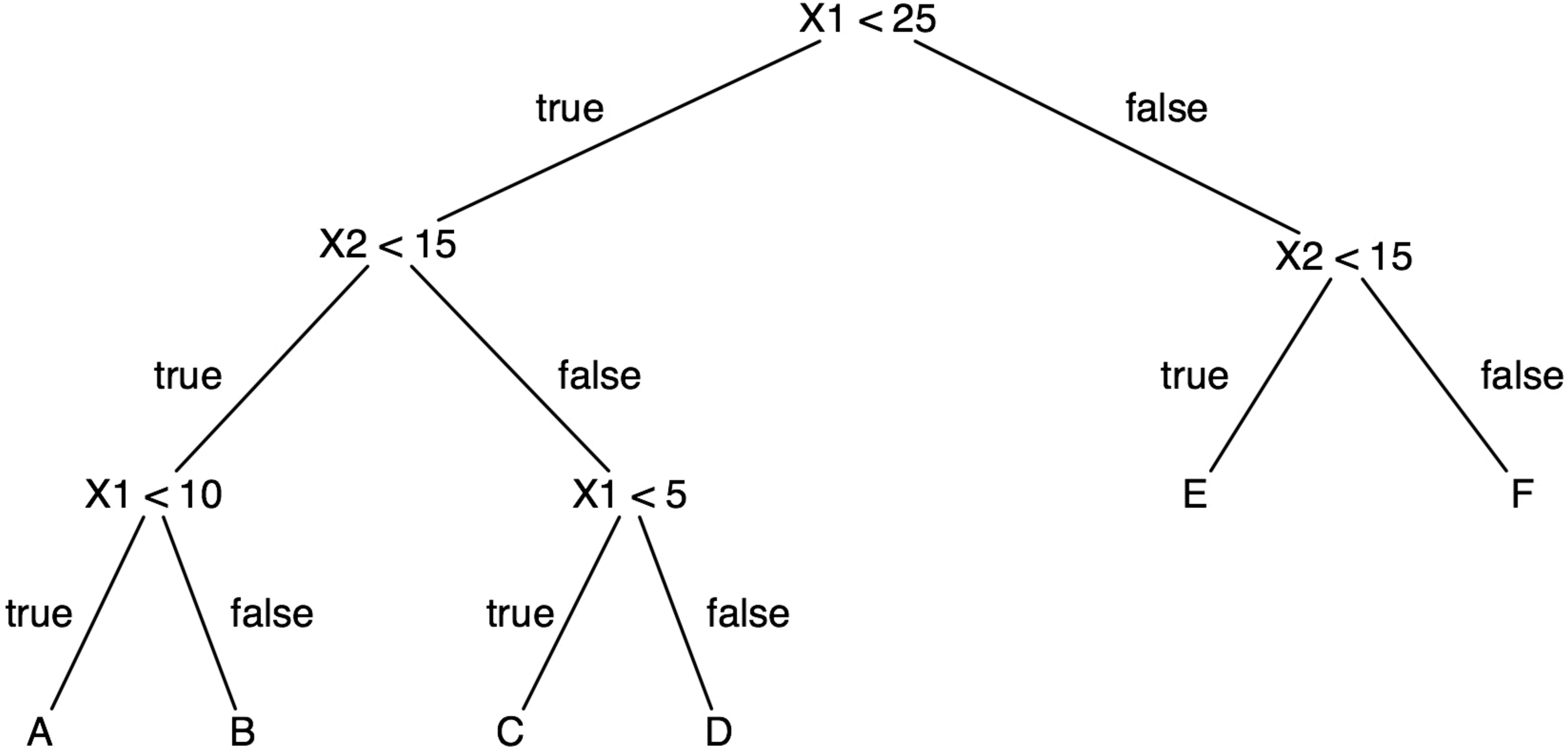
Decision Trees and Random Forests

Dalya Baron (Tel Aviv University)

XXX Winter School, November 2018

Decision Trees



Decision tree: a non-parametric model, constructed during training, which is described by a tree-like graph. It can be used for classification or regression.

Decision Tree Construction

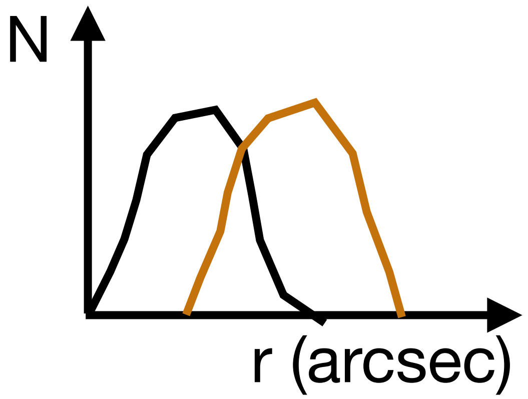
Input training set: a list of objects with measured features and known labels.
Classes: “black” and “brown” galaxies.
Measured features: r (arcsec), B (mag), V(mag).

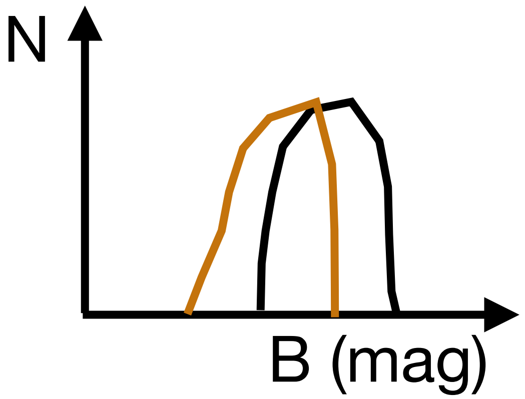
Decision Tree Construction

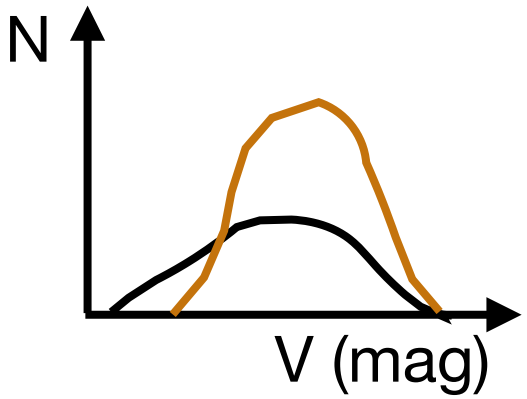
Input training set: a list of objects with measured features and known labels.

Classes: “black” and “brown” galaxies.

Measured features: r (arcsec), B (mag), V(mag).

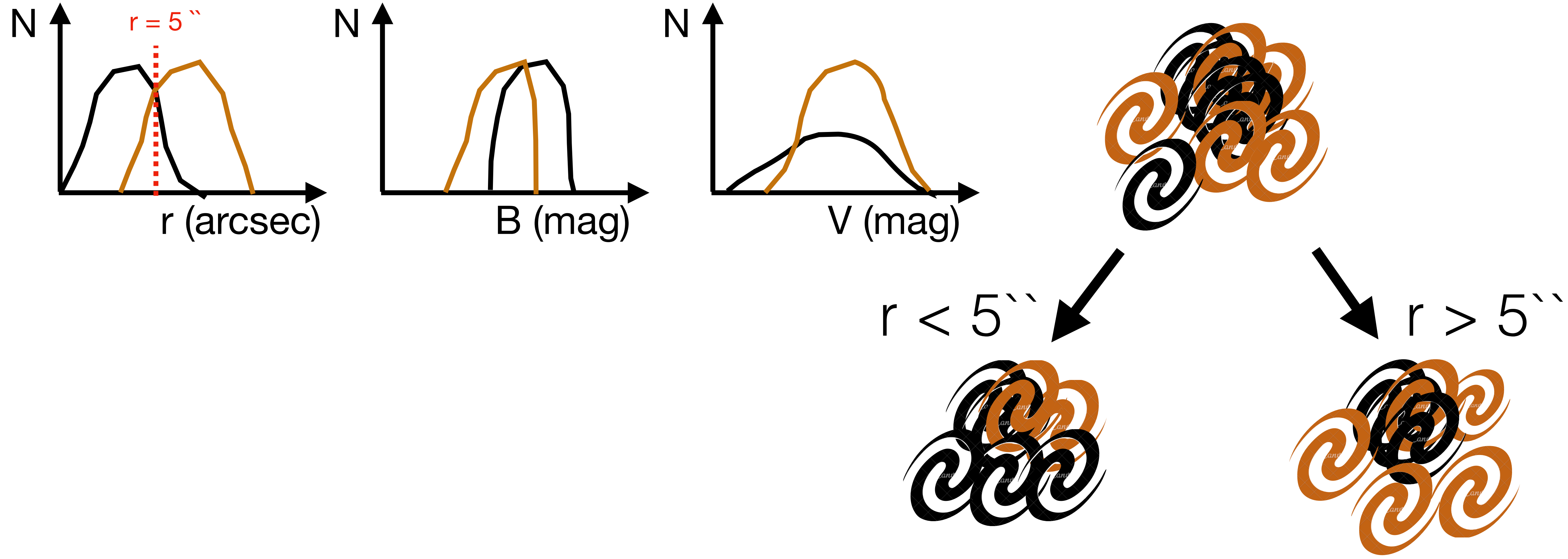






Decision Tree Construction

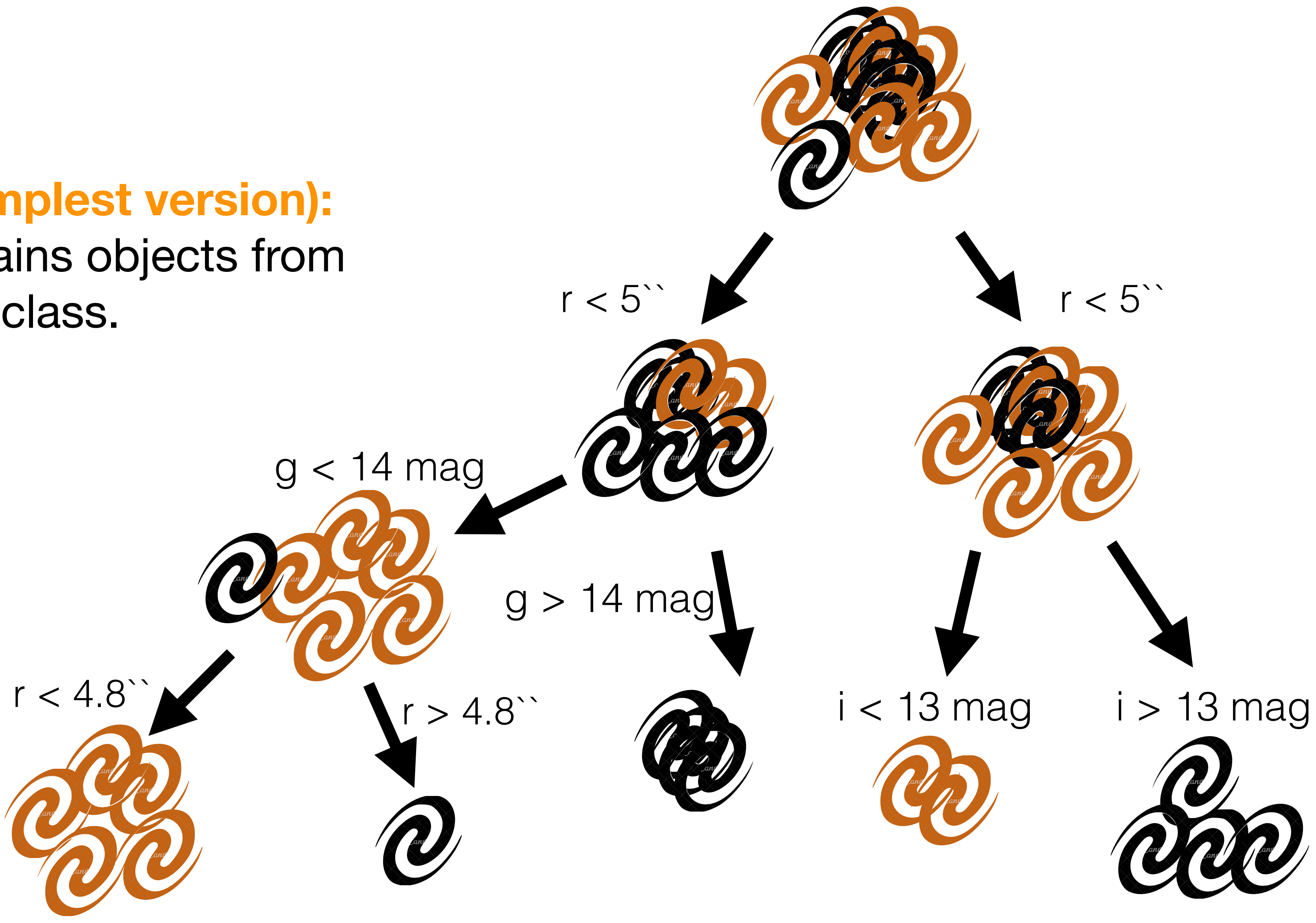
Input training set: a list of objects with measured features and known labels.
Classes: “black” and “brown” galaxies.
Measured features: r (arcsec), B (mag), V(mag).



Decision Tree Construction

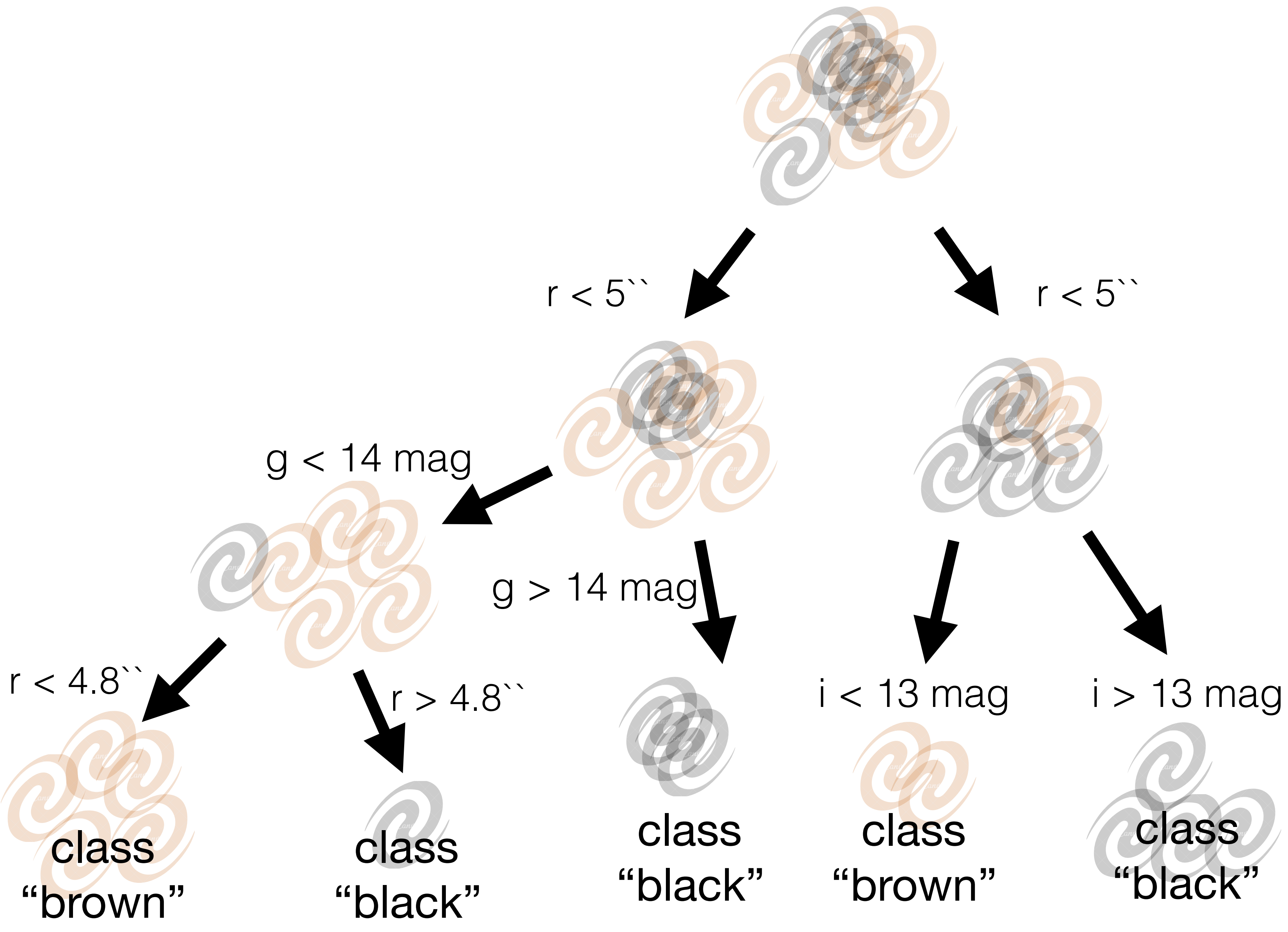
Input training set: a list of objects with measured features and known labels.
Classes: “black” and “brown” galaxies.
Measured features: r (arcsec), g (mag), i(mag).

Stop criterion (simplest version): each terminal contains objects from a single class.



Decision Tree Prediction

Input set: a list of objects with measured features and unknown labels.
Objects are propagated through the tree according to their measured features.



r < 5``

r < 5``

g < 14 mag

g > 14 mag

i < 13 mag

i > 13 mag

r < 4.8``

r > 4.8``

class “brown”

class “black”

class “black”

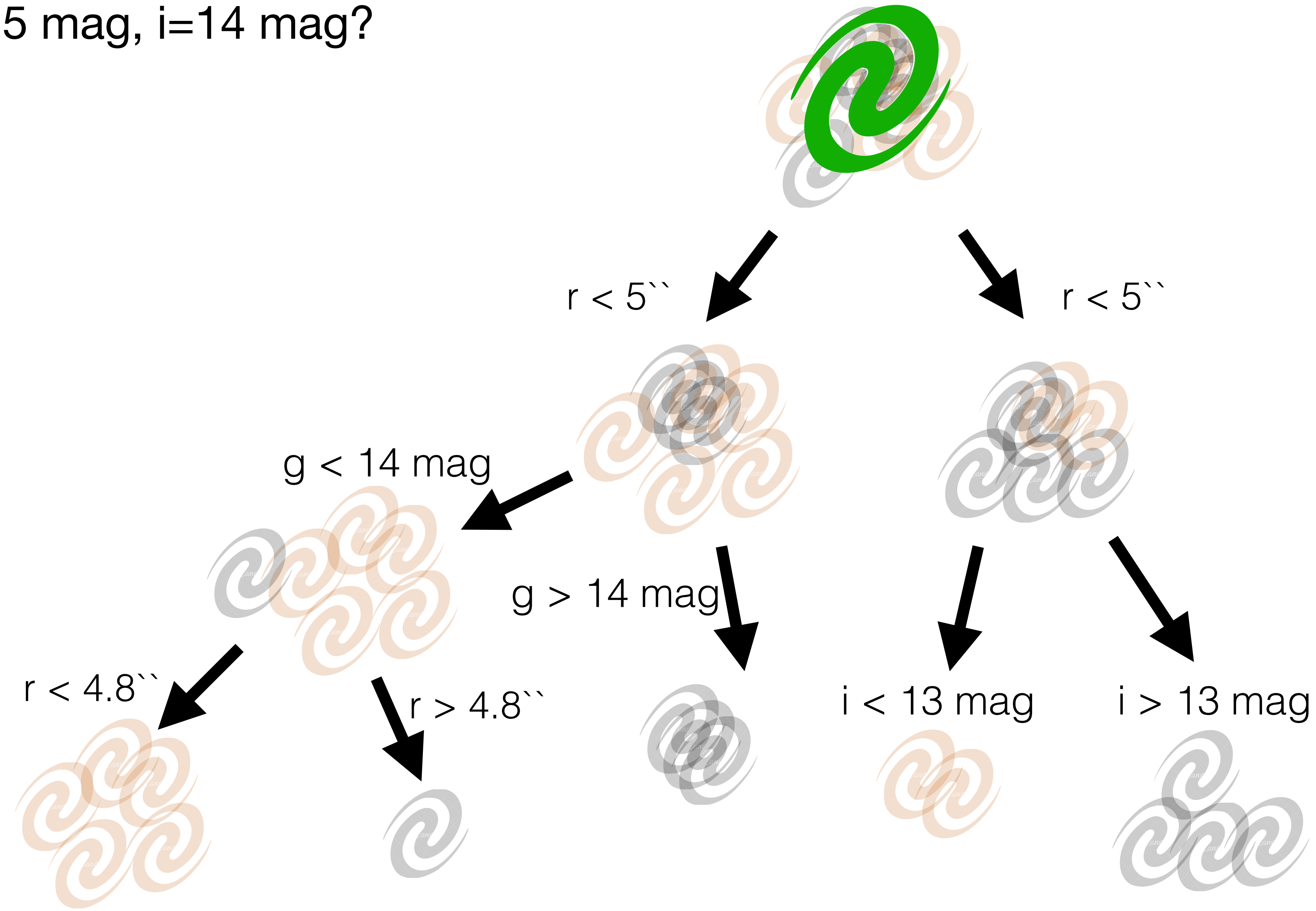
class “brown”

class “black”

Decision Tree Prediction

Input set: a list of objects with measured features and unknown labels.
Objects are propagated through the tree according to their measured features.

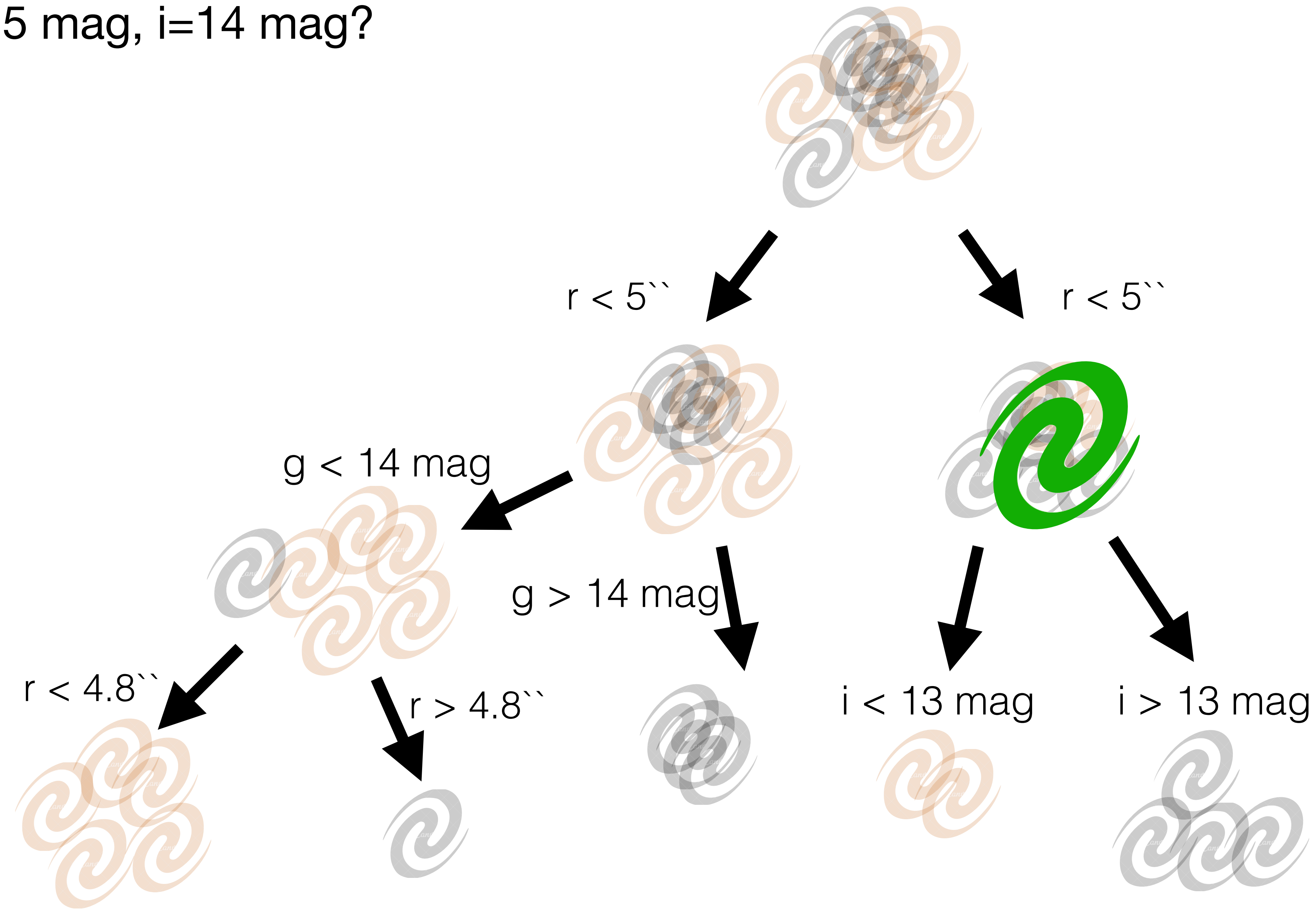
Example: what is the predicted label for a
galaxy with the measured features:
r=3``, g=15 mag, i=14 mag?



Decision Tree Prediction

Input set: a list of objects with measured features and unknown labels. Objects are propagated through the tree according to their measured features.

Example: what is the predicted label for a galaxy with the measured features: r=3``, g=15 mag, i=14 mag?

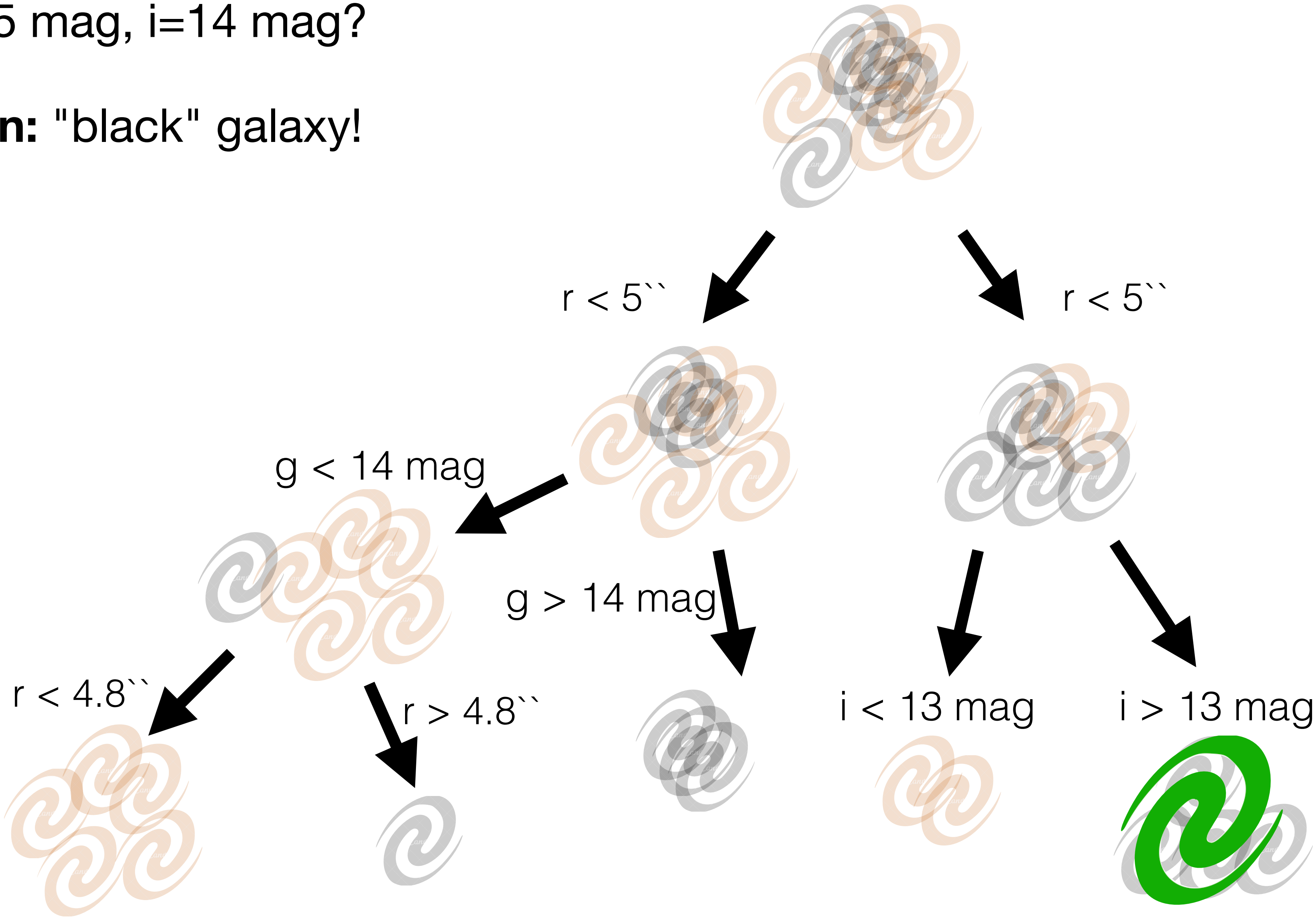


Decision Tree Prediction

Input set: a list of objects with measured features and unknown labels. Objects are propagated through the tree according to their measured features.

Example: what is the predicted label for a galaxy with the measured features: r=3``, g=15 mag, i=14 mag?

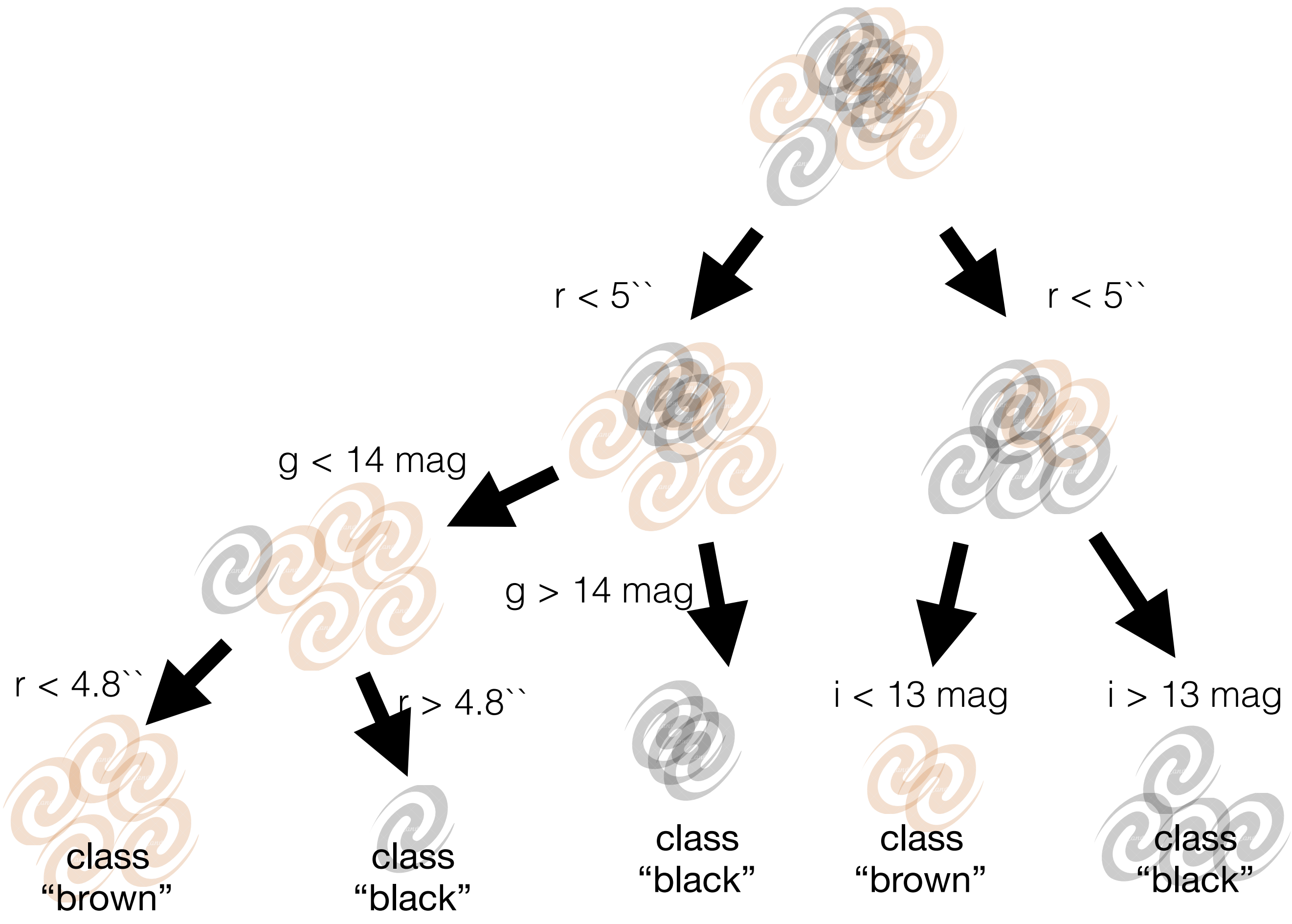
Prediction: "black" galaxy!



Decision Trees: Pros & Cons

Advantages:

(1) Non-linear model, which is constructed during training.
(2) In its simplest version, very few free parameters.
(3) Handles numerous features and numerous objects.
(4) No need to scale the feature values to the same “units”.
(5) Produces classification probability (in its more complex version).
(6) Produces feature importance.



r < 5``

r < 5``

g < 14 mag

g > 14 mag

r < 4.8``

r > 4.8``

i < 13 mag

i > 13 mag

class “brown”

class “black”

class “black”

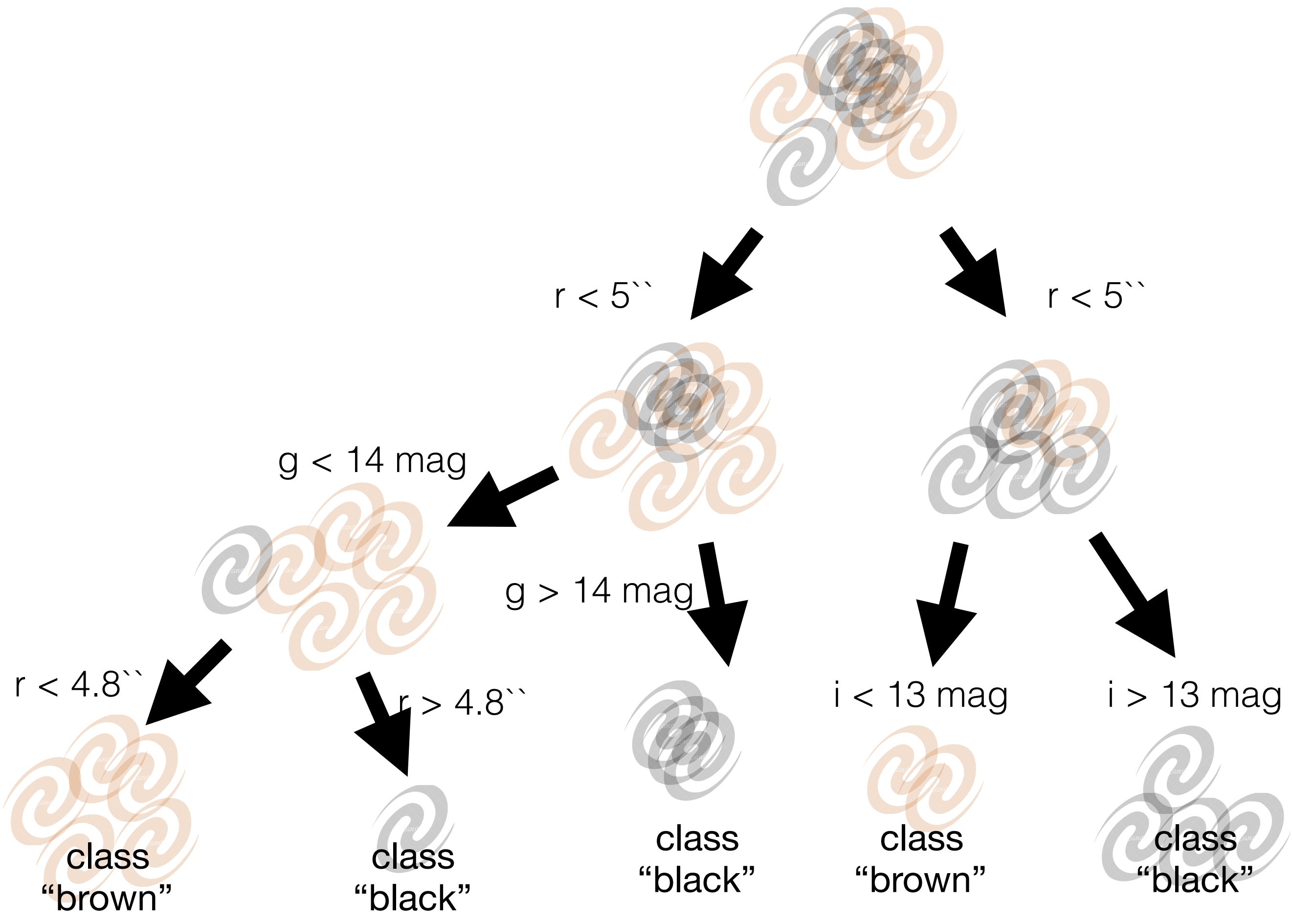
class “brown”

class “black”

Decision Trees: Pros & Cons

Advantages:

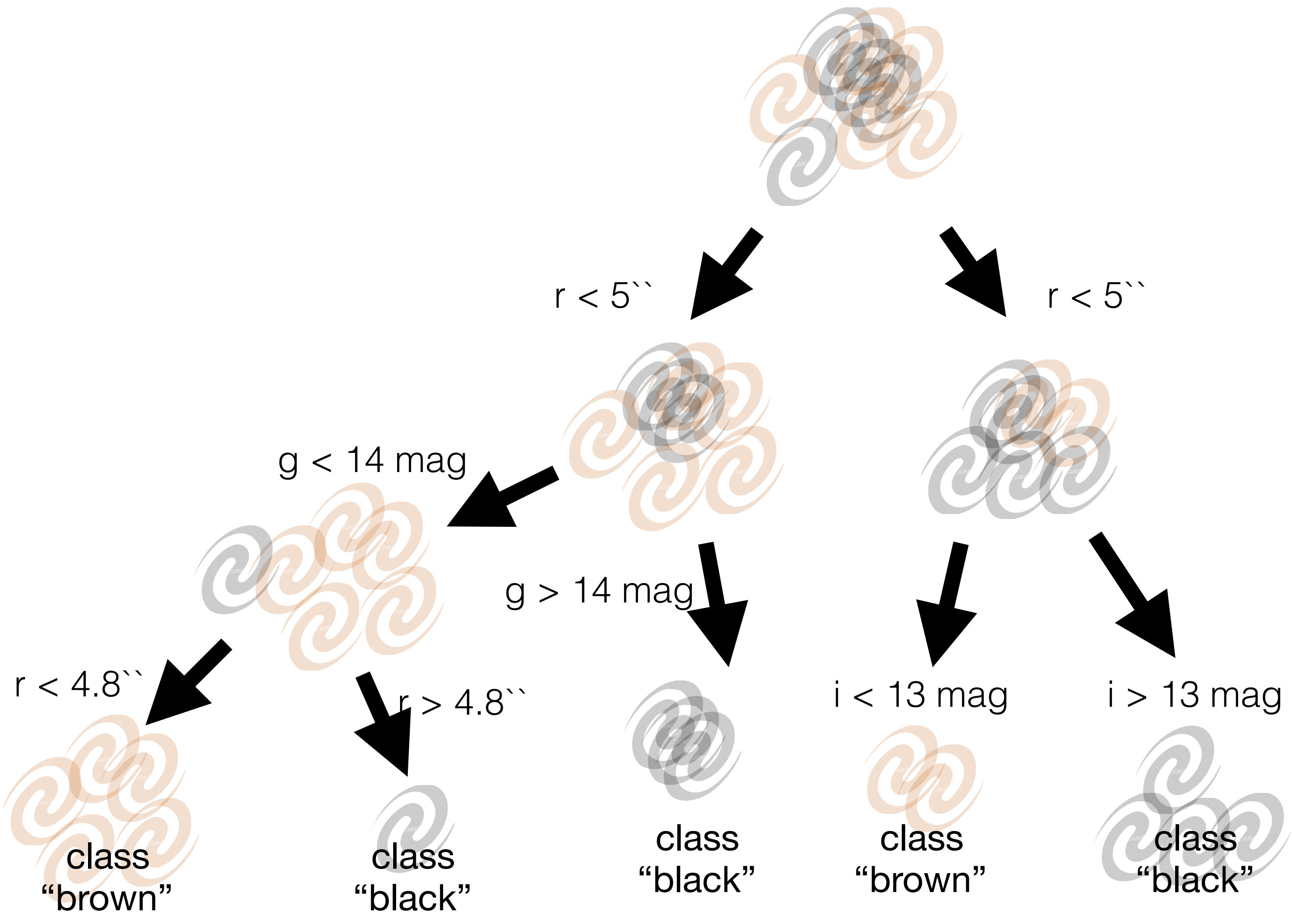
(1) Non-linear model, which is constructed during training.
(2) In its simplest version, very few free parameters.
(3) Handles numerous features and numerous objects.
(4) No need to scale the feature values to the same “units”.
(5) Produces classification probability (in its more complex version).
(6) Produces feature importance.



Feature importance & feature selection

Rule of thumb: the higher a feature is in a decision tree, the more important it is for the classification task. The locations of features within the tree can be used to produce feature importance.
In our example, feature importance: r, i, and then g.

Useful trick: add non-informative features to your dataset (a feature with random values, or a constant feature). If your physical features are ranked less important, remove them!



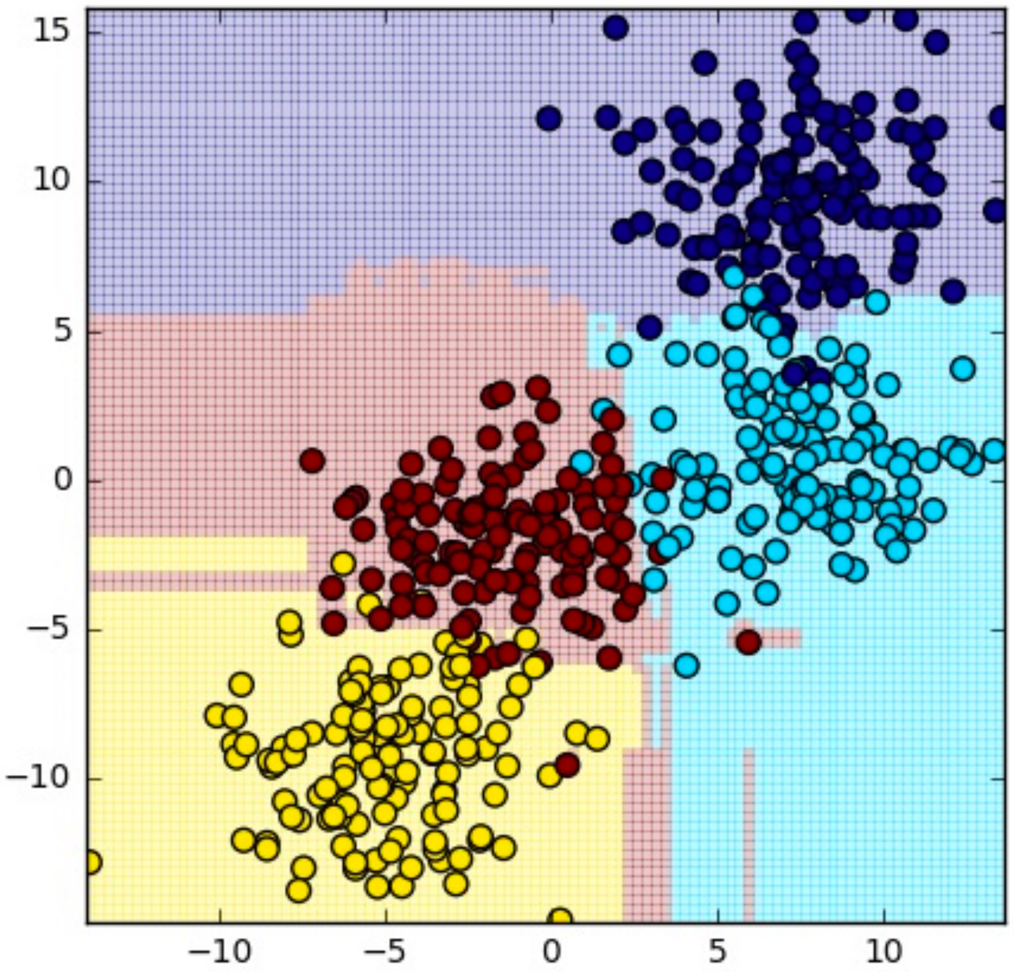
Decision Trees: Pros & Cons

Advantages:

1. Non-linear model, which is constructed during training.
2. In its simplest version, very few free parameters.
3. Handles numerous features and numerous objects.
4. No need to scale the feature values to the same “units”.
5. Produces classification probability (in its more complex version).
6. Produces feature importance.

Disadvantages:

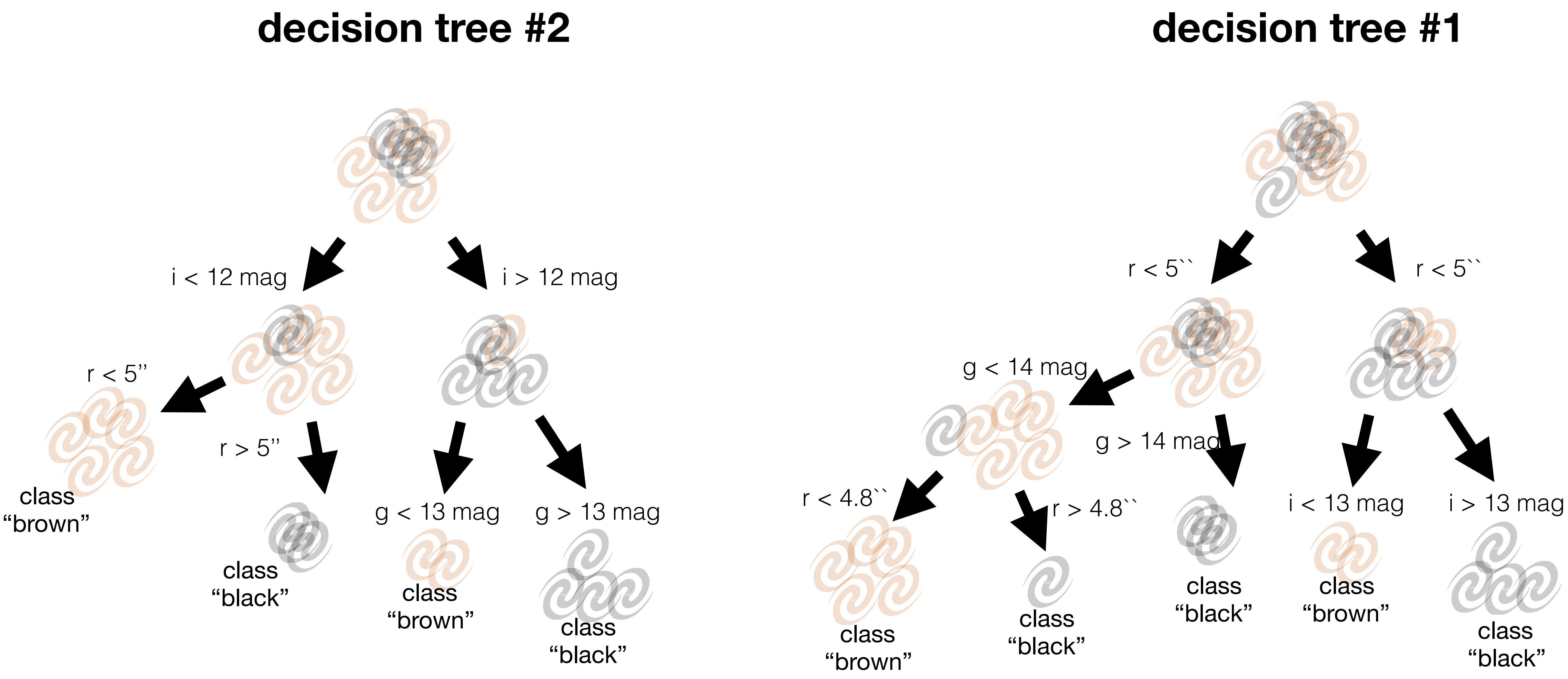
1. Usually does not generalize well to unseen datasets:
    1. Mediocre performance on test set.
    2. Tends to overfit.



Random Forests

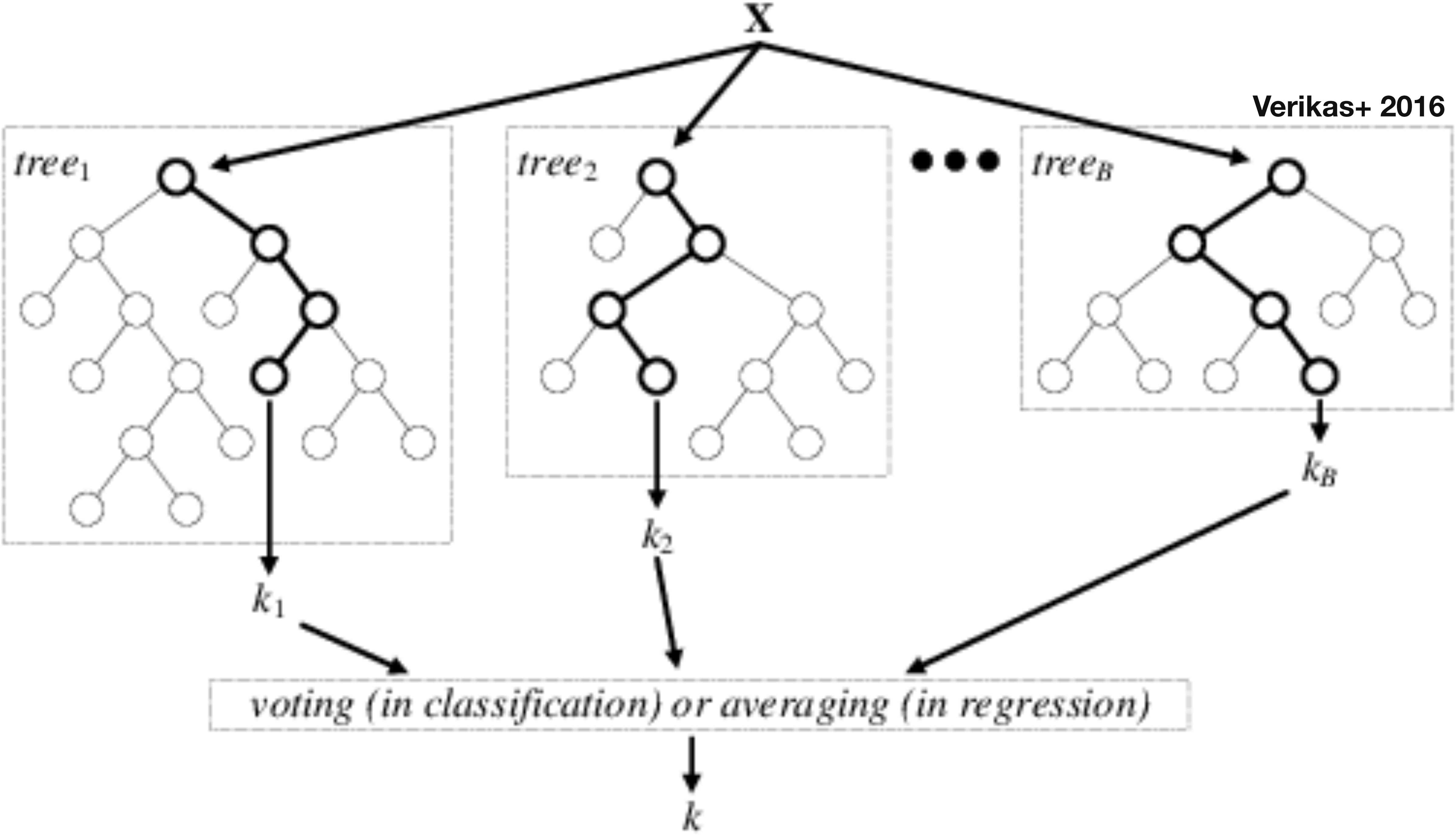
Random Forest is an ensemble of decision trees, where **randomness** is injected into the training process of each individual tree with a **bagging** approach.

Bagging: -The training set is split into randomly-selected subsets, and each decision tree is trained on a subset of the data.
-In each node in the decision tree, only a randomly-selected subset of the feature is considered.



Random Forest Prediction

Verikas+ 2016



Random Forest Prediction

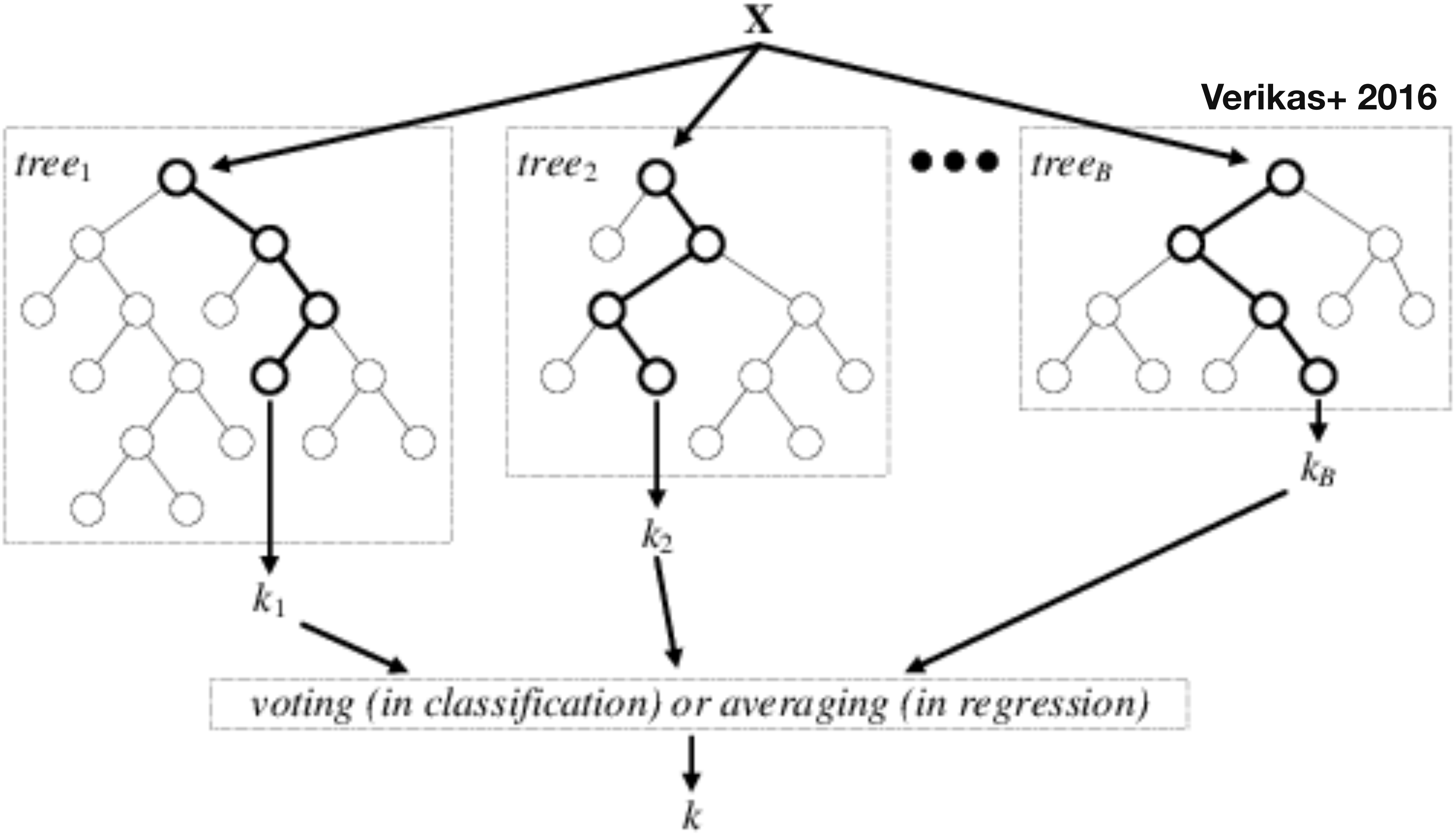
Hyper parameters:

(1) Number of trees in the forest

(2) Number of randomly-selected features to consider in each split.

(3) Splitting criterion (also for Decision Trees).

(4) Class weight.



Verikas+ 2016

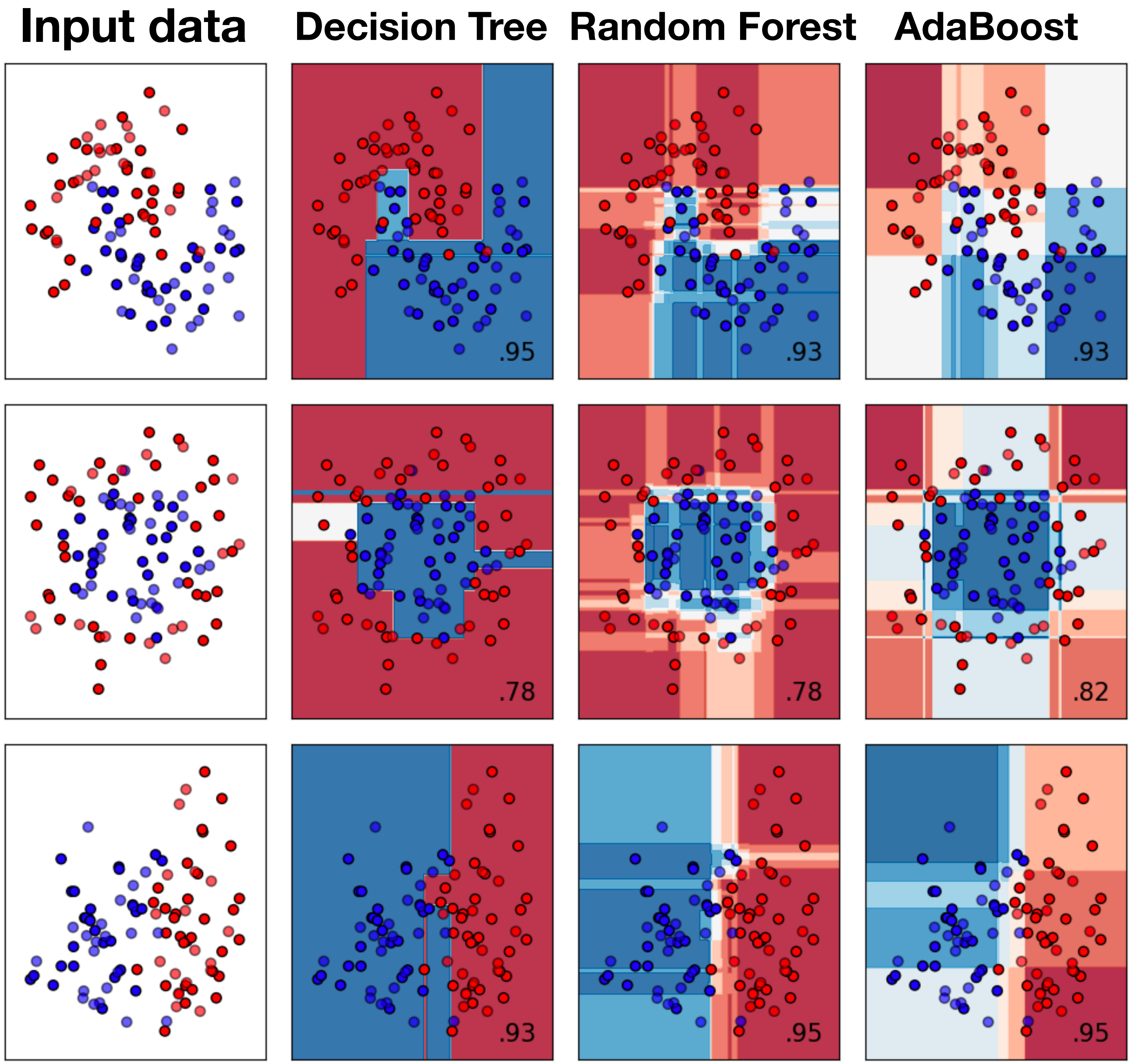
Random Forest: Pros & Cons

Advantages:

1. Same advantages as in a single Decision Tree.
2. Specifically, can handle thousands of features!
3. Generalizes well to unseen datasets.
4. Easily parallelizable.

Disadvantages:

1. Cannot handle measurement uncertainties (true for most ML algorithms!).



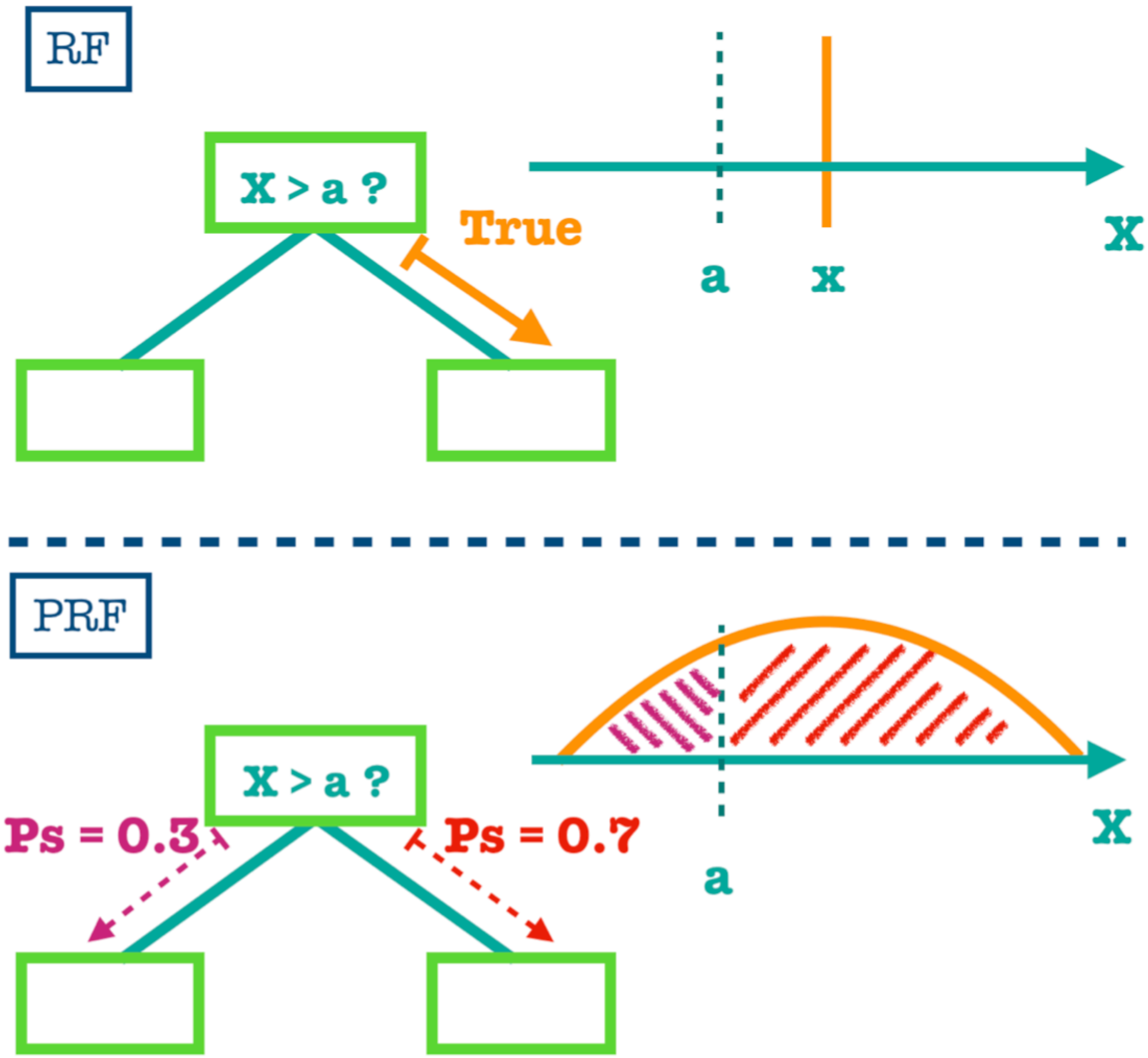
http://scikit-learn.org/stable

Random Forest: Examples

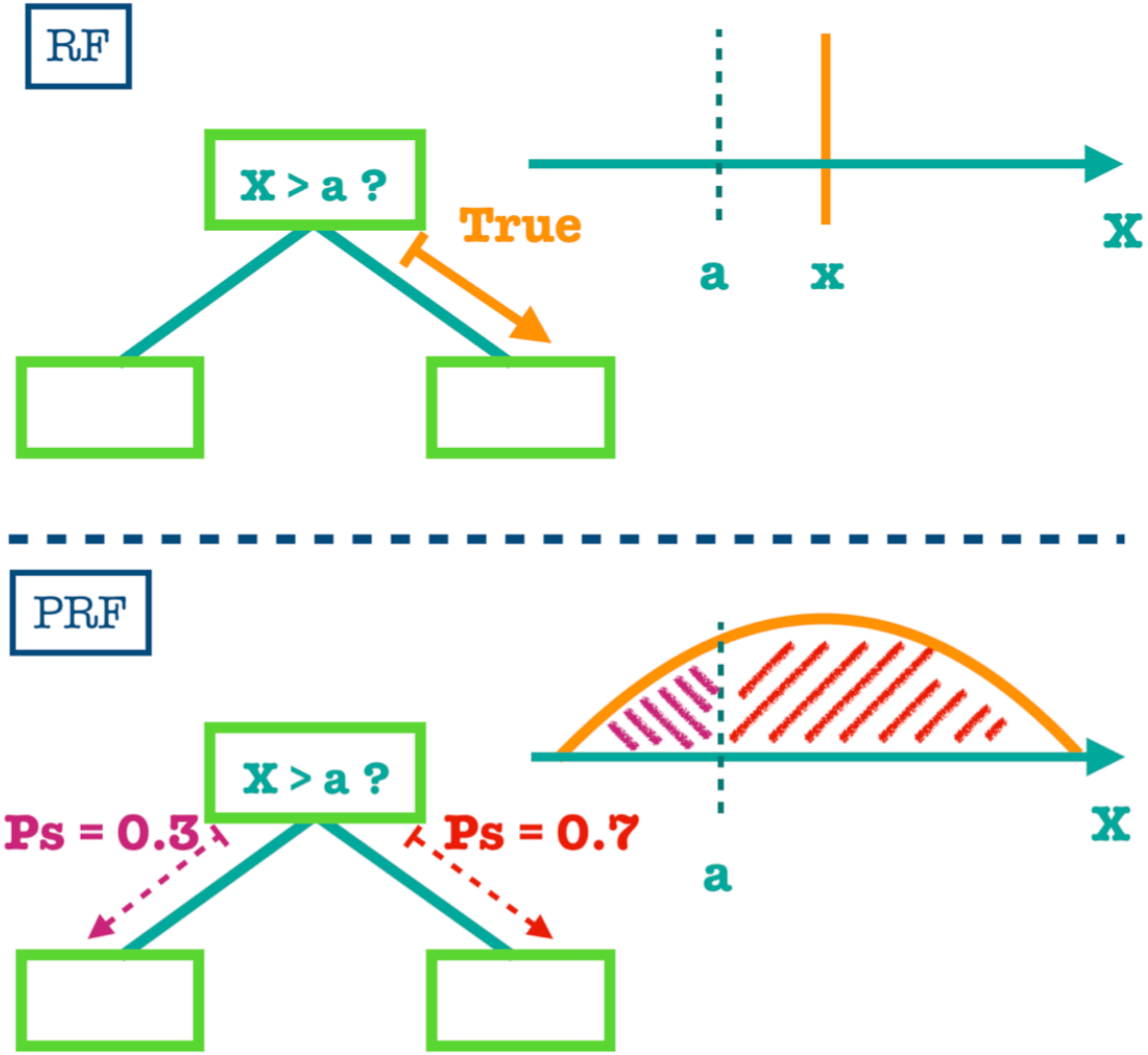
https://cs.stanford.edu/~karpathy/svmjs/demo/demoforest.html

Probabilistic Random Forest

A Random Forest that takes into account the uncertainties in both the features and the input labels. The Probabilistic Random Forest treats all measurements as random variables (see Reis+18).

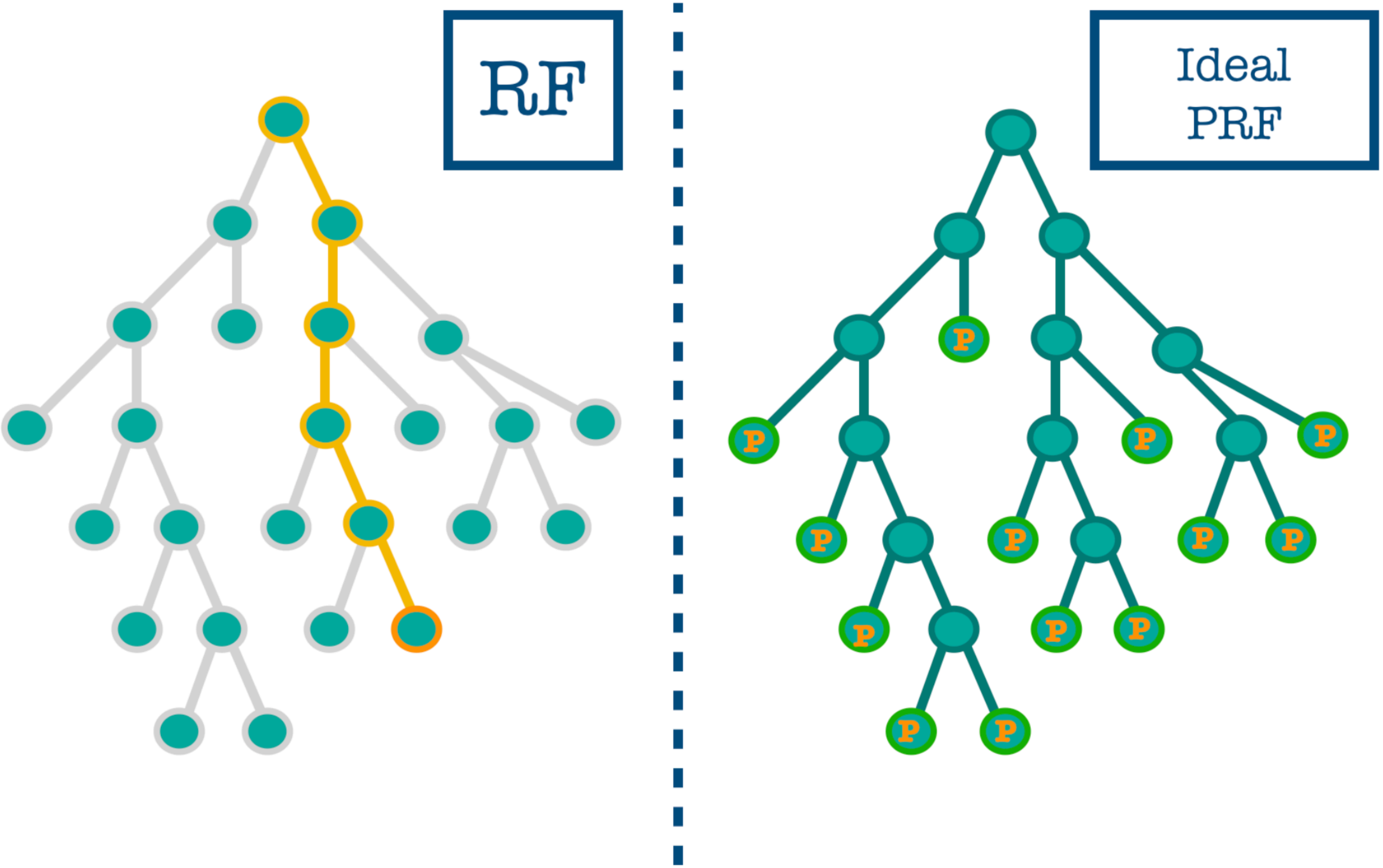


PRF is able to handle a dataset with missing values!!!



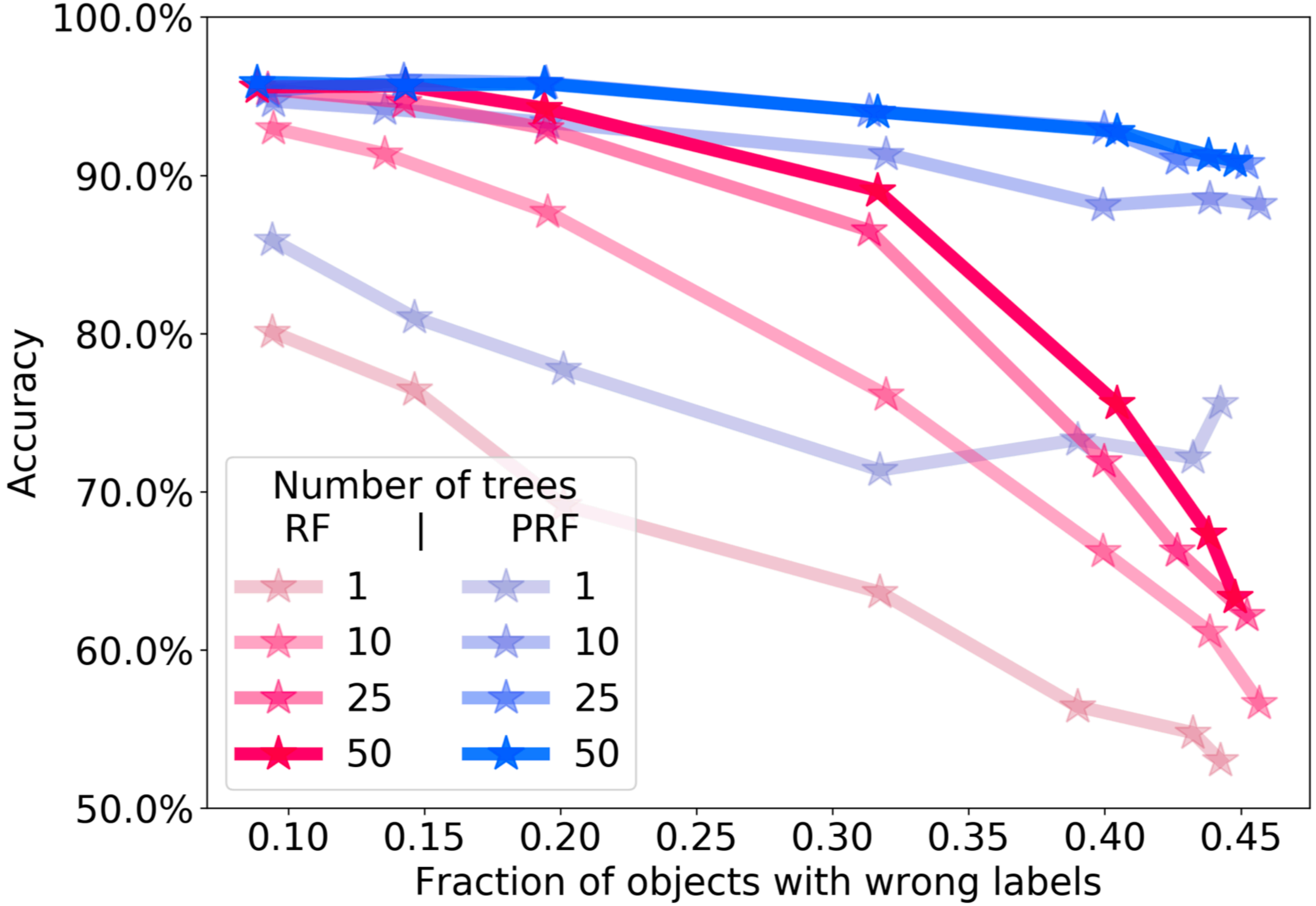
Probabilistic Random Forest

A Random Forest that takes into account the uncertainties in both the features and the input labels. The Probabilistic Random Forest treats all measurements as random variables (see Reis+18).



Probabilistic Random Forest

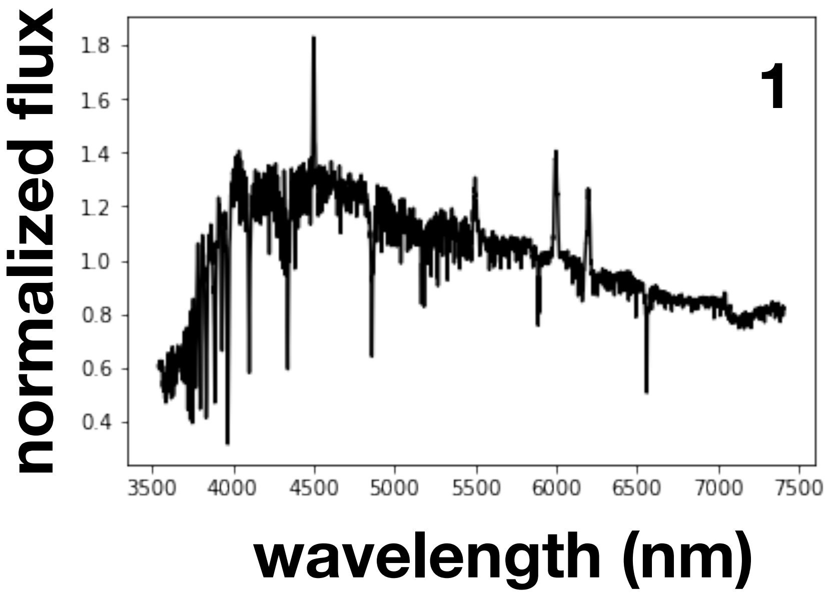
A Random Forest that takes into account the uncertainties in both the features and the input labels. The Probabilistic Random Forest treats all measurements as random variables (see Reis+18).

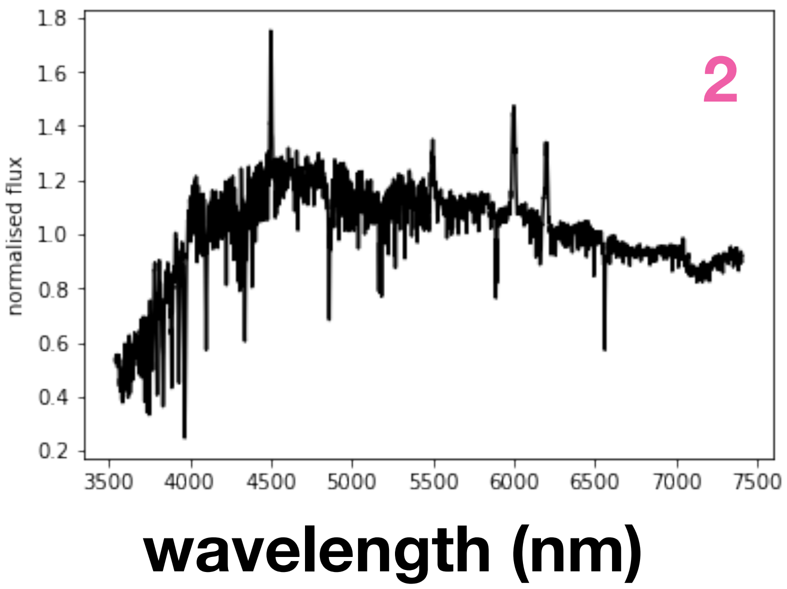


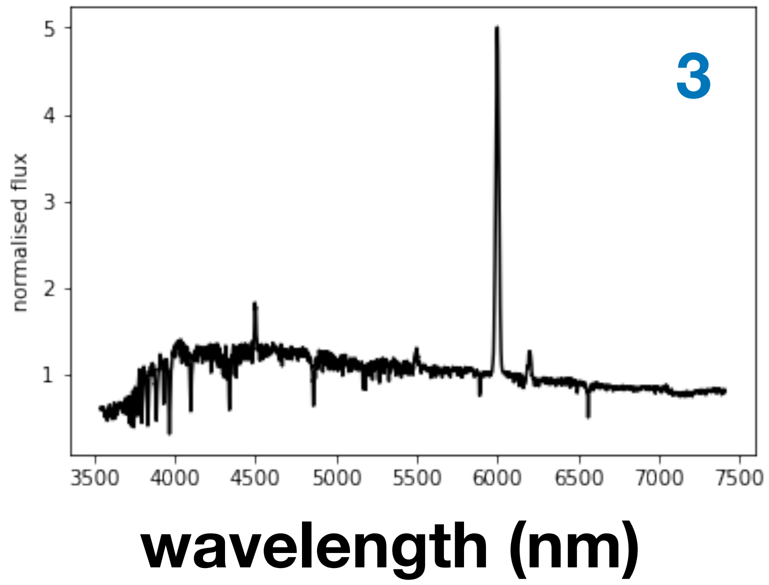
Unsupervised Random Forest

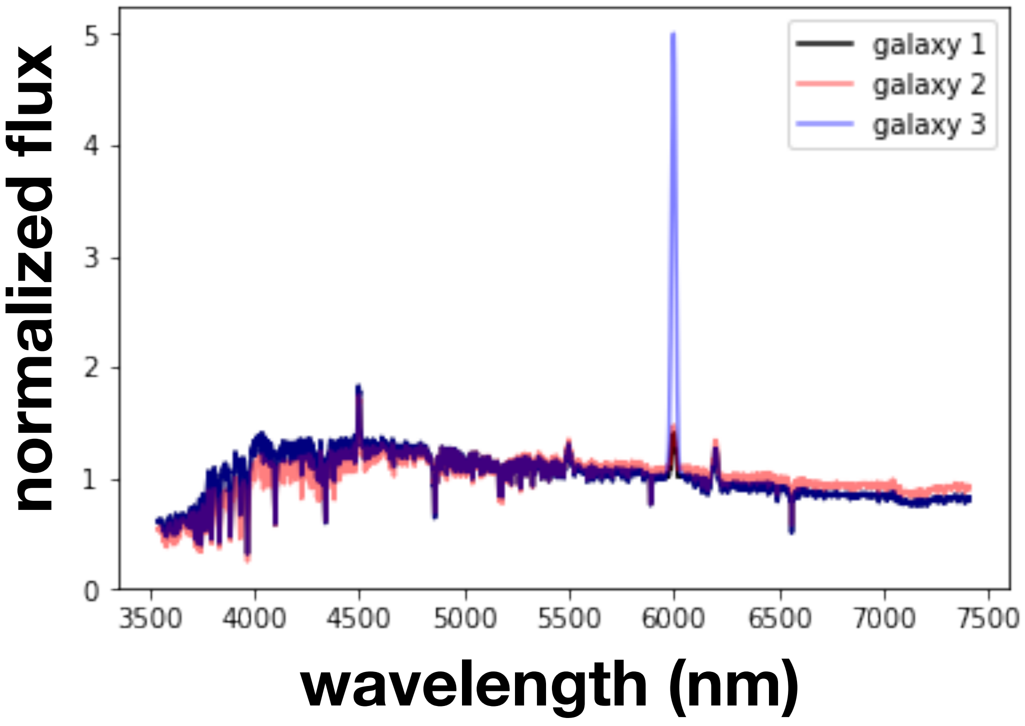
Random Forest can be used as an unsupervised algorithm, to produce pair-wise similarity for the objects in our sample.

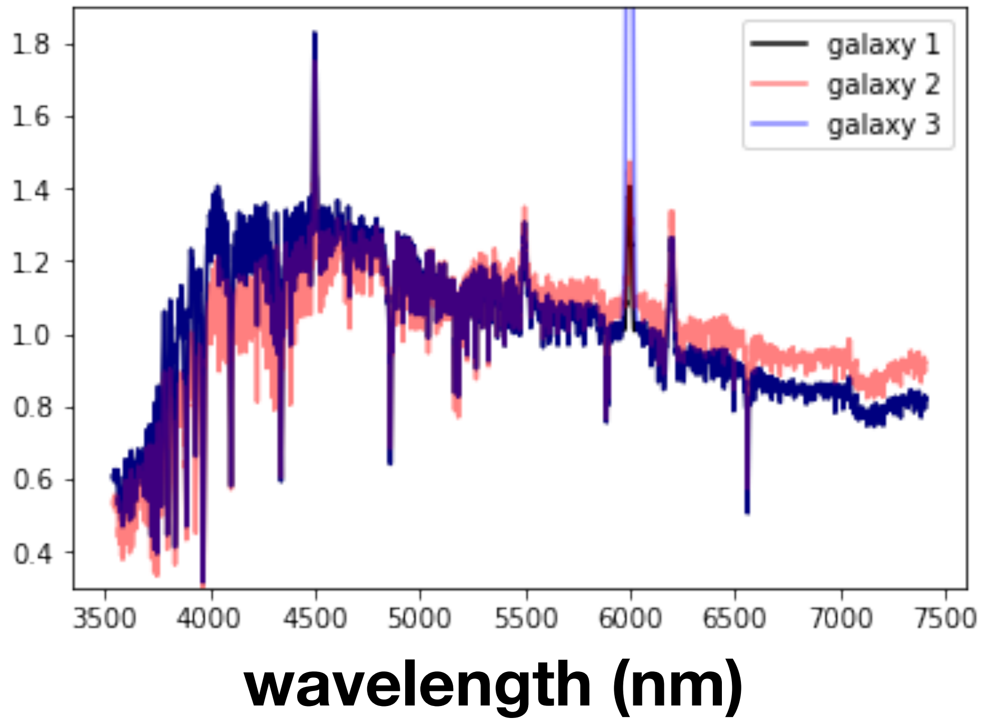
Why do we need to measure distances between objects?









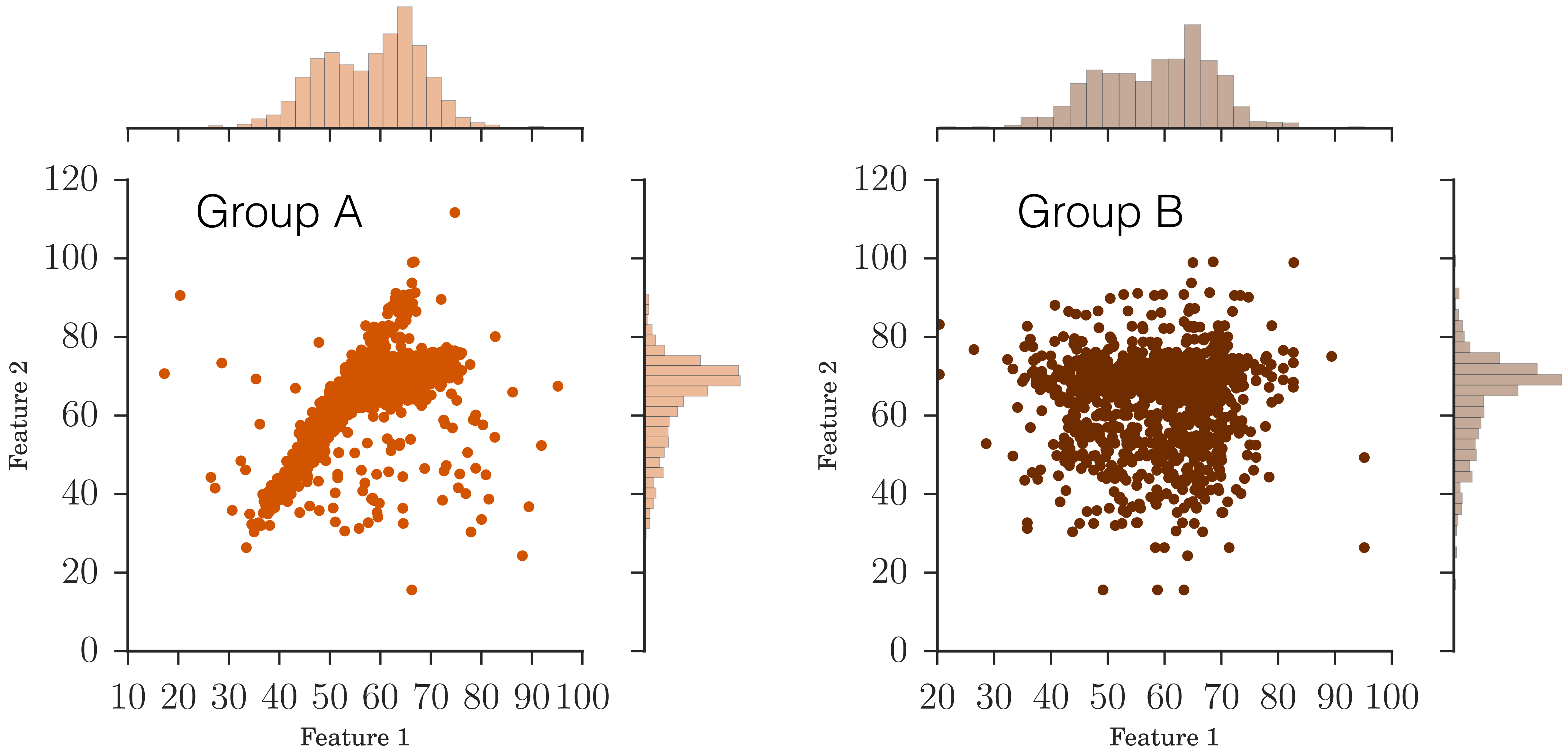


Unsupervised Random Forest

Random Forest can be used as an unsupervised algorithm, to produce pair-wise similarity for the objects in our sample.

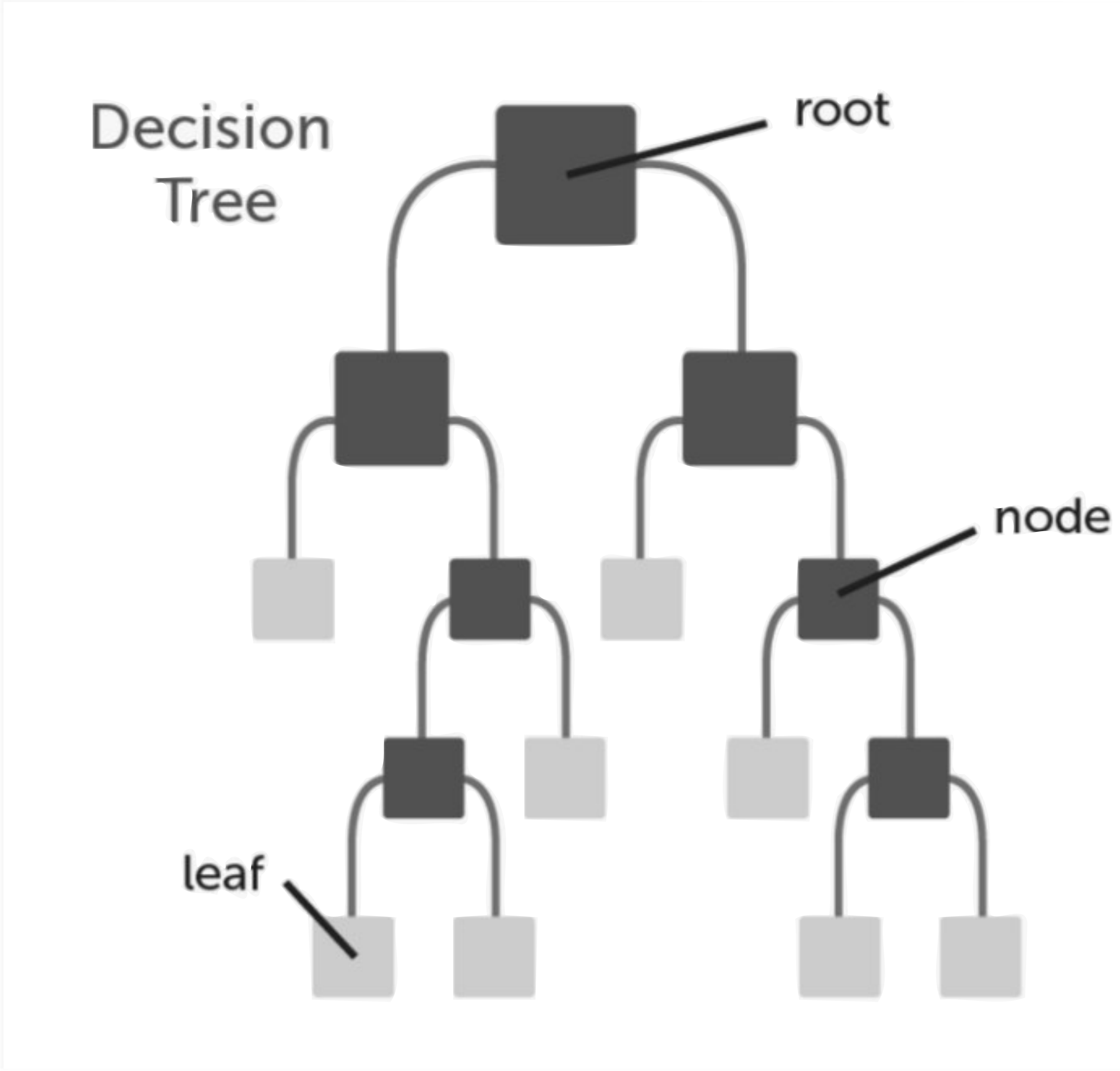
Input dataset: a list of objects with measured features, but no labels!

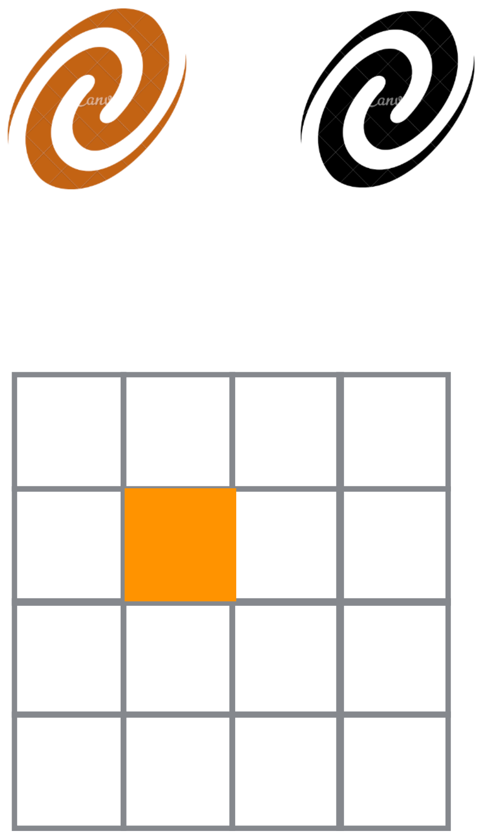
Random Forest is trained to distinguish between real and synthetic datasets.



Unsupervised Random Forest

We train the Random Forest to distinguish between groups A and B.
For group A (real data), we propagate the objects and obtain a similarity matrix.

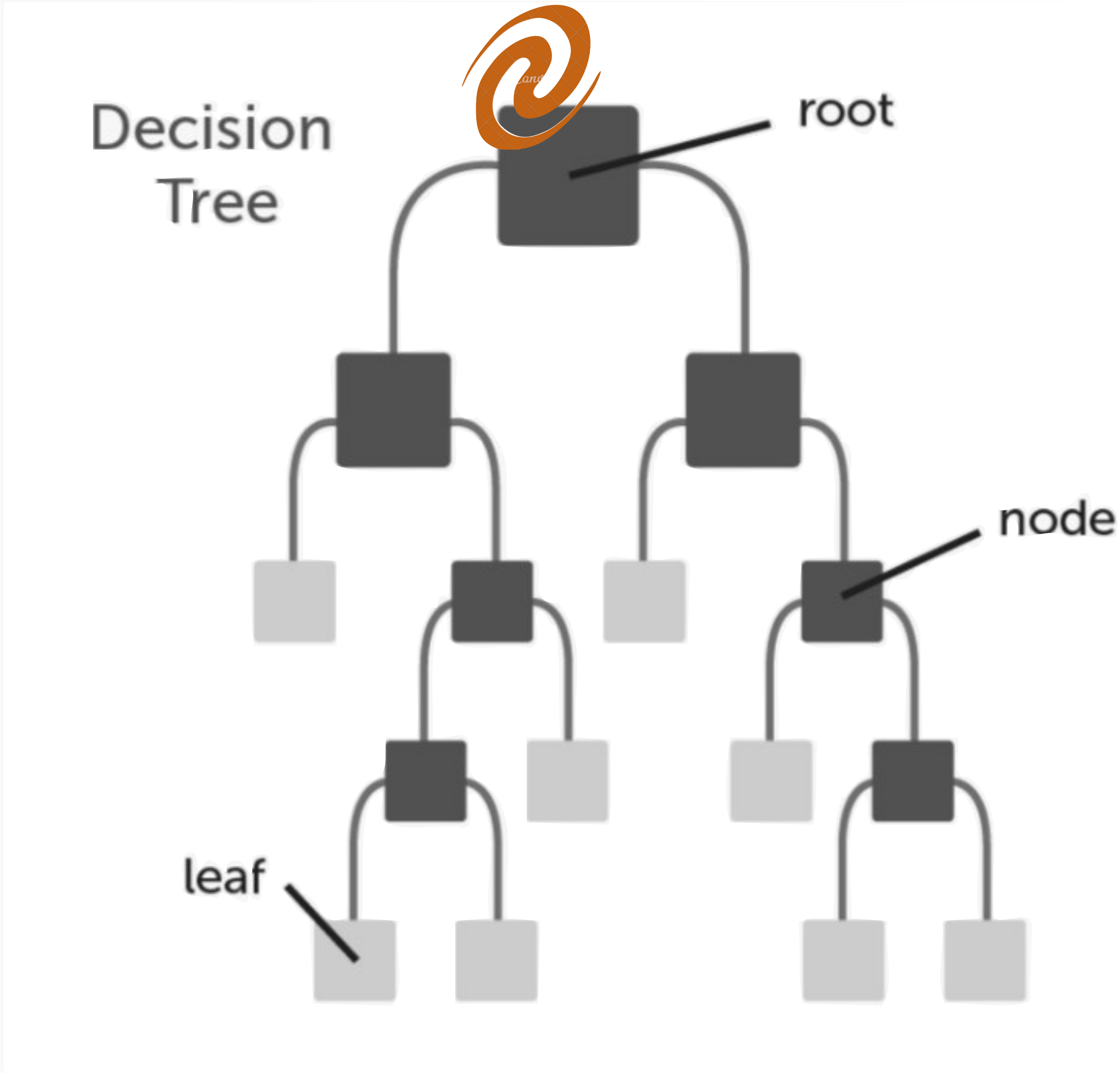


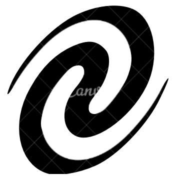


similarity matrix

Unsupervised Random Forest

We train the Random Forest to distinguish between groups A and B.
For group A (real data), we propagate the objects and obtain a similarity matrix.



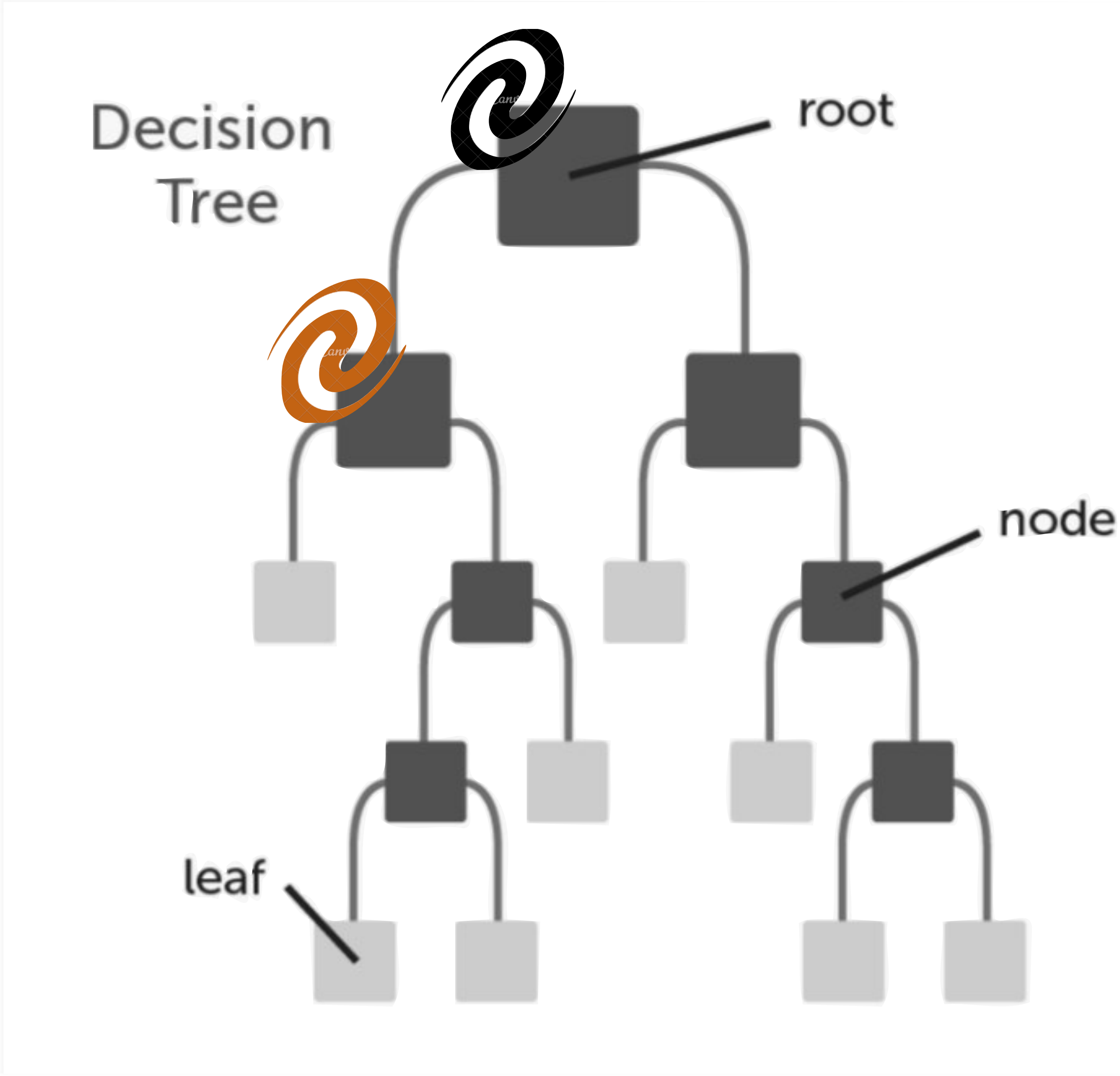


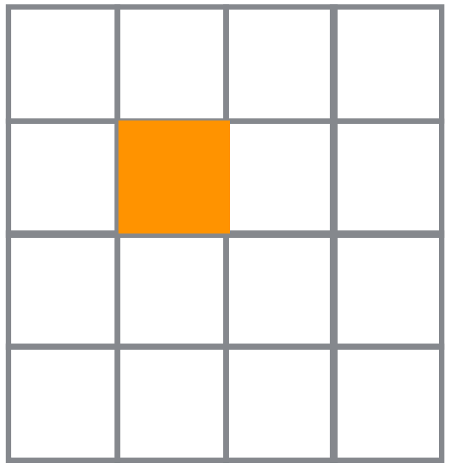
	🟧		

similarity matrix

Unsupervised Random Forest

We train the Random Forest to distinguish between groups A and B.
For group A (real data), we propagate the objects and obtain a similarity matrix.

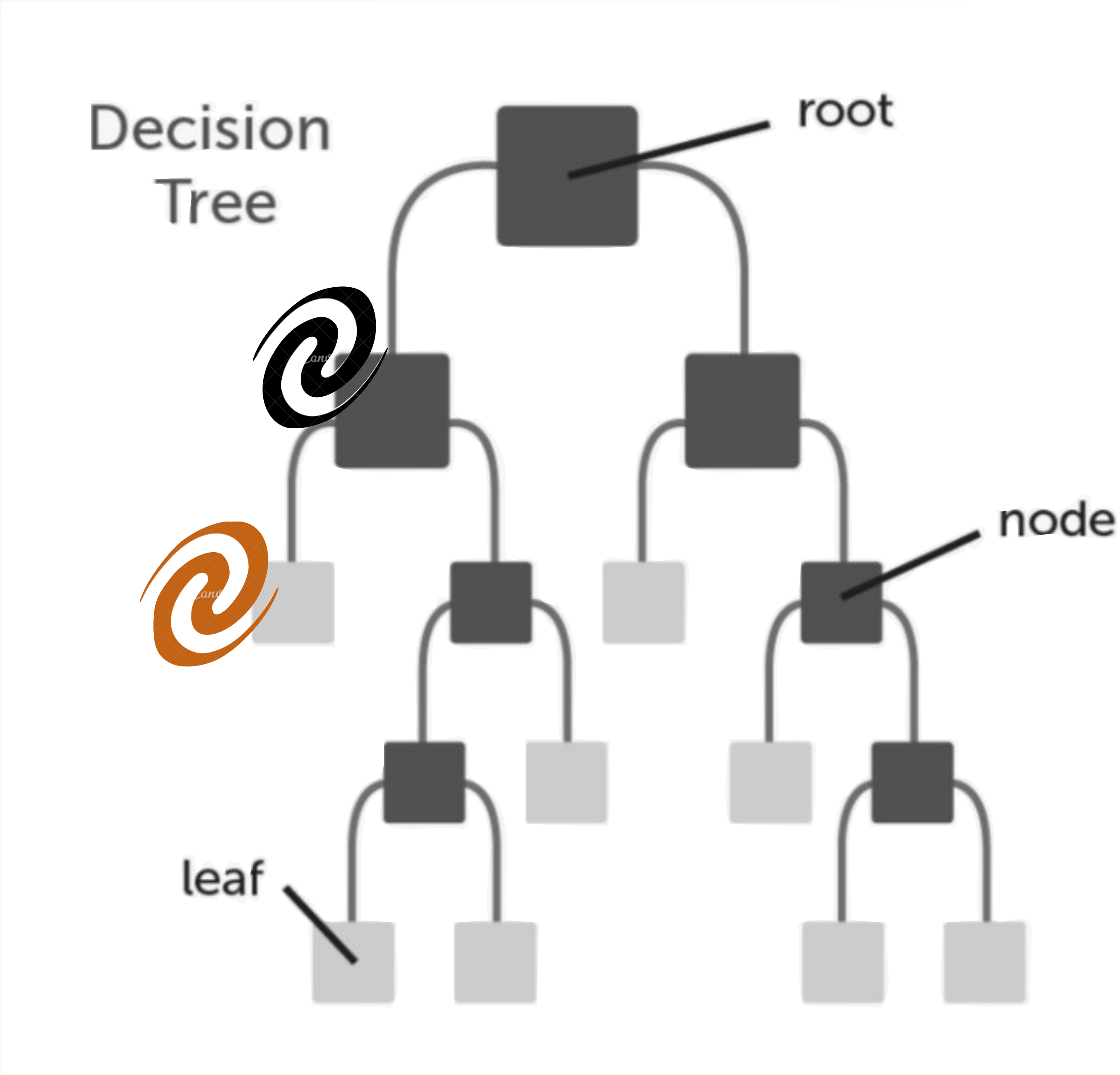


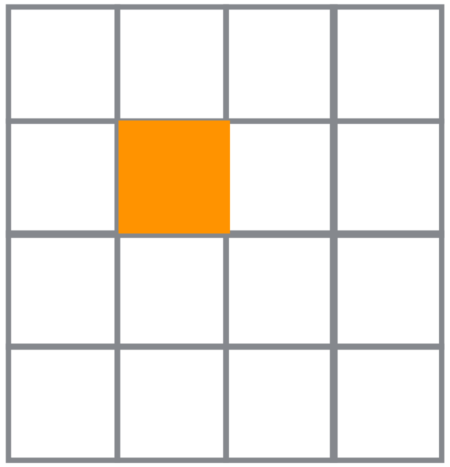


similarity matrix

Unsupervised Random Forest

We train the Random Forest to distinguish between groups A and B.
For group A (real data), we propagate the objects and obtain a similarity matrix.

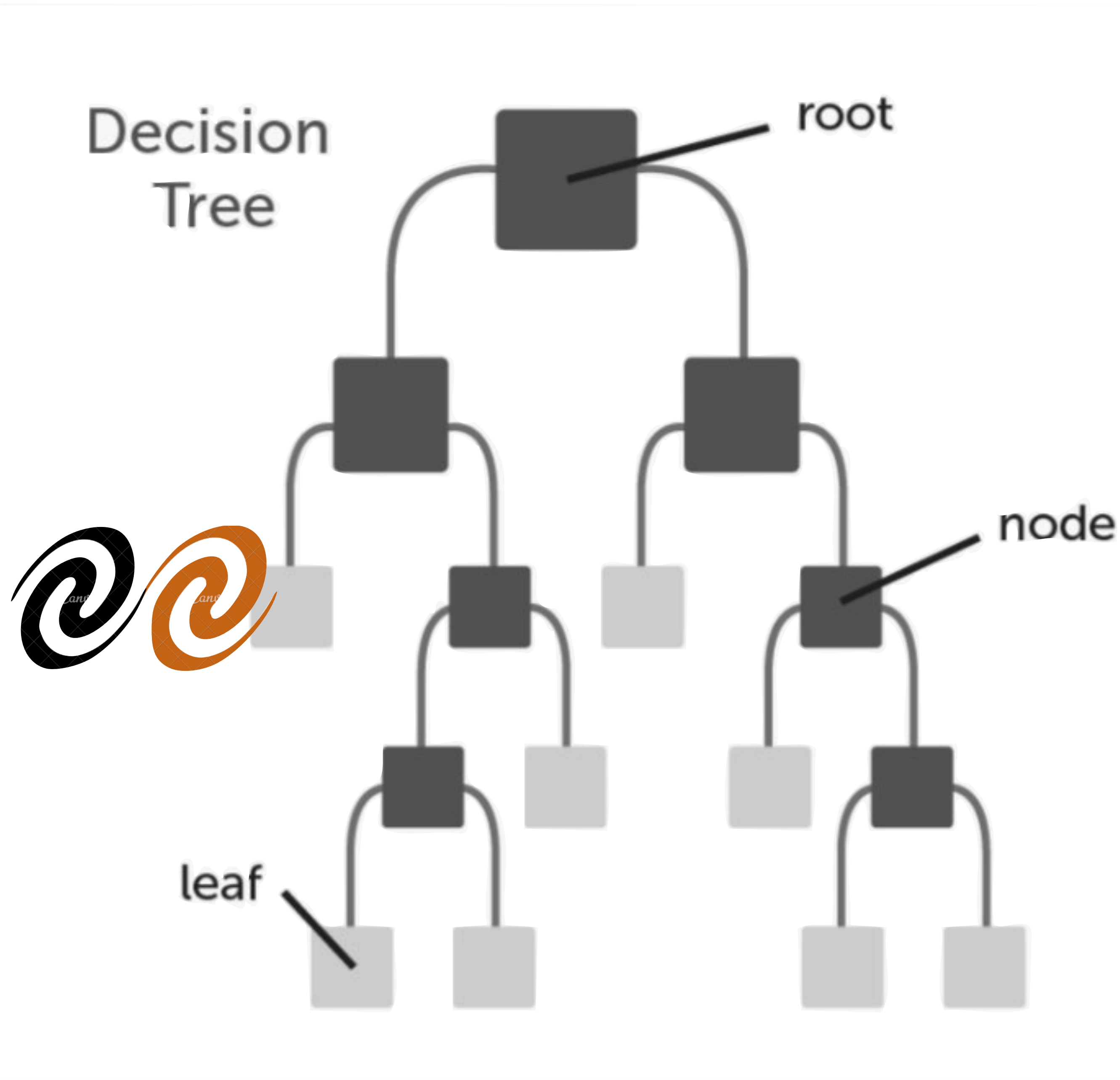




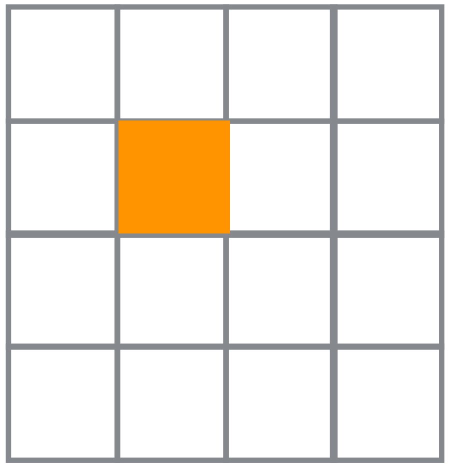
similarity matrix

Unsupervised Random Forest

We train the Random Forest to distinguish between groups A and B.
For group A (real data), we propagate the objects and obtain a similarity matrix.



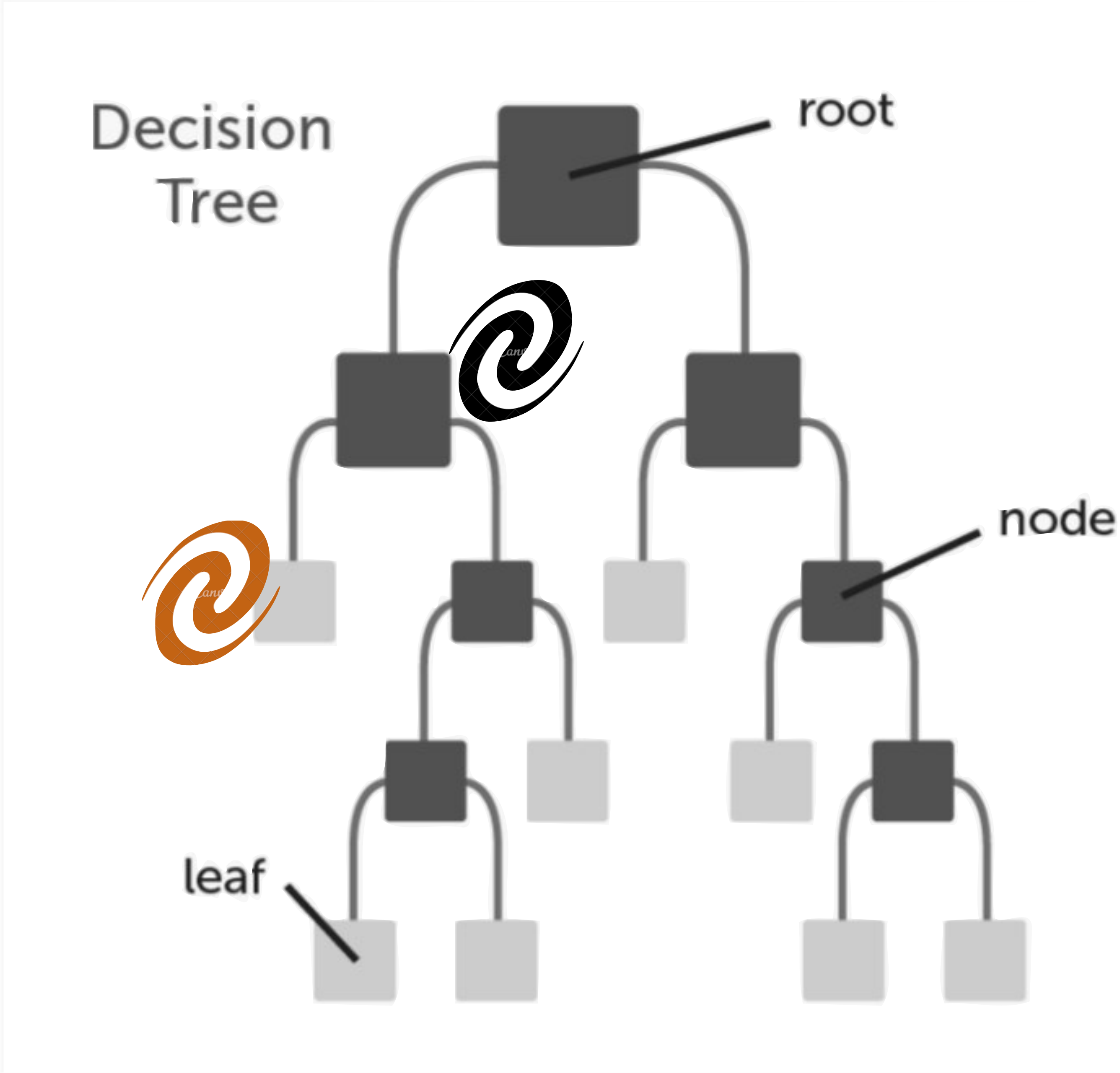
similarity += 1

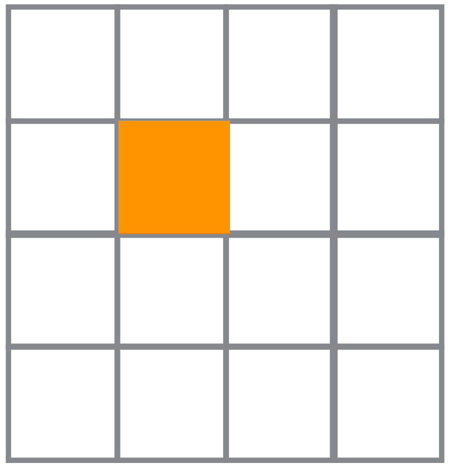


similarity matrix

Unsupervised Random Forest

We train the Random Forest to distinguish between groups A and B.
For group A (real data), we propagate the objects and obtain a similarity matrix.

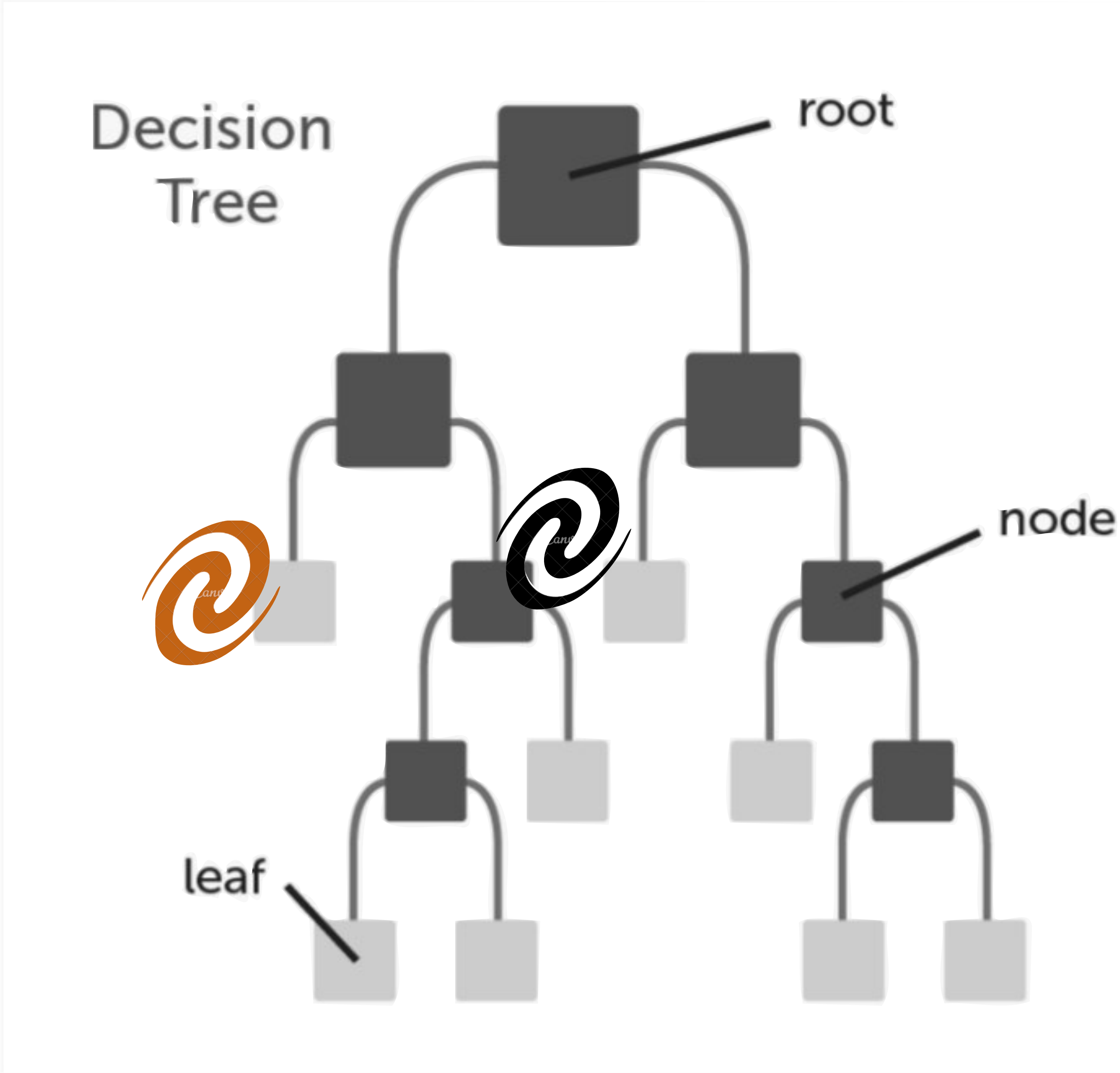


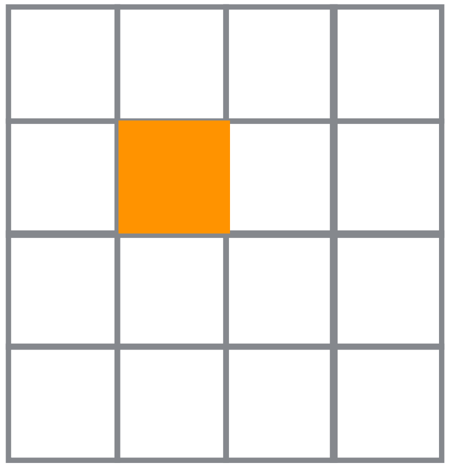


similarity matrix

Unsupervised Random Forest

We train the Random Forest to distinguish between groups A and B.
For group A (real data), we propagate the objects and obtain a similarity matrix.

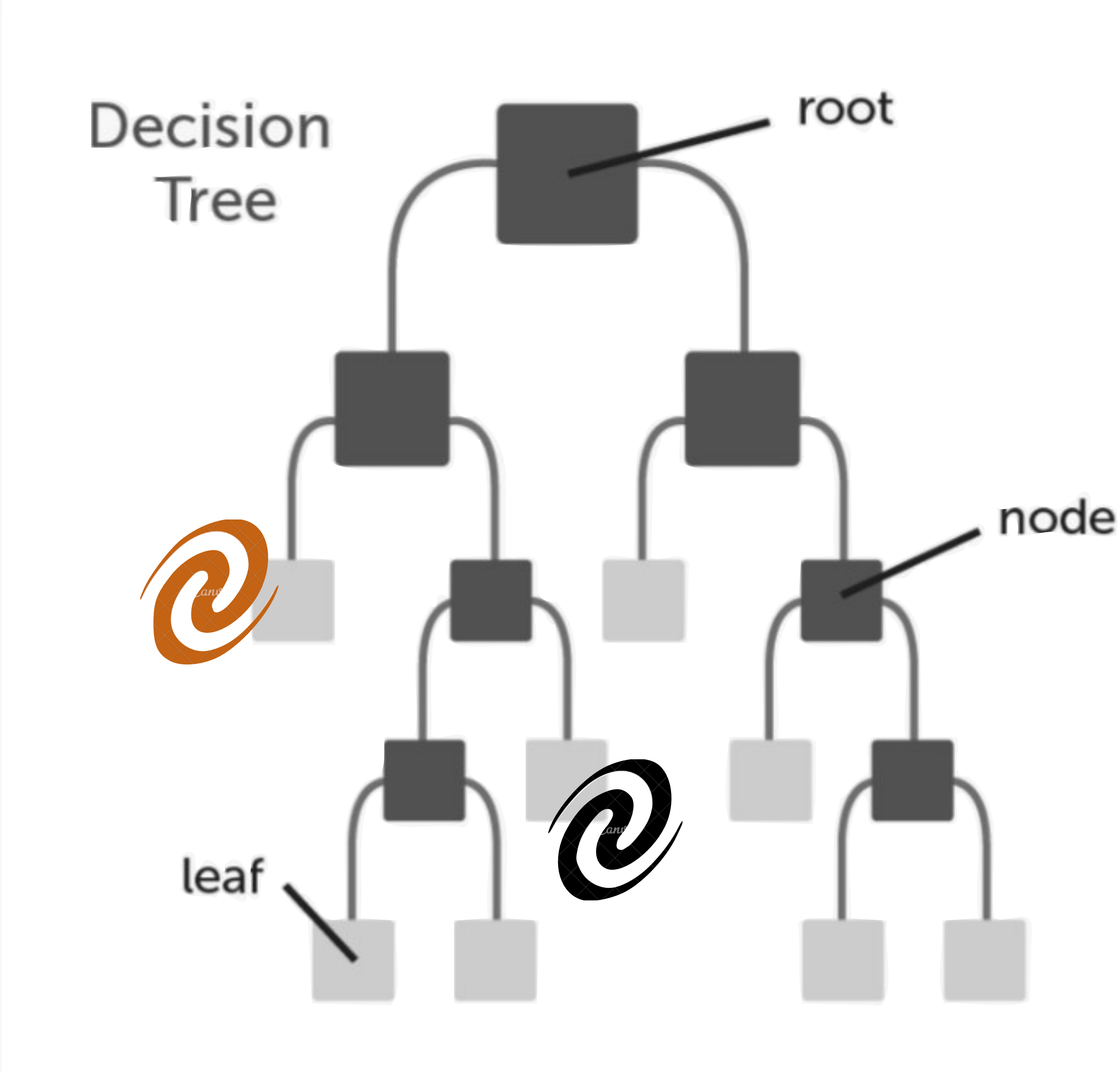


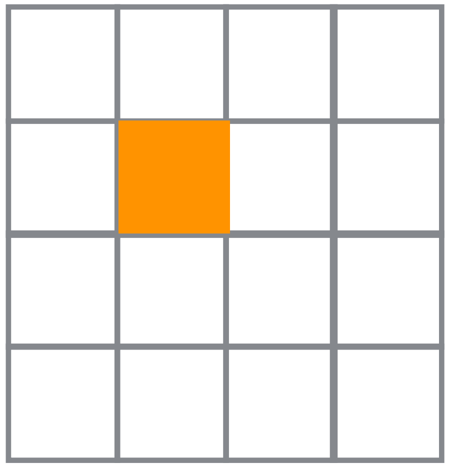


similarity matrix

Unsupervised Random Forest

We train the Random Forest to distinguish between groups A and B.
For group A (real data), we propagate the objects and obtain a similarity matrix.



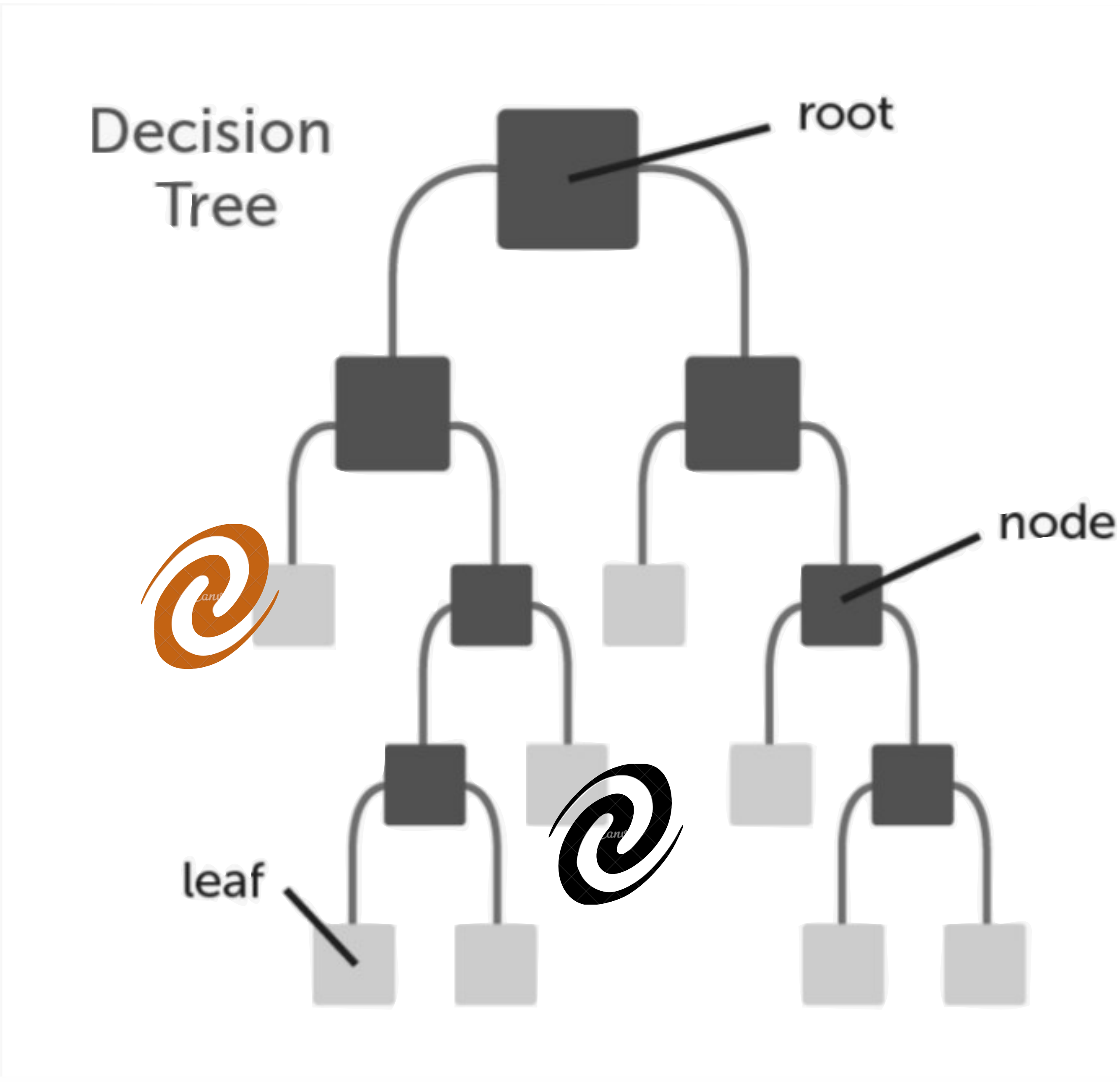


similarity matrix

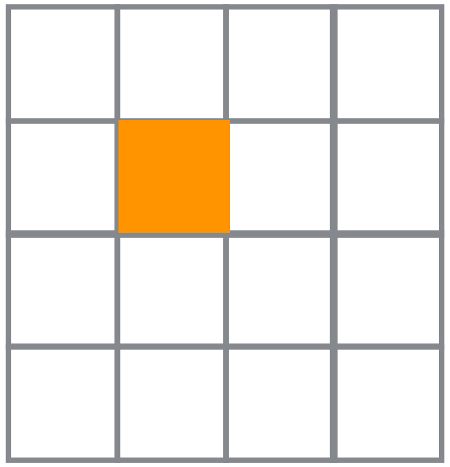
similarity += 0

Unsupervised Random Forest

We train the Random Forest to distinguish between groups A and B.
For group A (real data), we propagate the objects and obtain a similarity matrix.



similarity += 0



similarity matrix

The process is repeated for all the trees in the forest. Therefore, the similarity ranges from 0 to N, the number of trees in the forest.

Questions?