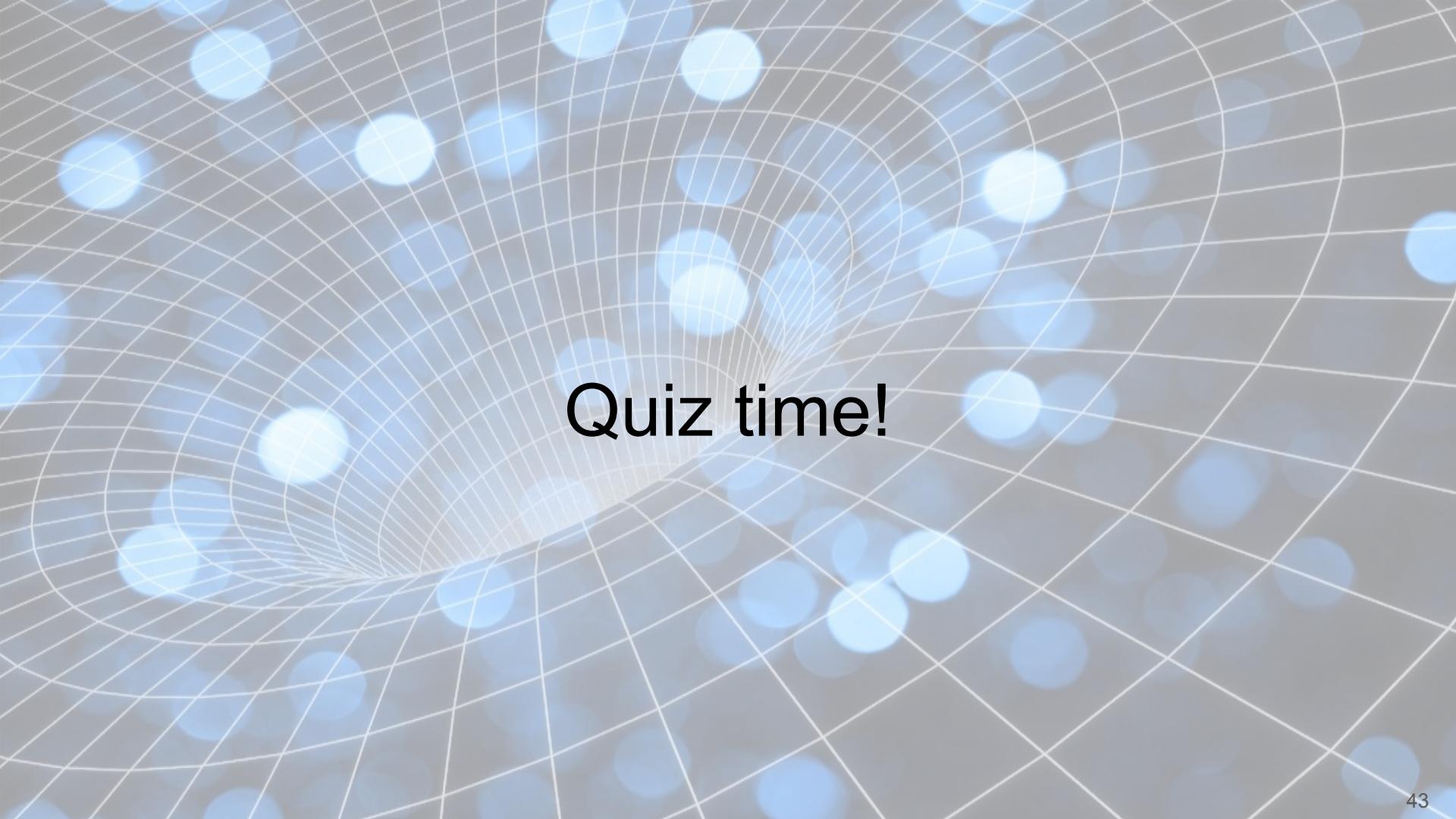
gt for domain errors

Results on mislabeling



Results on domain error (external images)





Quiz time!

Quiz time!

Go to the website:

PollEv.com/flaviopiccoli014



What is the most important research engine for researchers and data scientists?

Pandas

Google Scholar

Stackoverflow

Google



What is the most important advantage of using a benchmark in the state of art?

it is already pre-processed
correctly

it is required by the conduct
of honor of researchers

Allows the comparison
with sota methods

it has huge amount of data



What is the biggest downside of using a benchmark of the state of art?

The use can be hard and expensive

model trained on sota benchmark
hardly generalizes on project data

it may have pitfalls that are
unknown

model trained on sota benchmark
can be unreal



Which of the following is not a categorical quantity

Age group

Sex

Height

Animal type



Which of the following is not a time-dependent signal?

Stock price of Apple

Heart rate

hyperspectral signal

Audio amplitude



What is the correct order of data preprocessing?

Cleaning - Normalizing
- Encoding - Splitting

Encoding - Splitting -
Cleaning - Normalizing

Cleaning - Encoding -
Normalizing - Splitting

Splitting - Cleaning -
Encoding - Normalizing



Why is encoding necessary?

Because machine-learning models can not work with numerical values

Because machine-learning models can overfit on non-numerical values

Because machine-learning models can only work with numerical values

Because machine-learning models use complex features to perform predictions



Which one is an important method for outlier detection?

Isolation forests

Random forests

Multi-class SVM

Encoding step

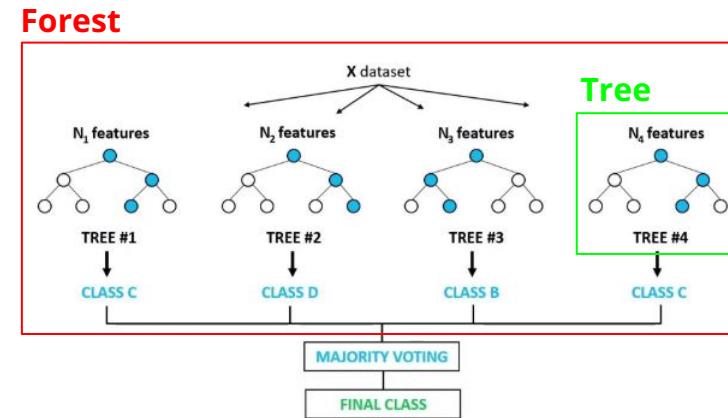


Random forests

Random Forests

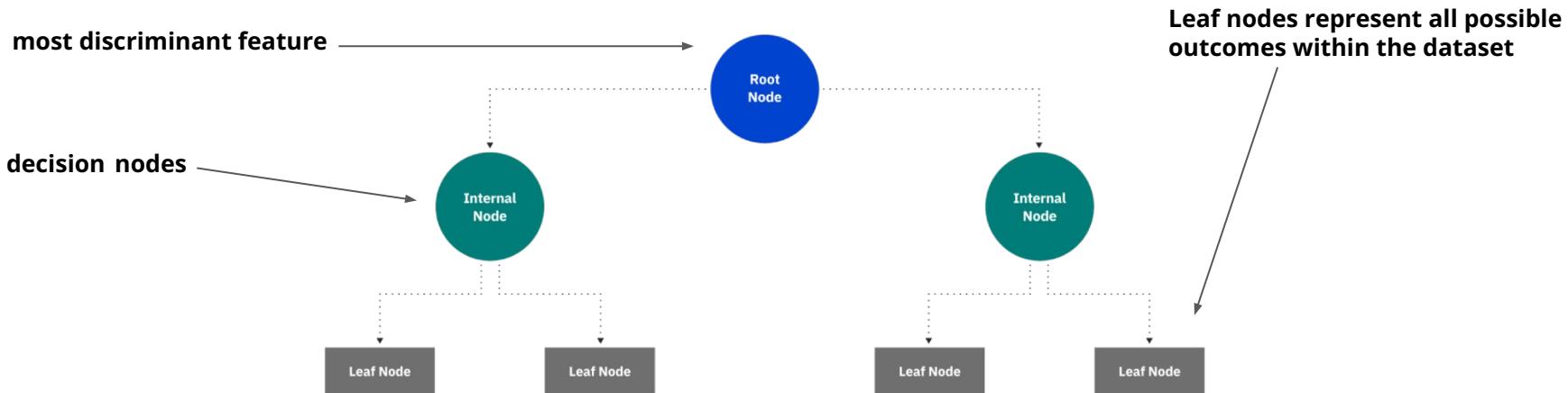
- it is one of the most popular tree-based supervised learning algorithms
- It is also the most flexible and easy to use
- Easy to backtrace features that mostly contributed to the final output

1. Select random samples from a given data or training set.
2. This algorithm will construct a decision tree for every training data.
3. Voting will take place by averaging the decision tree.
4. Finally, select the most voted prediction result as the final prediction result.



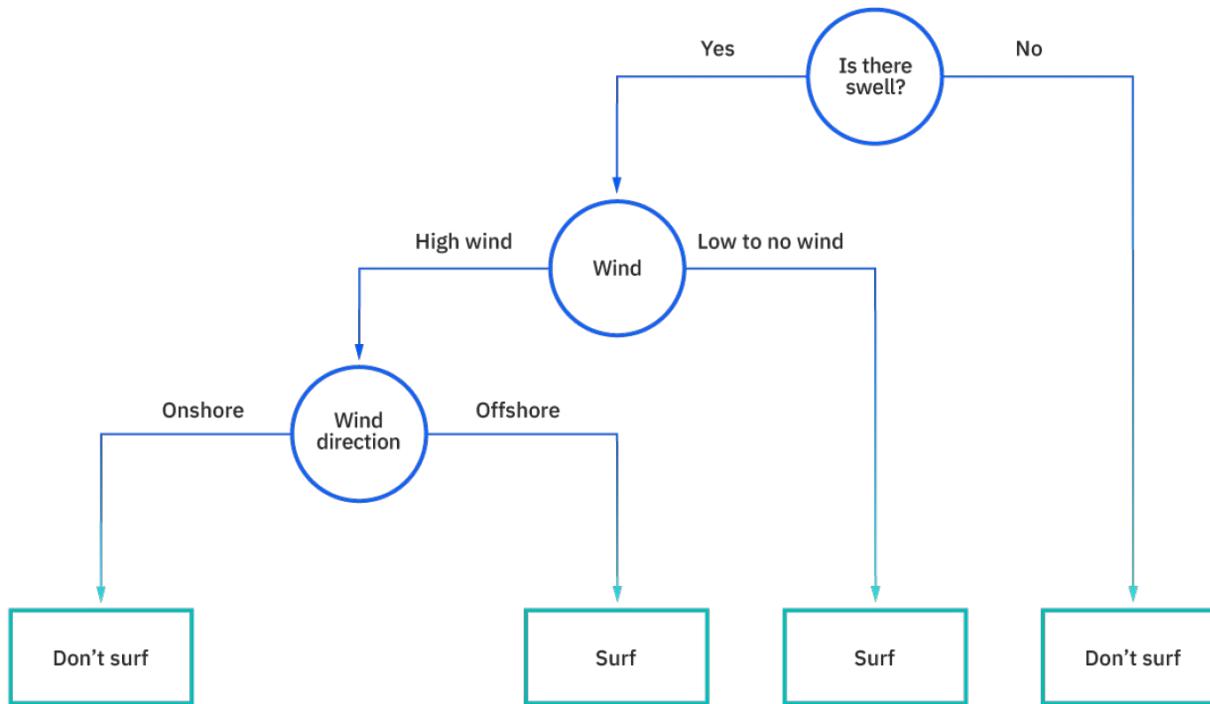
Decision tree

- it is the basis classifier of random forests
- non-parametric supervised learning algorithm
- hierarchical tree structure
- it has one root node, branches, internal nodes and leaf nodes



Decision tree

- An example



Decision tree

- Base idea: most important nodes must be up
 - importance = how much they split sample data (maximum: 50% - 50%)
- Several methods exist for creating the tree:
 - **ID3 (Iterative Dichotomiser 3)**: uses entropy and gain in information to evaluate nodes
 - **C4.5**: evolution of ID3. Uses gain in information to evaluate nodes
 - **CART (Classification And Regression Trees)**: uses Gini impurity method to identify the nodes
- Entropy: measures the impurity of the values of a sample

$$E(S) = - \sum_{c \in C} p(c) \log 2p(c)$$

S = set of data

c = classes in the set

p(c) = portion of the samples that belong to class c w.r.t. the total number of points in S

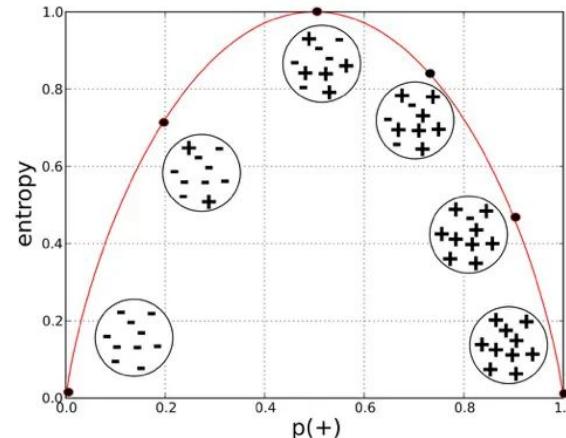


Entropy

- Entropy: measures the impurity of the values of a sample

$$E(S) = - \sum_{c \in C} p(c) \log 2p(c)$$

- Entropy values are between 0 and 1
- If all samples in S belong to a class, $E = 0$
- if half samples belong to a class and the other to another class, $E = 1$
- information gain represents the difference in entropy before and after a division based on a given attribute



S = set of data

c = classes in the set

$p(c)$ = portion of the samples that belong to class c w.r.t. the total # of points in S

Information gain

- Information gain represents the difference in entropy before and after a division based on a given attribute
- Measures the reduction of uncertainty given an additional piece of information

$$IG(S, a) = E(S) - \sum_{v \in a} \frac{|S_v|}{|S|} E(S_v)$$

Information gain if we split
the data in two sets
according to a specific value
of attribute a

Entropy of the dataset

For each value v of the attribute a, compute
entropy. Then, sum all entropies.

a = specific attribute or class label

E(S) = entropy of data set S

$|S_v| / |S|$ = proportion of values in S_v times the number of values in the set

- The attribute with the greatest information gain will produce the best subdivision

Decision tree

A simple example

- Forecast playing of tennis given env. vars
- Total samples = 14
- yes tennis = 9
- no tennis = 5

Day	Outlook	Temp	Humidity	Wind	Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Weak	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cold	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

$$E(Tennis) = -\frac{9}{14} \log_2 \frac{9}{14} - \frac{5}{14} \log_2 \frac{5}{14} = 0.94$$

yes tennis no tennis

Decision tree

we can then compute information gain on any attribute

For example, **Humidity**

- Total samples = 14
- high humidity = 7
- normal humidity = 7

Contingency table:

		Tennis		
		Yes	No	Tot
Hum.	High	3	4	7
	Normal	6	1	7

Day	Outlook	Temp	Humidity	Wind	Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Weak	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cold	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

$$E(Tennis|Humidity = \text{high}) = -\frac{3}{7} \log_2 \frac{3}{7} - \frac{4}{7} \log_2 \frac{4}{7} = -0.985$$

$$E(Tennis|Humidity = \text{norm}) = -\frac{6}{7} \log_2 \frac{6}{7} - \frac{1}{7} \log_2 \frac{1}{7} = -0.59$$



yes tennis no tennis

Decision tree

we can then compute information gain on any attribute

For example, **Humidity**

- Total samples = 14
- high humidity = 7
- normal humidity = 7

$$E(Tennis|Humidity = \text{high}) = 0.985$$

$$E(Tennis|Humidity = \text{norm}) = 0.59$$

Day	Outlook	Temp	Humidity	Wind	Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Weak	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cold	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

$$E(Tennis|Humidity) = -\frac{7}{14} * 0.985 - \frac{7}{14} * 0.592 = -0.7885$$



$$E(Tennis|Humidity = \text{high})$$



$$E(Tennis|Humidity = \text{norm})$$



Decision tree

we can then compute information gain on any attribute

For example, **Humidity**

- Total samples = 14
- high humidity = 7
- normal humidity = 7

$$E(Tennis) = 0.94$$

$$E(Tennis|Humidity) = 0.7885$$

Day	Outlook	Temp	Humidity	Wind	Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Weak	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cold	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

$$IG(Tennis, \text{Humidity}) = E(Tennis) - E(Tennis|Humidity) = 0.151$$



Decision tree

try with attribute **Wind**

$$E(Tennis) = 0.94$$

Tennis				
	Yes	No	Tot	
Wind	Weak	6	2	8
	Strong	3	3	6

Day	Outlook	Temp	Humidity	Wind	Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Weak	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cold	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

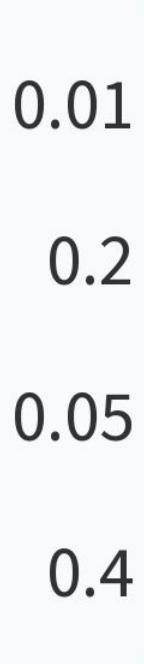
$$E(Tennis|Wind = weak) = ?$$

$$E(Tennis|Wind = stro) = ?$$

$$E(Tennis|Wind) = ?$$

$$IG(Tennis, Wind) = E(Tennis) - E(Tennis|Wind) = ?$$

What is the Information gain for the attribute "Wind"?



Decision tree

try with attribute **Wind**

$$E(Tennis) = 0.94$$

Tennis

	Yes	No	Tot
Wind	6	2	8
Strong	3	3	6

Day	Outlook	Temp	Humidity	Wind	Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Weak	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cold	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

$$E(Tennis|Wind = weak) = -\frac{6}{8} \log_2 \frac{6}{8} - \frac{2}{8} \log_2 \frac{2}{8} = -0.81$$

$$E(Tennis|Wind = stro) = -\frac{3}{6} \log_2 \frac{3}{6} - \frac{3}{6} \log_2 \frac{3}{6} = -1.00$$

$$E(Tennis|Wind) = -\frac{8}{14} 0.81 - \frac{6}{14} 1 = -0.89$$

$$IG(Tennis, Wind) = E(Tennis) - E(Tennis|Wind) = 0.05$$



Feature selection

The feature selected among Humidity and Wind is:

- Humidity

Day	Outlook	Temp	Humidity	Wind	Tennis
D1	Sunny	Hot	High	Weak	No
D2	Sunny	Hot	High	Strong	No
D3	Overcast	Hot	High	Weak	Yes
D4	Rain	Mild	High	Weak	Yes
D5	Rain	Cool	Normal	Weak	Yes
D6	Rain	Cool	Normal	Strong	No
D7	Overcast	Cool	Normal	Weak	Yes
D8	Sunny	Mild	High	Weak	No
D9	Sunny	Cold	Normal	Weak	Yes
D10	Rain	Mild	Normal	Strong	Yes
D11	Sunny	Mild	Normal	Strong	Yes
D12	Overcast	Mild	High	Strong	Yes
D13	Overcast	Hot	Normal	Weak	Yes
D14	Rain	Mild	High	Strong	No

$$IG(Tennis, \text{Humidity}) = E(Tennis) - E(Tennis|\text{Humidity}) = 0.151$$

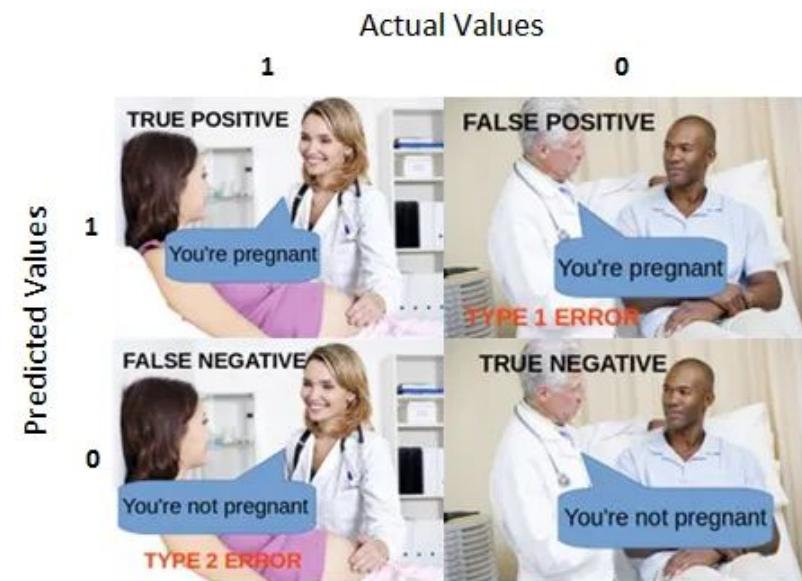
$$IG(Tennis, \text{Wind}) = E(Tennis) - E(Tennis|\text{Wind}) = 0.05$$

Performance assessment

Confusion matrix

The confusion matrix helps to understand the most common mistakes of our predictor

		Actual Values	
		Positive (1)	Negative (0)
Predicted Values	Positive (1)	TP	FP
	Negative (0)	FN	TN



Confusion matrix

A little example

`pred = [0, 0, 2, 2, 0, 2]`
`gr.tr. = [2, 0, 2, 2, 0, 1]`

	<code>pr_0</code>	<code>pr_1</code>	<code>pr_2</code>
<code>gt_0</code>	2	0	0
<code>gt_1</code>	0	0	1
<code>gt_2</code>	1	0	2

Confusion matrix

for class 0:

$\text{pred} = [0, 0, 2, 2, 0, 2]$
 $\text{gr.tr.} = [2, 0, 2, 2, 0, 1]$

Correctly classified (TP)

	pr_0	pr_1	pr_2
gt_0	2	0	0
gt_1	0	0	1
gt_2	1	0	2

Confusion matrix

for class 0:

pred = [0, 0, 2, 2, 0, 2]
gr.tr. = [2, 0, 2, 2, 0, 1]

	pr_0	pr_1	pr_2
gt_0	2	0	0
gt_1	0	0	1
gt_2	1	0	2

False Positives (FP)

Confusion matrix

for class 0:

pred = [0, 0, 2, 2, 0, 2]
gr.tr. = [2, 0, 2, 2, 0, 1]

	pr_0	pr_1	pr_2
gt_0	2	0	0
gt_1	0	0	1
gt_2	1	0	2

False Negatives (FN)



Accuracy

Guessed / total

pred = [0, 0, 2, 2, 0, 2]

gr.tr. = [2, 0, 2, 2, 0, 1]

$$\text{acc} = 4 / 6 = 0.66 = 66\%$$

	pr_0	pr_1	pr_2
gt_0	2	0	0
gt_1	0	0	1
gt_2	1	0	2



Hands on our first exercise!

Exercise 1 - Ephatitis Dataset

Predict whether a person infected with hepatitis virus will live or die

Attribute	Values
Class	DIE, LIVE
AGE	10, 20, 30, 40, 50, 60, 70, 80
SEX	male, female
STEROID	no, yes
ANTIVIRALS	no, yes
FATIGUE	no, yes
MALAISE	no, yes
ANOREXIA	no, yes
LIVER BIG	no, yes
LIVER FIRM	no, yes
SPLEEN PALPABLE	no, yes
SPIDERS	no, yes
ASCITES	no, yes
VARICES	no, yes
BILIRUBIN	0.39, 0.80, 1.20, 2.00, 3.00, 4.00
ALK PHOSPHATE	33, 80, 120, 160, 200, 250
SGOT	13, 100, 200, 300, 400, 500,
ALBUMIN	2.1, 3.0, 3.8, 4.5, 5.0, 6.0
PROTIME	10, 20, 30, 40, 50, 60, 70, 80, 90
HISTOLOGY	no, yes

Tasks:

1. Explore the dataset
2. Replace missing content
3. Pre-process variables
4. Find outliers
5. Split
6. Train a random forest
7. Evaluate performance



Exercise 2 - Titanic

- Try yourself with the Titanic Dataset
- Before starting, drop columns 'Name', 'Ticket', 'Cabin'



Exercise 2 - Titanic

- Try yourself with the Titanic Dataset
- Before starting, drop columns 'Name', 'Ticket', 'Cabin'

```
df = pd.concat([df,pd.get_dummies(df['Embarked']),prefix='Embarked']) ,  
axis=1)
```

```
df = df.drop('Embarked', axis=1)  
df
```