UNIVERSITÀ DEGLI STUDI DI PADOVA

Machine Learning 2024/2025

AMCO
ARTIFICIAL INTELLIGENCE, MACHINE LEARNING AND CONTROL RESEARCH GROUP

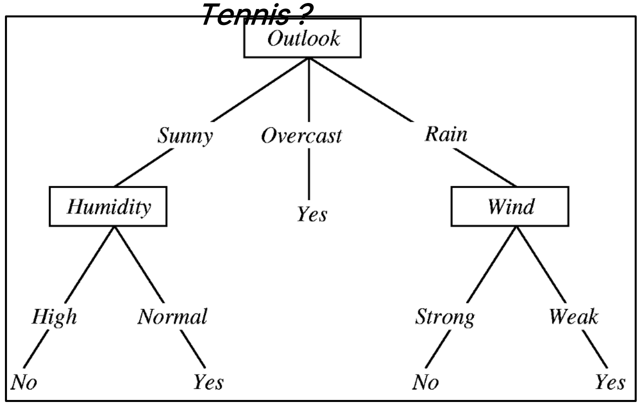
Lecture #34 eXplainable Artificial Intelligence (XAI)

Gian Antonio Susto

Feature statistics, Model specific - Tree-based methods

- Decision Trees are ‘classical’ solutions to supervised tasks
- The classification is done by following a tree-structure:
  - each interior node is a input variable (and there are edges to children for each possible value of that variable)
  - each leaf is a class
- Advantages
  - ‘Easily interpretable’
  - They require no data normalization
  - The outcome computation is almost immediate

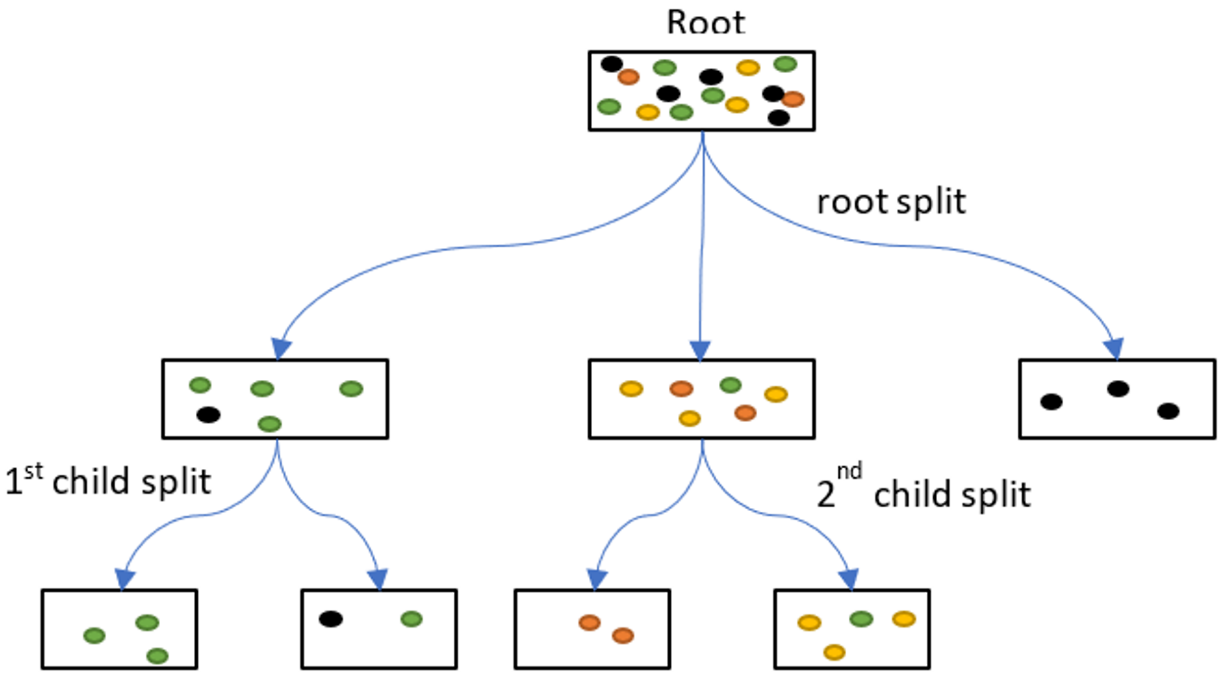
Playing Tennis ?



From: ‘Machine Learning’, Mitchell 1997

Feature statistics, Model specific - Tree-based methods

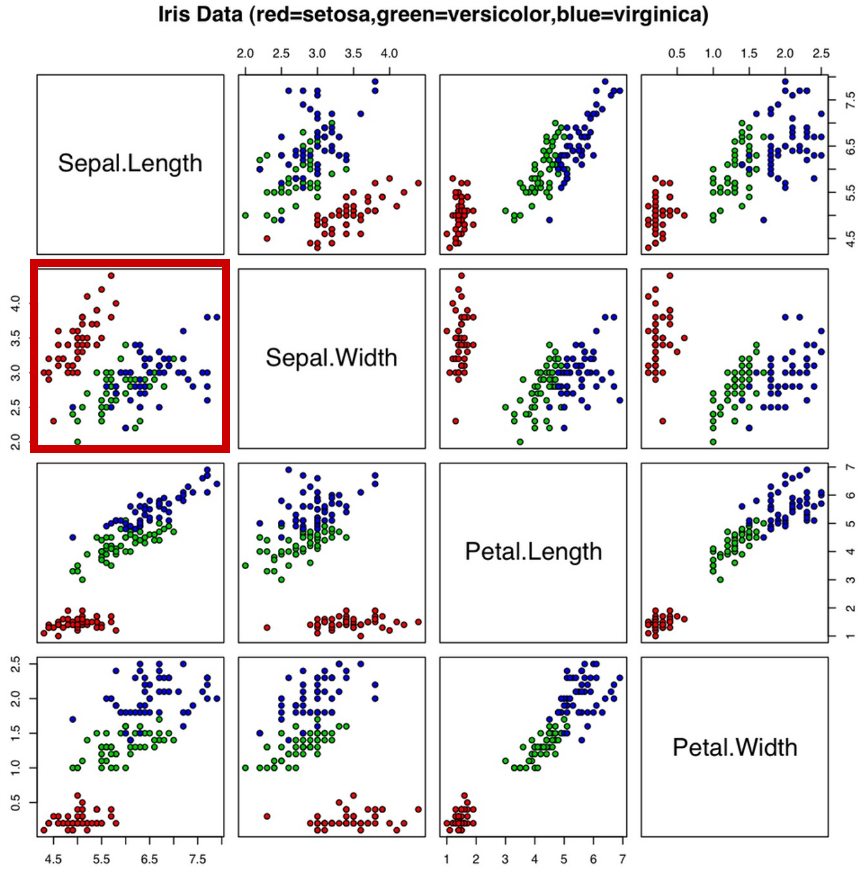
- DTs are constructed with *top-down* approaches: at each step of the algorithm is to choose a variable that 'best' splits the set of observations (recursive partitioning)
- Many criteria:
  - entropy and information gain
  - Gini impurity / Mean Decrease in impurity
  - Variance reduction



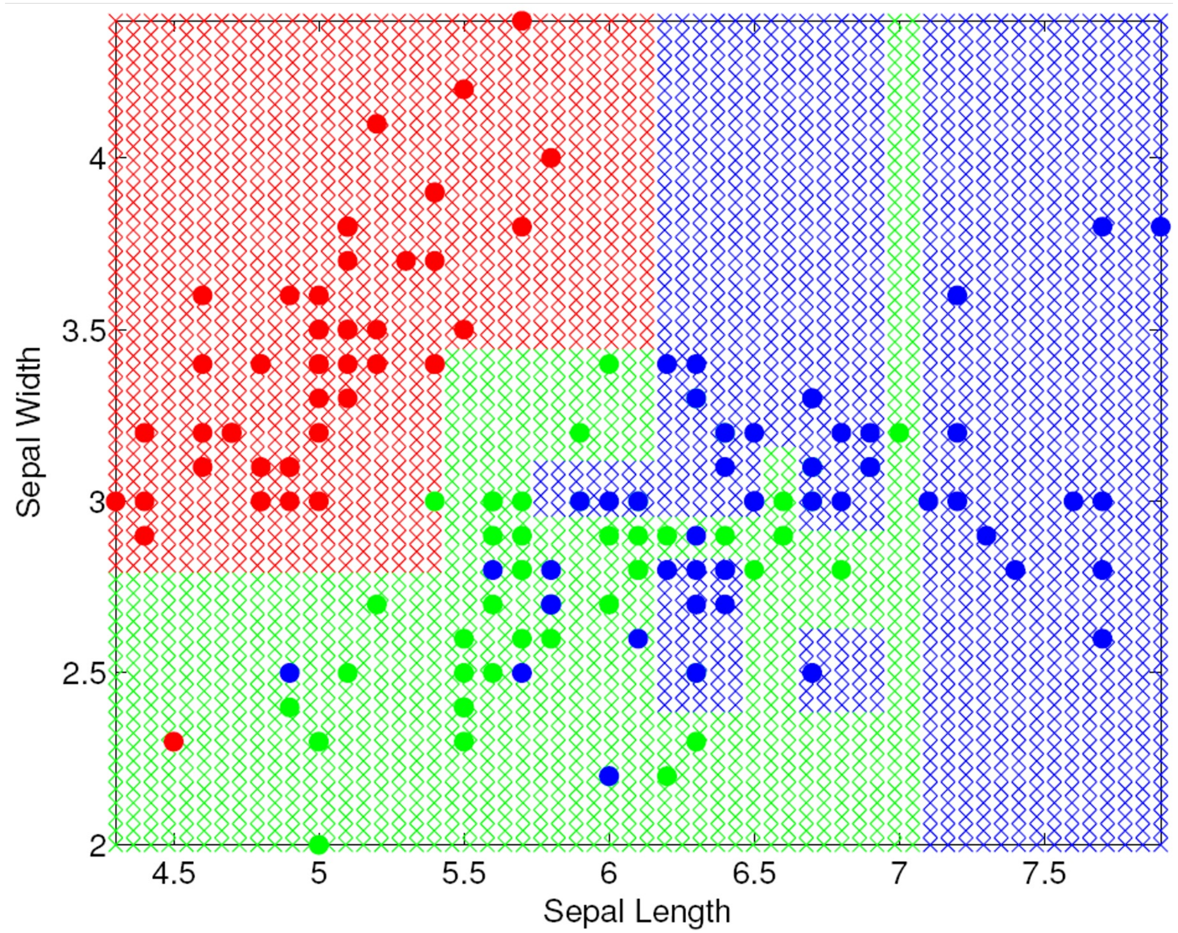
Feature statistics, Model specific - Tree-based methods

Example: ‘Iris Classification’ dataset, Ronald Fisher (1936) - UCI ML Repository

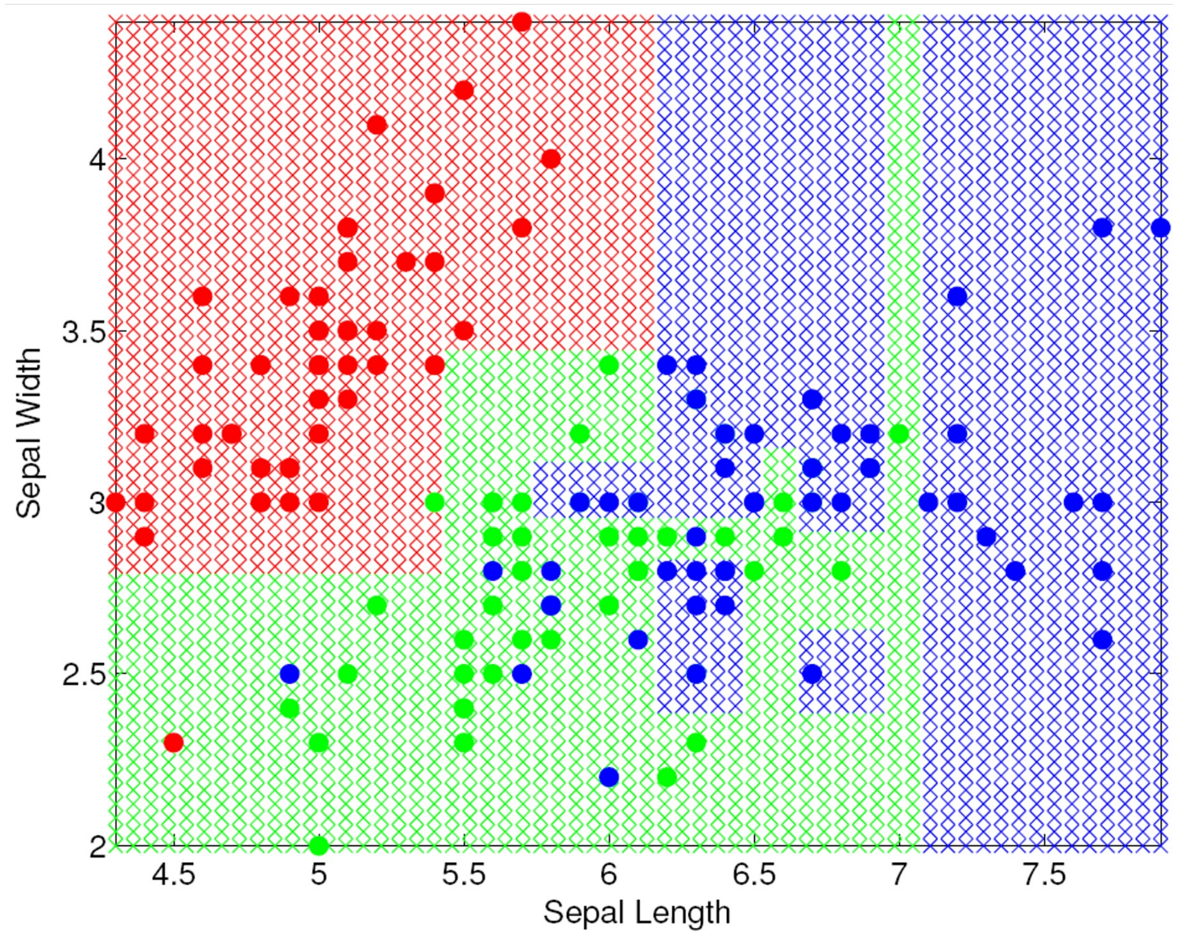
L = 3 classes problem: classify Setosa, Versicolour and Virginica iris from data containing sepal and petal width and length – n = 150 samples, p = 4 variables

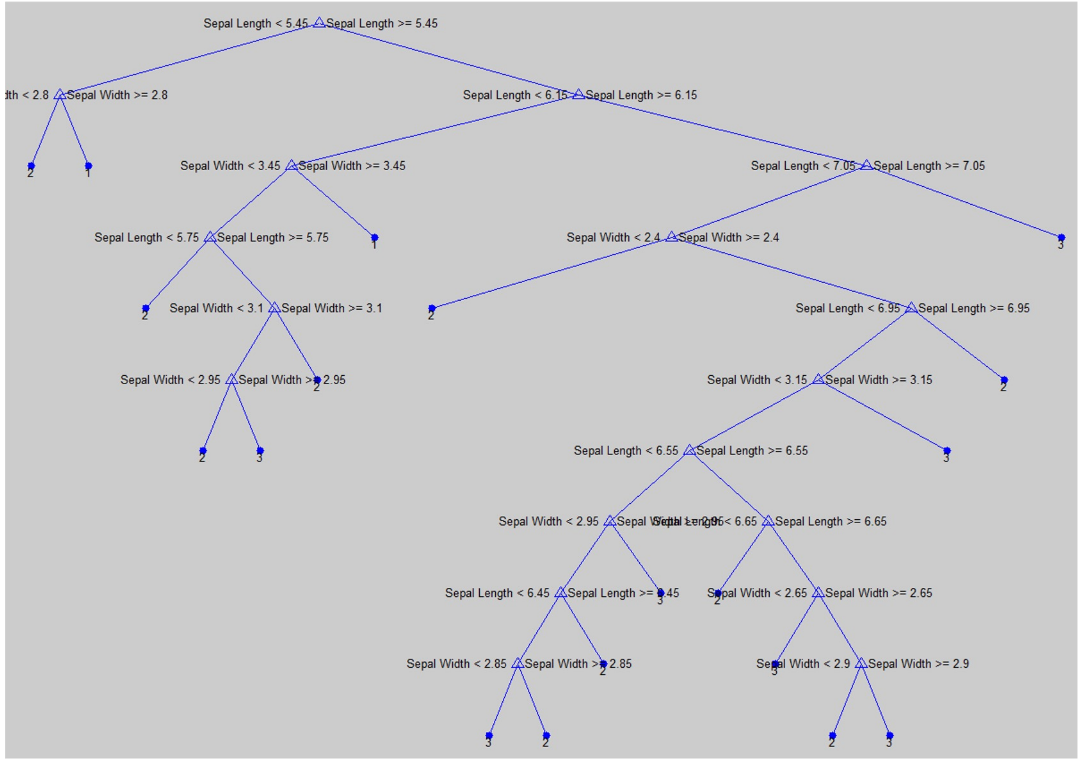


Feature statistics, Model specific - Tree-based methods



Feature statistics, Model specific - Tree-based methods



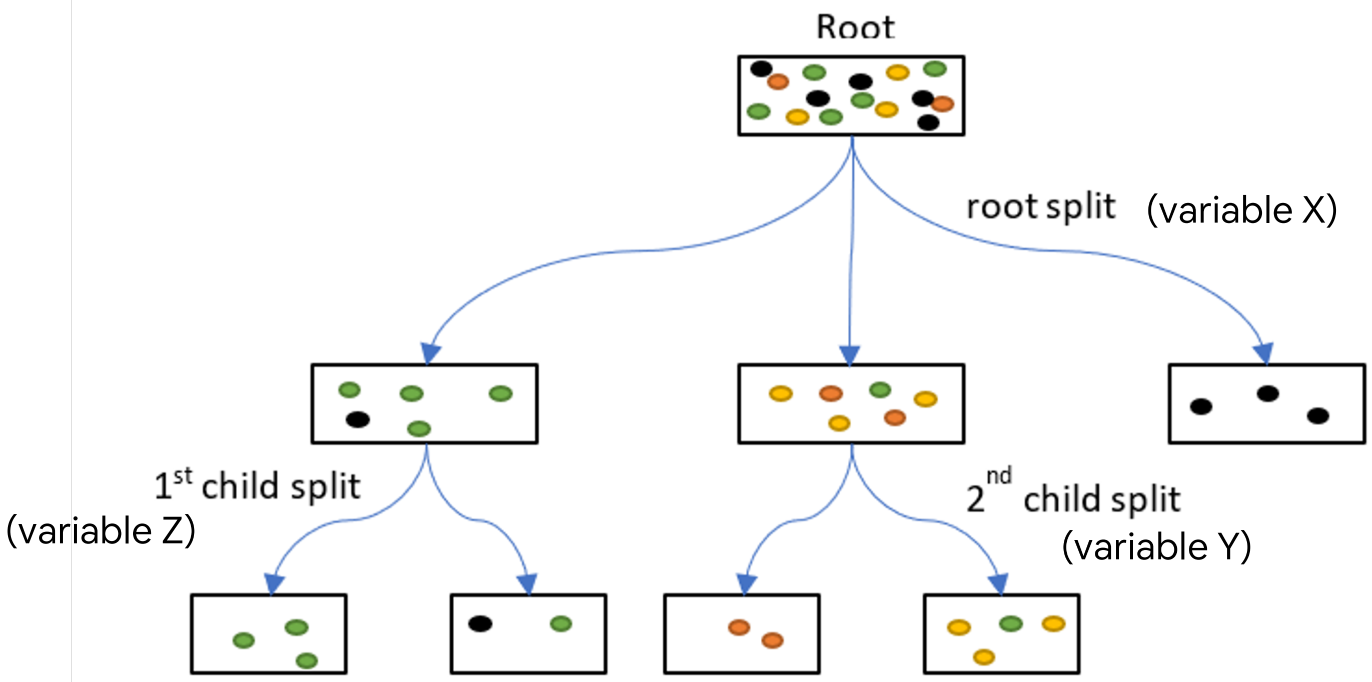


19 splits!

Complete! Interpretable?

Feature statistics, Model specific - Tree-based methods

- Also in this case we would like to provide feature statistics summary: what are the most important variables?
- **Gini Importance** or **Mean Decrease in Impurity (MDI)** calculates each feature importance as the sum over the number of splits that include the feature, proportionally to the number of samples it splits



Recap: Gini Index / Entropy / Information Gain

The Gini Index (or Gini Impurity) is a measure of how impure or mixed a dataset is.

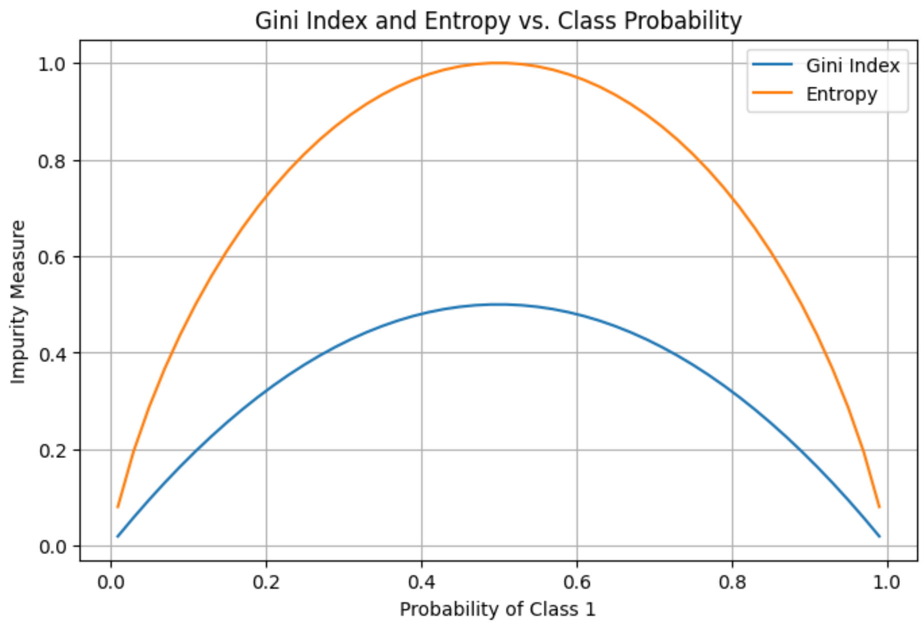
For a dataset S with c classes:

$$Gini(S) = 1 - \sum_{i=1}^{C} p_i^2$$

$$Gini_{split} = \frac{n_{left}}{n} \cdot Gini(left) + \frac{n_{right}}{n} \cdot Gini(right)$$

$$Entropy(S) = -\sum_{i=1}^{c} p_i \log_2(p_i)$$

$$IG(S, A) = Entropy(S) - \sum_{v \in Values(A)} \frac{|S_v|}{|S|} \cdot Entropy(S_v)$$



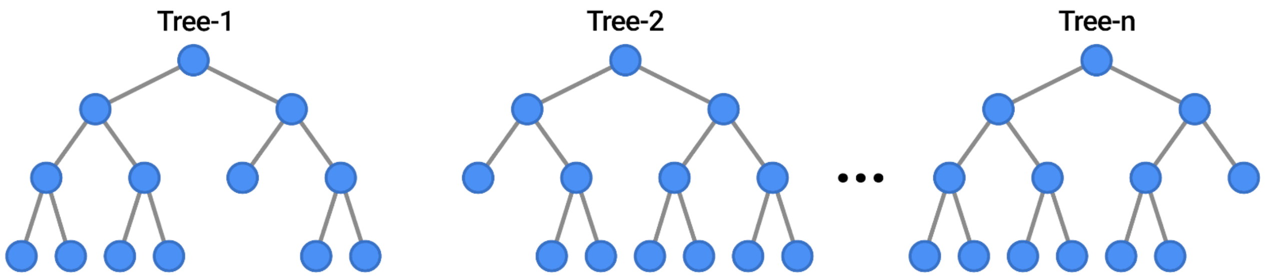
Recap: Random Forest (RF)

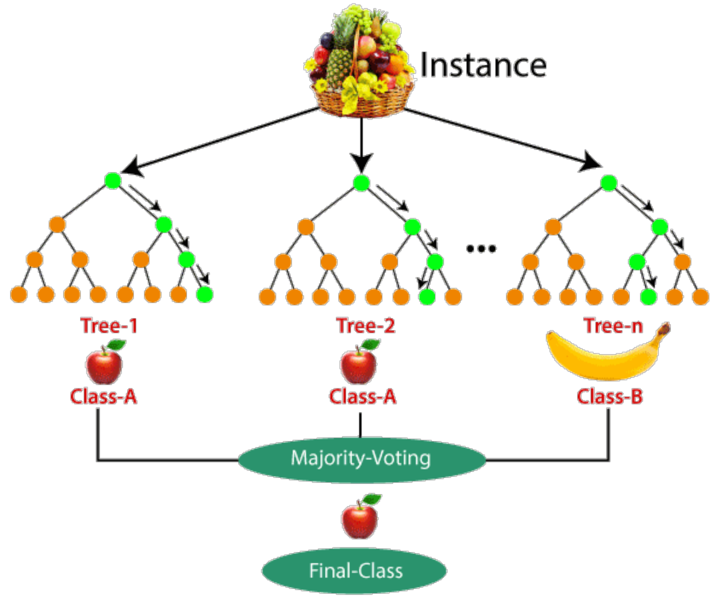
A RF is composed by many ‘weak’ learners (decision trees): we cleverly combine DTs reducing overfitting!

We construct slightly different DTs (more on this later) and, in classification, we decide by a majority-voting (we choose following the mode) the final class. In regression, the final decision is the average.

This in an ‘ensemble’ approach: we combine multiple models (often called base learners or weak learners) to produce a stronger model.

EXAMPLES



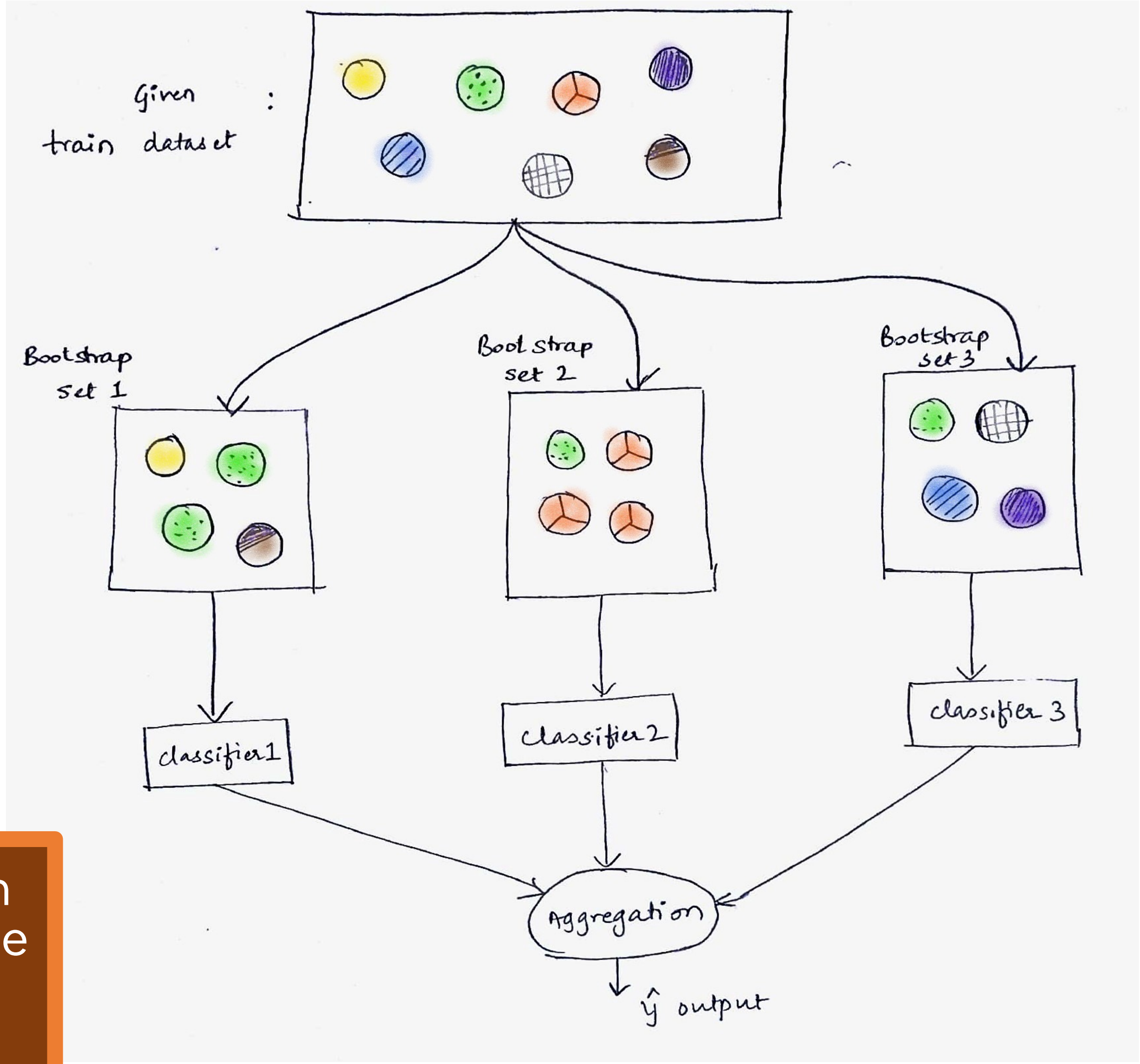


🔧 Recap: How to Build a Random Forest

Let's assume you want to build a forest with T trees.

For each tree:

- Sample the dataset with replacement (bootstrap sample). This procedure is called Bagging (bootstrap aggregating).
- Build a decision tree: but at each split, instead of evaluating all features, pick a random subset (e.g., $\sqrt{p}$). This procedure is called Feature Bagging.

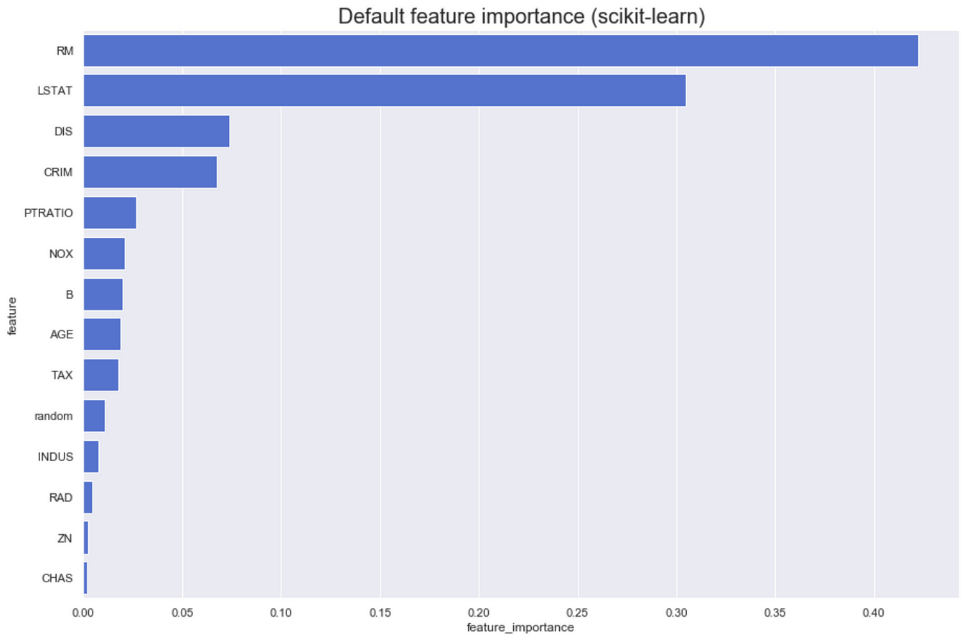


The reason for doing this is the correlation of the trees in an ordinary bootstrap sample: if one or a few features are very strong predictors for the response variable (target output), these features will be selected in many of the T trees, causing them to become correlated.

RF: feature importance

Feature importance reflects how useful or valuable each feature is for making predictions in a model. For decision trees (and ensembles like Random Forests), it's typically based on:

🔍 How much each feature decreases impurity (e.g., Gini index or entropy) when it’s used to split the data



📊 Intuition

- If a feature is consistently chosen for important splits (i.e., it helps reduce impurity a lot), it gets high importance.

- Features that are rarely used or don’t reduce impurity much get low or zero importance.

RF: feature importance

Feature importance reflects how useful or valuable each feature is for making predictions in a model. For decision trees (and ensembles like Random Forests), it's typically based on:

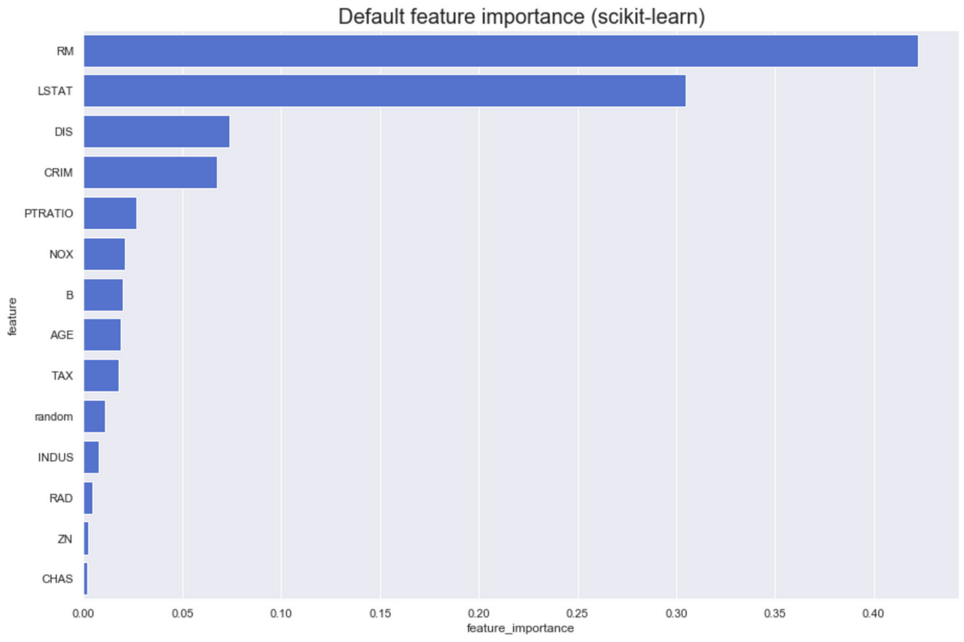
🔍 How much each feature decreases impurity (e.g., Gini index or entropy) when it’s used to split the data

📊 Intuition

- If a feature is consistently chosen for important splits (i.e., it helps reduce impurity a lot), it gets high importance.

- Features that are rarely used or don’t reduce impurity much get low or zero importance.

It is a ’global’ approach: provide us with info on the whole model structure



RF: feature importance - Derivation

Let's consider the Gini impurity, and we have a decision tree:

1. At every split, the algorithm calculates how much that split reduces impurity:

$$\Delta Gini = Gini(\text{parent}) - \left(\frac{n_{left}}{n_{\text{P}}} \cdot Gini(\text{left}) + \frac{n_{right}}{n_{\text{P}}} \cdot Gini(\text{right})\right)$$

2. The contribution of a feature is the sum of all impurity decreases where that feature was used to split:

$$Importance(feature) = \sum_{\text{nodes using feature}} \frac{n_p}{n_{TOT}} \Delta Gini$$

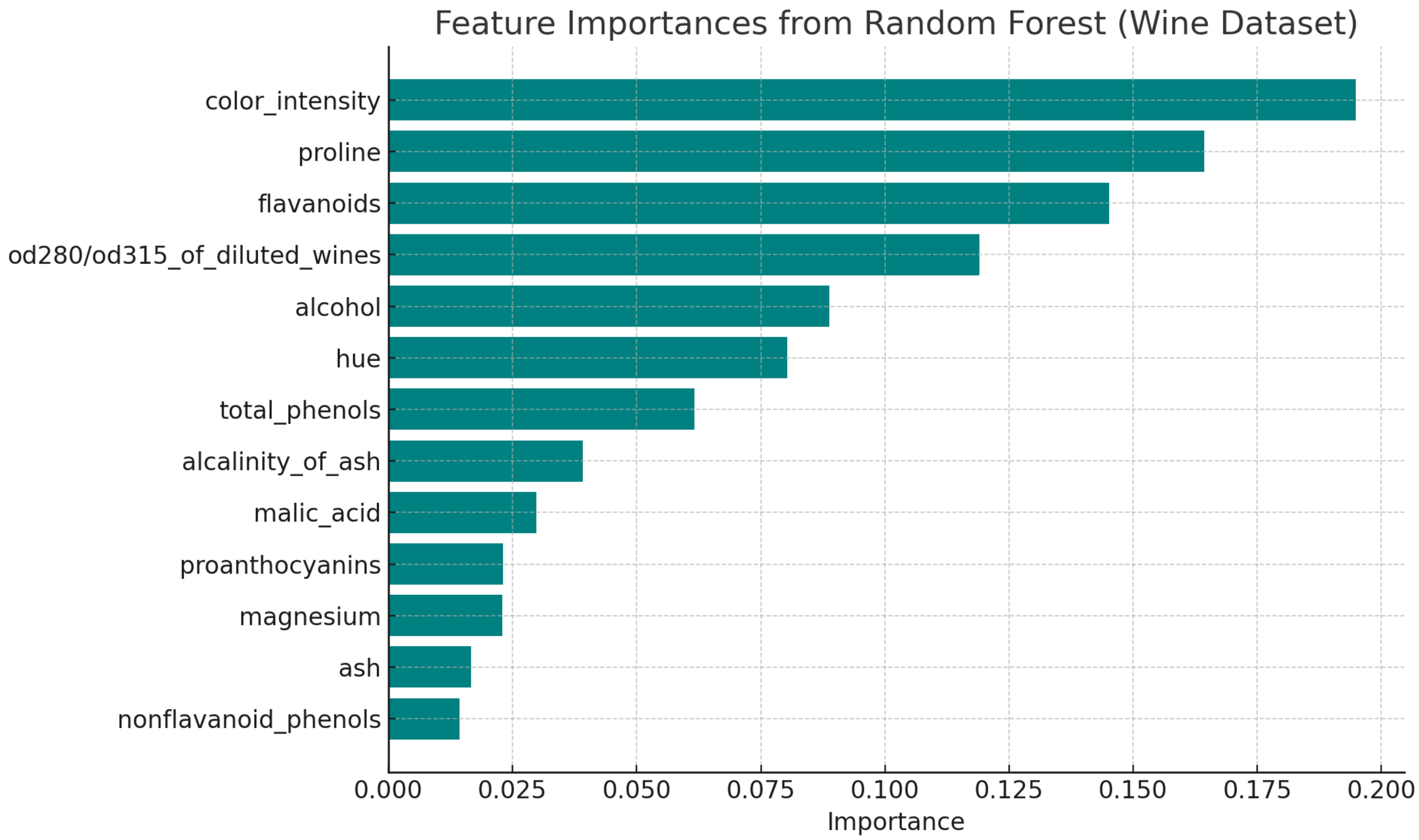
3. In a Random Forest, we average this importance over all the trees in the forest.

4. (Optional) the imp

$$\text{Normalized Importance} = \frac{\text{Raw Importance}}{\sum \text{Raw Importances}}$$

Where:
n_p is the number of samples at the parent node
n_{TOT} is the total number of samples

🍷 On the wine dataset



Feature statistics, Model specific - Tree-based methods (Optional)

Even though RF consists of a collection of Decision Trees (which are recognized as interpretable models), its interpretation isn't as trivial as it may seem

The most widely used feature importance measure in this context is again the Mean Decrease Impurity (MDI): think about averaging MDI of the individual Decision Trees

REMARK ⚠

Problem: MDI measure suffers from so-called "*feature selection bias*", i.e. it may erroneously assign high MDI values to features that are not highly correlated to the output

Feature statistics, Model specific - Tree-based methods (Optional)

Ev sists of a collection of Decision Trees (which are
re t as trivial as it
ma

REMARK ⚠

Solution: "[*A Debiased MDI Feature Importance Measure for Random Forests*]", by Li et al.

Th context is again the Mean Decrease Impurity (MDI): think about averaging MDI of the individual Decision Trees

REMARK ⚠

Problem: MDI measure suffers from so-called "*feature selection bias*", i.e. it may erroneously assign high MDI values to features that are not highly correlated to the output

Feature statistics, Model specific - Tree-based methods (Optional)

So, we have a robust model-specific method to compute feature importance for RF... are we done?

Not really... in several applications we may need to detect high-order interactions between features!

Feature statistics, Model specific - Tree-based methods (Optional)

So, we have a robust model-specific method to compute feature importance for RF... are we done?

Not really... in several applications we may need to detect high-order interactions between features!

REMARK ⚠

Solution: "*iterative Random Forests to discover predictive and stable high-order interactions*", by Basu et al. (THIS IS A 'NEW' INTERPRETABLE-ORIENTED MODEL)

Feature statistics, Model specific - Tree-based methods (Optional)

For other ensemble tree-based methods, similar approaches can be used.

The are other approaches, for example:

Boruta implements a different feature selection algorithm. It randomly permutes variables like Permutation Importance (next slides) does, but performs on all variables at the same time and concatenates the shuffled features with the original ones. The concatenated result is used to fit the model.

Miron B. Kursa, Witold R. Rudnicki (2010). Feature Selection with the Boruta Package.

Journal of Statistical Software, 36(11) , p. 1–13.

Taxonomy: model-agnostic vs model-specific

Portability of the interpretability algorithm?

In principle, it can be applied to all ML models

MODEL-AGNOSTIC

It is tailored for a specific ML model

MODEL-SPECIFIC

Feature statistics, Model-agnostic - Permutation Importance

Permutation Importance

Idea: evaluate performance degradation after all values of a specific feature have been shuffled (over all data points)

- post-hoc or intrinsic?
- model-agnostic or model-specific?
- global or local?

This method outputs so-called "*feature importance*" for each feature: a scalar number, the greater the value the more important the corresponding feature

Feature statistics, Model-agnostic - Permutation Importance

Permutation Importance

Idea: evaluate performance degradation after all values of a specific feature have been shuffled (over all data points)

- post-hoc
- model-agnostic (even though it was initially introduced for Random Forests)
- global

This method outputs so-called "*feature importance*" for each feature: a scalar number, the greater the value the more important the corresponding feature

Feature statistics, Model-agnostic - Permutation Importance

Permutation Importance

<u>Procedure</u>

1. Train a model and get predictions ("original predictions")

x_1	x_2	ground truth	original pred	pred with shuffled x_1	pred with shuffled x_2
3.4	7.5	0	1		
2.7	7.7	1	1		
3.5	6.9	1	1		
1.5	6.3	0	0		
1.8	6.4	1	1		

Feature statistics, Model-agnostic - Permutation Importance

Permutation Importance

Procedure

1. Train a model and get predictions ("original predictions")
2. evaluate performance (e.g. classification accuracy)

x_1	x_2	ground truth	original pred	pred with shuffled x_1	pred with shuffled x_2
3.4	7.5	0	1		
2.7	7.7	1	1		
3.5	6.9	1	1		
1.5	6.3	0	0		
1.8	6.4	1	1		

ERROR: 20%

Feature statistics, Model-agnostic - Permutation Importance

Permutation Importance

Procedure

1. Train a model and get predictions (“original predictions”)
2. evaluate performance (e.g. classification accuracy)
3. select a specific feature and...

x_1	x_2	ground truth	original pred	pred with shuffled x_1	pred with shuffled x_2
3.4	7.5	0	1		
2.7	7.7	1	1		
3.5	6.9	1	1		
1.5	6.3	0	0		
1.8	6.4	1	1		

ERROR: 20%

Feature statistics, Model-agnostic - Permutation Importance

Permutation Importance

Procedure

1. Train a model and get predictions ("original predictions")
2. evaluate performance (e.g. classification accuracy)
3. select a specific feature and... shuffle the values over all data points

x_1	x_2	ground truth	original pred	pred with shuffled x_1	pred with shuffled x_2
2.7	7.5	0	1		
3.4	7.7	1	1		
1.5	6.9	1	1		
1.8	6.3	0	0		
3.5	6.4	1	1		

ERROR: 20%

Feature statistics, Model-agnostic - Permutation Importance

Permutation Importance

Procedure

1. Train a model and get predictions ("original predictions")
2. evaluate performance (e.g. classification accuracy)
3. select a specific feature and... shuffle the values over all data points
4. get predictions for these new (artificially created) data points and evaluate performance

x_1	x_2	ground truth	original pred	pred with shuffled x_1	pred with shuffled x_2
2.7	7.5	0	1	1	
3.4	7.7	1	1	1	
1.5	6.9	1	1	0	
1.8	6.3	0	0	0	
3.5	6.4	1	1	1	
		ERROR:	20%	40%	

Feature statistics, Model-agnostic - Permutation Importance

Permutation Importance

<u>Procedure</u>

1. Train a model and get predictions ("original predictions")
2. evaluate performance (e.g. classification accuracy)
3. select a specific feature and... shuffle the values over all data points
4. get predictions for these new (artificially created) data points and evaluate performance

Repeat points 3 and 4 for all features

x_1	x_2	ground truth	original pred	pred with shuffled x_1	pred with shuffled x_2
3.4	6.9	0	1	1	1
2.7	6.3	1	1	1	0
3.5	7.5	1	1	0	1
1.5	6.4	0	0	0	1
1.8	7.7	1	1	1	0
		ERROR:	20%	40%	80%

Feature statistics, Model-agnostic - Permutation Importance

Permutation Importance

<u>Evaluation</u>

Notice that when we shuffled feature x_2, the performance got significantly worse (80% error) than when we shuffled feature x_1 (40% error)

→ x_2 is more important than x_1

ground truth	original pred	pred with shuffled x_1	pred with shuffled x_2
0	1	1	1
1	1	1	0
1	1	0	1
0	0	0	1
1	1	1	0
ERROR:	20%	40%	80%

Feature statistics, Model-agnostic - Permutation Importance

Permutation Importance

Note that by shuffling the values assumed by a specific feature, we break the underlying relation between that feature and the true output

⟶ if such relation is strong (i.e. if that feature is actually important to predict the output), the shuffling will dramatically decrease the performance

Feature statistics, Model-agnostic - Permutation Importance

<u>Disadvantages:</u>

- since we need to repeat the procedure for all the features, it may be computationally costly with high-dimensional data (i.e. high number of features); and since shuffling is random, we may even want to do it multiple times, making the procedure even more costly

- we shuffle one feature at a time, so we are not taking into consideration the correlations between features; this may lead to new (artificially created) data points which are improbable in practice

Feature statistics, Model-agnostic - Permutation Importance

Disadvantages:

- since we need to repeat the procedure for all the features, it may be computationally costly with high-dimensional data (i.e. high number of features); and since shuffling is random, we may even want to do it multiple times, making the procedure even more costly

- we shuffle one feature at a time, so we are not taking into consideration the correlations between features; this may lead to new (artificially created) data points which are improbable in practice

Let’s see it with a practical example...

Feature statistics, Model-agnostic - Permutation Importance

Example (I will not be ‘ethical’ or ‘fair’ in the following…)

We trained a ML model to predict the probability of a heart attack based on the following features:

- favourite football team
- emotional state
- weight
- systolic blood pressure
- diastolic blood pressure

Feature statistics, Model-agnostic - Permutation Importance

Example (I will not be ‘ethical’ or ‘fair’ in the following…)

We trained a ML model to predict the probability of a heart attack based on the following features:

- favourite football team
- emotional state
- weight
- systolic blood pressure
- diastolic blood pressure

We apply Permutation Importance on the feature “favourite football team”…

Feature statistics, Model-agnostic - Permutation Importance

ID	FAVOURITE TEAM	EMOTIONAL STATE	WEIGHT	SYSTOLIC BLOOD PRESSURE	DIASTOLIC BLOOD PRESSURE	P(HEART ATTACK)
Gian Antonio		Neutral	75	117	78	0.35
Mattia		Excited	70	130	83	0.23
Marco		Sad	92	105	72	0.70
Felice		Very happy	67	112	80	0.63
...	...	...	...	...	...	...

Feature statistics, Model-agnostic - Permutation Importance

ID	FAVOURITE TEAM	EMOTIONAL STATE	WEIGHT	SYSTOLIC BLOOD PRESSURE	DIASTOLIC BLOOD PRESSURE	P(HEART ATTACK)
Gian Antonio	JUVENTUS	Neutral	75	117	78	0.35
Mattia		Excited			83	0.23
Marco	ACM 1899	Sad			72	0.70
Felice	N	Very happy	67	112	80	0.63
...	...	...	...	...	...	...

SHUFFLE

Feature statistics, Model-agnostic - Permutation Importance

ID	FAVOURITE TEAM	EMOTIONAL STATE	WEIGHT	SYSTOLIC BLOOD PRESSURE	DIASTOLIC BLOOD PRESSURE	P(HEART ATTACK)
Gian Antonio		Neutral	75	117	78	0.35
Mattia		Excited	70	130	83	0.23
Marco		Sad	92	105	72	0.70
Felice		Very happy	67	112	80	0.63
...	...	...	...	...	...	...

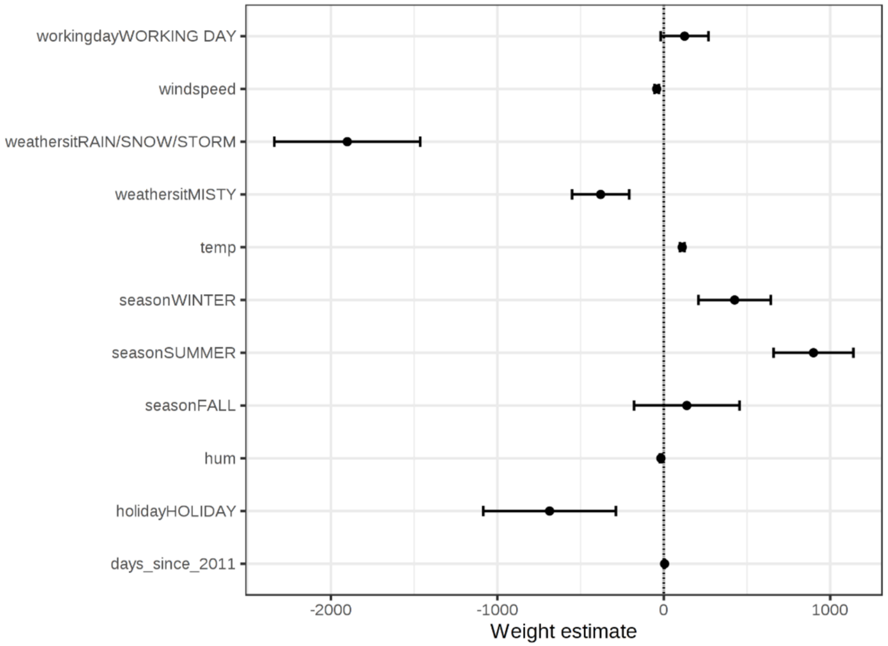
Feature statistics, Model-agnostic - Permutation Importance

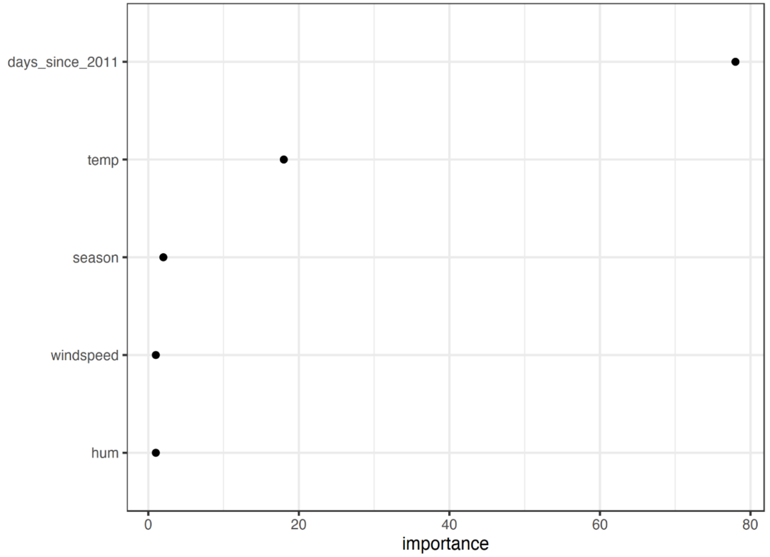
ID	FAVOURITE TEAM	EMOTIONAL STATE	WEIGHT	SYSTOLIC BLOOD PRESSURE	DIASTOLIC BLOOD PRESSURE	P(HEART ATTACK)
Gian Antonio		Neutral	70	117	78	0.35
Mattia		Excited	75	130	83	0.23
Marco		Sad	92	105	72	0.70
Felice		Very happy	67	112	80	0.63
...	...	...	...	...	...	...

Feature statistics, Model-agnostic - Permutation Importance

ID	FAVOURITE TEAM	EMOTIONAL STATE	WEIGHT	SYSTOLIC BLOOD PRESSURE	DIASTOLIC BLOOD PRESSURE	P(HEART ATTACK)
Gian Antonio		Neutral	70	117	78	0.35
Mattia		Excited	75			0.23
Marco		Sad	92			0.70
Felice		Very happy	67			0.63
...	...	...	...			...

Types of interpretations so far...





ground truth	original pred	pred with shuffled x_1	pred with shuffled x_2
0	1	1	1
1	1	1	0
1	1	0	1
0	0	0	1
1	1	1	0
ERROR:	20%	40%	80%

Feature Visualization, Model-agnostic - PDPs

Partial Dependence Plots

Idea: show the marginal effect a feature (or pair of features) has on the prediction

- post-hoc
- model-agnostic
- global

As the name suggests, this method outputs a plot: a curve in the case of a single feature or a surface in the case of a pair of features (usually displayed as a heat map)

Feature Visualization, Model-agnostic - PDPs

Partial Dependence Plots

$$\hat{f}_{x_S}(x_S) = E_{x_C}\left[\hat{f}(x_S, x_C)\right] = \int \hat{f}(x_S, x_C)\,d\mathbb{P}(x_C)$$

$$\hat{f}_{x_S}(x_S) = \frac{1}{n}\sum_{i=1}^{n} \hat{f}(x_S, x_C^{(i)})$$

partial function

selected feature(s)

other feature(s)

The partial function is a function of the feature(s) we are interested in for the analysis ("*selected feature(s)*")

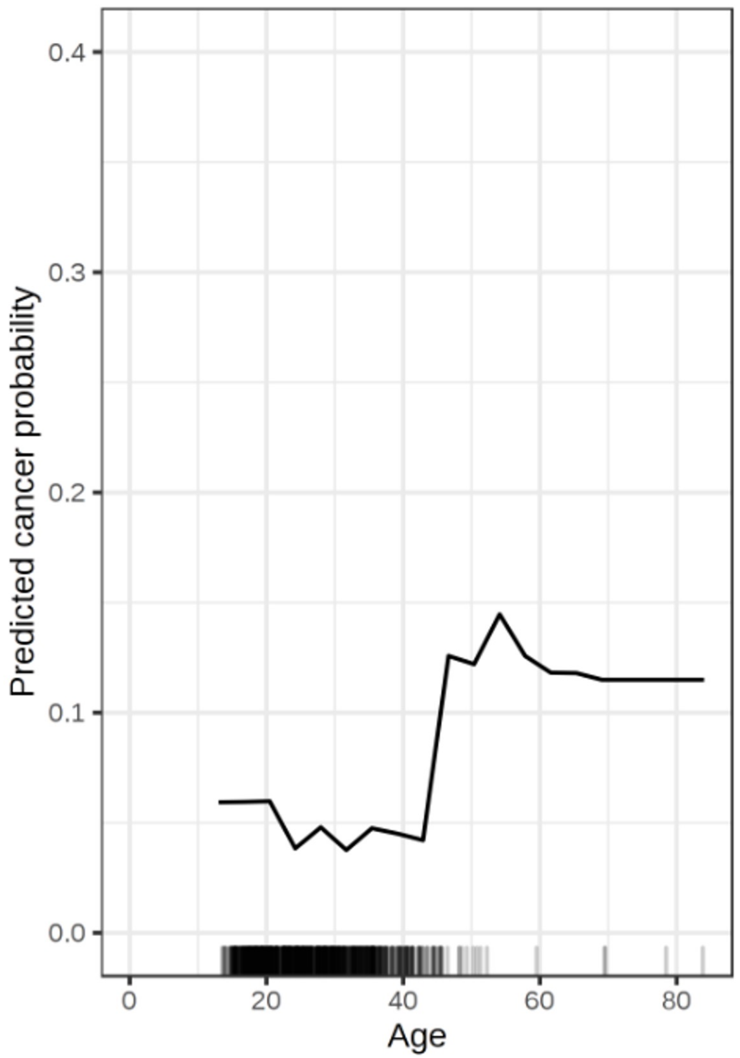
Feature Visualization, Model-agnostic - PDPs

Partial Dependence Plots

<u>Example</u> (single feature)

(from [Molnar]())

Prediction (“*cancer probability*”) as a function of the selected feature (“*Age*”)



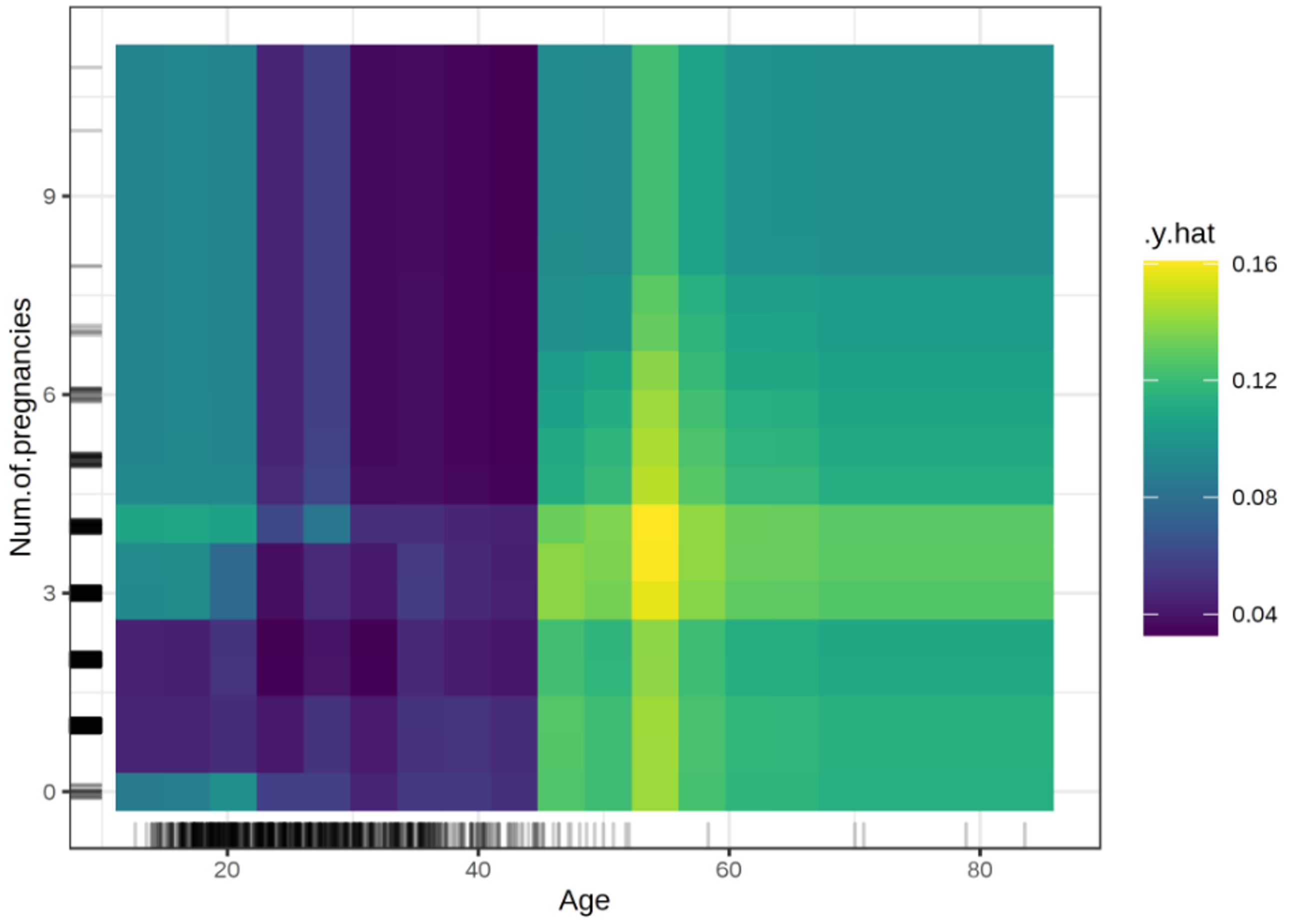
Feature Visualization, Model-agnostic - PDPs

Partial Dependence Plots

Example (pair of features)

(from Molnar)

Prediction ("*cancer probability*") as a function of the selected features ("*Age*" and "*Number of pregnancies*")



Feature Visualization, Model-agnostic - PDPs

Partial Dependence Plots

Disadvantages:

- assumption of independent features, as in Permutation Importance (strong assumption!)
- Partial Dependence Plots show only average effects

Feature Visualization, Model-agnostic - PDPs (Optional)

Partial Dependence Plots

REMARK ⚠

Variant: Accumulated Local Effects (ALE) plots

Disadvantages:

- assumption of independent features, as in Permutation Importance (strong assumption!)
- Partial Dependence Plots show only average effects

Feature Visualization, Model-agnostic - PDPs (Optional)

Partial Dependence Plots

Disadvantages:

- assumption of independent features, as in Permutation Importance (strong assumption!)
- Partial Dependence Plots show only average effe

REMARK

Variant: [Accumulated Local Effects (ALE) plots]

REMARK

Variant: [Individual Conditional Expectation (ICE) plots]

Local, Model-agnostic – LIME

Local Surrogate Models: LIME (Local Interpretable Model-agnostic Explanations)

Idea: train a local interpretable surrogate model to explain individual predictions

Only for theoretic part of the exam!

- post-hoc
- model-agnostic
- local

Let's assume we have a black-box model (e.g. a Deep Neural Network) and a new (single) data point x

GOAL: get the corresponding prediction y_{pred} and an explanation

Local, Model-agnostic - LIME

Local Surrogate Models: LIME

The prediction y_{pred} can be obtained, as usual, by feeding the black-box model with the input x and looking at his output

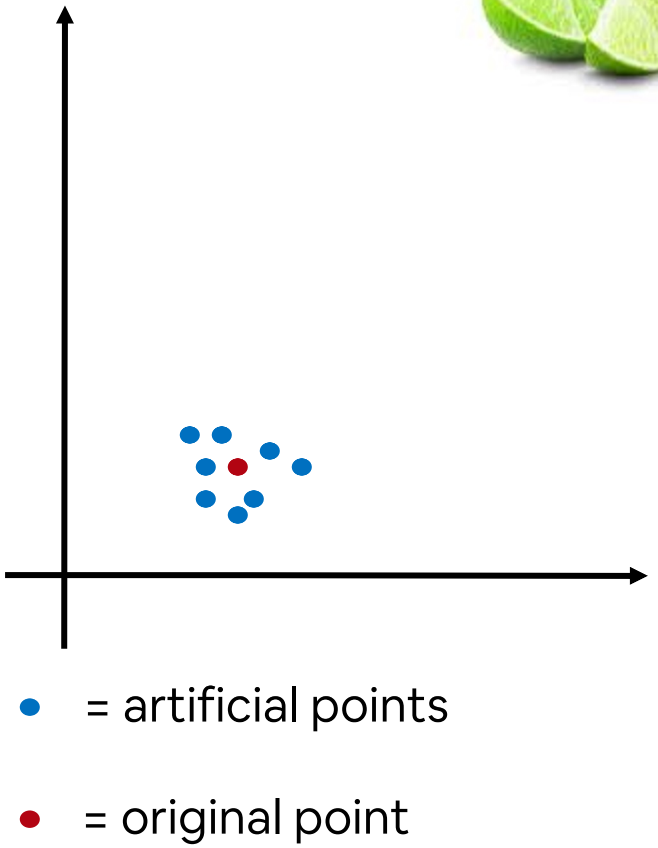
The explanation can be obtained through the LIME method, which is based on a local approximation of the black-box model by means of a simpler, interpretable model (so-called "*local surrogate model*")

Local, Model-agnostic - LIME

Local Surrogate Models: [LIME]

LIME procedure:

1. obtain new artificial data points x^a, x^b, ... by applying small perturbations to x
2. get the corresponding predictions y^a_{pred} , y^b_{pred} , ... made by the black-box model
3. train an interpretable model (say, a Decision Tree) in supervised settings on pairs (x^a, y^a_{pred}), (x^b, y^b_{pred}), ... (in other words, we are asking the interpretable model to learn the predictions made by the black-box model); each artificial point is weighted according to its proximity to the original point
4. exploit the interpretable nature of the local surrogate model (the Decision Tree, in this example) to see “what happens” in a neighborhood of the original point x



Local, Model-agnostic - LIME

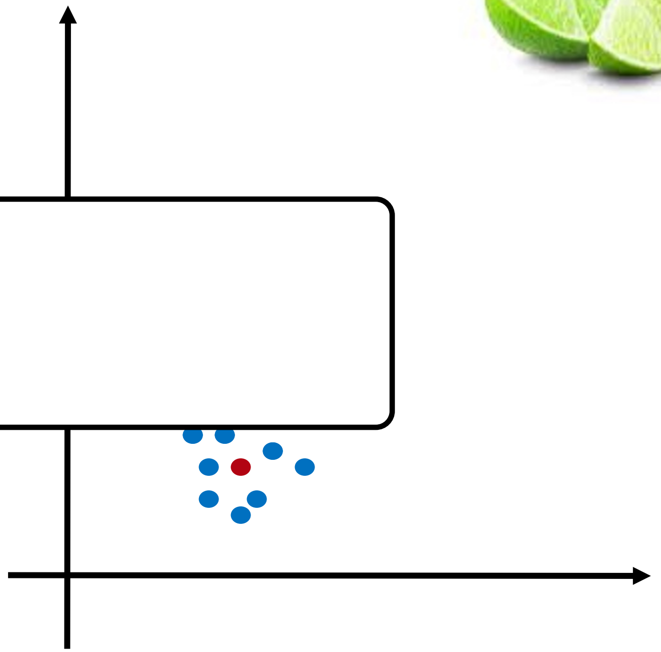
Local Surrogate Models: [LIME]

LIME procedu

REMARK ⚠

Challenge: how to properly define the neighborhood

1. obtain ne[…] perturbati[…]
2. get the c[…] by the bla[…]
3. train an interpretable model (say, a Decision Tree) in supervised settings on pairs (x^a, y^a_{pred}), (x^b, y^b_{pred}), … (in other words, we are asking the interpretable model to learn the predictions made by the black-box model); each artificial point is weighted according to its proximity to the original point
4. exploit the interpretable nature of the local surrogate model (the Decision Tree, in this example) to see "what happens" in a neighborhood of the original point x

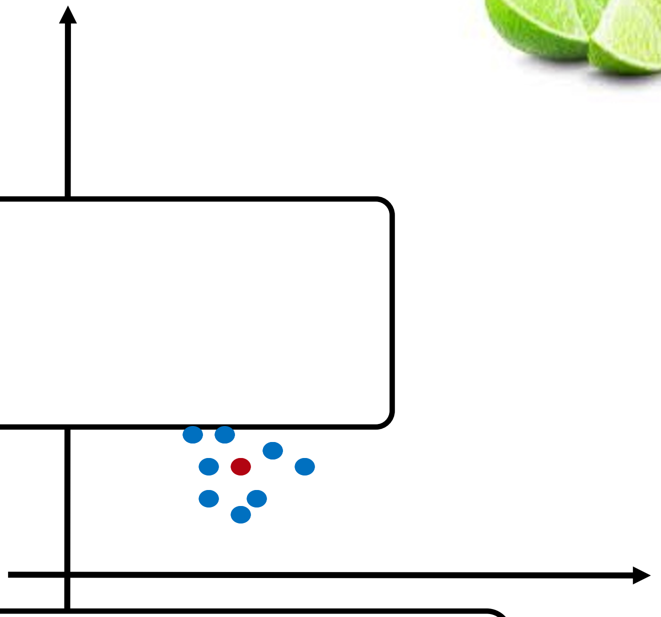


● = artificial points

● = original point

Local, Model-agnostic - LIME

Local Surrogate Models: LIME



LIME procedu

1. obtain new perturbati
2. get the co by the bla
3. train an interpretable model (say, a Decision Tree) in superv x^a, y^a_{pred}), (x^b, y^b_{pred}), … (in other w e interpretable model to learn t each a to the
4. exploit model happe

REMARK

Challenge: how to properly define the neighborhood

REMARK

Problem (Optional): Alvarez-Melis et al. have shown that LIME explanations are not always stable (very close points may have very different explanations)

SHAP (Optional)

SHapley Additive exPlanations

- Model-agnostic
- Post-hoc
- Based on the concept of Shapley value from cooperative game theory
- Can be used at both global and local scale

IDEA: the prediction produced by a ML model can be explained by treating it as the "payout" that has to be distributed across the features, which act as "players" in a coalition

Lundberg, S., & Lee, S. I. (2017). A unified approach to interpreting model predictions. *arXiv preprint arXiv:1705.07874*.

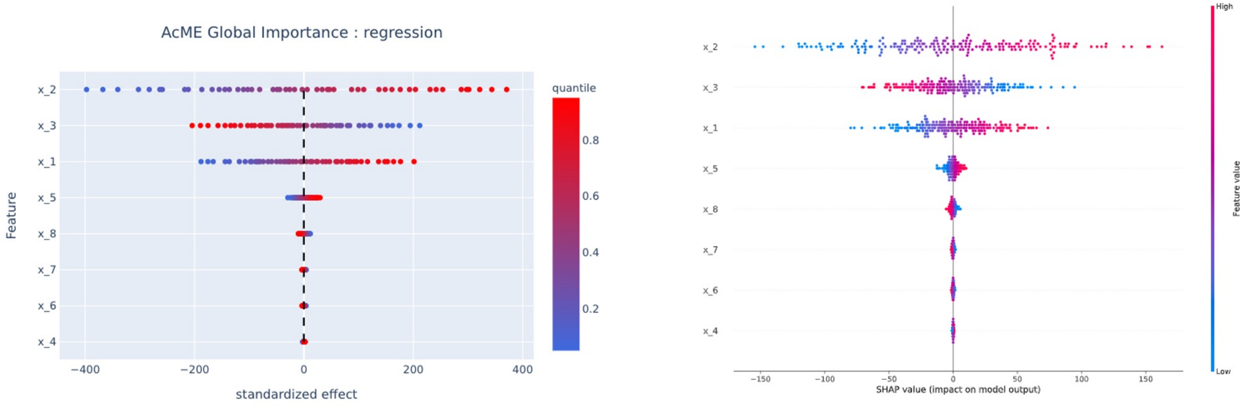
Our contribution: AcME (Optional)

ARTIFICIAL INTELLIGENCE, MACHINE LEARNING AND CONTROL RESEARCH GROUP

Accelerated Model Explanations (AcME)

- Loosely inspired by SHAP (but does not compute Shapley values!)
- Focused on the minimization of the computational cost
- Simplified visualization (human-centered approach)
- Tested on tabular data (for now)

https://www.sciencedirect.com/science/article/abs/pii/S0957417422021339



(a) AcME

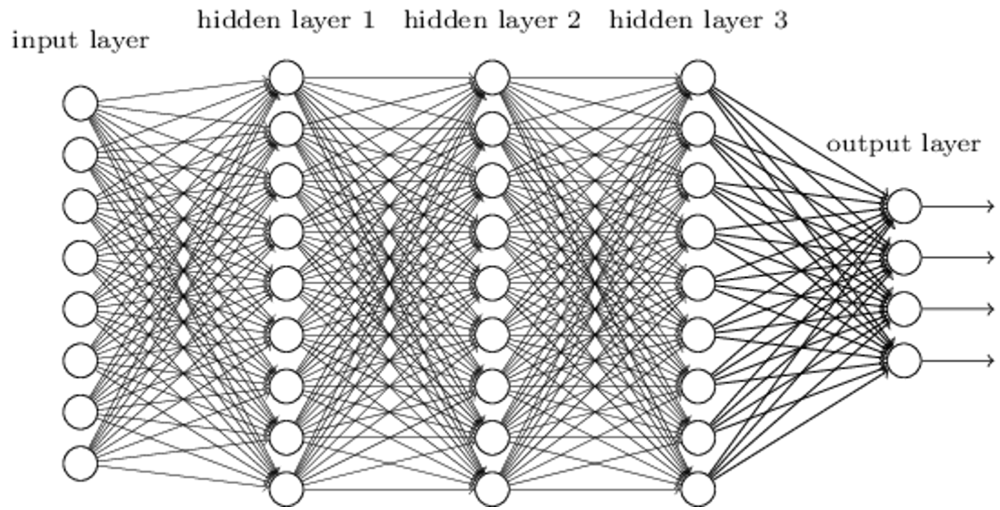
(b) KernelSHAP

	Number of samples	Elapsed Time (in seconds)
AcME	complete	0.36
KernelSHAP	5	357.23
KernelSHAP	10	425.61
KernelSHAP	20	875.85
KernelSHAP	100	1855.65

Table 2: [Boston Housing Dataset] Elapsed time for SHAP with different dataset sampled.

Model-specific methods for DNNs

Deep Neural Networks (DNNs) are arguably the hardest ML models to be interpreted by human beings



Despite their amazing performance on a wide variety of applications (e.g. Computer Vision, Natural Language Processing, ...), interpretation of DNNs and produced outputs is still an open research problem

In this lecture we just give a brief overview of the research works in this field and refer the curious readers to the work of Gilpin et al. for further details and analyses

Model-specific methods for DNNs

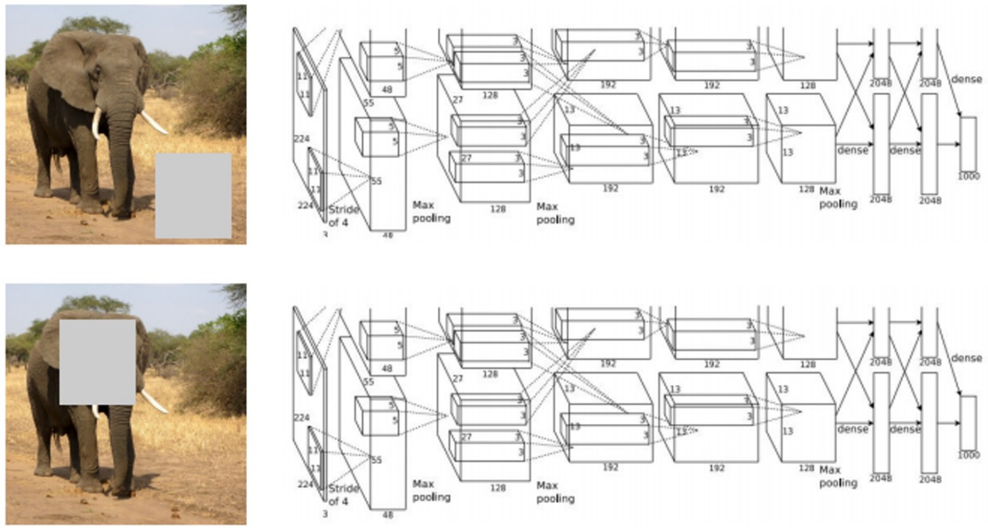
As described in Gilpin et al., interpretability methods for DNNs can be roughly divided into three main categories:

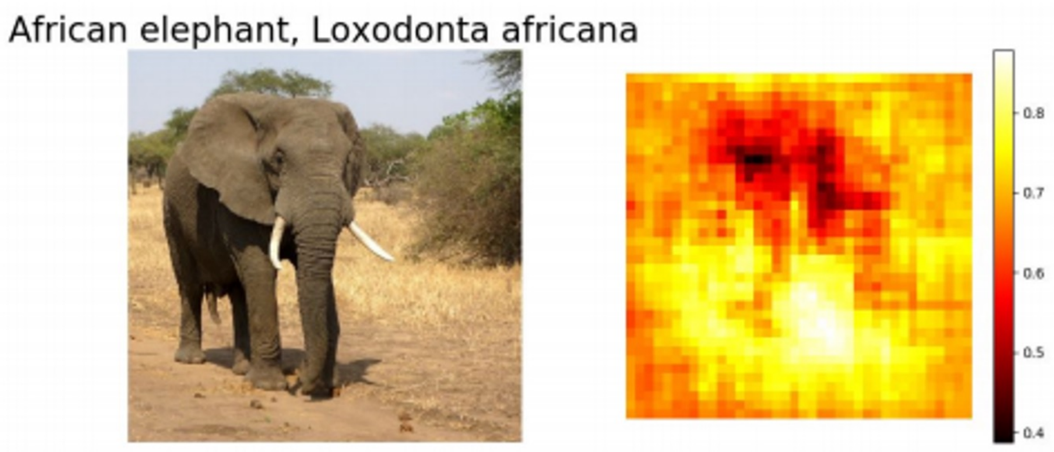
- methods focused on the explanation of the processing of the data by a DNNs
- methods focused on the explanation of the representations generated within the DNNs
- methods focused on the design of architectures that facilitate interpretations of the network's behavior

Model-specific methods for DNNs

Explanation of the processing

Saliency maps:





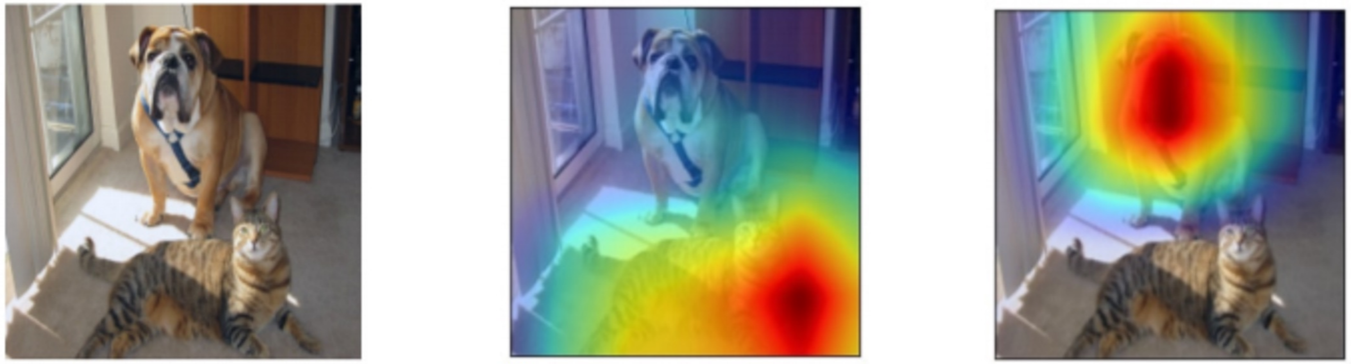
Zeiler and Fergus, “Visualizing and Understanding Convolutional Networks”, ECCV 2014

Model-specific methods for DNNs (Optional)

Explanation of the processing

Some examples:

- surrogate models specifically tailored for DNNs (such as DeepRED, ANN-DT)
- saliency mapping (such as DeepLIFT, Grad-CAM)



(a) Original Image (f) ResNet Grad-CAM 'Cat' (l)ResNet Grad-CAM 'Dog'

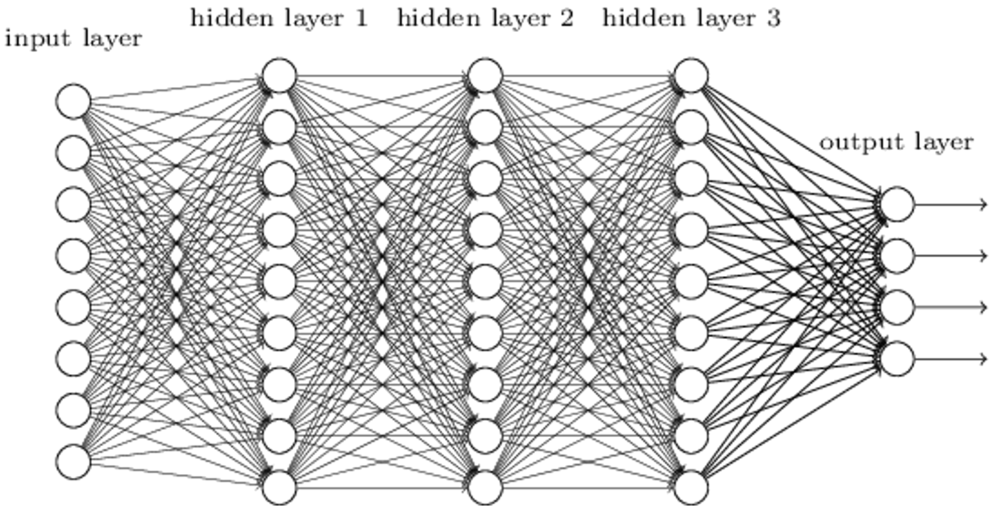
From "*Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization*", Selvaraju et al.

Model-specific methods for DNNs (Optional)

Explanation of the representations

Some examples:

- role of layers - example Razavian et al.
- role of individual units, both units or filters (like in CNN) - example Network dissection)
- role of other representation vectors - example Concept Activation Vectors

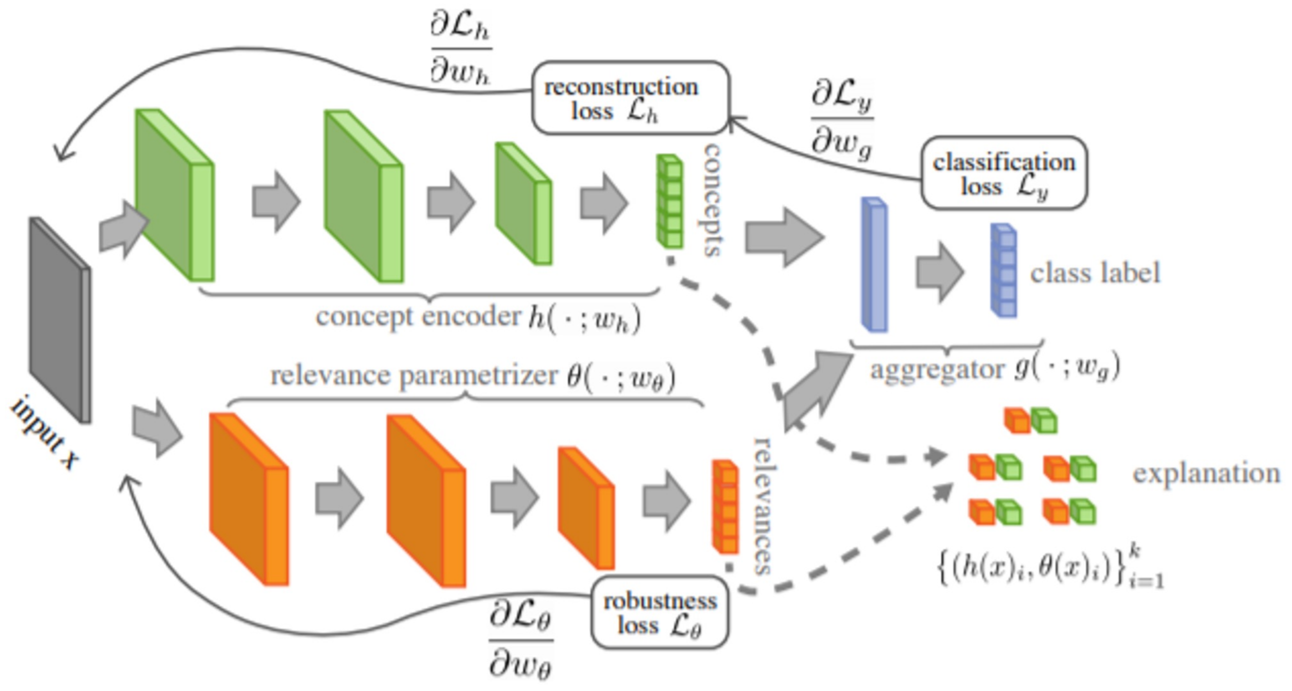


Model-specific methods for DNNs (Optional)

Explanation-producing systems

Some examples:

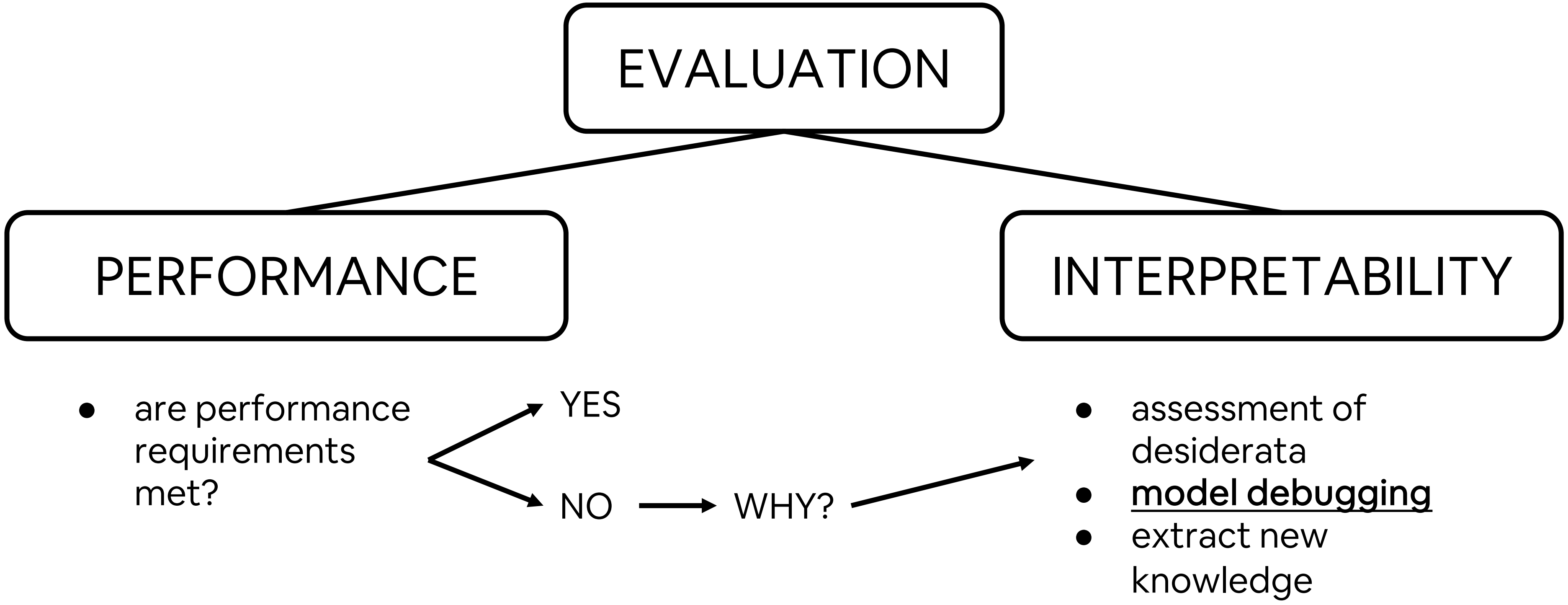
- attention networks (such as Xiao et al.)
- Self-Explaining Neural Networks (Alvarez-Melis et al.)



From "*Towards Robust Interpretability with Self-Explaining Neural Networks*", Alvarez-Melis et al.

Evaluation of model interpretability

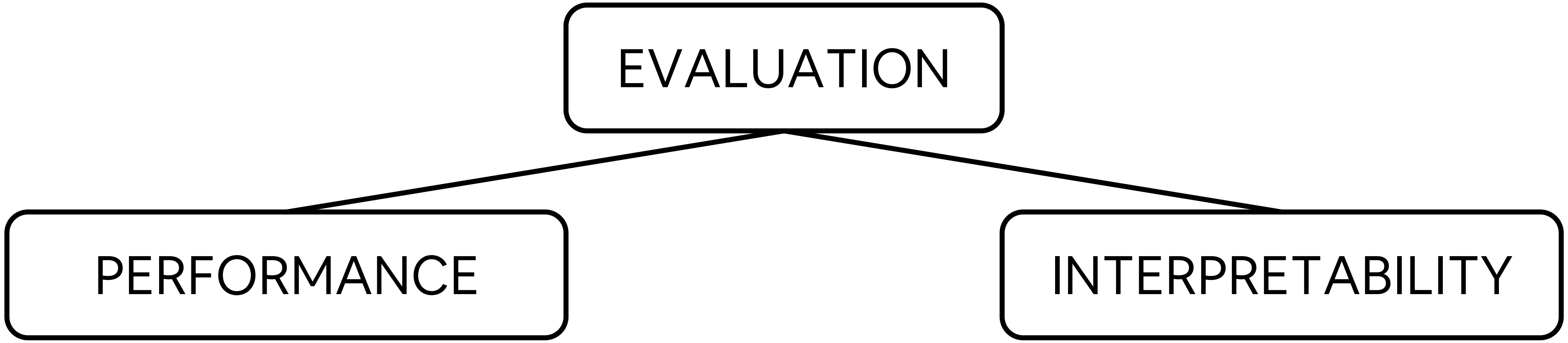
Just like the assessment of a ML model performance, the evaluation of interpretability is among the most delicate procedures in the overall pipeline



Evaluation of model interpretability

REMARK: High performances help interpretability: if the model is not accurate, insights and added knowledge may not be accurate as well!

Just like the assessment of a ML model performance, the evaluation of interpretability is among the most delicate procedures in the overall pipeline



- are performance requirements met?

- assessment of desiderata
- <u>model debugging</u>
- extract new knowledge

Evaluation of model interpretability

Evaluation of model performance

Luckily, we can rely on established metrics, which are easy and inexpensive to compute:

- Classification problems
  - Overall classification accuracy (balanced datasets)
  - Per-class classification accuracy (unbalanced datasets)
  - Precision, Recall and F1-score (binary classification)

- Regression problems
  - Mean Squared Error (MSE)
  - Mean Absolute Error (MAE)
  - R2 score

Evaluation of model interpretability

Evaluation of model interpretability

No well-established and recognized metrics

First concrete effort to organize ideas about interpretability evaluation procedures can be found in "*Towards a rigorous science of interpretable machine learning*" (Doshi-Velez et al.)

Three levels:

- Application level evaluation
- Human level evaluation
- Functional level evaluation

Evaluation of model interpretability

Evaluation of model interpretability – Application level

Real humans: model's interpretability is assessed through experiments involving domain experts

Real task: model's interpretability is assessed w.r.t. the task the model is supposed to solve

Example: ML model trained to detect fractures from X-ray images

(adapted from Molnar)

→ Radiologists are asked to assess the quality of explanations

Evaluation of model interpretability

REMARK

This is the more direct way to evaluate interpretability because it reflects exactly deployment conditions

Evalua

Real hu ... eriments involvi

Real task: model's interpretability is assessed w.r.t. the task the model is supposed to solve

Example: ML model trained to detect fractures from X-ray images

(adapted from Molnar)

→ Radiologists are asked to assess the quality of explanations

Evaluation of model interpretability

REMARK

This is the more direct way to evaluate interpretability because it reflects exactly deployment conditions

Evalua

Real hu ... eriments involvi

Real ta ... pretability is assessed w.r.t. the task the model is suppo

REMARK

Problem: it's difficult to recruit radiologists… and their expertise may be costly!

Examp ... ages

(adapted from Molnar)

→ Radiologists are asked to assess the quality of explanations

Evaluation of model interpretability

Evaluation of model interpretability – Human level

Real humans: model's interpretability is assessed through experiments involving non-experts

Simplified task: model's interpretability is assessed w.r.t. a simplified version of the task the model is supposed to solve

Example: ML model trained to detect fractures from X-ray images

(adapted from [Molnar])

⟶ Non-radiologists are asked to rank different types of explanations

Evaluation of model interpretability

REMARK ⚠

Evalu

Real h... ents
involvi...

Simpli... ed version of
the ta...

Advantages:

- recruitment process is way easier
- experiments are cheaper

→ we can afford a larger number of experiments, so that results on the interpretability evaluation are statistically more significant

Example: ML model trained to detect fractures from X-ray images
(adapted from [Molnar]())

→ Non-radiologists are asked to rank different types of explanations

Evaluation of model interpretability

Evaluation of model interpretability – Functional level

No humans: model's interpretability is assessed without involving humans

Proxy task: model's interpretability is assessed by relying on a quantifiable measure recognized to be related to interpretability (e.g. sparsity, depth in tree-based models,...)

Example: if a Decision Tree is being used, we assess its interpretability through the analysis of its depth (shallow trees are more interpretable than deep ones)

Evaluation of model interpretability

REMARK

Advantage: the evaluation is straightforward (since it doesn't require human experiments) and less subjective

Evalua

No hu ... ing humans

Proxy task: model's interpretability is assessed by relying on a quantifiable measure recognized to be related to interpretability (e.g. sparsity, depth in tree-based models,...)

Example: if a Decision Tree is being used, we assess its interpretability through the analysis of its depth (shallow trees are more interpretable than deep ones)

Evaluation of model interpretability

REMARK

Evalua

Advantage: the evaluation is straightforward (since it doesn't require human experiments) and less subjective

No hu ing humans

Proxy task: model's interpretability is assessed by relying on a quantifiable measure recognized to be related to interpretability (e.g. sparsity, depth in tree-based

REMARK

Challenge: which proxies to use! Not all ML models offer useful quantities for the purpose of interpretability evaluation

Example: if a D
through the a
than deep one

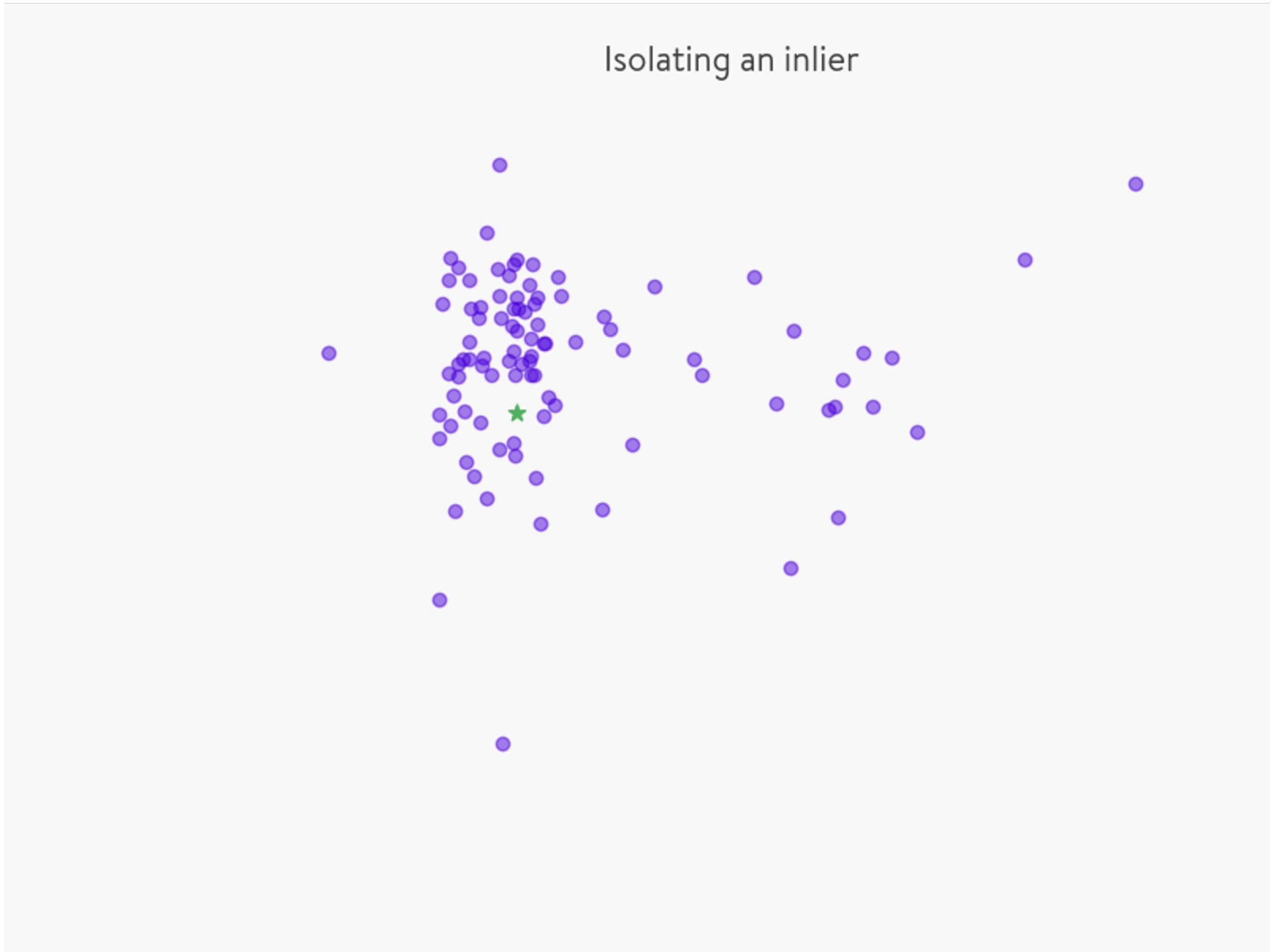
An example of Proxy Task Choice

Anomaly detection is an unsupervised task that aims at identifying data points that are 'different' from the majority

There are tree-based approaches for anomaly/outlier detection that are quite powerful and widely adopted

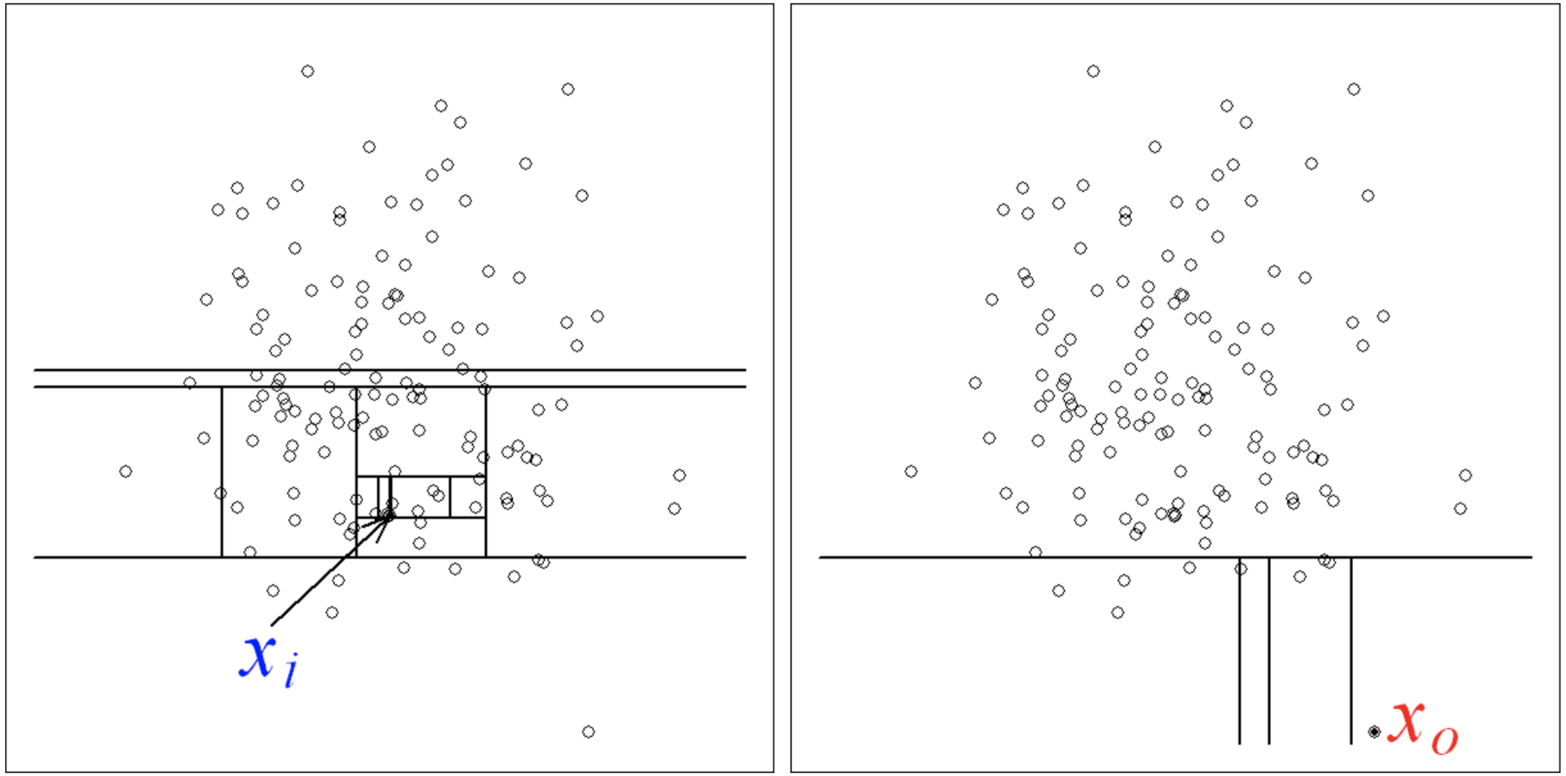
An example of Proxy Task Choice

Isolation Forest is based on recursive partitioning



An example of Proxy Task Choice

Isolation Forest is based on recursive partitioning



An example of Proxy Task Choice

We derived a method to make Isolation Forest interpretable (DIFFI): similarly to MDI in Random Forest, but based on unsupervised principles

DIFFI provides both global and local feature rankings

M. Carletti, M. Terzi, G.A. Susto.
Interpretable Anomaly Detection with DIFFI: Depth-based Feature Importance for the Isolation Forest
https://arxiv.org/abs/2007.11117

An example of Proxy Task Choice

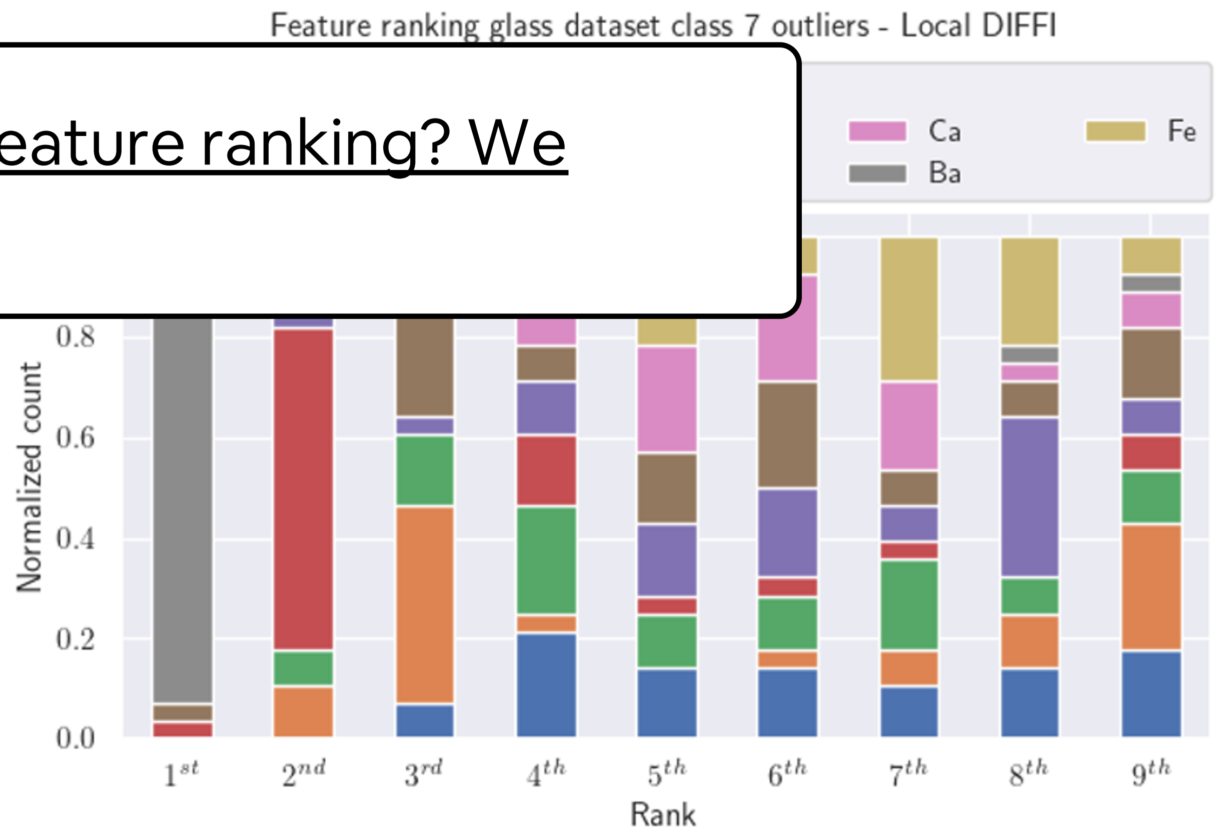
ARTIFICIAL INTELLIGENCE, MACHINE LEARNING AND CONTROL RESEARCH GROUP

We derived a method to make
Isolati
(DIFFI)
Rando
unsup

How do we evaluate the feature ranking? We don’t have a ground truth

DIFFI provides both global and local feature rankings

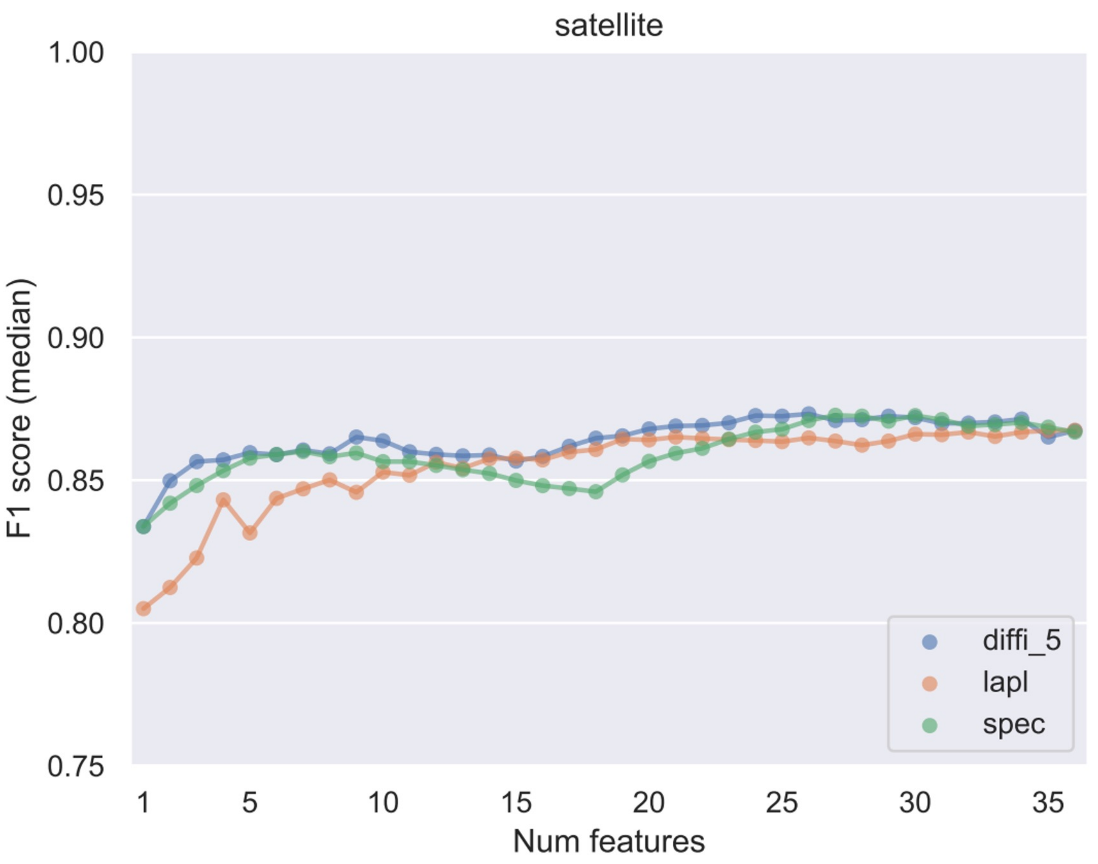
M. Carletti, M. Terzi, G.A. Susto.
Interpretable Anomaly Detection with DIFFI: Depth-based Feature Importance for the Isolation Forest
https://arxiv.org/abs/2007.11117

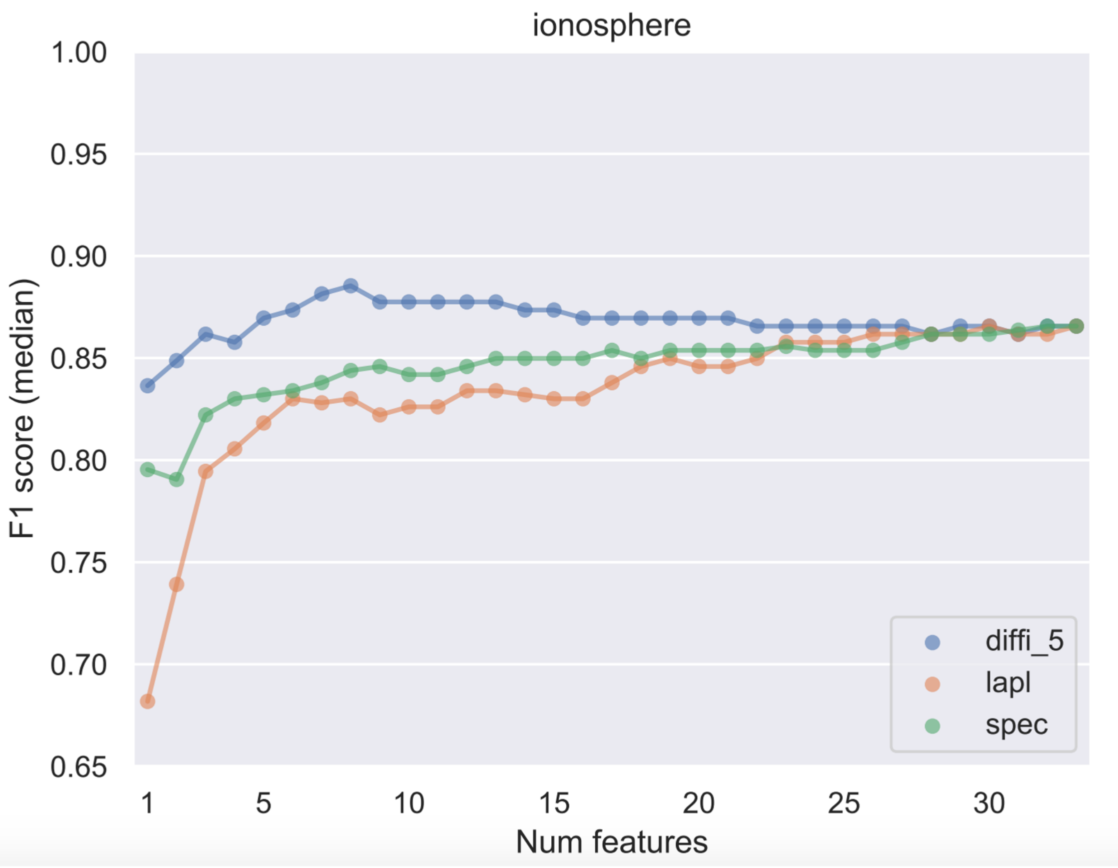


An example of Proxy Task Choice

ARTIFICIAL INTELLIGENCE, MACHINE LEARNING AND CONTROL RESEARCH GROUP

Proxy task: feature selection





Evaluation of model interpretability

<u>Open research problem:</u> make the three levels of evaluation inform each other

Questions to be addressed:

- [functional $\longrightarrow$ application] what proxies for what applications?
- [application $\longrightarrow$ human] which are the factors that should be considered for the simplified task, in order to maintain the essence of the original one?
- [human $\longrightarrow$ functional] which are the important factors to consider for proxies, in order to provide good explanations?

Want to try things on your own? Python-based answer...

- LASSO https://scikit-learn.org/stable/modules/generated/sklearn.linear_model.Lasso.html
- MDI for Random Forest https://scikit-learn.org/stable/auto_examples/ensemble/plot_forest_importances.html
- Permutation Importance https://scikit-learn.org/stable/modules/permutation_importance.html
- LIME https://github.com/marcotcr/lime
- SHAP https://github.com/slundberg/shap
- Available notebook on the class page!

Bonus:

- DIFFI https://github.com/mattiacarletti/DIFFI
- ACME https://github.com/dandolodavid/ACME

Università degli Studi di Padova

Machine Learning 2024/2025

Artificial Intelligence, Machine Learning and Control Research Group

Thank you!

Gian Antonio Susto