School of COMPUTING Clemson University

2024

CPSC 4300/6300: Applied Data Science

Tree based classification

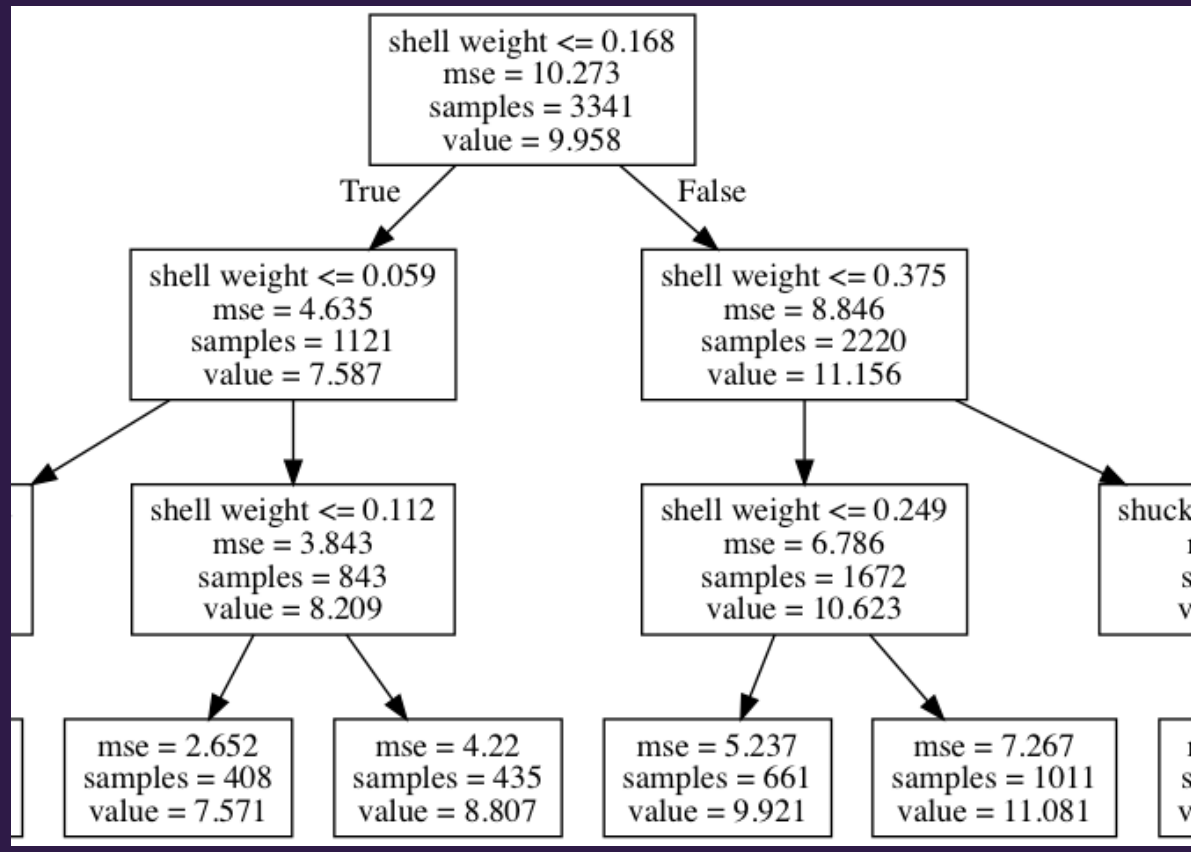
Dr. Aaron J. Masino
Associate Professor, School of Computing

College of ENGINEERING, COMPUTING AND APPLIED SCIENCES

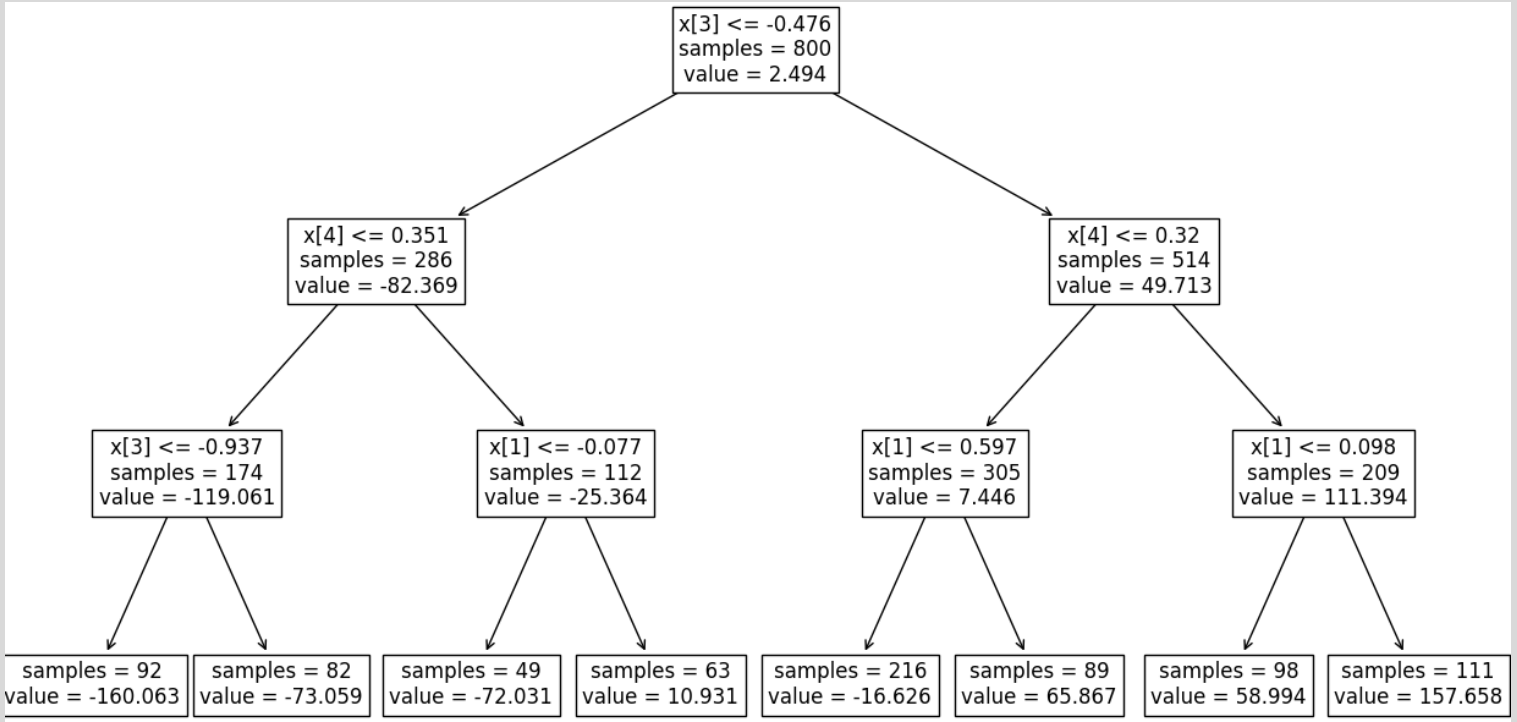
Outline

- Decision tree terminology
- Decision trees for classification
- Tree ensembles

Decision tree terminology

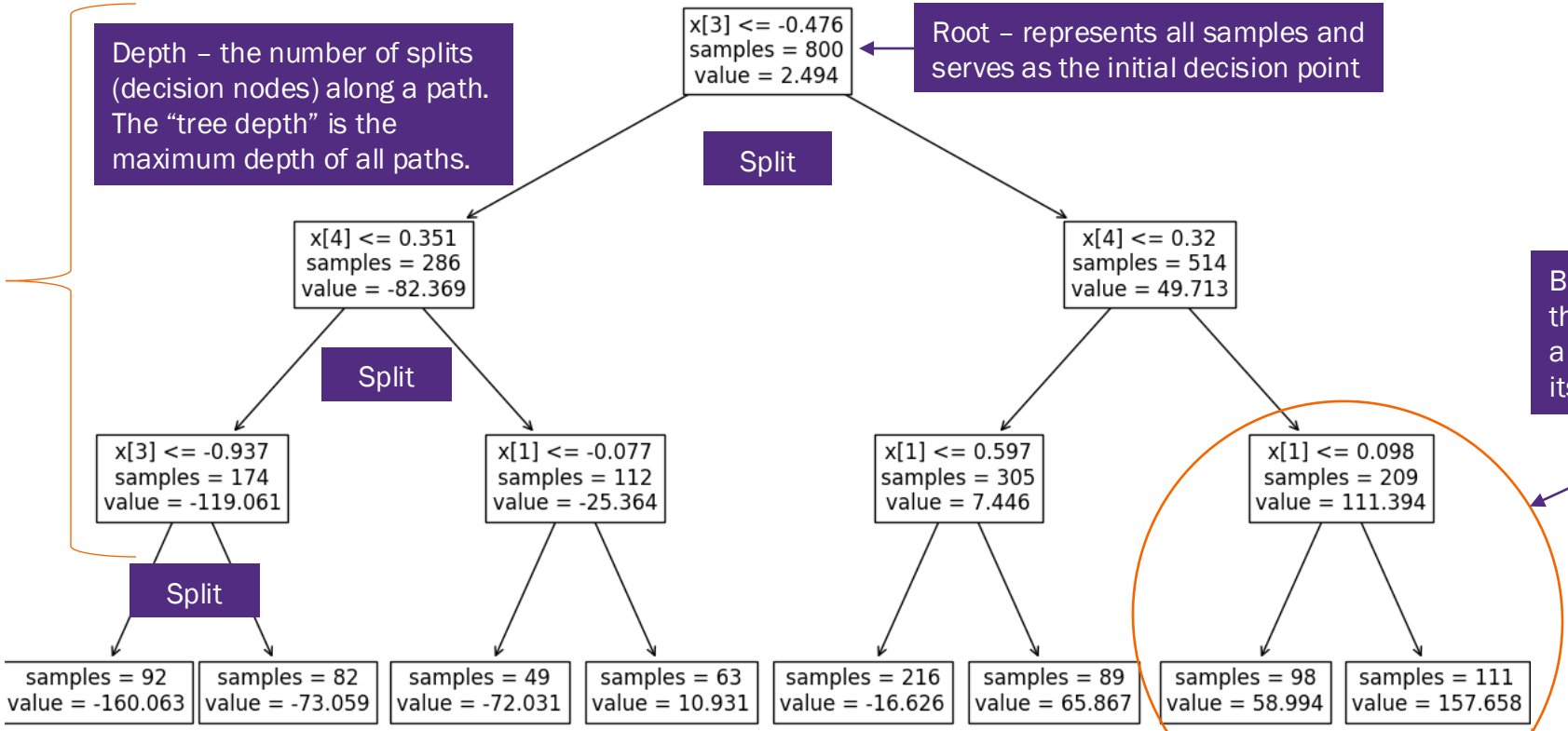


What is a decision tree?



- Directed acyclic graph with a single root node
- Can be read as a flow-chart where a dichotomous test on a single attribute is made at each node
- Path is followed until reaching a terminal node
- Can be used for regression and classification

Decision tree terminology



Root – represents all samples and serves as the initial decision point

Split

Depth – the number of splits (decision nodes) along a path. The "tree depth" is the maximum depth of all paths.

Decision nodes - any node that splits the samples into two subsets

Split

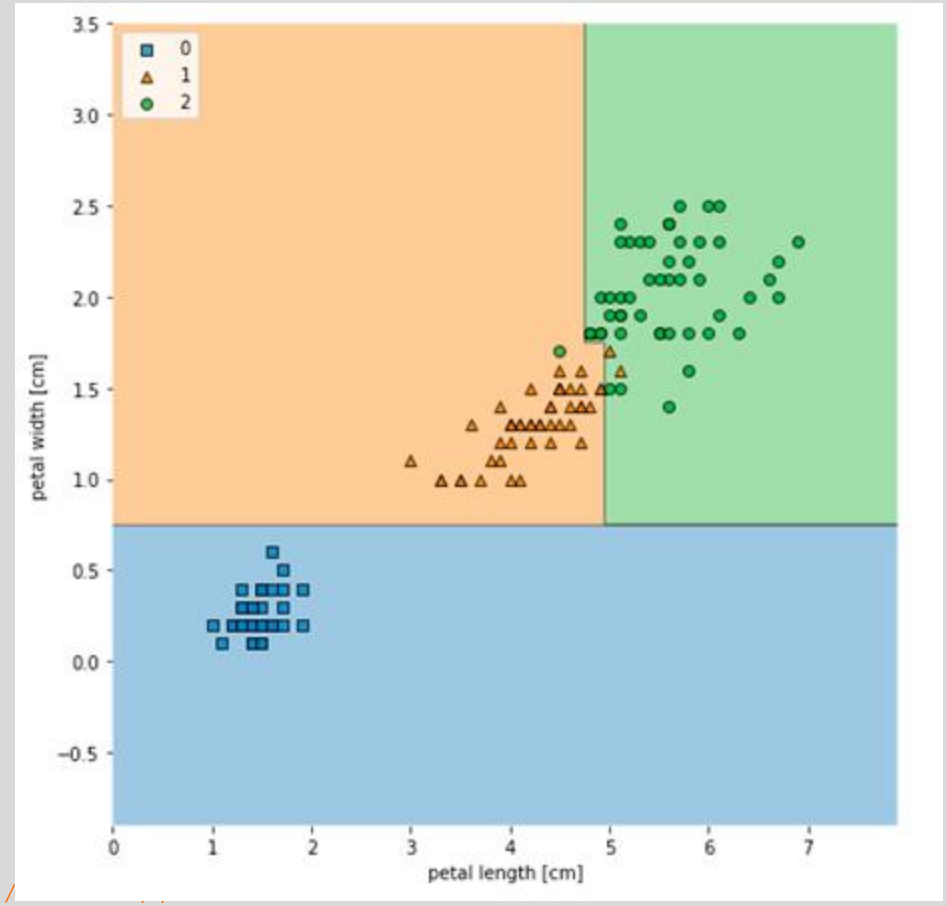
Split

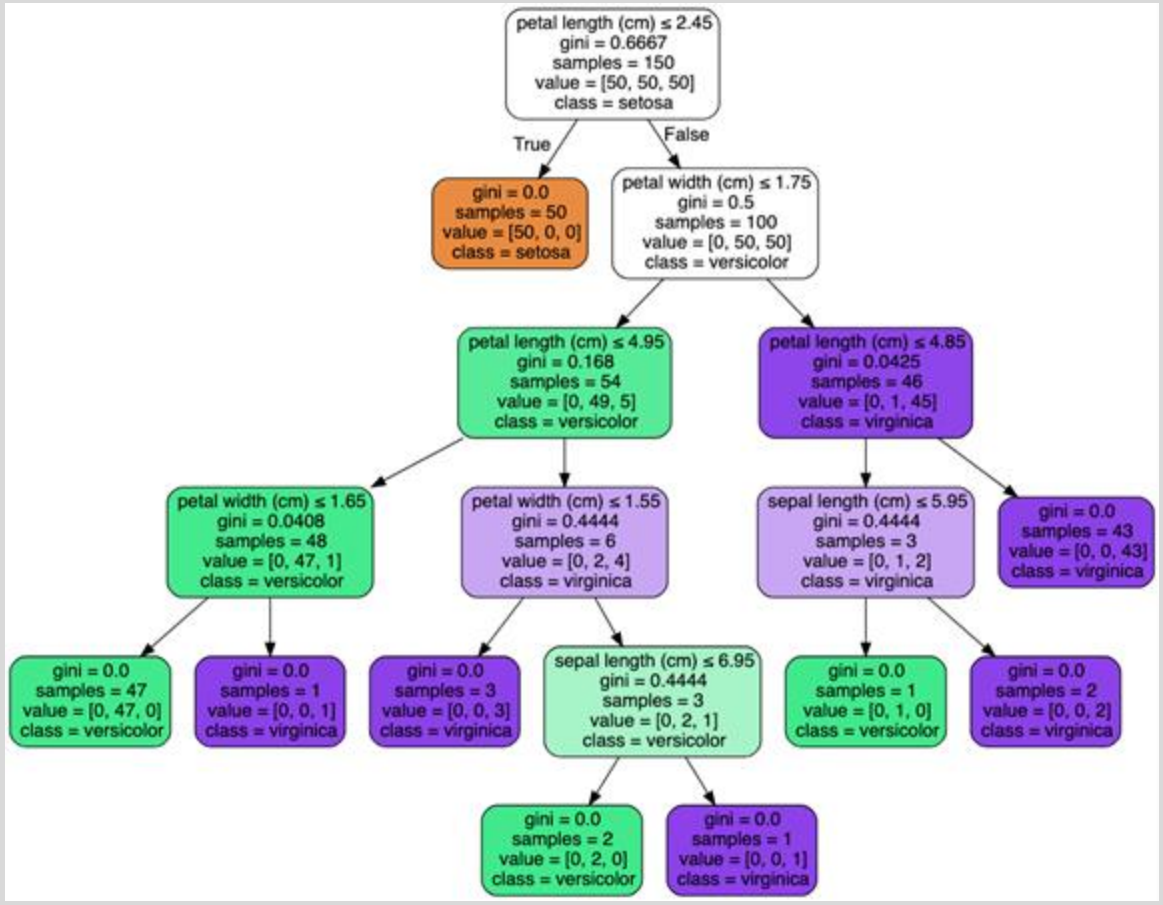
Branch – a subsection of the tree that encompasses a node (branch root) and all its descendant nodes

Leaf - any node that doesn't split. Provides the sample outcome for the path leading to that node. In regression, the outcome is typically the mean of the training samples at the leaf.

Decision Tree

Decision trees build complex decision boundaries by dividing the feature space into rectangles.





Decision tree advantages and disadvantages

Advantages

- Can model non-linear and non-additive relationships
- Easy to interpret / explain
- May mirror human decision-making
- Small trees are easy to display graphically (aids communication)
- No need for dummy variables

Disadvantages

- Prone to high variance (overfitting) – small changes in training sample can lead to large changes in resulting tree
- Poor prediction performance relative to other methods (addressable w/ ensembles)
- Not well suited for complex feature interactions

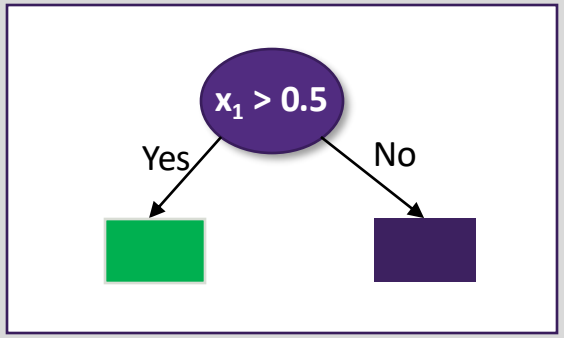
School of COMPUTING
Clemson University

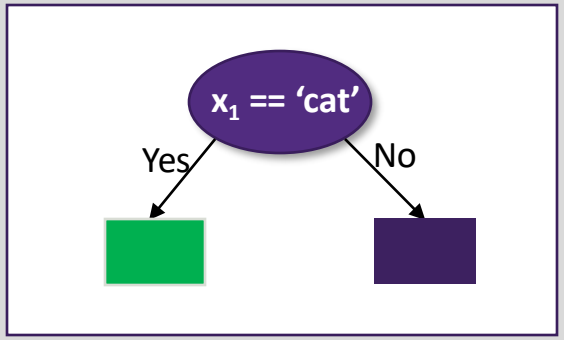
Decision trees for classification

Decision tree learning for classification

- Iterative two step process
  - Select an attribute and test to split the data into two regions
  - For all samples in each region, assign the majority class of those samples as the target class
- Repeat the process until some stopping criteria is met
- Most popular methods:
  - (CART) Classification and Regression Trees
  - C5.0 (update of C4.5 that is an extension of ID3) – proprietary license

Decision tree learning for classification





Example: x_1 is tested. All samples for which $x_1 > 0.5$ will follow the path to the left. All others will go to the right.

When splitting on categorical features, all samples that match the test value go to the left, and all other samples go to right.

The estimate for the target, *y*, for all samples in the green box ($x_1 > 0.5$) is set to the majority class of y for those samples

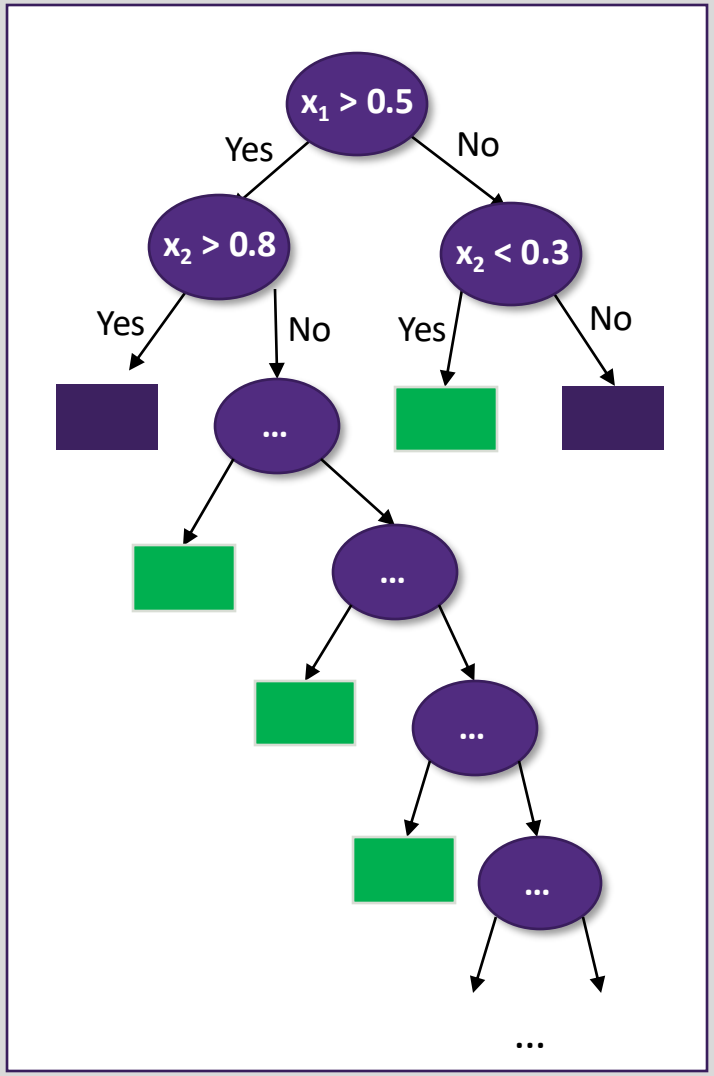
The estimate for the target, y, for all samples in the purple box ($x_1 \leq 0.5$) is set to the majority class of y for those samples

Decision tree learning for classification

- How do we select features and cut points for splitting?
- We will apply *recursive binary splitting* to iteratively select a predictor, X_J, and the cut point, s, that most reduces the objective function
- In regression, the objective function is the sum of the RSS in each region created $R_1, R_2, \cdots, R_J$ by the splits

$$RSS = \sum_{j=1}^{J} \sum_{i \in R_j} \left(y_i - \bar{y}_{R_j}\right)^2$$

- For classification, we typically use the *Gini index* or *entropy* as the objective



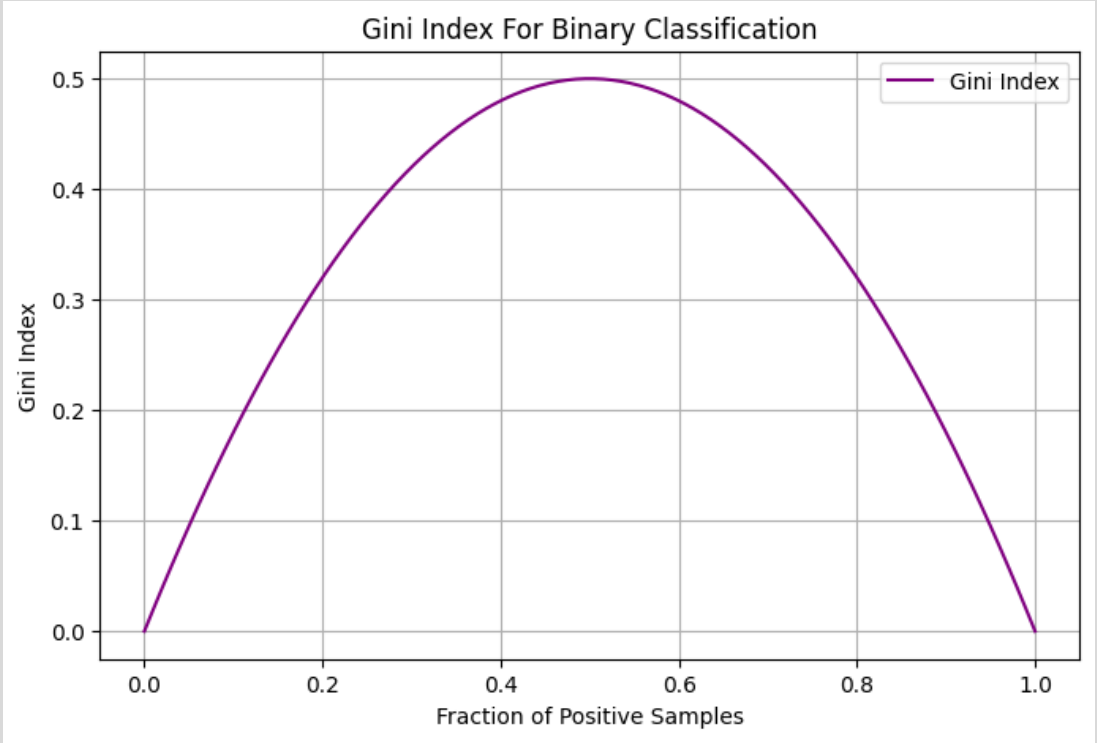
Gini index

- Let there be K classes as possible outcomes and M sample observations, then *Gini index*, G, is given by

$$G = \sum_{k=1}^{K} \widehat{p_k}(1 - \widehat{p_k})$$

where $\widehat{p_k} = {}^{m_k}/_{M}$ and m_k are the fraction and count of observations in class $k \in K$, respectively

- For split feature and cut point selection, we iterate over the features and observed feature values (the possible cut points) to minimize the sum of the *Gini index* for the two child nodes



Node purity – The *Gini index* approaches zero if all observed samples are in exactly one class. In decision trees, this is an indication of the *purity* of the node.

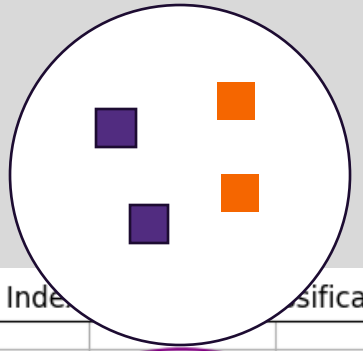
Gini index

- Negative sample k=0
- Positive sample k =1

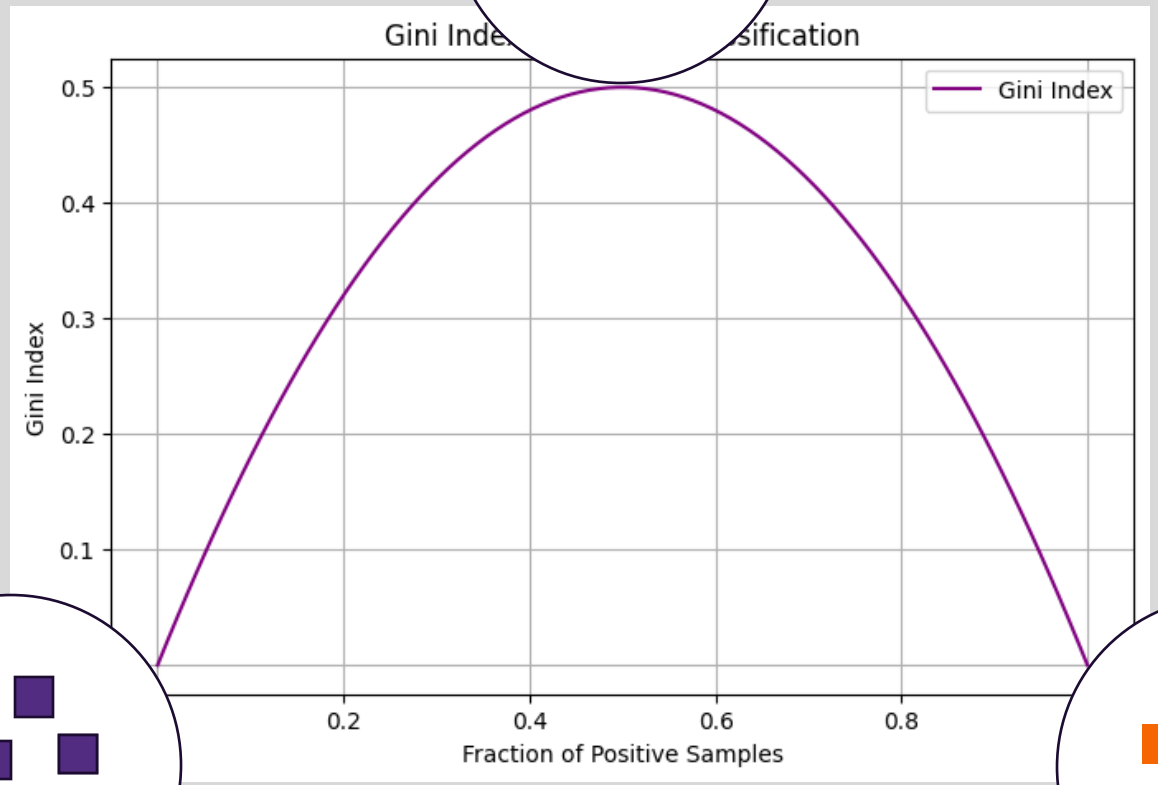
$$G = \sum_{k=1}^{K} \widehat{p_k}(1 - \widehat{p_k})$$

$$\widehat{p_k} = {}^{m_k}/_{M}$$

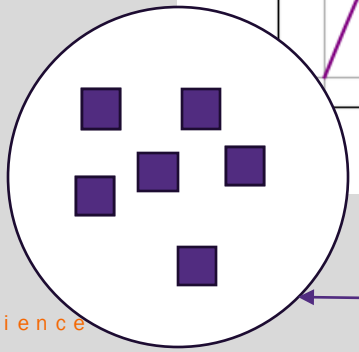
Balanced Node



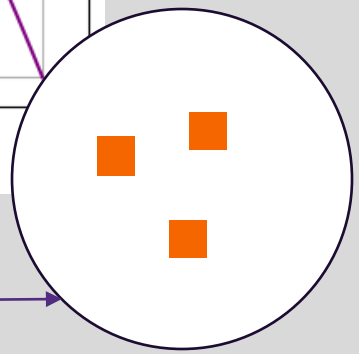
$$M = 4$$
$$m_0 = 2, m_1 = 2$$
$$\widehat{p_0} = {}^1/_2, \widehat{p_1} = {}^1/_2$$
$$G = {}^1/_2(1 - {}^1/_2) + {}^1/_2(1 - {}^1/_2) = {}^1/_2$$



$$M = 6$$
$$m_0 = 6, m_1 = 0$$
$$\widehat{p_0} = 1, \widehat{p_1} = 0$$
$$G = 1(1 - 1) + 0(1 - 0) = 0$$



$$M = 3$$
$$m_0 = 0, m_1 = 3$$
$$\widehat{p_0} = 0, \widehat{p_1} = 1$$
$$G = 0(1 - 0) + 1(1 - 1) = 0$$



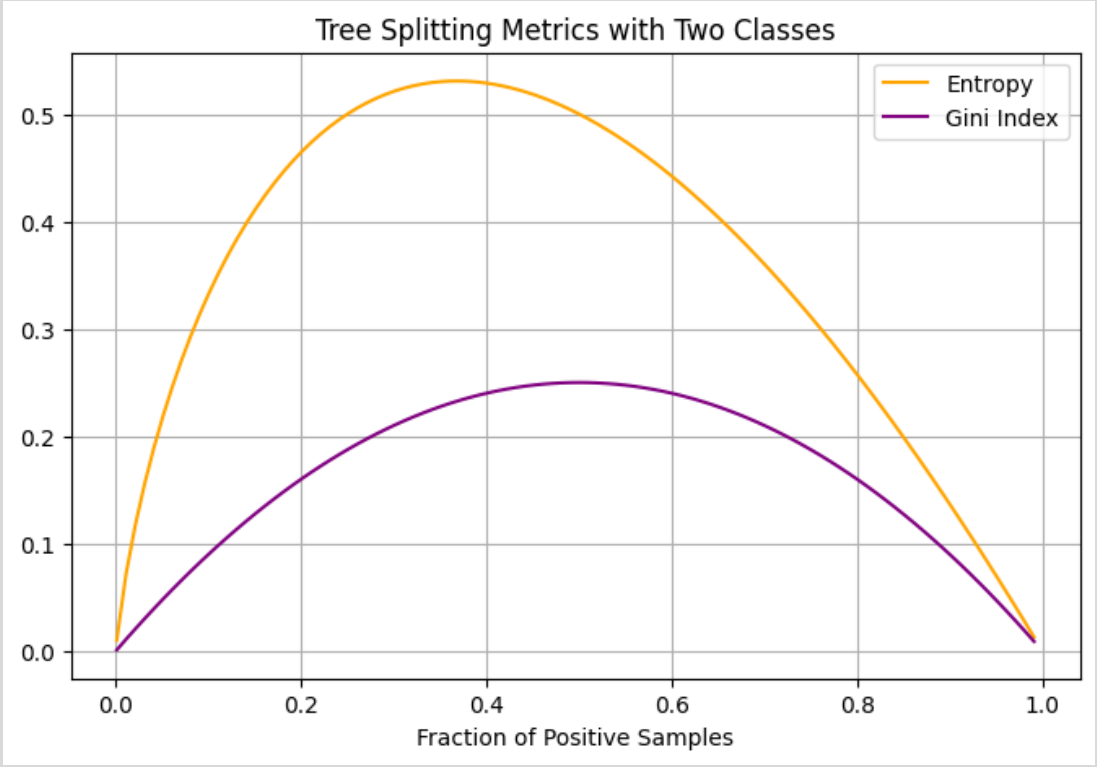
Pure Nodes

Entropy

- As with the *Gini Index*, let there be K classes and M sample observations, then *entropy*, H, is given by

$$H = -\sum_{k=1}^{K} \widehat{p_k} \log_2 \widehat{p_k}$$

where $\widehat{p_k} = {}^{m_k}/_{M}$ and m_k are the fraction and count of observations in class $k \in K$, respectively

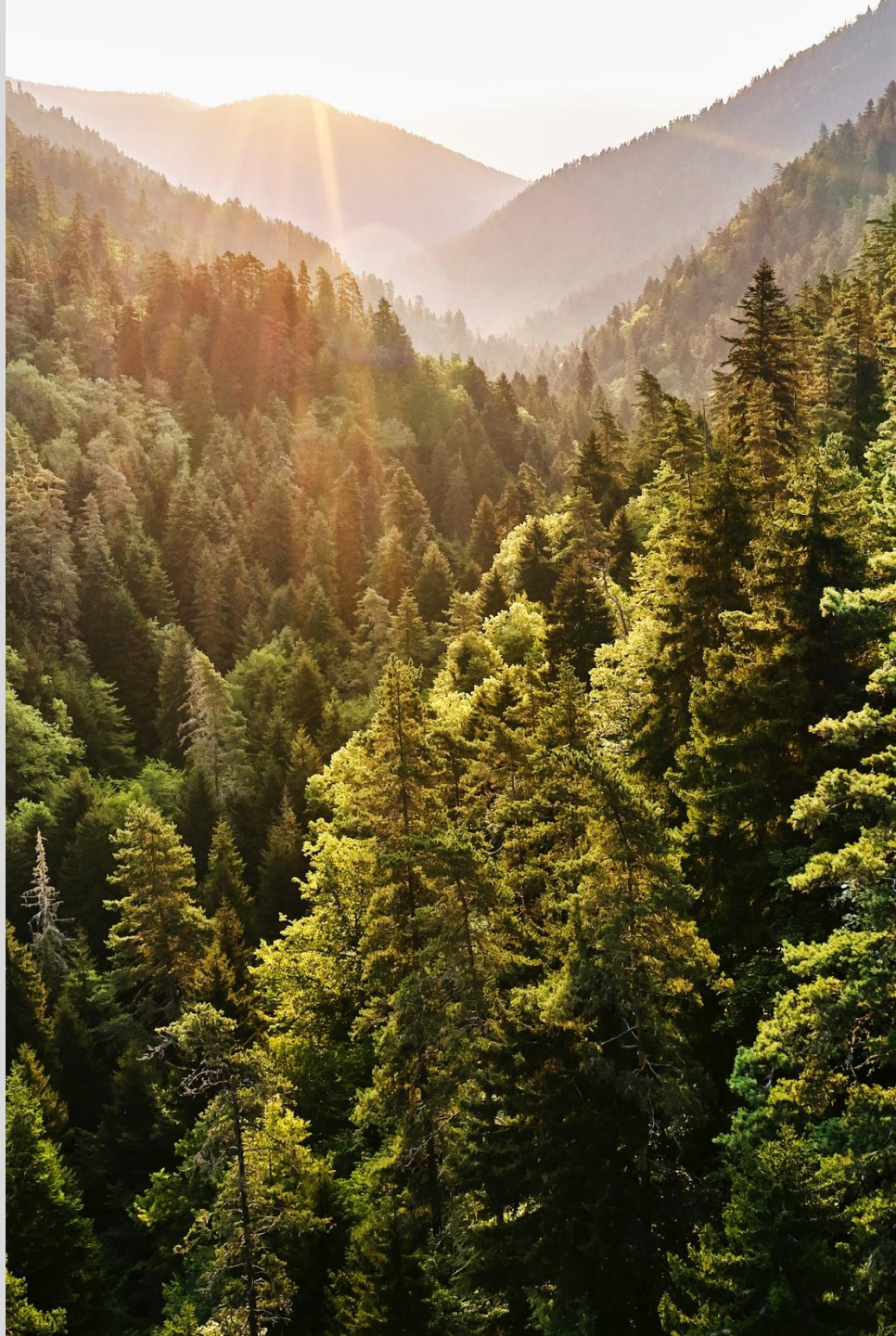


School of COMPUTING
Clemson University

Ensembles of Trees

Decision tree ensembles

- Ensemble – in the context of machine learning, an ensemble is a collection of "simpler" models
- Decision tree ensemble is a collection of decision trees
- The ensemble model prediction is a function of the individual model outputs (e.g., the mode for classification)
- Why use ensembles?
  - Reduces variance (overfitting) even if the individual models are overfit
  - Methods can be highly parallelizable (e.g., bagging and Random Forest)
- What's the catch? Mainly, we lose interpretability



Bootstrap aggregation (bagging)

- Given *n* training samples:
  1. Randomly sample with replacement *m* samples from *n* (typically m=n)
  2. Generate a model using the random sample
  3. Repeat steps 1 and 2 *B* times
- Yields B models from which the final output is produced (e.g., by averaging)
- Model variance (overfitting) is NOT dependent on B – so creating many trees won't lead to overfitting

Bagging Decision Trees

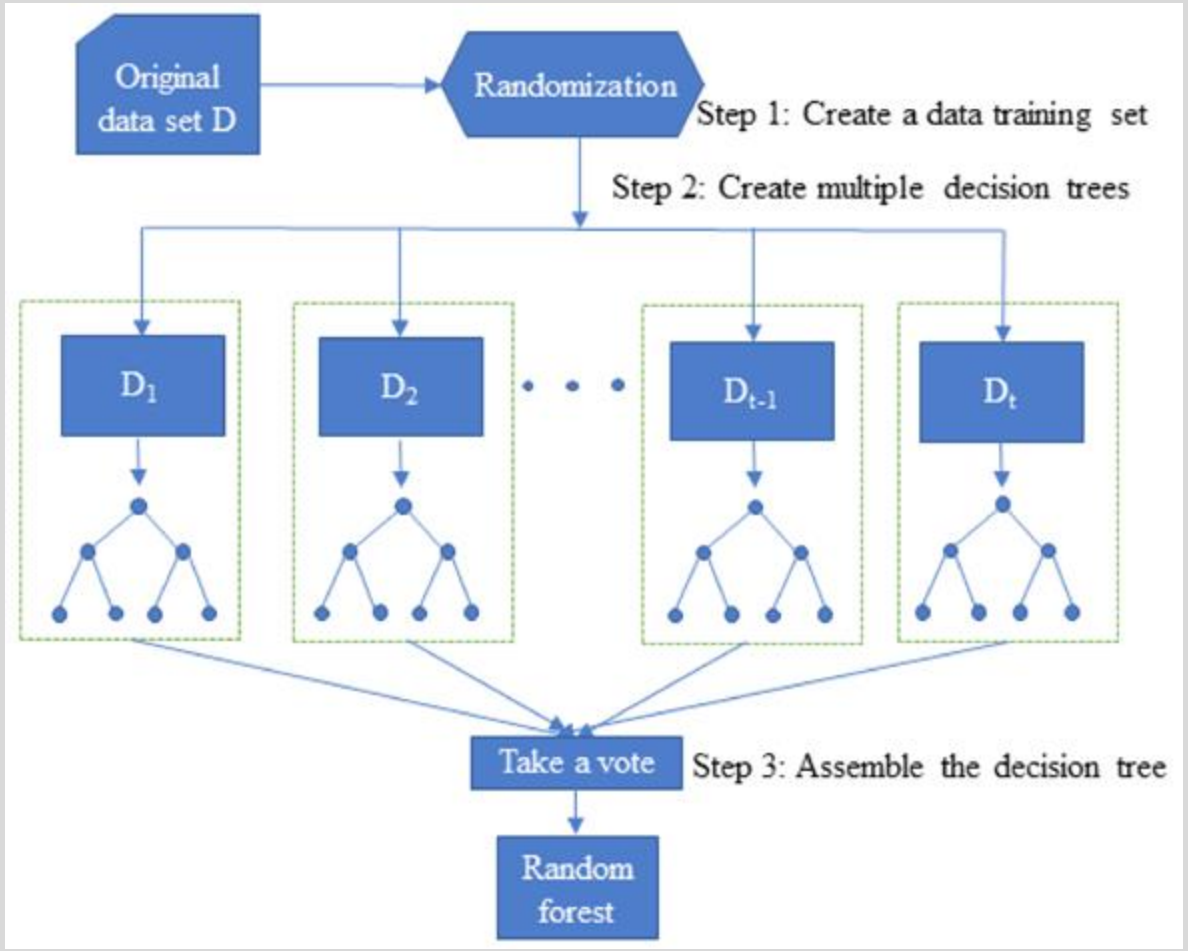
- How can bagging be used with decision trees for regression?

1. Construct *B* bootstrapped datasets from available training data
2. For each bootstrapped dataset, construct a deep, unpruned tree
3. For a given test sample, the final output is the average of the *B* trees for that sample

ISSUE: trees are highly correlated

Random forests

- Extension of bagging that decorrelates the trees
- Follows the same procedure as bagging, except that for each split, only $m < p$ predictors are considered
  - Limits which predictors are considered at each split (including initial split) so that no features can dominate all trees
  - Typical choice $m = \sqrt{p}$
- As with normal bagging, the number of trees (bootstrapped samples) does not affect variance



Boosting trees – conceptual description

- Similar to bagging in that multiple trees are created
- No bootstrapping
- Instead, we fit a sequence of small trees to the residuals $r_i = y_i - \hat{y}_i$ (initially all $r_i = y_i$)
- At each iteration, b, the current model, $\hat{f}(x)$ is updated with tree fitted to the current residuals $\hat{f}^b(x)$ by:

$$\hat{f}(x) \leftarrow \hat{f}(x) + \lambda \hat{f}^b(x)$$

- The residuals are update by:

$$r_i \leftarrow r_i - \lambda \hat{f}^b(x_i)$$

- The final model is

$$\hat{f}(x) = \sum_{b=1}^{B} \lambda \hat{f}^b(x)$$

Each new tree is constrained to a shallow depth based on the tuning parameter, *d*. Typically $1 \leq d \leq 3$

λ is a *shrinkage* parameter that controls the learning rate. Typical values are in [0.001, 0.01]

Unlike bagging, boosting variance is sensitive to the value of B. Creating too many trees can lead to overfitting.

Boosting trees – implementations

Gradient Boosted Trees (in scikit-learn)

XGBoost – essentially boosting with regularization

CatBoost – grows symmetric trees, likely to achieve the highest performance in datasets with many categorical variables

Questions?