Artificial Intelligence and Machine Learning

Unit II

Introduction to ML

Iacopo Masi

My own latex definitions

Introduction and administrative stuff

About Me

- **Associate Professor** with Sapienza since late 2020
- **Adjunct Research Assistant Professor** with University of Southern California (USC), Los Angeles till August 2022

- Worked as Research Scientist on big DARPA projects (Dept. of Defense) of USA.

- My Background:
  - Computer Vision
  - Machine Learning

Course Schedule

- Tuesday, 2pm - 4m (2 hours)

- Thursday, 4am - 7pm (3 hours)

From February 21 till end of May 30 (one week break for Easter vacation)

Lecture Modality

- Lectures only in presence
- I will record them (video + microphone)
- Content:
  - ##### Theoretical Sessions (yes, you have to know the math behind!)
  - ##### Embedded with practicals (even how to make it computable!)
  - ##### With some cool applications (have fun!)

Lecture Modality

- **When:** second semester - Tuesday 2-4pm; Thursday 4-7pm
- **Where:** Aula 1, Building RM018
- **Forum:** We will use this Google Classroom

Course Material & Interaction

Google Classroom (Very Important):

- Material uploaded before every lecture (if time permits)
- Use Google Classroom for most and private communication with course staff
- Ask questions about logistics, homework, etc.
- Participate to Q.A. (live) sessions on Zoom

Very important: write down now!

Code to enter classroom: qgirexd

classroom.google.com/c/NTkxMzc5NDc1NzYw?cjc=qgirexd

Course Material & Interaction

Google Classroom (Very Important):

- Material uploaded before every lecture (if time permits)
- Use Google Classroom for most and private communication with course staff
- Ask questions about logistics, homework, etc.

Course Material & Interaction

- Github Website (Public for everyone) I will upload the material here too
- Our private classroom I will mainly used it to send you notifications

Course Material & Textbook

- Slides and material will be uploaded before every lecture on Google Classroom.
  - Good starting point but **but may be not enough**.
  - Textbooks are required.

Topic	Authors	Book	Difficulty
Generic ML	H. Daumé III	**"A Course in Machine Learning"**, download the book	Easy
Generic ML	Christopher M. Bishop	**"Pattern Recognition and Machine Learning"** download the book	Difficult

- The course is inspired and follows CS229 by Stanford while other material is inspired from other courses

Textbooks

There is not a single textbook but suggested are:

Topic	Authors	Book
Generic ML	H. Daumé III	**"A Course in Machine Learning"**, download the book
Generic ML	Christopher M. Bishop	**"Pattern Recognition and Machine Learning"** download the book
Generic ML	Kevin P. Murphy	"Probabilistic Machine Learning: An introduction", MIT Press, 2021
Deep Learning	Ian Goodfellow and Yoshua Bengio and Aaron Courville	"Deep Learning", MIT Press 2016
Deep Learning	Ston Zhang, Zack C. Lipton, Mu Li, Alex J. Smola	"Dive into Deep Learning"

You can find online most of these or part of them.

How to study

1. Use my slides! Most of question/answers in the exam will be coming out from my slides or a remix of them.
2. If you do not understand the slide, search for a matching chapter in one of the book I mentioned.
3. Watch again and again the lecture in the part that is not clear.

Credits

Credits: This program and material was inspired by the following courses:

- Stanford CS299
- Doretto CS691A
- Intro to ML Padova
- Stanford CS231
- Sapienza DLAI
- Sapienza ML

Exam (your payback)

- **Written exam** (open questions, exercises to solve, proof sketch)
  - Grade range $\in [0, \ldots, 17]$
  - 17 points = 15 points + 2 bonus points.
- **Bonus for slide correction:** $+[0..3]$ Give `+0.25` points for each fix (not a single typo will not do it).
- The rest of the 17 points will come from the exam of Unit I
- Final grade can arrive up to 34
- We cannot register the grade of a single Unit (AI&ML is an exam as a whole, the final score is inseparable).
- **Note: a Unit is passed if score >= 9 (18/30)**

How to submit slide correction

- Slides are all public on Github as Jupyter Notebook. As such you can:
  - Fork my repository
  - Edit the bug and fix it push it to your repository
  - Do a pull request (PR) to incorporate the slides into my GitHub repo
- Steps:
  1. Before adventuring: make sure to **have found a substantial fix (couple of typos at least)**
  2. Fill out this Google form instead of sending the email
  3. As soon as I have time, I will incorporate the fix and you the bonus points.

Exam (your payback)

Sum of the grade of Unit I with grade of Unit II

Advise: ML is widespread now.

Do not study this course just to pass the exam

Find internal motivation to do it

Establish me as a scientist in AI, help neural scientists to understand how brain works using AI

Exam: Caveat [especially for Erasmus students]

Sum of the grade of Unit I with grade of Unit II

we CANNOT record on infostud just a single Unit!

Course Objective

- Introducing you to the basic principles of **Machine Learning**
- Knowledge on the **main learning modalities** (supervised, unsupervised, parametric/non parametric)
- Knowledge on the **main ML algorithm strengths and weaknesses** (no free lunch theorem)
- Develop awarness of the **mathematical tools** behind.
- Setting **strong foundations** for more advances courses (i.e. Deep Learning)
- Develop **critical thinking**/raise next generation of **scientists**
- Show a few **cool, practical applications**

Good to know

No mandatory requirements but math tools that come in handy

- **Linear algebra:** vector/matrix manipulations (geometry in high dimensions)
- **Calculus:** partial derivatives (cost function, gradients)
- **Probability:** common distributions; bayes Rule (learn how NOT thinking deterministic)
- **Statistics:** mean/median/mode; maximum likelihood

We will review these in the first lectures

Technology is power (toolset to use)

Toolsets:

- Python (widely used in ML)
- **NumPy** (matrix manipulation and linear algebra) **I will cover the basics in the course**
- Scikit learn (basic ML) **We will try to avoid this and use our code as much as possibile**
- PyTorch (automatic differentiation and neural nets) **Basic Concepts**

You may be covering this in AI Lab class so I will not go much in details.

Technology is power (toolset to use)

Install a Python 3.8 environment with:

- python 3.8+
- numpy
- scikit learn
- matplotlib

Provisional Course Agenda at a glance

Topic	Hours
Intro, Math Recap	5
Unsupervised Learning	
Dimensionality Reduction (PCA, Eigenvectors, SVD)	5
Clustering (kmeans, GMM)	5
Supervised Learning, Non-parametric	
Nearest Neighbours	5
Decision trees	5
Self-assessment on the first part	
Supervised Learning, Parametric	
Linear Regression with Least Squares	5
Polynomial regression, under/overfitting	5
Perceptron, Logistic Regression (LR)	5
SVM	5
Deep Learning	
from LR to Neural Nets	15
Total	60

Why using Machine Learning?

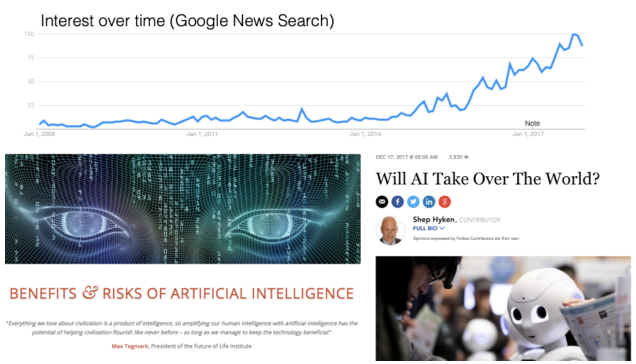
Everyone is using it now.... (Impact in applications)...

...but this is not a good answer.

We will get back on the answer later

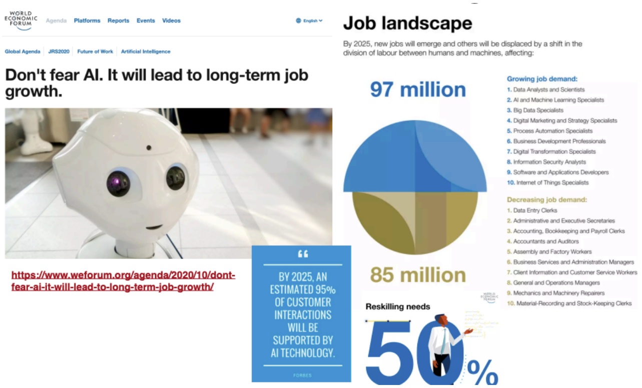
```python
from IPython.core.display import HTML

HTML("""
<style>
.output_png {
    display: table-cell;
    text-align: center;
    vertical-align: middle;
}
</style>
""")
```

Rise of AI



Graphics from [ballan2019_introML]

AI Job Landscape



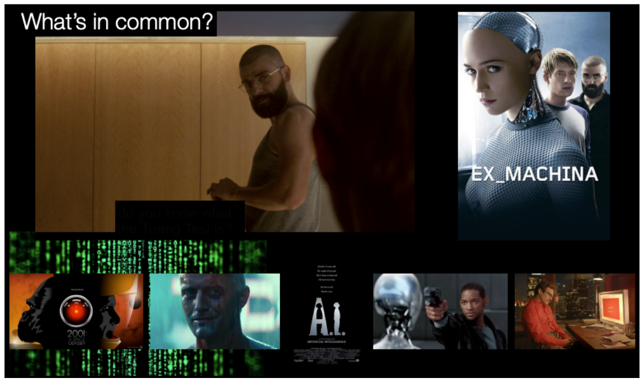
Graphics from [ballan2019_introML]

AI Job Landscape - An example

An AI&ML student from previous year given that he/she studied hard AI&ML (along with Unit I and AI Lab) was able to **secure an internship with Hewlett-Packard Enterprise (HPE)**. The student told me that:

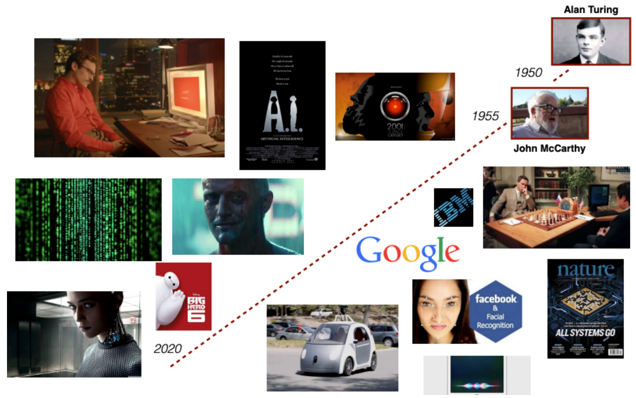
- she/he was selected among 40 candidates (1:40)
- she/he was preferred to graduated students (students obtained the master)
- she/he will work with international team

AI in Science Fiction

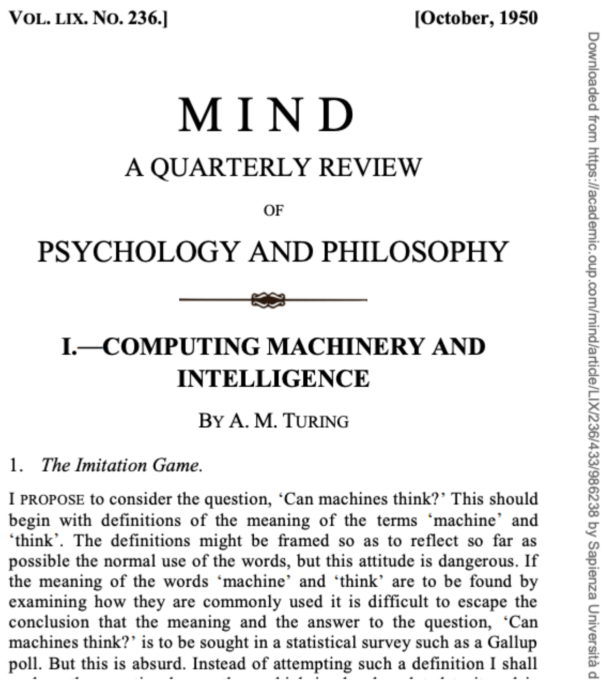


Graphics from [ballan2019_introML]

Quick History



Graphics from [ballan2019_introML]



VOL. LIX. NO. 236.] [October, 1950

MIND

A QUARTERLY REVIEW

OF

PSYCHOLOGY AND PHILOSOPHY

I.—COMPUTING MACHINERY AND INTELLIGENCE

BY A. M. TURING

1. *The Imitation Game.*

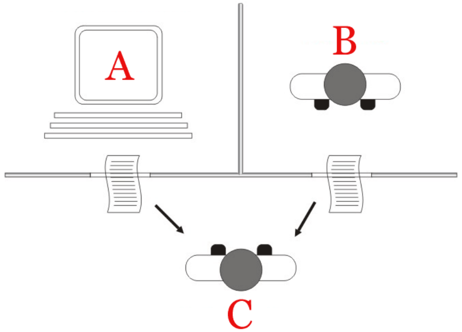
I PROPOSE to consider the question, 'Can machines think?' This should begin with definitions of the meaning of the terms 'machine' and 'think'. The definitions might be framed so as to reflect so far as possible the normal use of the words, but this attitude is dangerous. If the meaning of the words 'machine' and 'think' are to be found by examining how they are commonly used it is difficult to escape the conclusion that the meaning and the answer to the question, 'Can machines think?' is to be sought in a statistical survey such as a Gallup poll. But this is absurd. Instead of attempting such a definition I shall replace the question by another, which is closely related to it and is expressed in relatively unambiguous words.

The new form of the problem can be described in terms of a game which we call the 'imitation game'. It is played with three people, a man (A), a woman (B), and an interrogator (C) who may be of either sex. The interrogator stays in a room apart from the other two. The object of the game for the interrogator is to determine which of the other two is the man

Turing Test

The imitation game (based on language):

- The interrogator (C) is unable to see players (A, B) and can communicate with them only through written notes
- The interrogator tries to determine which player is a computer and which is a human

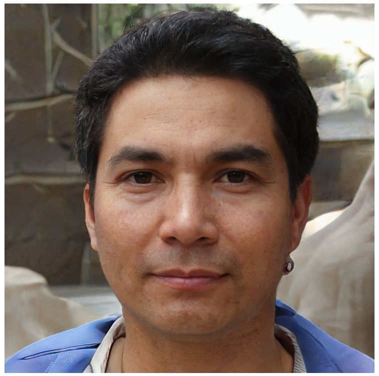


Let's do a VISUAL Turing test

Who believes this image is real?

Let's do a VISUAL Turing test

Who believes this image is real?



Let's do a VISUAL Turing test

Who believes this image is real?

What is AI (Informal)

- J. McCarthy, who coined the term in 1956, defines AI as

  > the science and engineering of making intelligent machines

- A modern definition of AI:

  > "The ability of a digital computer or computer-controlled robot to perform tasks commonly associated with intelligent beings"

What is ML (Informal)

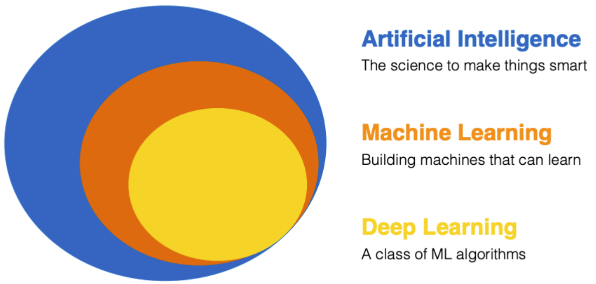
First definition in 1959 by Arthur Lee Samuel

> ML is the field of study that gives computers the ability to learn without being explicitly programmed.

- Common definition (by Tom Mitchell):

> ML is the study of computer algorithms that improve automatically through experience

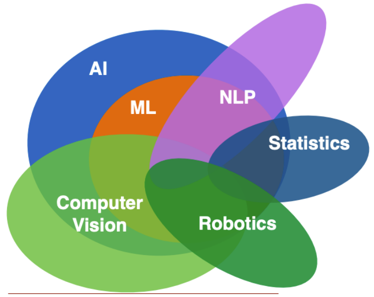
AI vs Machine Learning vs Deep Learning



Deep Learning ⊂ Machine Learning ⊂ AI

Graphics from [ballan2019_introML]

AI and beyond



- Computer Vision, Robostics, NLP in some sense they are all applications of AI to a domain.
- `vision = let machine see the world`

Graphics from [ballan2019_introML]

Yes, but why using it?

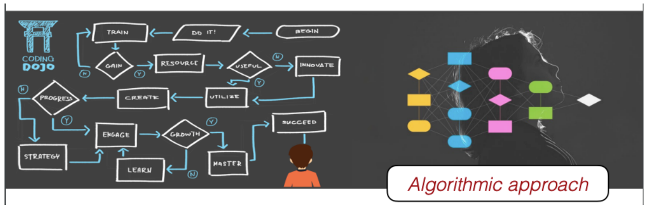
To solve problems, but which kind of problems?

There are two types of problems.

1) Problems solvable using **algorithms** developed by humans with a set of rules:

> As computer scientists (or mathematicians) we design an algorithm and write a program that encodes a set of rules that are useful to solve the problem

Algorithmic approach

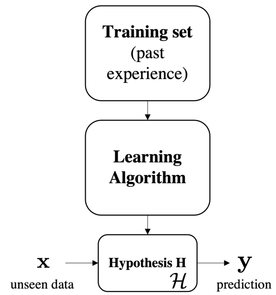


2) Problems that are **very hard** to solve with a set of rules

> As ML engineer/data scientist/research scientist we design and optimize a model that learns patterns and extract "rules" from data that that are useful to solve the problem

Big difference is: instead of writing the algorithm, we write the optimization for the hypothesis.

ML approach



Why not to use a traditional algorithmic approach?

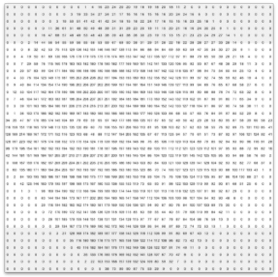
- Impossible to exactly formalize the problem (and so to give an algorithmic solution)
- Presence of noise, uncertainty, too many variations in the data
- High complexity in formulating a solution, i.e. it cannot be done manually
- Lack of compiled knowledge with respect to the problem to be solved

Example: Write a program that recognizes faces (face recognition) over a closed-set of identity

- Very hard to exactly formalize the problem
- Noise may be present and data may be ambiguous
- **Algoritmic approach:** `Store a predefined templates of faces as images with those closed set identities. Take all the pixel at position (x,y) and if then else then...`
- **ML approach:** `Learn a function that maps input images to an identity using prior data`. We will soon see that **learning** ≈ **optimizing.**

Example: Face Recognition. Humans can do it, why hard for machines?

- No one trained humans (maybe "God"/evolution/X did...)
- Can you recognize this face?
  - **...but let's do it like the computer does it**

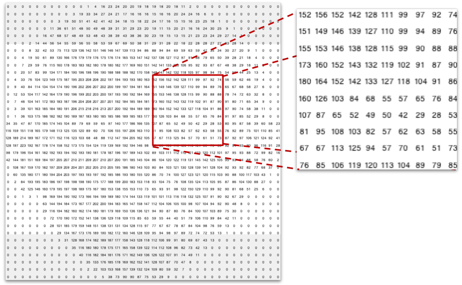


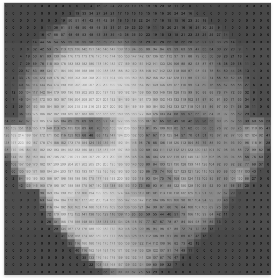
Example: Face Recognition. Humans can do it, why hard for machines?

- No one trained humans (maybe "God"/evolution/X did...)
- Can you recognize this face?
    - **...but let's do it like the computer does it**
    - **right I forgot to zoom in**

Example: Face Recognition. Humans can do it, why hard for machines?

- No one trained humans (maybe "God"/evolution/X did...)
- Can you recognize this face?
    - **...but let's do it like the computer does it**



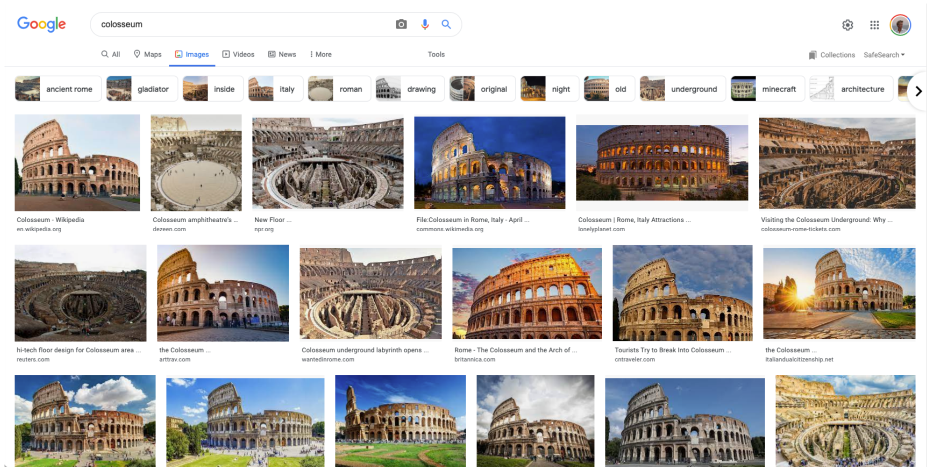


ML is widespread

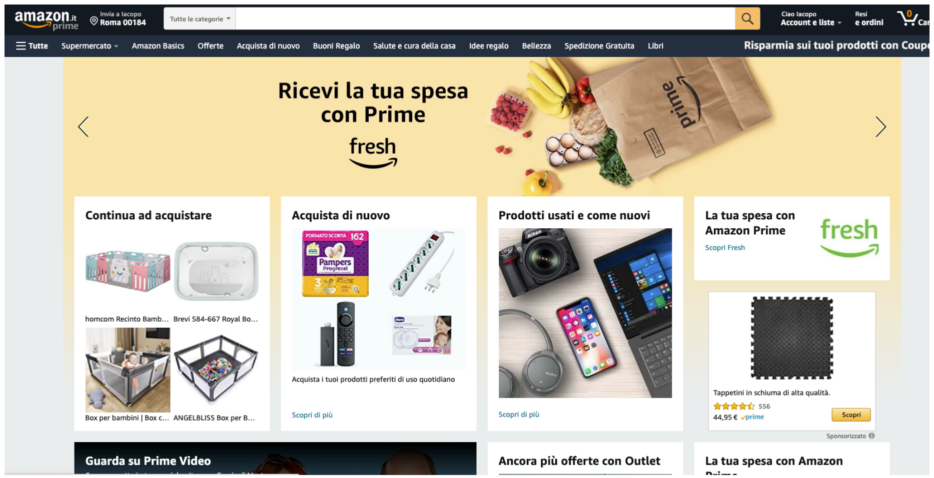
You probably use ML dozens of times a day without even knowing it:

- **[Information Retrieval]** A web search on Google works works well because a software based on ML has figure out how to rank pages
- **[Spam Filter/Classifier]** Each time you check your e-mail a spam filter has learned how to distinguish spam from not-spam e-mails
- **[Face Recognition]** When Facebook or Apple's photo application recognizes your friends in your pictures, that's also because of ML ## and useful in many tasks

Image/Text Retrieval



Reccomandation Systems



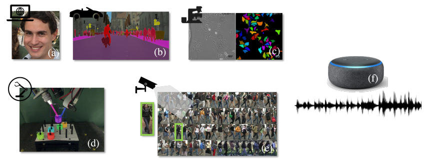
Classification/Recognition

Is this a dog?

What about this?

Applications

1. **Classification:** Determine which discrete category the example is
2. **Recognizing patterns:** Speech Recognition, Facial identity, etc
3. **Recommender Systems:** Noisy data, commercial pay-off (e.g., Amazon, Netflix).
4. **Information retrieval:** Find documents or images with similar content
5. **Computer vision:** detection, segmentation, depth estimation, optical flow,
6. **Robotics:** perception, planning, Autonomous Driving (Tesla)
7. **Learning to play games**: AlphaGO, IBM DeepBlue
8. **Recognizing anomalies:** Unusual sequences of credit card transactions, panic situation at an airport



Artificial Intelligence and Machine Learning

Unit II

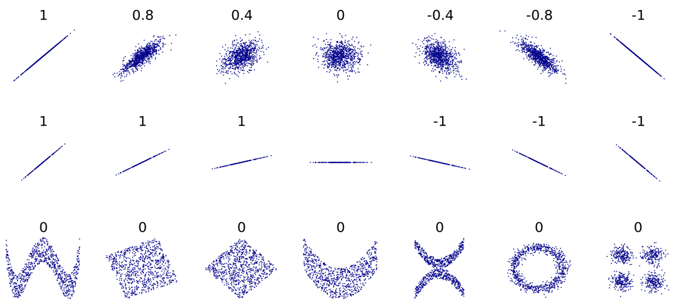
Introduction to ML

Iacopo Masi

Limits of Machine Learning

- **Causality vs Correlation**
- Noise in the data or in the labels
- Datasets could have historical bias
- In some cases, ML = blackbox that cannot explain why a prediction was made

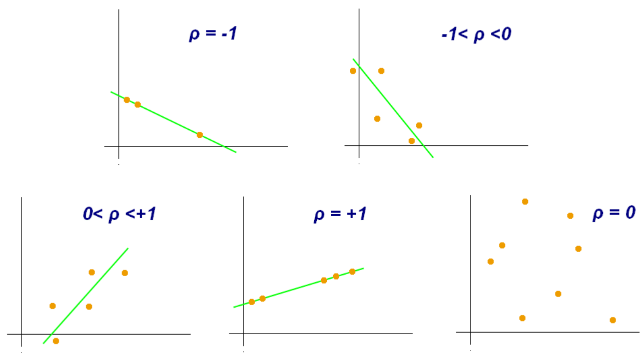
Correlation



Graphics from Wikipedia

Graphics from [this link](https://wtmaths.com/correlation.html)

Measuring Correlation



X	Y
0.1	45
0.1	65
0.2	28
0.3	76
0.5	55
0.6	48
0.9	64
1.1	41
1.5	30
1.8	52
1.8	75
1.9	35
2.1	42
2.2	65
3.0	30
3.6	71

Pearson Correlation Coefficient

$\rho_{X,Y} = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y}$

where:

- cov is the covariance of the two series
- σ_X is the standard deviation of X
- σ_Y is the standard deviation of Y

Covariance of two series

The formula for ρ can be expressed in terms of mean and expectation.

$$\text{cov}(X,Y) = \mathbb{E}[(X-\mu_X)(Y-\mu_Y)]$$

So Pearson correlation ρ can also be written as:

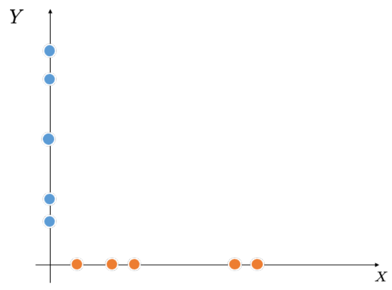
$$\rho_{X,Y} = \frac{\mathbb{E}[(X-\mu_X)(Y-\mu_Y)]}{\sigma_X \sigma_Y}$$

- **Normalized Measure of the Covariance**
- Takes values in `[-1,+1]`

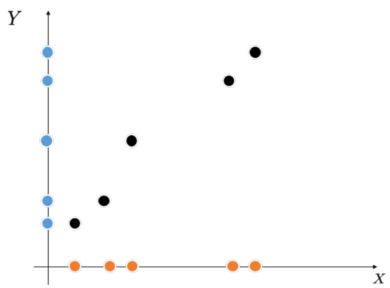
Pearson Correlation Coefficient

- The correlation coefficient ranges from −1 to 1.
- An absolute value of exactly ± 1 implies that a **linear equation describes the relationship between X and Y perfectly, with all data points lying on a line**.
- The correlation **sign is determined by the regression slope**: a value of +1 implies that all data points lie on a line for which Y increases as X increases, and vice versa for −1.
- `0` means that there is no linear dependency between variables

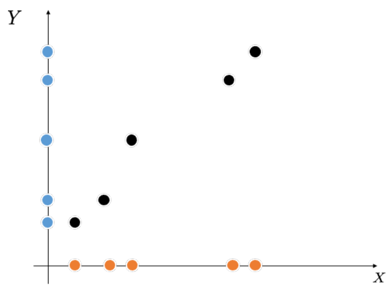
Pearson Correlation Coefficient Geometry



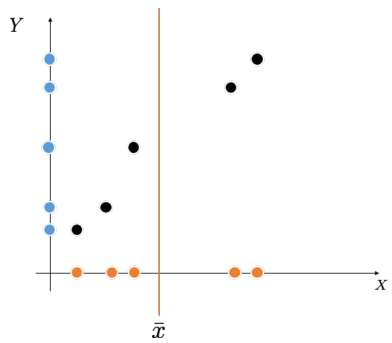
Pearson Correlation Coefficient Geometry



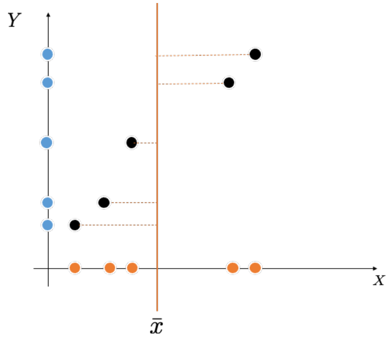
Pearson Correlation Coefficient Geometry



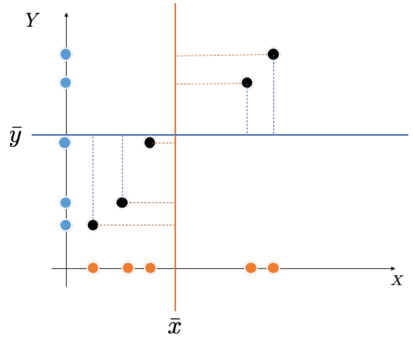
Pearson Correlation Coefficient Geometry



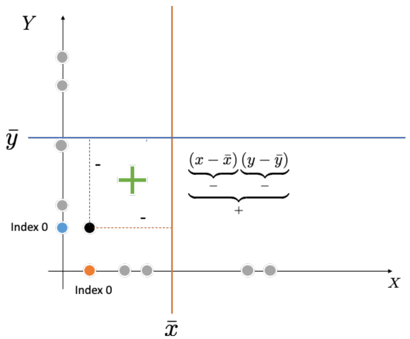
Pearson Correlation Coefficient Geometry



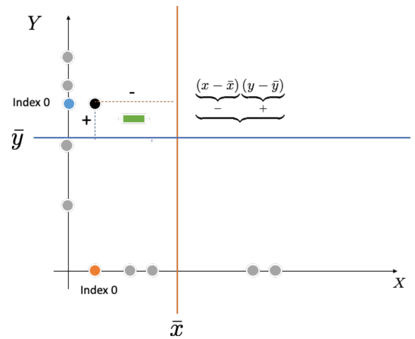
Pearson Correlation Coefficient Geometry



Pearson Correlation Coefficient Geometry



Pearson Correlation Coefficient Geometry

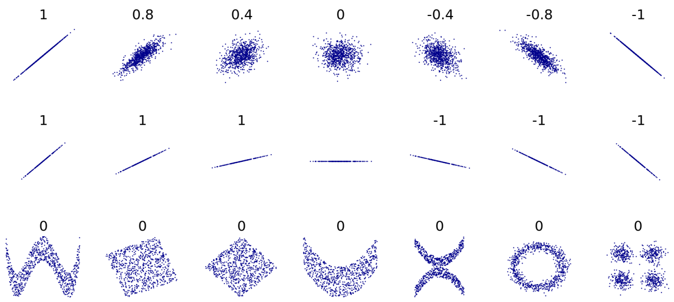


Pearson Correlation Coefficient Geometry

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}$$

- It takes maximum intensity when numerator is equal to denumerator. Otherwise Covariance is Always less than the product of the std. deviation
- The **sign of the covariance** tells you if the data is **correlated** or **anticorrelated**

Now Interpret again the plot

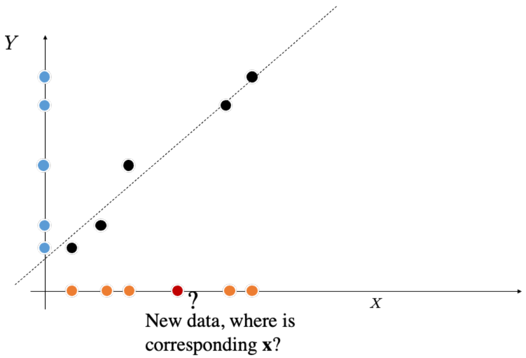


Graphics from Wikipedia

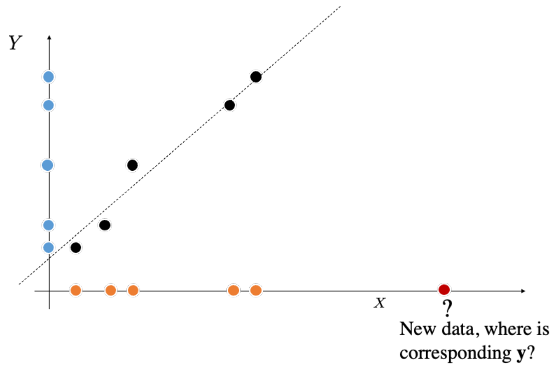
Final Note: Estimation $\longrightarrow$ Predictive Power for Future data

...but we have to be careful when predicting....

Final Note: Estimation $\longrightarrow$ Predictive Power for Future data

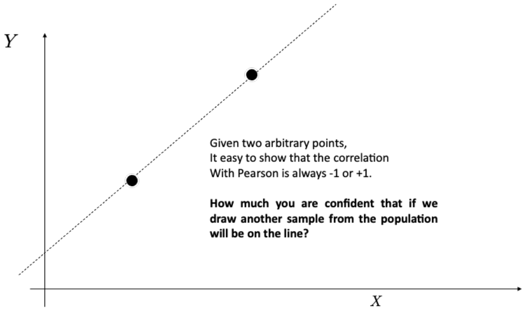


Final Note: Estimation $\longrightarrow$ Predictive Power for Future data

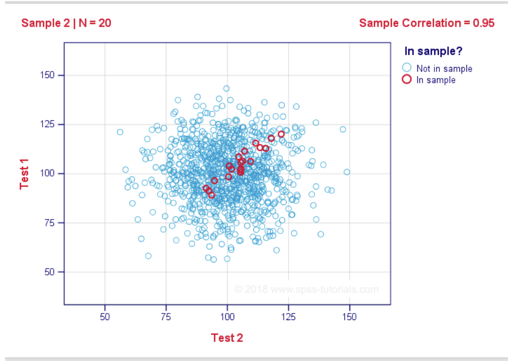


Final Note: More Samples you have, the better you predict!

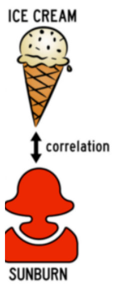
We will see what happens with ML when you have a **low number of samples for training.**



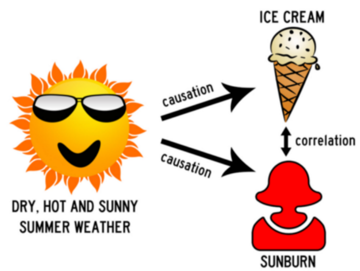
Final Note: More Samples you have, the better you predict!



Correlation DOES NOT imply Causation



Correlation DOES NOT imply Causation

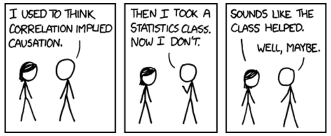


Correlation does NOT imply Causation

If given two variable A and B, we see that by increasing A, B increases as well:

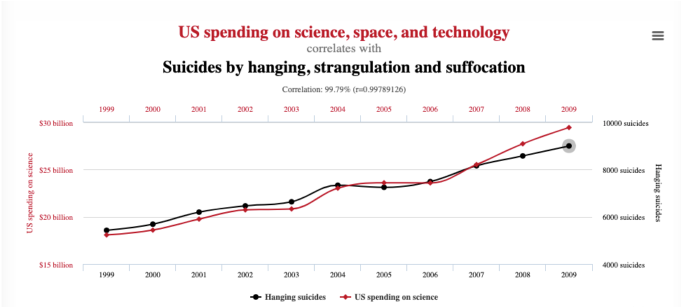
- they are positively correlated *(it could be spurious)*
- It is **NOT** sufficient condition for causality. **It may be OR may be not.**
- It could be that $B \rightarrow A$ or $A \rightarrow B$ (or even that they both co-imply)
- It could also be that another unknown variable C, $C \rightarrow A$ and $C \rightarrow B$.

Graphics from [this link](https://sundaskhalid.medium.com/correlation-vs-causation-in-data-science-66b6cfa702f0)



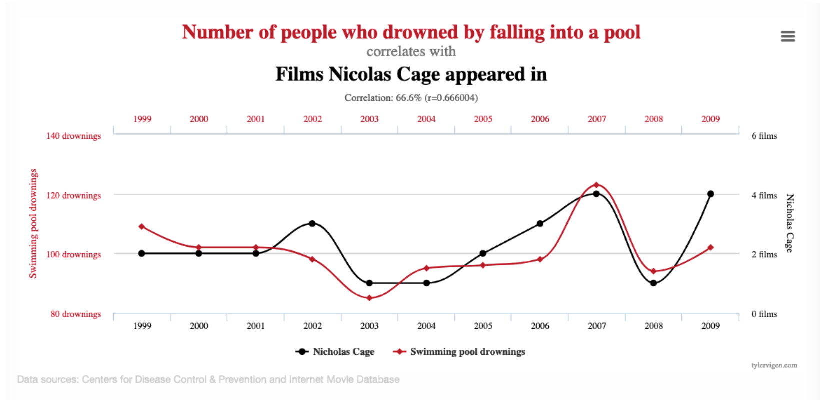
Graphics xcd comic

Spurious Correlations



Check this link out

Spurious Correlations

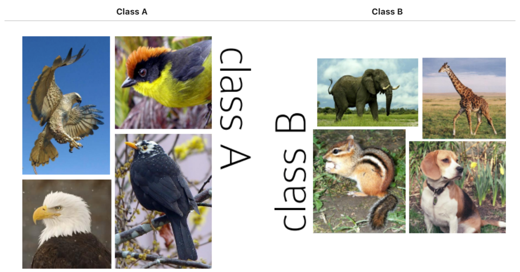


Check this link out

Inductive Bias: What We Know Before the Data Arrives

Let's play a learning "game"

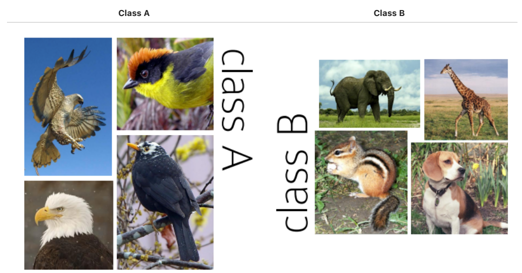
Training data



Classify these images with A or B from left to right, top to bottom

Write down your answer, then I will ask a few answers

Training data



Test data

Answers?

parrot	squirrel	cat	penguin
A	B	B	A
A	A	B	B

- ~70% ABBA prediction (Inferred **bird vs non bird**)
- ~30% AABB (Inferred **fly vs not fly**)

> This preference for one distinction (bird/non-bird) over another (fly/no-fly) is a **bias** that different **human learners** have.
>
> In the context of machine learning, it is called **inductive bias**: in the absense of data that narrow down the relevant concept, what type of solutions are we more likely to prefer?

Learning Paradigms

1. Supervised Learning (we have labels)

2. Unsupervised Learning (we do NOT have labels)

There are others: *Reinforcement Learning/Active Learning (not covered in this course)*

Introduction to Supervised Learning

Assume that there is a unknown and complex generator $\mathcal{D}$ that provides output pairs $(\mathbf{x}, y)$.

- We refer to this **unknown generator process as an unkown probability distribution** $\mathcal{D}$ over input pairs $(\mathbf{x}, y) \in \mathcal{X} \times \mathcal{Y}$.
- **Example:** Pairs of images and a label as in the case of bird/non-bird
  - $\mathbf{x}$ corresponds to the image;
  - y to the label

- Given paired $(\mathbf{x}, y)$, we learn to predict the label when given as input unseen data.
  - **Classification: the output is a discrete value (category)**
    - Binary Classification (0/1)
    - Multi-Class Classification (1...N)
  - **Regression: the output is a continous value (real-valued output)**

In pratice, in a real-world problem **no one has access to** $\mathcal{D}$ **because problems are too complex**

> Try to write a computer program to generate all possible natural images that you can find in the word. Is it easy?

Let's assume here that we have access to $\mathcal{D}$ as a python function `get_prob_under_D(x,y)` that takes as input a pair `(x,y)` and returns the probability of the pair under $\mathcal{D}$.

If so, we can define the **Bayes optimal classifier** as the classifier that:

- for any test input $\mathbf{x}'$, simply returns the y' that maximizes `get_prob_under_D(x,y)`
- Or else, try all possible labels and return the label which yields maximum prob.

$$h(x') = \arg\max_{y' \in \mathcal{Y}} \mathcal{D}(x', y') \tag{1}$$

Take away

The take-home message is that if someone gave you access to the "data distribution", forming an **optimal classifier would be trivial.**

Real world

Unfortunately, no one gave you the implementation of this distribution.

- We need to figure out ways of **learning the mapping from x to y**
- given **only access to a training set sampled from** $\mathcal{D}$, rather than $\mathcal{D}$ itself.

Training set

$$\underbrace{\{\mathbf{x}_i, y_i\}_{i=1}^{N}}_{\text{known}} \sim \underbrace{\mathcal{D}}_{\text{unknown}} \tag{2}$$

where:

- N is the number of training samples
- the vector $\mathbf{x}$ is the input data
- y is the associated (scalar) label

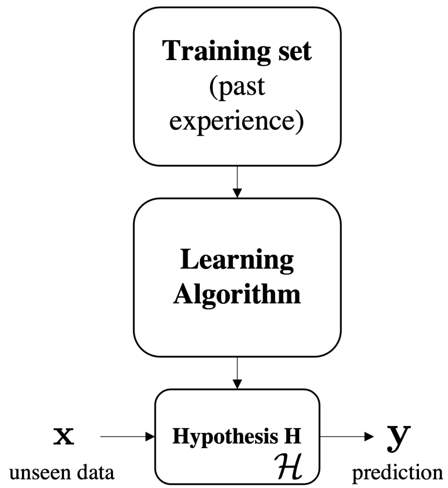
Supervised Learning

Goal: given a training set with labels, learn a function over a set of possibile functions (hypothesis over a Hypothesis set)

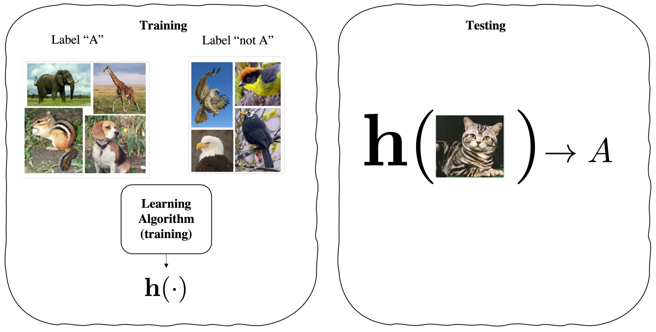
$$h \in \mathcal{H} \text{ so that } h : \mathbf{x} \mapsto y$$

Output of the learning is $h(\cdot)$ that can be used to do prediction at test-time.

Prediction: Classification (discrete-valued) vs Regression (real-valued output)



Supervised Learning for our game



Output of training is h

Output of testing is the prediction

Cardinal Rule of Machine Learning

The cardinal rule of machine learning is: never touch your test data.

Ever! If that's not clear enough:

Never ever touch your test data!

There is a specific **validation** set for that.

From cimi book:

> Do not look at your test data. Even once. Even a tiny peek. Once you do that, it is not test data any more. Yes, perhaps your algorithm hasn't seen it. **But you have. And you are likely a better learner than your learning algorithm.** Consciously or otherwise, you might make decisions based on whatever you might have seen. Once you look at the test data, your model's performance on it is no longer indicative of it's performance on future unseen data. This is simply because future data is unseen, but your "test" data no longer is.

Tools

We are going to use tools such as:

Base programming **Python**

Matrix and array manipulation **Numpy**

Basic ML methods implemented **Scikit Learn**

Plotting and Visualization Tool: **MatplotLib**

Course Material & Interaction

Google Classroom (Very Important):

- Material uploaded before every lecture (if time permits)
- Use Google Classroom for most and private communication with course staff
- Ask questions about logistics, homework, etc.
- Participate to Q.A. (live) sessions on Zoom

The End

Thank you for your attention

Unsupervised Learning

$$\underbrace{\{\mathbf{x}_i\}_{i=1}^N}_{\text{known yet no labels}} \sim \underbrace{\mathcal{D}}_{\text{unknown}} \tag{3}$$

- We do not have any labels paired with the data.
- Create an internal representation of the input, **capturing regularities/structure** in data
  - Examples: **form clusters**; **extract features**
  - How do we know if a representation is good?

Clustering (unsupervised)

- Each column is the result of a clustering algorithm
- The input data lives in a 2D space
- Colors indicates the clustering results (which points should be considered together)

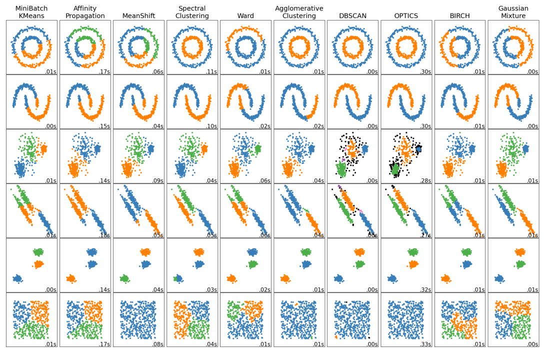


Image from [scikit-learn](https://scikit-learn.org/stable/modules/clustering.html#clustering)