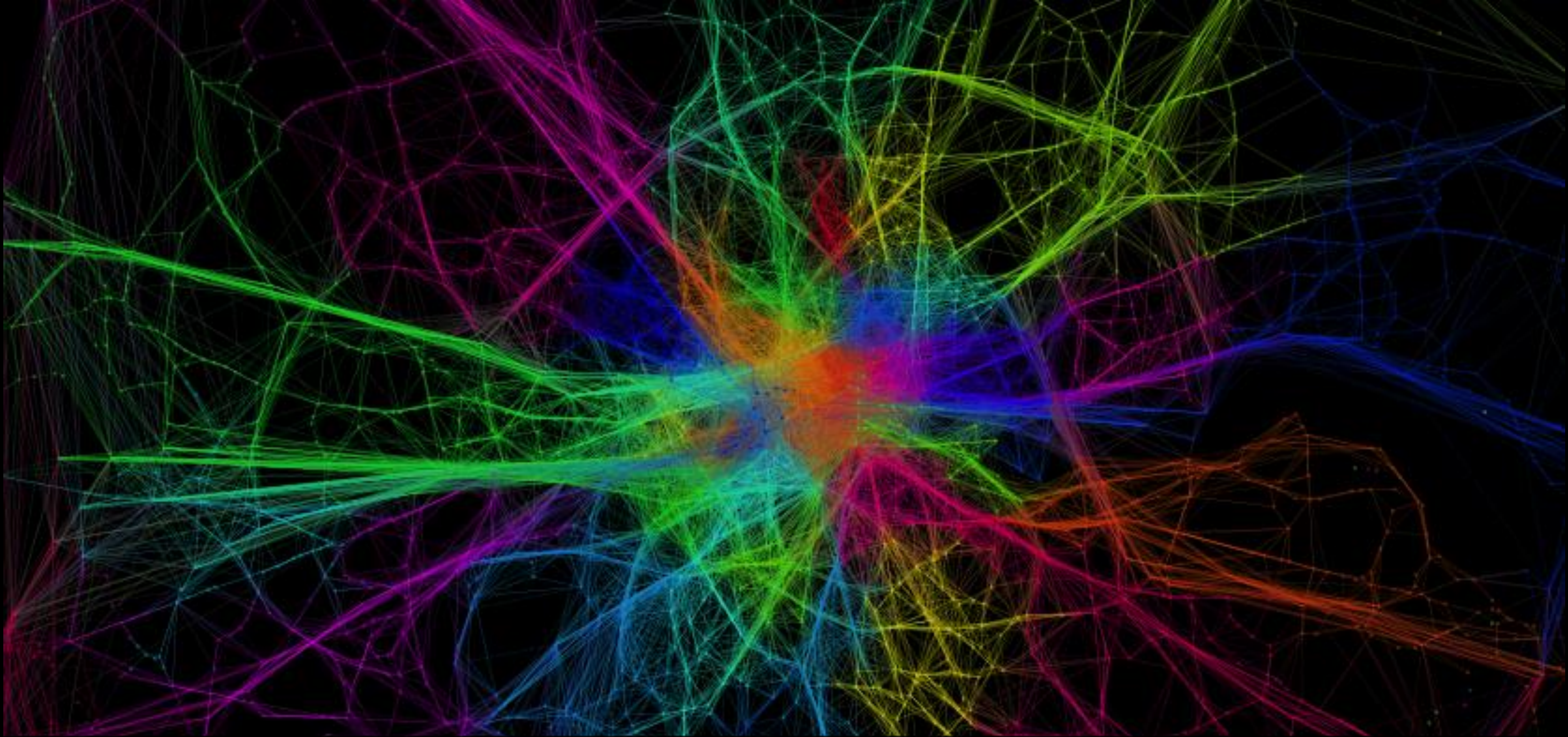


Tree-based Methods

CASA0006: Data Science for Spatial Systems

Huanfa Chen

CASA0006

1. Introduction to Module
2. Supervised Machine Learning
3. Tree-based Methods
4. Artificial Neural Networks
5. Analysis Workflow
6. Panel Regression
7. Difference in Difference
8. Regression Discontinuity
9. Dimensionality Reduction
10. Spatial Clustering

Objectives

- Learn the basics of decision tree
- Understand the idea of ensemble learning
- Learn the principle of random forest (RF) and gradient boosting decision tree (GBDT)
- Understand the permutation feature importance to interpret tree-based models

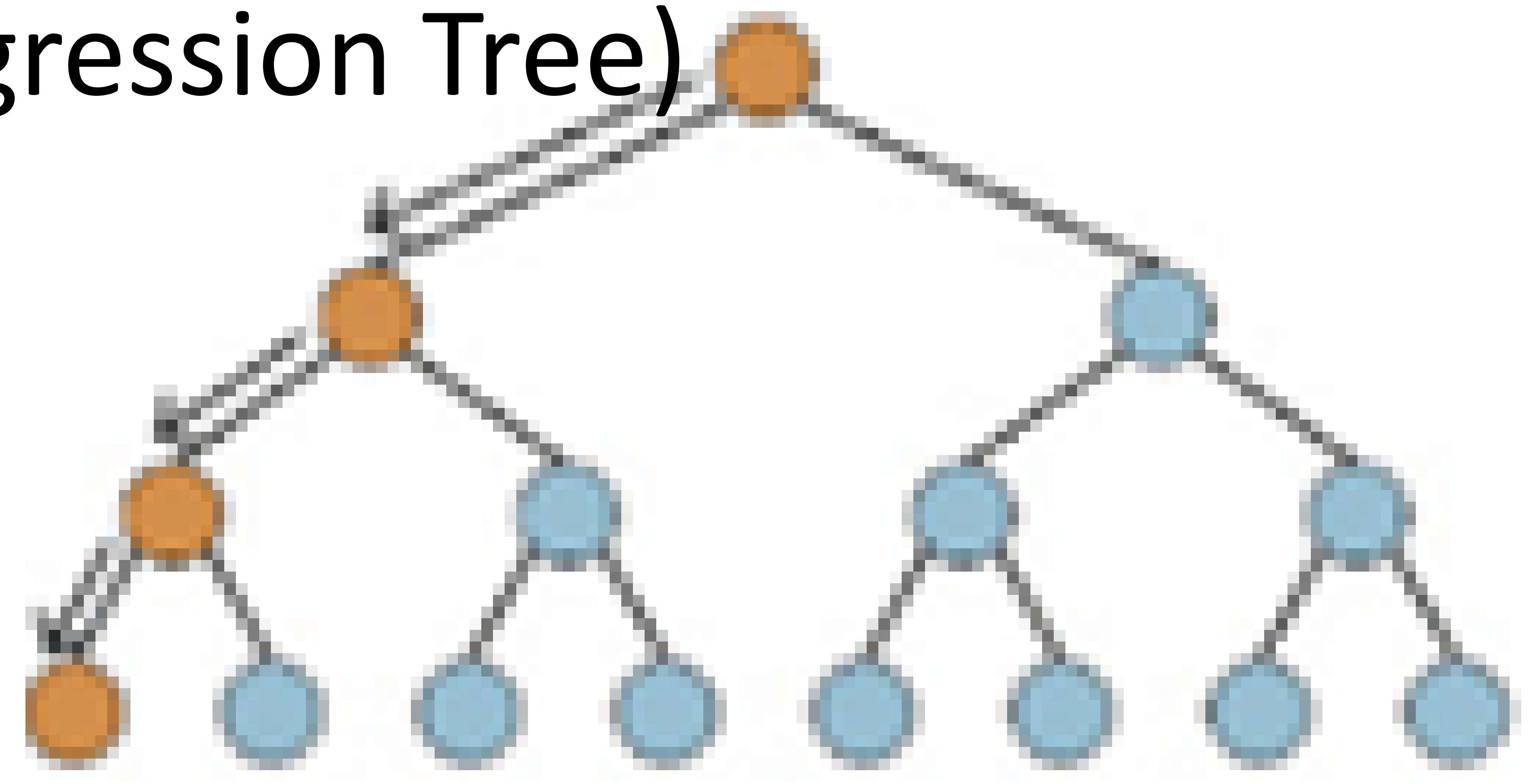
Outline

1. Decision trees
2. Ensemble learning
3. Random forest
4. GBDT
5. Model interpretation

Decision trees

Decision trees

- Simply speaking, CART consists of a flow diagram or a ‘tree’ of decisions about the explanatory variables of a dataset. The structure is similar to a list of if-else statements
- Data-driven approach
- No assumptions about the data relationship
- There are different types of decision trees (CART, ID3, others)
- We focus on the CART (Classification and regression Tree)



Example: to play tennis or not?

- Imagine you play tennis every Sunday and you invite your best friend, Clare to come with you every time.
- Clare sometimes comes to join but sometimes not, and it seems to depend on some factors – including weather, temperature, humidity and wind. You would like to use the historical dataset below to predict whether or not Clare will play tennis.

Adapted from this source: https://www.vebuso.com/2020/01/decision-tree-intuition-from-concept-to-application

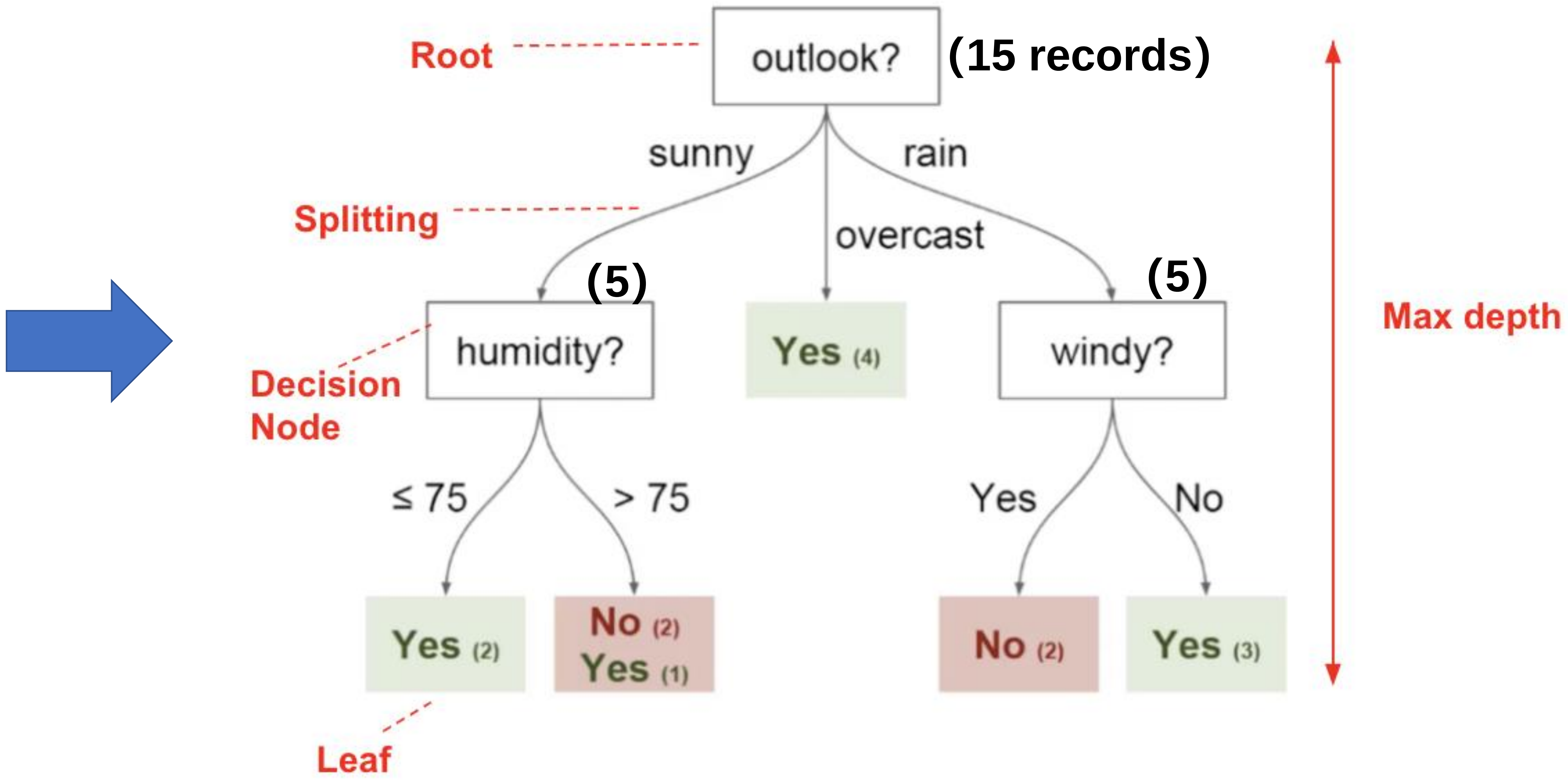
Example: to play tennis or not?

Dataset

Temperature	Outlook	Humidity	Windy	Played?
Mild	Sunny	80	No	Yes
Hot	Sunny	75	Yes	No
Hot	Overcast	77	No	Yes
Cool	Rain	70	No	Yes
Cool	Overcast	72	Yes	Yes
Mild	Sunny	77	No	No
Cool	Sunny	70	No	Yes
Mild	Rain	69	No	Yes
Mild	Sunny	65	Yes	Yes
Mild	Overcast	77	Yes	Yes
Hot	Overcast	74	No	Yes
Mild	Rain	77	Yes	No
Cool	Rain	73	Yes	No
Mild	Rain	78	No	Yes

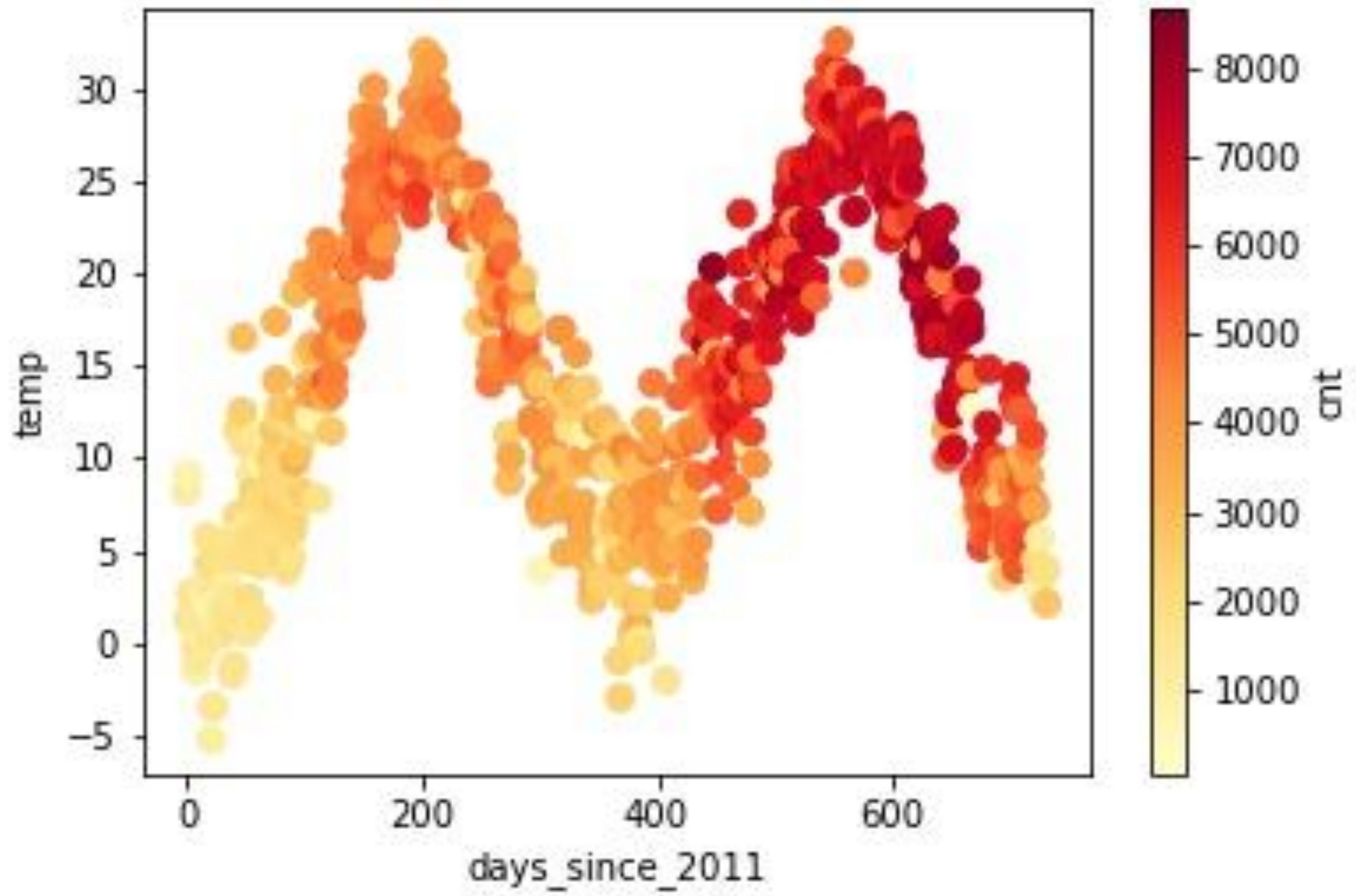
(15 records): NO. of records in each node

Decision Tree Diagram



Adapted from this source: https://www.vebuso.com/2020/01/decision-tree-intuition-from-concept-to-application

Another CART: predict daily bike rental

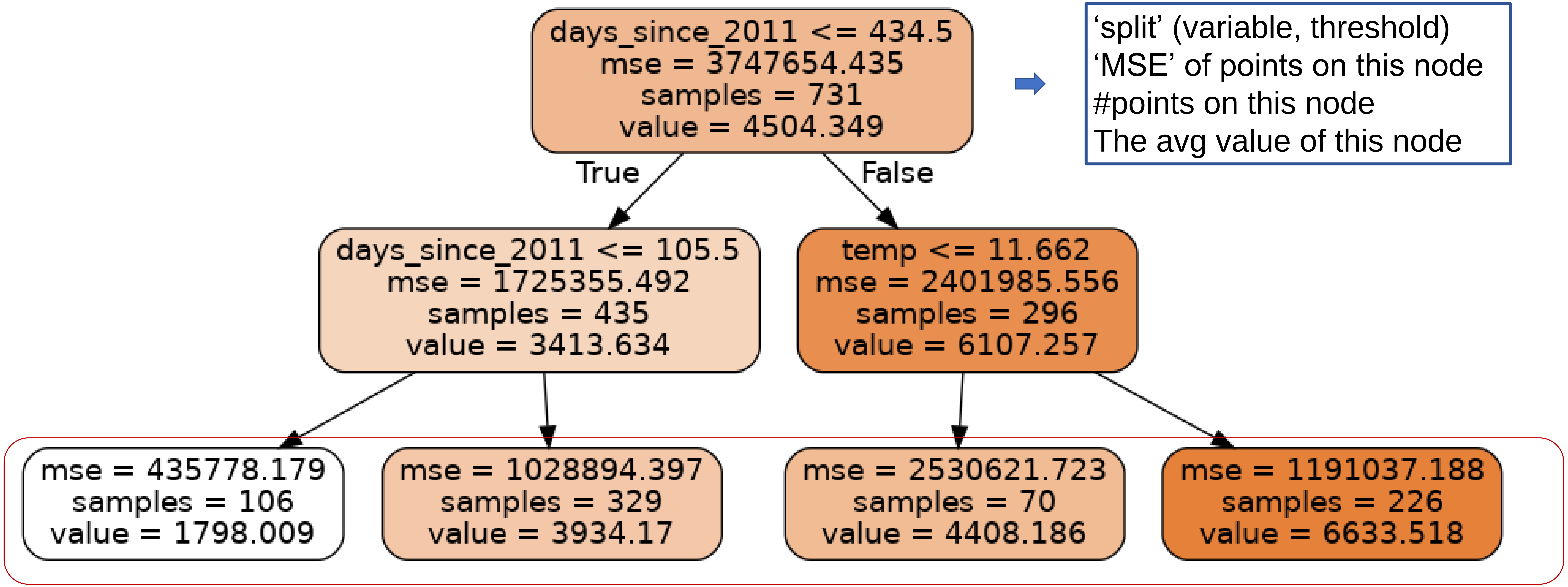


X_1: number of days since 2011
X_2: temperature
Y (colour): daily bike rental (or count)

We used only two x variables in order to simplify this example. CART is applicable to any dimension of variables

Data provided by bicycle rental company Capital-Bikeshare in Washington D.C.

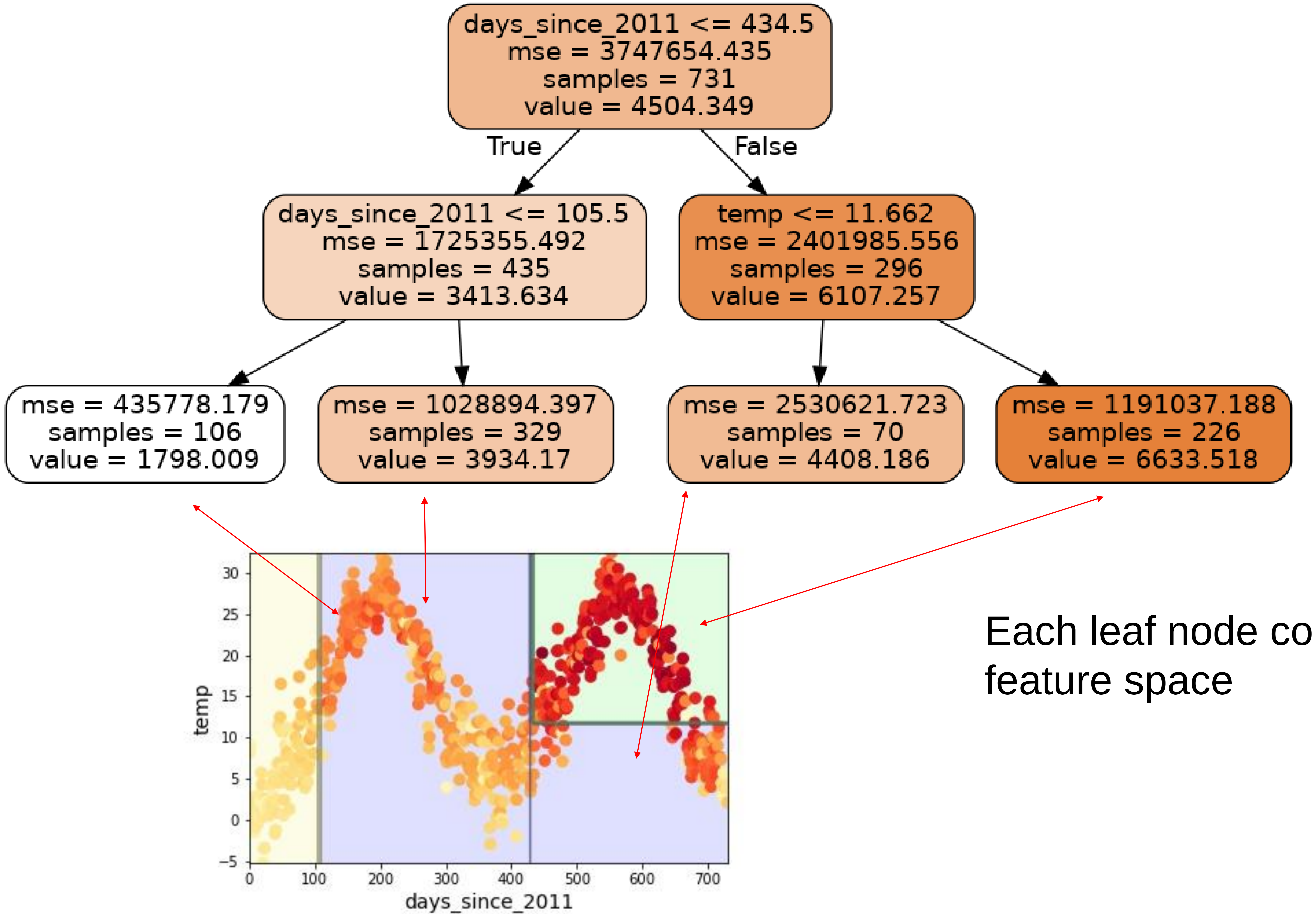
Another CART: predict daily bike rental



'split' (variable, threshold)
'MSE' of points on this node
#points on this node
The avg value of this node

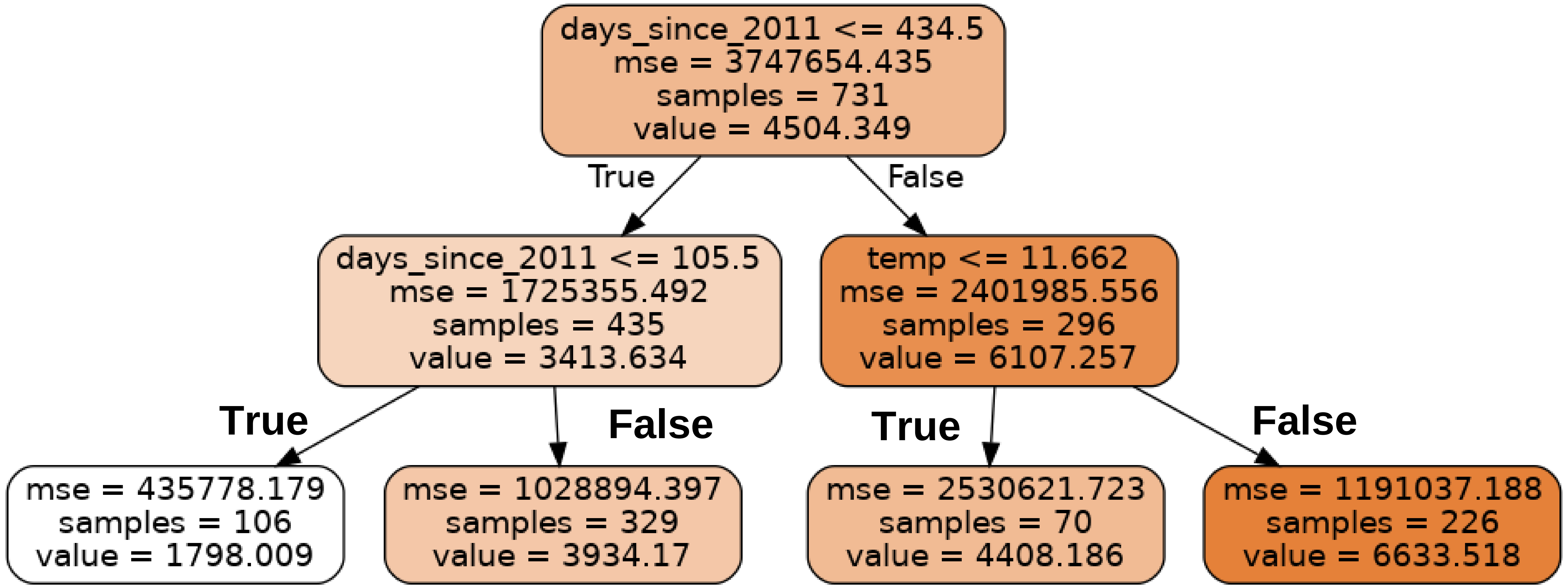
Leaf node. If a data point fall into a leaf, then it is predicted as the 'value' of this leaf.

Another CART: predict daily bike rental



Each leaf node corresponds to a subset of the feature space

Using CART for prediction



(days_since_2011, temp)

(435, 12): predicted_bike_rental = ??

(434, 12): predicted_bike_rental = ??

How to train a CART

- Q1: how to decide which split to adopt? (metrics)
- Q2: when to stop the split? (the stopping criteria)
- Q3: what is the similarity and difference between CART for regression and for classification?

Training of a CART (for regression)

- For each node, splits the sample into two subsets using a single variable *k* at threshold t_k (note only splits into two)
- Chooses *k* and t_k by finding a split that minimise the cost function

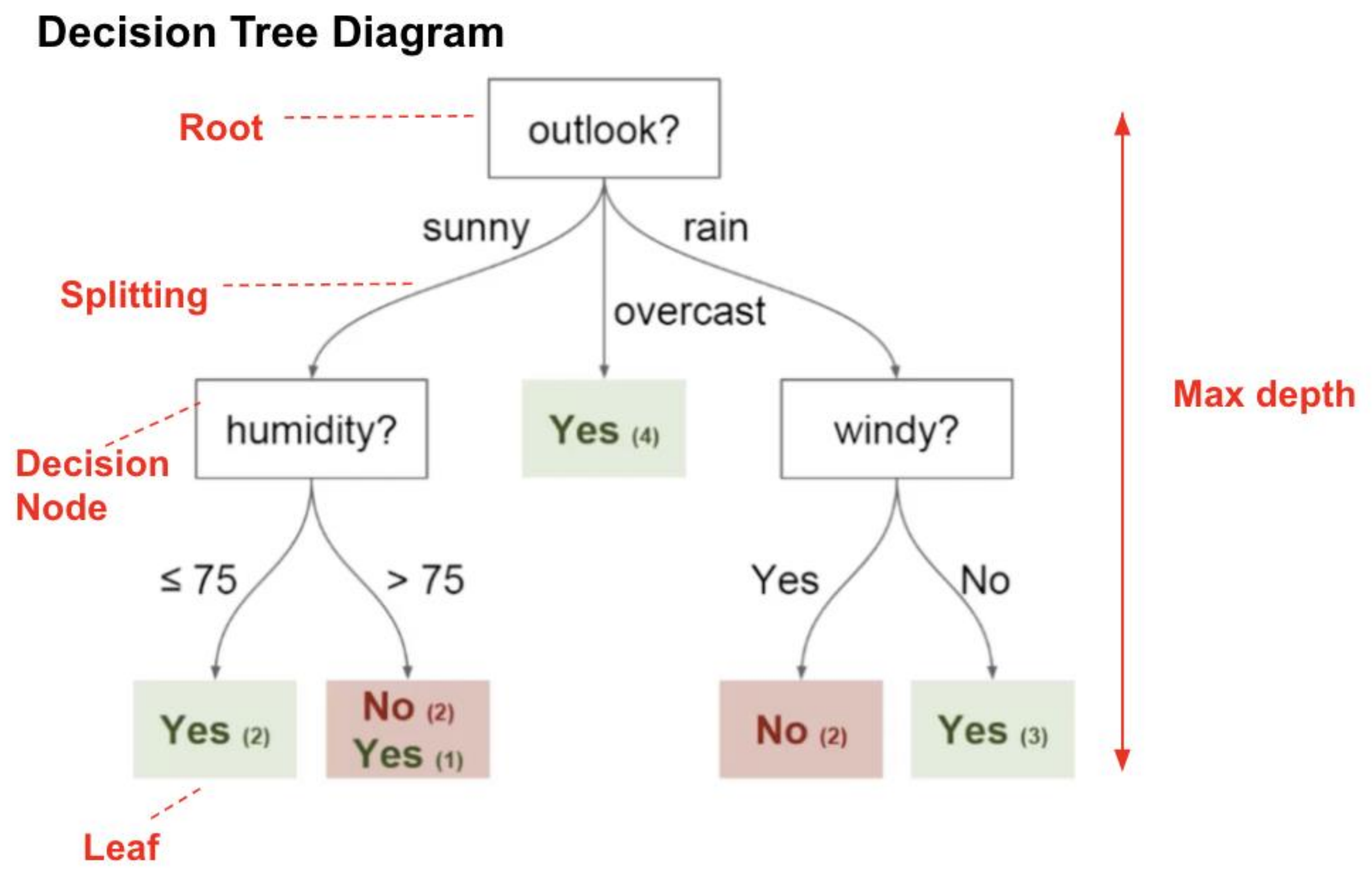
$$J(k, t_k) = \frac{m_{\text{left}}}{m} MSE_{\text{left}} + \frac{m_{\text{right}}}{m} MSE_{\text{right}}$$

  - 'left' and 'right' refer to two groups and m_{left} refers to the number of points in group left. $m = m_{\text{left}} + m_{\text{right}}$.
  - MSE: mean square error (representing within-group variation)
- Repeat the splitting until stop criteria are met

Stopping criteria of CART

- Usually there are two stopping criteria, which are hyperparameters of CART and are predefined by users
- A trade-off between model fitness and the extent of overfitting
- The larger max_tree_depth (or smaller min_instances), the more splits, the better fitting on the training data, the more likely to overfit

Stopping criteria	Meaning
Max tree depth	If the layer of a node is deeper than this value, it stops split.
Minimal instances in a node	If #instances of a node is smaller than this value, it stops split.



CART for regression and classification

- Similarity: overall idea, stopping criteria, etc.
- Difference

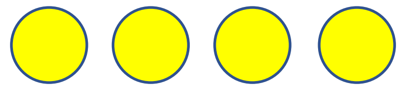
	Cost function of split	Value of a node	Prediction
Regression	Mean square error	Mean of all records on this node	A number
Classification	Gini impurity	Majority class	A class or probability distribution over classes

Gini impurity

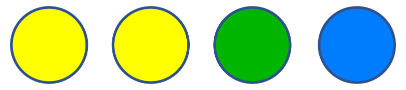
- CART for classification: choose the best split that maximises the increase of purity (compared to before split)

- ***Gini impurity:*** measures the impurity of a group containing different classes (where p_i is the probability of a class)

$$I_G(p) = \sum_{i=1}^{J} p_i(1 - p_i)$$



- Gini = 0. (if and only if only one class in the set)

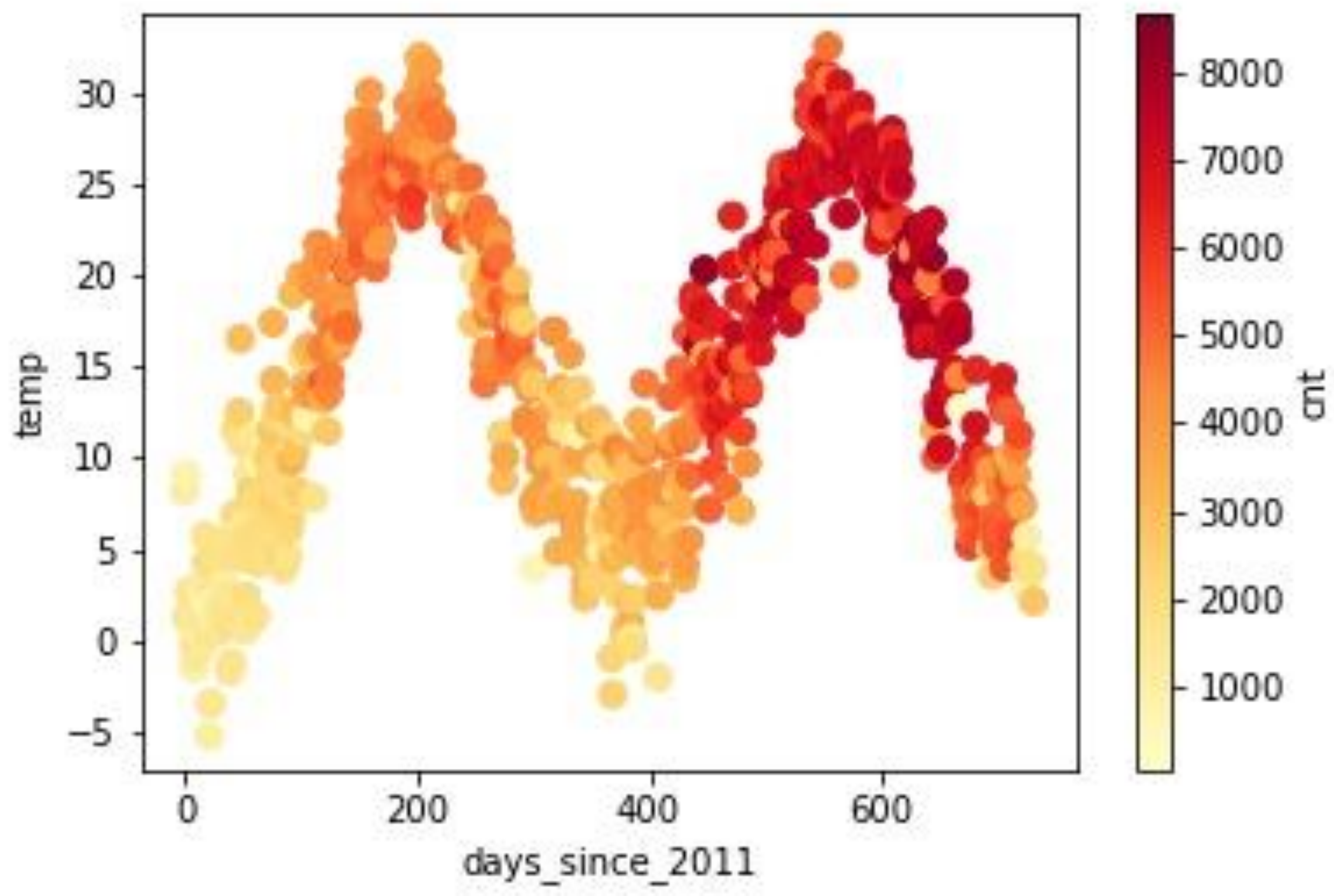


- Gini = 0.5*(1-0.5) + 0.25*(1-0.25) + 0.25*(1-0.25) = 0.625

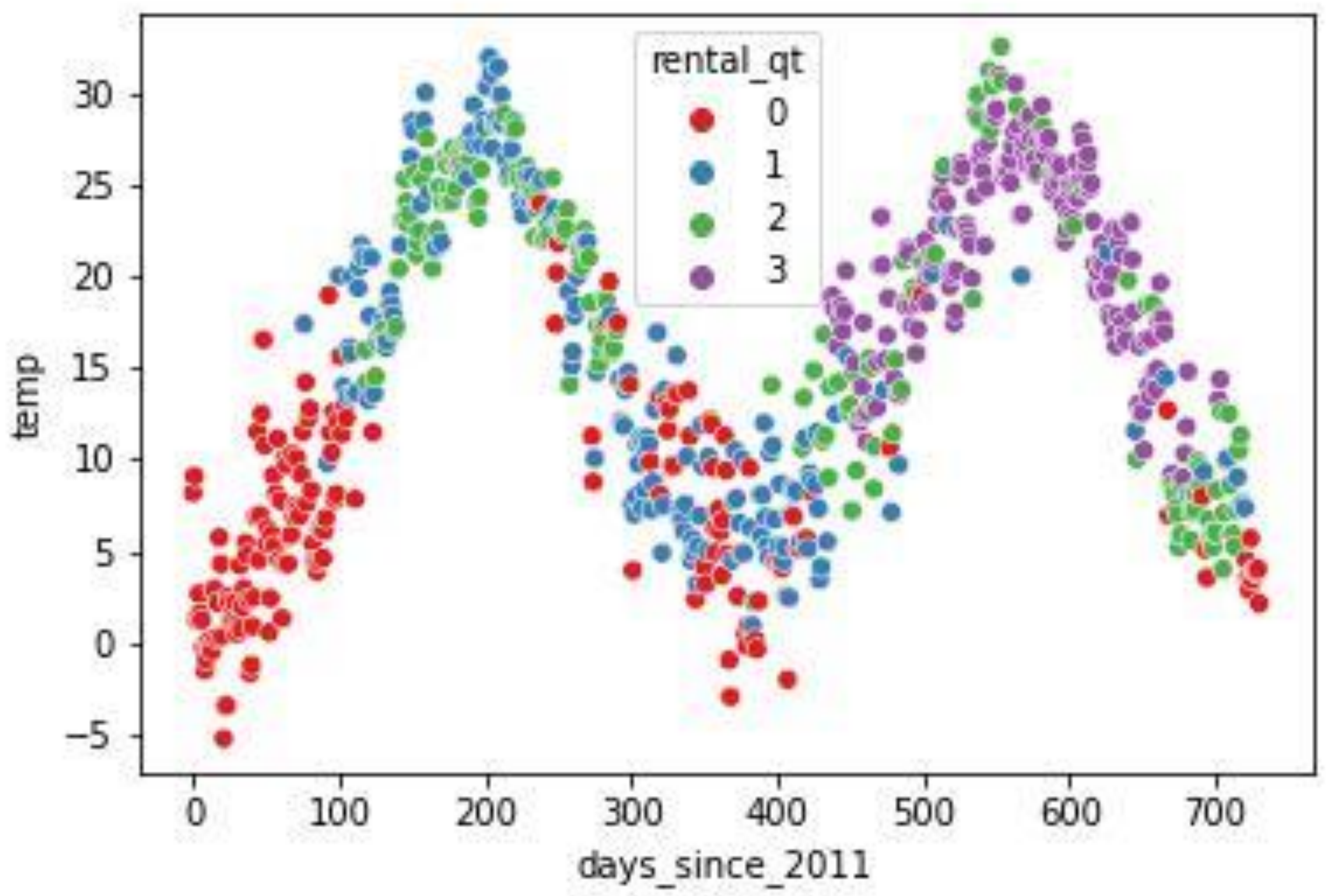
CART for classification

We will illustrate CART for classification by tweaking the bike rental example.

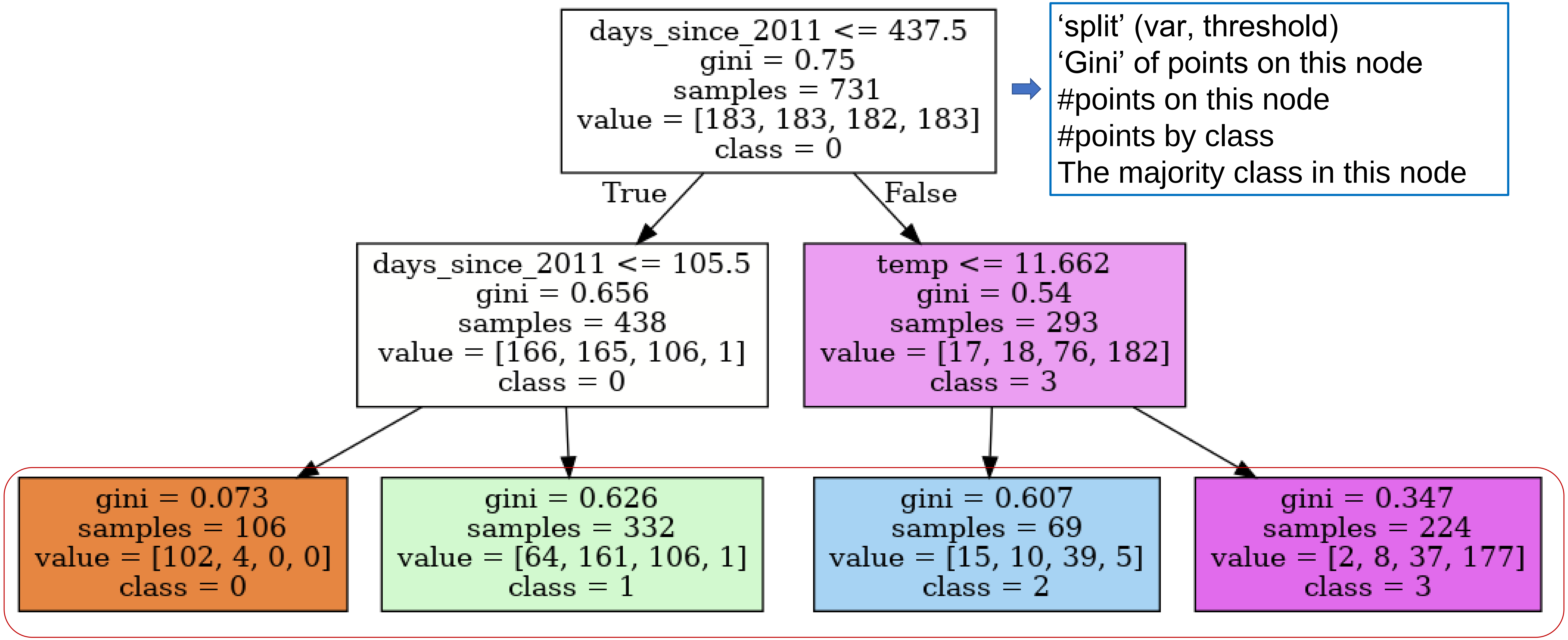
Regression



Classification: for illustration, we transformed this task into a classification task using the quantile (0,25,50,75,100)



CART for classification



'split' (var, threshold)
'Gini' of points on this node
#points on this node
#points by class
The majority class in this node

Leaf node. If a data point fall into a leaf, then it is predicted as the 'majority class' of this leaf.

The prediction can be either a label or a prob distribution on four classes

Summary of CART

- Advantages of CART
  - Interpretability: relatively easy to understand (compared to many trees)
  - Flexibility: no assumptions of data distribution and no transformations needed

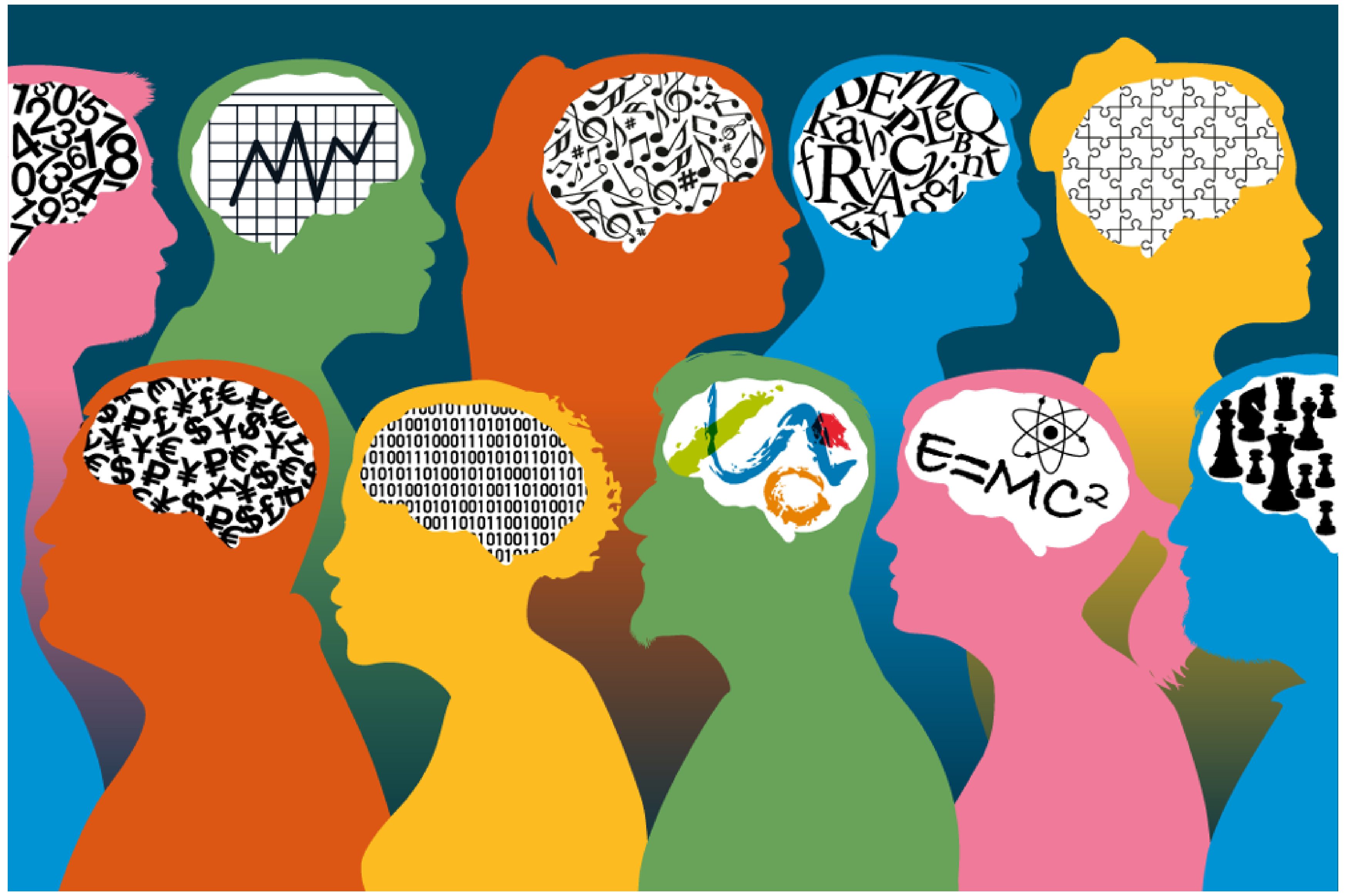
- Disadvantages
  - Lack of smoothness. Slight changes in the predicators can have a big impact on the response
  - Tendency of overfitting: meaning that the tree fits well to the training data but is unable to generalise to new data

- Key points
  - CART can be used for both regression and classification
  - The problem of CART will be tackled by RF or GBDT
  - It is uncommon to use CART to directly make predictions. Rather, CART is used to construct RF or GBDT.

Ensemble learning

Ensemble learning

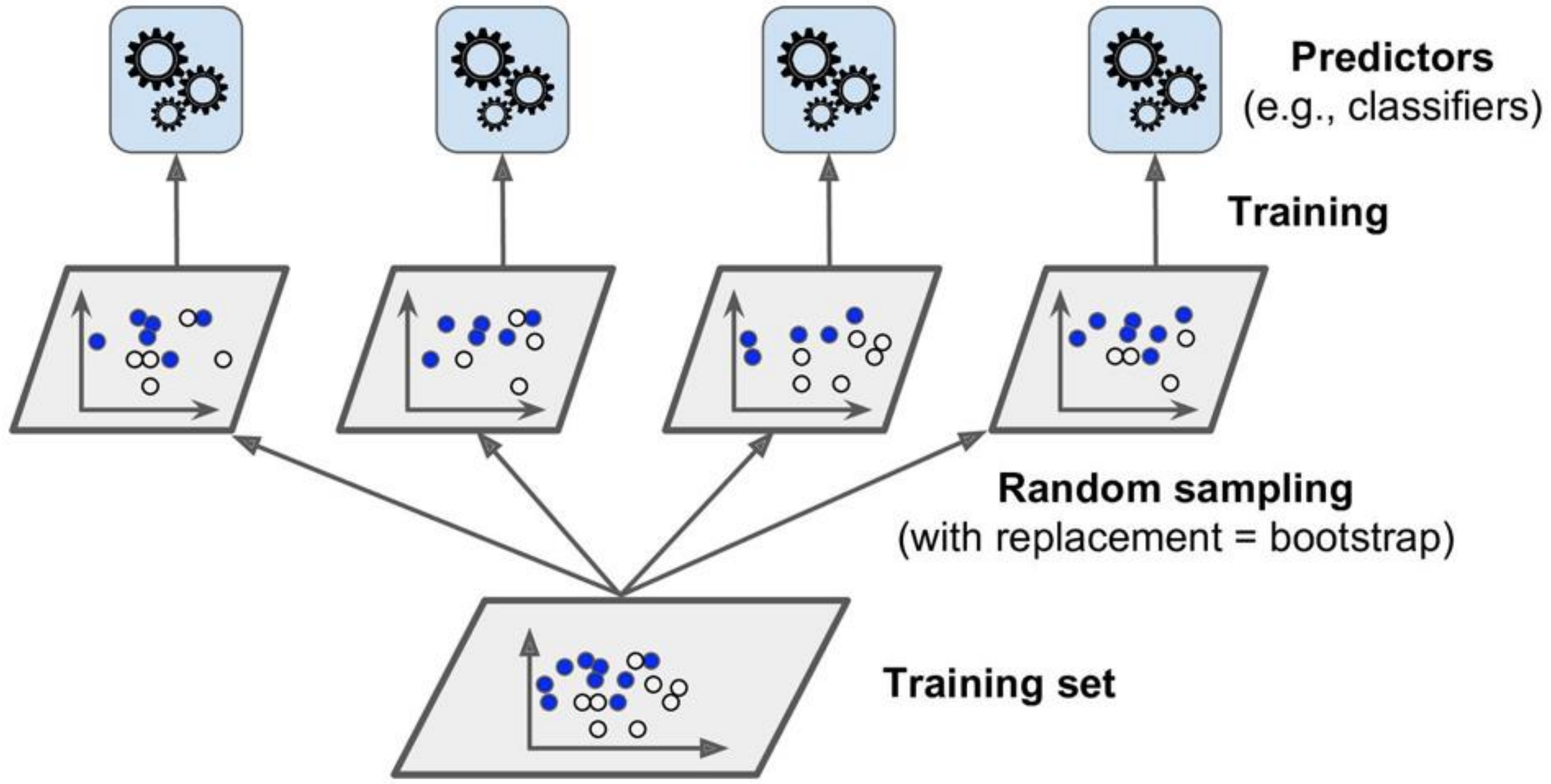
- <u>Wisdom of the Crowd</u>
- The average from many predictors may be more accurate any single given predictor
- Even if individual predictors are weak (only slightly better than random), an ensemble can be strong (accurate).
- In machine learning, group of predictors called an ensemble

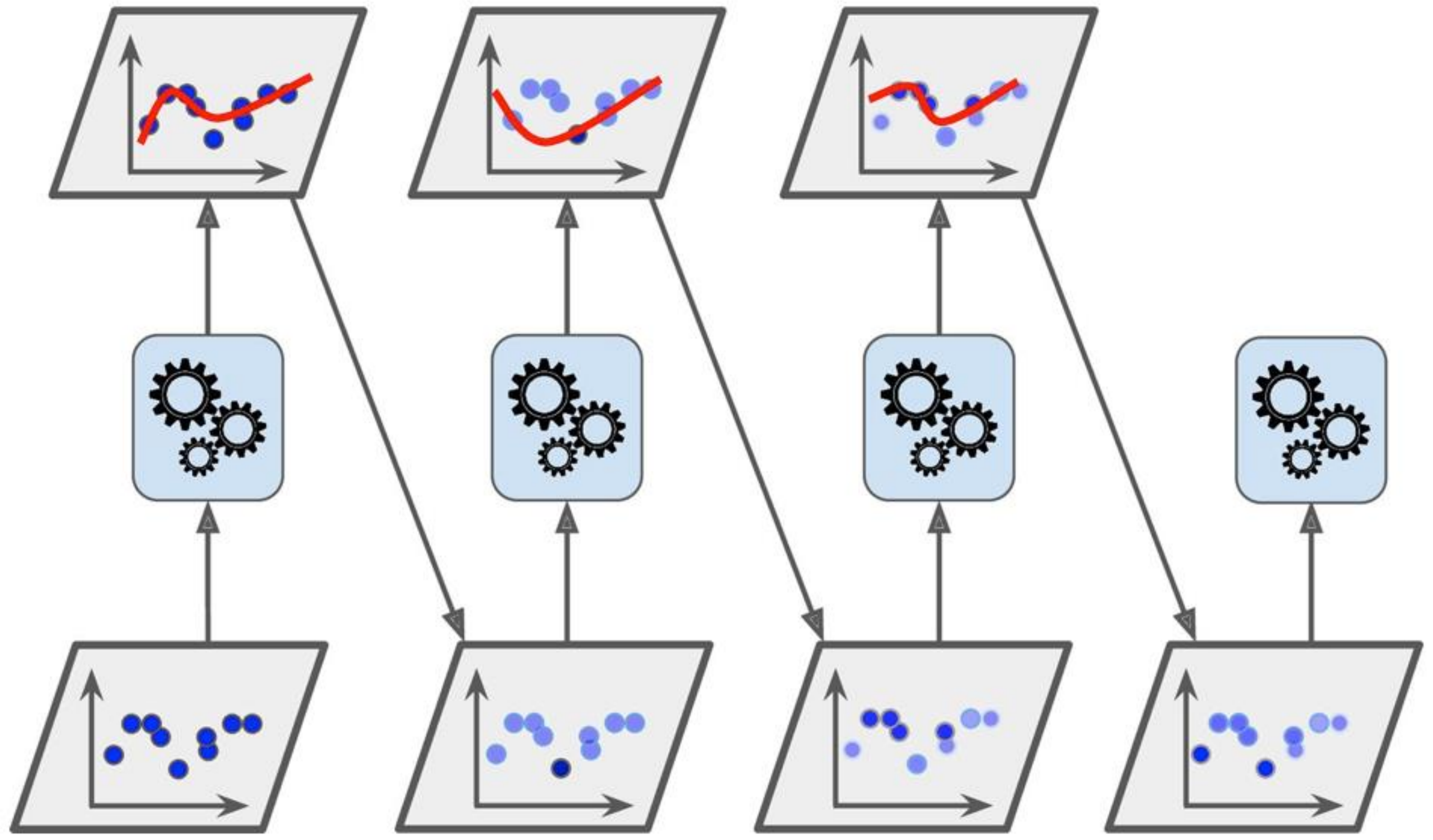


Ensemble learning

- CART is a good unit for ensemble learning
  - Training a CART is relatively easy and cheap
  - CART makes no assumptions on input data

- Two common approaches of ensemble learning
  - Bagging (random forest)
  - Boosting (gradient boosting decision tree, GBDT)



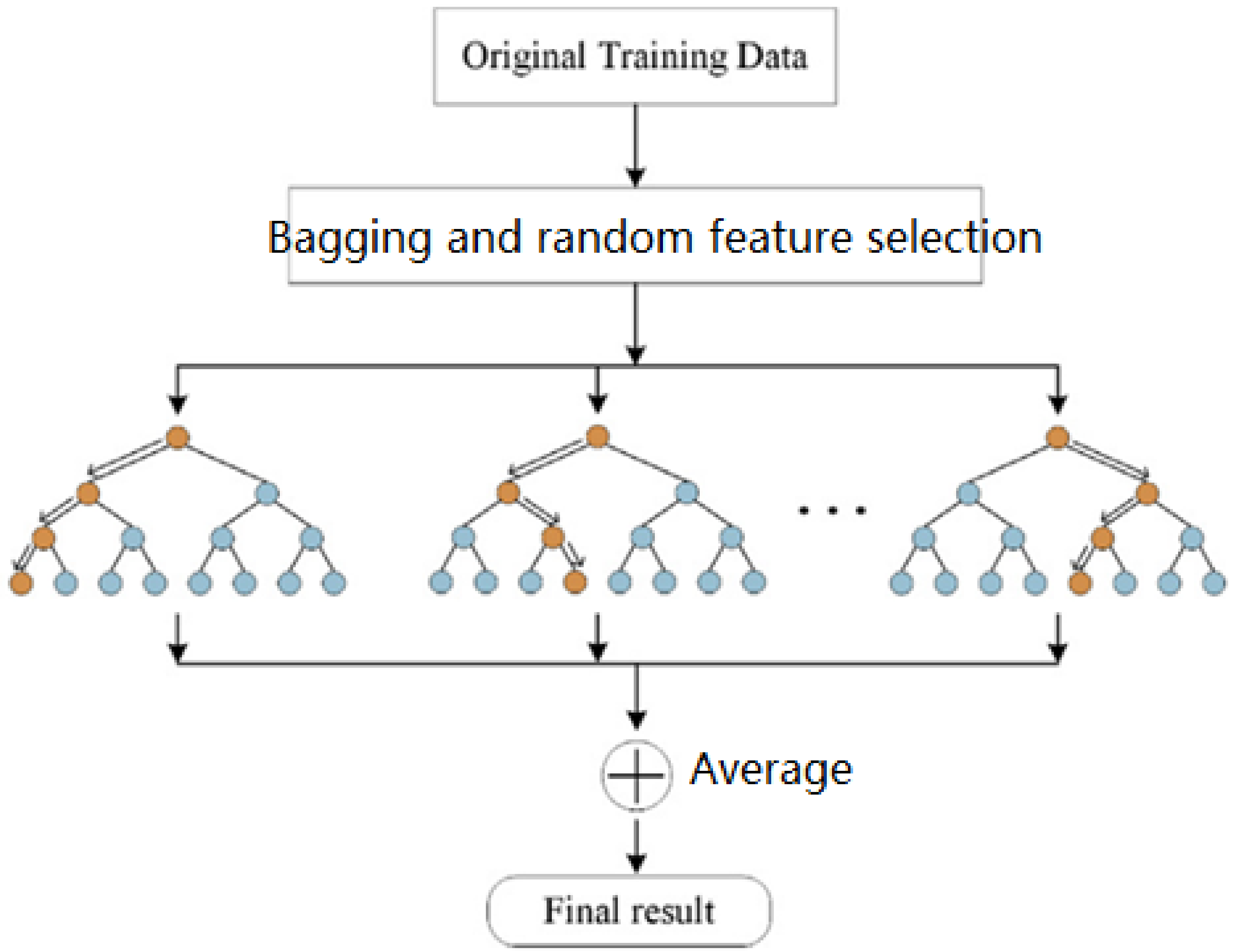


Random forest

Random Forest

- RF is a collection of many different CARTs.

- Given an input, the prediction of RF is a combination (e.g. average or the majority votes) of the output of all trees.



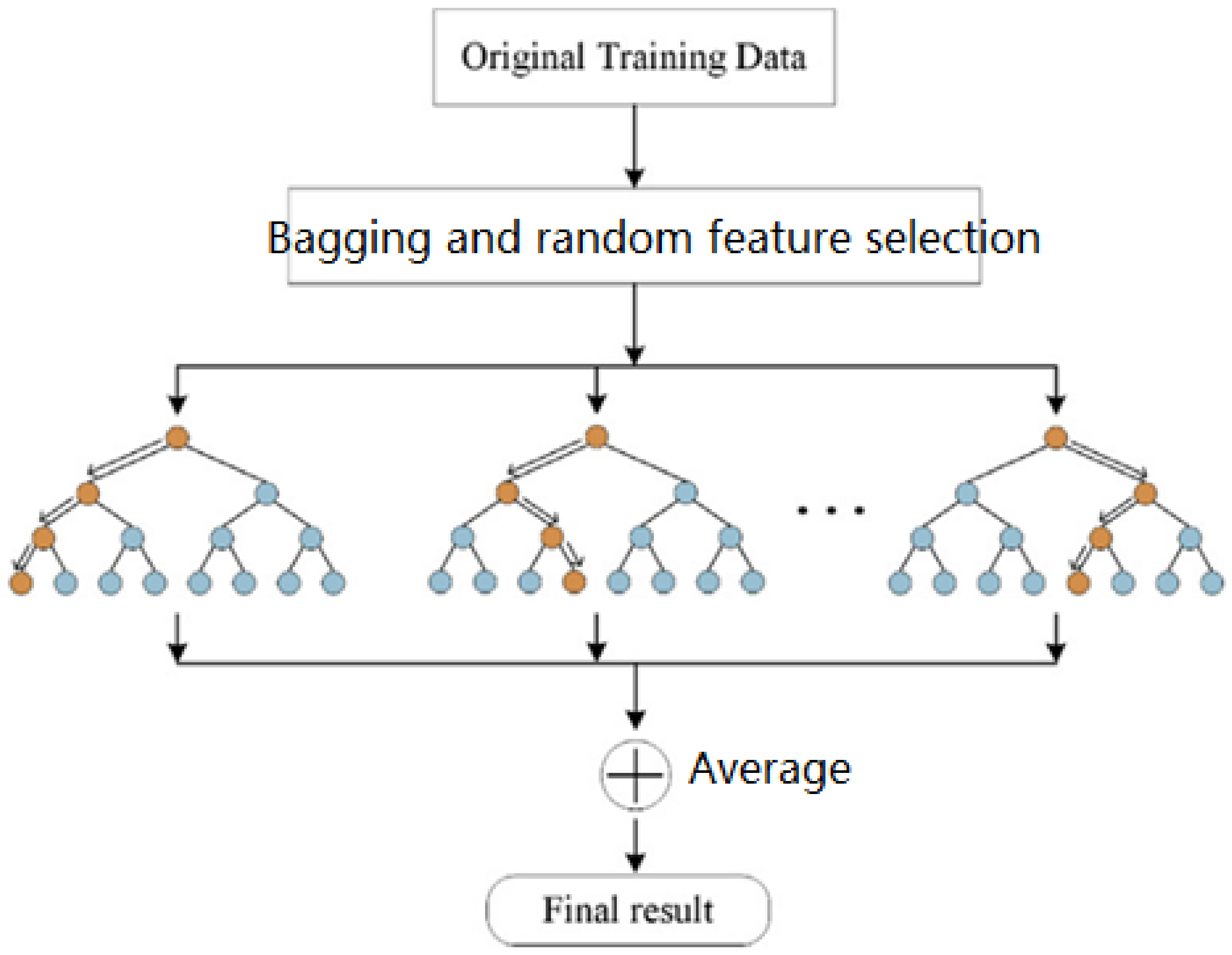
Amended from image source

Random Forest

- Two techniques to grow different and diverse trees (the beauty of randomness)

1. Bagging (short for bootstrap aggregating): sampling instances ('rows')
2. Random feature selection: sampling features ('columns')

- As each CART sees different training data, the trees are different.



Amended from image source

Random Forest

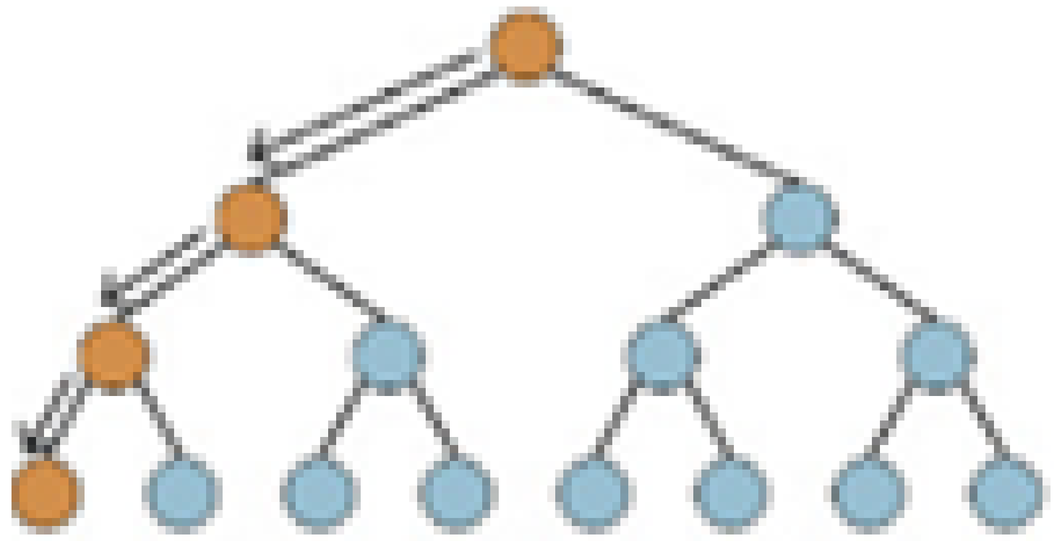
- Bootstrap: sampling with replacement. It guarantees that the sample has the same distribution as population; some instances may be sampled repeatedly.
- Example of bagging and random feature selection

Index	x1	x2	x3
1	2	1.0	2
2	3	1.5	3
3	5	2.0	4
4	4	2.6	6

Sampling:
2 features,
7 samples

Index	x1	x3
1	2	2
2	3	3
3	5	4
4	4	6
2	3	3
4	4	6
2	3	3

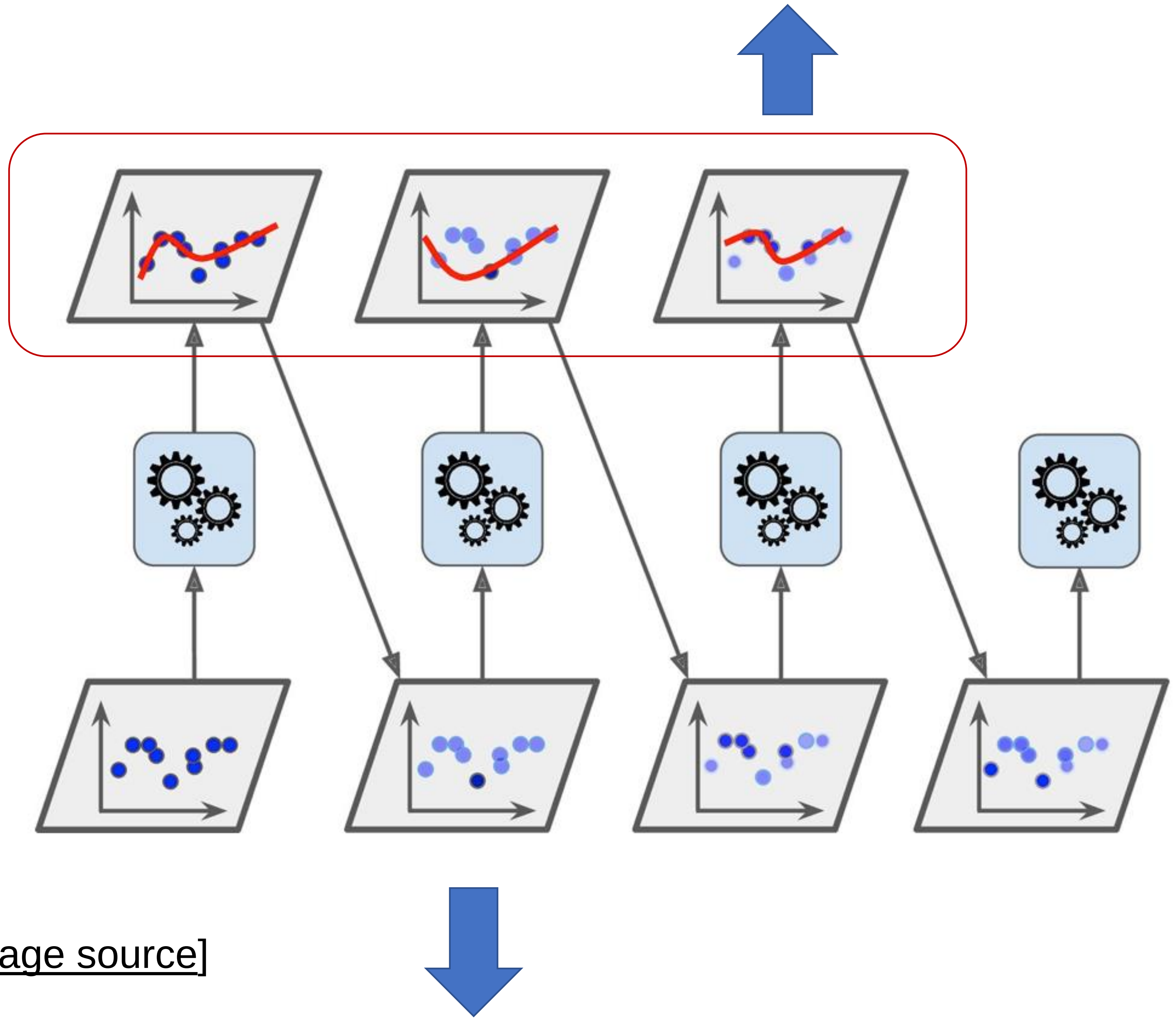
Train a CART



GBDT

GBDT

GBDT predictor



[image source]

A deeper colour means larger residual and then larger weight for the next predictor

- While RF grows trees horizontally (or in parallel), GBDT grows trees vertically (or sequentially)
- A new CART predictor is trained using the residual from the last CART as the weight. It focuses on the inaccurate prediction (with larger residual).
- All trees are combined to form the ensemble (similar to RF)

GBDT

Implementations

- GradientBoostingRegressor from sklearn
  - Good for small projects, but not scalable
- XGBoost (another package)
  - Efficient, robust, Industry-level implementation of GBDT
  - Winner of many data science competitions
  - Highly recommended
- Machine learning = theory + engineering

RF and GBDT

- Advantages
  - No assumptions on data distribution
  - Able to model non-linear relationship and feature interactions
  - Good predictive performance (especially for tabular data)
  - Good generalisation

- Disadvantages
  - Low interpretability: not intuitive, although there are some interpretation methods

Model interpretation

Interpreting ML models

- ‘Interpretation of ML models’ is an emerging field and there are many new methods coming out every year.
- Why is it important –
  - Many ML models are black-box models
  - “The problem is that a single metric, such as classification accuracy, is an incomplete description of most real-world tasks.” (Doshi-Velez et al.)
  - Some real-world tasks require safety measures and testing
  - We need to detect and understand bias in ML models
- One of the classic methods for interpreting tree-based models is <u>permutation feature importance</u>

Doshi-Velez, Finale, and Been Kim. “Towards a rigorous science of interpretable machine learning,” no. Ml: 1–13. http://arxiv.org/abs/1702.08608 (2017).

Permutation feature importance (PFI)

- The idea is straightforward. We measure the importance of a feature by calculating the increase in the model's prediction error after permuting the feature.
- A feature is "important" if shuffling its values increases the model error, because in this case the model relied on the feature for the prediction.
- In contrast, a feature is "unimportant" if shuffling its values leaves the model error unchanged, because in this case the model ignored the feature for the prediction.
- This method is model-agnostic
  - Applicable to linear regression, CART, RF, GBDT, etc.
  - Applicable to regression and classification task

Permutation feature importance (PFI)

- The idea is straightforward. We measure the importance of a feature by calculating the increase in the model's prediction error after permuting the feature.
- A feature is "important" if shuffling its values increases the model error, because in this case the model relied on the feature for the prediction.
- In contrast, a feature is "unimportant" if shuffling its values leaves the model error unchanged, because in this case the model ignored the feature for the prediction.
- This method is model-agnostic
  - Applicable to linear regression, CART, RF, GBDT, etc.
  - Applicable to regression and classification task

Permutation feature importance (PFI)

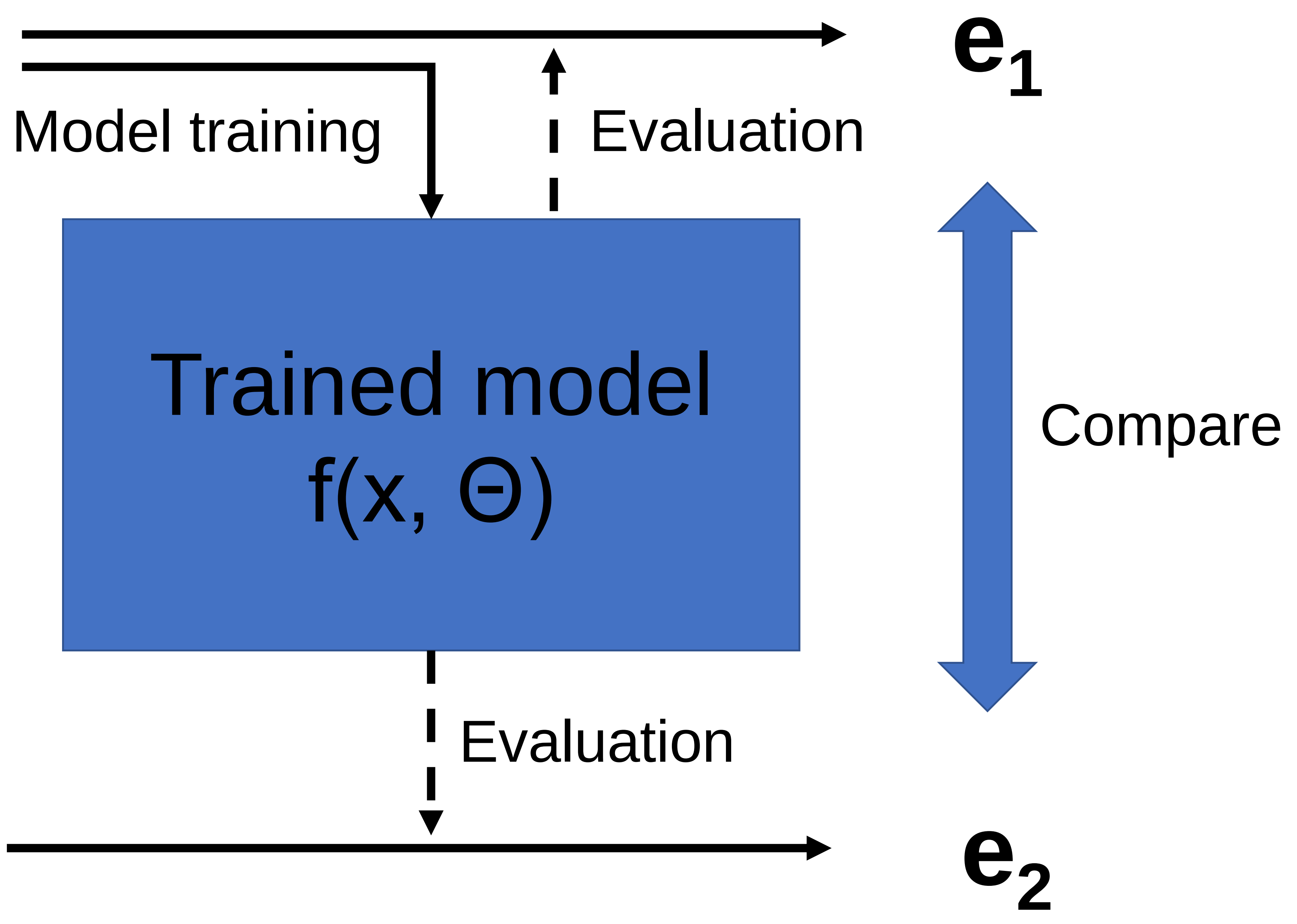
x1	x2	x3
2	1.0	2
3	1.5	3
5	2.0	4
4	2.6	6

y
12
15
16
20

Shuffle x_1

x1'	x2	x3
3	1.0	2
5	1.5	3
4	2.0	4
2	2.6	6

y
12
15
16
20



1. Train the model, and estimate the error on the dataset: $e_1=L(y, f([x_1,x_2,x_3])$
2. Shuffle x_1 and get a new dataset $[x_1',x_2,x_3]$
3. Re-estimate the error on the shuffled data $e_2=L(y, f([x_1',x_2,x_3])$
4. The PFI of x_1 is the difference between e_2 and e_1.
5. Repeat Step 3-4 for x_2 and x_3. Then, you can rank x_1, x_2, x_3 from the most important to least.

Other interpretation

- Other types of feature importance, such as Gini importance for RF, standardised coefficients for regression. Note that some feature importance measures are model-specific, e.g. only applicable for regression
- Partial dependence plot shows the marginal effect one or two features have on the predicted outcome of a ML model
- Section 8.1 and 8.5 of this book: https://christophm.github.io/interpretable-ml-book/

Summary

- Basics of CART for regression and classification
- The idea of ensemble learning
- Random forest and GBDT (XGBoost): two primary ensemble learning methods based on CART
- Interpretation of tree-based models: permutation feature importance

Workshop

- Weekly quiz on Moodle: please finish them before the workshop and we will discuss the quiz in the workshop
- Python notebooks for workshop: will be ready by 5pm Thursday.
- See you in the workshop on Friday 1-3pm