emata

Don’t Get Lost in the Random Forests: A Beginner’s Guide

Moses Bomera, Emata

April 29th, 2021

Moses Bomera

Data Scientist
Emata (& Laboremus)

Works on alternative credit scoring for microfinance institution Emata, a spin-off of fintech company Laboremus.

Prior to joining Laboremus, worked on research in computer vision and natural language processing at netLabs!UG Research Centre, Makerere University.

We are on a mission to provide affordable digital loans to millions of farmers in East Africa

Today, agri-financing in Africa does NOT work.

Farmers are risky clients

NO (CREDIT) HISTORY,
NO COLLATERAL

Farmers need small loans

HIGH OPERATING COSTS

50-100%

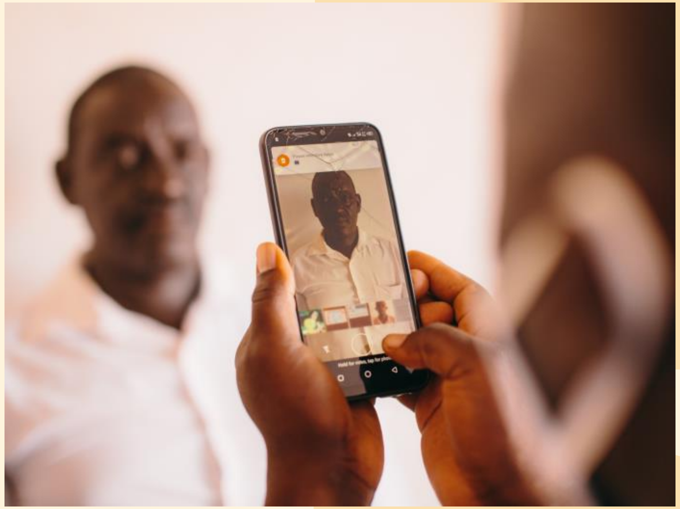
ANNUAL
INTEREST RATE

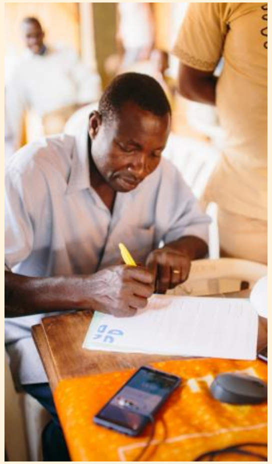
What does Emata do?

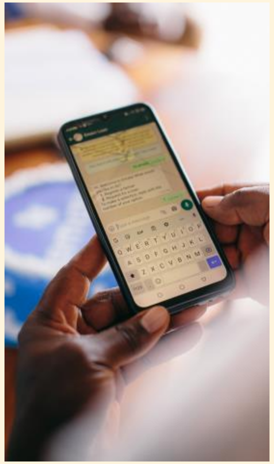
1. Digitise agricultural cooperatives to get data

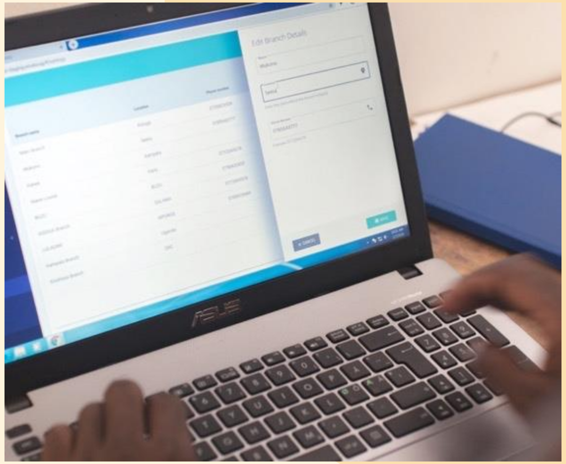
2. Turn data into credit scoring

3. Offer digital and affordable loans to farmers









Edit Branch Details

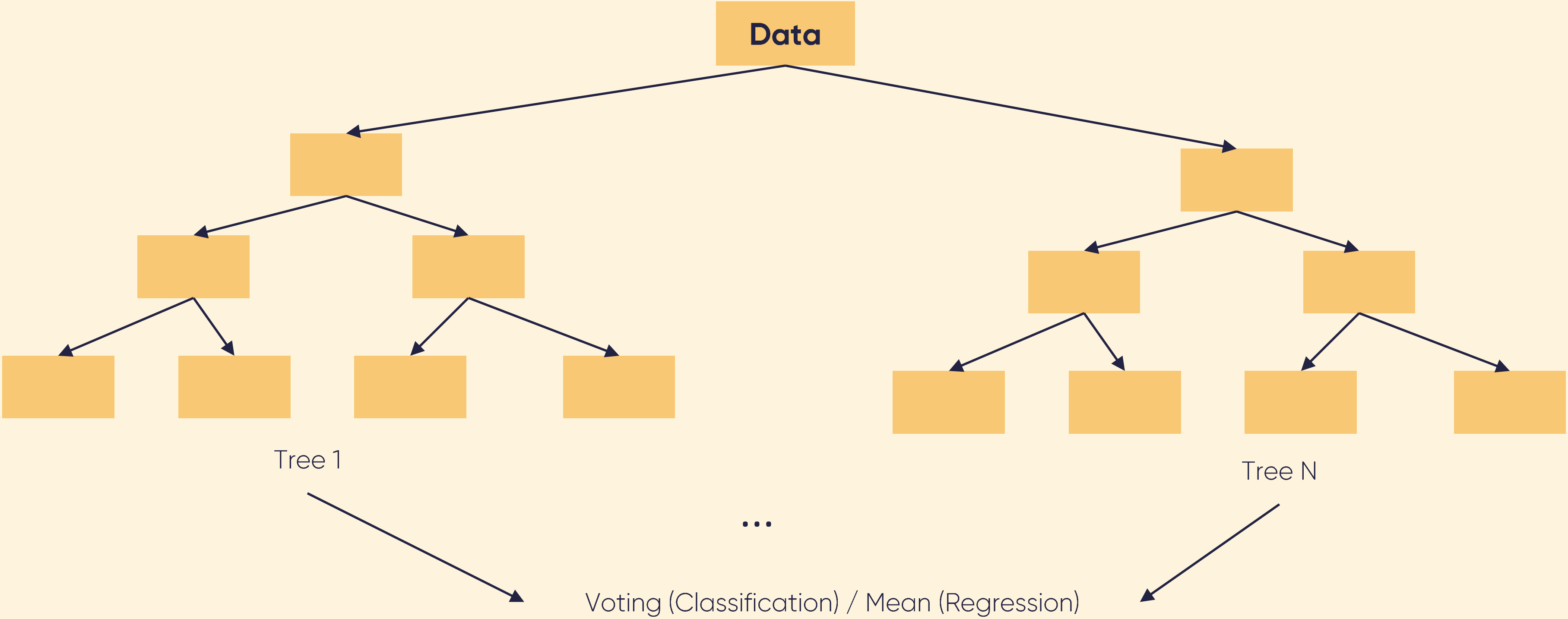
Agenda

- What is a Random Forest?
- Principal component – A Decision Tree
- Calculating the gini value
- The CART Training Algorithm
- Regression
- Gini impurity or Entropy?
- How does it work?
- Regularization Hyperparameters
- TLDR
- Reference Material
- Tutorial

What is a Random Forest?

What is a Random Forest?

It is an ensemble of decision trees.



What have Random Forests been used for?

Classification, regression, time series forecasting etc.

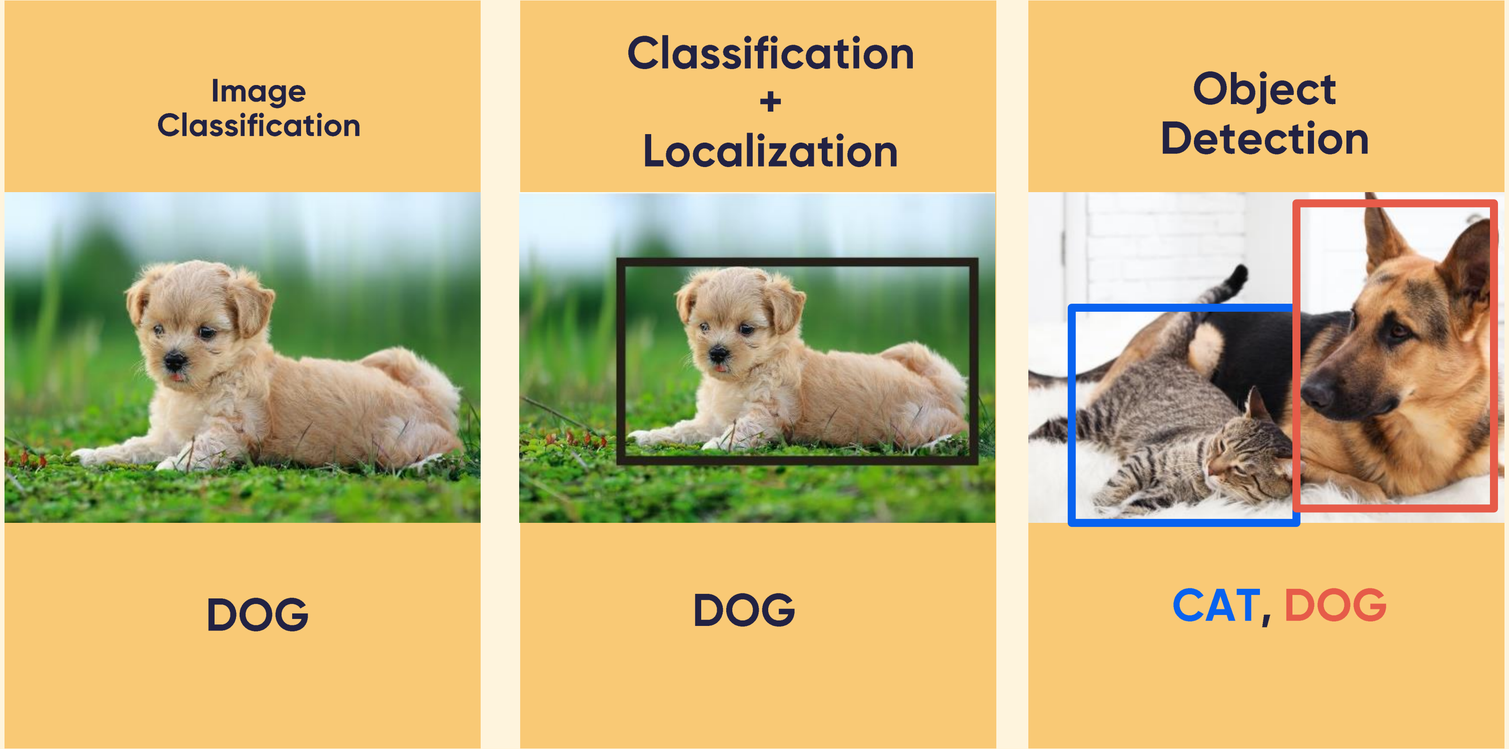


Image Classification

DOG

Classification + Localization

DOG

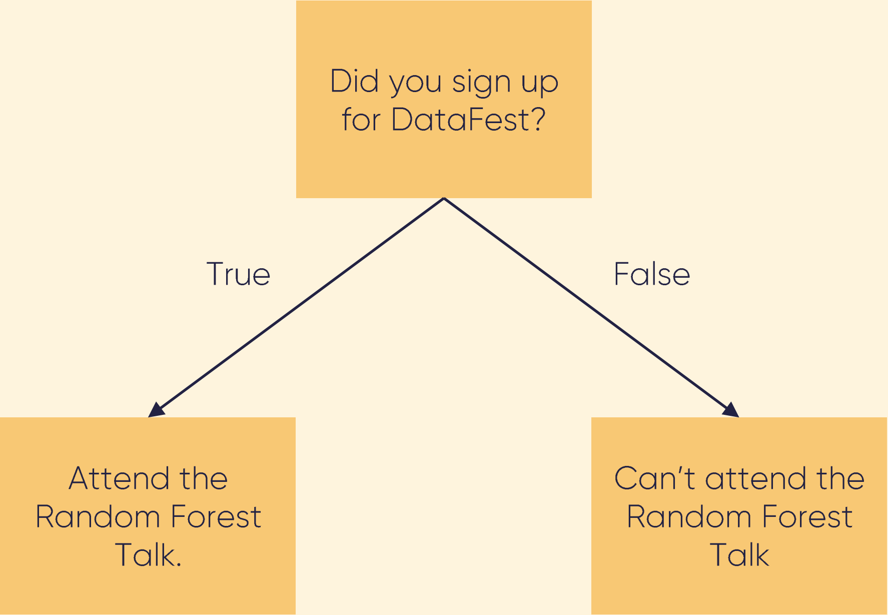
Object Detection

CAT, DOG

Photo source: morioh.com

Random Forests

A decision tree (1/3)



Did you sign up for DataFest?

True

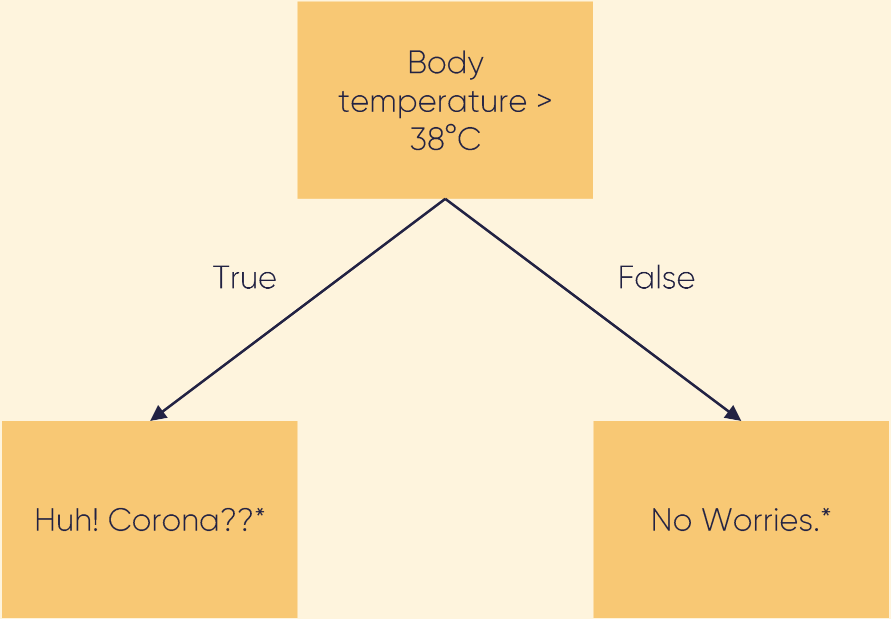
False

Attend the Random Forest Talk.

Can’t attend the Random Forest Talk

Random Forests

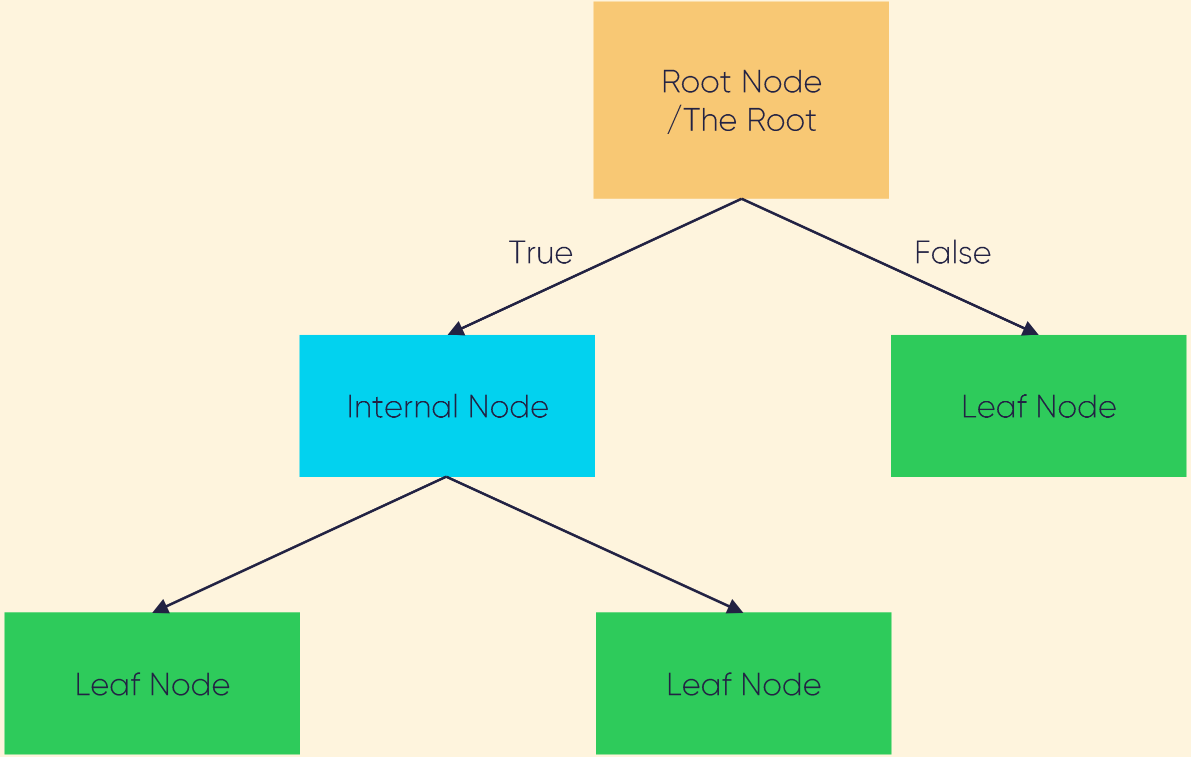
A decision tree (2/3)



*For tutorial purpose only.

Random Forests

A decision tree (3/3)



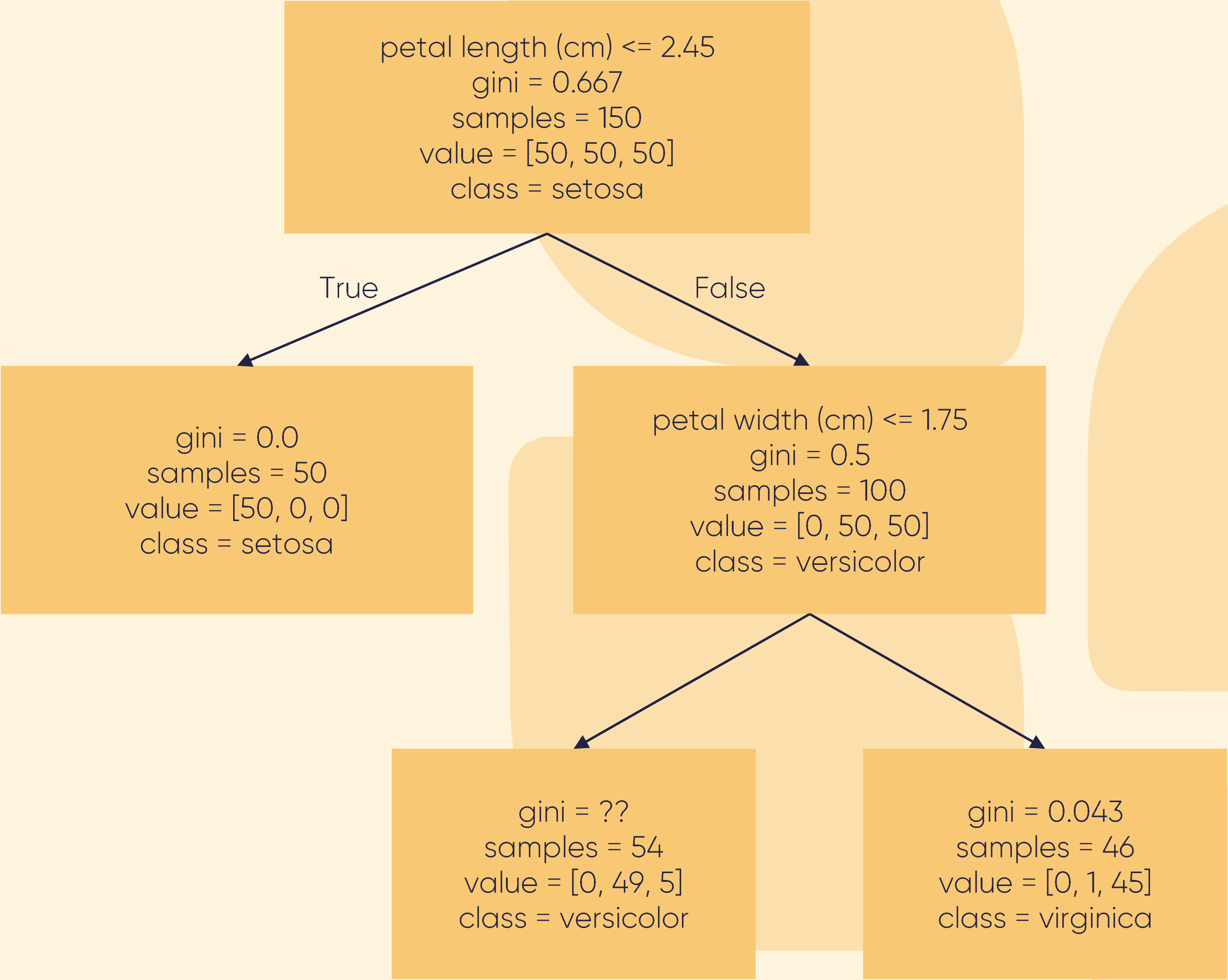
Dataset

petal length	petal width	target
1.4	0.2	setosa
1.3	0.2	setosa
...	...	...
5.1	1.8	virginica

Iris dataset

Left; Iris setosa,
Top; Iris virginica,
Bottom; Iris veriscolor

The Theory


petal length (cm) <= 2.45
gini = 0.667
samples = 150
value = [50, 50, 50]
class = setosa

True

False

gini = 0.0
samples = 50
value = [50, 0, 0]
class = setosa

petal width (cm) <= 1.75
gini = 0.5
samples = 100
value = [0, 50, 50]
class = versicolor

gini = ??
samples = 54
value = [0, 49, 5]
class = versicolor

gini = 0.043
samples = 46
value = [0, 1, 45]
class = virginica

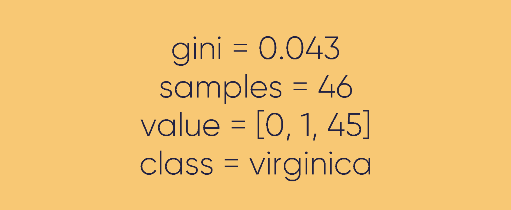
Node Features

Samples: the number of training instances that particular node applies to.

Gini: measures how pure (or impure) a node is.

Class: the target represented by a given node.

$p_{i,k}$ is the ratio of class k instances to all the training instances in i^{th} node.



gini = 0.043
samples = 46
value = [0, 1, 45]
class = virginica

$$G_i = 1 - \sum_{k=1}^{n} {p_{i,k}}^2$$

Calculating the gini value

gini = ??
samples = 54
value = [0, 49, 5]
class = versicolor

$$G_i = 1 - \sum_{k=1}^{n} {p_{i,k}}^2$$

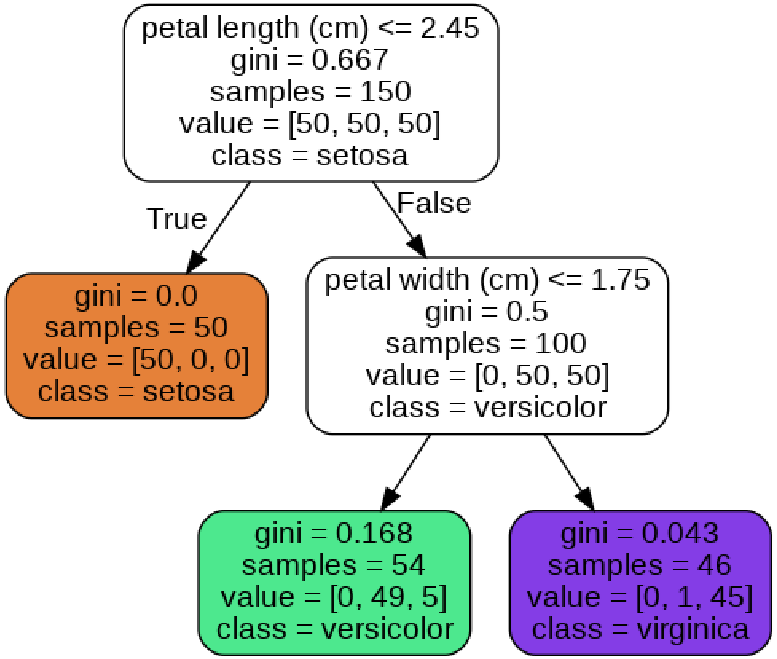
$$1 - \left(\frac{0}{54}\right)^2 - \left(\frac{49}{54}\right)^2 - \left(\frac{5}{54}\right)^2 =$$

Classification and Regression Tree (CART) Algorithm

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

$$\text{where } \begin{cases} G_{\text{left/right}} \text{ measures the impurity of the left/right subset,} \\ m_{\text{left/right}} \text{ is the number of instances in the left/right subset.} \end{cases}$$

1. CART splits the training set into two subsets using a single feature, k and a threshold purity, t_k.
2. The selection process involves minimizing the cost function
3. 1 and 2 are repeated recursively for each subset until the maximum depth is reached.



petal length (cm) <= 2.45
gini = 0.667
samples = 150
value = [50, 50, 50]
class = setosa

True

False

gini = 0.0
samples = 50
value = [50, 0, 0]
class = setosa

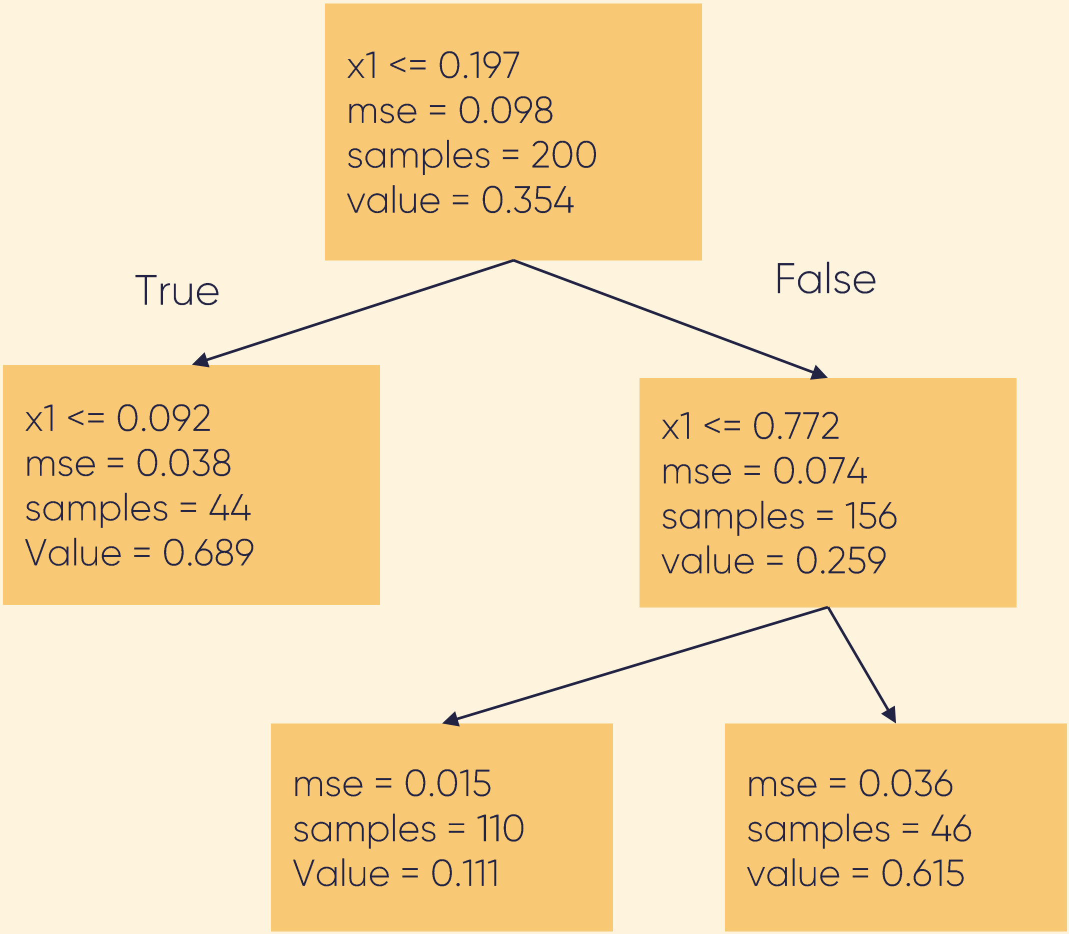
petal width (cm) <= 1.75
gini = 0.5
samples = 100
value = [0, 50, 50]
class = versicolor

gini = 0.168
samples = 54
value = [0, 49, 5]
class = versicolor

gini = 0.043
samples = 46
value = [0, 1, 45]
class = virginica

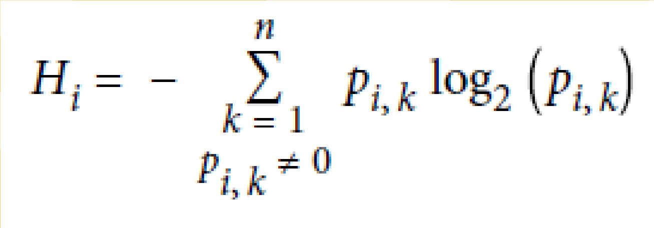
Regression

- Similar to the classification approach.
- The prediction is the average of the samples associated with the leaf node.
- For the gini impurity split, the regression tree uses mean squared error i.e. attempts to minimize the mse.



Gini impurity or Entropy?

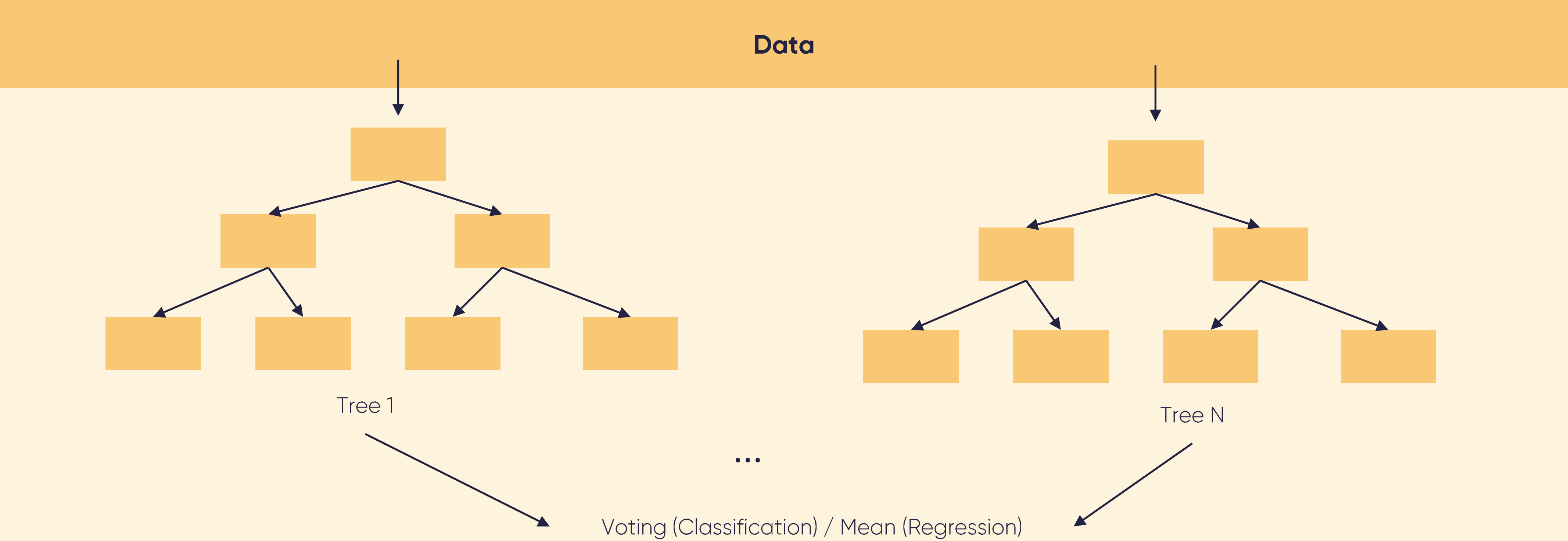
- **While gini is the go-to impurity measure**, you can also use entropy as a measurement.
- Entropy is based on the same concept from thermodynamics.
- Entropy approaches zero when molecules are still and well ordered.
- In machine learning, entropy is zero when a set contains instances of only one class.



$$H_i = - \sum_{\substack{k=1 \\ p_{i,k} \neq 0}}^{n} p_{i,k} \log_2 \left(p_{i,k} \right)$$

What is a Random Forest?

It is an ensemble of decision trees.



Data

Tree 1

...

Tree N

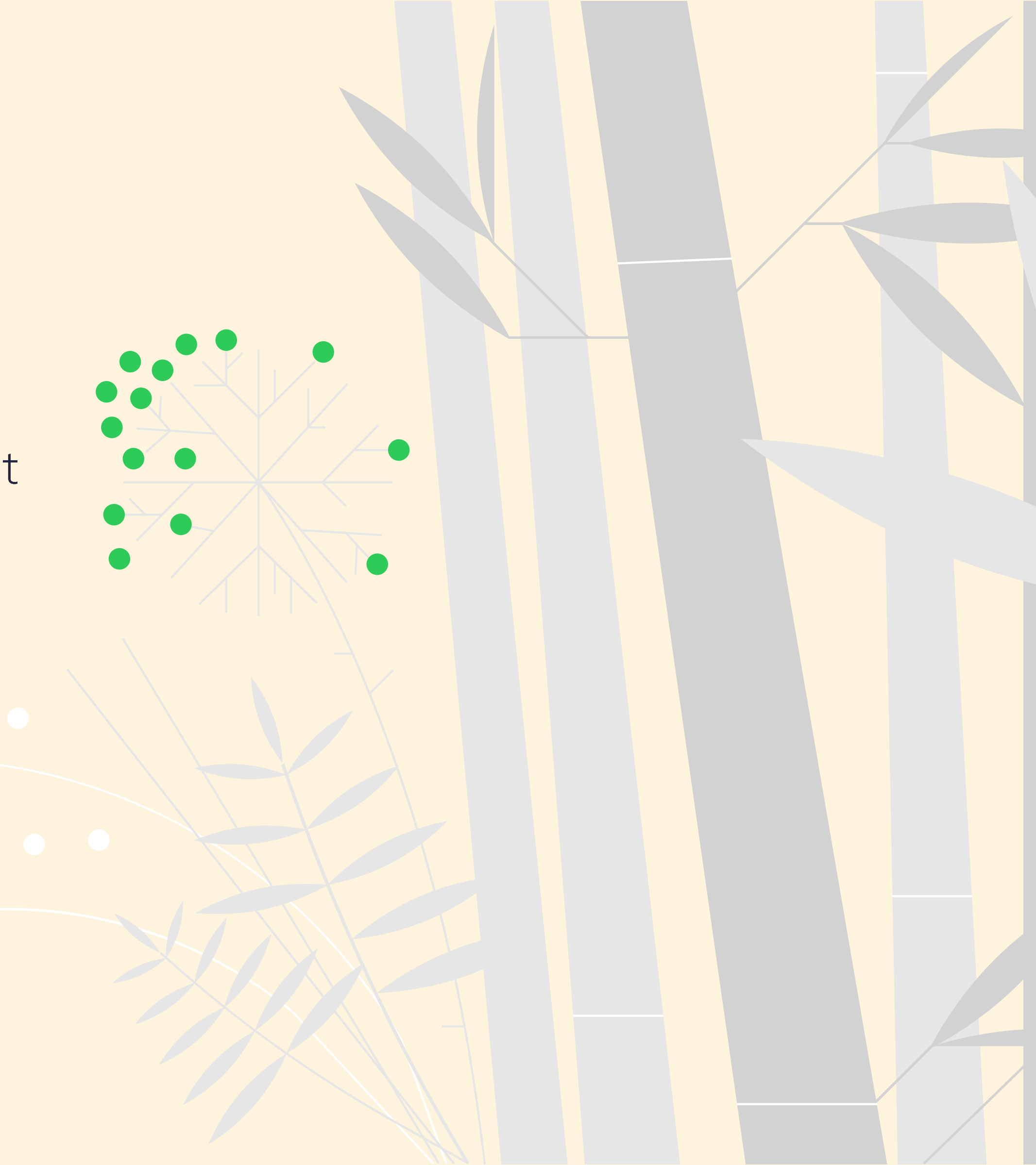
Voting (Classification) / Mean (Regression)

How does it work? (1/2)

- A random forest is an ensemble of decision trees.
- Trained usually through the bagging method (or pasting).
- Each tree in a random forest gives a prediction, for classification, the class with the majority of votes is the prediction.
- In regression, the average of each tree's prediction is the prediction.

How does it work? (2/2)

- RF introduces randomness when growing trees.
- Doesn't make a split using the very best feature, instead it selects the best feature among a random subset of features.
- Results in greater tree diversity.
- Trading a higher bias with lower variance yielding a better model than an individual decision tree.



Regularization Hyperparameters

- *n_estimators* - the number of trees to use.
- *max_depth* – the depth of the tree.
- *min_samples_split* - the minimum number of samples an internal node must have before it can be split.
- *min_samples_leaf* - the minimum number of samples required to be at a leaf node
- *max_leaf_nodes* - the maximum number of leaf nodes.
- *max_features* - the maximum number of features that are evaluated for splitting at each node.

Increasing *min_** hyperparameters or reducing *max_** hyperparameters will regularize the model.

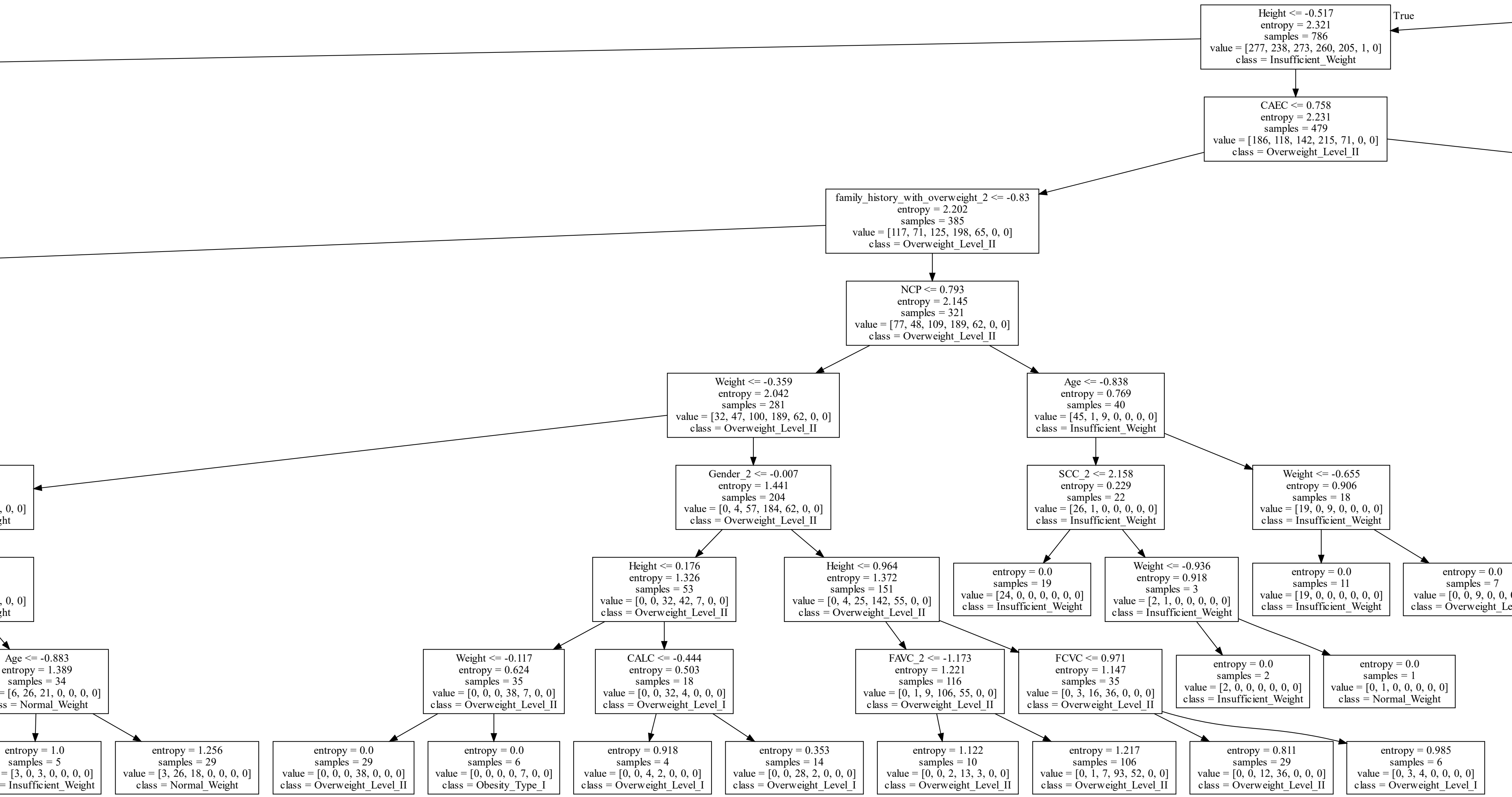
TLDR;

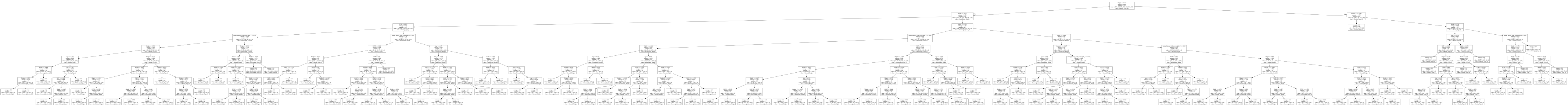
PROS of using Random Forests

- Random Forests can be used for both classification and regression.
- Can handle large datasets with high dimensionality, (can be used for dimensionality reduction).
- Works well with unscaled datasets.

CONS of using Random Forests

- It is a high variance model, so it is important to regularize the model when training.
- Given the numerous hyper-parameters, and the number of trees, the decision process becomes a bit of a black box.


True
Height <= -0.517
entropy = 2.321
samples = 786
value = [277, 238, 273, 260, 205, 1, 0]
class = Insufficient_Weight
CAEC <= 0.758
entropy = 2.231
samples = 479
value = [186, 118, 142, 215, 71, 0, 0]
class = Overweight_Level_II
family_history_with_overweight_2 <= -0.83
entropy = 2.202
samples = 385
value = [117, 71, 125, 198, 65, 0, 0]
class = Overweight_Level_II
NCP <= 0.793
entropy = 2.145
samples = 321
value = [77, 48, 109, 189, 62, 0, 0]
class = Overweight_Level_II
Weight <= -0.359
entropy = 2.042
samples = 281
value = [32, 47, 100, 189, 62, 0, 0]
class = Overweight_Level_II
Age <= -0.838
entropy = 0.769
samples = 40
value = [45, 1, 9, 0, 0, 0, 0]
class = Insufficient_Weight
, 0, 0]
ght
Gender_2 <= -0.007
entropy = 1.441
samples = 204
value = [0, 4, 57, 184, 62, 0, 0]
class = Overweight_Level_II
SCC_2 <= 2.158
entropy = 0.229
samples = 22
value = [26, 1, 0, 0, 0, 0, 0]
class = Insufficient_Weight
Weight <= -0.655
entropy = 0.906
samples = 18
value = [19, 0, 9, 0, 0, 0, 0]
class = Insufficient_Weight
, 0, 0]
ght
Height <= 0.176
entropy = 1.326
samples = 53
value = [0, 0, 32, 42, 7, 0, 0]
class = Overweight_Level_II
Height <= 0.964
entropy = 1.372
samples = 151
value = [0, 4, 25, 142, 55, 0, 0]
class = Overweight_Level_II
entropy = 0.0
samples = 19
value = [24, 0, 0, 0, 0, 0, 0]
class = Insufficient_Weight
Weight <= -0.936
entropy = 0.918
samples = 3
value = [2, 1, 0, 0, 0, 0, 0]
class = Insufficient_Weight
entropy = 0.0
samples = 11
value = [19, 0, 0, 0, 0, 0, 0]
class = Insufficient_Weight
entropy = 0.0
samples = 7
value = [0, 0, 9, 0, 0, 0
class = Overweight_Le
Age <= -0.883
entropy = 1.389
samples = 34
= [6, 26, 21, 0, 0, 0, 0]
ss = Normal_Weight
Weight <= -0.117
entropy = 0.624
samples = 35
value = [0, 0, 0, 38, 7, 0, 0]
class = Overweight_Level_II
CALC <= -0.444
entropy = 0.503
samples = 18
value = [0, 0, 32, 4, 0, 0, 0]
class = Overweight_Level_I
FAVC_2 <= -1.173
entropy = 1.221
samples = 116
value = [0, 1, 9, 106, 55, 0, 0]
class = Overweight_Level_II
FCVC <= 0.971
entropy = 1.147
samples = 35
value = [0, 3, 16, 36, 0, 0, 0]
class = Overweight_Level_II
entropy = 0.0
samples = 2
value = [2, 0, 0, 0, 0, 0, 0]
class = Insufficient_Weight
entropy = 0.0
samples = 1
value = [0, 1, 0, 0, 0, 0, 0]
class = Normal_Weight
entropy = 1.0
samples = 5
= [3, 0, 3, 0, 0, 0, 0]
= Insufficient_Weight
entropy = 1.256
samples = 29
value = [3, 26, 18, 0, 0, 0, 0]
class = Normal_Weight
entropy = 0.0
samples = 29
value = [0, 0, 0, 38, 0, 0, 0]
class = Overweight_Level_II
entropy = 0.0
samples = 6
value = [0, 0, 0, 0, 7, 0, 0]
class = Obesity_Type_I
entropy = 0.918
samples = 4
value = [0, 0, 4, 2, 0, 0, 0]
class = Overweight_Level_I
entropy = 0.353
samples = 14
value = [0, 0, 28, 2, 0, 0, 0]
class = Overweight_Level_I
entropy = 1.122
samples = 10
value = [0, 0, 2, 13, 3, 0, 0]
class = Overweight_Level_II
entropy = 1.217
samples = 106
value = [0, 1, 7, 93, 52, 0, 0]
class = Overweight_Level_II
entropy = 0.811
samples = 29
value = [0, 0, 12, 36, 0, 0, 0]
class = Overweight_Level_II
entropy = 0.985
samples = 6
value = [0, 3, 4, 0, 0, 0, 0]
class = Overweight_Level_I



Tutorial

Tutorial Notebook

Reference Material

1. Chapter 6, 7 Hands-On Machine Learning with Scikit-Learn, Keras, and TensorFlow by Aurélien Géron
2. https://ieeexplore.ieee.org/abstract/document/598994 – Random Forest academic paper.
3. https://towardsdatascience.com/understanding-random-forest-58381e0602d2
4. https://github.com/kjw0612/awesome-random-forest - Projects that have used Random Forests to achieve amazing solutions.
5. https://www.quora.com/What-are-the-advantages-and-disadvantages-for-a-random-forest-algorithm - Discussions on the advantages and disadvantages of Random Forests
6. https://www.youtube.com/watch?v=7VeUPuFGJHk - StatQuest: Decision Trees
7. https://www.youtube.com/watch?v=J4Wdy0Wc_xQ - StatQuest: Random Forests Part 1 - Building, Using and Evaluating

Thank you!

Questions?

emata

The future of farmer financing

www.emata.ug