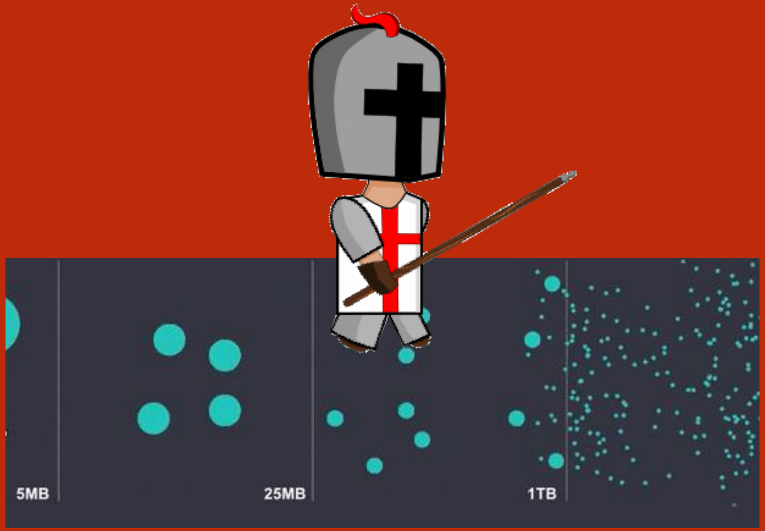
ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA



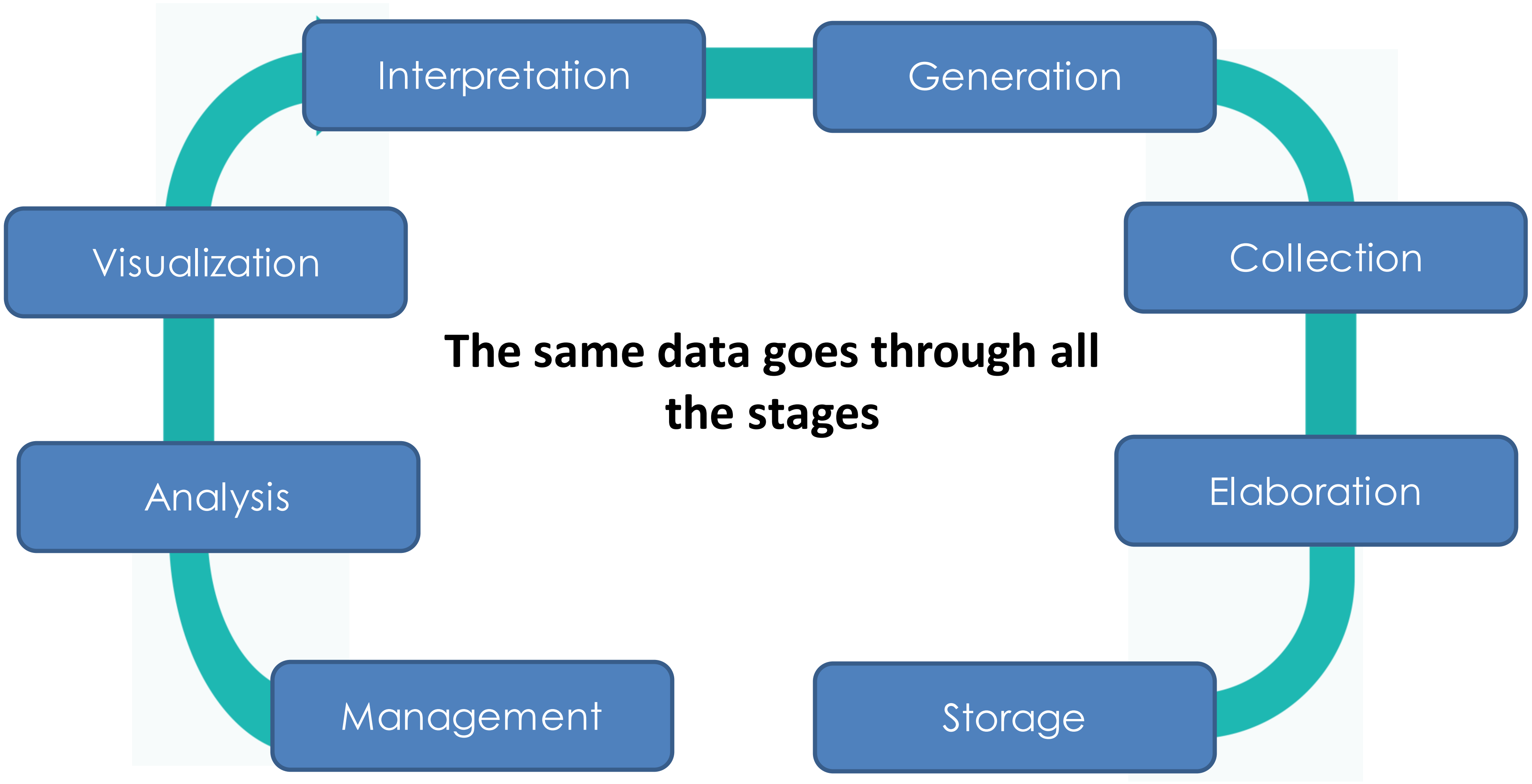
The Machine learner path

Lorenzo Stacchio

PhD student in Computer Science

Department for Life Quality studies

Big data cycle



Generation

Collection

Elaboration

Storage

Management

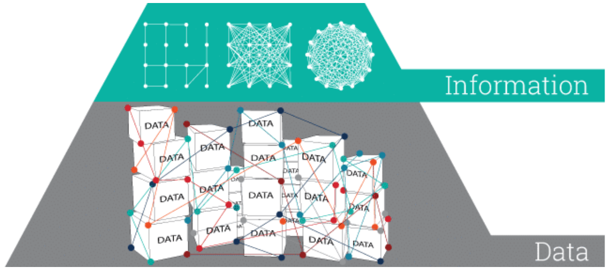
Analysis

Visualization

Interpretation

The same data goes through all the stages

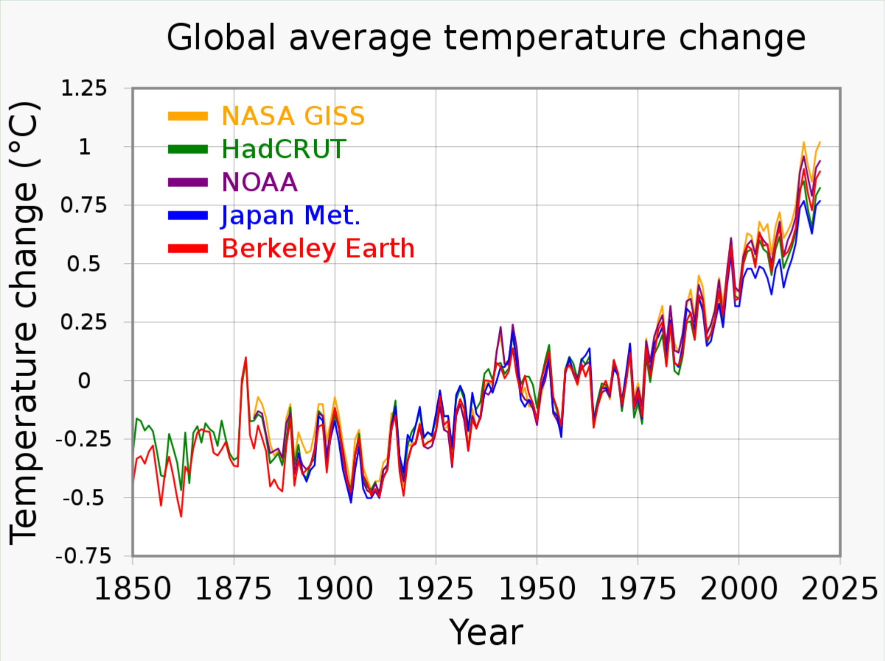
Data, Information, Knowledge and Wisdom



- **Data** is defined as a collection of rough and unorganized facts, such as numbers and characters. Without context, the data makes little sense. For example, 1 is a number, but we can see it as a temperature degree (1 °). We have therefore transformed the data into information!

- **Information** is a "clean and processed" set of data in a way that makes manipulation, analysis, and visualization easier; For example, we could make a graph to analyze the global mean temperature values over the past 170 years!

- By asking pertinent questions about "who", "what", "when", "where", etc., we can derive valuable insights from information by making it more useful. But when we come to the "how" question, we are forced to make the leap from information to knowledge!



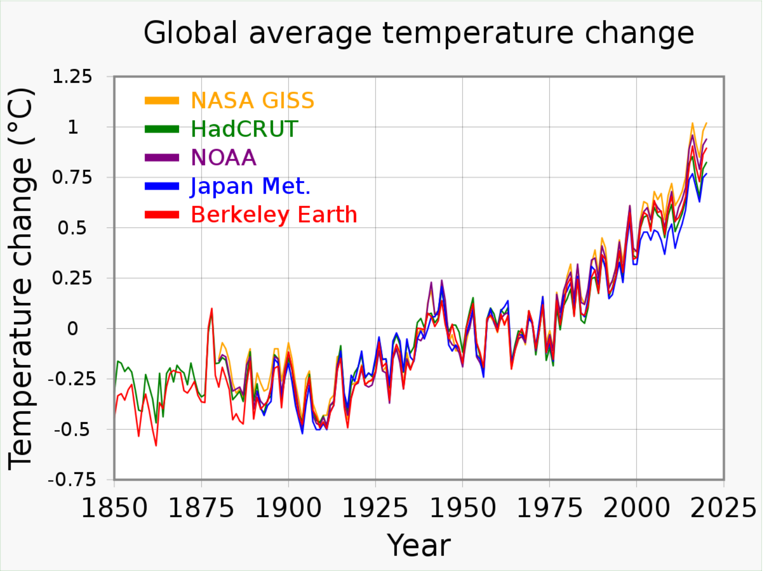
- How is the information derived from the collected data relevant to our objectives? "How" are the pieces of this information linked to other pieces to add more meaning and value? And most importantly, "how" can we apply the information to achieve our goal?

- When we understand how to apply information to achieve our goals, we turn it into knowledge.

- When we use the knowledge and insights gained from information to make proactive decisions, we can say that we have reached the final step of the pyramid: wisdom.

- Wisdom is at the top of the hierarchy and is nothing more than applied knowledge to make the best possible decision!

Wisdom
Knowledge



- What are the causes of global warming? (**knowledge**)

- What can we do to stop it? (**wisdom**)

The Machine learner path

Domain knowledge

Programming

Database

No SQL

SQL

Big data

APACHE Spark

hadoop

Machine & Deep Learning

scikit learn

Math

Analysis, Linear Algebra, Probability and statistics

Learn how to map data into knowledge

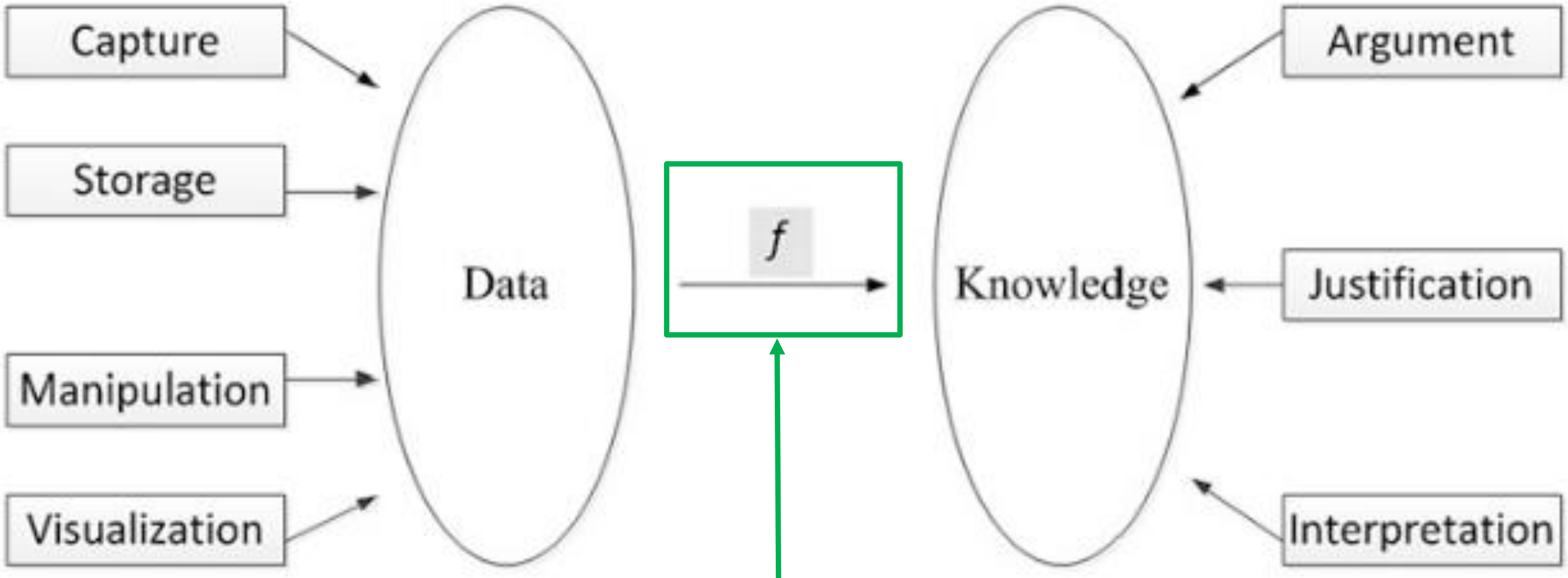
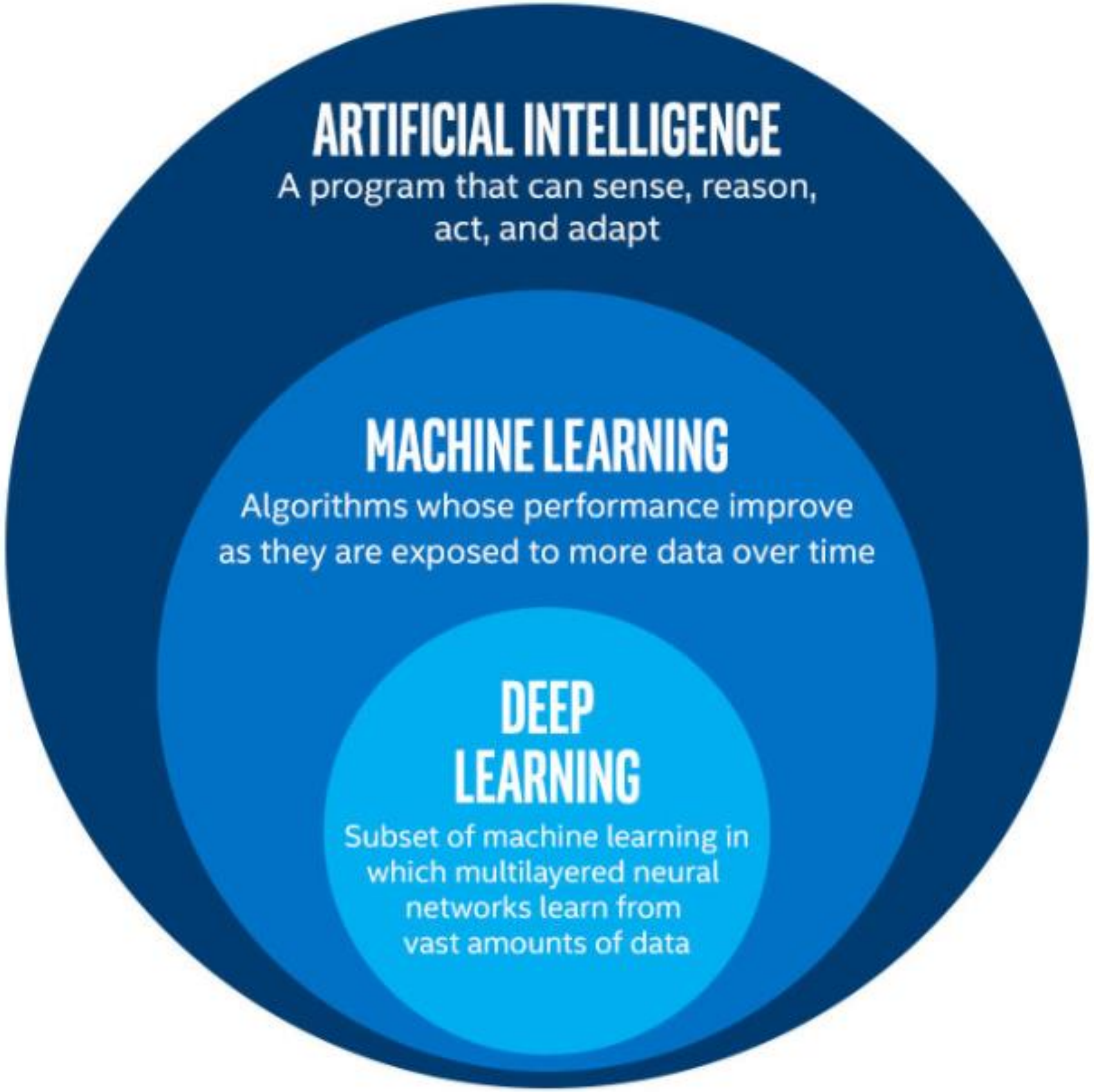


Fig. 1.1 Transformation of data into knowledge

Machine & Deep Learning

Artificial Intelligence (AI), Machine Learning (ML) e Deep Learning (DL)


ARTIFICIAL INTELLIGENCE
A program that can sense, reason, act, and adapt
MACHINE LEARNING
Algorithms whose performance improve as they are exposed to more data over time
DEEP LEARNING
Subset of machine learning in which multilayered neural networks learn from vast amounts of data

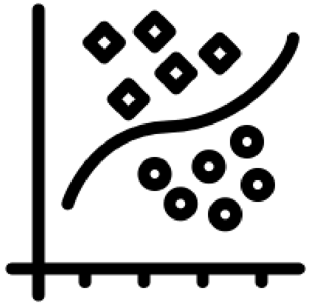
https://towardsdatascience.com/cousins-of-artificial-intelligence-

Machine Learning – A way to solve non classical problems

Classical algorithms can't solve many problems, among them:

- **Regression problems**;

- **Classification problems**;



- **Data Mining**.

Machine Learning – Characteristics

- Low prior knowledge of information;
- Data with a high number of attributes (input features);
- High volume of data for training;
- Adaptability.

https://medium.com/@lokeshpara17/tiny-imagenet-using-pytorch-42a3f2ee3c9d

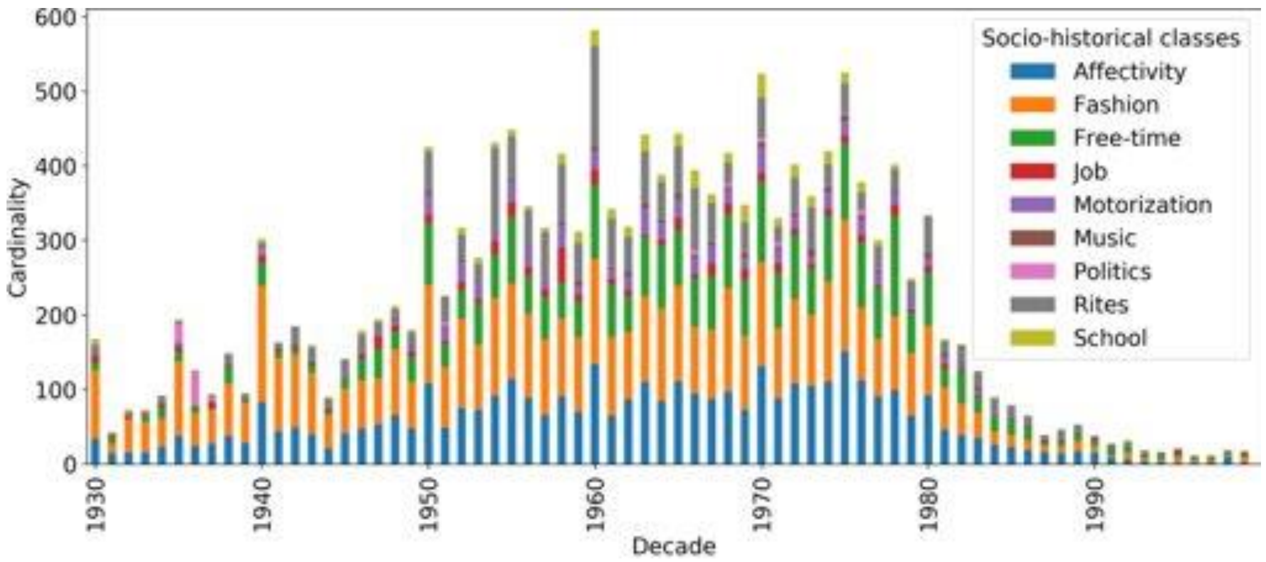
https://www.image-net.org/

Imago: A family photo album dataset for a socio-historical analysis of the twentieth century

- The IMAGO data collection project launched in 2004 by socio-historical scholars to study the evolution of Social History through the lenses of photographs from family albums;

- The collection includes about 80,000 photos, taken between 1845 and 2009, belonging to about 1,500 Italian family albums, offering the opportunity to study the evolution of Italian society during the twentieth century.

- Among these, 16,642 images were tagged by the students of the degree course in Fashion Cultures and Practices, under the supervision of the social-history faculty.



Each image is tagged with different attributes:

- Shooting year;
- Socio-historical context;
- Textual description;

- Where is the low information? In the knowledge of the model!

- What can I say about image distribution?
- What can I say about the pixels in the 1948 images?
- Why do certain pixels belong to the year 1945 and not 1946?

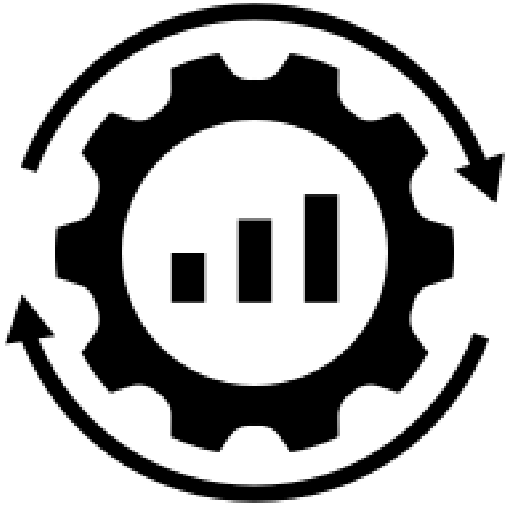
- Machine learning must learn to provide these answers and at that point we can use it wisely!

Scientific papers on IMAGO

- …
- Stacchio, L., Angeli, A., Lisanti, G., Calanca, D., & Marfia, G. (2022). Towards a holistic approach to the socio-historical analysis of vernacular photos. ACM Transactions on Multimedia Computing, Communications, and Applications (TOMM).
- Stacchio, L., Angeli, A., Hajahmadi, S., & Marfia, G. (2021, October). Revive Family Photo Albums through a Collaborative Environment Exploiting the HoloLens 2. In 2021 IEEE International Symposium on Mixed and Augmented Reality Adjunct (ISMAR-Adjunct) (pp. 378-383). IEEE;
- Stacchio, L., Hajahmadi, S., & Marfia, G. (2021, March). Preserving Family Album Photos with the HoloLens 2. In 2021 IEEE Conference on Virtual Reality and 3D User Interfaces Abstracts and Workshops (VRW) (pp. 643-644). IEEE.

Machine Learning – Learning Algorithm

- Define a model for the problem to be solved: the model depends on a set of parameters;
- Define a metric to measure performance: a measure of error to evaluate the model;
- Train the model (update the parameters), to minimize the error, on the training data.



A LEARNING ALGORITHM IS AN OPTIMIZATION PROCESS

Machine Learning – Parameters ed Hyper-parameters

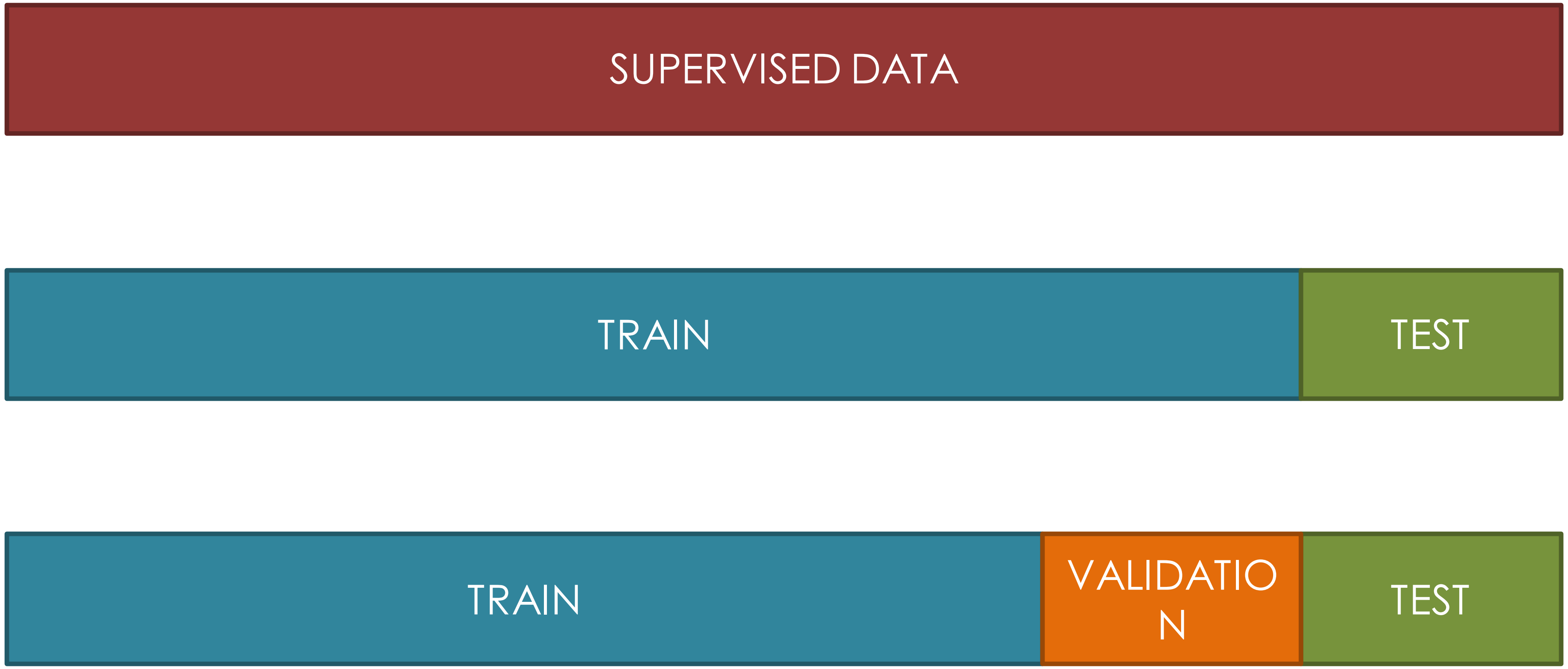
Parameters

With parameters of a machine and / or deep learning model we mean the parameters whose value changes during the training of the model. The value of the parameters then changes based on the data of the training set on the "instruction" of the algorithm used during training to minimize the error function.

Hyper-parameters

By hyper-parameters of a machine and / or deep learning model we mean (if any) the parameters whose value is decided by the programmer before training the model. The programmer can then decide to use part of the data he has available (not belonging to the training set) to form another set of data (validation set) and use it to optimize the choice of hyper-parameters.

Machine Learning – Training, Validation e Test set



SUPERVISED DATA

TRAIN

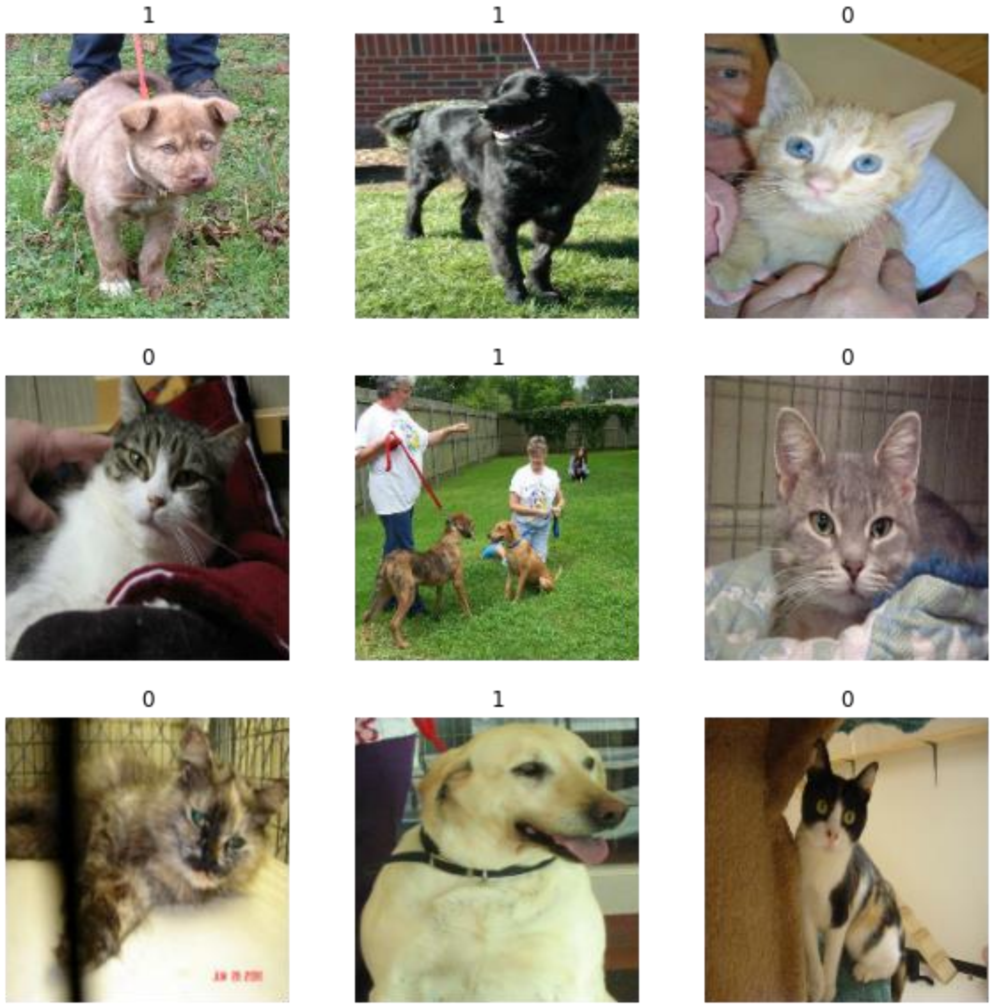
TEST

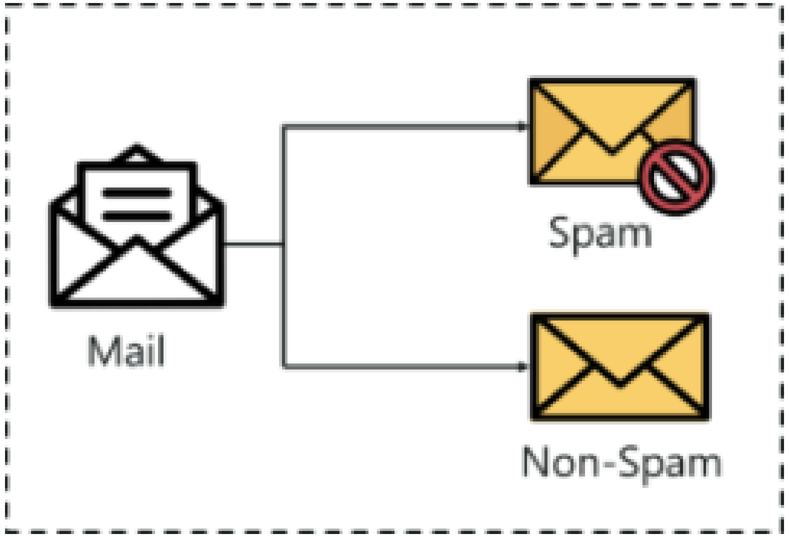
TRAIN

VALIDATIO
N

TEST

Machine Learning and classification: Supervised learning





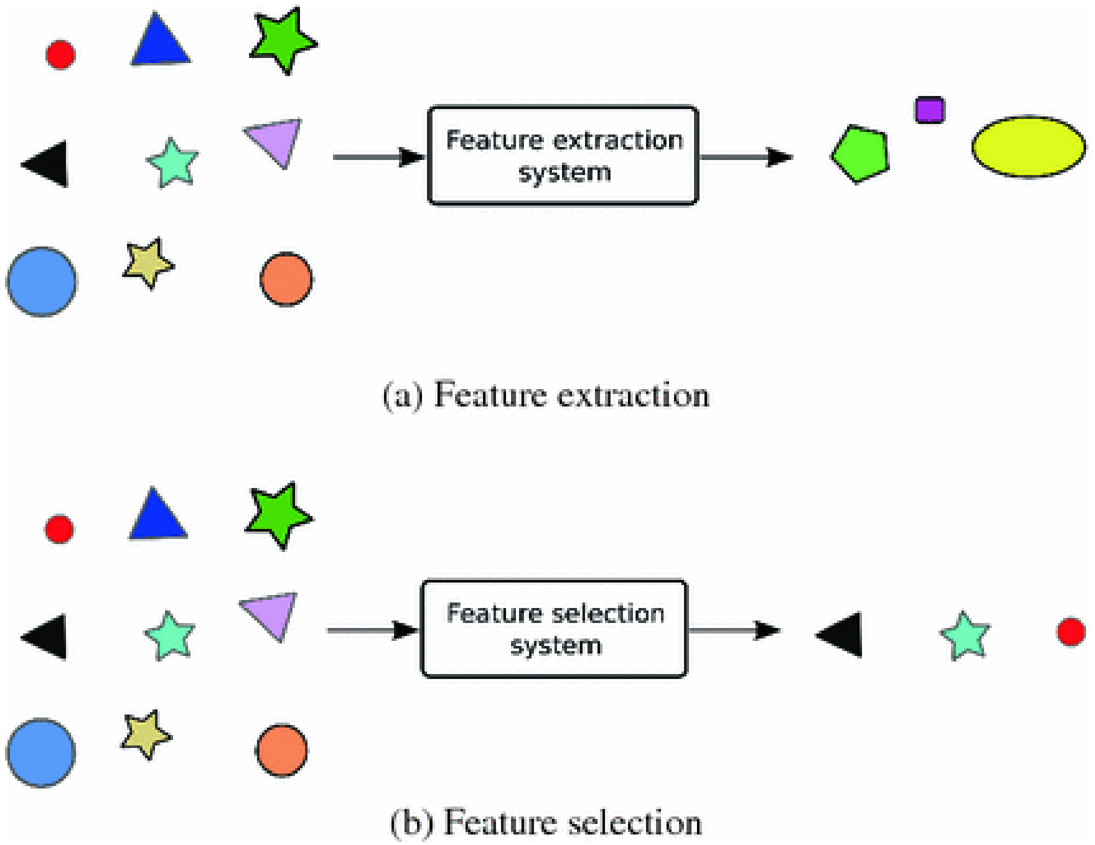
Machine Learning – Features

- Any information relating to a datum is called a feature;

- Features are the inputs in a learning process / algorithm;

- Machine learning models are highly sensitive to the choice of features;

- Choosing "good" features (feature selection and / or feature extraction) can be very difficult (but it becomes fundamental).



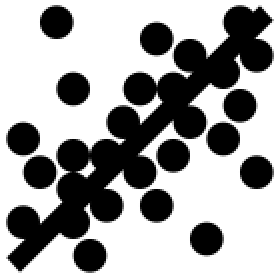
(a) Feature extraction

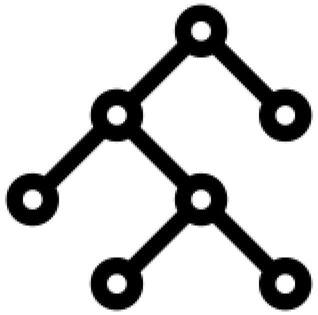
(b) Feature selection

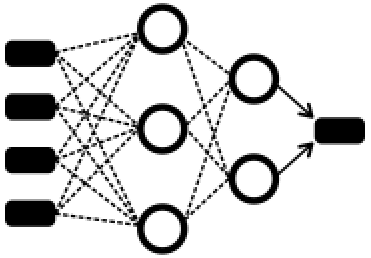
https://link.springer.com/chapter/10.1007/978-3-319-90080-3_2

ML and classification – A forest of models (it's a joke, you will get it later)

- Different techniques for defining models:
  - Linear models;
  - Tree models;
  - Neural networks;

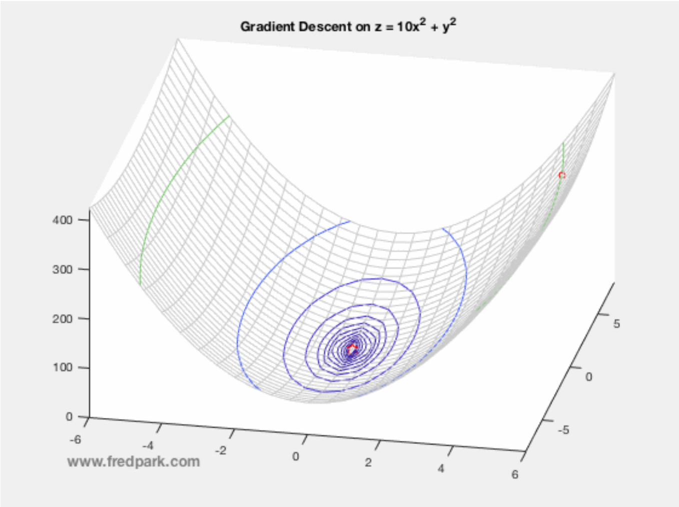






- Several techniques for defining error functions:
  - Cross-entropy loss;
  - Cosine distance;
  - Logistics function;

- Several techniques for optimizing models:
  - Entropy and Information gain;
  - Conditional probability;
  - Gradient descent;



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

no-meme
sticker
¡¡Hola!!
Aweonao!
MUA!
meme
TE LA VENDO!!!

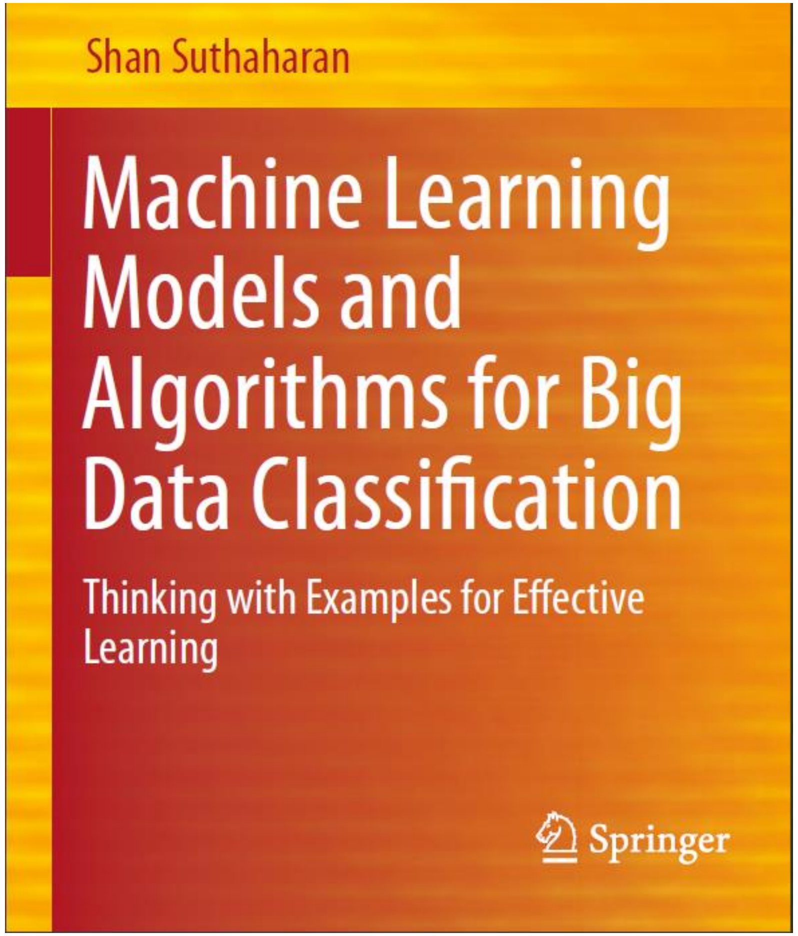
Machine learning for classification

Lorenzo Stacchio

PhD student in Computer Science

Department for Life Quality studies

Study material



Shan Suthaharan

Machine Learning Models and Algorithms for Big Data Classification

Thinking with Examples for Effective Learning

Springer

Chapter 1 and 6

Why we refer to models?

- Machine learning is about exploring and developing mathematical models and algorithms for learning from data.

- It often focuses on classification, which is implemented by modeling an optimal (parametric) mapping function between the data domain and the set of known classes between which we want to discriminate.

- This mapping could be a parameterized function or parameterized processes that learn the characteristics of a system from the input data.

- The term algorithm is often confused in the context of machine learning.

- Indeed, it is the modeling that can consist of different learning algorithms for the derivation of a model;

- The learning algorithm is used to train, validate, and test the model using a given data set to find an optimal parameter value, validate it, and evaluate its performance.

Supervised and unsupervised learning for classification

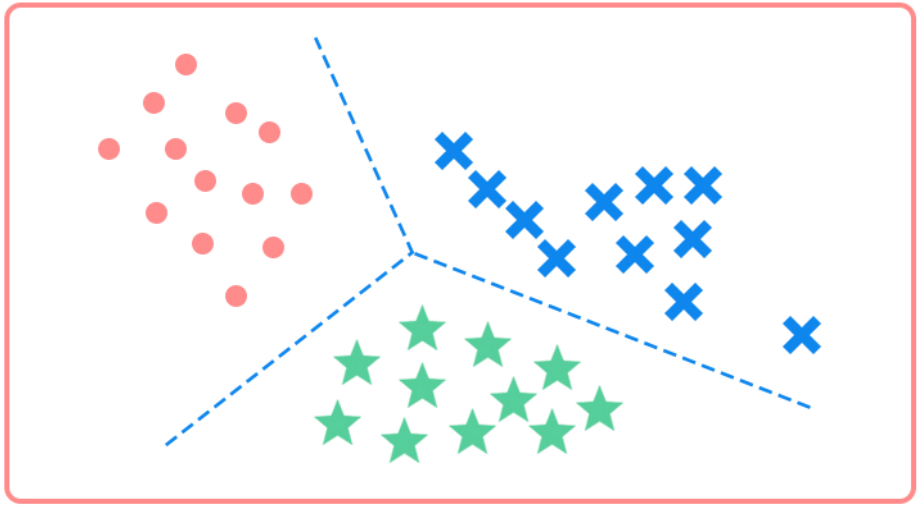
SUPERVISED LEARNING – Inputs + outputs (labels)

- Classification.

UNSUPERVISED LEARNING – Input (no labels)

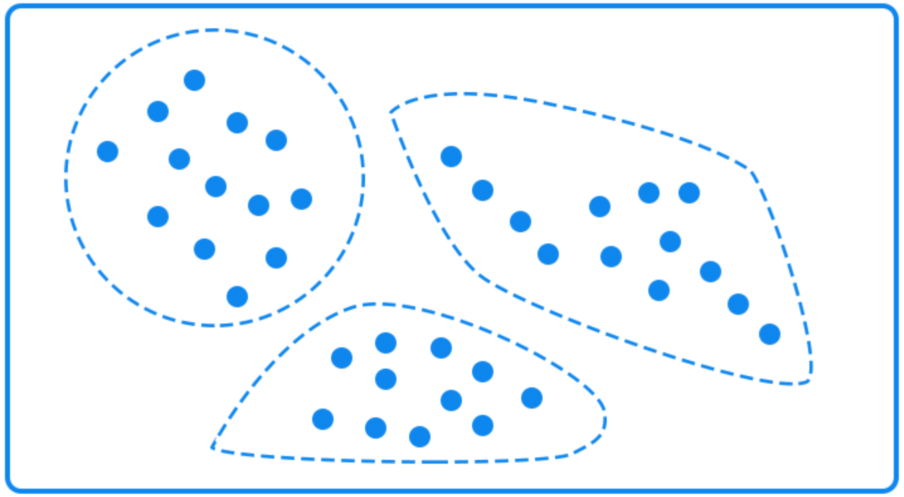
- Clustering;
- Feature extraction;

Classification



Supervised learning

Clustering



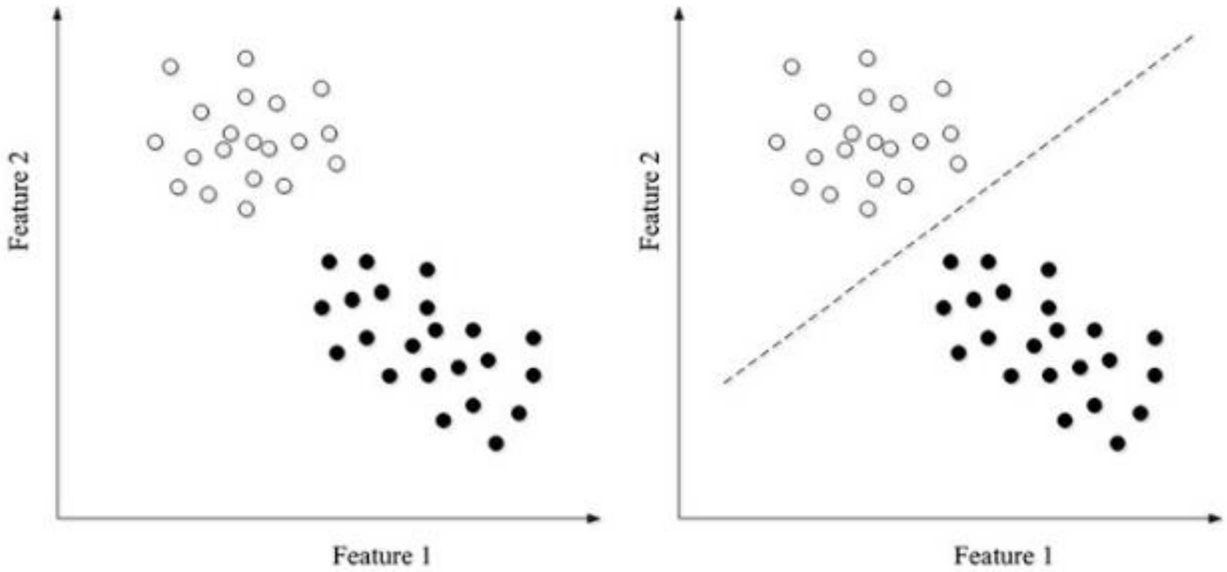
Unsupervised learning

Classification (supervised machine learning)

- When we face classification problems, we assume that we have labeled data (with classes) to generate rules (through training) to understand what to say about a certain input data, even when we have never seen it!

- Now suppose we need to find a mathematical function to separate the white points from the black points;

- We can model a parametric (linear) classification function that learns the optimal configuration from the data itself!



- In general, a classification process supposes to have data belonging to a certain domain D formed by the relation R_l, where l indicates the number of features available!

- If we assume that there are n classes, then the function to be modeled takes the following form:

$$f : R^l \Rightarrow \{0, 1, 2, \ldots, n\}$$

- There are different machine learning models to learn this type of functions:
    - Decision trees and random forests;
    - Univariate and multi-varied logistics functions;
    - K-nearest neighbor;
    - Deep Learning (which is a separate field);

Clustering (unsupervised)

- When we face clustering problems, we assume that we have unlabeled data available, from which we can derive approximate rules for labeling new data.

- Now let's look at an example, where all the dots are white. Although we do not have a priori knowledge of which points belong to which classes, we can clearly identify geometric patterns that indicate, in this case, that we are in the presence of two groups (ie clusters) for distinct!

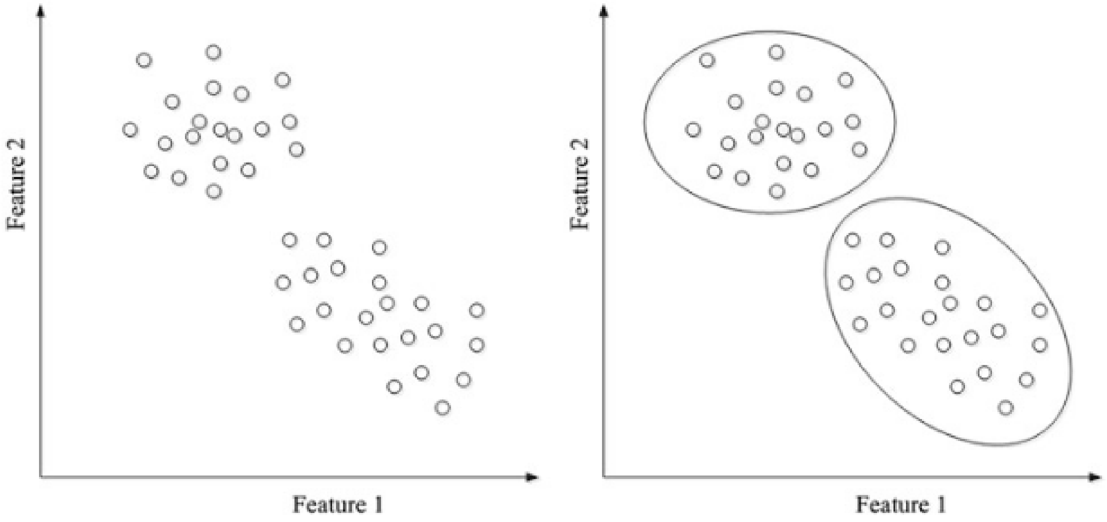


Fig. 1.4 Clustering is defined

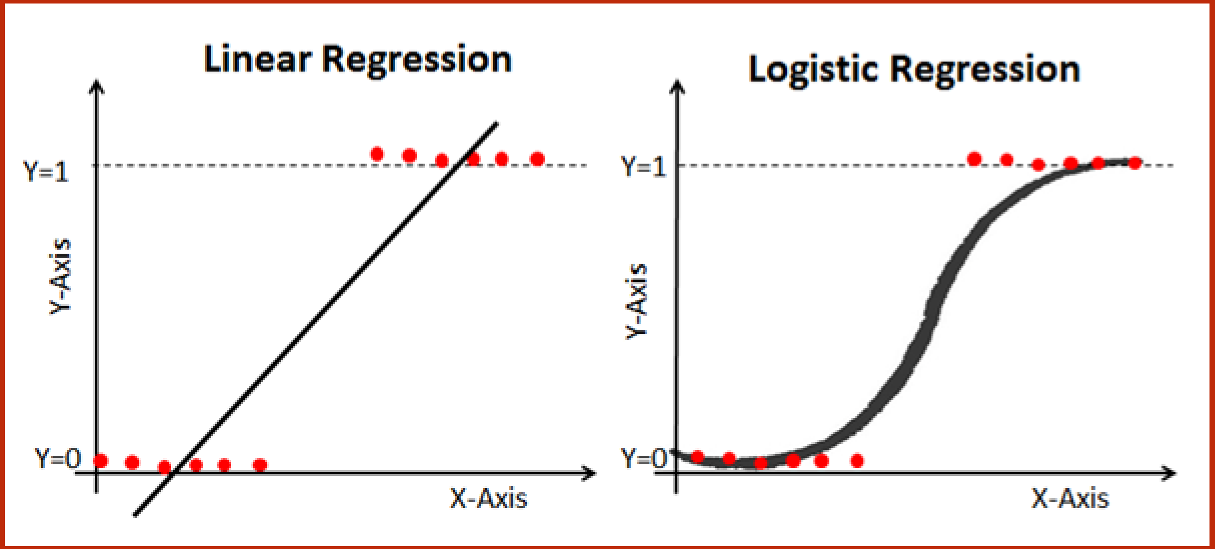
- As before, a classification process supposes to have data belonging to a certain domain D formed by the relation R_l, where l indicates the number of features available!

- If we assume that we can extract n ^ classes, then the function to be modeled takes the following form:

$$\hat{f} : R^l \Rightarrow \{0, 1, 2 \ldots, \hat{n}\}$$

- There are different machine learning models to learn this type of functions:
    - K-means clustering;
    - Hierarchical clustering.

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA



Logistic regression

Lorenzo Stacchio

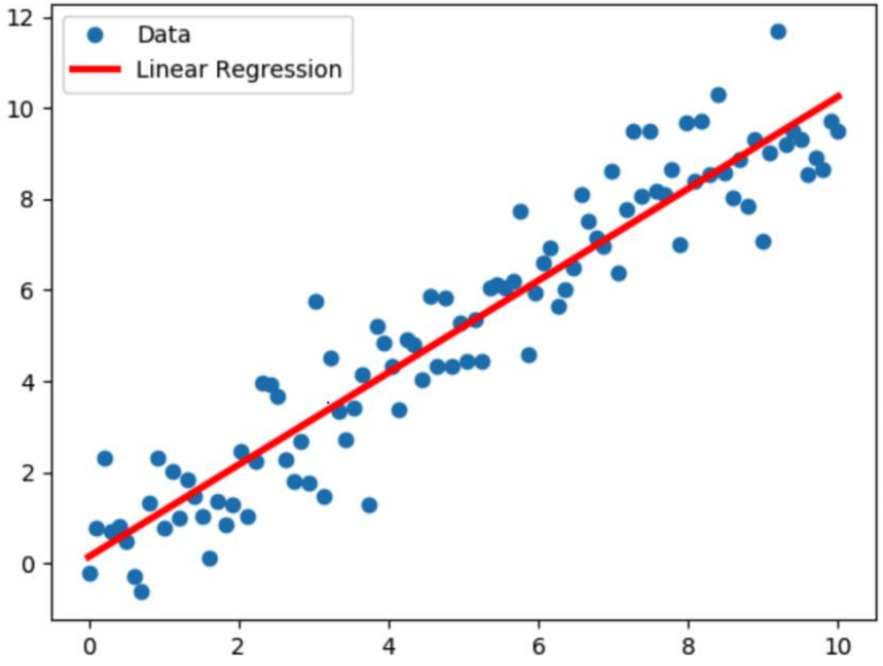
PhD student in Computer Science

Department for Life Quality studies

Regressione lineare (univariata)

- Linear regression attempts to model the relationship between two variables using a linear equation (= a straight line) to the observed data;

- One variable is considered an independent variable (to be exploited to predict, for example your income) and the other is considered a dependent variable (to be predicted, for example your expenses).

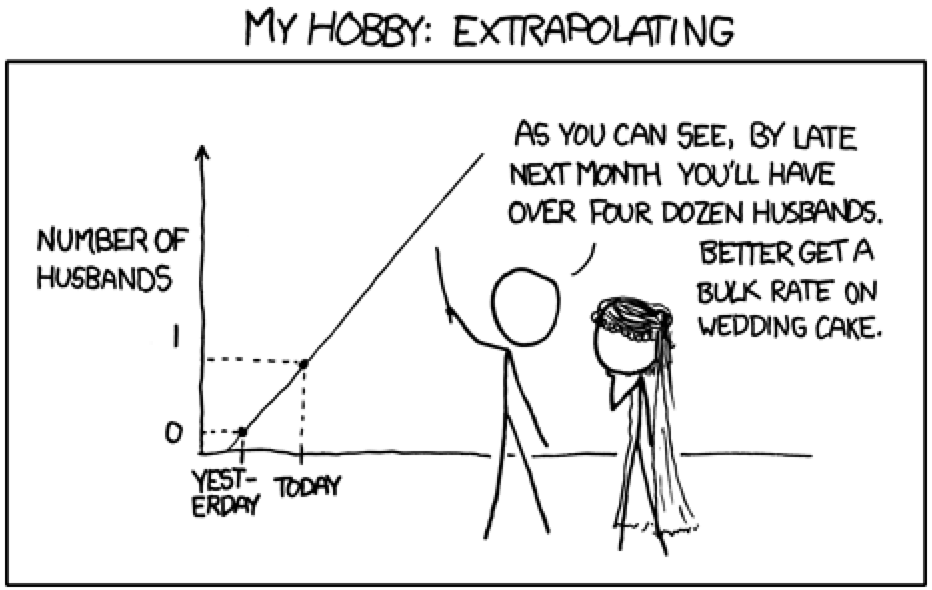


When to use Linear Regression?

- From a machine learning standpoint, it's the simplest model you can try on your data;

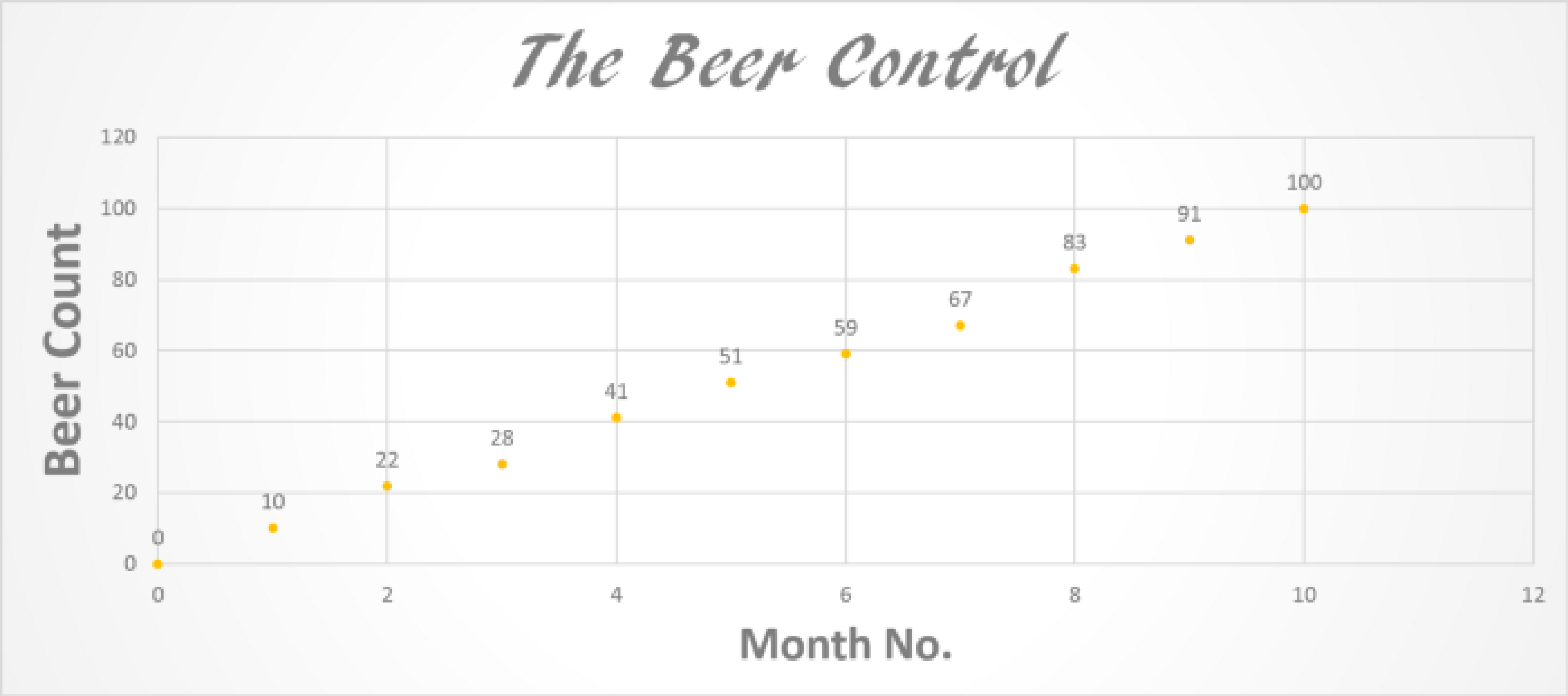
- If you feel that the data follows a linear trend, linear regression can give you fast and reasonably accurate results;

- Generally, complex data does not have a linear nature, and therefore linear regression is used in very few and controlled contexts;


MY HOBBY: EXTRAPOLATING
NUMBER OF HUSBANDS
1
0
YEST-ERDAY
TODAY
AS YOU CAN SEE, BY LATE NEXT MONTH YOU'LL HAVE OVER FOUR DOZEN HUSBANDS. BETTER GET A BULK RATE ON WEDDING CAKE.

Linear regression finds the best parameter to represent the line

- Let's try to understand what happens during the linear regression, taking as an example the beers per month drunk by an average graduate student on his way;



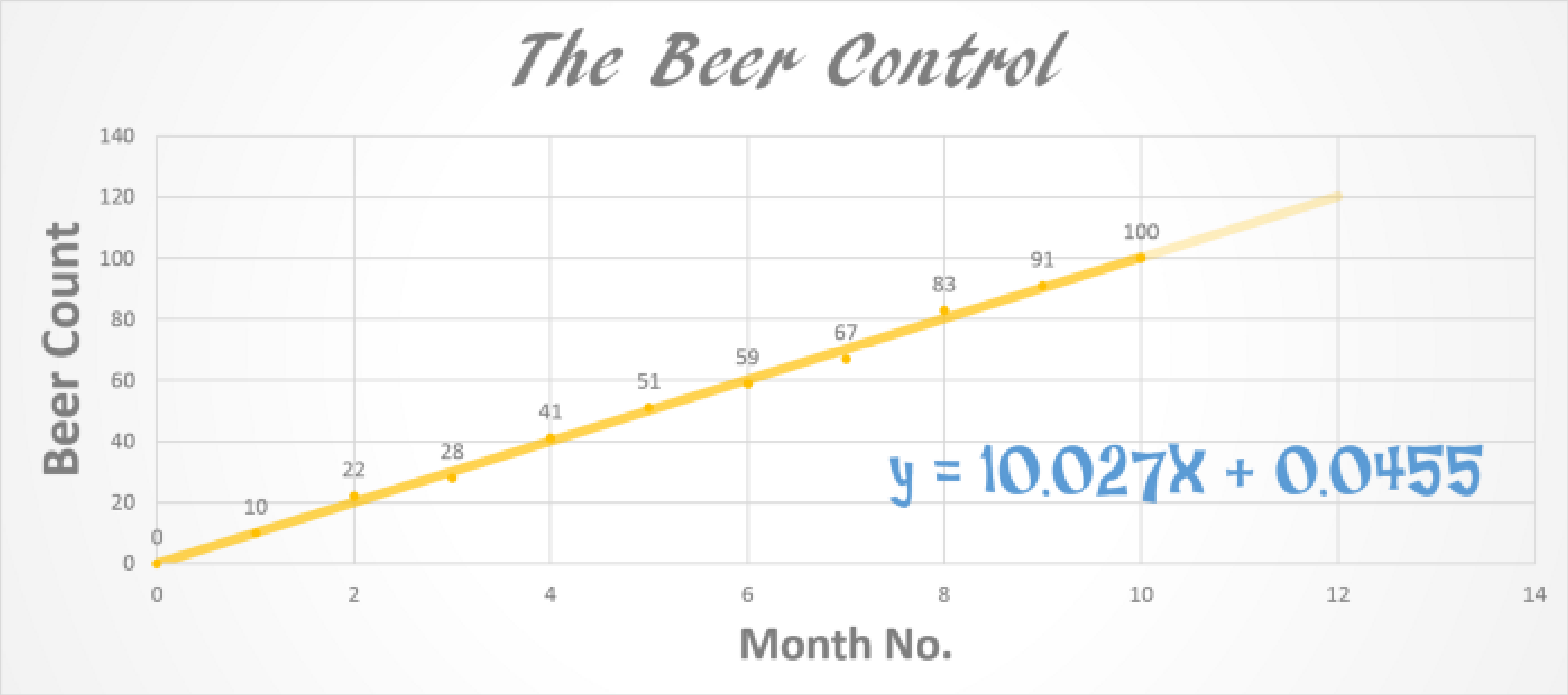
- The equation of a line is defined as $Y = (m*x) + q$;

- **Y is the dependent variable (Beer count), whose value is derived from the product between m and X to which q is then added;**

- **X is the value of the independent variable (Month Number);**

- **m is defined as the angular coefficient, that is the parameter that gives the "slope" of the line;**

- **q is instead the intercept and defines the deviation value of the function from the origin and corresponds to the intersection point of the line with the ordinate axis (y);**



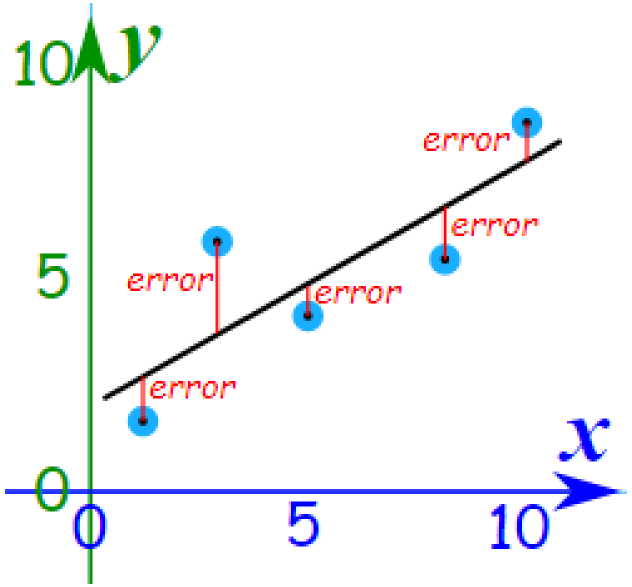
- What happened?
- **We have defined the best values for m and q for the available data;**
- **How to measure when these values are good?**
- **We have to use a metric!**

Least squares method: optimizing "residuals"

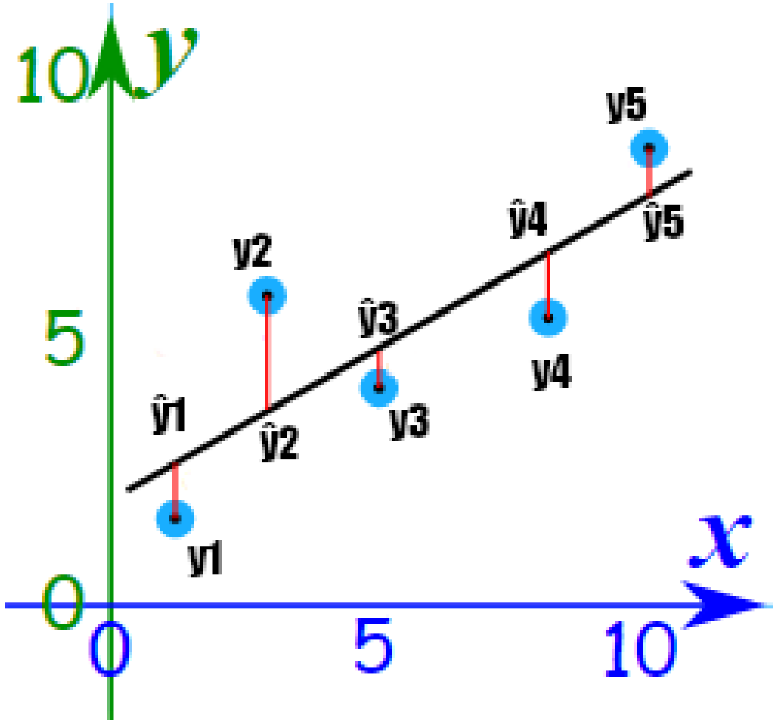
- As you have surely already guessed, we want our tuition to be as "in line" with our data as possible;

- For this, we would like to have a metric that measures how well our line defines our data set, looking at the proximity of the predicted points to the real ones, described by the dependent variable and predicted according to the independent variable!

- The distances between the predicted and real points are called residuals;



Mean square error



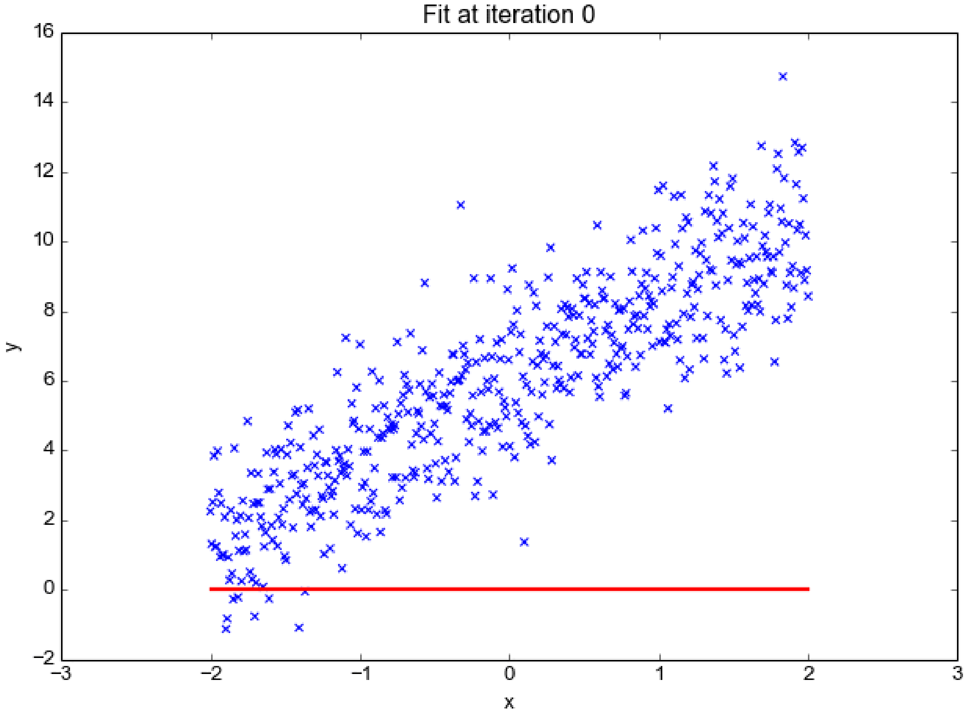
$$RMSE = \sqrt{\sum_{i=1}^{n} \frac{(\hat{y}_i - y_i)^2}{n}}$$

- When we accumulate these errors (difference between predicted and real point) on all the data at our disposal we can calculate what is defined as RMSE (root mean square error);

- RMSE is a frequently used measure of the differences between the values (samples or population values) predicted by a model and the actual observed values;

- We want to minimize this error!

Error minimization



- Through optimization algorithms (gradient descent) also used to teach neural network models, we can choose the "best line", i,.e., the parameters that minimize the distance between the line and the points in our data domain;

Multivariate linear regression?

Linear regression: a regression method

- We're talking about classification… so why explain what linear regression is?

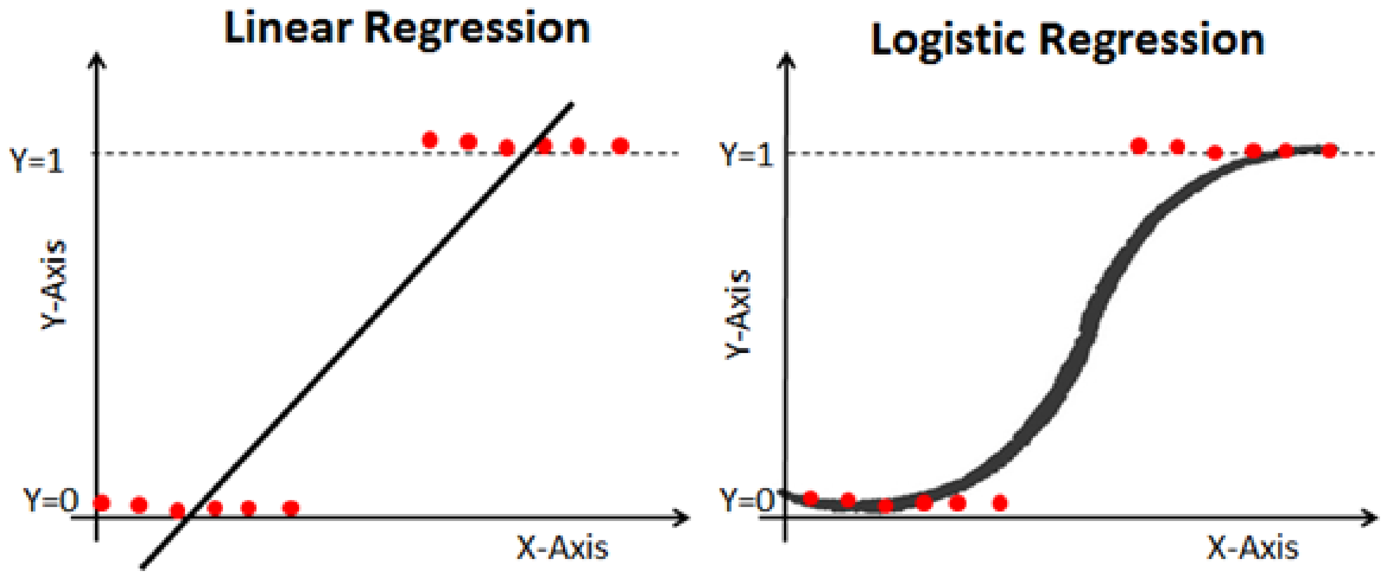
- **Because the linear regression method is used in the simplest classification model in existence: logistic regression!**

- **Classification is the problem of predicting an output of a discrete class (dog vs cat; shirt vs pants);**

- **Regression is the problem of predicting an output described by a continuous variable (price of a house, price of a dress over time).**

Logistic regression: the naive method for classification

- Classification models are created to provide discrete answers based on any type of input;
- There are various classification models, the first model, which only allows you to perform a binary classification (eg spam / non spam) is the logistic regression;
- In fact, classical linear regression is not suitable for discriminating between two classes as the model is optimized on a continuous and non-discrete data domain!



- We can initially refer to logistic regression as a classic equation, where Y represents the discrete response (e.g., true / false, 0/1);

- Y is related to the data domain X with a product with the parameter a.

$$Y = aX$$

- However, in this equation, **X can be continuous or discrete while Y is discrete;**

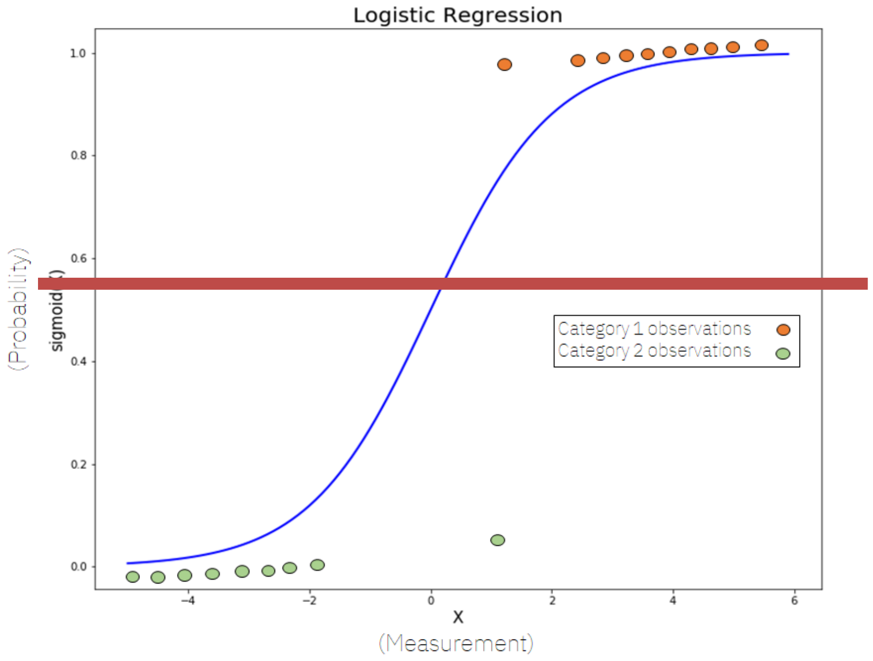
- In a geometric sense, this relationship forms a scale effect making it difficult to fit a mathematical model to the data;

- Therefore, we need to define a new intermediate equation and finally apply what is defined as the threshold value to bring out the final discrete value (true / false).

- Dense logistic regression is not a straight-line model, but a so-called logistic (or sigmoid) function;

$$sigmoid(x) = \frac{1}{1 + e^{-\boxed{x}}} \longleftarrow aX$$

- This function maps any value, into a value between 0 and 1, which is a probability! Based on the threshold value (0.5), the probability is converted to its extreme values (0 and 1, spam and not spam, true and false).



Threshold

- What is the x in the formula?

$$x = \theta \times feature + b$$

$$sigmoid(x) = \frac{1}{1 + e^{-(\theta \times feature + b)}}$$

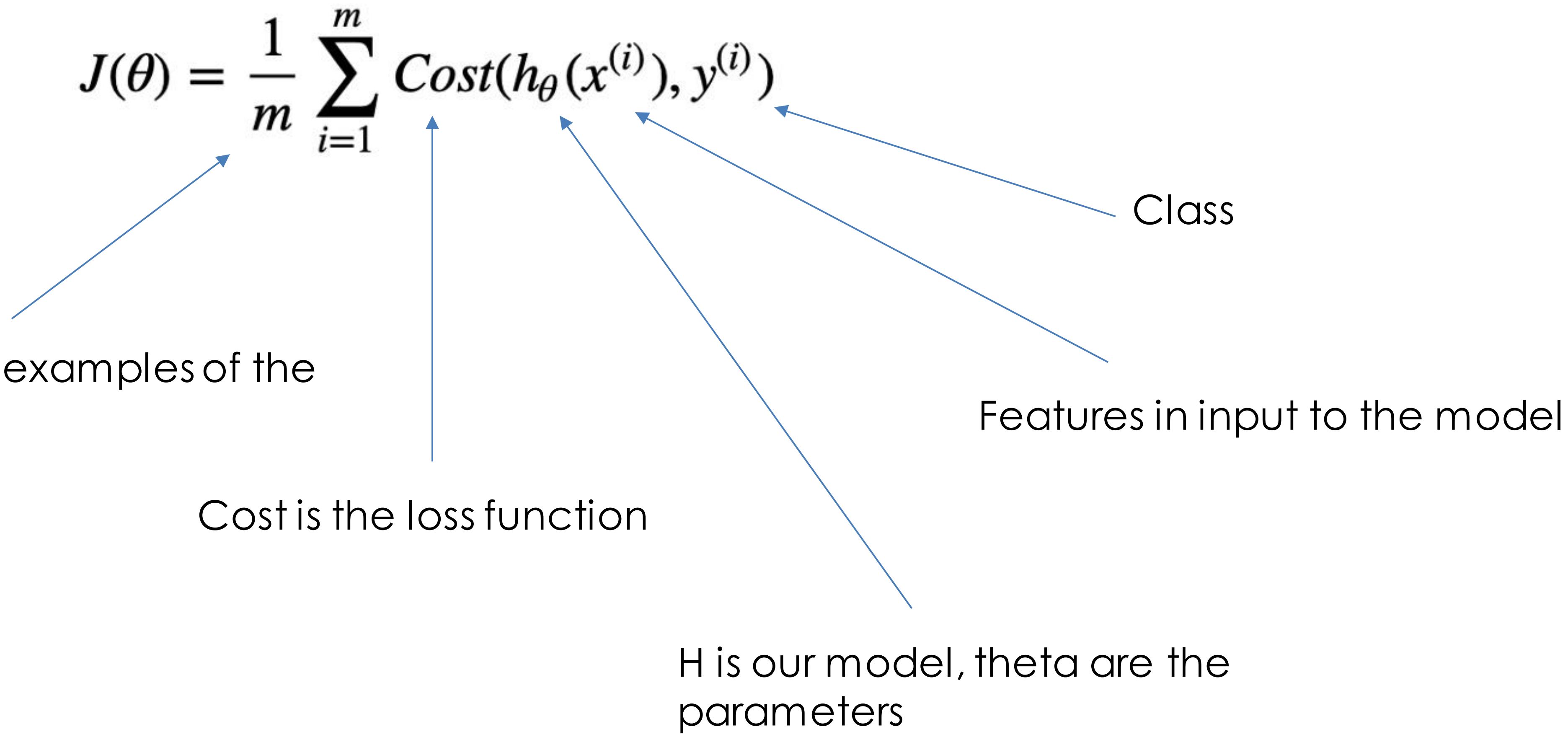
- What does this formula remind you of?

- What can we get from the fact that this is exactly the function of a line? In reality the parameters are optimized on a continuous intermediate step, which is then mapped into a probability value to classify our example into two classes!

- How do we train it? With iterative optimization methods such as gradient descent;

- What is the loss function we need to minimize?

Binary cross-entropy



$$J(\theta) = \frac{1}{m} \sum_{i=1}^{m} Cost(h_\theta(x^{(i)}), y^{(i)})$$

Class

M is the number of examples of the considered set

Features in input to the model

Cost is the loss function

H is our model, theta are the parameters

$$Cost(h_\theta(x), y) = \boxed{-ylog(h_\theta(x))} - \boxed{(1 - y)log(1 - h_\theta(x))}$$

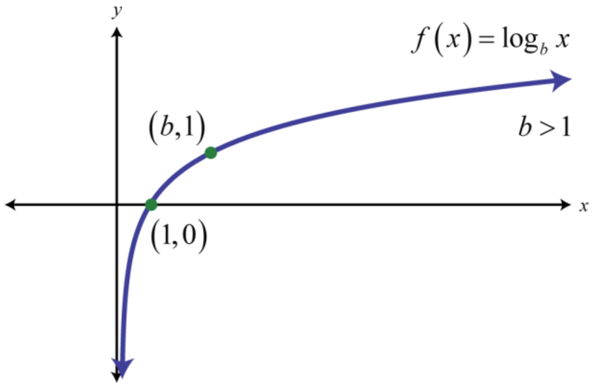
Y is the class, when y is 0 only this term is active

Y is the class, when y is 1 only this term is active

H is our model, theta are the parameters

Why is this formula correct?

$$-y\log(h_\theta(x))$$



- When y = 1 (dog) we would like this term to be equal to 0;

- The logarithm is a positively defined function which however has a negative range when it has values between 0 and 1;

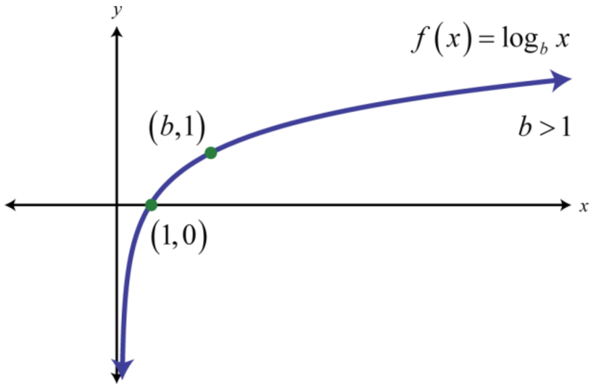
- When the value is 1 instead, its logarithm corresponds to 0;

- This is exactly what we want!

- **When the logarithm is 0, we have correctly classified the class, when it is below zero, the cost function will grow according to the magnitude of the error (remember that the product between negatives is positive).**

Why is this formula correct? Pt. 2

$$-(1-y)log(1-h_\theta(x))$$



- When y = 0 (cat) we would like this term to be equal to 0;
- The logarithm is a positively defined function which however has a negative range when it has values between 0 and 1; When the value is 1 instead, its logarithm corresponds to 0;
- **When the model returns 0 the logarithm is 0! And so we have correctly classified the class;**
- **If the class returns a value other than 0, the logarithm value will be negative, thus providing a positive addition to the cost function!**

Multi-varied linear/logistic regression: what happens if we have more than one feature?

- What is the x in the formula?

$$x = \theta \times feature + \theta_2 \times feature_2 + \theta_3 \times feature_3 \ldots + \theta_n \times feature_n + b$$

Let's code!

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Boomer vs Non boomer

Lorenzo Stacchio

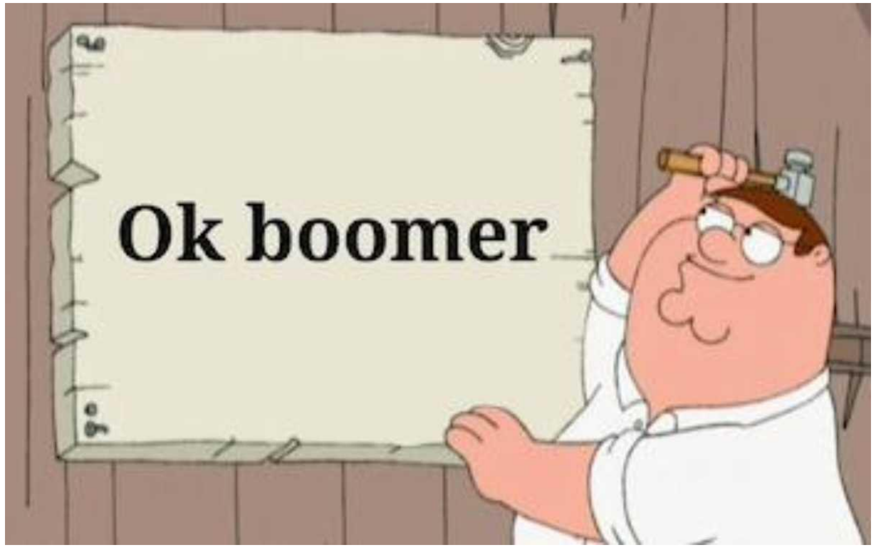
PhD student in Computer Science

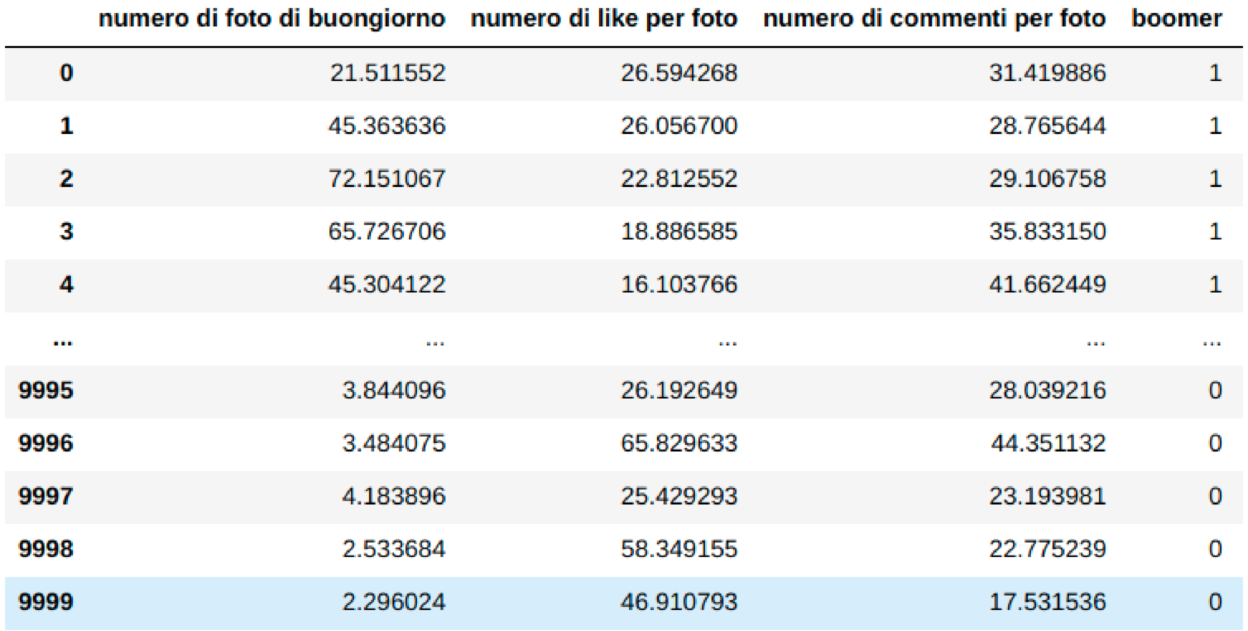
Department for Life Quality studies

Logistic regression to classify a boomer from a non-boomer

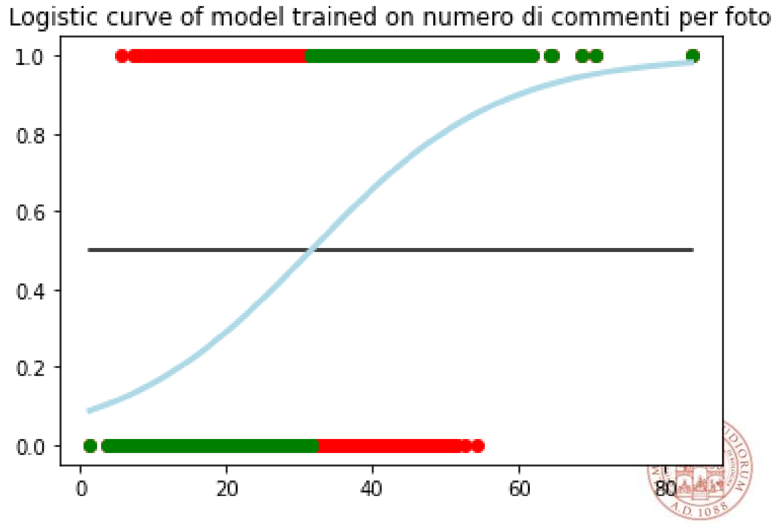
- To have fun together, I created a synthetic dataset assuming the existence of features that were discriminatory or not to distinguish a boomer in a generic social network;

- Is the very fact of creating a synthetic dataset to recognize boomers considered boomer stuff? Probably yes.



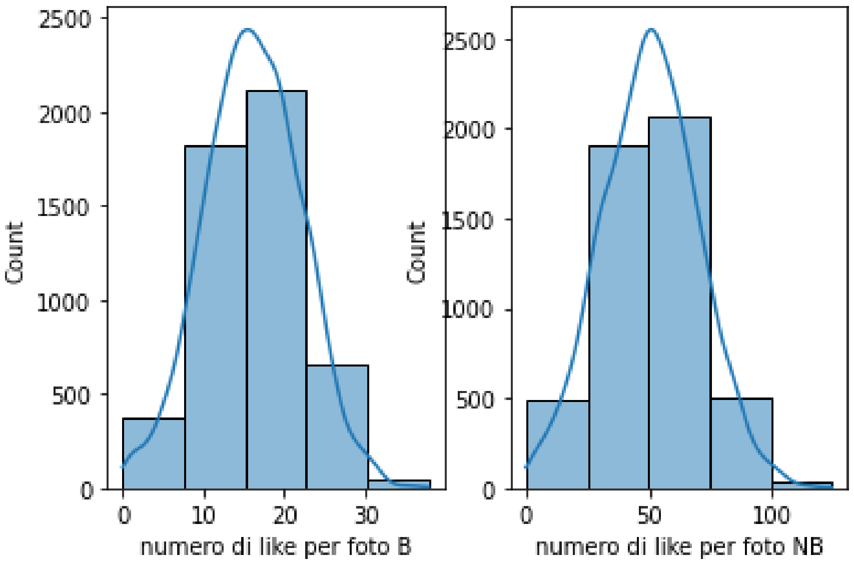


	numero di foto di buongiorno	numero di like per foto	numero di commenti per foto	boomer
0	21.511552	26.594268	31.419886	1
1	45.363636	26.056700	28.765644	1
2	72.151067	22.812552	29.106758	1
3	65.726706	18.886585	35.833150	1
4	45.304122	16.103766	41.662449	1
...	...	...	...	...
9995	3.844096	26.192649	28.039216	0
9996	3.484075	65.829633	44.351132	0
9997	4.183896	25.429293	23.193981	0
9998	2.533684	58.349155	22.775239	0
9999	2.296024	46.910793	17.531536	0

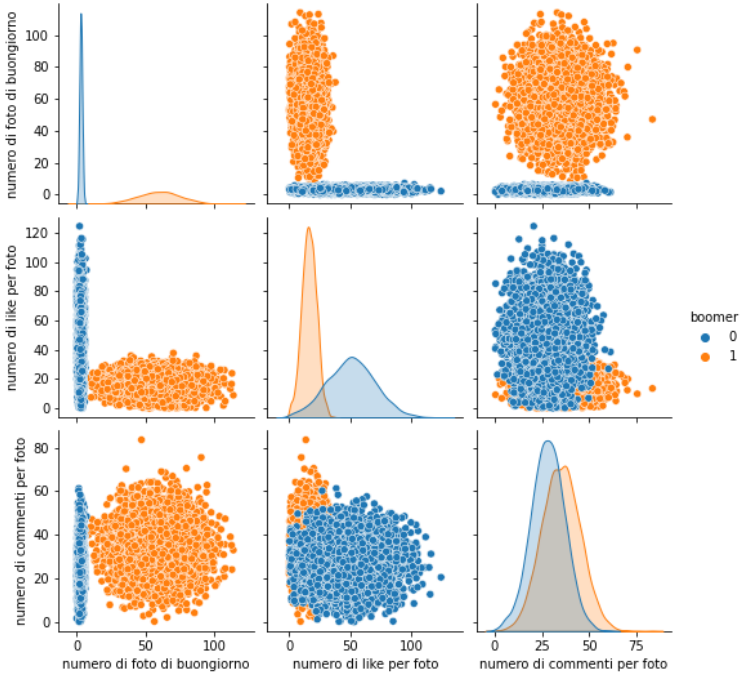


Let’s colab it!

Data Visualization and explorative analysis

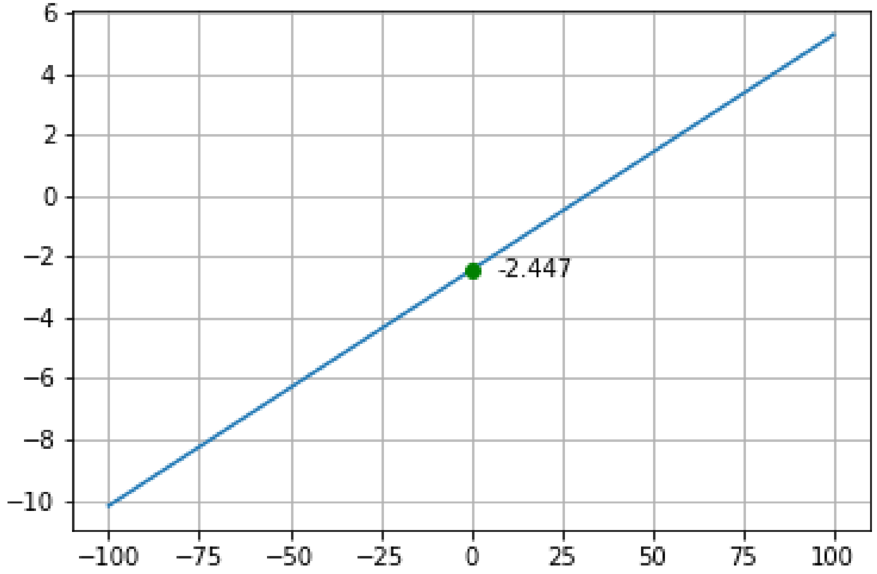


Histograms

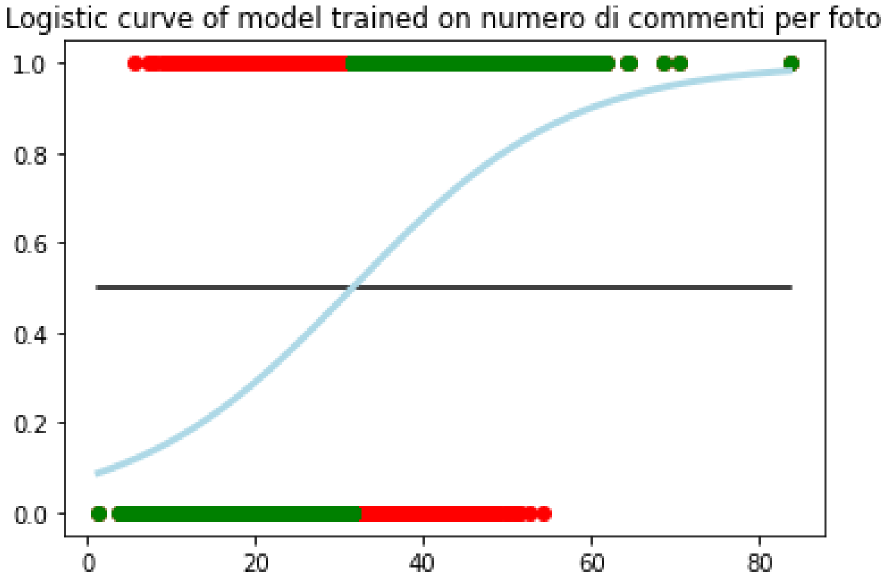


Pair plots

Logistic regression



Linear coefficients



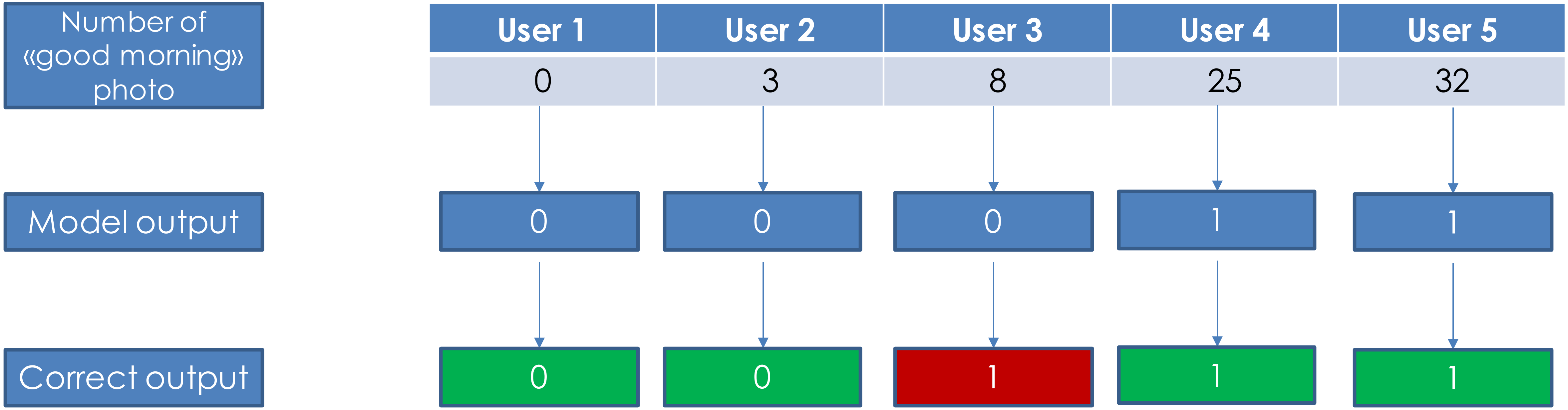
Sigmoid & Threshold

Measuring the performance of a classifier: accuracy

- Accuracy is a metric for evaluating classification models.

- Informally, accuracy is the fraction of predictions that our model has correctly obtained. Formally, accuracy has the following definition:

$$Accuracy = \frac{Numero\ di\ predizioni\ corrette}{Numero\ totale\ di\ predizioni}$$



Number of «good morning» photo	User 1	User 2	User 3	User 4	User 5
	0	3	8	25	32
Model output	0	0	0	1	1
Correct output	0	0	1	1	1

- Our model got 4 out of 5 predictions right, so we have an accuracy of 4/5 = 0.8, or 80%;

- However, the accuracy has limits and works optimally only to the extent that we are in a dataset situation with balanced classes;

- In fact, accuracy alone does not tell the right story about the performance of our model with a dataset with unbalanced classes, let's try to understand this concept with an example;

True Positives, True Negatives, False Positives e False Negatives

- A true positive is a result where the model correctly predicts the positive class. Likewise, a true negative is a result where the model correctly predicts the negative class.

- A false positive is a result where the model incorrectly predicts the positive class. And a false negative (false negative) is a result where the model incorrectly predicts the negative class.

- In our case, the positive class corresponds to being a boomer and the negative class to not being! Suppose we have 55 examples of boomers and 5 of non-boomers;

		Labels of the considered dataset	
		Boomer	Non boomer
Model Prediction	Boomer	True positive: 55	False positive : 5
	Non boomer	False negative: 0	True negative: 0

- Considering the last example, the accuracy would correspond to a very high value, equal to 55/60 = 0.92, or 92%

- However, we see that there is a clear problem: all errors by the classifier are false positives;

- All examples of the Non-Boomer class have been classified as Non-Boomer by the classifier;

- We could not have appreciated this phenomenon by looking only at accuracy, and above all we could not have run for cover to solve the problem!

- What do you think is the problem? The classifier has clearly overfitted on the Boomer class!

Precision, Recall and F1-score

- In these cases of imbalance, it is good to find deeper metrics than simple accuracy!

- For this purpose, there are precision and recall!

- Precision: What proportion of positive ratings was correct?

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

- In our example:
  - **TP = 55;**
  - **FP = 5;**
  - **Precision = 55/60 = 0.92**

- Recall: What proportion of correct positives was correctly identified?

$$\text{Recall} = \frac{TP}{TP + FN}$$

- In our example:
  - **TP = 55;**
  - **FN = 0;**
  - **Recall = 55/55 = 1**

- **F1: Harmonic mean between precision and recall (to be used instead of accuracy)**

$$F_1 = 2 \times \frac{precision \times recall}{precision + recall}$$

- In our example:
  - **precision = 0.92;**
  - **recall = 1;**
  - **F1 = 0.96**

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Moment, moment, moment… Why are these results so good?

- Because we have considered the boomer class as a positive class!
- What would happen if we considered the non-boomers as a positive class?

		Labels of the considered dataset	
		Non boomer	Boomer
Model Prediction	Non boomer	True positive: 0	False positive: 0
	Boomer	False negative : 5	True negative: 55

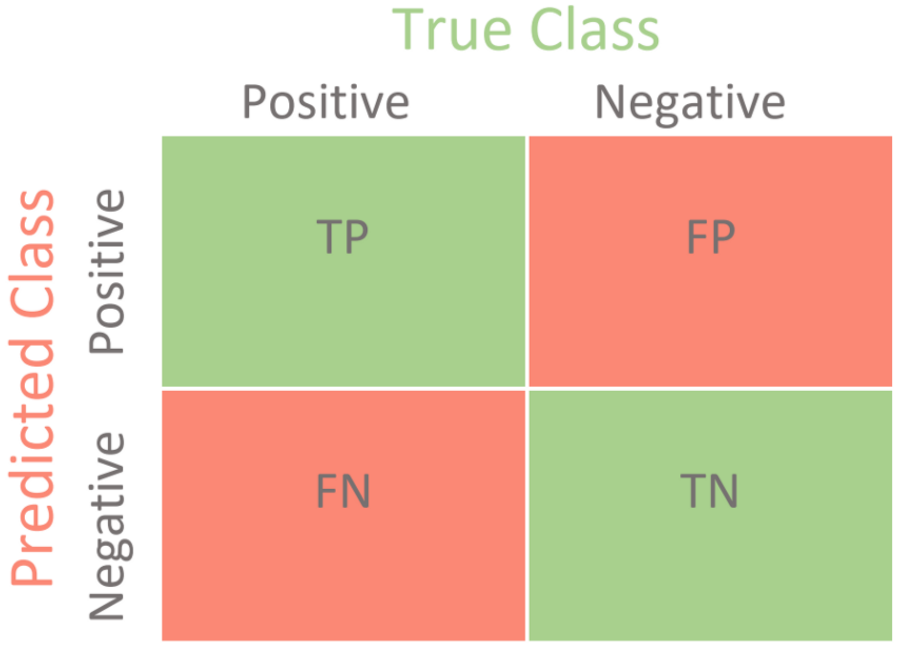
- **Precision = 0/0 is impossible, it is considered as 0;**
- **Recall = 0/5 = 0;**
- **F1- score = 0**

Confusion matrix

- The Confusion Matrix is a tabular way to visualize the performance of the prediction model.

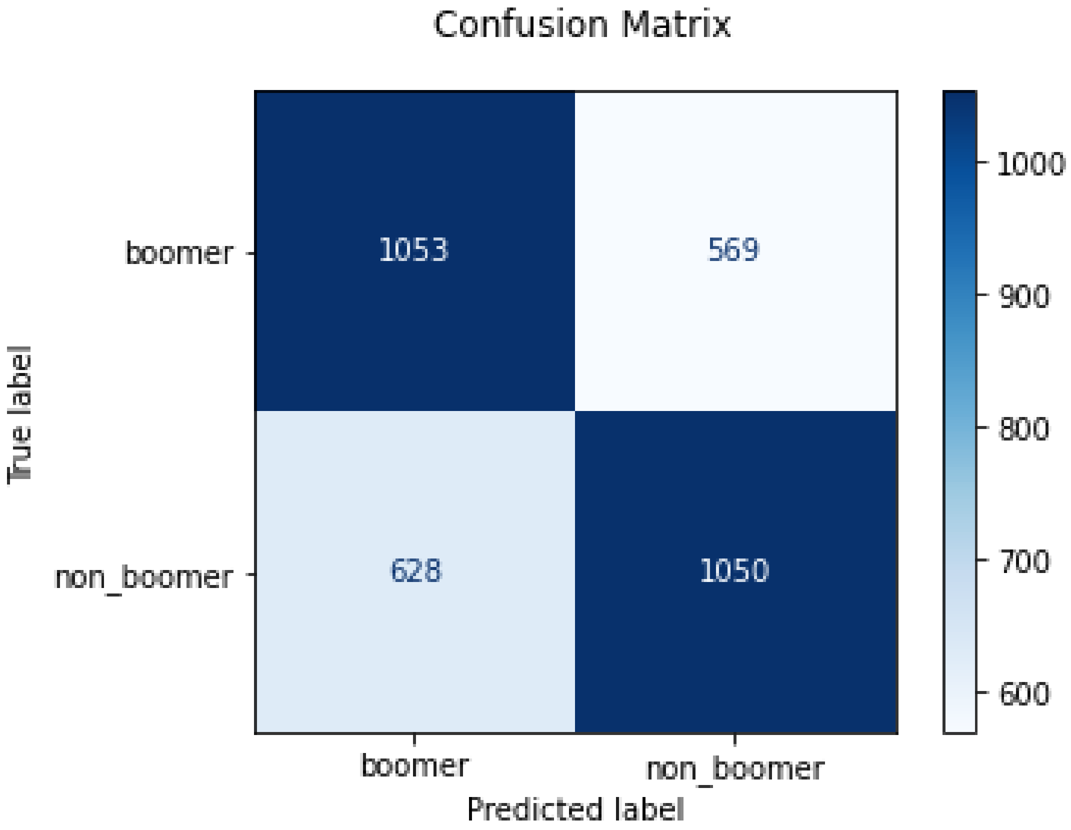
- Each entry in the confusion matrix denotes the number of predictions made by the model into which it classified the classes correctly or incorrectly.

- **It is based on the concepts we have just seen!**

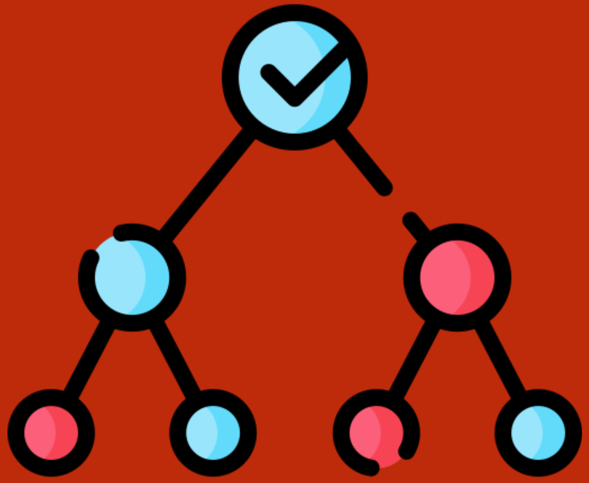


		True Class	
		Positive	Negative
Predicted Class	Positive	TP	FP
	Negative	FN	TN

Confusion matrix on our boomer vs non boomer classifier



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA



Decision trees

Lorenzo Stacchio

PhD student in Computer Science

Department for Life Quality studies

Decision Tree

- In the world of Machine Learning, decision trees **are a kind of non-parametric models**, which can be used for both classification and regression;

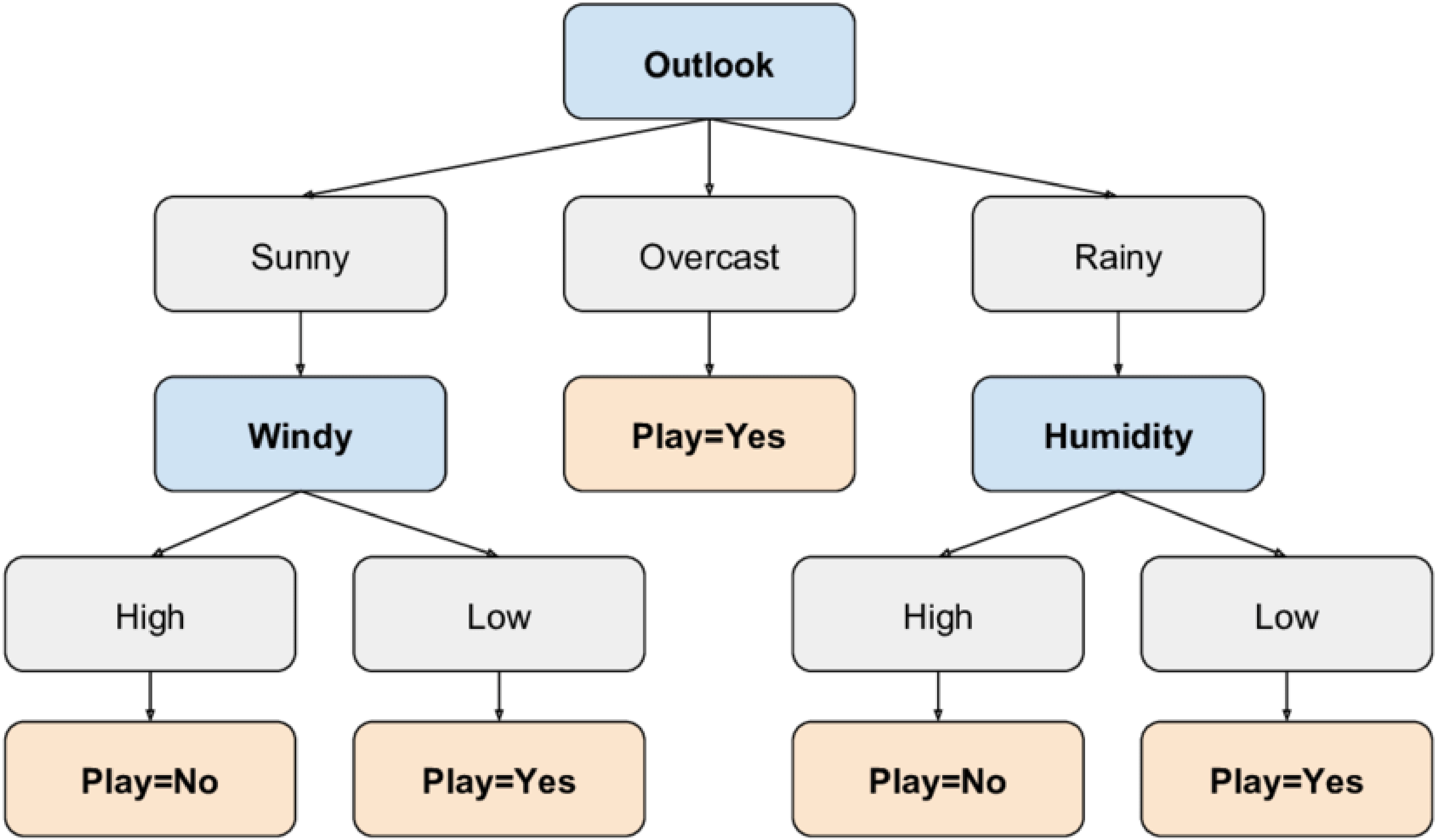
- These models are therefore **flexible** since they do not increase the number of parameters when we add more features (if we build them correctly) and can produce a categorical forecast (such as whether a plant is of a certain type or not) or a numerical forecast (such as the price of a house);

- They are built using two types of elements: **nodes and branches**. At each node, one of our data features is evaluated to break down the observations in the training process or to make a specific data point follow a certain path when making a prediction.

- In practice, for each feature, a boolean test is carried out to understand which path we will have to take to provide a correct classification!

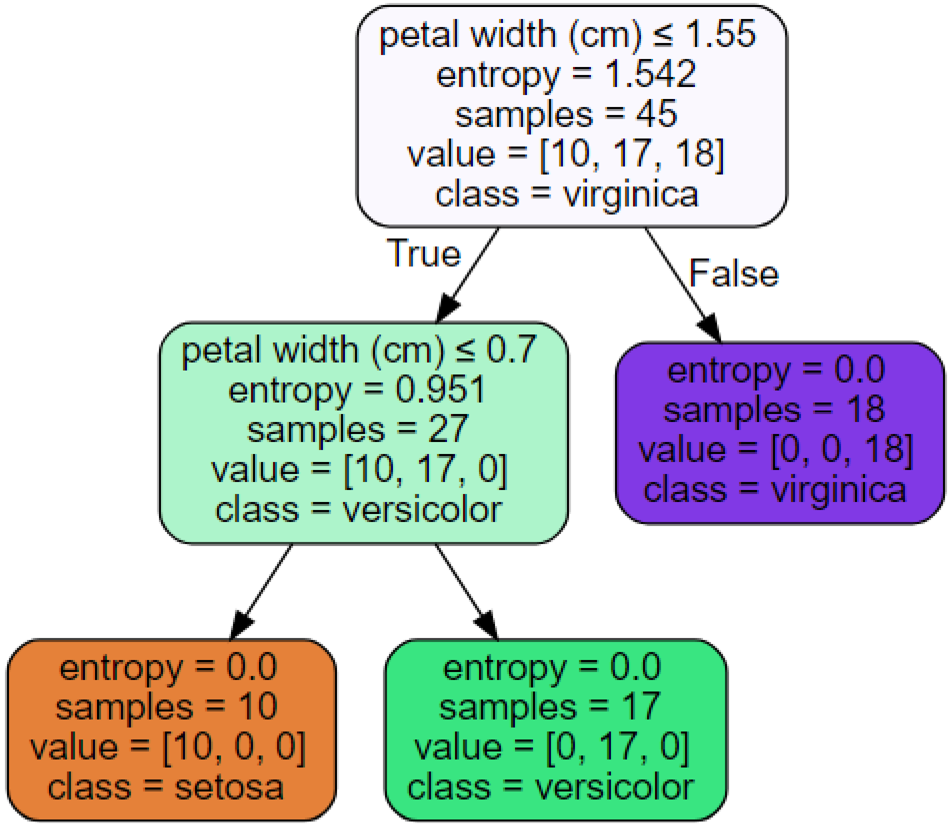
- In case of categorial variables, each node will evaluate the equality respect to a discrete value, and in case of numerical values, each node will compare its values respect to a calculated threshold;

Decision tree with categorical features: Tennis match

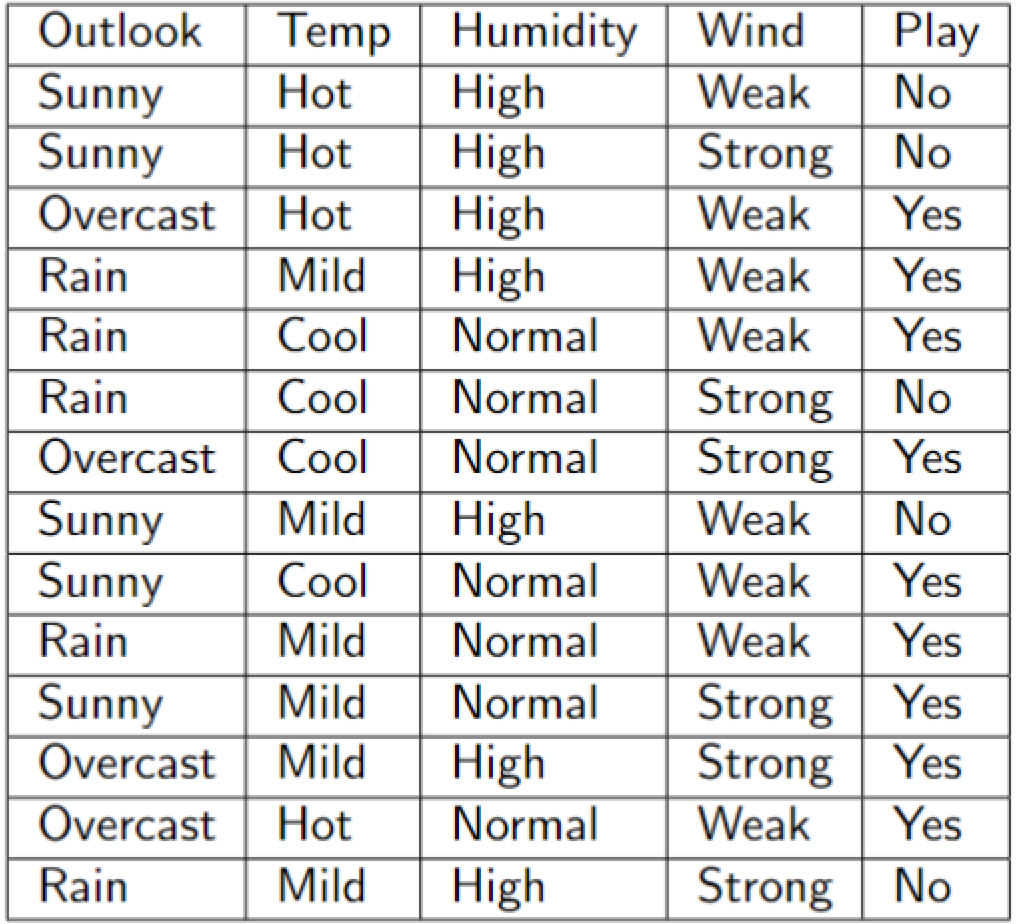


Source

Decision tree with numerical features: Iris



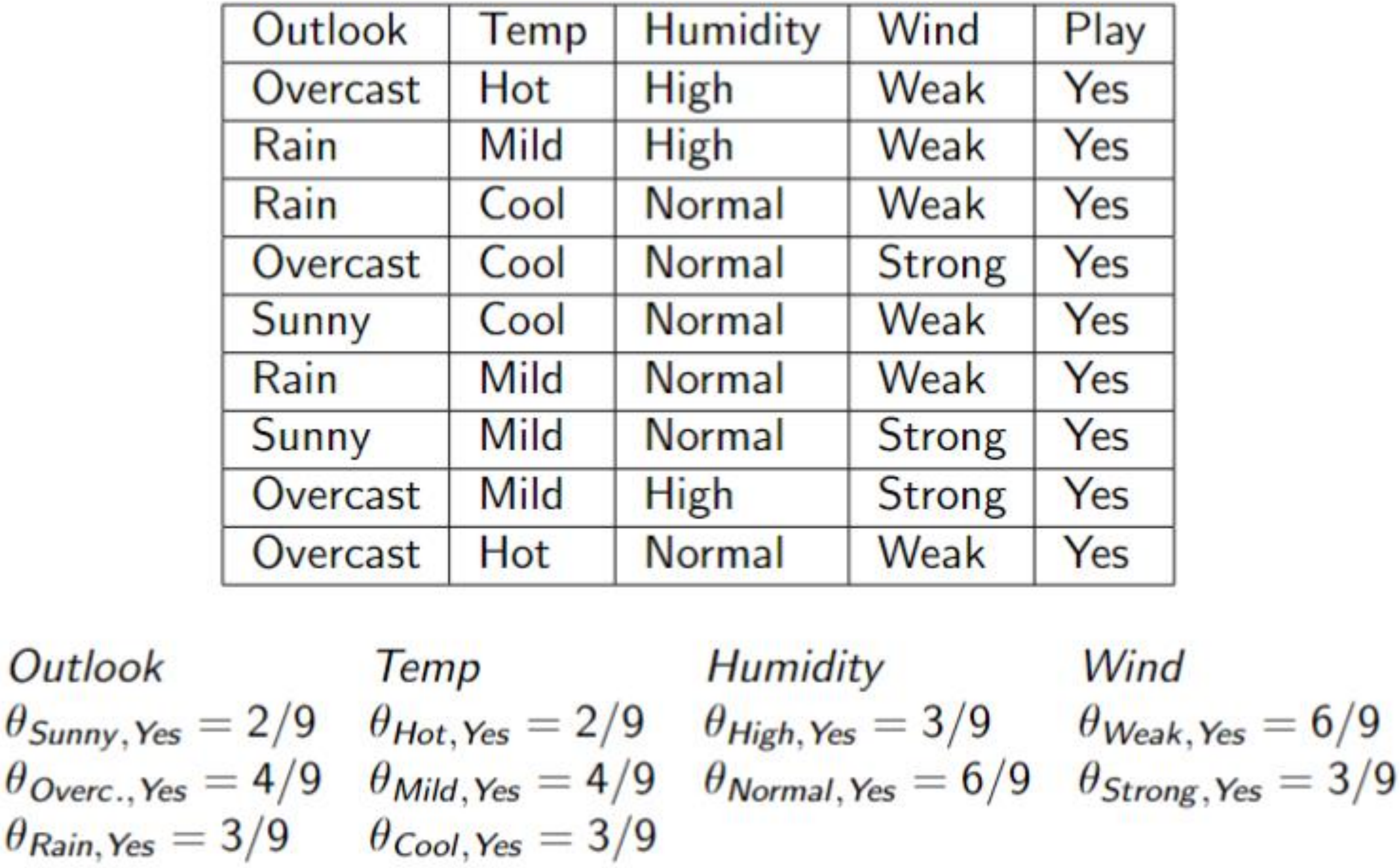
How to build a decision tree? It's a matter of probability



Outlook	Temp	Humidity	Wind	Play
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	No

- Starting from this Table, we can guess the mathematical reasoning behind how decision trees are built;

- What is the probability to play tennis if we consider just the outlook?



Outlook	Temp	Humidity	Wind	Play
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Overcast	Cool	Normal	Strong	Yes
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes

Outlook

$\theta_{Sunny,Yes} = 2/9$

$\theta_{Overc.,Yes} = 4/9$

$\theta_{Rain,Yes} = 3/9$

Temp

$\theta_{Hot,Yes} = 2/9$

$\theta_{Mild,Yes} = 4/9$

$\theta_{Cool,Yes} = 3/9$

Humidity

$\theta_{High,Yes} = 3/9$

$\theta_{Normal,Yes} = 6/9$

Wind

$\theta_{Weak,Yes} = 6/9$

$\theta_{Strong,Yes} = 3/9$

Outlook	Temp	Humidity	Wind	Play
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Rain	Cool	Normal	Strong	No
Sunny	Mild	High	Weak	No
Rain	Mild	High	Strong	No

Outlook

$\theta_{Sunny,No} = 3/5$

$\theta_{Overc.,No} = 0$

$\theta_{Rain,No} = 2/5$

Temp

$\theta_{Hot,No} = 2/5$

$\theta_{Mild,No} = 2/5$

$\theta_{Cool,No} = 1/5$

Humidity

$\theta_{High,No} = 4/5$

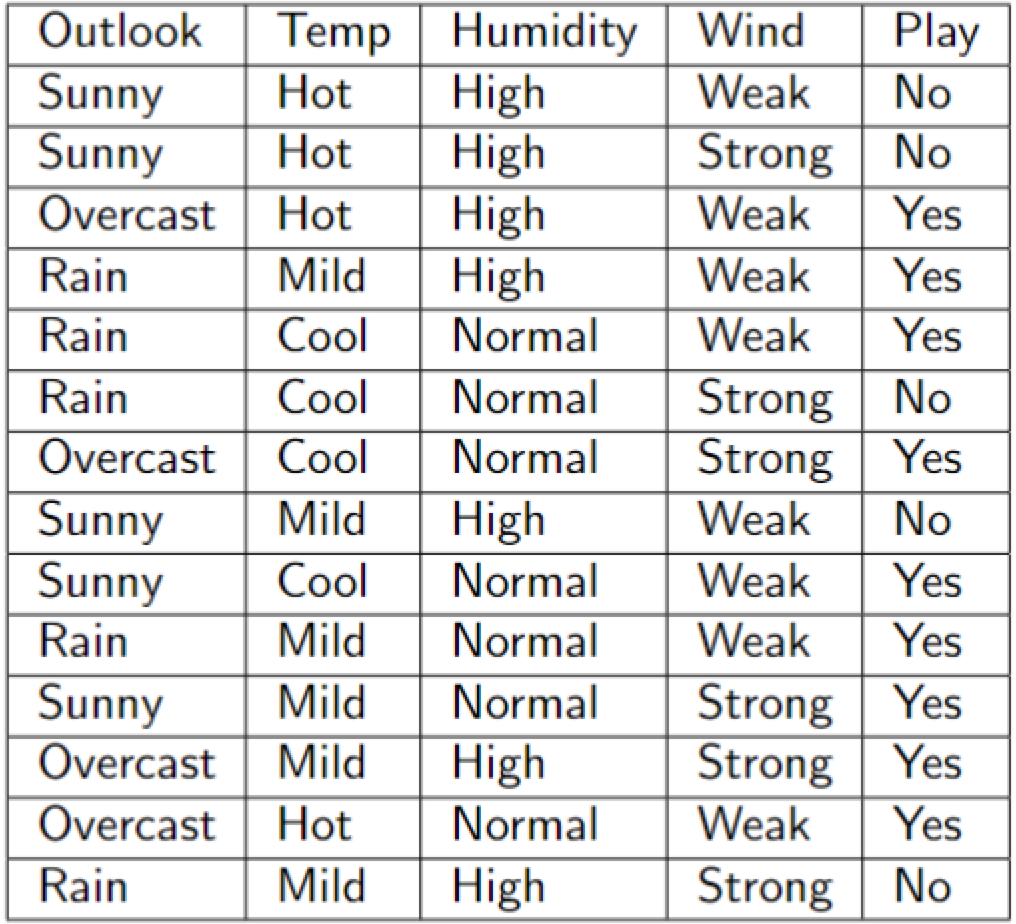
$\theta_{Normal,No} = 1/5$

Wind

$\theta_{Weak,No} = 2/5$

$\theta_{Strong,No} = 3/5$

How to build a decision tree? It's a matter of probability



Outlook	Temp	Humidity	Wind	Play
Sunny	Hot	High	Weak	No
Sunny	Hot	High	Strong	No
Overcast	Hot	High	Weak	Yes
Rain	Mild	High	Weak	Yes
Rain	Cool	Normal	Weak	Yes
Rain	Cool	Normal	Strong	No
Overcast	Cool	Normal	Strong	Yes
Sunny	Mild	High	Weak	No
Sunny	Cool	Normal	Weak	Yes
Rain	Mild	Normal	Weak	Yes
Sunny	Mild	Normal	Strong	Yes
Overcast	Mild	High	Strong	Yes
Overcast	Hot	Normal	Weak	Yes
Rain	Mild	High	Strong	No

- Starting from this Table, we can guess the mathematical reasoning behind how decision trees are built;

- What is the probability to play tennis if we consider just the outlook?

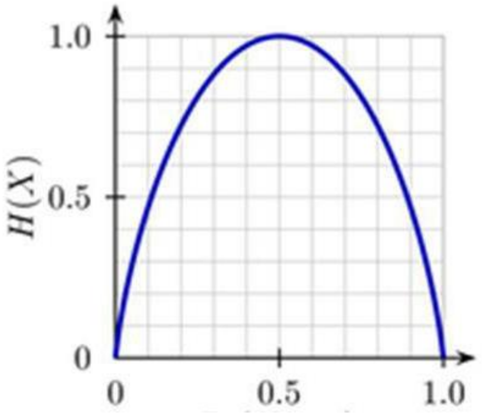
  - P(PlayTennis = “Yes” and Outlook Sunny): 2/9 = 23 %
  - P(PlayTennis = “Yes” and Outlook Rainy): 3/9 = 33%
  - P(PlayTennis = “Yes” and Outlook Overcast): = 4/9 = 44%

- What is the probability of play tennis if we consider the outlook and the temperature?

How to choose the order in which to evaluate the features? Entropy is often the answer!

- The entropy of information or Shannon entropy quantifies the amount of uncertainty (or surprise) involved in the value of a random variable;
- Mathematically, it is defined as the sum of the products between the probabilities of an event and the logarithm of the inverse probability, for all events associated with a causal variable;
- Its meaning in the decision tree allows us to estimate the impurity or heterogeneity of the variable we are considering;

$$\text{Entropy} = \sum \boxed{p(x)} \boxed{\log\left(\frac{1}{p(x)}\right)}$$



Probability associated with a certain event

Logarithm of the inverse of the probability of that event

Estimation of the surprise in the entire causal variable

- Suppose we consider a random variable (feature) that defines two events: heads or tails;
- Suppose this coin is rigged that 9 times out of 10 heads returns;
- We will therefore have the 90% probability of having heads and 10% of having tails;

$$Moneta = (testa\ , croce\ )$$

$$p\,(Moneta = testa\ ) = 0.9$$

$$p\,(Moneta = croce\ ) = 0.1$$

What is the degree of surprise behind each event?

Logaritmo

$$\boxed{\log}\boxed{\left(\frac{1}{p(x)}\right)}$$

Inversa della probabilità di un evento

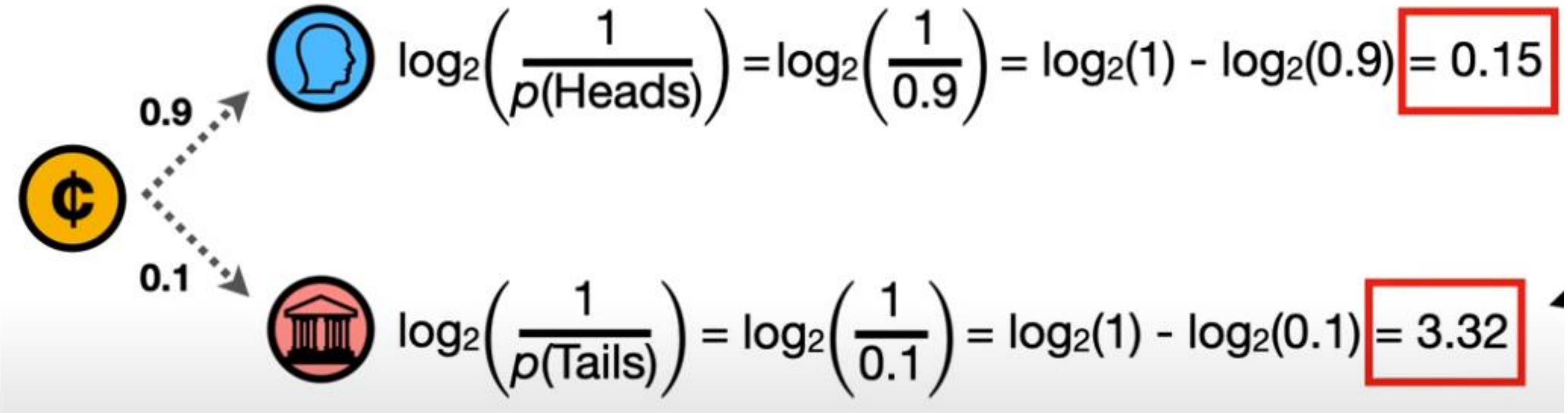
- The inverse of probability tries to provide significant information about the surprise, intuitively, the higher the probability, the less we would be surprised that that event happened (and vice versa);

- The logarithm is used to handle two problems:
  - if p (x) were 0, the operation would be impossible but by exploiting the properties of logarithms we can make it computable;

  - The logarithm function gives us a smoother function, allowing us to derive more precise calculations;

$$\log\left(\frac{1}{p(x)}\right) = \log(1) - \log(p(x))$$

N.B., the base of the logarithm depends on the number of events defined in x (in this case 2);



$$\log_2\left(\frac{1}{p(\text{Heads})}\right) = \log_2\left(\frac{1}{0.9}\right) = \log_2(1) - \log_2(0.9) \boxed{= 0.15}$$

$$\log_2\left(\frac{1}{p(\text{Tails})}\right) = \log_2\left(\frac{1}{0.1}\right) = \log_2(1) - \log_2(0.1) \boxed{= 3.32}$$

https://www.youtube.com/watch?v=YtebGVx-Fxw

Probability associated with an event and the concept of "surprise"

$$Moneta=(testa, croce)$$

$$p(Moneta=testa)=0.9$$

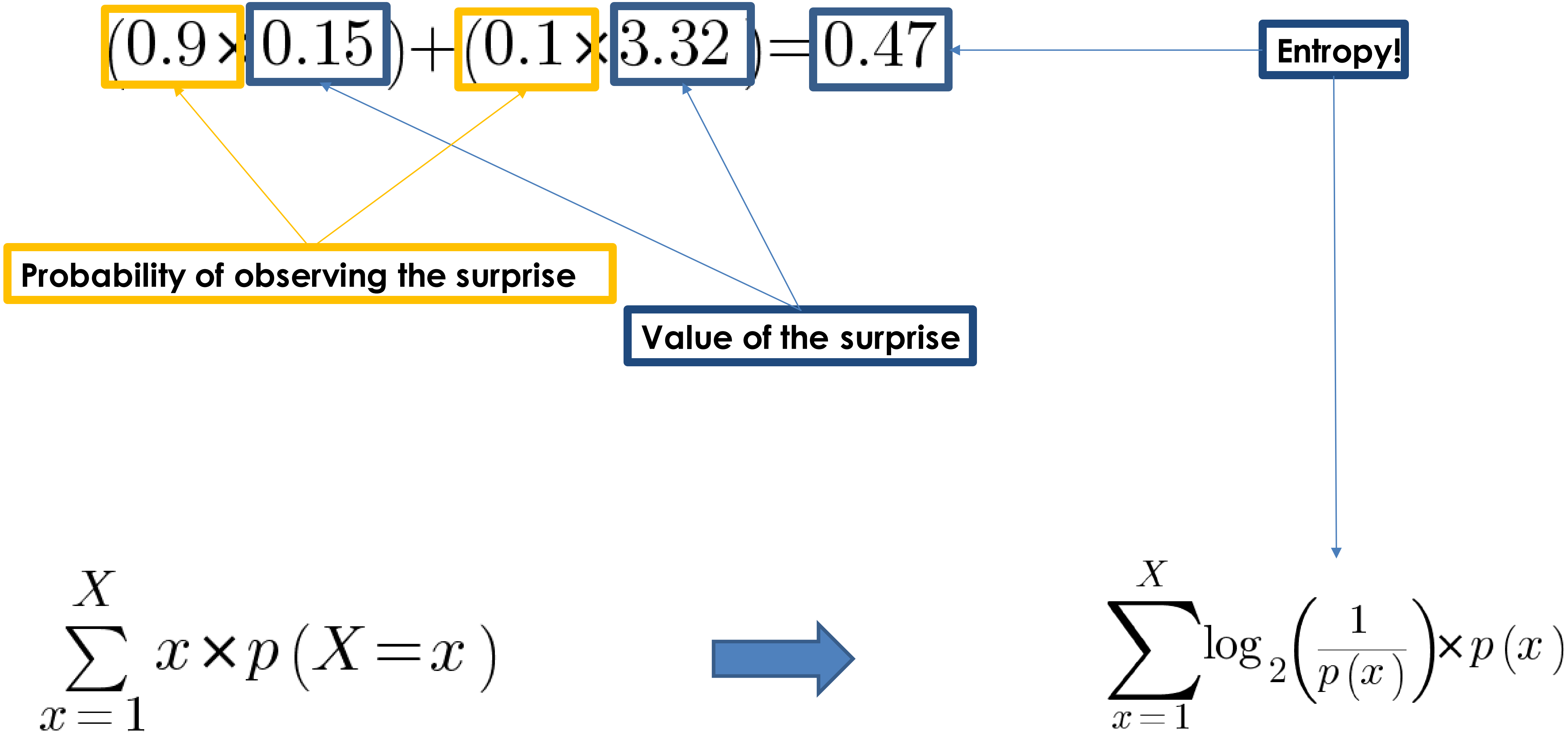
$$\log_2\left(\frac{1}{p(testa)}\right)=0.15$$

$$p(Moneta=croce)=0.1$$

$$\log_2\left(\frac{1}{p(croce)}\right)=3.32$$

- At this point, we want to calculate the surprise level for the random variable in its entirety!
- We can approximate the level of entropy, using the expected probability value formula;

$$\sum_{x=1}^{X} x \times p(X=x)$$



$$(\boxed{0.9}\times\boxed{0.15})+(\boxed{0.1}\times\boxed{3.32})=\boxed{0.47}$$

Entropy!

Probability of observing the surprise

Value of the surprise

$$\sum_{x=1}^{X} x\times p(X=x) \Rightarrow \sum_{x=1}^{X}\log_2\left(\frac{1}{p(x)}\right)\times p(x)$$

Entropy of an unfixed coin

$$Moneta = (testa, croce)$$

$$p(Moneta = testa) = 0.5$$

$$\log_2\left(\frac{1}{0.5}\right) = 1$$

$$p(Moneta = croce) = 0.5$$

$$\log_2\left(\frac{1}{0.5}\right) = 1$$

$$entropy(moneta) = (0.5 \times 1) + (0.5 \times 1) = 1$$

- We have maximum entropy!
- If you think about it, that's correct: at every coin toss, we never know what to expect!

Information gain

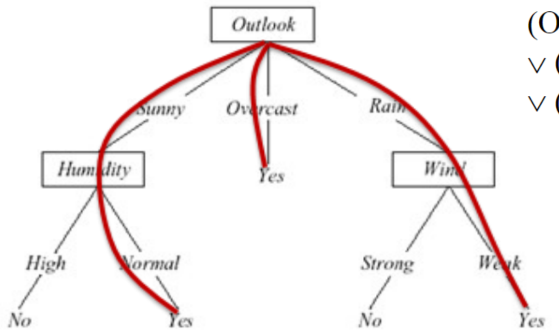
- Information gain or reduction in randomness is simply the subtraction between the entropy of the target variable (playing tennis) and the entropy of a feature variable (outlook, temperature, humidity, windy).

- **Information gain is fundamental: it tells us which variable, among the many we are considering, can make it easier for us to make a decision on whether to play tennis or not!**

- **Without doing mathematical calculations, we intuitively think: if I have a high degree of uncertainty in playing or not, the weather forecast gives me a factor that decreases my uncertainty!**

- **Once I have seen the weather forecast, I clearly must understand if the humidity and wind have the correct values to allow for a good game;**

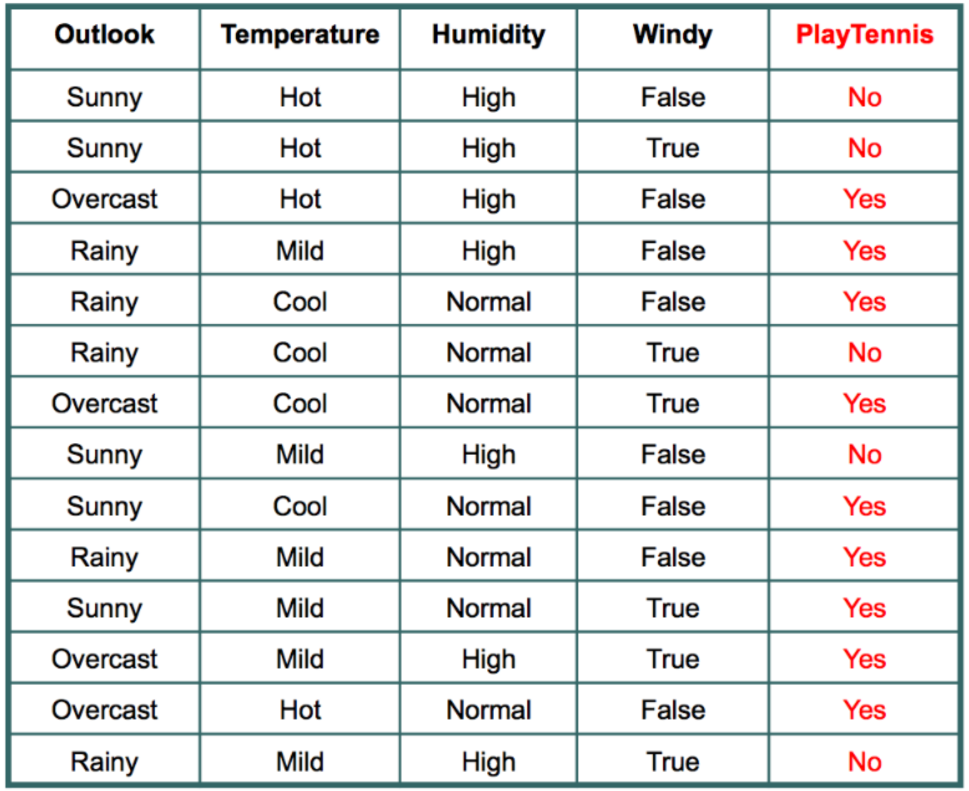
PlayTennis?



$$\begin{aligned}&(\text{Outlook} = \text{Sunny} \wedge \text{Humidity} = \text{Normal})\\ &\vee (\text{Outlook} = \text{Overcast})\\ &\vee (\text{Outlook} = \text{Rain} \wedge \text{Wind} = \text{Weak})\end{aligned}$$

Back to our trees

- Entropy in the decision tree allows us to estimate the impurity or heterogeneity of the variable we are considering;



Outlook	Temperature	Humidity	Windy	PlayTennis
Sunny	Hot	High	False	No
Sunny	Hot	High	True	No
Overcast	Hot	High	False	Yes
Rainy	Mild	High	False	Yes
Rainy	Cool	Normal	False	Yes
Rainy	Cool	Normal	True	No
Overcast	Cool	Normal	True	Yes
Sunny	Mild	High	False	No
Sunny	Cool	Normal	False	Yes
Rainy	Mild	Normal	False	Yes
Sunny	Mild	Normal	True	Yes
Overcast	Mild	High	True	Yes
Overcast	Hot	Normal	False	Yes
Rainy	Mild	High	True	No

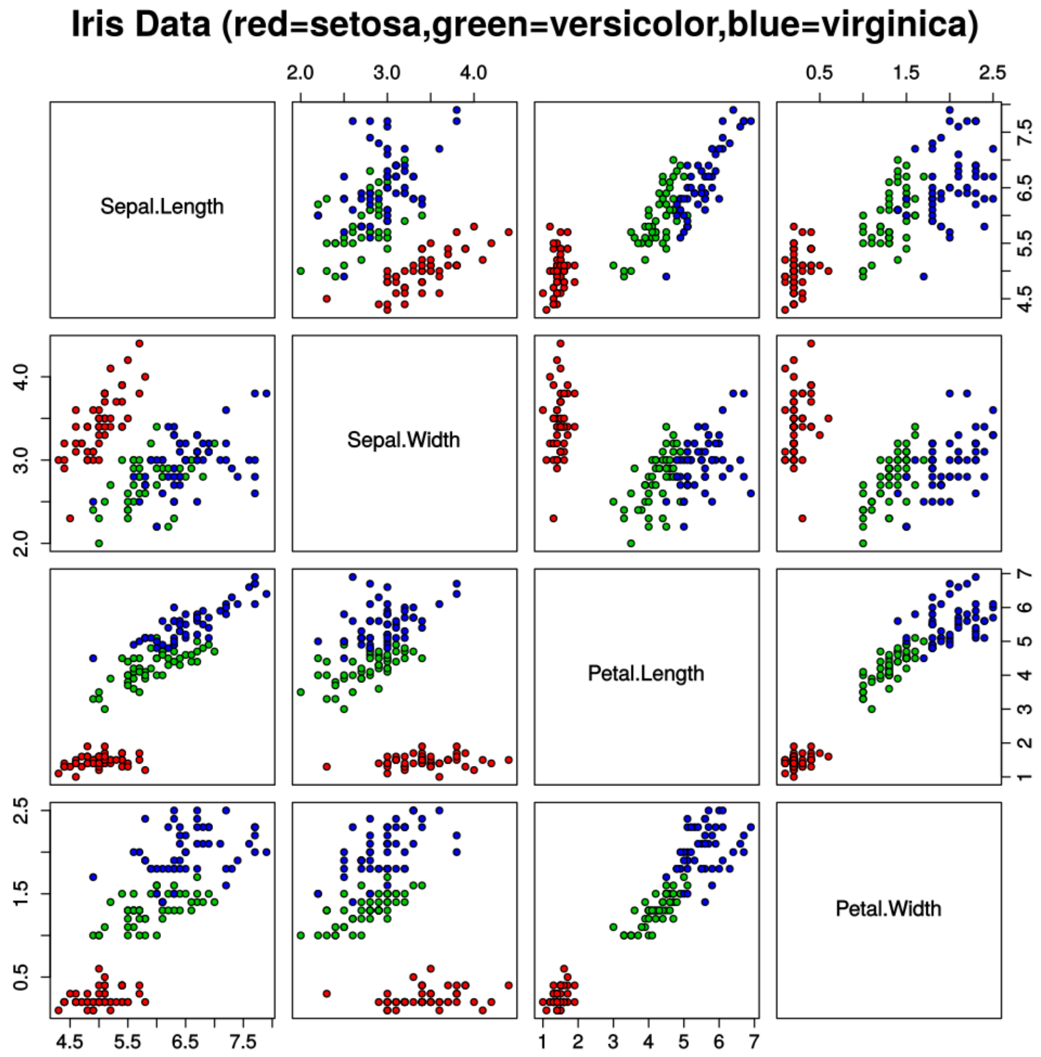
- Outlook is a casual variable with 3 events:
  - Sunny, probability 5/14
  - Overcast, probability 4/14
  - Rainy, probability 5/14

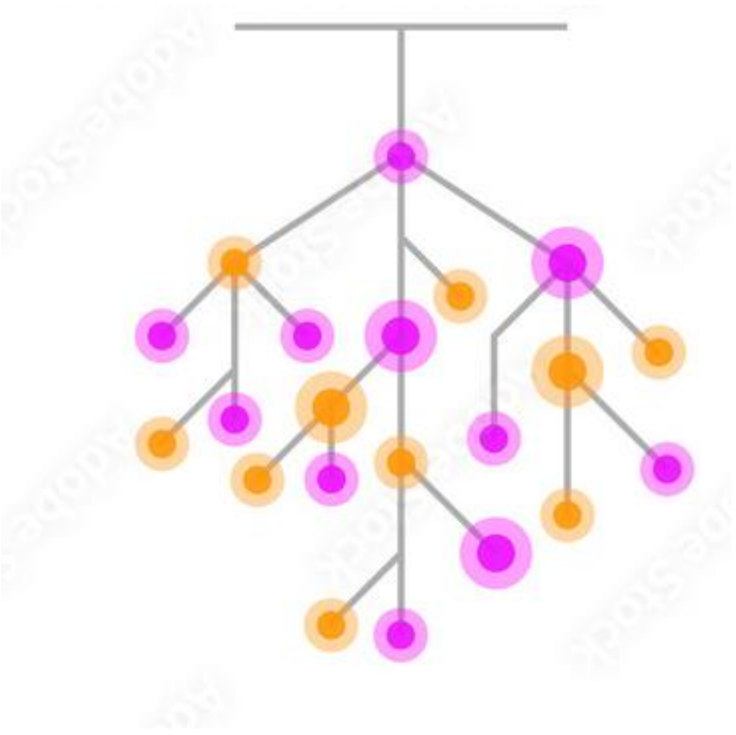
- We can do the same reasoning for the other variables ... and we can clearly calculate their entropy!

- Why calculate the entropy of these variables?

Let's code!

A green classification process: classify the iris flower dataset with a decision tree





Iris dataset

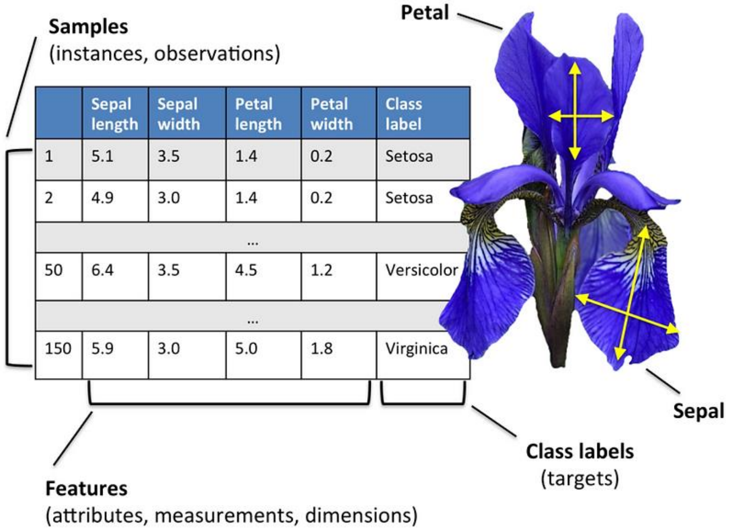
- This is perhaps the best-known database found in the pattern recognition literature;

- The dataset contains 3 classes of 50 instances each, where each class refers to a type of iris plant;

- It contains a small challenge from a mathematical point of view;

- Expected attribute: class of the iris plant;

- Characteristics:
    - length of the sepal in cm
    - sepal width in cm
    - petal length in cm
    - petal width in cm
    - classes: Iris Setosa, Iris Versicolor, Iris Virginica

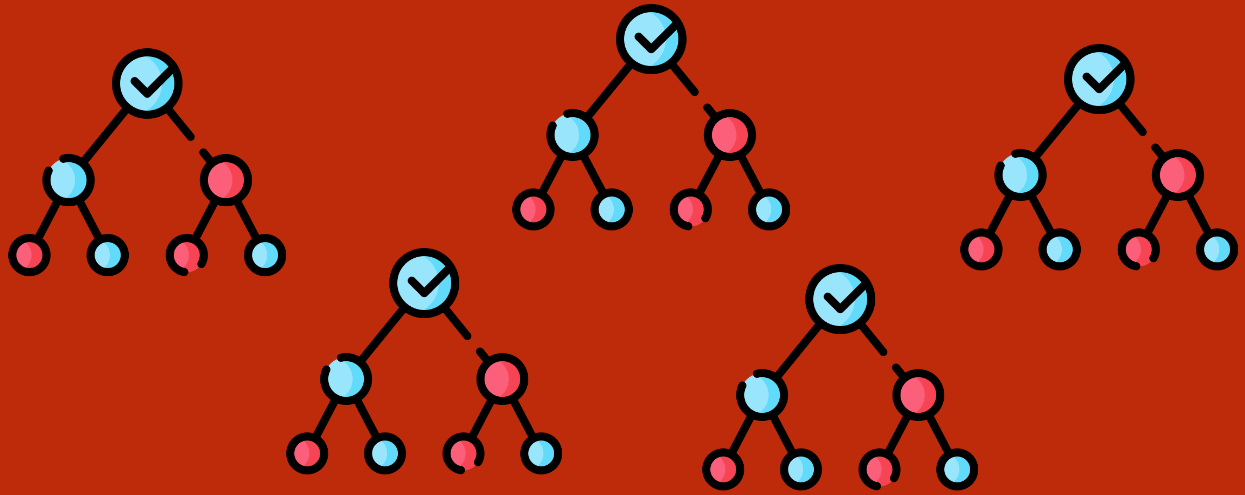


	Sepal length	Sepal width	Petal length	Petal width	Class label
1	5.1	3.5	1.4	0.2	Setosa
2	4.9	3.0	1.4	0.2	Setosa
...					
50	6.4	3.5	4.5	1.2	Versicolor
...					
150	5.9	3.0	5.0	1.8	Virginica

Samples
(instances, observations)
Petal
Sepal
Class labels
(targets)
Features
(attributes, measurements, dimensions)

Let’s colab it!

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA



Random forest

Lorenzo Stacchio

PhD student in Computer Science

Department for Life Quality studies

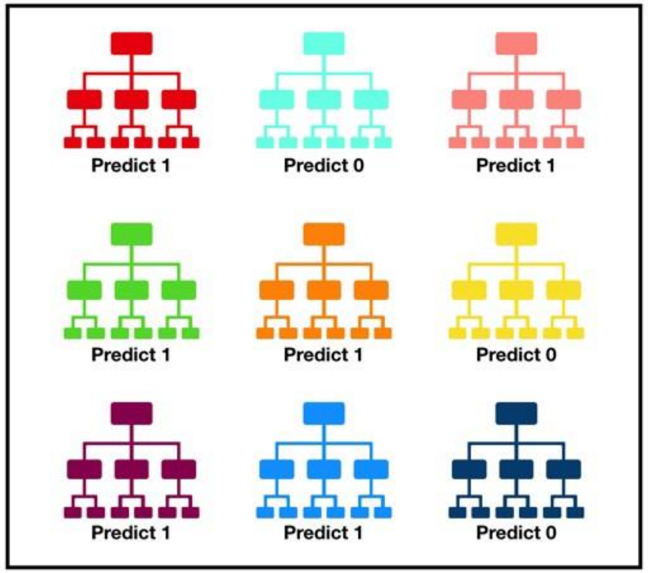
Random forest

- A random forest, as the name suggests, consists of many individual decision trees that operate as what is called an ensemble;

- Every single tree in the random forest issues a class prediction, and the class with the most votes becomes our model prediction;

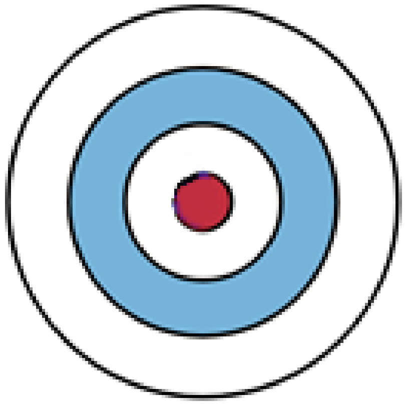
- The fundamental concept behind the random forest is simple but powerful: the wisdom of crowds.

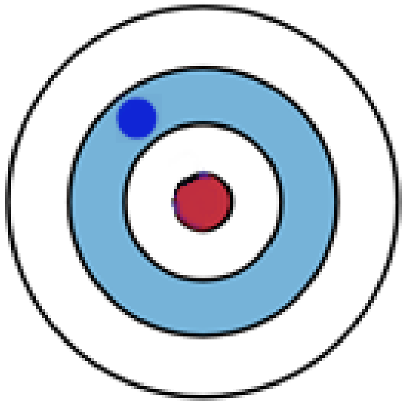
- In data science, the reason the random forest model works so well is that many relatively unrelated models operating as a set will outperform the individual constituent models (average effect).

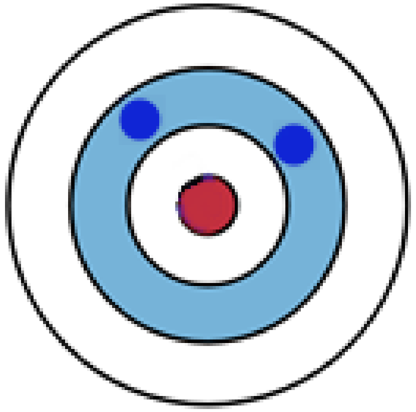


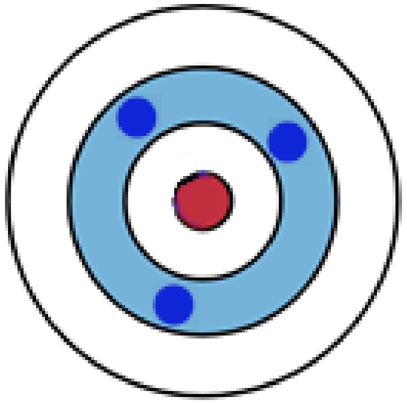
Tally: Six 1s and Three 0s
Prediction: 1

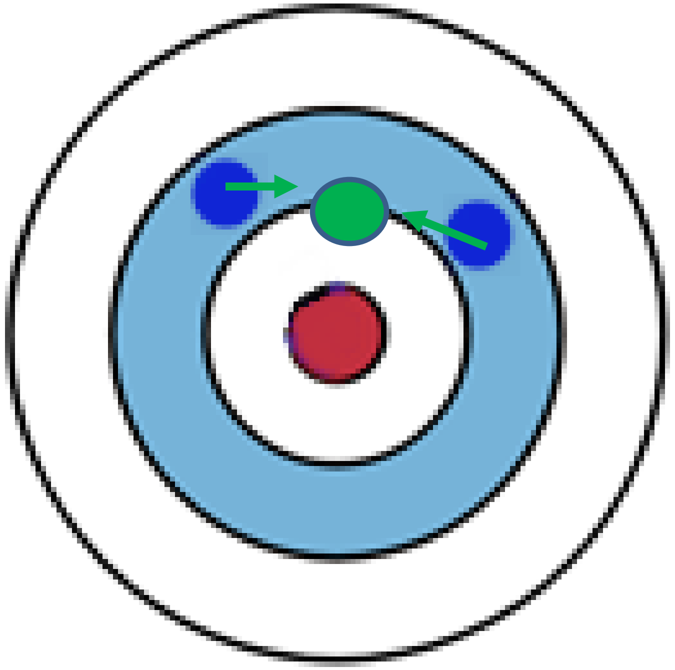
- To achieve this effect, however, trees do not have to go in the same direction of choice (unrelated trees)!
- This way, some trees will protect others from mistakes!


Function loss plane

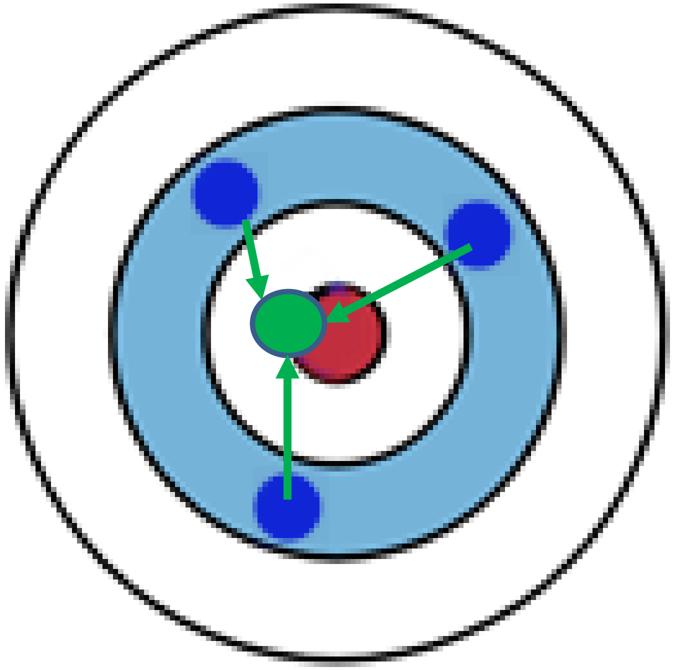

1° tree


2 uncorrelated trees


3 uncorrelated trees



2 uncorrelated trees



3 uncorrelated trees

How to avoid related trees? The Bagging technique

- Decision trees are very sensitive to the data they are trained on: small changes to the training set can result in significantly different tree structures;

- Random forest takes advantage of this by allowing each individual tree to randomly sample from the dataset **with replacement**, resulting in different trees (bagging process).

- It should be noted that the various trees do not divide the dataset into equal parts separate from each other, but simply draw different samples from the same dataset!

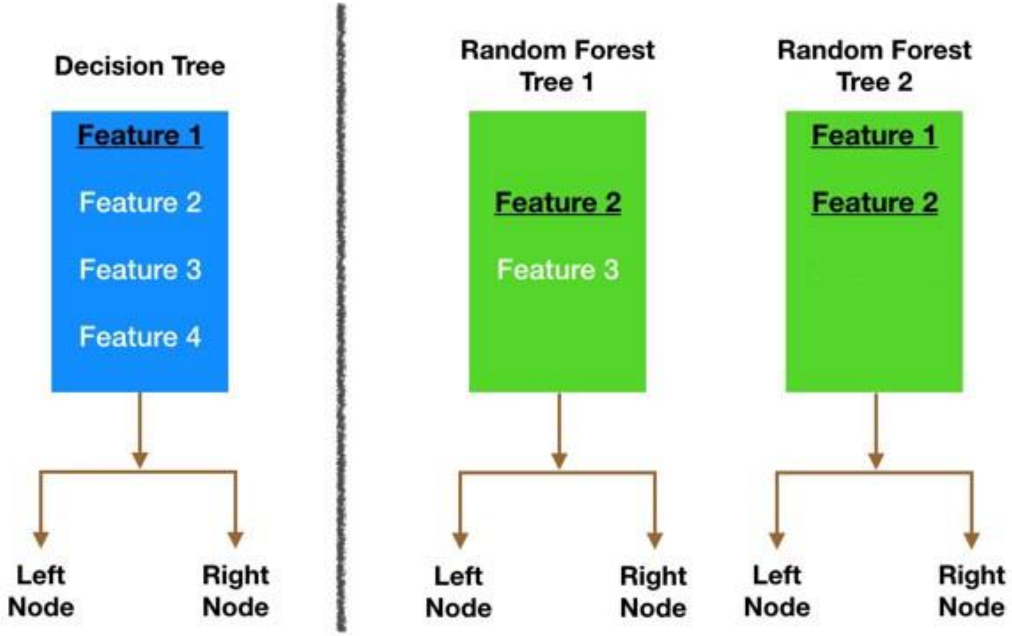
- For simplicity, think of a box that each of you can draw a number from. Normally, once a number is drawn, it is never repeated but not here, where each player would put the number back into the box, giving the next player the opportunity to draw it!

How to avoid related trees? Feature randomness

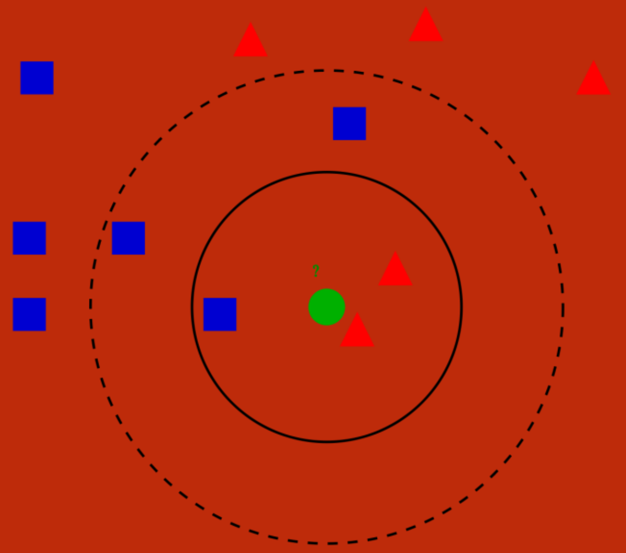
- Normally, each decision tree at each node would analyze the value of a certain feature to figure out what to do;

- However, we want trees that are not only uncorrelated in the samples but also in the features!

- For this, a random number of features to be assigned to each tree is decided (re-insertion always applies);



ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA



A world of classifiers

Lorenzo Stacchio

PhD student in Computer Science

Department for Life Quality studies

A world of classifiers

- There are tons of ML-based classifiers (supervised and unsupervised) that we haven't talked about:
    - K-means;
    - KNN;
    - SVM;
    - Naive Bayes;
    - Multi-layer perceptron;

- Then there is an entire branch dedicated to deep learning, which is now considered the state of the art for the classification of images;

    - However, many of these are based on concepts we've talked about:
    - Chance;
    - Entropy;
    - Gradient descent;
    - ...;

Python has an active community of Machine Learning developers

Big Data & Data Science

NumPy

PySpark

pandas

matplotlib

scikit learn

Deep Learning

PyTorch

TensorFlow

mxnet

ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

Lorenzo Stacchio

Dipartimento di Scienze per la Qualità della Vita

lorenzo.stacchio2@unibo.it