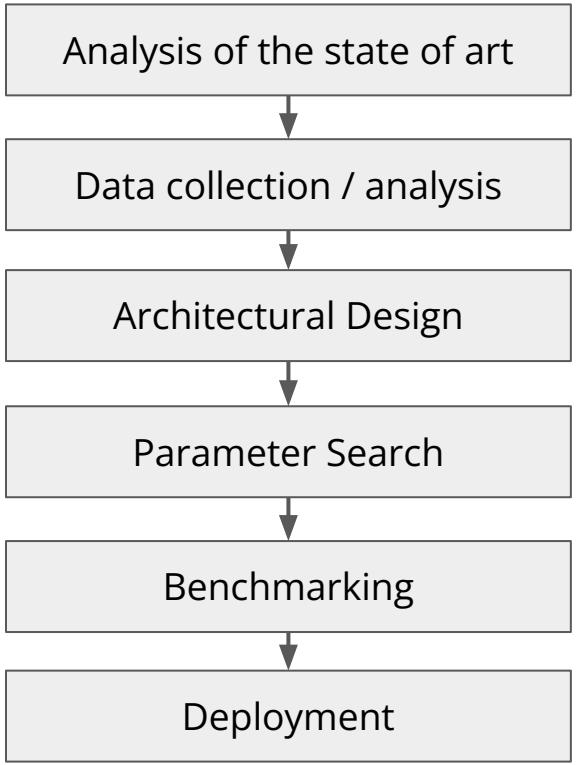
Università degli
Studi di Milano-Bicocca

Data collection & analysis

Prof. Flavio Piccoli

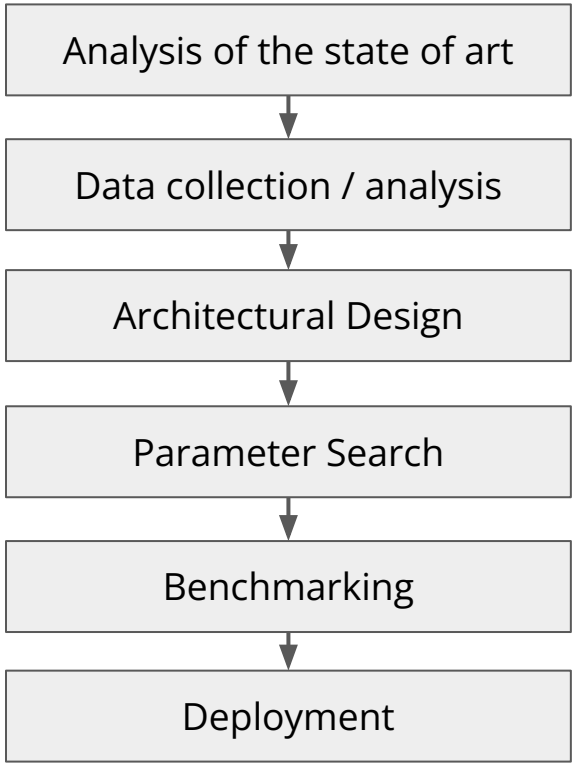
a.a. 2022-2023

R&D process



- Analyze the state of art to find similar works
- Collect or retrieve data relative to the task
- Create the neural architecture to reach the goal
- Find the most suitable hyper-parameters
- Measure the performance of the proposed system
- Send to production

R&D process



Google Scholar

pandas

scikit learn

PyTorch

PyTorch Lightning

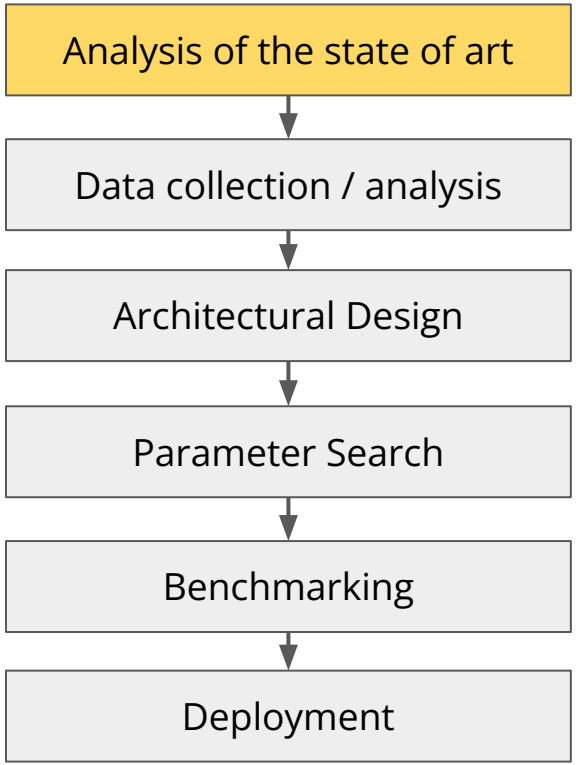
RAY tune

Streamlit

Flask

ONNX

R&D process



Google Scholar

pandas

scikit learn

PyTorch

PyTorch Lightning

RAY tune

Streamlit

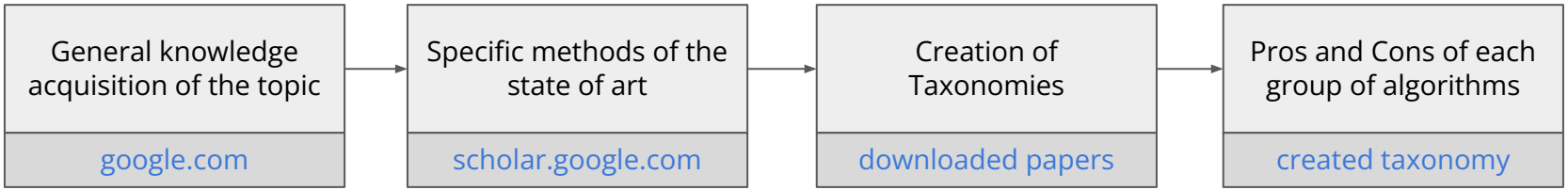
Flask

ONNX

Analysis of the state of art

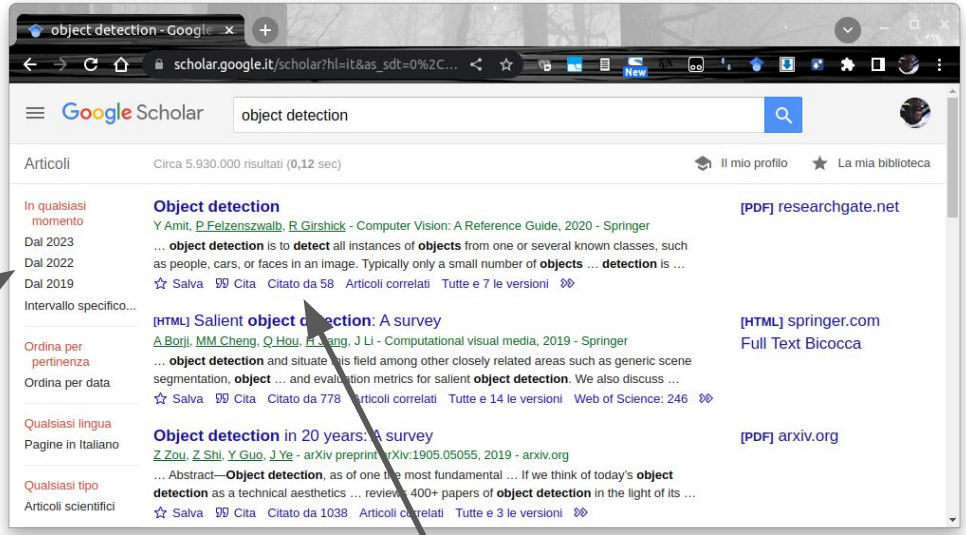
- Before starting to design a method, it is very important to analyze the state of art to find:
  - **methods** doing the same task
  - **benchmarks** to compare your method with the state of art
  - **metrics** for assessing your method

- Two important tools:
  - Google and
  - Google Scholar https://scholar.google.it



Google Scholar

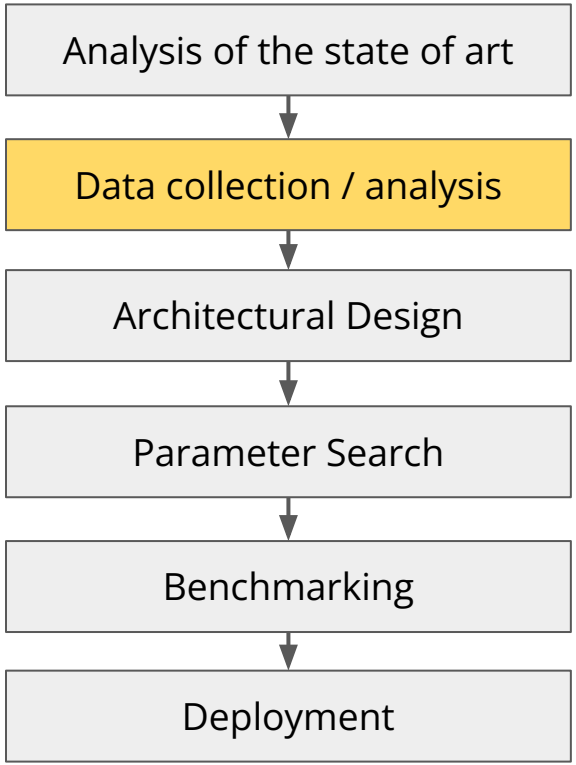
- Search engine for scientific papers



It is useful to select newest methods

once you find an interesting paper, read also citing literature

R&D process



Google Scholar

pandas

scikit learn

PyTorch

PyTorch Lightning

RAY tune

Streamlit

Flask

ONNX

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	1. comparison with sota methods 2. reliable results 3. huge amount of data available	1. data may not be replicable in real life 2. model may not work
Custom benchmark	1. same data of the final application 2. full knowledge of the dataset	1. comparison with sota methods may not be replicable 2. 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

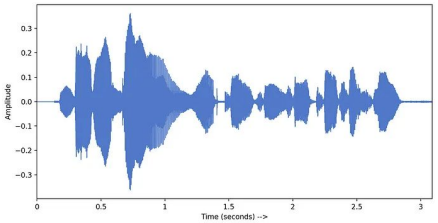
- it's a value that measures a quantity
    - can belong to a specific interval
    - can be an unlimited value

Digital signal data

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

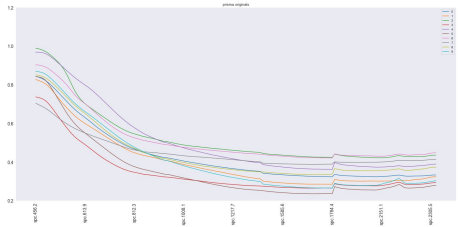
It can be:

- **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



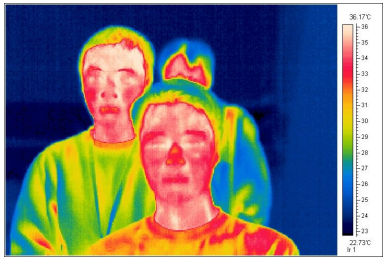
hyperspectral signal = wide spectrum of light

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

- Cathegorical label

cat	dog	frog

- Scalar feature

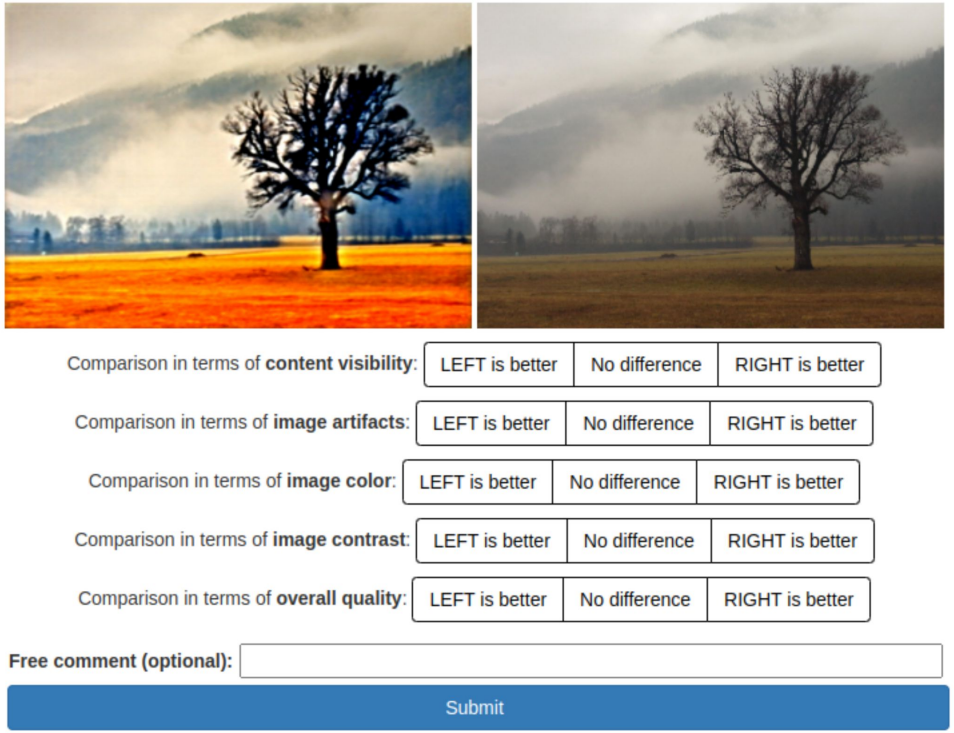
5.3 2 False

- Image level feature

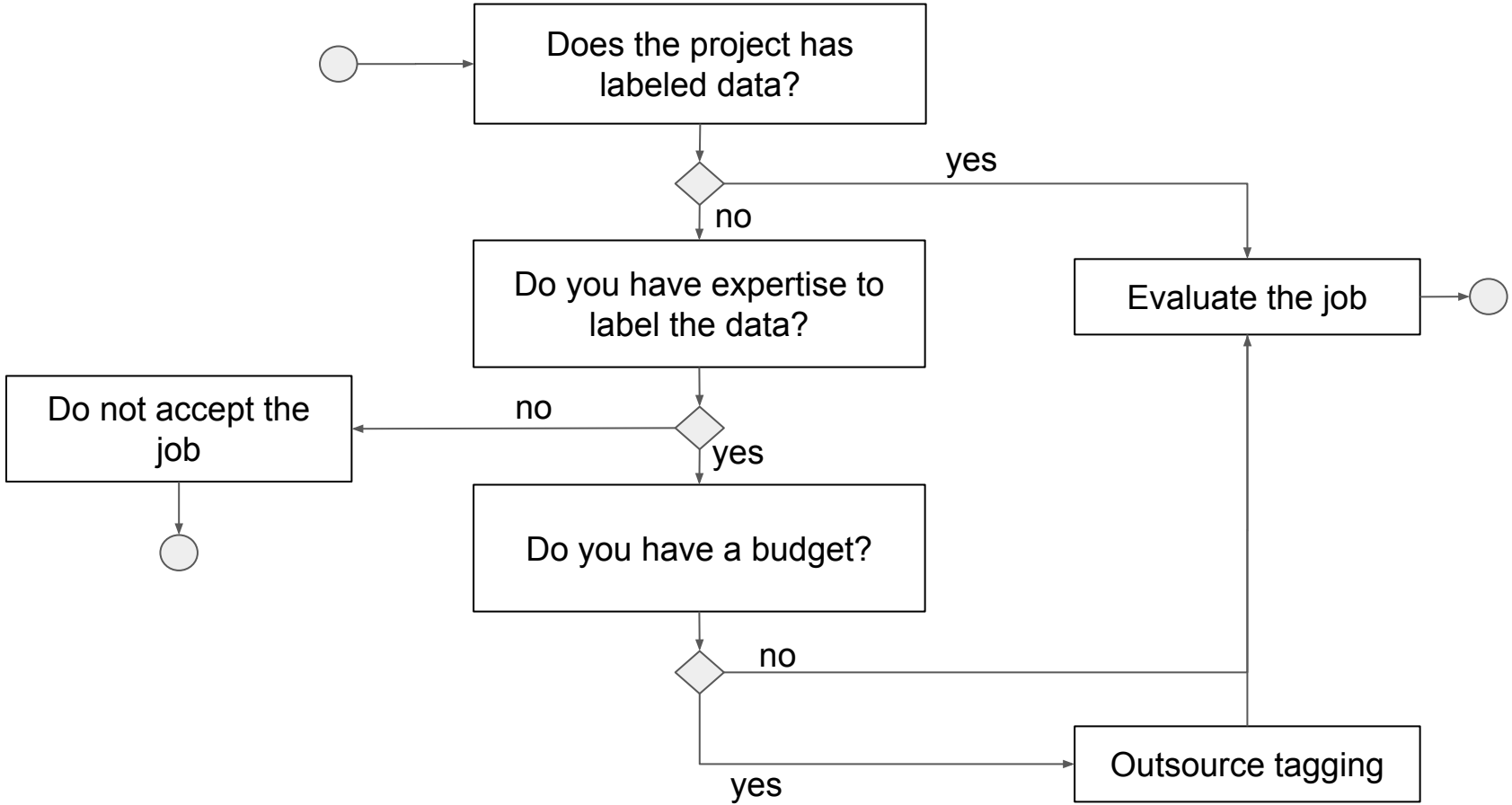
Existing tools & frameworks for data collection

- 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)

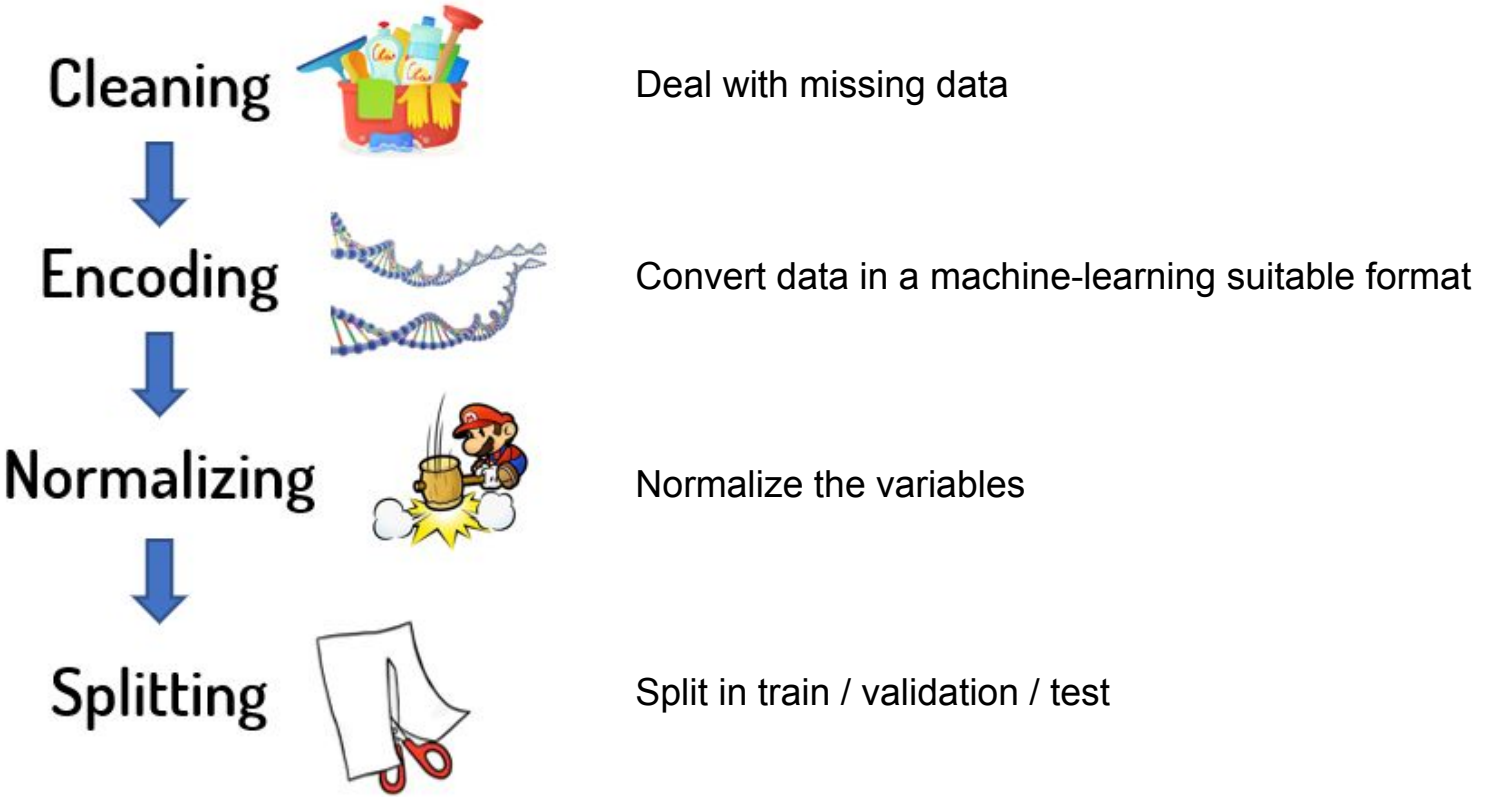
amazon mechanical turk



Feasibility evaluation



Data preprocessing



Deal with missing data

Convert data in a machine-learning suitable format

Normalize the variables

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)

brand
Fiat
BMW
Lamborghini
Fiat

brand
0
1
2
0

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

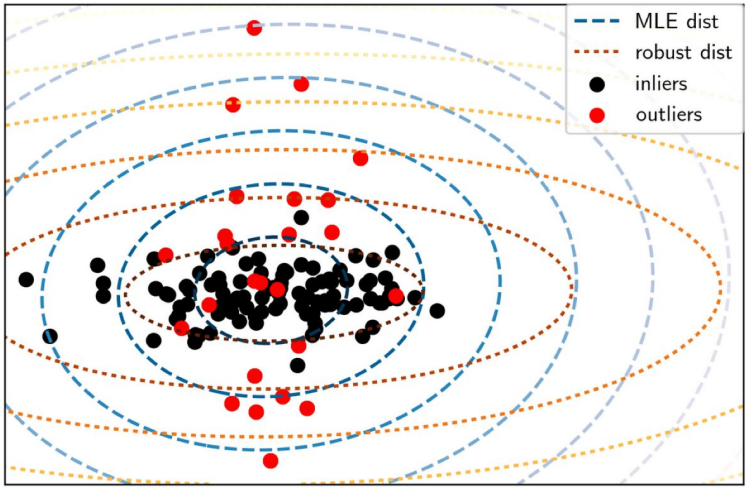
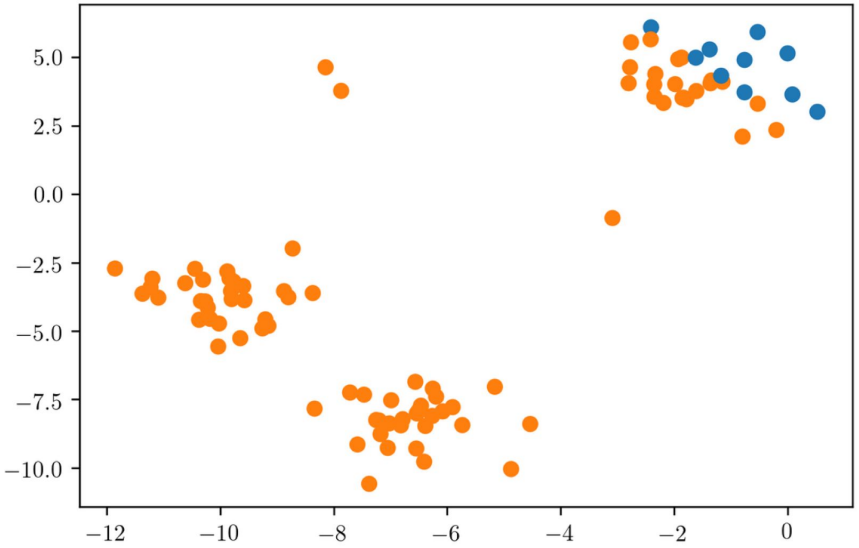


image courtesy: https://amueller.github.io/aml/03-unsupervised-learning/03-outlier-detection.html

Outlier detection - elliptic envelope

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

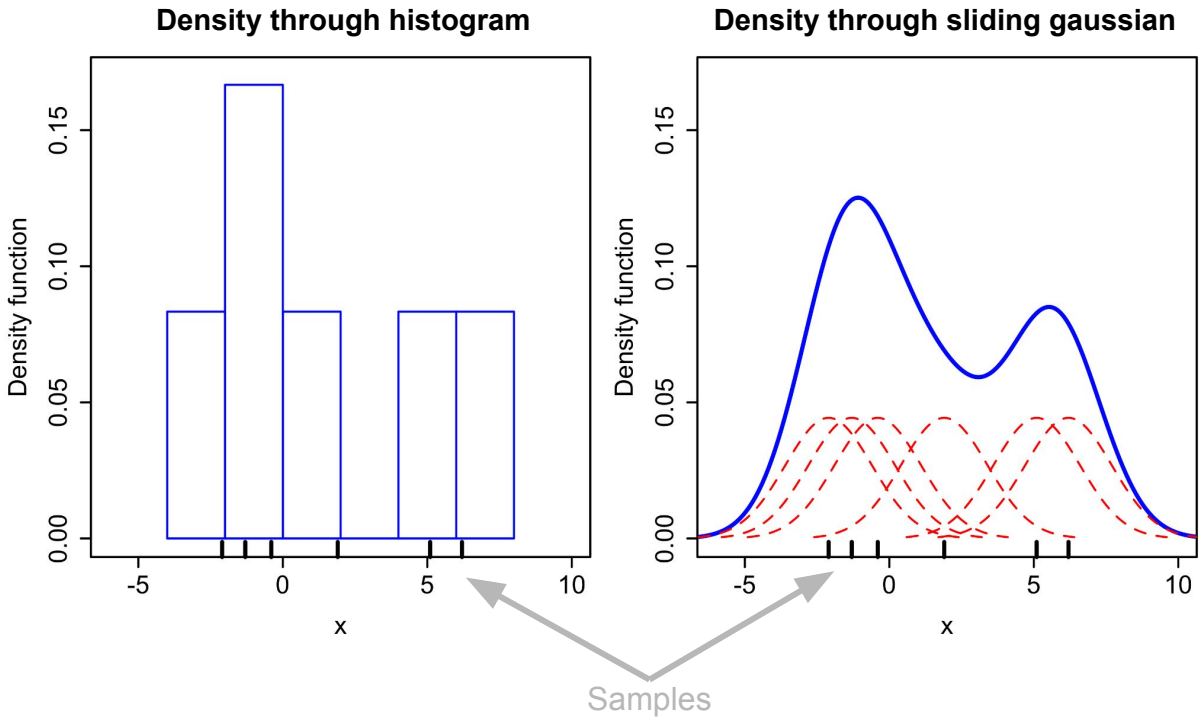


It fails!

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

Outlier detection - Kernel density estimation

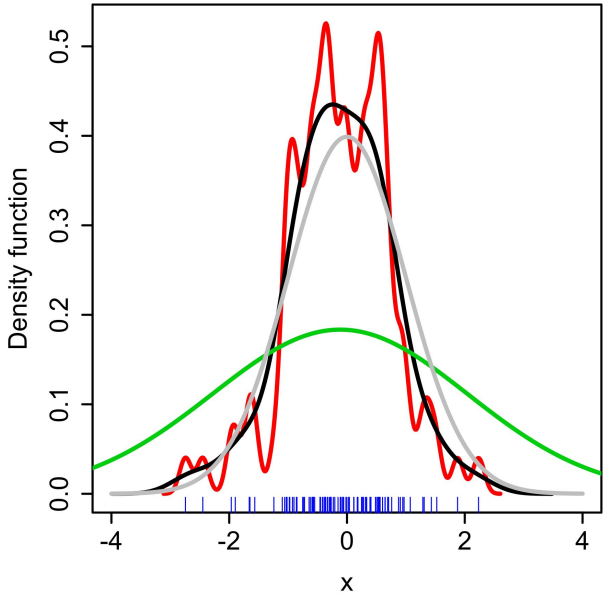
- Another approach is to study the density of the parameters and mark as outliers the ones with lower repr.



Outlier detection - Kernel density estimation

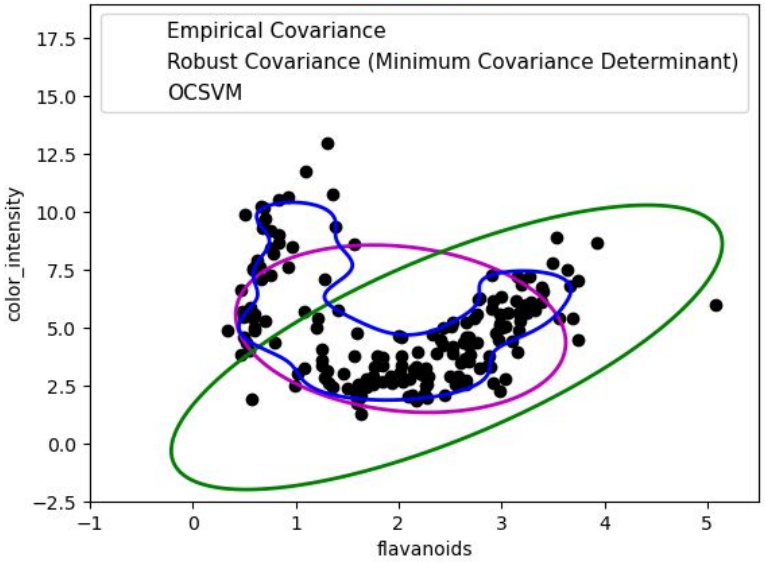
Problems:

- 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*)

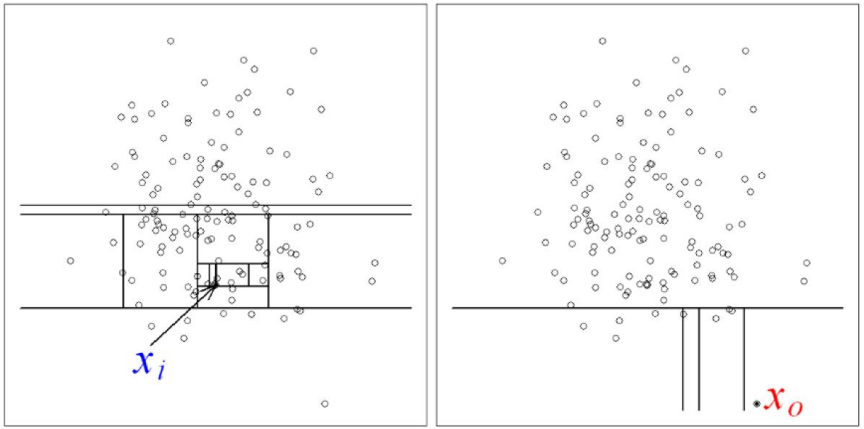


- here again there is a parameter..

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

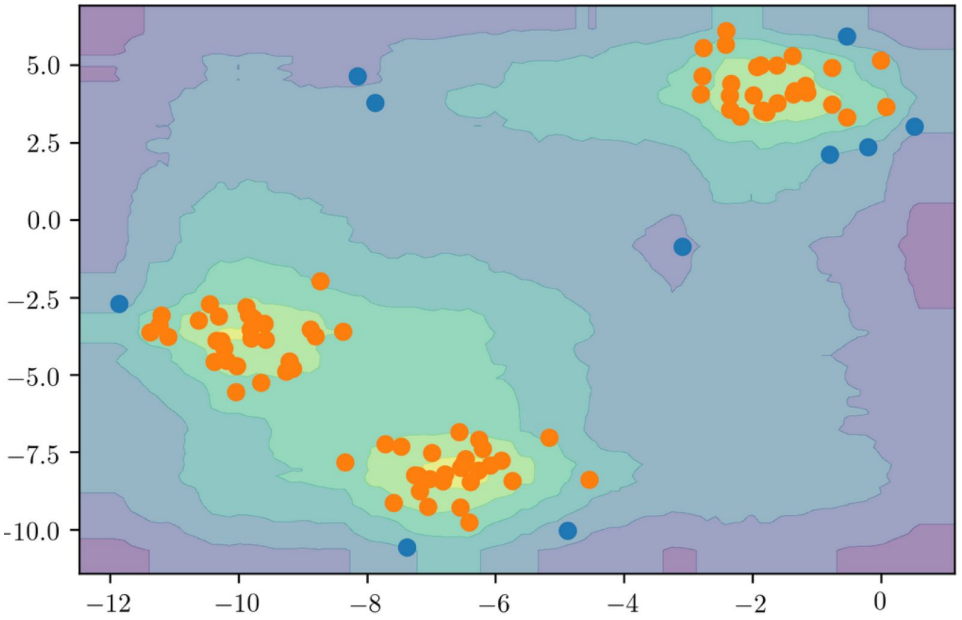


(a) Isolating x_i (b) Isolating x_o

- this method does not require any parameter!

Outlier detection - isolation forests

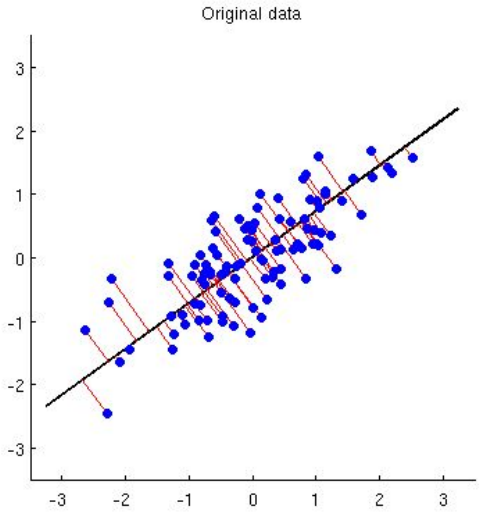
- Here there is a visual representation of the isolation through iso-curves on the two features

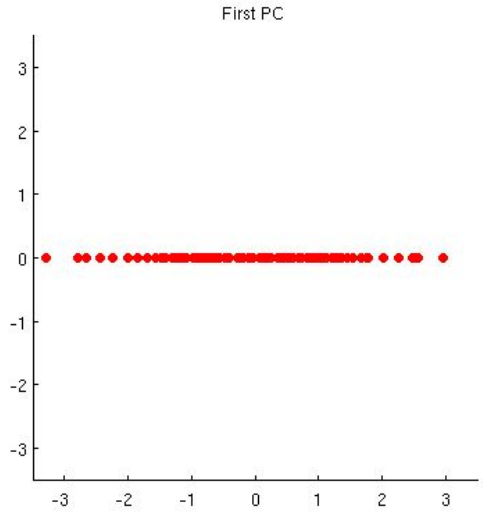


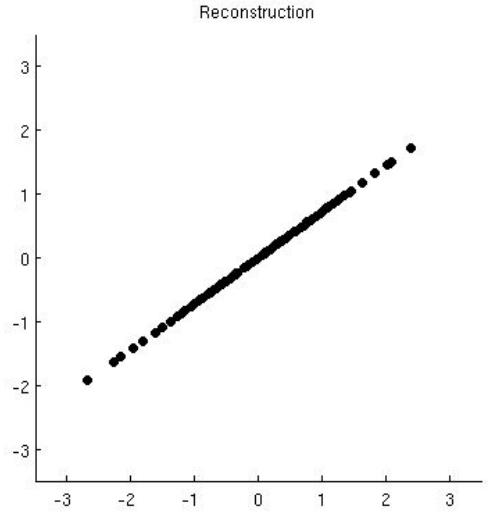
Outlier detection - PCA

idea:

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







Outlier or under-represented class?

- When you collect data, there can be some samples distant from the others

Idea: identify something that differs from the standard distribution

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

Hands on our first exercise!

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

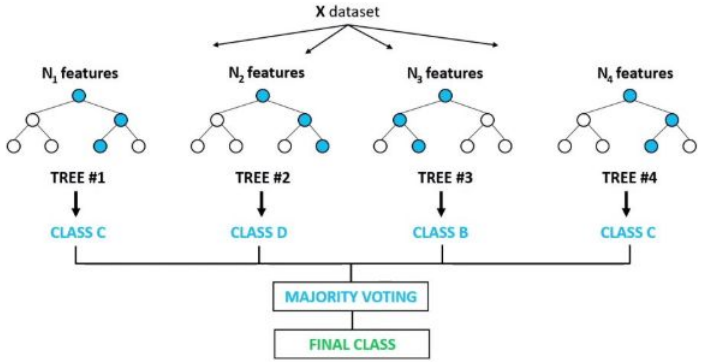
Tasks:

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

Random Forests

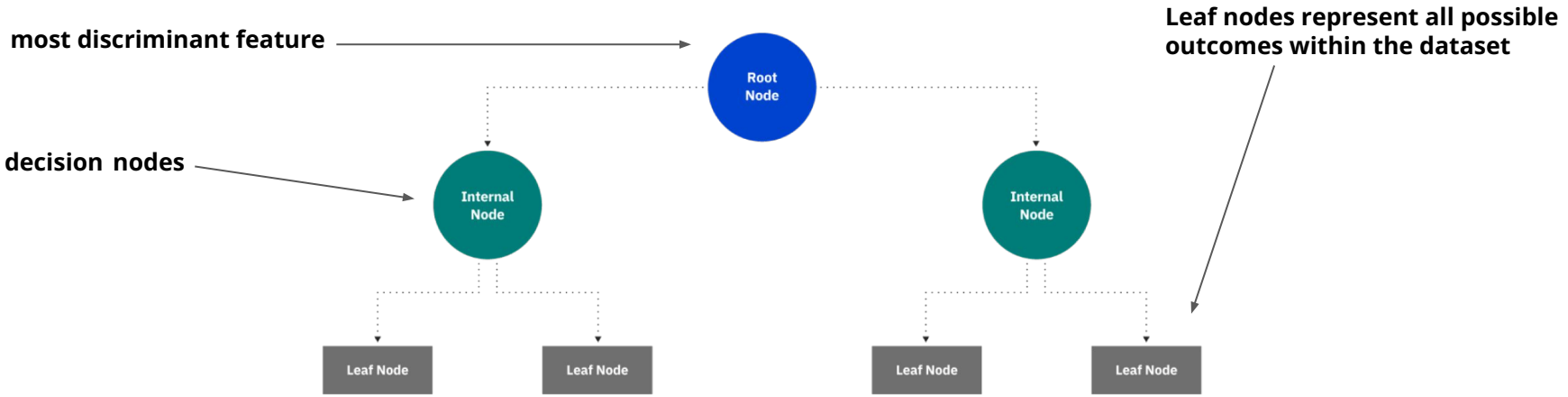
- it is one of the most popular tree-based supervised learning algorithms
- It is also the most flexible and easy to use
- Easy to backtrace features that mostly contributed to the final output

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.



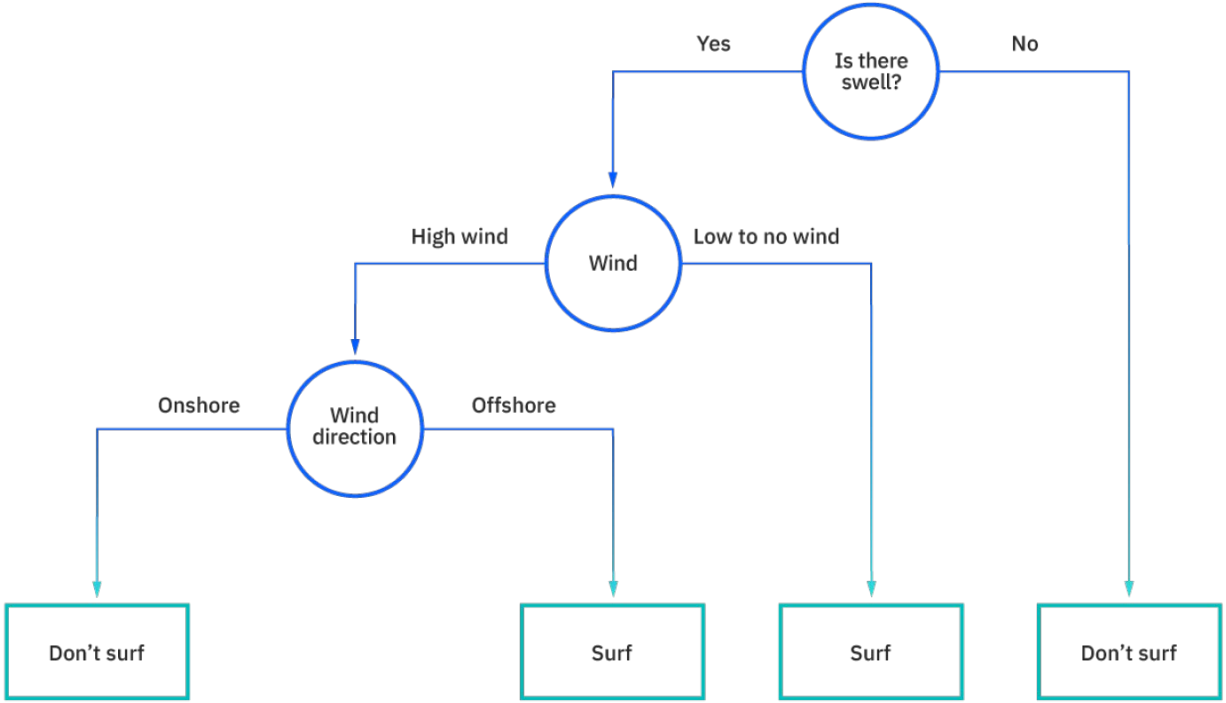
Decision tree

- 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 (Cassification 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

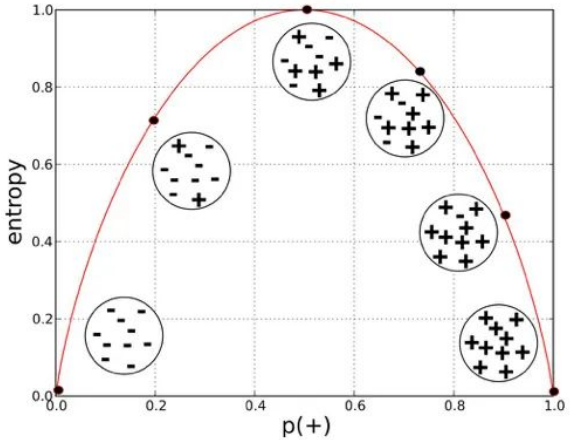
Entropy

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

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

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

- 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



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

$$IG(Y,X) = E(Y) - E(Y|X)$$

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

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

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

$$E(Tennis) = -\underbrace{\frac{9}{14} log_2 \frac{9}{14}}_{\text{yes tennis}} - \underbrace{\frac{5}{14} log_2 \frac{5}{14}}_{\text{no tennis}} = 0.94$$

Decision tree

we can then compute information gain on any attribute

For example, **Humidity**

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

Contingency table:

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

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

$$E(Tennis|Humidity = norm) = -\underbrace{\frac{6}{7}log_2\frac{6}{7}}_{\text{yes tennis}} - \underbrace{\frac{1}{7}log_2\frac{1}{7}}_{\text{no tennis}} = -0.59$$

Decision tree

we can then compute information gain on any attribute

For example, **Humidity**

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

$$E(Tennis|Humidity = high) = 0.985$$

$$E(Tennis|Humidity = norm) = 0.59$$

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

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

$$IG(Tennis, Humidity) = E(Tennis) - E(Tennis|Humidity) = 0.151$$

Decision tree

try with attribute **Wind**

$$E(Tennis) = 0.94$$

Wind	Tennis: Yes	Tennis: No	Tot
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

Total samples = 14
Weak wind = 8
Strong wind = 6

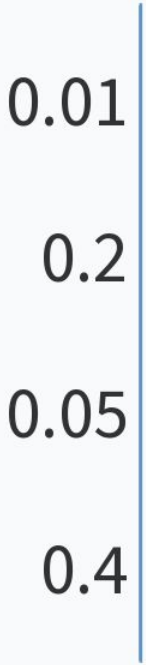
$$E(Tennis|Wind = weak) = ?$$ -6/8 log2(6/8) - 2/8 log2(2/8) = 0,81

$$E(Tennis|Wind = stro) = ?$$ -3/6 log2(3/6) - 3/6 log2(3/6) = 1

$$E(Tennis|Wind) = ?$$ -8/14*0,81 - 6/14*1 = -0,89

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

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



Decision tree

try with attribute **Wind**

$$E(Tennis) = 0.94$$

		Tennis		
		Yes	No	Tot
Wind	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

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

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

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

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

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

$$IG(Tennis, Humidity) = E(Tennis) - E(Tennis|Humidity) = 0.151$$

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

Confusion matrix

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

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

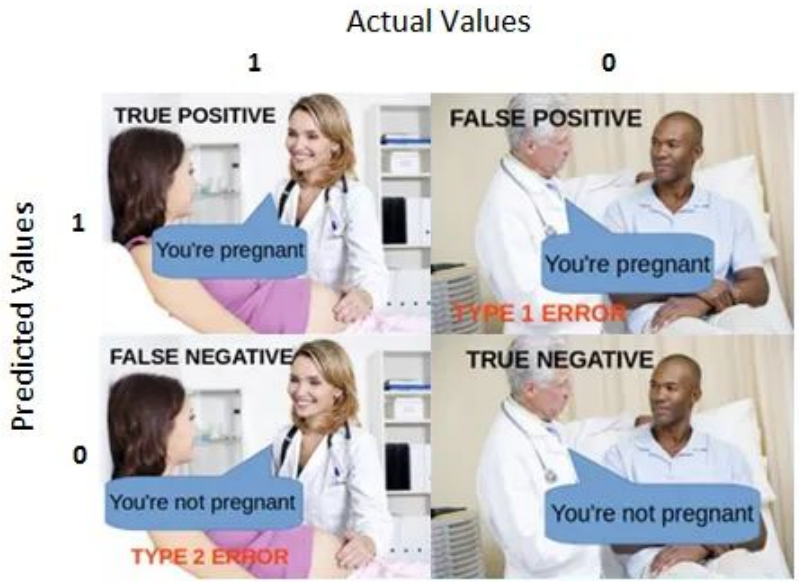


Image courtesy: https://towardsdatascience.com/understanding-confusion-matrix-a9ad42dcfd62

Confusion matrix

A little example

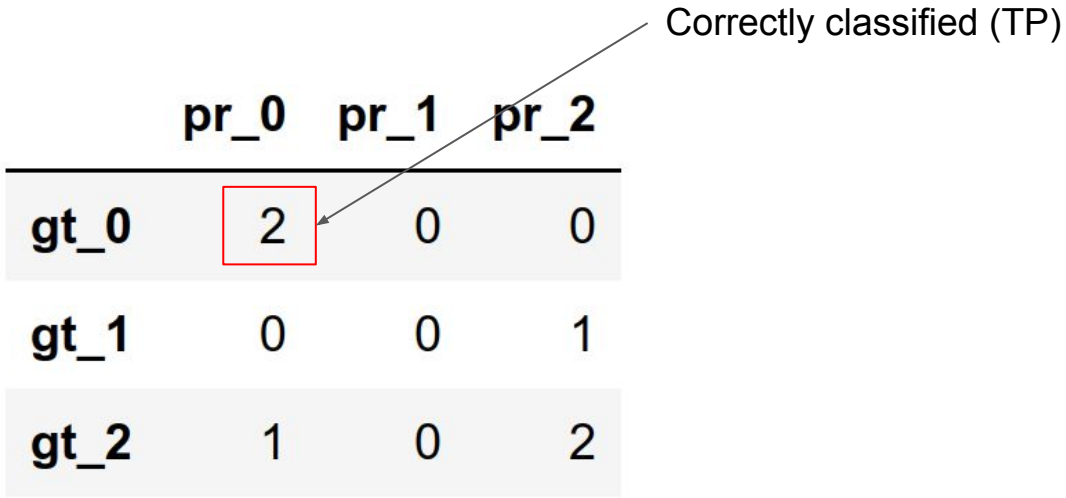
pred = [0, 0, 2, 2, 0, 2]
gr.tr. = [2, 0, 2, 2, 0, 1]

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

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

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]

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

Exercise 2 - Titanic

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