Sviluppi e applicazioni di algoritmi di Intelligenza Artificiale

Loredana Bellantuono

Dipartimento di Scienze Mediche di Base, Neuroscienze e Organi di Senso
Università degli Studi di Bari Aldo Moro

Istituto Nazionale di Fisica Nucleare, Sezione di Bari

2° Congresso della Sezione **INFN** e del **Dipartimento Interateneo di Fisica** di Bari

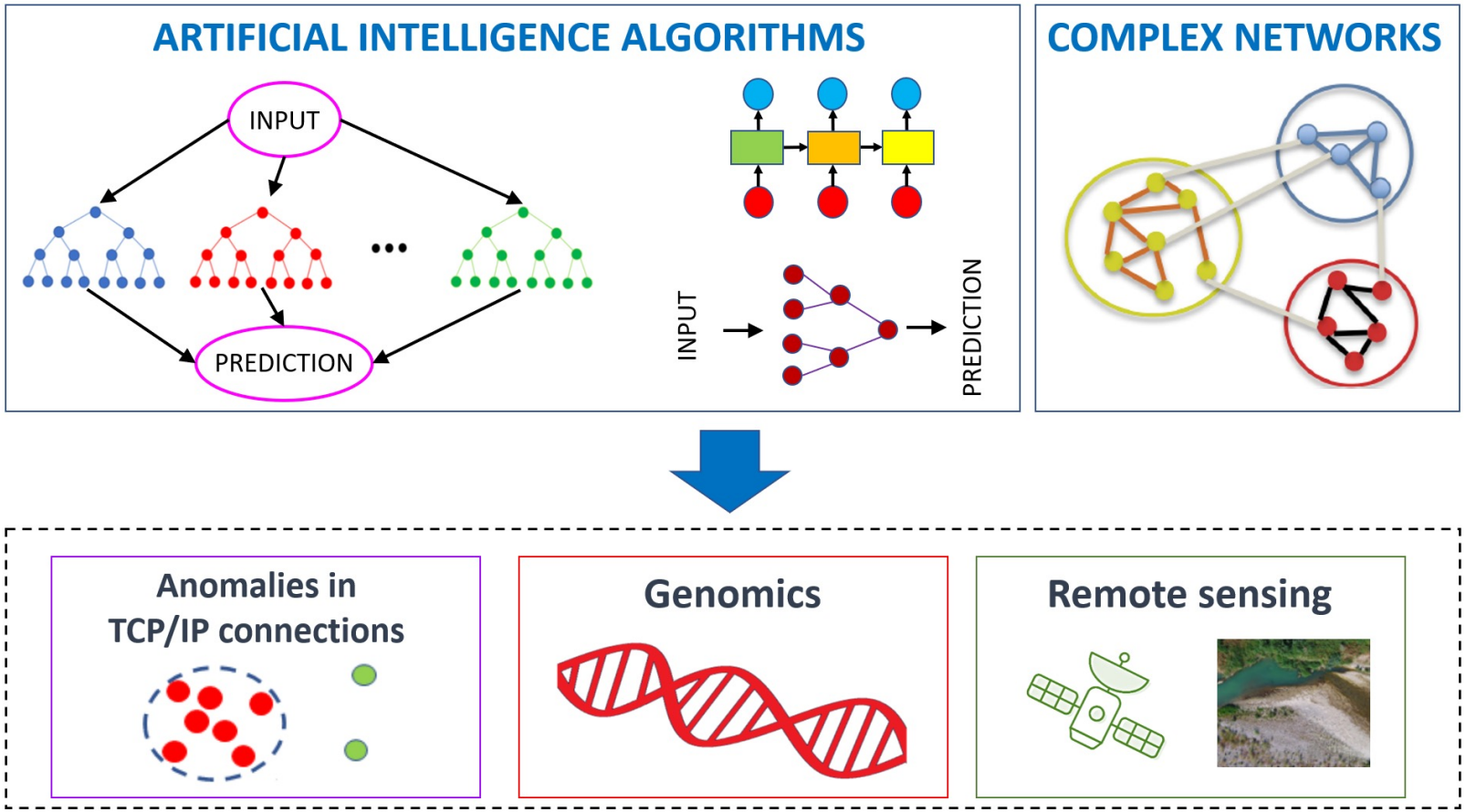
Bari, 4 Febbraio 2022

Outline

Our focus → **Feature engineering** in the activities of **Gruppo V** in Bari

METHODS

APPLICATIONS



New GPU Cluster @ ReCaS-Bari

All the applications shown in the following are based on the **ReCaS-Bari high-performance computing** capability

ReCaS upgrade → new GPU Cluster [more details in Gioacchino Vino's talk]

- Currently, the cluster is composed of:
  - 38 GPUs (18 Nvidia A100 and 20 Nvidia V100)
  - 1755 CPU cores
  - 13.7 TB RAM
  - 55 TB Disk Space
- Thanks to the high number of cores, the GPUs allow the running of high-performance **parallel algorithms reducing the overall execution time**
- The configured cluster is able to run batch jobs or open **interactive environments** where users are able to write code and test it in real time

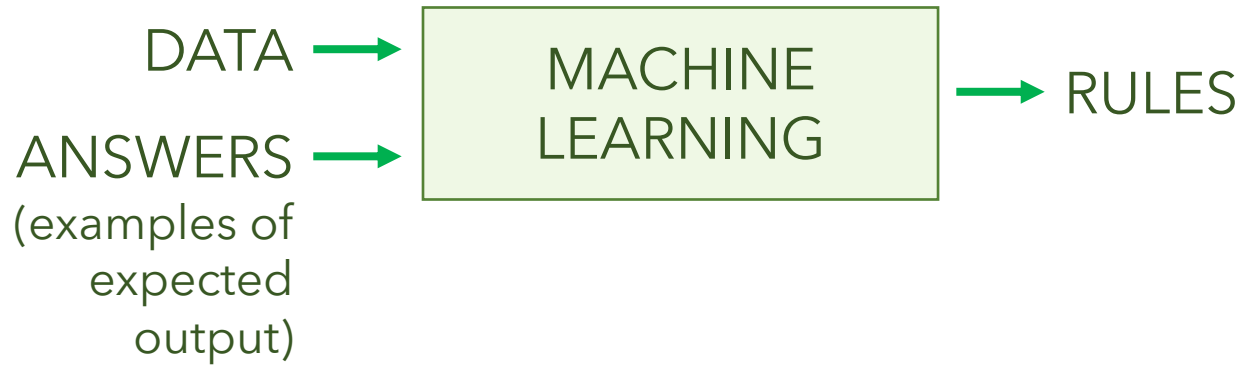
Artificial Intelligence (AI)

the effort to automate intellectual tasks normally performed by humans

François Chollet

From **classical programming**...

...to **Machine Learning (ML)**



searching for useful representations of input data, within a hypothesis space, using guidance from a feedback signal

Feature engineering

the process of transforming raw data into features that better represent the underlying problem to the predicting models, resulting in improved model accuracy on unseen data

Jason Brownlee

Manual **FEATURE CONSTRUCTION** from raw data

FEATURE EXTRACTION using dimensionality reduction algorithms (e.g. PCA)

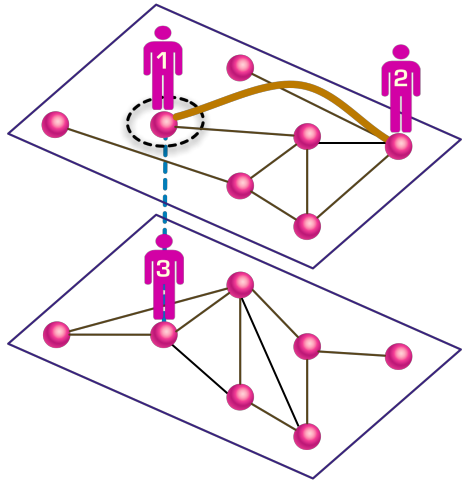
FEATURE IMPORTANCE and **SELECTION:** from many features to a few ones that capture the main structure of the data

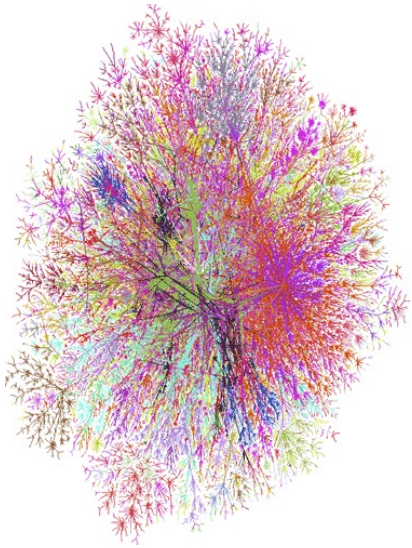
FEATURE LEARNING: automatic construction of abstract representations from raw data

Feature engineering in complex systems

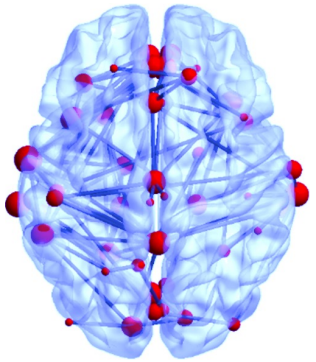
The whole is more than the sum of its parts

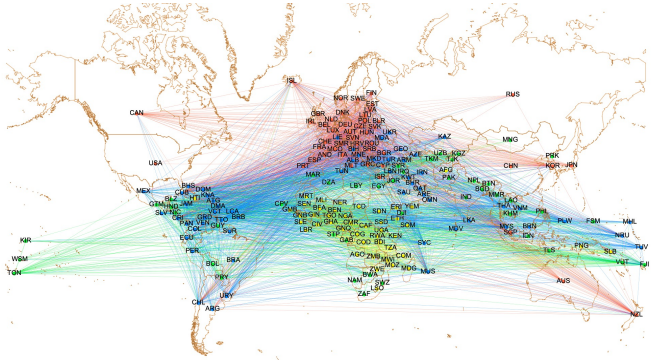
Aristotle





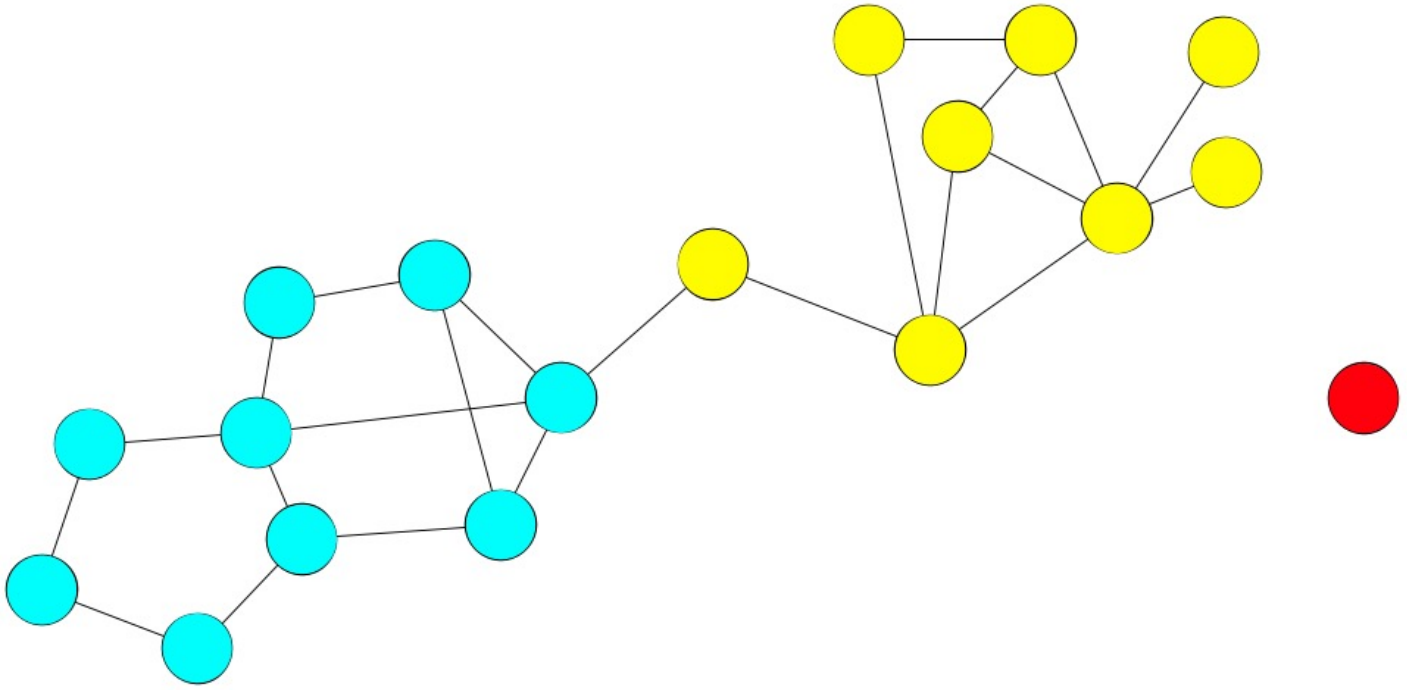
- Complex systems are such that the **collective behavior** of constituents cannot be inferred from properties of the parts.
- Example of **real-world complex systems** include: funding relations in global economies, climate, human brain, social interactions, and modern infrastructures.
- The **pattern of interactions** among components in a complex system can be represented as a **network**, which provides the framework for **feature engineering**





Network-based feature engineering

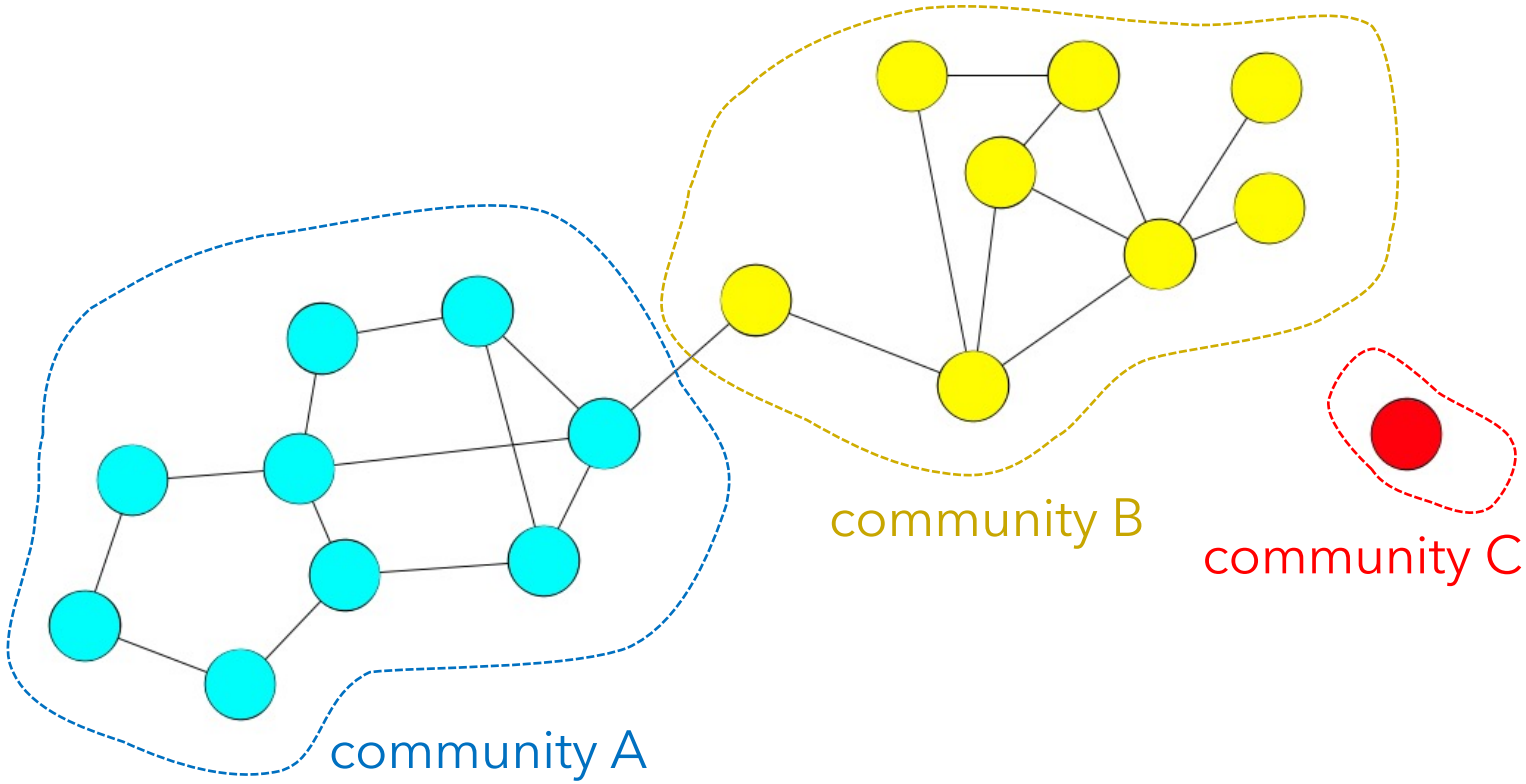
Complex network: a collection of objects (*nodes*), whose mutual relations are modeled as *links*. The strength of these relations is quantified by assigning a *weight* to each link.



The **theory of graphs** provides a set of **quantitative methods** to unveil and characterize network properties at all scales: from **single-node centralities** to **global connectivity structures**.

Network-based feature engineering

Complex network: a collection of objects (*nodes*), whose mutual relations are modeled as *links*. The strength of these relations is quantified by assigning a *weight* to each link.

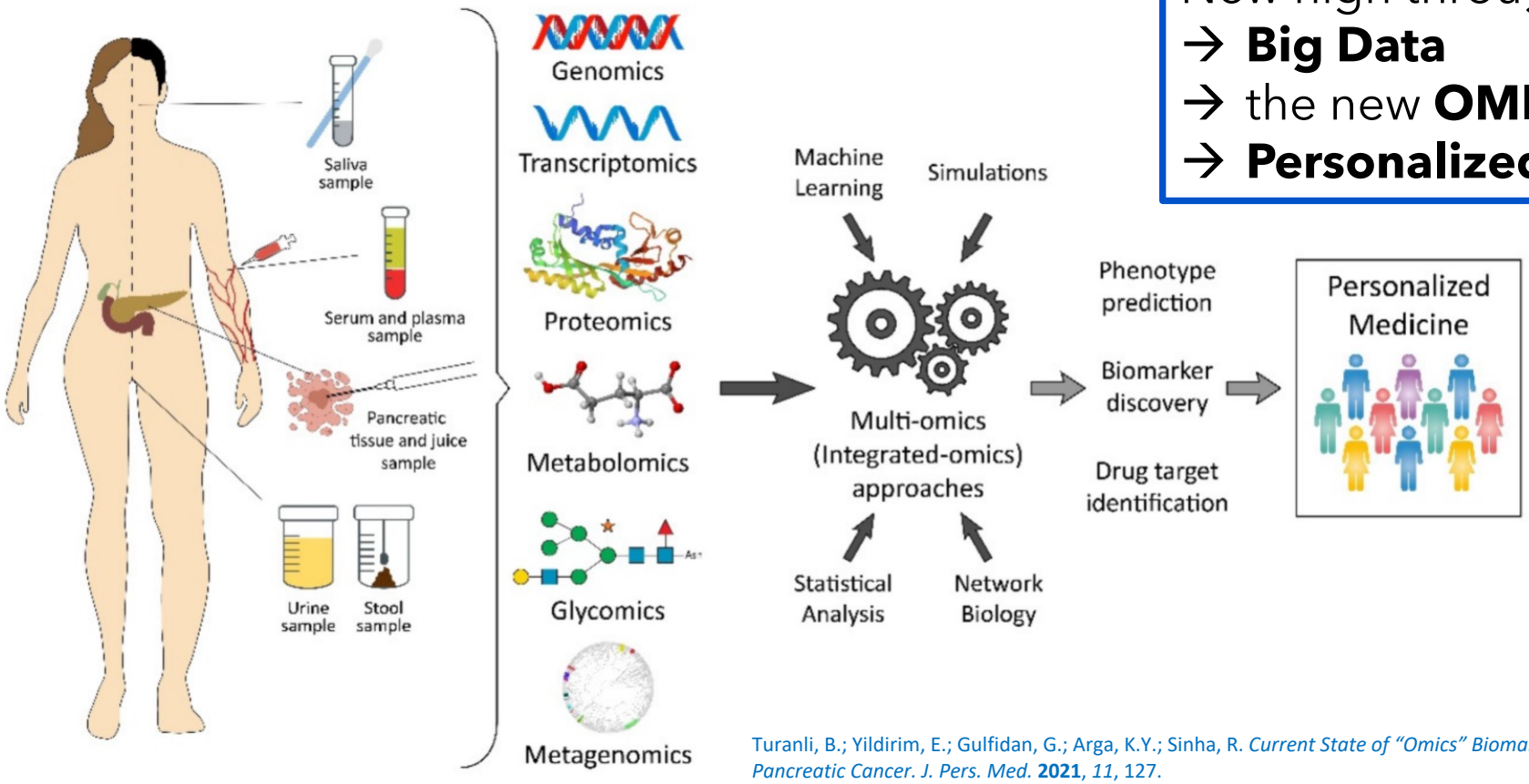


Community detection algorithms allow to identify sets of nodes with a tendency to interact among themselves more than with nodes in the rest of the network.

Genomics

Managing, Analysing, and Integrating **Big Data** in Medical **Bioinformatics**

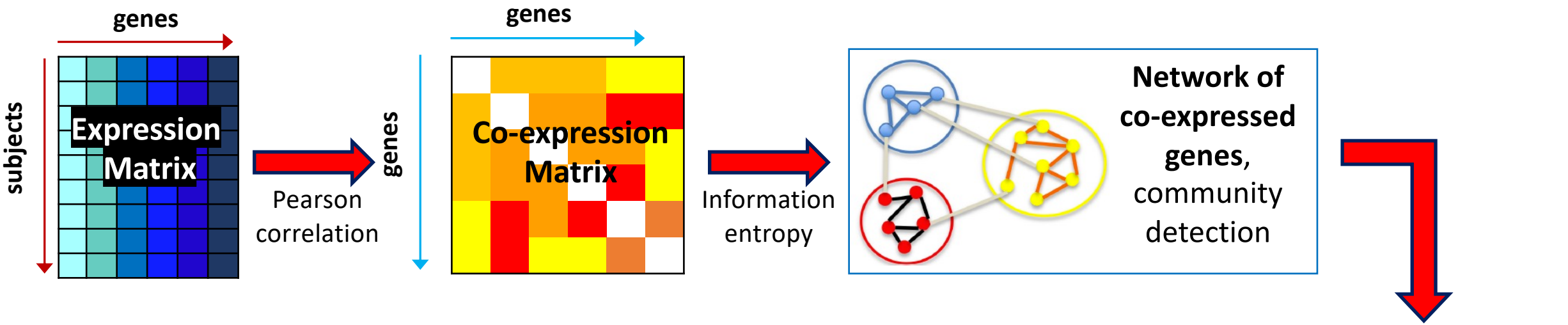
New high throughput technologies in Biology
→ **Big Data**
→ the new **OMIC SCIENCES**
→ **Personalized Medicine**



1 individual:
6 Gbases in the genome
10^{13} cells
10^{14} cells in the microbiome

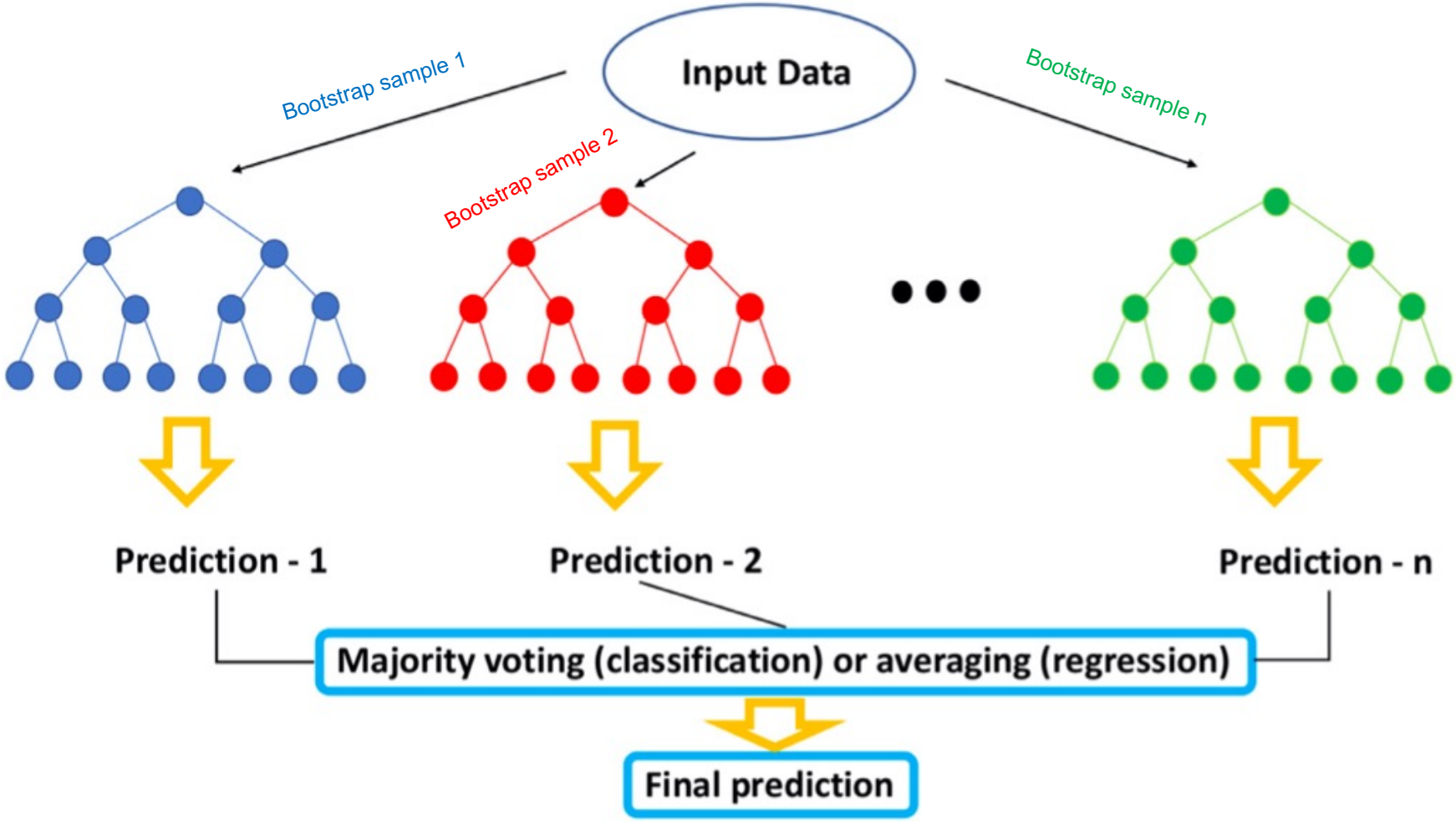
Turanli, B.; Yildirim, E.; Gulfidan, G.; Arga, K.Y.; Sinha, R. *Current State of "Omics" Biomarkers in Pancreatic Cancer. J. Pers. Med.* **2021**, *11*, 127.

Relevant gene communties in PD/NC classification



Random Forest framework

Training: **randomly subsampled input data** is processed in **decision trees**, grown on **randomly selected features**



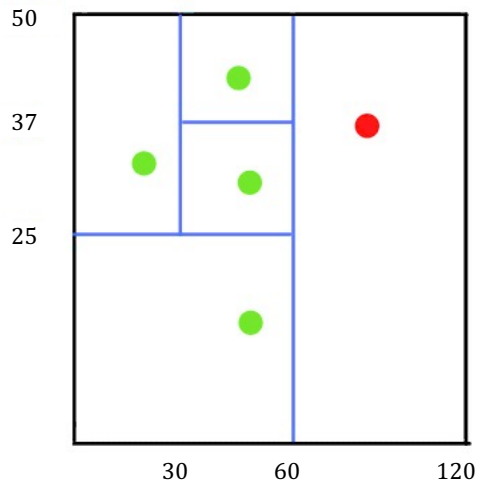
Isolation Forest framework

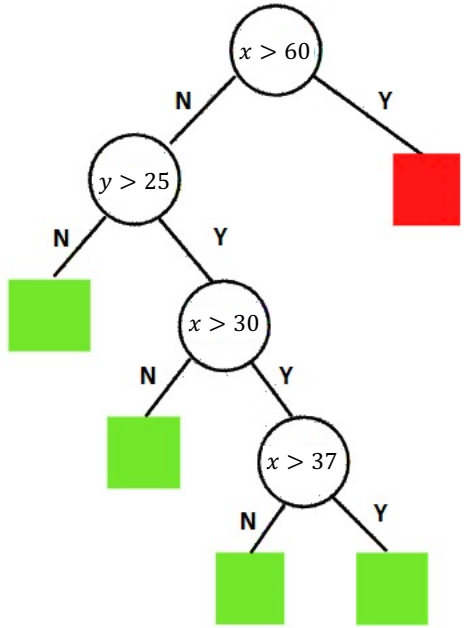
Isolation Forest is an **unsupervised anomaly detection technique**

Given some input data, an observation that significantly deviates from the others is called an **anomaly/outlier**.

Training: **randomly subsampled input data** is processed in decision trees.

At each node, branching is done on **a randomly selected feature**, using **a random threshold**.





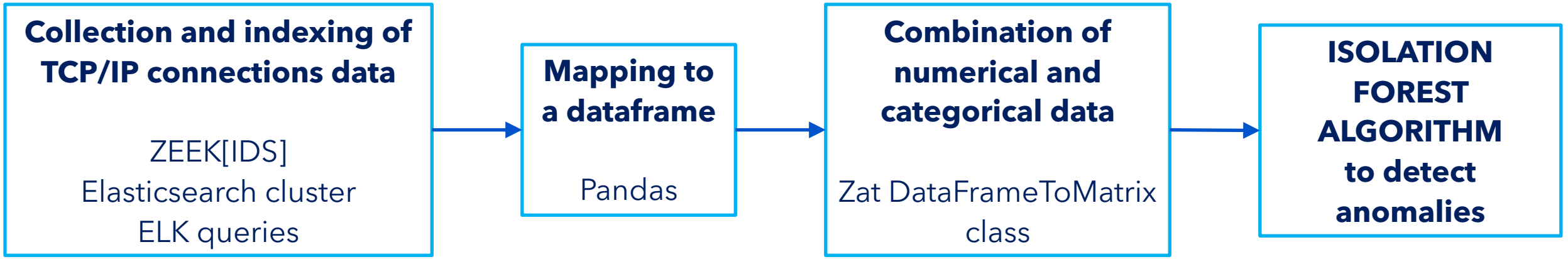
The samples which end up in shorter branches indicate **anomalies** as it was easier for the tree to separate them from other observations.

Anomaly detection with Isolation Forest @ReCaS-Bari

Key idea: using data to teach a computer the proper (*normal*) behavior of a system, to be able to understand when it **deviates from *normality***

Anomaly detection finds its application in various domains , to efficiently **spot abnormal events** such as fraudulent bank transactions, network intrusion, sudden rise/drop in sales, change in customer behavior.

Specific **use-case**: highlight **traffic anomalies in TCP/IP connections to** the **central switch/router** of the **ReCaS-Bari datacenter**, providing a **first-level monitoring system** of the server data.

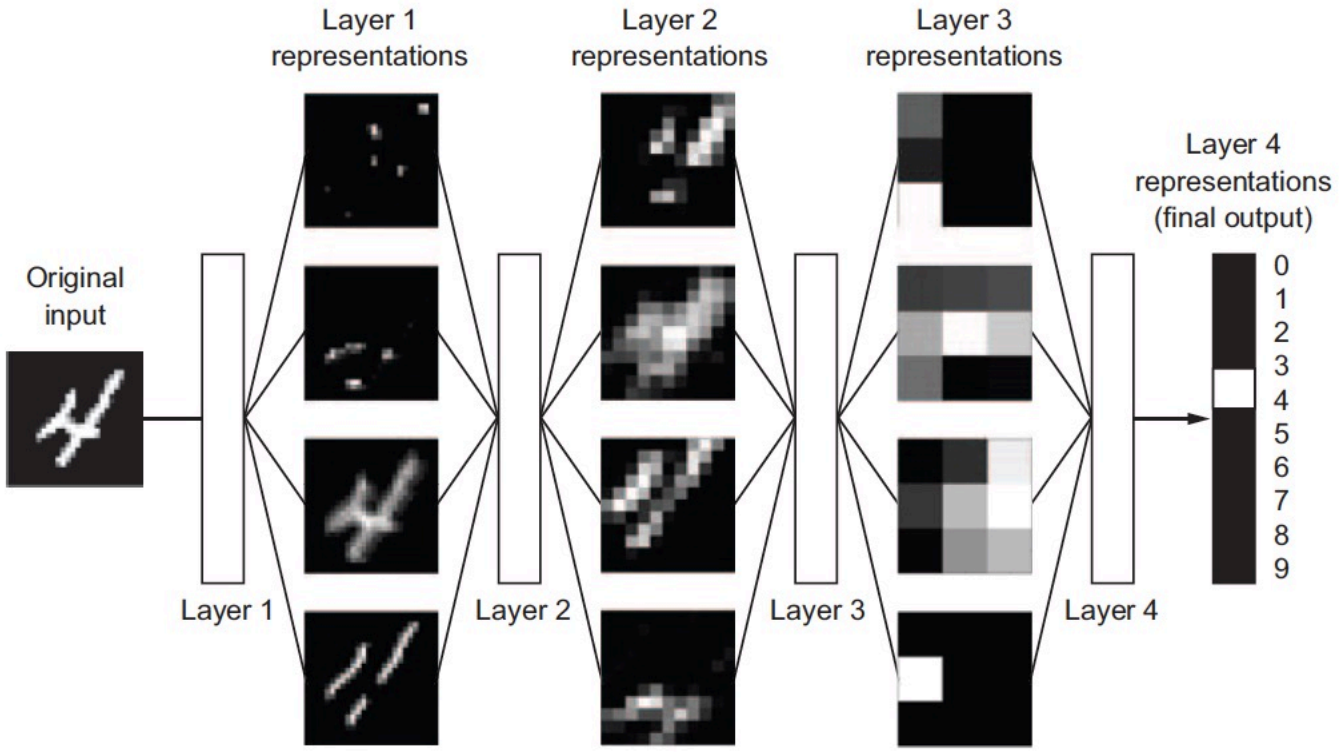


Other approach for the same use-case: **AUTOENCODER**

Deep Learning

Full **automation** of **feature engineering**

DEEP NEURAL NETWORK (DNN)



F. Chollet, Deep Learning with Python (2018)

State of the art for all **perceptual tasks**

The learning workfow is

- **INCREMENTAL**: the input image is transformed into **successive representations** that are **increasingly different from the original one** and **informative about the output**
- **COLLECTIVE:** when an intermediate layer is updated, all the others **automatically adapt** to the change.

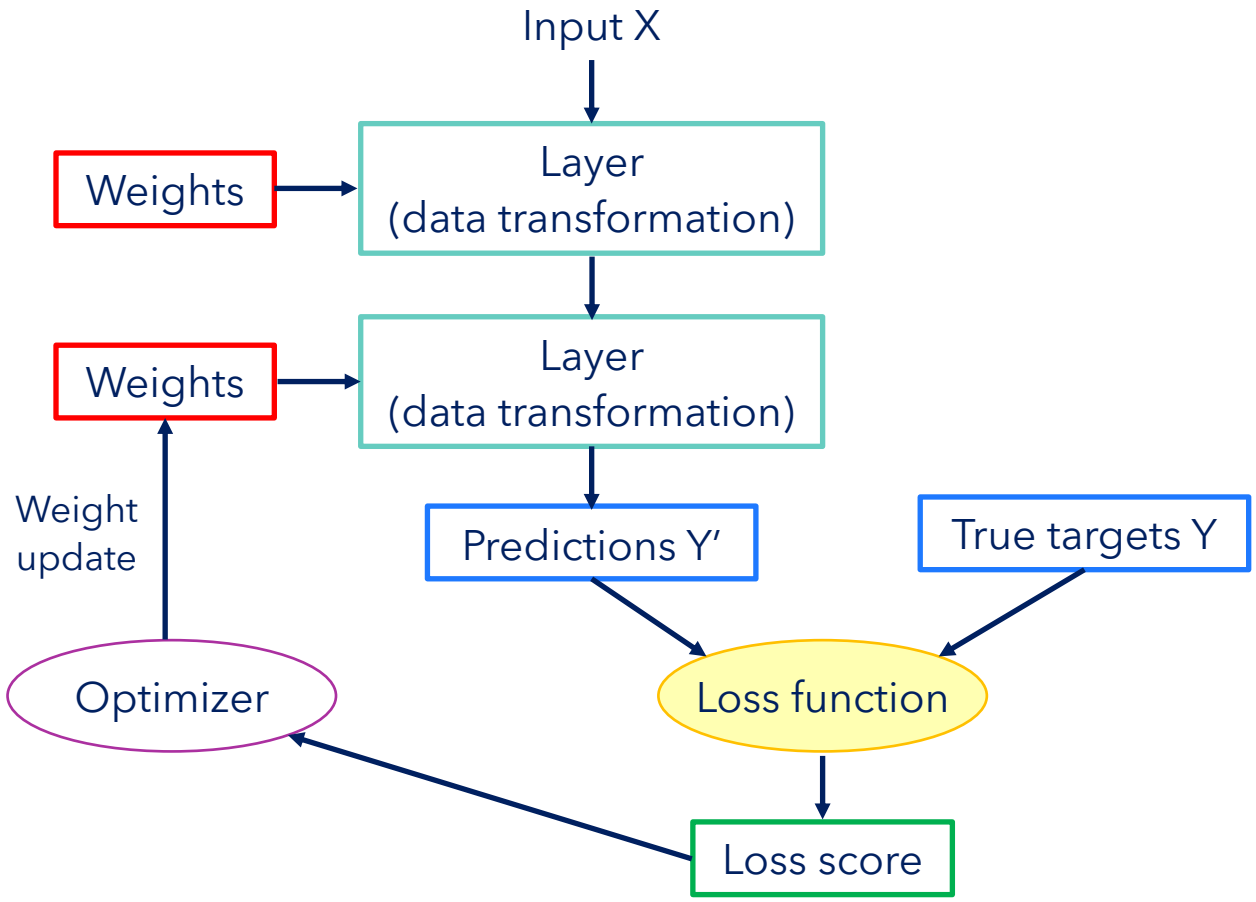
Deep Learning: how it works

The transformation implemented by a layer is parametrized by its **weights**.

Learning means finding **optimal weights**, such that the DNN correctly maps **inputs** to **targets**.

A **loss function** measures the discrepancy between prediction and target.

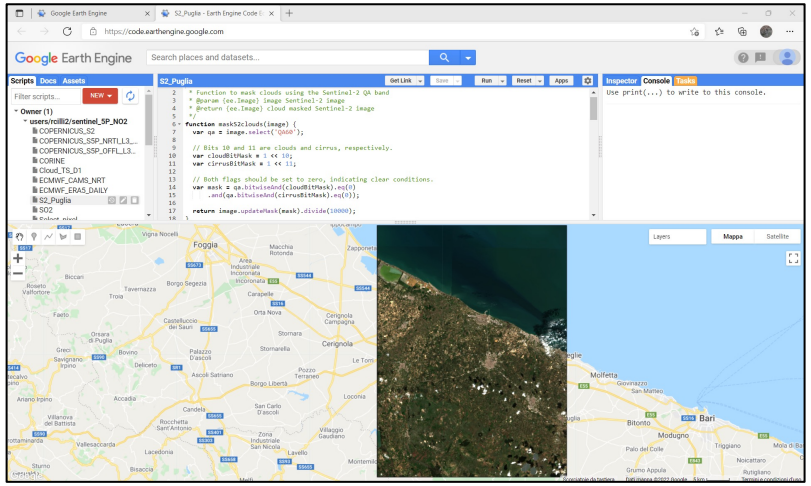
Backpropagation: the loss score is used as a **feedback signal** to adjust the weights.



F. Chollet, Deep Learning with Python (2018)

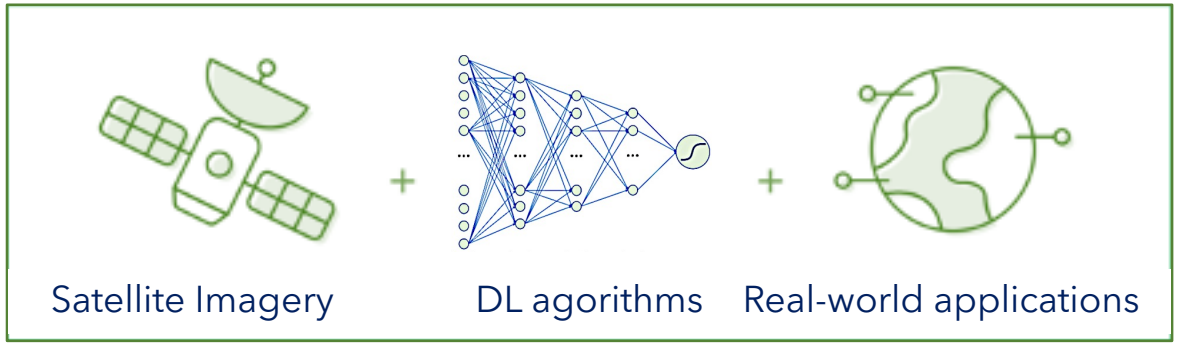
Remote sensing for Earth observation

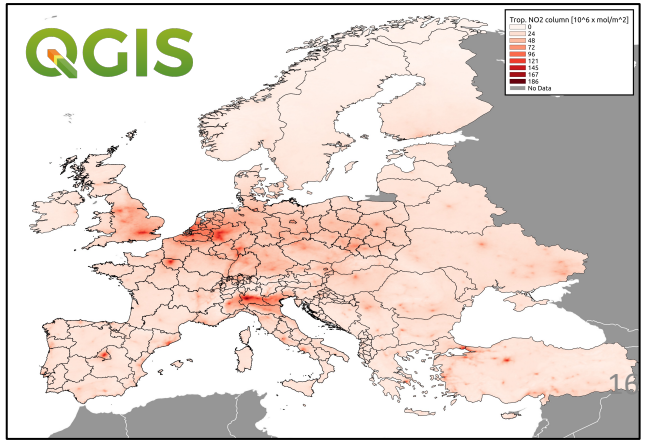
Workflows based on **satellite imagery**, **photogrammetric data**, and **geospatial information systems** for generating new datasets and enabling photointerpretations.



SOME GOALS

- Developing models for **object** and **change detection**
- Monitoring **hydro-geomorphological high-risk** areas
- Supporting **search** and **rescue** activities

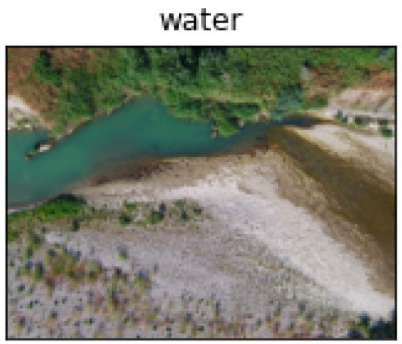


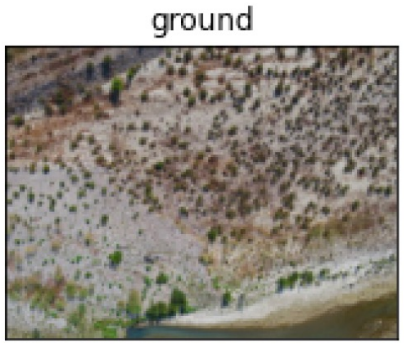


Remote sensing applications with Deep Learning

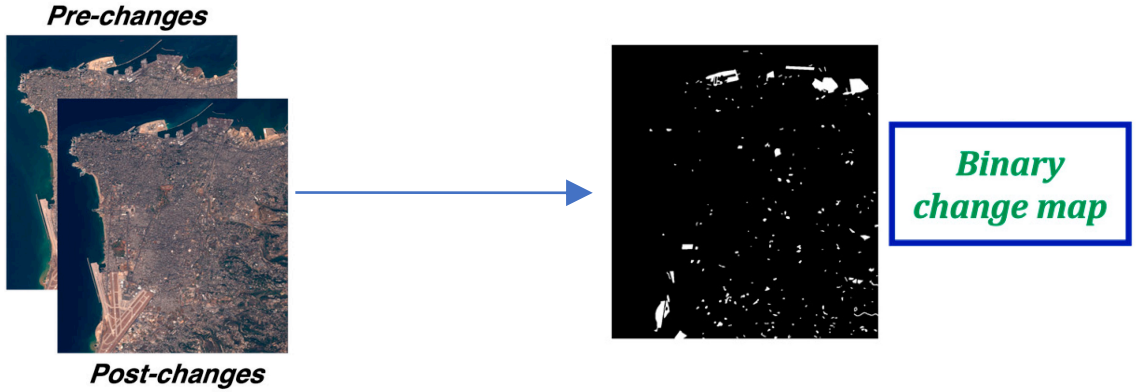
- **LAND COVER CLASSIFICATION**: optimized photogrammetric workflow baed on drone images to perform land-cover classification and change-detection monitoring using DL

3 basic classes





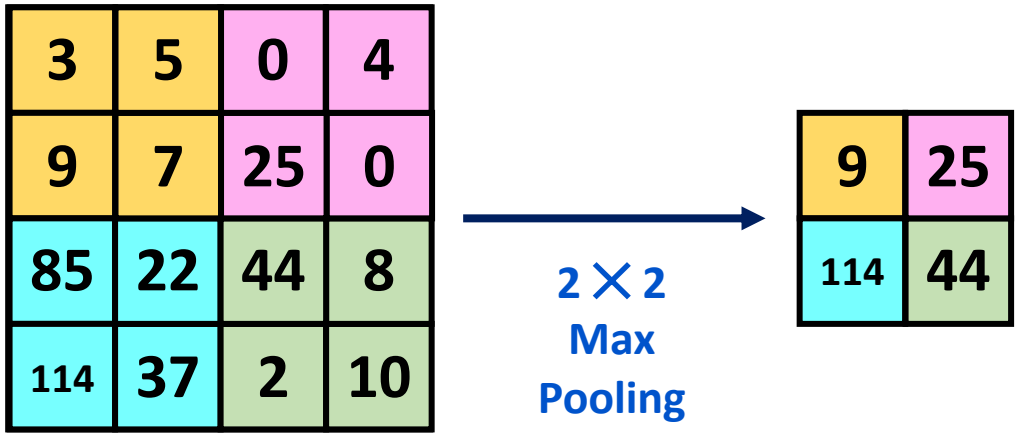
- **URBAN CHANGE DETECTION**: DL agorithms to detect changes in urban area extension through the analysis of multispectral satellite images



Convolutional networks

Convolutional networks (CONVNETS) consist of stacks of **convolution** and **max pooling** layers

apply the **same geometric transformation** to **different** spatial **locations** in the input tensor



- **Spatially downsample** the data
- Keep feature maps to a reasonable size as the number of features increases

CONVNETS can learn

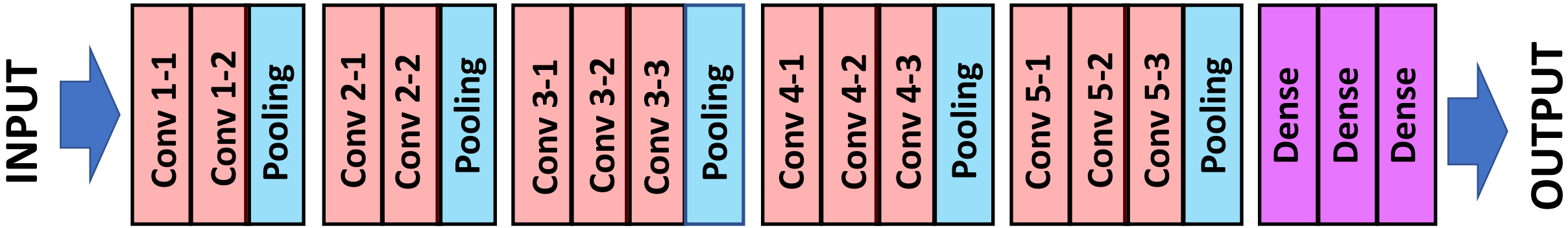
- **translation-invariant** representations
- **spatial hierarchies** of patterns

The pre-trained VGG16 model

Using a pretrained CONVNET (**weights assigned**) to exploit **data augmentation** (rotation, horizontal and vertical flip)

VGG16 keras architecture

- 13 **convolutional** layers, 5 **max pooling** layers and 3 **dense** layers
- 16 in VGG16 means that only **16** layers have weights (**learnable parameter layers**)

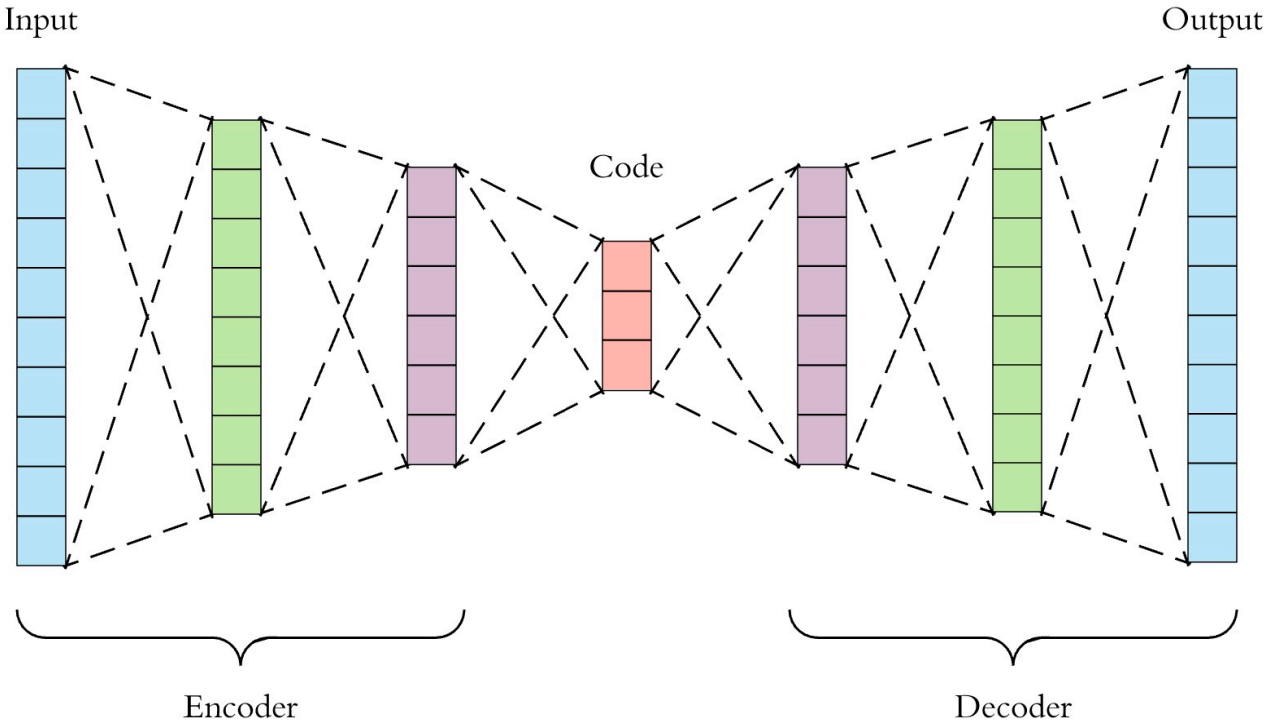


Autoencoders

An **autoencoder** is a neural network made of two conceptual blocks: **encoder** and **decoder**

The input layer is encoded into a **lower-dimensional layer (code)**, which is then decoded to the output layer

The network is trained to **reproduce** a specific class of inputs as faithfully as possible



When the input does not belong to the training class, the autoencoder **fails (in a measurable way)** to reproduce the output

The **decrease in reproduction accuracy** allows to spot an **anomaly**

Anomaly detection with autoencoders @ReCaS-Bari

Use-case: highlight **traffic anomalies in TCP/IP connections** to the **ReCaS-Bari central switch/router**

ADVANTAGES:

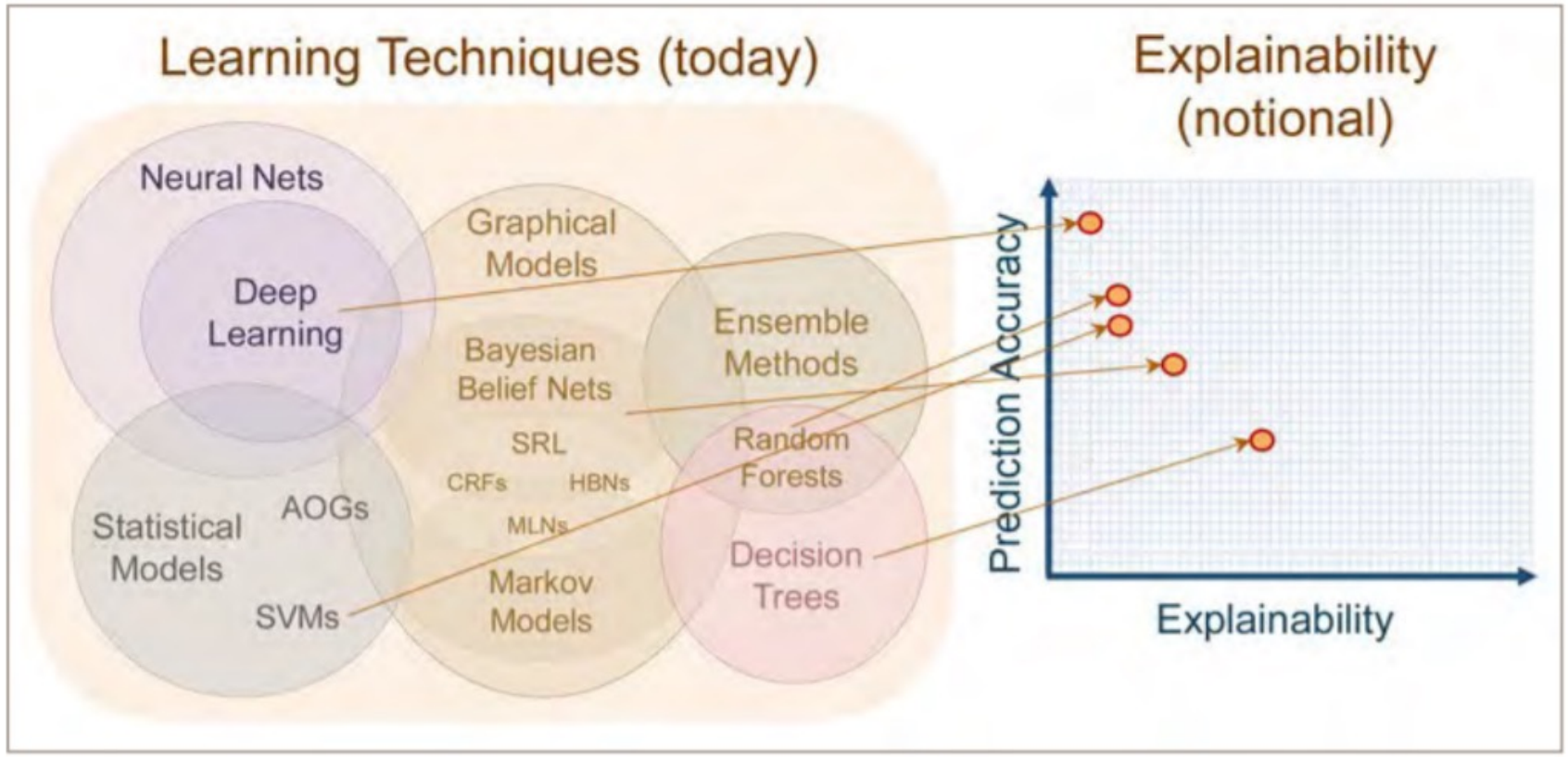
- Only data labelled as "normal" are needed for training
- No need to train the network about the anomalies one wants to detect → the method can find previously **unknown anomalies**
- Very **limited computing power** to run after training

TOOLS AND RESOURCES:

- **Python, Tensorflow** (or any other similar tool), **Jupyter** (optional)
- A decent GPU on a decent server (preliminary tests on a server equipped with **96 CPU cores, 950 GB of RAM, and 4 NVIDIA Tesla V100 GPU**)
- Computational time to **train the network** (once): **minutes/hours**

The performance-interpretability tradeoff

High performance of **complex AI model**s often comes at the **cost** of **interpretability**



Source: DARPA

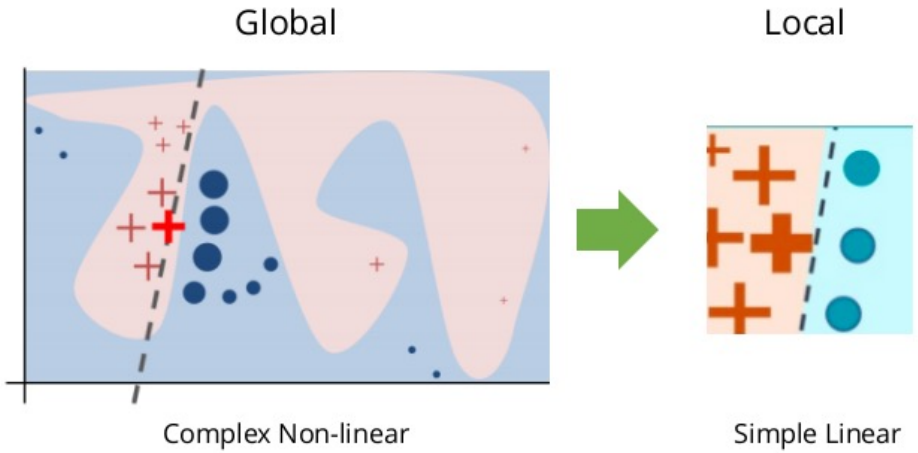
We are interested in understanding how **each predictor** is **contributing to** a given **ML model** and how the different predictors **interact.**

Explainable Artificial Intelligence (XAI) comprises many methods that **combine ML** algorithms **with explanatory techniques** to develop explainable solutions.

Local XAI methods: SHAP

SHAP is a XAI method that provides both **instance feature contribution**, and **overall feature importance**.

Local and **model-agnostic**: predictions are explained at the level of individual instances regardless of the specific ML predictive model.



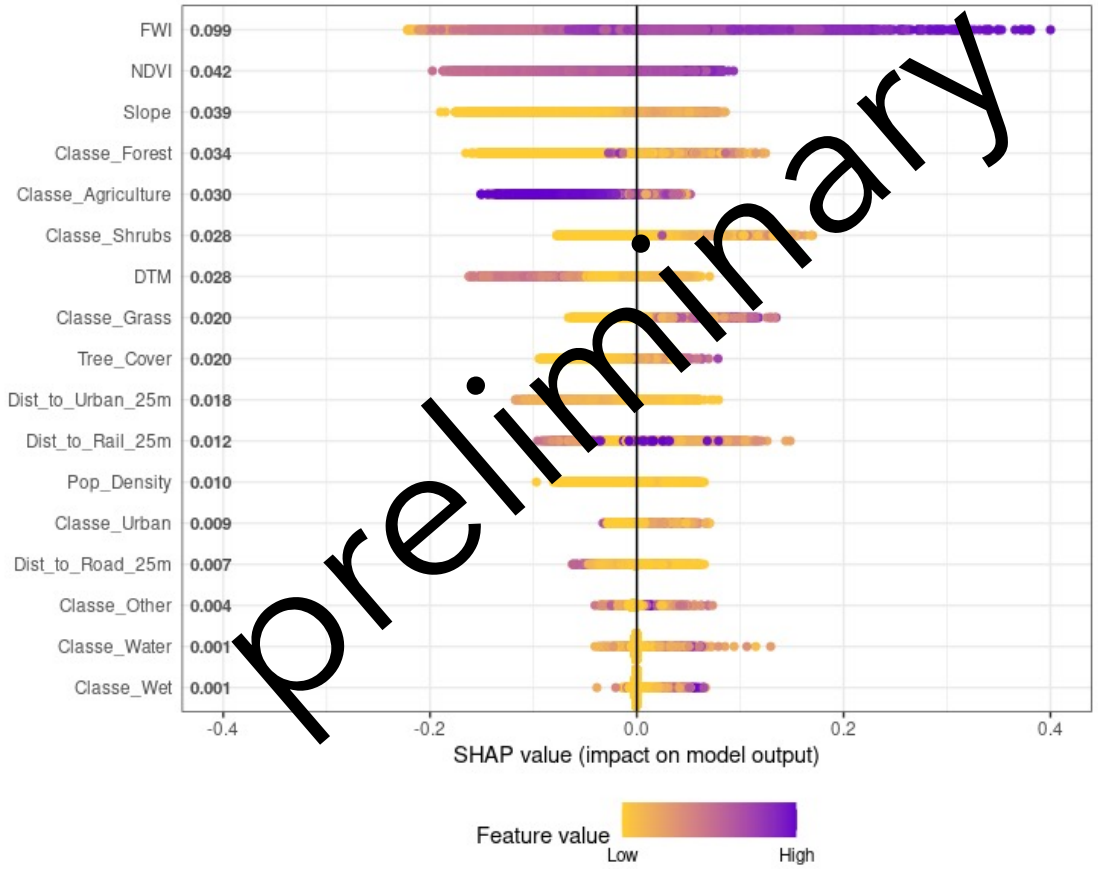
SHAP learns an interpretable linear model g around each test instance $\boldsymbol{p}$ and estimates feature importance locally:

- dataset $[(\boldsymbol{p}_1, y_1), (\boldsymbol{p}_2, y_2), \ldots, (\boldsymbol{p}_T, y_T)]$, with $\boldsymbol{p}_i$ feature vector and y_i target for sample i
- the generic pre-trained model f returns a prediction $f(\boldsymbol{p}_i) = \hat{y}_i$ based on a single input sample $\boldsymbol{p}_i$
- SHAP returns a linear model g to explain f by using a simplified input $\boldsymbol{p}'$ that maps to the original input through a mapping function $\boldsymbol{p} = h_{\boldsymbol{p}}(\boldsymbol{p}')$ trying to ensure $g(\boldsymbol{q}') \approx f(h_{\boldsymbol{p}}(\boldsymbol{q}'))$ whenever $\boldsymbol{q}' \approx \boldsymbol{p}'$

SHAP values and prediction of wildfire occurrence in Italy between 2007 and 2017

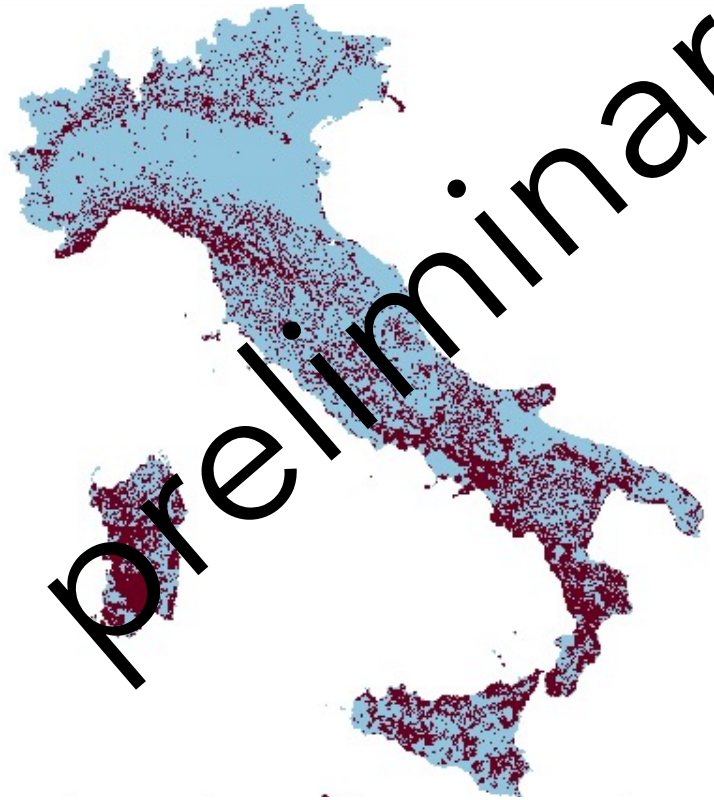
SHAP values quantify the **contribution** of **territorial features** to RF predictions of **wildfire occurrence**

SHAP summary plot



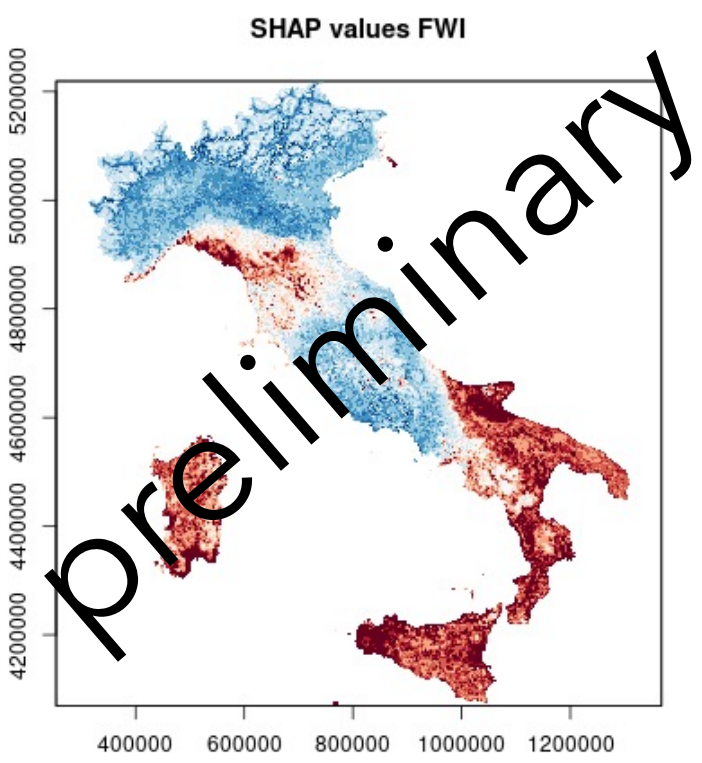
preliminary

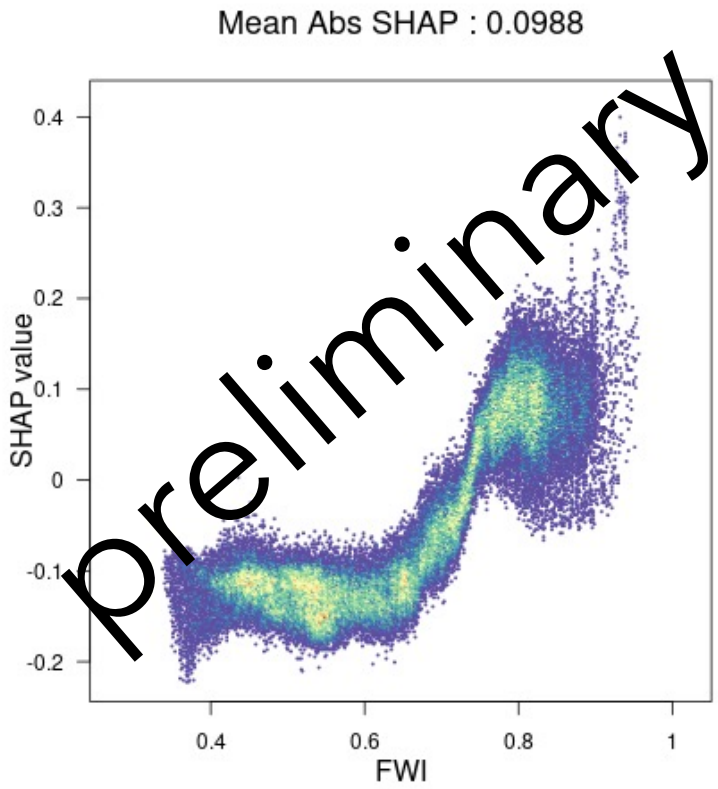
SHAP values computed for a binary classification model FIRE YES/NO

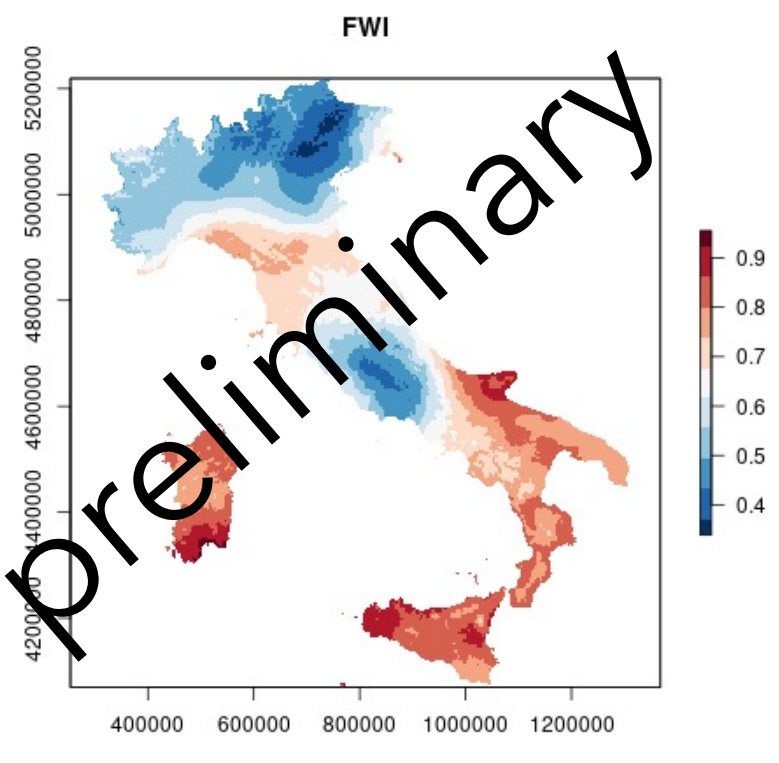


preliminary

SHAP values and prediction of wildfire occurrence in Italy between 2007 and 2017







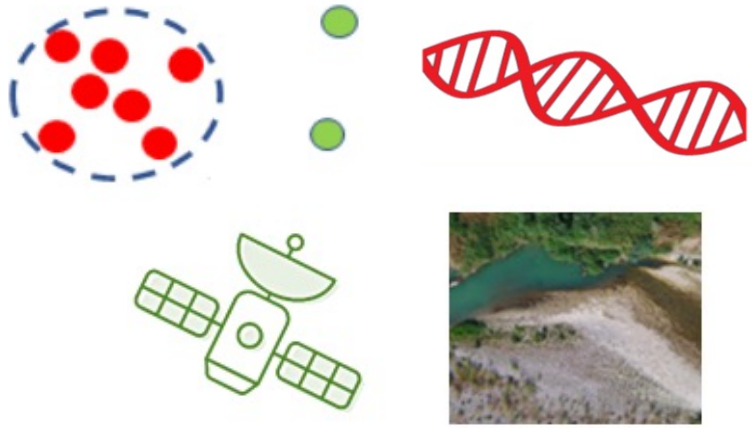
The SHAP analysis highlights the pivotal role of the **Fire Weather Index (FWI)**, a meteorologically based index used worldwide to estimate fire danger.

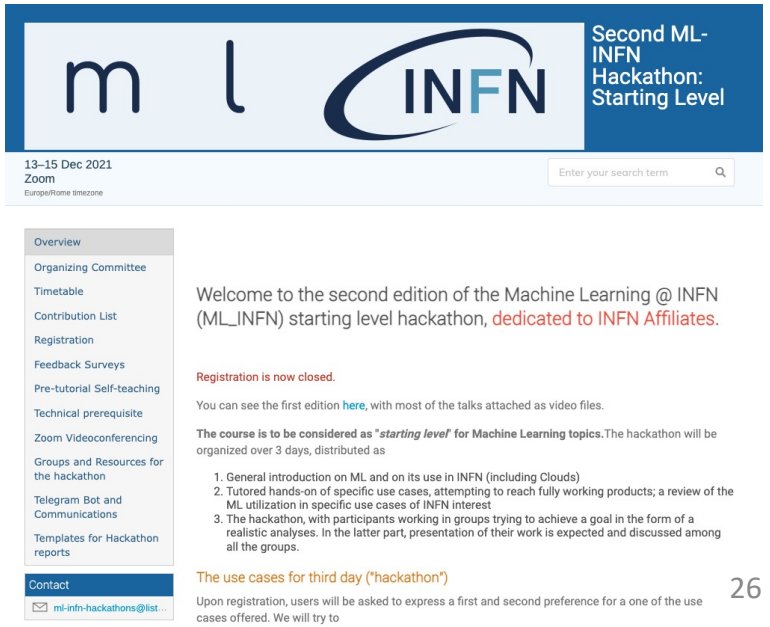
Conclusions

- **Artificial Intelligence (AI)** methods are characterized by a very wide range of potential **interdisciplinary applications**

- The **INFN groups** are developing an increasing expertise in applying cutting-edge Artificial Intelligence techniques to reserarch **areas of general interest** such as **genomics, remote sensing and anomaly detection**

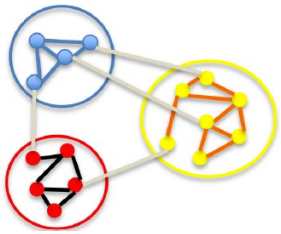
★INFN Bari joined the Second Edition of **ML_INFN Hackathon** on 13-15 December 2021, which offered students a tutored-hands-on specific **ML use cases of INFN interest** (Medical Physics, Classification@LHC and Autoencoding for VIRGO/GW)

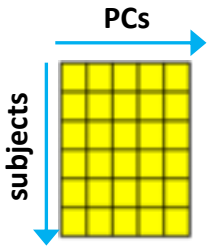


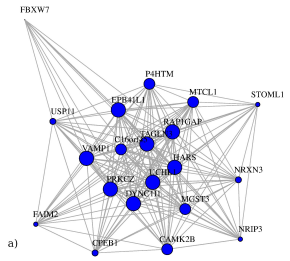


Credits

Genomics

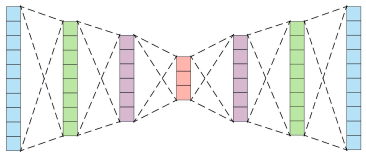


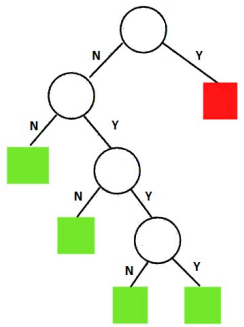




A. Monaco, E. Pantaleo, N. Amoroso, A. Lombardi, S. Tangaro, R. Bellotti and collaborators

Anomaly detection in TCP/IP connections to the **ReCaS-Bari central switch/router**





D. Diacono, A. Italiano, S. Nicotri, V. Spinoso, G. Vino

Remote sensing

G. Miniello and collaborators

R. Cilli, N. Amoroso, A. Lombardi, R. Bellotti and collaborators

A. Di Pilato, A. Pompili, A. Di Florio and collaborators

Grazie per l’attenzione!