DEPARTMENT OF KNOWLEDGE TECHNOLOGIES
Jožef Stefan Institute

EXPLAINABLE MACHINE LEARNING

Sašo Džeroski
Head of Department of knowledge technologies
Jožef Stefan Institute, Ljubljana, Slovenia

DEPARTMENT OF KNOWLEDGE TECHNOLOGIES
Jožef Stefan Institute

THE NEED FOR EXPLANATIONS IN MACHINE LEARNING

Complex machine learning models often operate as "black boxes" – we can see what goes in and what comes out,
but not how decisions are made inside.

Models in ML may use hundreds or thousands of features. Their internal calculations are complex and non-linear. Understanding which inputs drive outcomes is difficult.

For students, scientists & researchers,

understanding the "why" behind ML/AI predictions is crucial for:

- Building trust in research findings
- Ensuring scientific accountability
- Applying AI responsibly in practice

DEPARTMENT OF KNOWLEDGE TECHNOLOGIES
Jožef Stefan Institute

THE PROBLEM OF BLACK-BOX MODELS

A) ***Deep neural networks***

may have millions of parameters

B) ***Random forests***

can contain hundreds of decision trees

C) ***Complex feature interactions***

defy simple explanations

D) ***Lack of transparency***

limits trust and adoption

E) ***Explanation is at the very foundation of science:***

Lack thereof limits use of ML in science, incl. physics

EXPLAINABILITY IN MACHINE LEARNING (lecture outline)

Machine learning tasks (of predictive modelling)

- *Classification & regression*
- *Multi-target prediction (MT classification; MT regression)*

Interpretable models

- *Trees & rules*
- *Relational trees & rules*
- *Linear models & other equations*

Explaining (black-box) models and their predictions

- *Internal feature importance estimation (e.g., in forests)*
- *External feature importance estimation (e.g., permutation)*
- *Explaining predictions: SHAP & LIME*

INTERPRETABLE MODELS

*An **interpretable model** is one whose **internal mechanics** (structure and parameters) are **inherently understandable** to humans **without** the need for **post-hoc analysis***

Key characteristics of interpretable models

- ***Transparency:** You can understand how the model arrives at predictions by examining its components.*
- ***Simplicity:** The structure is relatively simple (e.g., small decision trees, sparse linear models).*
- ***Self-contained Explanation:** No additional tools or methods are required to explain the output.*

Examples of interpretable models: Decision trees, classification rules, linear models, naïve Bayes, …

EXPLAINABLE MODELS

*An **explainable model** is one where the output or **behavior can be explained post-hoc**, even if the model itself is not inherently understandable*

Key characteristics

- ***Post hoc explanation:*** *Explanations are derived after the model is trained*
- ***Model complexity:*** *Usually refers to black-box models (e.g., neural networks, ensemble methods)*
- ***Explanation interface:*** *Requires an external method to interpret predictions*

Example explanation techniques: Feature importance plots, LIME (Local Interpretable Model-agnostic Explanations); SHAP (SHapley Additive exPlanations); Saliency maps (in comp. vis.)

MACHINE LEARNING TASKS: DIMENSIONS

Different types of (structured) outputs

- Single target classification/ regression
- MT/ML Classification, MT Regression

Different degrees of supervision

- Fully supervised
- Semi-supervised
- Unsupervised

Batch vs. streaming

	Descriptive space				Target space		
…	…				…		
Example n+5	6	TRUE	0.46	0.61	0.66	0.00	7.91
Example n+1	4	FALSE	0.08	0.07	0.10	1.69	7.57
Example n+2	6	FALSE	0.08	0.07	0.08	0.77	8.86
Example n+3	8	TRUE	0.00	1.00	0.11	3.51	2.50
Example n+4	6	TRUE	0.00	0.00	0.43	2.10	8.09
…	…				…	…	…

SINGLE-TARGET PREDICTION (classification, regr.)

	Descriptive space				Target space
Example 1	1	TRUE	0.49	0.69	Yes
Example 2	2	FALSE	0.08	0.07	Yes
Example 3	1	FALSE	0.08	0.07	No
Example 4	2	TRUE	0.49	0.69	Yes
Example 5	3	TRUE	0.49	0.69	No
Example 6	4	FALSE	0.08	0.07	Yes
…	…				…

	Descriptive space				Target space
Example 1	1	TRUE	0.49	0.69	0.84
Example 2	2	FALSE	0.08	0.07	0.75
Example 3	1	FALSE	0.08	0.07	0.11
Example 4	2	TRUE	0.49	0.69	0.52
Example 5	3	TRUE	0.49	0.69	0.35
Example 6	4	FALSE	0.08	0.07	0.78
…	…				…

AN EXAMPLE FROM MEDICINE

Classical problem: Medical diagnosis

- Predictive modeling/classification

An example: Neurodegenerative diseases

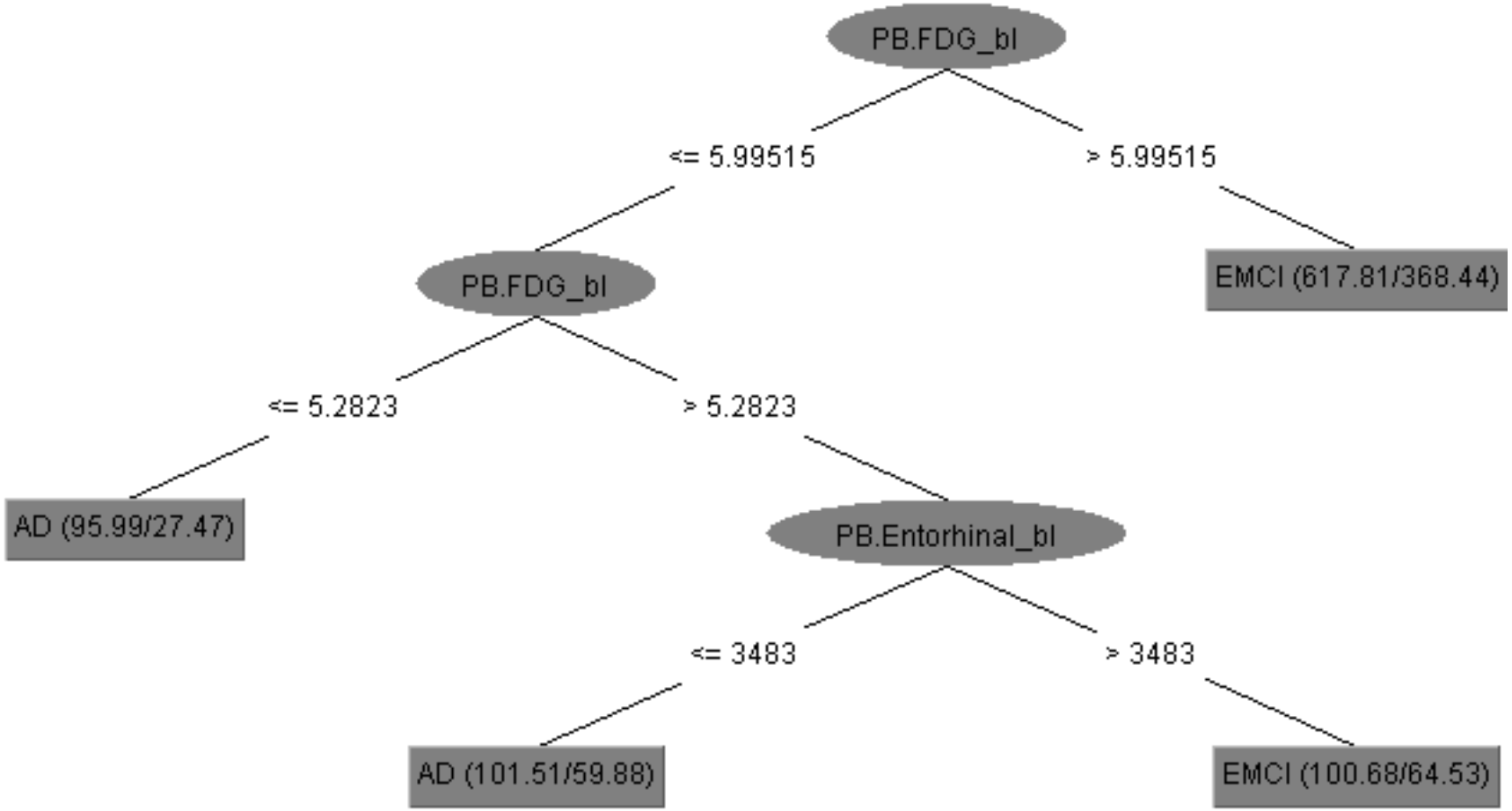
Target variable: Diagnosis; Possible values:

- CN - Cognitively Normal (0)
- SMC - Significant Memory Concern
- EMCI - Early Mild Cognitive Impairment
- LMCI - Late Mild Cognitive Impairment
- AD - Alzheimer‘s Disease (4)

Descriptive/input variables

1. APOE4 – Genetic variations of APOE4 related gene
2. FDG – Positron emission tomography (PET) imaging results with [^{18}F]fluorodeoxyglucose
3. AV45 – Positron emission tomography (PET) imaging results with [^{18}F]-labeled amyloid imaging agent AV45
4. Ventricles
5. Hippocampus
6. WholeBrain
7. Entorhinal
8. Fusiform – Fusiform gyrus
9. MidTemp – Middle Temporal Gyrus
10. ICV – Intracerebral volume [Volumetric data 4-10]

EXAMPLE DECISION TREE FOR DIAGNOSIS



MULTI-TARGET PREDICTION TASKS (MT classif., MT regr.)

	Descriptive space				Target space		
Example 1	1	TRUE	0.49	0.69	Yes	Blue	Rain
Example 2	2	FALSE	0.08	0.07	Yes	Green	Sun
Example 3	1	FALSE	0.08	0.07	Yes	Blue	Cloudy
Example 4	2	TRUE	0.49	0.69	Yes	Green	Sun
Example 5	3	TRUE	0.49	0.69	No	Blue	Sun
Example 6	4	FALSE	0.08	0.07	Yes	Red	Cloudy
…	…				…	…	…

	Descriptive space				Target space		
Example 1	1	TRUE	0.49	0.69	0.68	0.60	3.91
Example 2	2	FALSE	0.08	0.07	0.56	0.99	7.59
Example 3	1	FALSE	0.08	0.07	0.10	1.69	7.57
Example 4	2	TRUE	0.49	0.69	0.08	0.77	8.86
Example 5	3	TRUE	0.49	0.69	0.11	3.51	2.50
Example 6	4	FALSE	0.08	0.07	0.43	2.10	8.09
…	…				…	…	…

MULTI-TARGET PREDICTION: Clinical scores for Alzheimer's

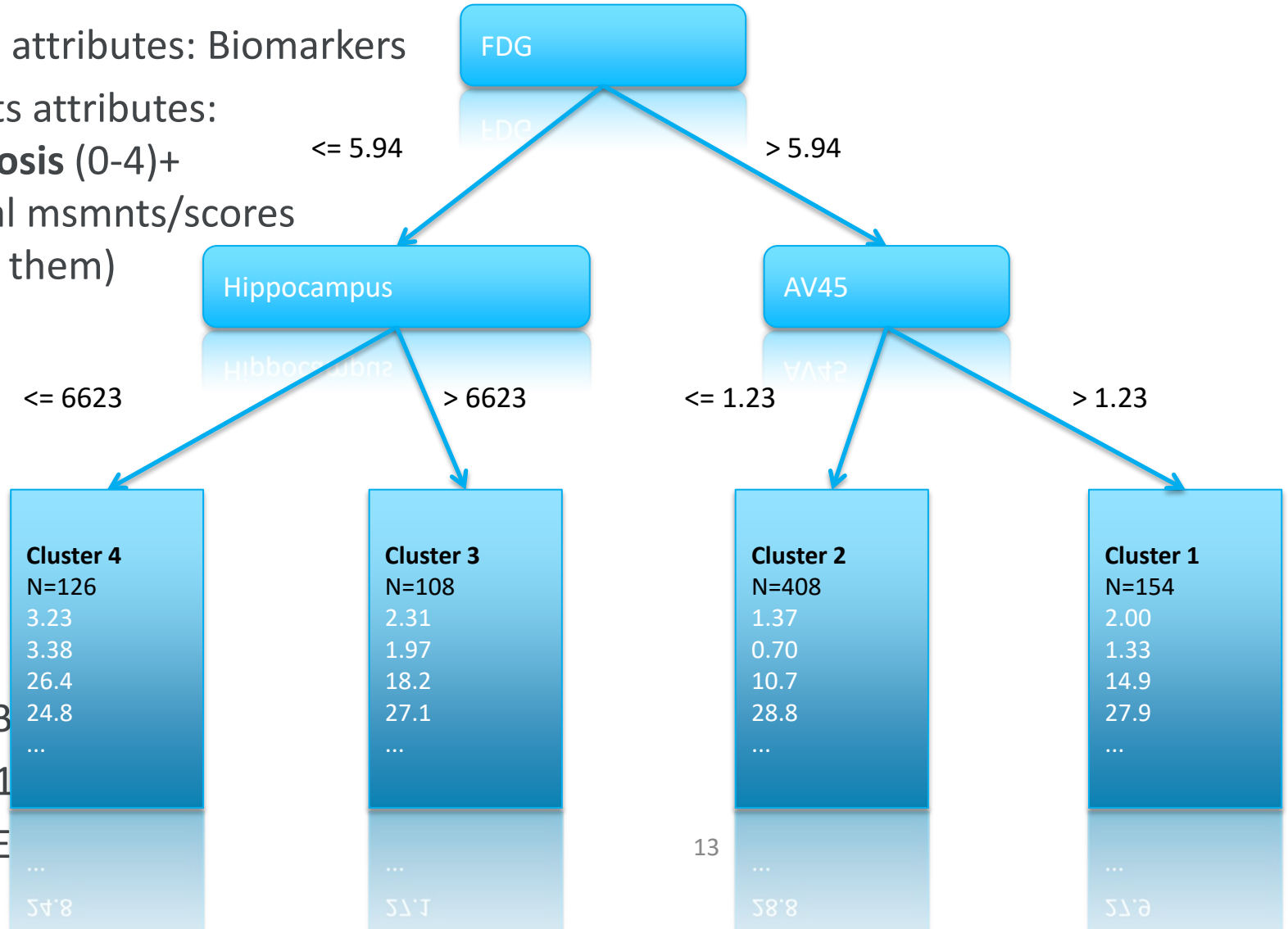
1. CDRSB – Clinical Dementia Rating Sum of Boxes
2. ADAS13 – AD assessment scale
3. MMSE – Mini Mental State Examination
4. RAVLT (immediate, learning, forgetting, perc. forgetting) – Rey Auditory Verbal Learning Test (4 features)
5. FAQ – Functional Assessment Questionnaire
6. MOCA – Montreal Cognitive Assessment
7. Ecog**Pt** (Memory, Language,Visuospatial Abilities,Planning,Organization,Divided Attention, Total score) – Everyday cognition questionnaire – filled in by patient
   (7 features)
8. Ecog**SP** (Memory, Language,Visuospatial Abilities,Planning,Organization,Divided Attention, Total score) – Everyday cognition questionnaire – filled in by study parter (7 features)

Descriptive/input variables (same as for diagnosis)

1. APOE4 – Genetic variations of APOE4 related gene
2. FDG – Positron emission tomography (PET) imaging results with [^{18}F]fluorodeoxyglucose
3. AV45 – Positron emission tomography (PET) imaging results with [^{18}F]-labeled amyloid imaging agent AV45
4. Ventricles
5. Hippocampus
6. WholeBrain
7. Entorhinal
8. Fusiform – Fusiform gyrus
9. MidTemp – Middle Temporal Gyrus
10. ICV – Intracerebral volume [Volumetric data 4-10]

MULTI-TARGET REGRESSION TREE RELATING BIOMARKERS TO CLINICAL SCORES

- Descr. attributes: Biomarkers
- Targets attributes: **diagnosis** (0-4)+ clinical msmnts/scores (23 of them)
  - DX
  - CDRSB
  - ADAS1
  - MMSE
  - ...



13

SEMI-SUPERVISED LEARNING: USE BOTH LABELED AND UNLABELED DATA

	Descriptive space				Target space
Example 1	1	TRUE	0.49	0.69	Yes
Example 2	2	FALSE	0.08	0.07	**?**
Example 3	1	FALSE	0.08	0.07	**?**
Example 4	2	TRUE	0.49	0.69	Yes
Example 5	3	TRUE	0.49	0.69	No
Example 6	4	FALSE	0.08	0.07	**?**
…	…				…

	Descriptive space				Target space
Example 1	1	TRUE	0.49	0.69	0.84
Example 2	2	FALSE	0.08	0.07	**?**
Example 3	1	FALSE	0.08	0.07	0.11
Example 4	2	TRUE	0.49	0.69	**?**
Example 5	3	TRUE	0.49	0.69	**?**
Example 6	4	FALSE	0.08	0.07	0.78
…	…				…

SEMI-SUPERVISED MULTI-TARGET PREDICTION: LEARNING FROM PARTIALLY LABELED DATA

In single target SSL, we have present or missing labels

In multi-target SSL, we have

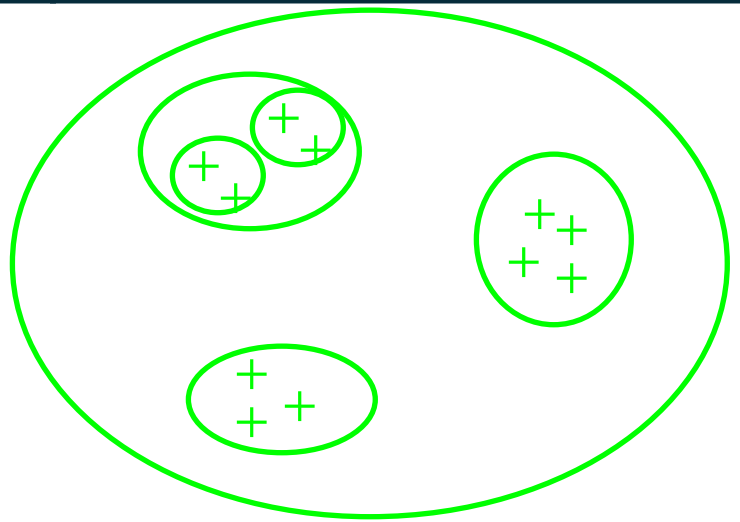
- Fully labeled examples (with complete labels)
- Fully unlabeled examples (all target values missing), and
- Partially labeled examples (some labels missing, some not)

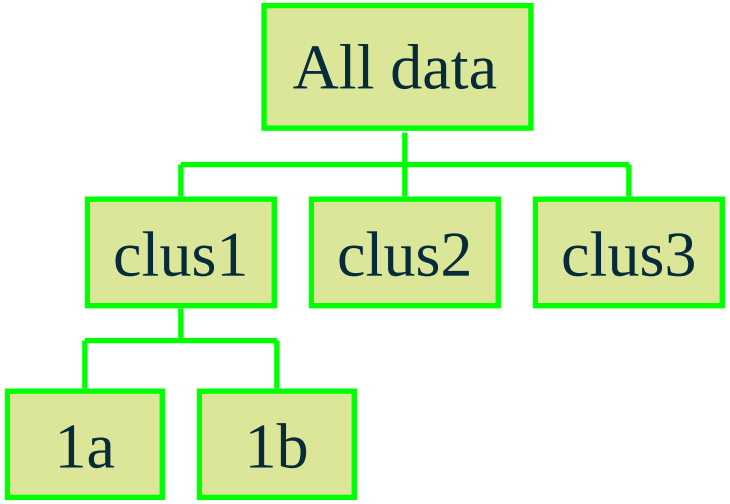
	Descriptive space				Target space		
Example 1	1	TRUE	0.49	0.69	**?**	0.60	3.91
Example 2	2	FALSE	0.08	0.07	0.56	0.99	7.59
Example 3	1	FALSE	0.08	0.07	**?**	**?**	**?**
Example 4	2	TRUE	0.49	0.69	0.08	0.77	8.86
Example 5	3	TRUE	0.49	0.69	0.11	**?**	**?**
Example 6	4	FALSE	0.08	0.07	0.43	2.10	8.09
…	…				…	…	…

UNSUPERVISED LEARNING: LEARNING FROM UNLABELED DATA

Data have no labels, need to be grouped into clusters of similar points

	Descriptive space				Target space		
Example 1	1	TRUE	0.49	0.69	?	0.60	3.91
Example 2	2	FALSE	0.08	0.07	0.56	0.99	7.59
Example 3	1	FALSE	0.08	0.07	?	?	?
Example 4	2	TRUE	0.49	0.69	0.08	0.77	8.86
Example 5	3	TRUE	0.49	0.69	0.11	?	?
Example 6	4	FALSE	0.08	0.07	0.43	2.10	8.09
…	…				…	…	…





PREDICTIVE CLUSTERING

Unifies predictive modelling clustering

Views trees as hierarchical clusterings

Builds such trees from data

With each cluster, predictive clustering provides
A description of the cluster
A prediction of the selected targets for that cluster

The output of Predictive Clustering can be viewed both as a clustering and as a predictive model

LEARNING TREES FOR MULTI-TARGET PREDICTION WITH PREDICTIVE CLUSTERING

To construct a tree T from a training set S:

If **the examples in S have low variance**,

construct a leaf labeled *target(prototype(S))*

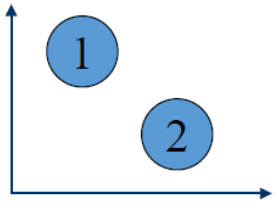
Otherwise:

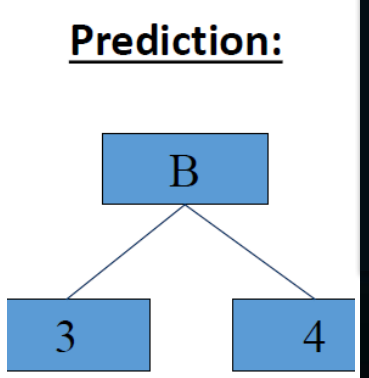
- Select the best attribute A with values v1, …, vn,

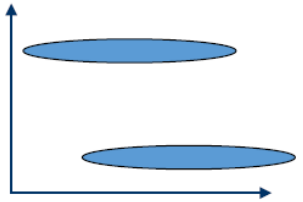
  which **reduces the most the variance** (*measured according to a given distance function d*)
- Partition S into S1, …, Sn according to A
- Recursively construct subtrees T1 to Tn for S1 to Sn
- Result: a tree with root A and subtrees T1, …, Tn

The variance is assessed across the multiple targets

LEARNING
PREDICTIVE CLUSTERING TREES

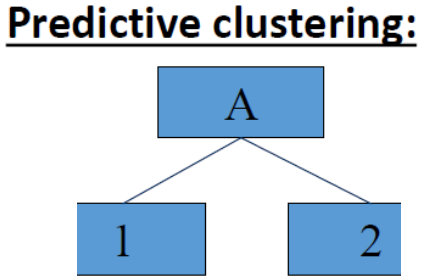




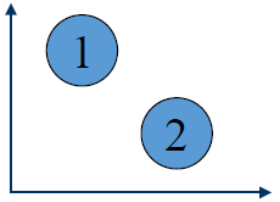


- Recursively partition data set into subsets (clusters) with low intra-cluster variance
  - Variance = avg. squared distance to prototype

$$ICV(S) = \sum_{y_j \in S} d(y_j, p(S))^2$$



- For the variance, the distance is measured
  - In standard clustering, along all dimensions
  - In prediction, along a single target dimension
  - In predictive clustering, along a structured target, e.g., several target dimensions



SELECTING THE BEST TEST IN A PCT

Select the test that maximizes variance reduction

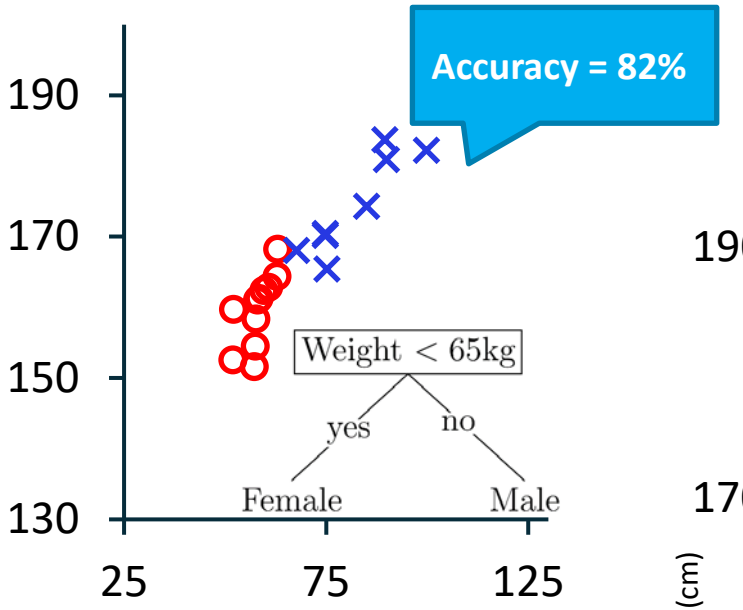
Calculated in line 4

procedure BestTest(E)

1: $(t^*, h^*, \mathcal{P}^*) = (none, 0, \emptyset)$
2: **for each** possible test t **do**
3: $\quad \mathcal{P} =$ partition induced by t on E
4: $\quad h = Var(E) - \sum_{E_i \in \mathcal{P}} \frac{|E_i|}{|E|} Var(E_i)$
5: $\quad$ **if** $(h > h^*) \wedge$ Acceptable$(t, \mathcal{P})$ **then**
6: $\quad\quad (t^*, h^*, \mathcal{P}^*) = (t, h, \mathcal{P})$
7: **return** $(t^*, h^*, \mathcal{P}^*)$

$$Var(E) = \sum_{i=1}^{T} Var(Y_i).$$

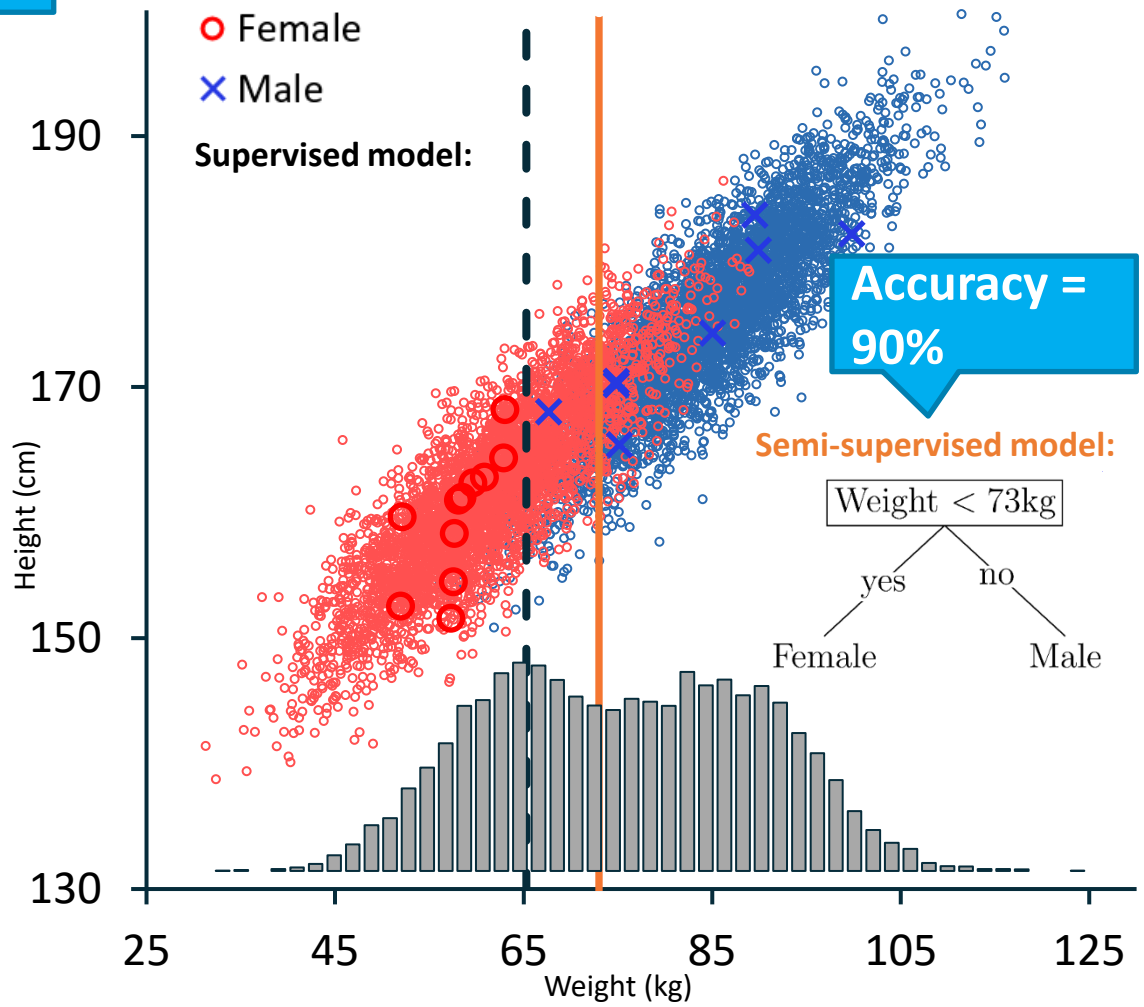
SEMI-SUPERVISED LEARNING: WHY USING UNLABELED DATA CAN HELP?



Task: Predict gender from person‘s weight and height

Data: 10000 entries of height, weight and gender

Assumptions about the labels with respect to the structure of unlabeled data



SEMI-SUPERVISED LEARNING WITH PCTs

New definition of variance that includes both targets and attributes, e.g., for MTR

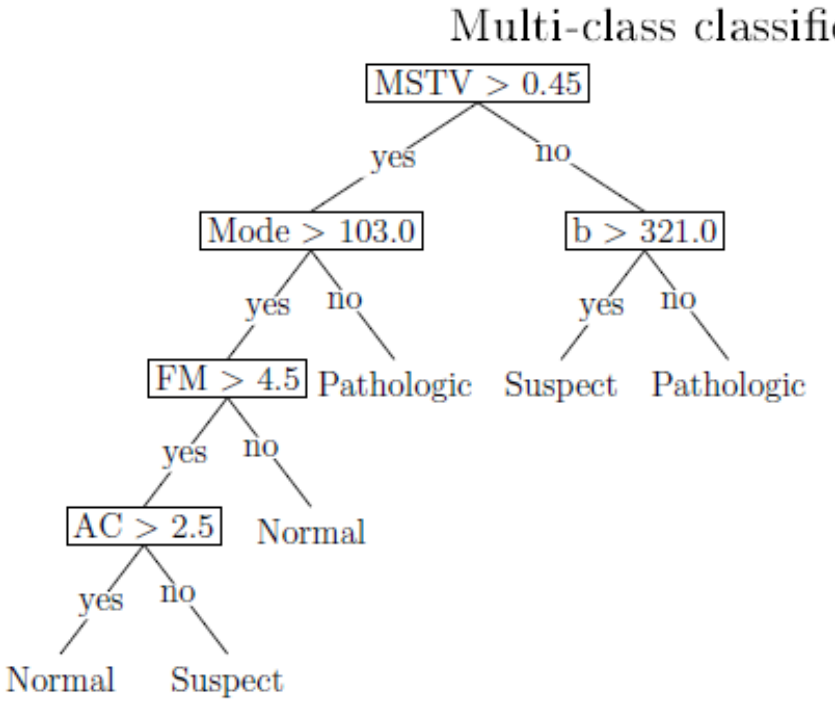
$$Var(E) = \frac{1}{T+D} \cdot \left(w \cdot \sum_{i=1}^{T} Var(Y_i) + (1-w) \cdot \sum_{j=1}^{D} Var(X_j) \right)$$

T = #target attributes, D = #descriptive attributes

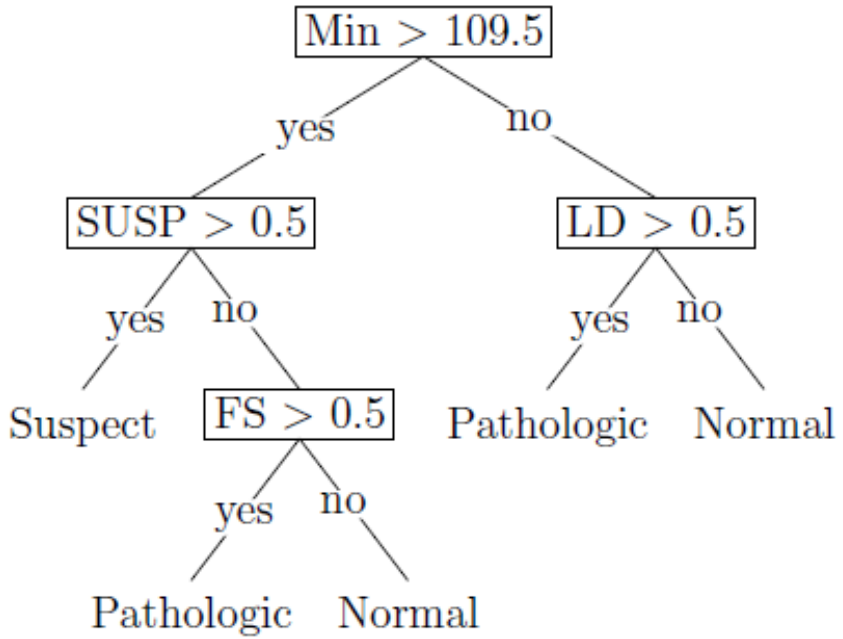
$$E = E_{\text{Labeled}} \cup E_{\text{Unlabeled}}$$

Variances only calculated for non-missing values

SSL OF DECISION TREES: ACCURACY & INTERPRETABILITY



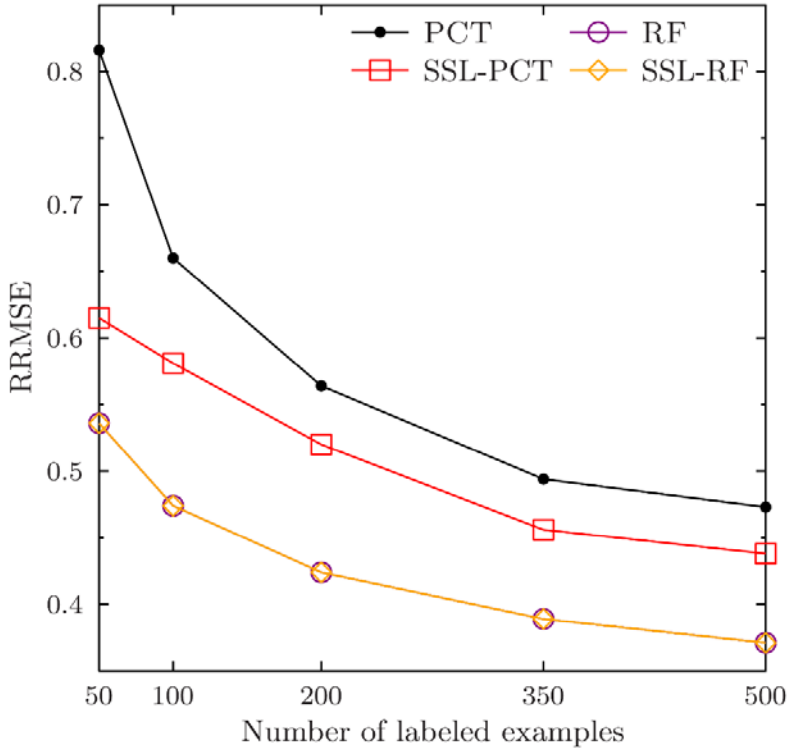
(c) SL-PCT, 50 labeled examples



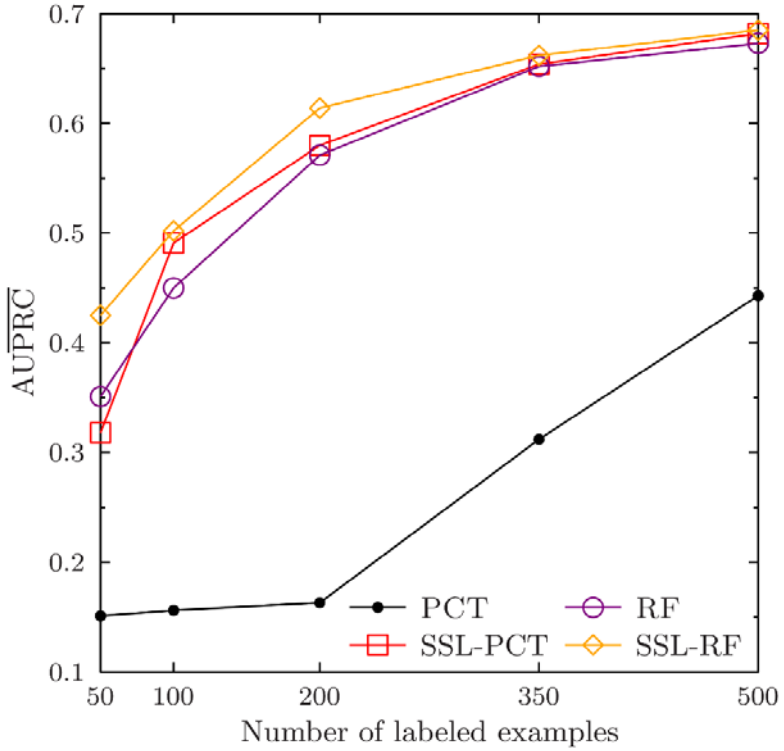
(d) SSL-PCT, 50 labeled and 2076 unlabeled examples

SSL vs. SL PERFORMANCE

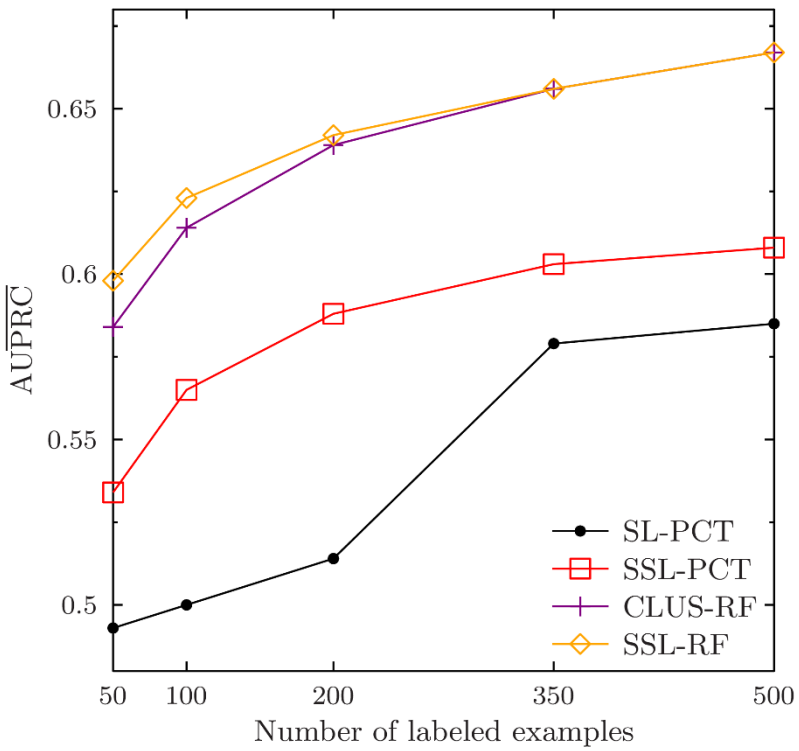
RF1 (MTR)



Medical (MLC)



Enron (HMLC)



DEPARTMENT OF KNOWLEDGE TECHNOLOGIES
Jožef Stefan Institute

INTERPRETABLE MODELS: DECISION TREES

The classical task of predictive modeling:

Single-target classification

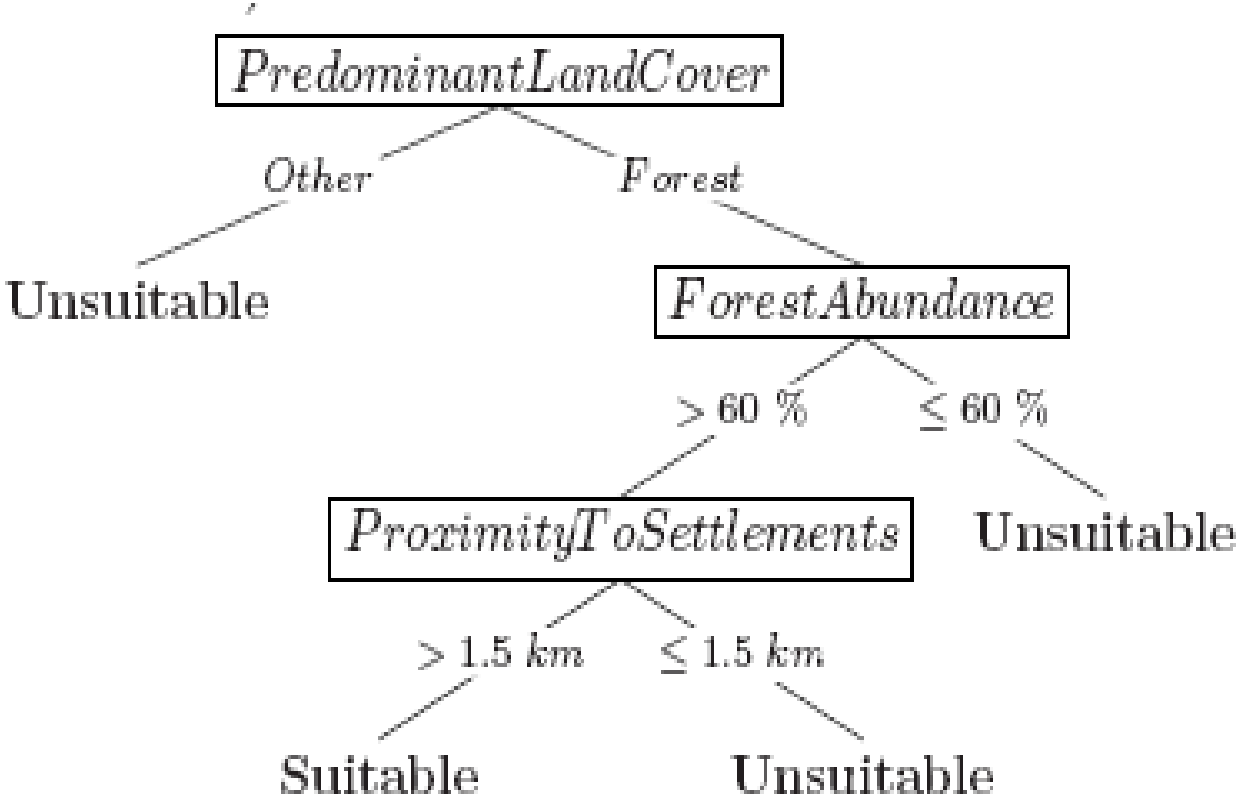
- *Given is a table of data, a single target to predict*
- *Learn a model to predict target from rest*
- *Modeling habitat suitability / species distribution*

Location	PLC	FOREST-ABUNDANCE	PTS	OtherEnvVariables	BBH
l1	Forest	80	21.4	...	Yes
l2	Forest	66	13.9	...	Yes
l3	Forest	55	50.0	...	No
l4	Forest	72	1.2	...	No
l5	Grassland	6	19.1	...	No
l6	Grassland	0	11.4	...	No
l7	Wetland	3	5.8	...	No
l8	Water	0	3.9	...	No

DEPARTMENT OF KNOWLEDGE TECHNOLOGIES
Jožef Stefan Institute

INTERPRETABLE MODELS: DECISION TREES

Models of habitat suitability for brown bears in Slovenia



DEPARTMENT OF KNOWLEDGE TECHNOLOGIES
Jožef Stefan Institute

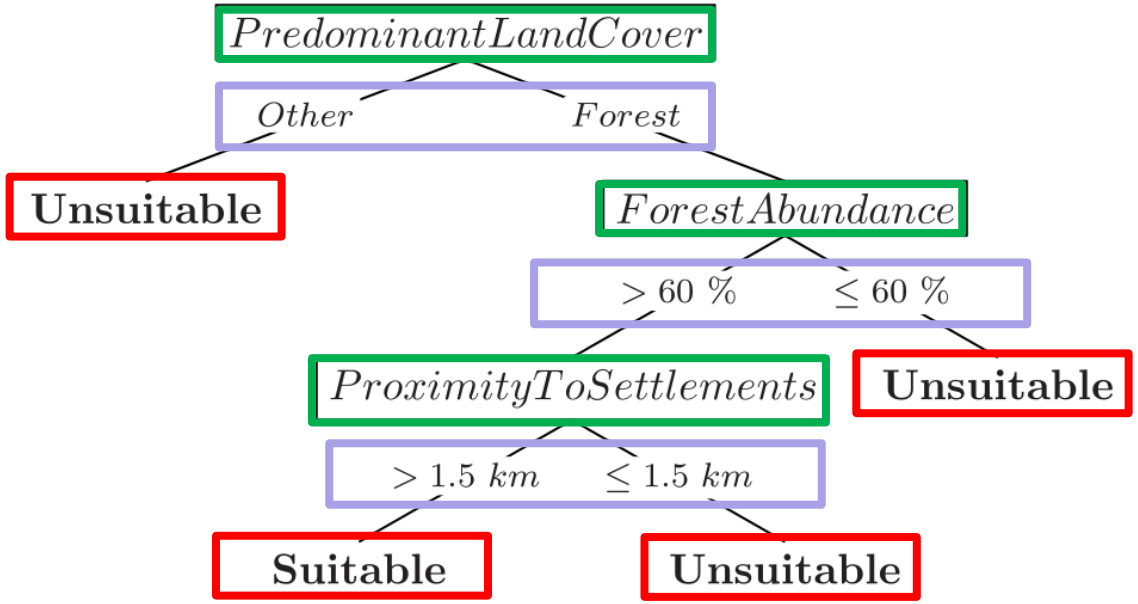
INTERPRETABLE MODELS: DECISION TREES

Hierarchically structured predictive model

Nodes – correspond to (environmental) variables

Arcs – possible values of the variables

Leafs – prediction of the target variable



DEPARTMENT OF KNOWLEDGE TECHNOLOGIES
Jožef Stefan Institute

INTERPRETABLE MODELS: DECISION TREES

To make a prediction with a decision tree:

Take as input values of attributes/ independent vars.

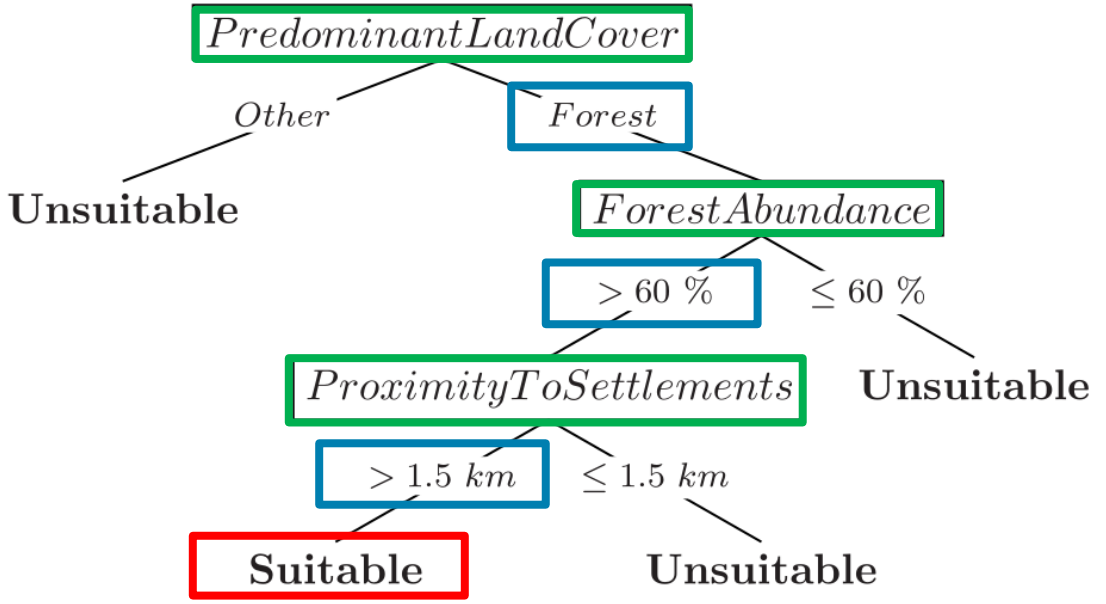
Follow branches according

to the values of these

Until you reach a leaf

PLC	FOREST-ABUNDANCE	PTS	BBH
Forest	80	21.4	?

Yes



DEPARTMENT OF KNOWLEDGE TECHNOLOGIES
Jožef Stefan Institute

INTERPRETABLE MODELS: IF-THEN RULES

Location	PLC	FOREST-ABUNDANCE	PTS	OtherEnvVariables	BBH
l1	Forest	80	21.4	...	Yes
l2	Forest	66	13.9	...	Yes
l3	Forest	55	50.0	...	No
l4	Forest	72	1.2	...	No
l5	Grassland	6	19.1	...	No
l6	Grassland	0	11.4	...	No
l7	Wetland	3	5.8	...	No
l8	Water	0	3.9	...	No

```
IF    PREDOMINANT-LAND-COVER = Forest
AND   FOREST-ABUNDANCE > 60%
AND   PROXIMITY-TO-SETTLEMENTS > 1.5 km
THEN BrownBearHabitat = Suitable
```

DEPARTMENT OF KNOWLEDGE TECHNOLOGIES
Jožef Stefan Institute

INTERPRETABLE MODELS: LOGISTIC REGRESSION

Logistic regression models the probability that an instance belongs to a particular class using the logistic (sigmoid) function:

$$P(y = 1 \mid \mathbf{x}) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x_1 + \beta_2 x_2 + \cdots + \beta_n x_n)}}$$

Where:

- $\mathbf{x} = (x_1, x_2, \ldots, x_n)$: feature vector
- β_0: intercept
- β_i: coefficient (weight) for feature x_i

The **log-odds** is modeled linearly:

$$\log\left(\frac{P}{1-P}\right) = \beta_0 + \sum_{i=1}^{n} \beta_i x_i$$

Each coefficient β_i represents the **change in the log-odds** of the outcome per unit increase in x_i, holding all other features constant

For logistic regression, **local and global interpretability are aligned**.

- **Global**: Coefficients explain the model's behavior across the dataset.
- **Local**: For single prediction, log-odds contributions from each feature βixi

DEPARTMENT OF KNOWLEDGE TECHNOLOGIES
Jožef Stefan Institute

INTERPRETABLE MODELS: LOGISTIC REGRESSION

Feature contributions example

Predicting whether a patient has diabetes, for patients described with the features age, glucose & BMI

Prediction: Probability of Diabetes = 0.85

Intercept: -2.0

Feature Contributions ($\beta_i x_i$):

+ Glucose (β=0.04, x=180) → +7.2

+ BMI (β=0.08, x=30) → +2.4

+ Age (β=0.01, x=50) → +0.5

Log-odds = -2.0 + 7.2 + 2.4 + 0.5 = 8.1

Predicted probability = 1 / (1 + e^{-8.1}) ≈ 0.85

DEPARTMENT OF KNOWLEDGE TECHNOLOGIES
Jožef Stefan Institute

INTERPRETABLE MODELS: NAÏVE BAYES

Naive Bayes is a **probabilistic classifier** based on **Bayes' Theorem** with a **naive assumption** that features are conditionally independent given the class label:

$$P(C_k \mid \mathbf{x}) = \frac{P(C_k) \cdot \prod_{i=1}^{n} P(x_i \mid C_k)}{P(\mathbf{x})}$$

Where:

- C_k: class
- $\mathbf{x} = (x_1, x_2, \ldots, x_n)$: feature vector
- $P(C_k)$: prior probability of class C_k
- $P(x_i \mid C_k)$: likelihood of feature x_i given class C_k

Since $P(\mathbf{x})$ is the same across all classes, classification is done via the **maximum a posteriori (MAP)**:

$$\hat{y} = \arg\max_{C_k} P(C_k) \cdot \prod_{i=1}^{n} P(x_i \mid C_k)$$

DEPARTMENT OF KNOWLEDGE TECHNOLOGIES
Jožef Stefan Institute

INTERPRETABLE MODELS: NAÏVE BAYES

Naive Bayes makes decisions based on how strongly each feature **supports or refutes a class**, through P(xi|Ck). For explanation:

- Each feature's **likelihood** contributes multiplicatively to the total posterior.
- In **log-space**, this becomes additive:

$$\log P(C_k \mid \mathbf{x}) \propto \log P(C_k) + \sum_{i=1}^{n} \log P(x_i \mid C_k)$$

- This makes explanation **similar in form** to linear models, as the log-probability is decomposed into **additive contributions** from each feature

An example from text classification

Suppose a Naive Bayes classifier is trained to classify documents as “spam” or “not spam.” For a given email:

- Words like “offer” or “winner” will have high P(xi|spam)
- Words like “meeting” or “report” may favor P(xi|not spam)

You can explain the prediction by ranking the most influential words:

- “offer” → log-likelihood +2.3
- “winner” → log-likelihood +1.8
- “meeting” → log-likelihood −0.7

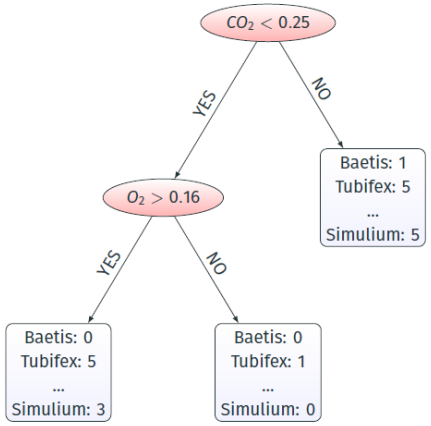
EXPLAINABLE MODELS: TREE ENSEMBLES

Ensembles are collections of models (e.g., forests of trees)

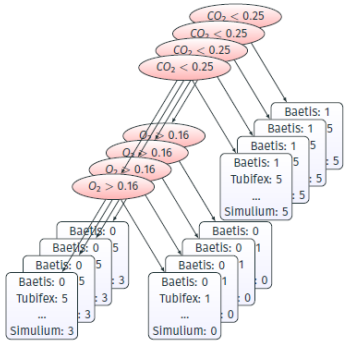
Whose predictions are combined to obtain a final prediction

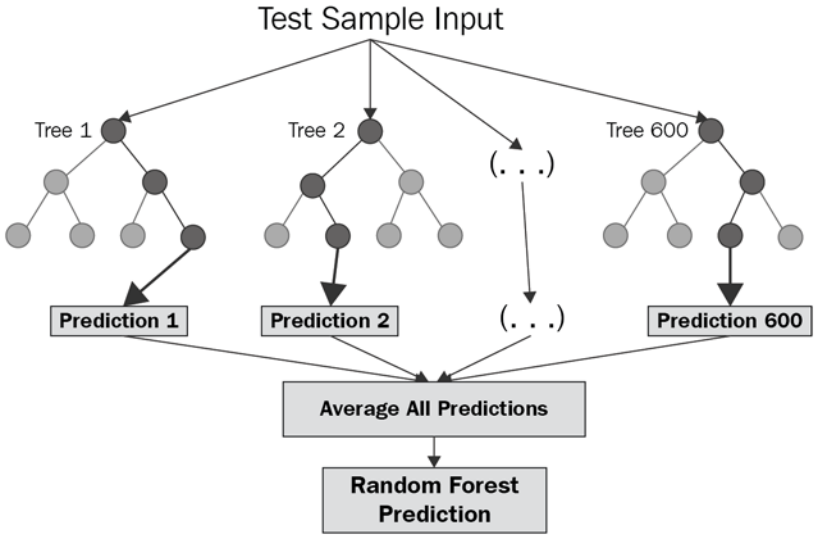
Can have much higher predictive performance vs. single models

A single decision tree



An ensemble of decision trees





Example: Random forests, which can be very efficient and accurate

But, are not interpretable

However, tree ensembles can have internal mechanism, which can provide feature importance estimates and thus some insight

Why Use Many Trees?

The Power of Randomization

Random forests employ

two key randomization techniques:

- Bootstrap sampling: Each tree trains on a different random subset of data
- Feature randomization: Each node considers only a random subset of features

The Wisdom of the Crowd

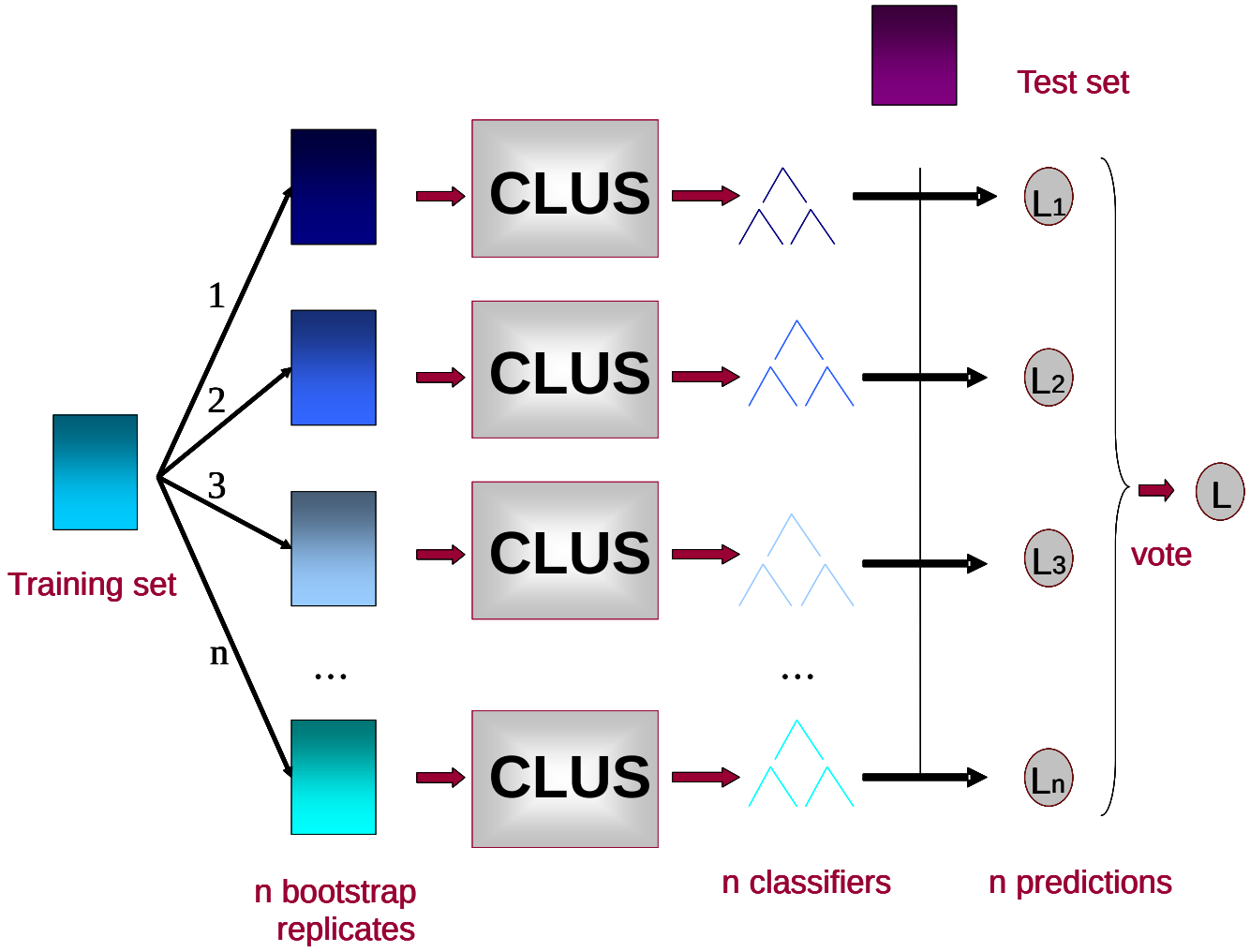
The final prediction combines all trees:

- For regression: Average of all tree predictions
- For classification: Majority vote among all trees

This aggregation dramatically reduces overfitting and improves generalization to new data.

EXPLAINABLE MODELS: LEARNING TREE ENSEMBLES

Typical approach: Generate different samples of the data (subsets of rows, subset of columns, or both), then learn a tree on each



INTRODUCING FEATURE IMPORTANCE

DEPARTMENT OF KNOWLEDGE TECHNOLOGIES
Jožef Stefan Institute

Relative Influence

Not all input variables contribute equally to the prediction.

Feature importance provides a quantitative ranking of each feature's influence on the model's output.

Model Interpretation

Reveals which inputs drive model decisions, turning the "black box" of machine learning into a more transparent, interpretable system.

Feature Selection

Helps identify which variables are worth measuring in future experiments, potentially saving research time and resources.

FEATURE IMPORTANCE ESTIMATION

Assess the importance of features through a score

A ranking is a list of features, sorted wrt. the scores

x_1	x_2	$\cdots$	x_F	y
3.1	M	$\cdots$	2.7	$\{\ell_3\}$
6.6	F	$\cdots$	2.8	$\{\ell_1, \ell_4, \ell_{159}\}$
$\vdots$	$\vdots$		$\vdots$	$\vdots$
1.3	F	$\cdots$	1.1	$\{\ell_{26}\}$

$\Longrightarrow$

feature	relevance
x_{17}	4.32
x_{12}	4.12
$\vdots$	$\vdots$
x_{41}	0.01

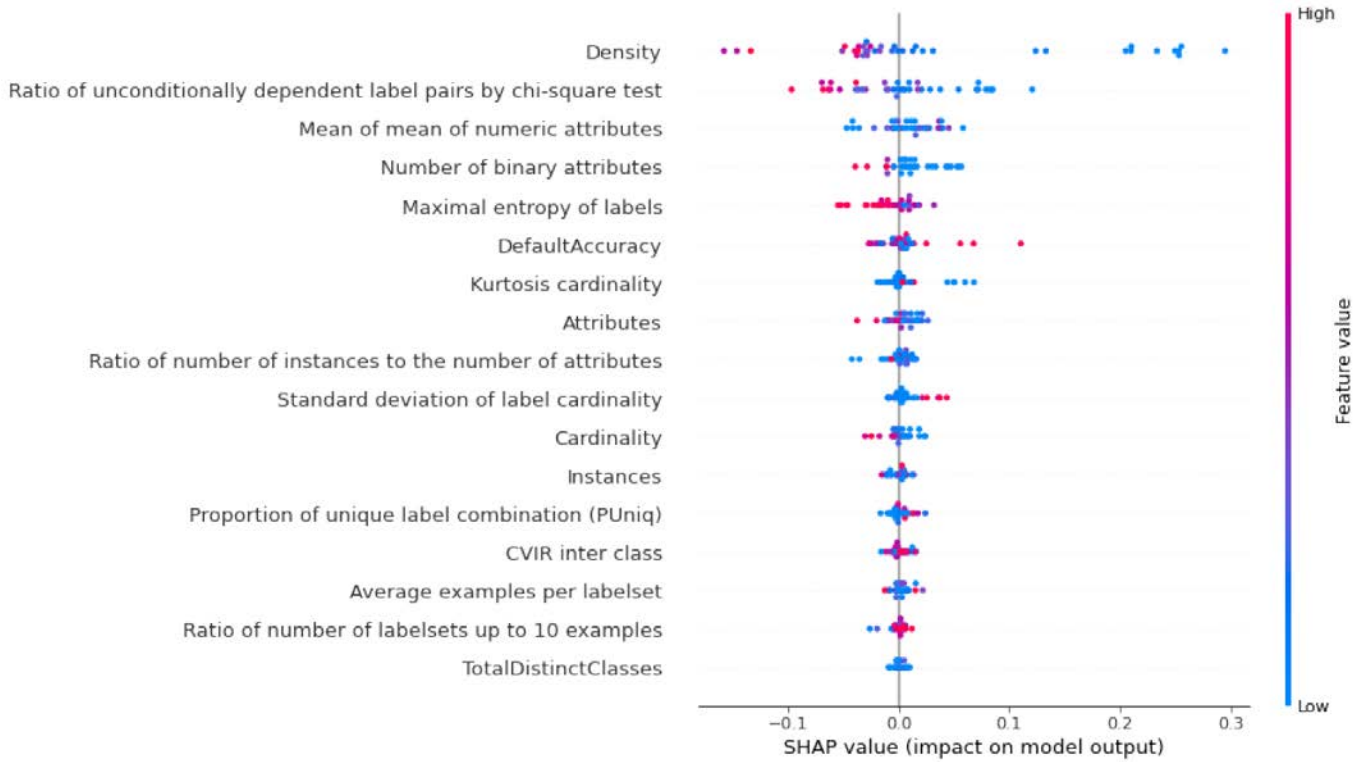
Example task: biomarker discovery from gene expr. data

Typically for a single disease/diagnosis

With MTP also for sets of clinical scores or a set of diseases

Allows for nice visualisations of (relative) feature importances

SOME VISUALISATIONS OF FEATURE IMPORTANCES



FEATURE IMPORTANCE ESTIMATION IN TREE ENSEMBLES

Extrinsic (external) methods: Model-agnostic, i.e., can be used with any kind of model, not just tree ensembles

- Permutation importance
- SHAP

Intrinsic (internal methods): Use the internal structure of the model (tree ensemble) or information available during model construction

- Works for different approaches to ensemble learning (e.g., bagging, random forest, boosting/gradient-boosting)
- Works for different supervision (fully-, semi-, & un-supervised)
- Works for different targets (classification/regression, ST/MT)

DEPARTMENT OF KNOWLEDGE TECHNOLOGIES
Jožef Stefan Institute

FEATURE IMPORTANCE ESTIMATION IN TREE ENSEMBLES: The Symbolic Score

Relies on the fact that we have an ensemble of trees

But can be run after the ensemble has been constructed

The symbolic score counts the appearances (and position) of each attribute xi in all trees in the forest

Attributes that appear higher in the tree (closer to the root) have lower depth and are more important

The counting of appearances is weighted as follows

$$importance_{\text{SYMB}}(x_i) = \frac{1}{|\mathcal{F}|} \sum_{\mathcal{T} \in \mathcal{F}} \sum_{\mathcal{N} \in \mathcal{T}(x_i)} w^{\text{DEPTH}(\mathcal{N})}$$

where w is between 0 and 1

FEATURE IMPORTANCE ESTIMATION IN TREE ENSEMBLES: The GENIE3 Score

Mean Decrease of Impurity: Features that consistently lead to large reductions in impurity across many trees receive higher importance scores.

Measure Node Impurity

Calculate how mixed or "impure" the data is at each node before splitting (using Gini impurity for classification or variance for regression)

Calculate Impurity Reduction

Compute how much purity improves after splitting on a particular feature

Aggregate Across Forest

Sum the impurity reductions for each feature across all nodes in all trees of the forest

DEPARTMENT OF KNOWLEDGE TECHNOLOGIES
Jožef Stefan Institute

FEATURE IMPORTANCE ESTIMATION IN TREE ENSEMBLES: The GENIE3 Score

Mean Decrease of Impurity: Features that consistently lead to large reductions in impurity across many trees receive higher importance scores.

For a forest $\mathcal{F}$ of trees $\mathcal{T}$, the Genie3 feature importance is defined as

$$importance_{\text{GENIE3}}(x_i) = \frac{1}{|\mathcal{F}|} \sum_{\mathcal{T} \in \mathcal{F}} \sum_{\mathcal{N} \in \mathcal{T}(x_i)} |E(\mathcal{N})| h^*(\mathcal{N}),$$

where $E(\mathcal{N})$ is the set of examples that come to the node $\mathcal{N}$, and $h^*(\mathcal{N})$ is the value of the variance reduction function.

$$h = Var(E) - \sum_{E_i \in \mathcal{P}} \frac{|E_i|}{|E|} Var(E_i)$$

Caveats and Biases in Feature Importance

Potential Biases

- High-cardinality bias: Features with more unique values tend to be favored
- Correlation bias: When features are correlated, importance can be split between them
- Scale sensitivity: Features with wider ranges may appear more important

Validity Checks

Always validate feature importance results:

- Compare with domain knowledge
- Use multiple importance metrics to confirm findings
- Test importance stability across different random seeds

Remember that feature importance is a model-specific measure, not an absolute property of the data.

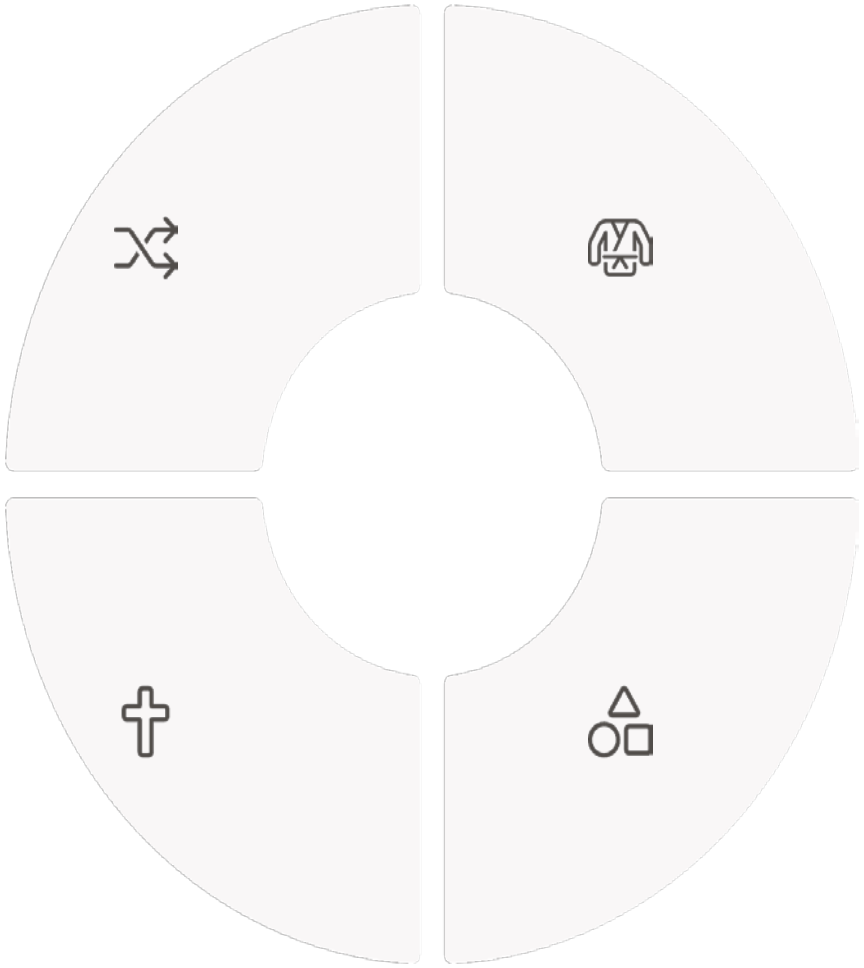
Alternatives: Model-agnostic approaches

Permutation Importance

Measures accuracy drop when feature's values randomly shuffled
Less biased for correlated features, but more computationally expensive

Mean Decrease of Impurity

Takes into account how much the impurity is reduced by each feature in each tree of the forest



Symbolic Score

Depends only on the forest, on the data only indirectly (through the forest)
Very efficient to compute

SHAP Values

Based on game theory, provides consistent feature attribution
Shows both magnitude and direction of feature impact

DEPARTMENT OF KNOWLEDGE TECHNOLOGIES
Jožef Stefan Institute

THE PERMUTATION IMPORTANCE SCORE

Originally introduced by Leo Breiman in the context of Random Forests, but can be used with any model

- *Randomly shuffle feature values, measure decrease in model performance.*
- *Reflects importance based on predictive contribution.*

- *Train a Random Forest.*
- *Measure the baseline accuracy of the model on out-of-bag (OOB) data.*
- *For each feature:*
  - *Randomly permute its values in the OOB data.*
  - *Measure the drop in accuracy (or increase in error) caused by this permutation.*
- *The larger the drop, the more important the feature.*

DEPARTMENT OF KNOWLEDGE TECHNOLOGIES
Jožef Stefan Institute

THE PERMUTATION IMPORTANCE SCORE

Originally introduced by Leo Breiman in the context of Random Forests, but can be used with any model

- Procedure:
  - Grow a tree $\mathcal{T}$ and evaluate its performance on
    - $\text{OOB}_\mathcal{T}$: out of bag examples ($\mathscr{D}_{\text{TRAIN}} \setminus \mathcal{B}_\mathcal{T}$)
    - $\text{OOB}^i_\mathcal{T}$: $\text{OOB}_\mathcal{T}$, where (only) the values of x_i are permuted
  - Compute average relative increase of the error over the trees in the forest, namely

$$importance_{\text{RF}}(x_i) = \frac{1}{|\mathcal{F}|} \sum_{\mathcal{T} \in \mathcal{F}} \frac{Err(\text{OOB}^i_\mathcal{T}) - Err(\text{OOB}_\mathcal{T})}{Err(\text{OOB}_\mathcal{T})}.$$

CALCULATING THE PERMUTATION IMPORTANCE SCORE IN RFs

Intuition: Shuffling important attributes should decrease the predictive performance of trees

procedure Induce_RF($E, k, f(x)$)
returns Forest, Importances

1: $F = \emptyset$
2: $I = \emptyset$
3: **for** $i = 1$ **to** k **do**
4: $E_i = \text{Bootstrap\_sample}(E)$
5: $Tree_i = PCT_{rand}(E_i, f(x))$
6: $F = F \bigcup Tree_i$
7: $E_{OOB} = E \setminus E_i$
8: $\text{Update\_Imp}(E_{OOB}, Tree, I)$
9: $I = \text{Average}(I, k)$
10: **return** F, I

procedure Update_Imp($E_{OOB}, Tree, I$)

1: $Err_{OOB} = \text{Evaluate}(Tree, E_{OOB})$
2: **for** $j = 1$ **to** D **do**
3: $E_j = \text{Randomize}(E_{OOB}, j)$
4: $Err_j = \text{Evaluate}(Tree, E_j)$
5: $I_j = I_j + (Err_j - Err_{OOB})/Err_{OOB}$
6: **return**

procedure Average(I, k)

1: $I^T = \emptyset$
2: **for** $l = 1$ **to** $size(I)$ **do**
3: $I_l^T = I_l/k$
4: **return** I^T

EXPLAINING PREDICTIONS: UNDERSTANDING FEATURE CONTRIBUTIONS WITH SHAP

SHAP = SHapley Additive exPlanations

SHAP's foundation comes from cooperative game theory, specifically Shapley values, developed by Lloyd Shapley in 1953.

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} [f(S \cup \{i\}) - f(S)]$$

The key insights:

- Consider each feature as a "player" in a team game
- The "game" is producing accurate predictions
- Shapley values calculate each player's (feature's) fair contribution to the team's success
- SHAP takes into account all possible combinations of players

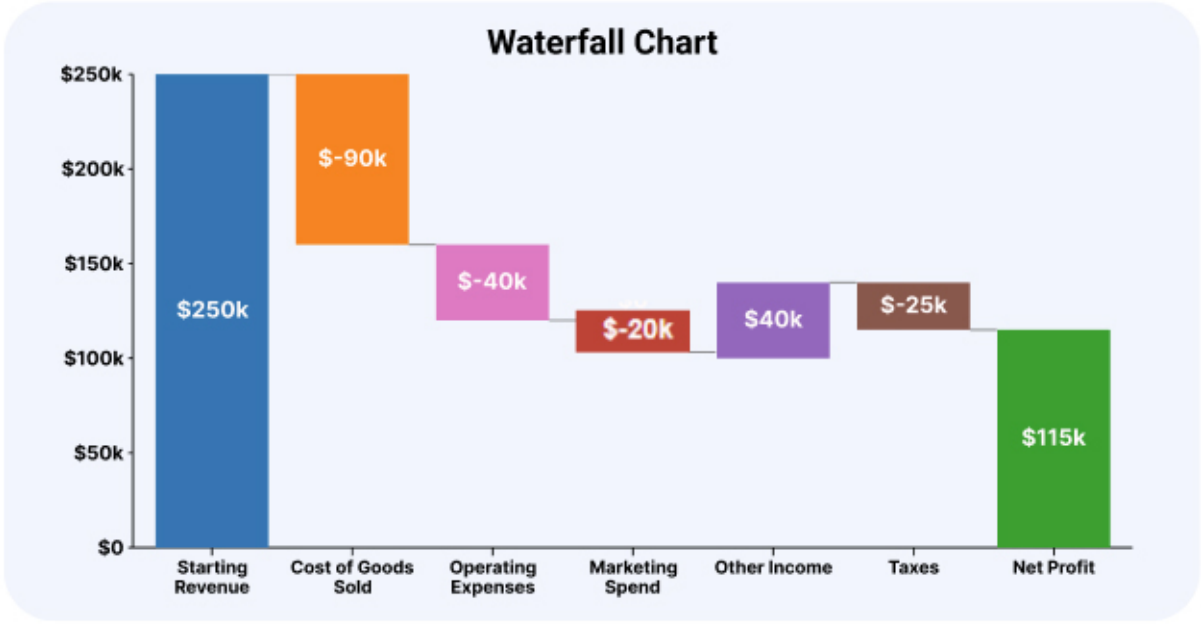
SHAP IN SIMPLE TERMS

Decomposing a prediction into contributions from each input variable

Exploit additivity: SHAP values sum up to the difference between the prediction and the average output

- Start with a baseline prediction (average model output)
- Calculate how each feature moves the prediction away from the baseline
- Sum all feature effects to reach the final prediction

A waterfall chart can show how features push a prediction higher or lower from the baseline value



Waterfall Chart: From Initial Value to Final Insight

SHAP Example: Predicting House Prices

Location: +$50K

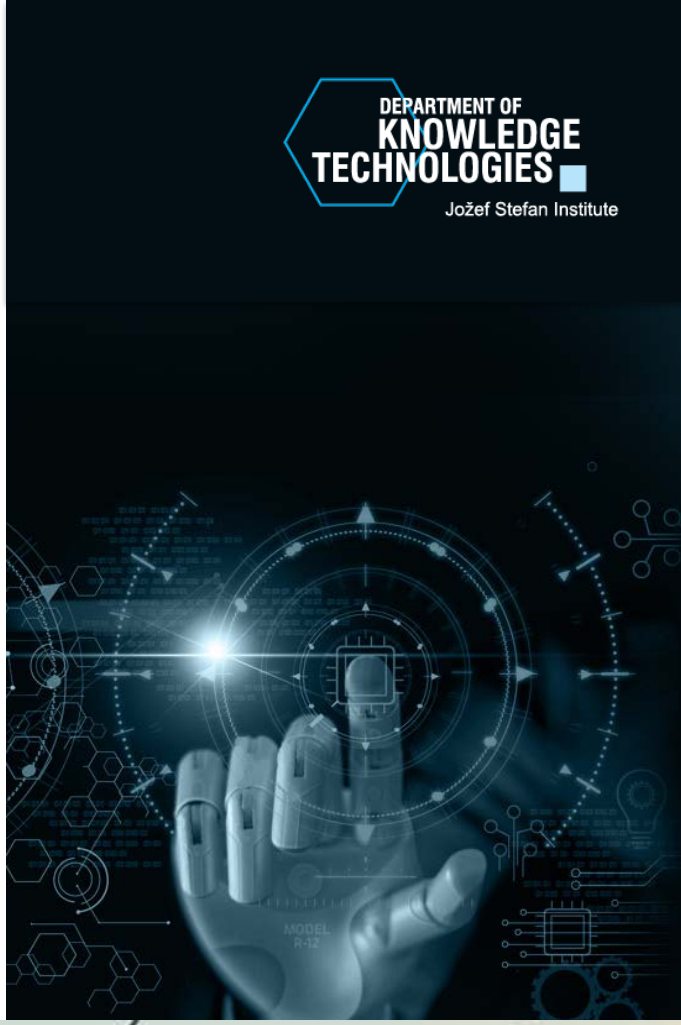
Being in a desirable neighborhood significantly increases the prediction

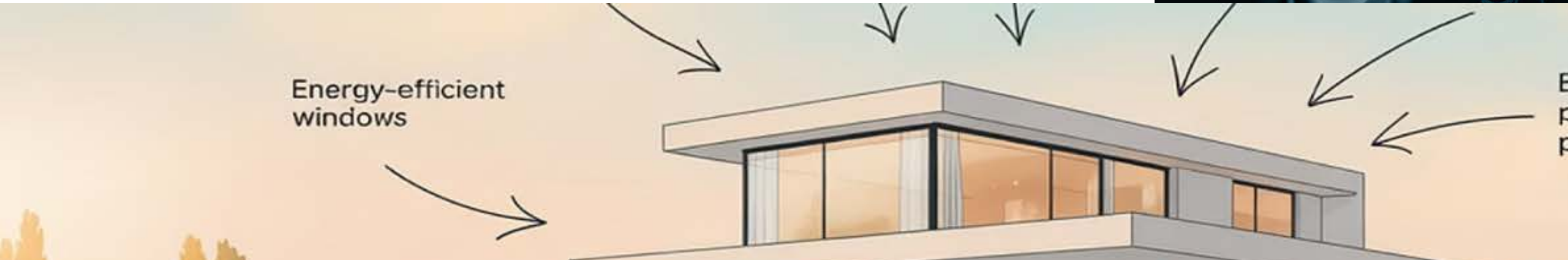
Size: +$30K

Square footage adds substantial value to the prediction

Age: -$10K

Older construction age decreases the predicted price





Another SHAP Example: Predicting Diabetes

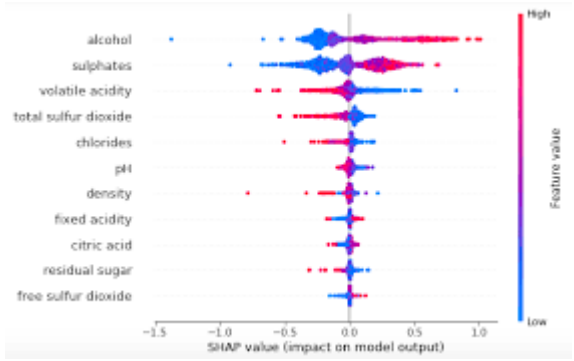
Input features: Age, BMI, Glucose

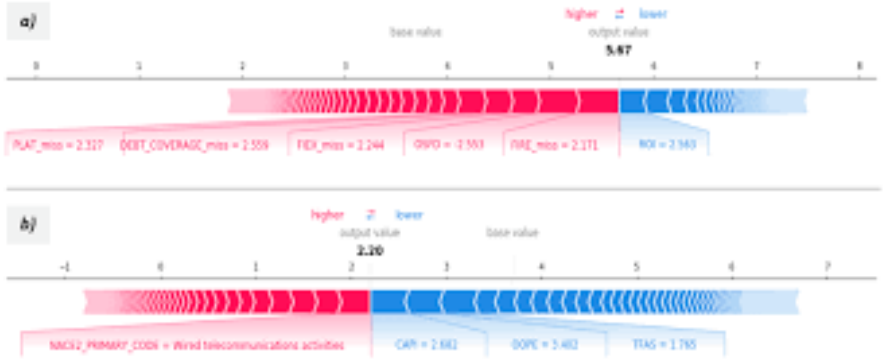
- Model predicts risk of diabetes (0.9 for patient X)
- Average prediction across population: 0.5
- SHAP values:
  - Glucose: +0.25
  - BMI: +0.10
  - Age: +0.05
- Total SHAP values = 0.4 → aligns with 0.9 - 0.5

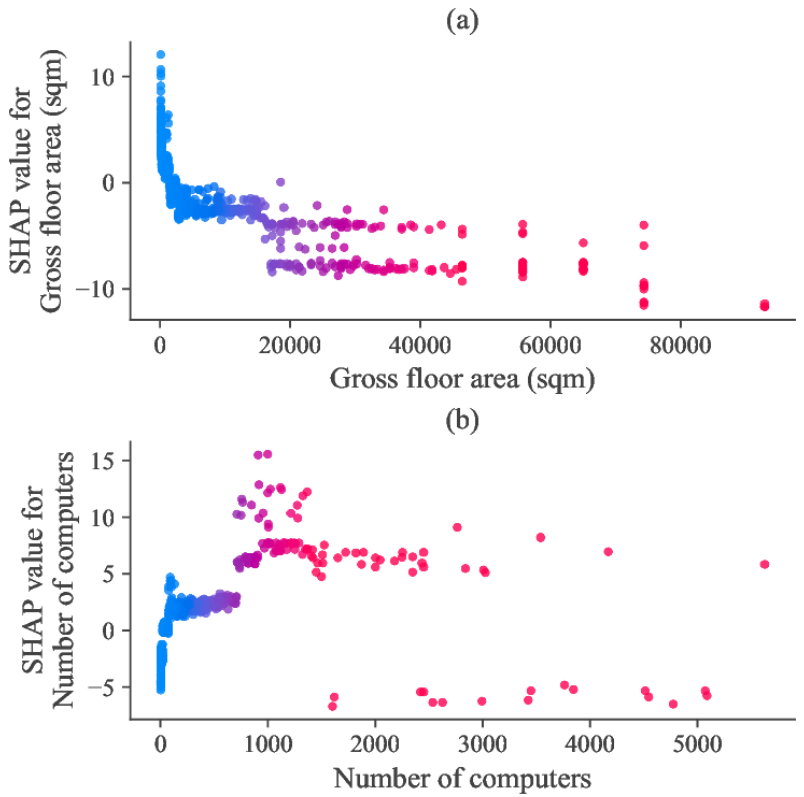
Visualizing SHAP Values

Different visualization

- SHAP Summary Plot: Global view of feature importance.
- SHAP Force Plot: Local explanation for individual predictions.
- SHAP Dependence Plot: Feature value dependency.







Key Properties of SHAP

Model-Agnostic

Works with any machine learning algorithm, from simple linear models to complex neural networks

Fair Attribution

Consistent mathematical foundation ensures features receive appropriate credit

Consistency

Same explanation for identical predictions, providing reliability

Handles Interactions

Accounts for complex feature interactions that other methods miss

DEPARTMENT OF KNOWLEDGE TECHNOLOGIES
Jožef Stefan Institute

Explaining Predictions with LIME: Local Interpretable Model-Agnostic Explanations

- *Local = Focused explanatons: Explains individual predictions (for a given example) rather than the entire model*
- *Model-Agnostic = Universal: Works with any predictive model*
- *Proposed by Ribeiro, Singh, & Guestrin (2016 KDD paper) "Why Should I Trust You?: Explaining the Predictions of Any Classifier"*
- ***LIME explains individual predictions***

  by building local surrogate models
- *A surrogate models is*
  - *a simple, interpretable, and efficient model that*
  - *approximates the behavior of a complex model in a limited region*
- *Common types of surrogate models (used in LIME)*
  - *Decision trees (of limited depth)*
  - *Classification rules*
  - *Linear equations*

DEPARTMENT OF KNOWLEDGE TECHNOLOGIES
Jožef Stefan Institute

THE LIME ALGORITHM

1. **Choose the model M (Y=f(X)) and a reference point x_0**
   (for which an explanation is to be provided)
2. **Generate points from the domain of M**
   *(sample **X** values from a Normal distribution inferred from the training set)*
3. **Predict the *Y* coordinate(s) of the sampled points**,
   using the model M
4. **Assign weights based on the closeness to the chosen point**
   (use RBF Kernel, assign higher weights to points closer to $\mathbf{x_0}$*)*

$$RBF(x^{(i)}) = \exp\left(-\frac{||x^{(i)} - x^{(ref)}||^2}{kw}\right)$$

5. **Train explanation model E (**$Y = \beta_0 + \sum \beta_j X_j$**) using Linear Ridge Regression on the weighted dataset:**
   The β coefficients are regarded as the LIME explanation
6. **Extract explanation** *(e.g., interpret the coefficients of E)*

DEPARTMENT OF KNOWLEDGE TECHNOLOGIES
Jožef Stefan Institute

LEARNING THE SURROGATE MODEL IN LIME

- When learning the surrogate model,
  LIME optimizes the following objective function:

$$\xi(x) = \underset{g \in G}{\operatorname{argmin}} L(f, g, \pi_x) + \Omega(g)$$

- *Where*
  - *f is the complex model*
    (for whose prediction an explanation is needed)
  - *g is the interpretable surrogate*
  - π_x *is the locality weighting kernel*
  - *Ω(g) is a complexity penalty*

DEPARTMENT OF KNOWLEDGE TECHNOLOGIES
Jožef Stefan Institute

LIME:
An example explanation

- *Consider a black-box model D = f(X)*
  predicting whether a patient has diabetes
- *For patients described with the features age, glucose & BMI*

- *Given a patient* x_0
  - *LIME perturbs* x_0 *to generate samples around it*
  - *It then trains a local linear model on these samples*
  - *It explains the prediction by showing the following feature weights:*
    - *Glucose: +0.7*
    - *BMI: +0.25*
    - *Age: -0.10*

Strengths and Limitations of LIME

Strengths

- Model-agnostic: works with any black box model
- Intuitive explanations for non-experts
- Flexible across data types
- Customizable interpretable models
- Open-source implementations available
- Computationally efficient for single predictions

Limitations

- Local explanations may miss global patterns
- Sensitive to perturbation method choices
- Sensitive to proximity kernel
- Unstable explanations in some cases
- Computationally intensive for large models/ datasets
- Explanations may oversimplify model complexity

DEPARTMENT OF KNOWLEDGE TECHNOLOGIES
Jožef Stefan Institute

LIME Generalizations and Extensions

- *Different types of data*
  (require different perturbations/ sampling)
  - *Tabular data: Sampling from normal dist. inferred from data*
  - *Text: Remove or replace words*
  - *Images: Turn pixels/superpixels on/off*

- *Differrent LIME extensions*
  - *LIME-SUP: Supervised partitioning of example space into regions where surrogates built, results in improved stability*
  - *Anchor-LIME: Rule-based explanations*
  - *OptiLIME: Finds the best kernel width (optimizing the trade-off between explanation stability and fidelity)*
  - *And many others: DL-LIME, Green LIME, BayLIME, …*