Decision Trees, Random Forests and Extreme Trees

Marcin Kuta

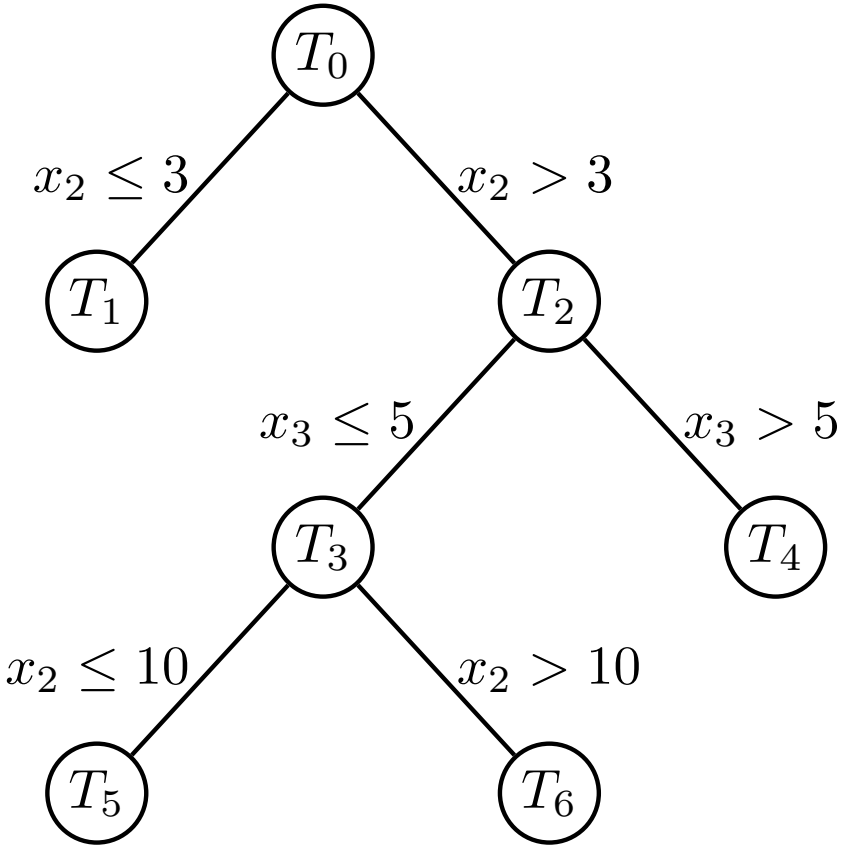
Decision Trees

- Node splitting
  - Binary
  - Multi-way
- Decision lines
  - Axis-aligned
  - Oblique
- Optimization
  - Greedy
  - Non-greedy

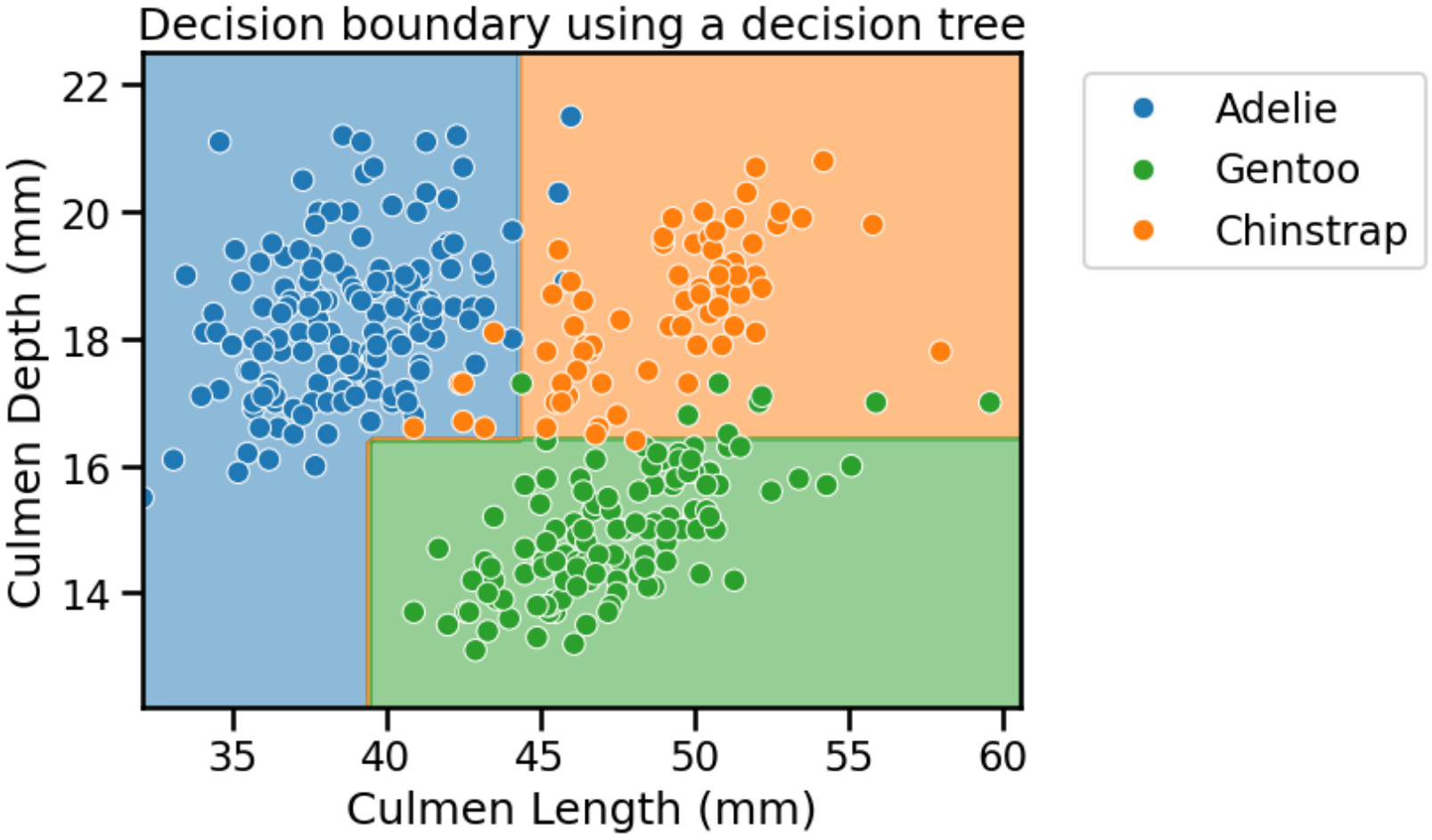
Ordinary Binary Decision Trees

- Binary
- Axis-aligned
- Greedy

Decision Trees



Decision Trees



Measures of node impurity in classification

$I(t)$ – impurity of node t

- Entropy

$$-\sum_{i=1}^{K} p(C_i|t) \log_2 p(C_i|t)$$

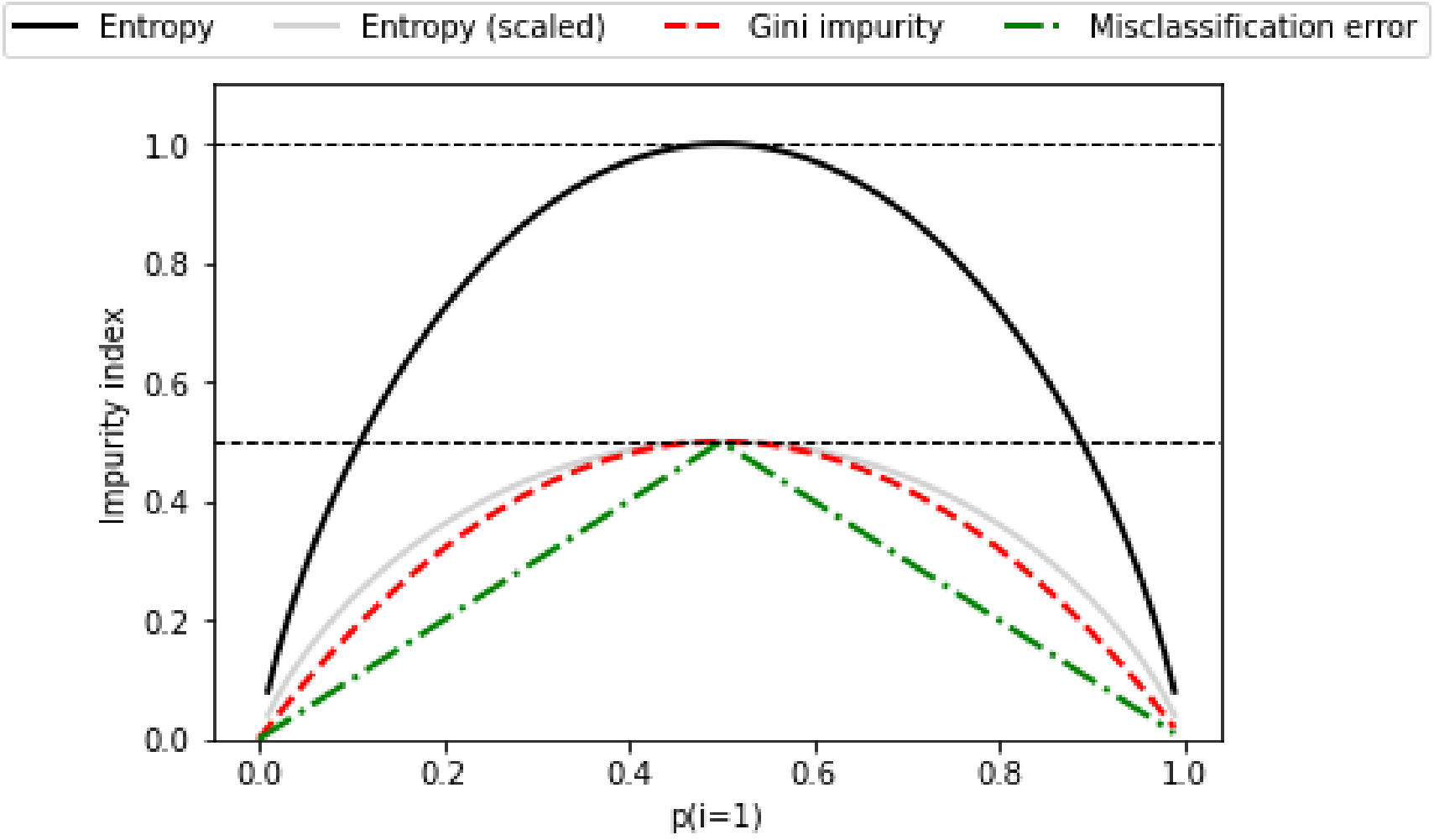
- Gini impurity

$$\sum_{i=1}^{K} p(C_i|t)(1 - p(C_i|t)) = 1 - \sum_{i=1}^{K} p(C_i|t)^2$$

- Misclassification error

$$1 - \max_i p(C_i|t)$$

Measures of node impurity



Measures of node impurity in regression

- Mean Squared Error

$$\frac{1}{n} \sum_{i:x_i \in t} (y_i - \bar{y})^2$$

- Half Poisson deviance

$$\frac{1}{n} \sum_{i:x_i \in t} (y_i \log \frac{y_i}{\bar{y}} - y_i + \bar{y})^2$$

Attribute selection criteria

Impurity reduction

$$\Delta I(t, \alpha) = I(t) - \frac{N_{\text{left}}}{N_t} I(t_{\text{left}}) - \frac{N_{\text{right}}}{N_t} I(t_{\text{right}}) \tag{1}$$

- When entropy is used as node impurity, impurity reduction is called information gain

Stop splitting criteria

- node is pure: all instances belong to one class
- all instances have the same attribute values

Pre-prunning (early stopping)

- number of instances in a node below a certain threshold
- limiting the maximum depth of the tree
- limiting the maximum number of leaves
- $\Delta I(t)$ below a certain threshold

Pre-prunning can lead to underfitting

Post-prunning

Cost-complexity prunning (weakest link pruning)

$$error'(T) = error(T) + \alpha \cdot L \tag{2}$$

α is a tuning parameter estimated through cross-validation

Class assignment rule

$$C_j = \arg\max_i p(C_i|t) \tag{3}$$

Advantages

Advantages

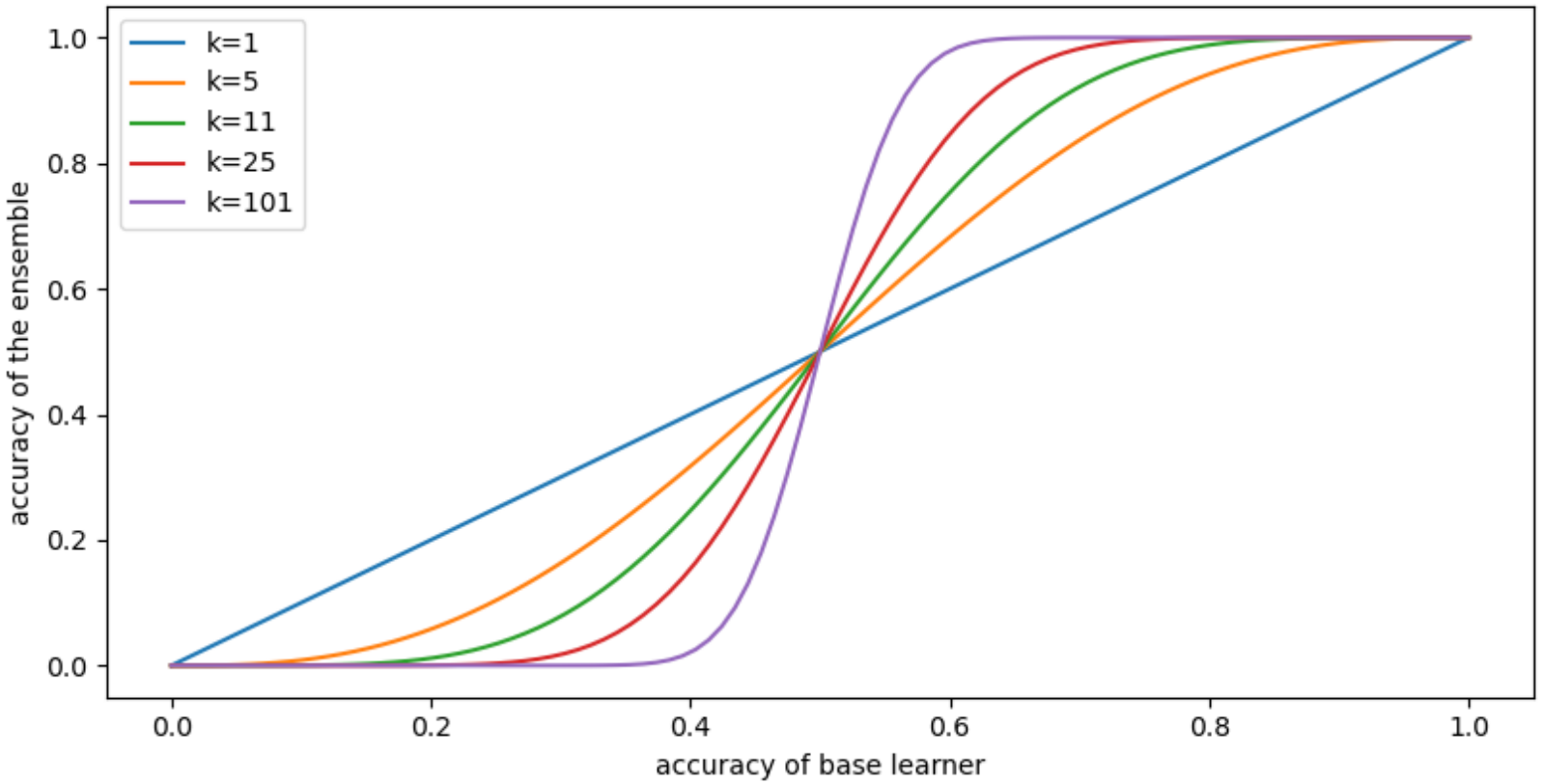
- Interpretability and visualization
- Little preprocessing is needed
  - Naturally handle categorical features, no need for dummy variables
  - Can handle missing values
  - Data normalization is not needed
- Naturally handle multi-output problems
- Fast inference time, $O(\log N)$
- Base learners for ensemble methods (Random Forests, XGBoost)

Disadvantages

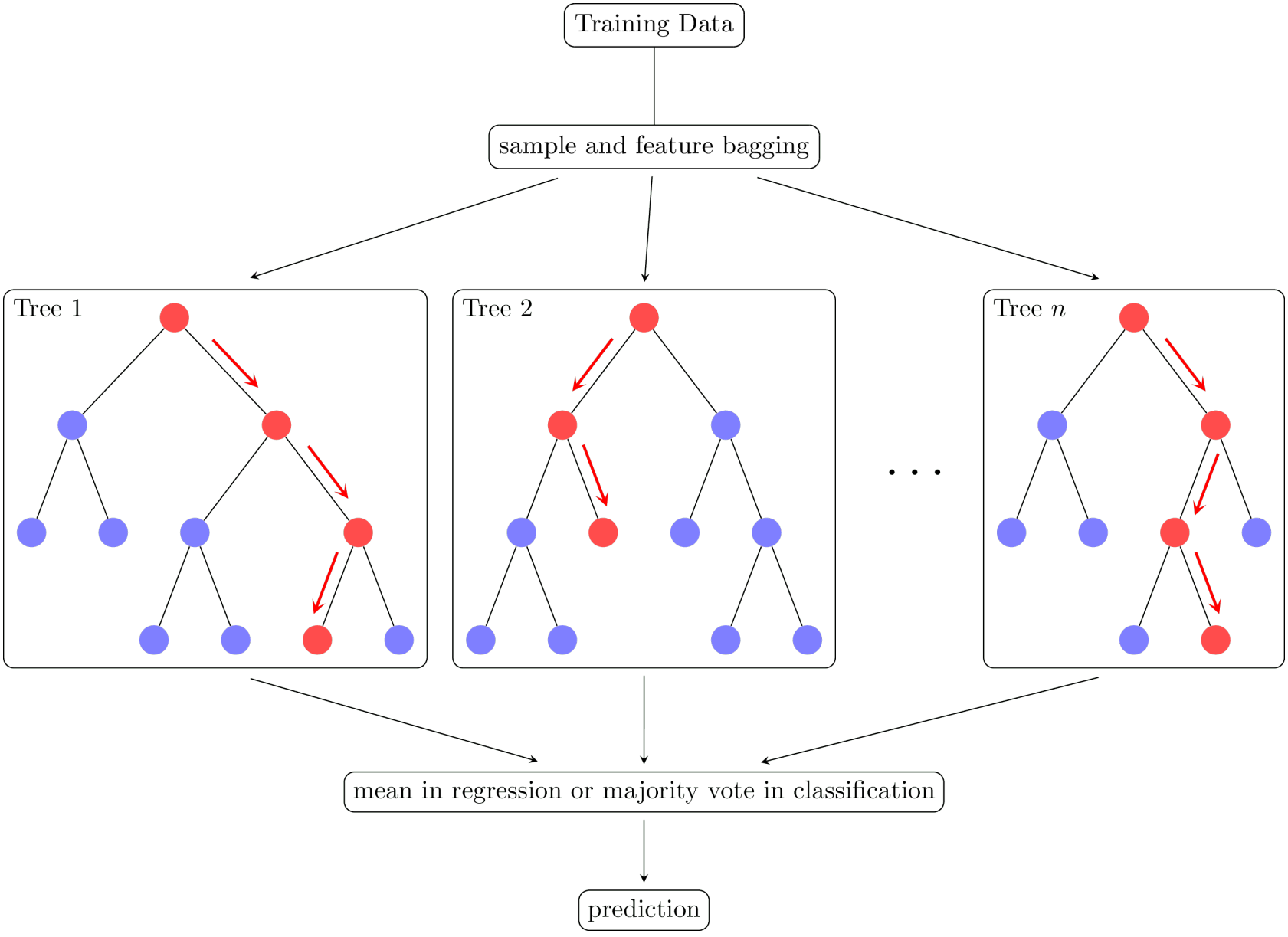
Disadvantages

- Training is difficult, decision trees easily overfit
- Instability of trees: a small change in the data can cause a large change in decision boundaries
- Lack of smoothness of the prediction surface
- The dataset should be balanced. For unbalanced data biased trees are created

Random Forests



Random Forests



Random Forests

- Ensemble of decision trees
- Bagging: Bootstrap Aggregation

$$\mathcal{D} = \{ (x_1, y_1), (x_2, y_2), (x_3, y_3), (x_4, y_4) \}$$

$$\mathcal{D}^{(1)} = \{ (x_1, y_1), (x_2, y_2), (x_3, y_3), (x_3, y_3) \}$$
$$\mathcal{D}^{(2)} = \{ (x_1, y_1), (x_4, y_4), (x_4, y_4), (x_4, y_4) \}$$
$$\mathcal{D}^{(3)} = \{ (x_1, y_1), (x_1, y_1), (x_2, y_2), (x_2, y_2) \}$$

- Random feature selection
  - In classification $p = \sqrt{m}$ features are selected in each split
  - In regression $p = m/3$ features are selected in each split

Random Forests

Data: Training data $\{ (x_i, y_i) \}_{i=1}^{N}$
Result: Random Forest

▷ `can be run in parallel`
for $i = 1$ *to num_learners* **do**
    draw a bootstrap sample $\mathcal{D}^{(i)}$ of size N from the traning data
    grow a tree T_i:
    repeat
        select p features at random from m variables
        pick the best split (feature and threshold) among p variables
        split the node into two child nodes
    until *minimum node size is reached*
Output ensemble of trees $\{ T_i \}_{i=1}^{num\_learners}$

Random Forests

Prediction for a new data point x

- Classification

  Majority vote

$$y(x) = \arg\max_C |\{ i : T_i(x) = C \}| \tag{4}$$

- Regression

  Arithmetic mean

$$y(x) = \frac{1}{N} \sum_{i=1}^{N} T_i(x) \tag{5}$$

Advantages and disadvantages

Advantages

- Random forest are resistant to overfitting
- No need for pruning trees
- Easy to set parameters
- Interpretability: variable importance
- Not very sensitive to outliers
- Embarassingly parallelizable

Disadvantages

- Extreme values are not predicted accurately
- Regression cannot predict values beyond range in the training data

Extra Trees

Extremely randomized trees

- Similar to random forests
- Each tree is trained with the whole learning sample (rather than bootstrap sample)
- A random cut-point is selected (rather than locally optimal)

References

[1] https://github.com/pietroventurini/machine-learning-notes/blob/main/3%20-%20Decision%20Trees.ipynb

[2] https://dm.cs.tu-dortmund.de/mlbits/class-dtree-learning

[3] https://jakevdp.github.io/PythonDataScienceHandbook/05.08-random-forests.html

[4] https://github.com/rasbt/machine-learning-book/blob/main/ch03/ch03.ipynb
Decision tree learning

[5] https://inria.github.io/scikit-learn-mooc/trees/trees_module_intro.html