



Image processing for Earth Observation

3b semantic segmentation with random forests

Devis TUIA

EPFL, fall semester 2025

Content (6 weeks)

- W1 General concepts of image classification / segmentation
Traditional supervised classification methods (RF)
- W2 Traditional supervised classification methods (SVM)
Best practices
- W3 Elements of neural networks
- W4 Convolutional neural networks
- W5 Convolutional neural networks for semantic segmentation
- W6 Sequence modeling, change detection

What do ML models look like?

Phase 1: extract a feature representation
= extract relevant variables for the task at hand



Extract feature
representations
Last two
weeks!

Features?

Variables issued from the data that are more expressive to solve the problem

They are specific to the type of data:

- Images: textures, color gradients, ...
- Environmental: temperature, pressure, altitude gradients, ...
- Text: occurrences of words, length of words, ...
- Regions: administrative statistics, area, ...
- Road networks: length, number of intersections, typology, centrality measures, ...

What do these models look like?

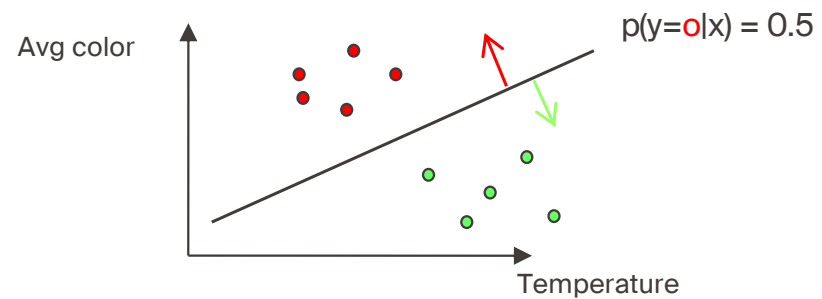
Phase 2: training, (learning)
= using a series of X / Y pairs, learn how to relate them



Extract
feature
representa-
tions

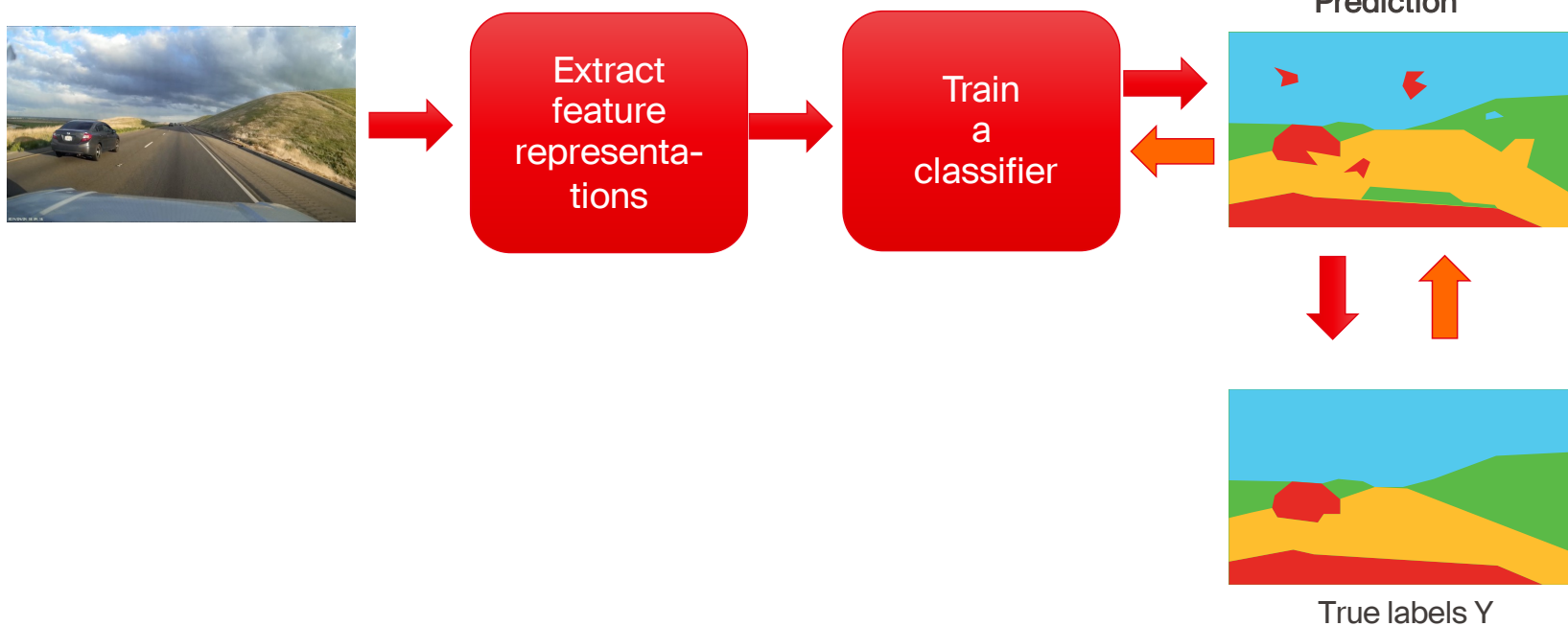
Train
a
classifier

- Max. Likelihood
- Random forest
- SVM
- Neural network



What do these models look like?

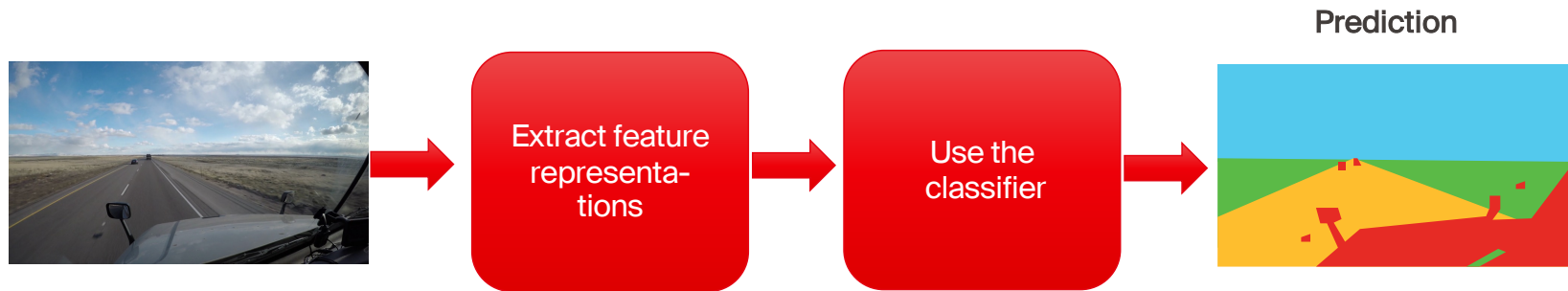
Phase 2: training, (learning)
= using a series of X / Y pairs, learn how to relate them



What do these models look like?

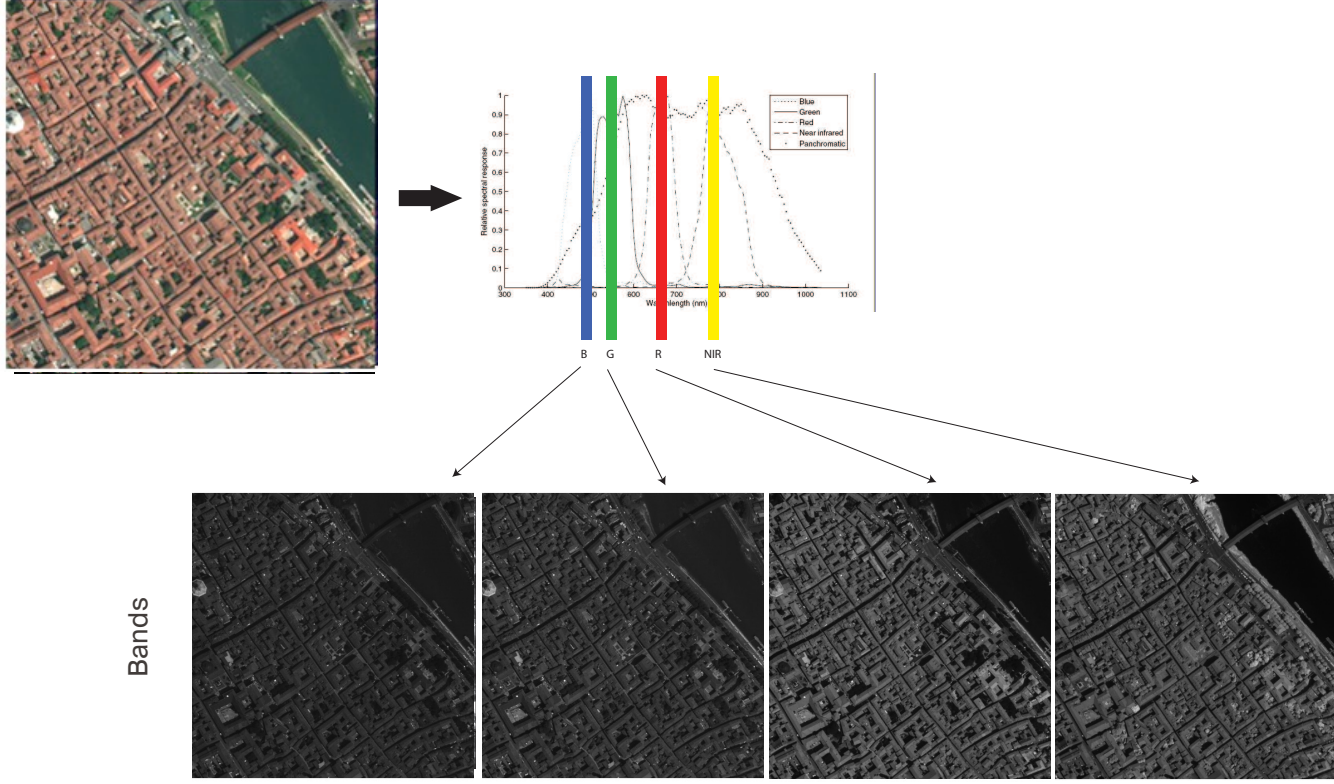
Phase 3: inference

= given a new image, you pass it through the model and retrieve the response



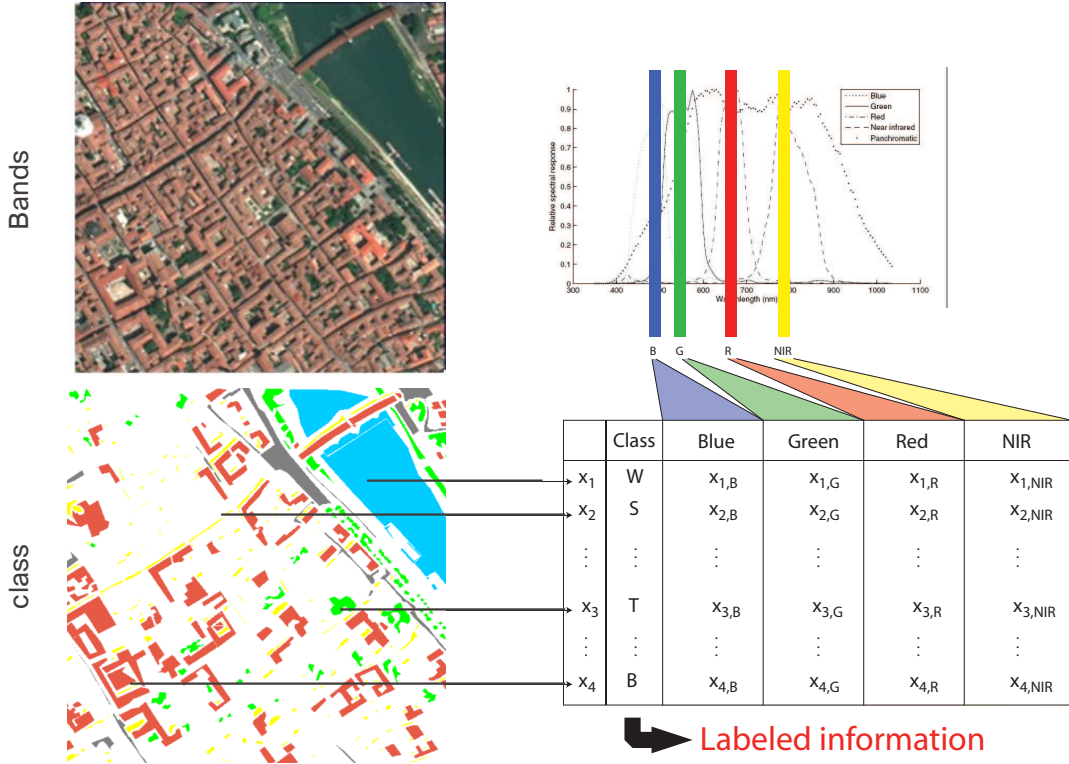
In this case, for each pixel it will attribute the most probable class
It remains a prediction, so it will still have errors!

Step by step: 1 – digital information

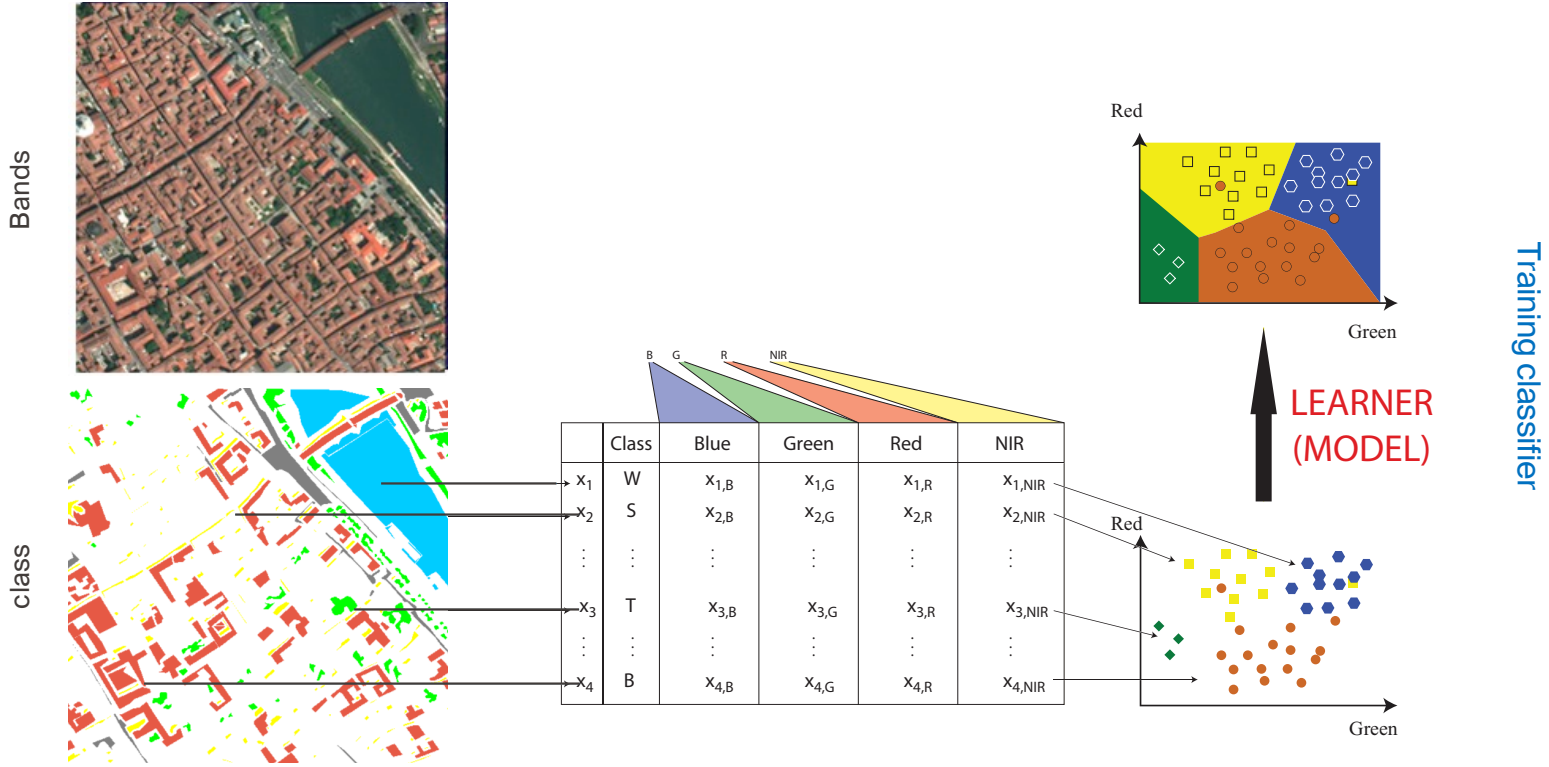


Step by step: 2 – labeled examples

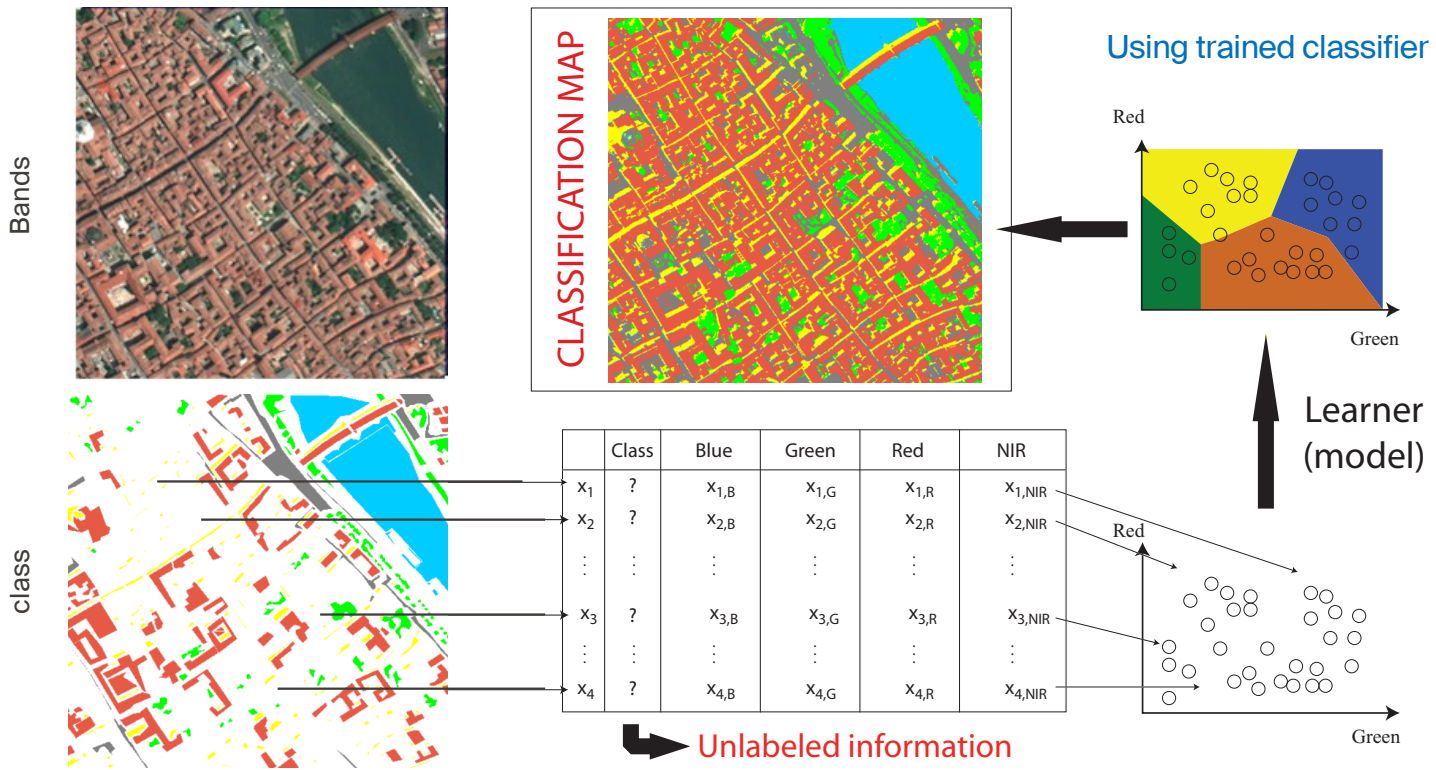
input-output pairs

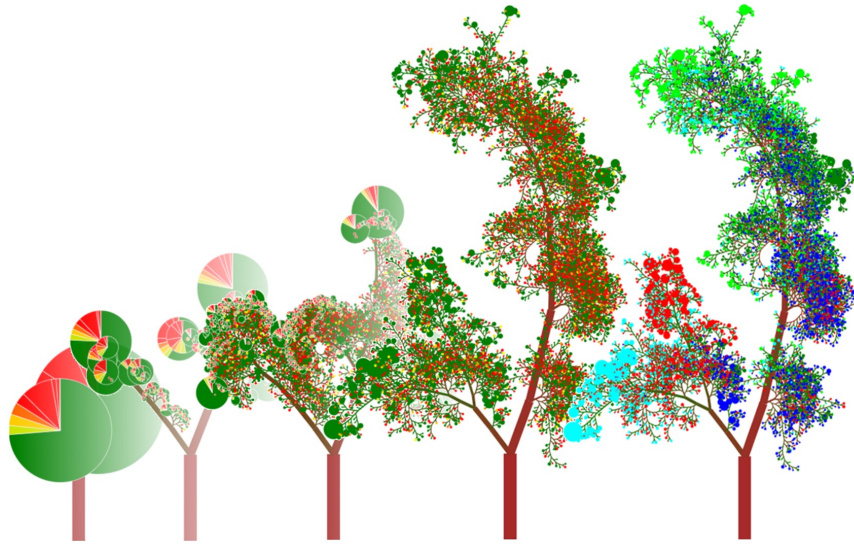


Step by step: 3 – build the model



Step by step: 4 – predict unseen data





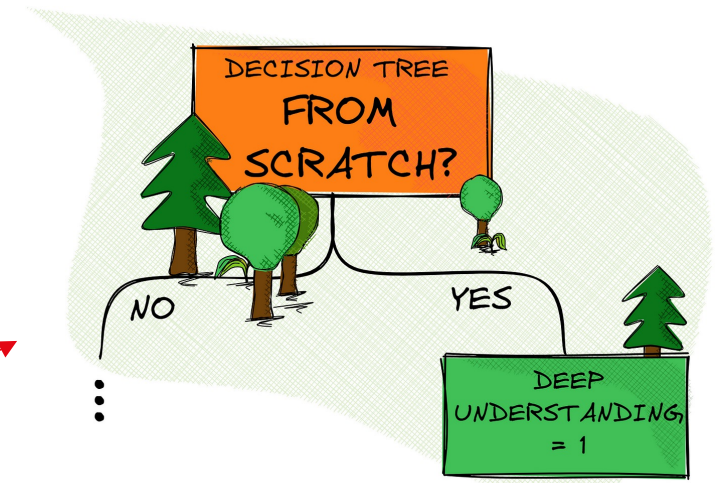
source: Rhaensch.de

Decision trees and random forests

Random forests

- It an ensemble model (it classifies every pixel according to the majority vote of many simple models)
- Each model is called a **decision tree**
- Imagine you had to classify all students in Europe into “**IPEO students**” or “**IPEO teaching team**”

how would you do it?



Source: towards data science



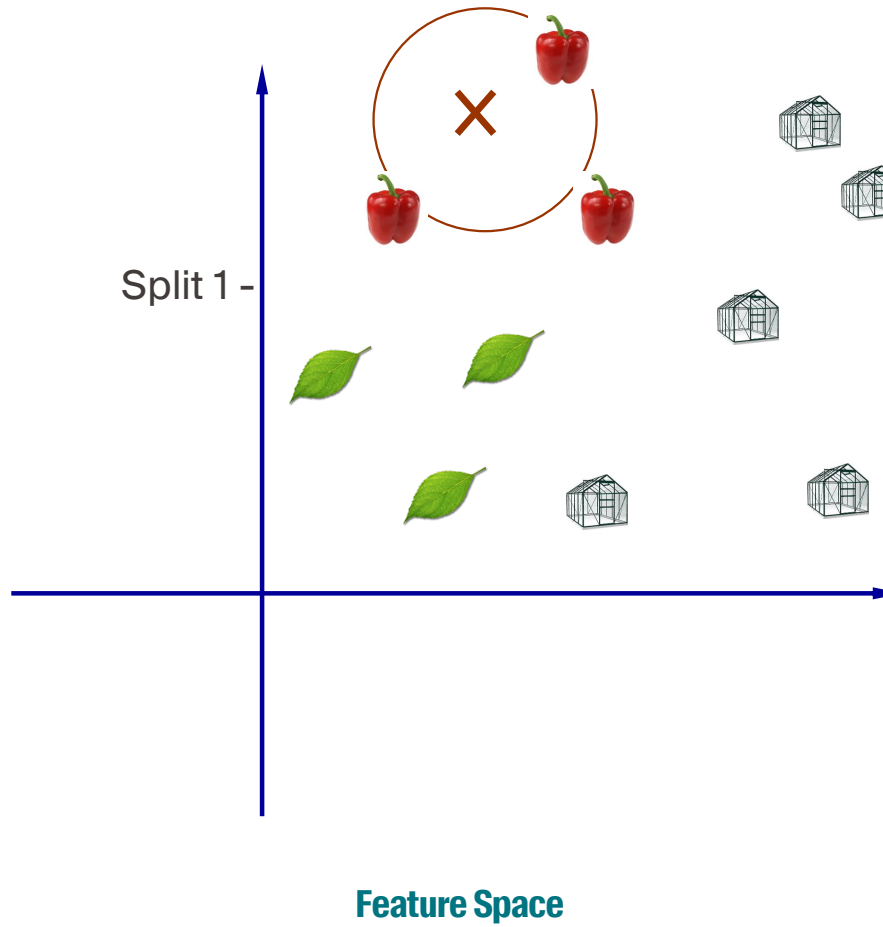
Source: bangkok post

Segmenting the feature space

- Basically a decision tree segments the input space
- It does that using a supervised rule
- Something like: “*if I divide there, would the two resulting segments be clearer about classes*”?

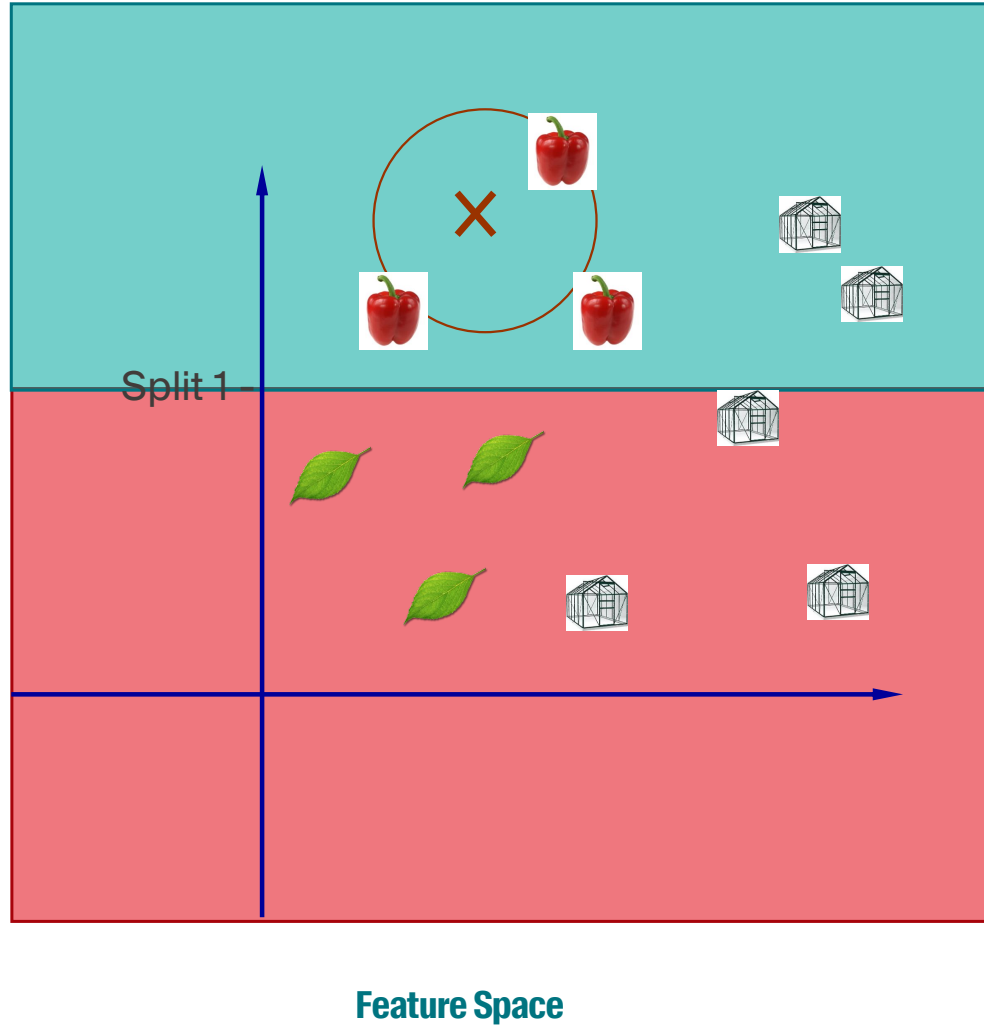
The basic idea:

- You can find the nonlinear solution in 3 splits.



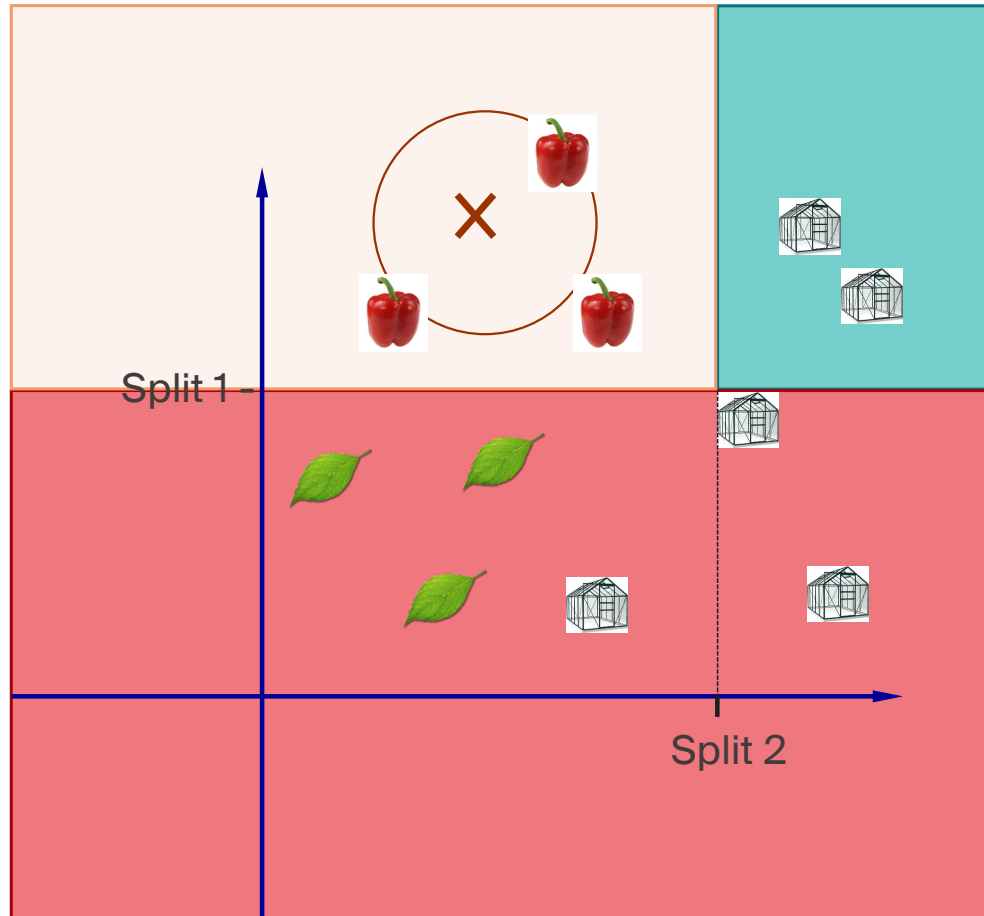
The basic idea:

- You can find the nonlinear solution in 3 splits.



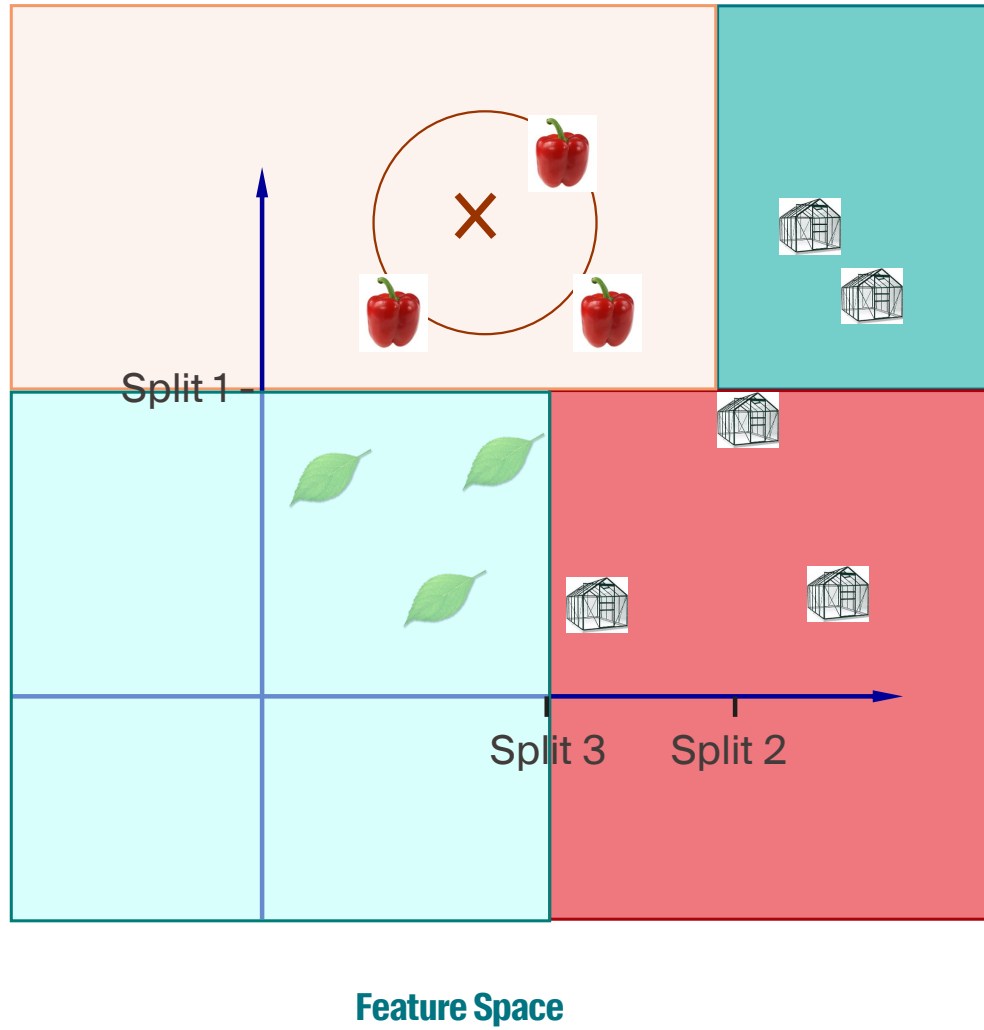
The basic idea:

- You can find the nonlinear solution in 3 splits.
- (with 2 you would get the bell pepper right)



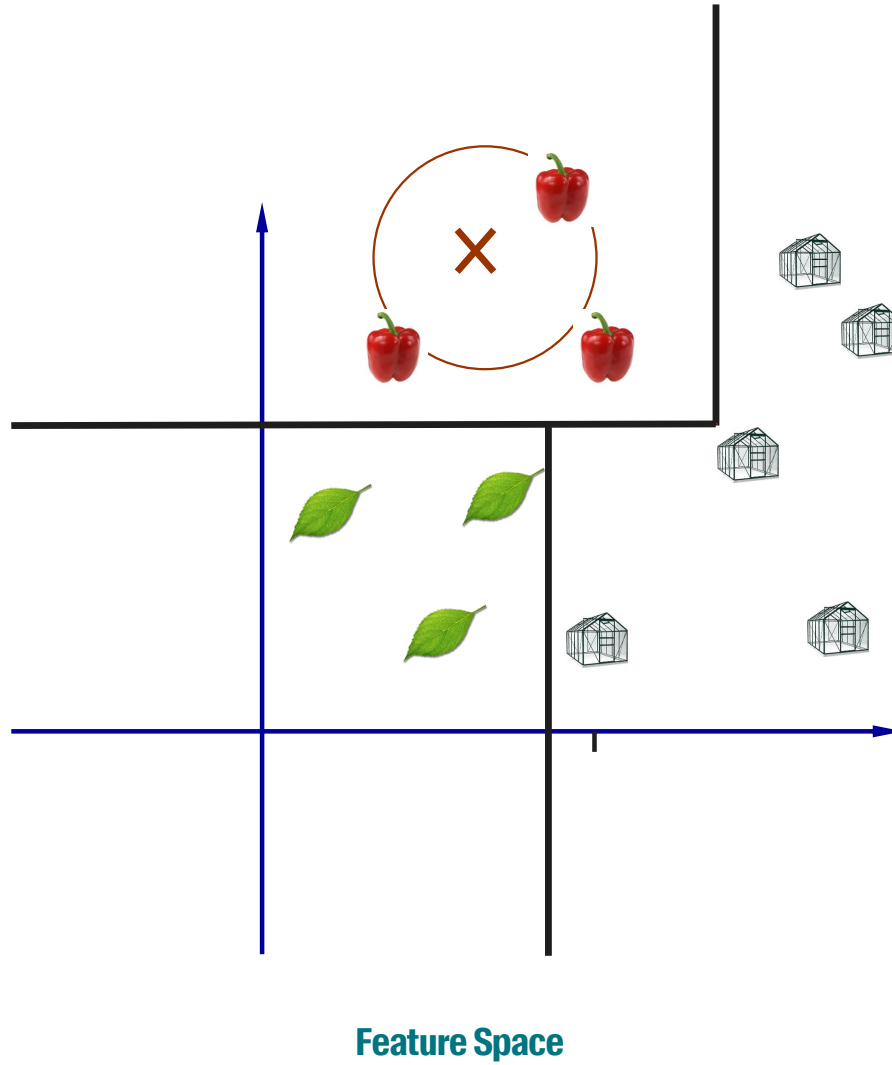
The basic idea:

- You can find the nonlinear solution in 3 splits.

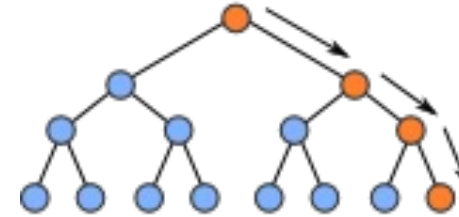


The basic idea:

- You can find the nonlinear solution in 3 splits.



The decision tree in a nutshell



<http://www.r2d3.us/visual-intro-to-machine-learning-part-1/>

1. A decision tree selects one feature at the time and splits the data in 2.
2. The split variable and threshold are those maximizing class homogeneity in the children nodes.
3. Then you work on the remaining data in the children nodes and split them again (and again)
4. When you stop, you classify a leaf by the majority label.

How to split?

- In the website, decisions are taken according to the “best split”.
- A split is a division choice within the decision tree:
 - Given all possible variables x ,
 - Chose one variable $x^{(i)}$,
 - Chose one split value τ of that variable. For each sample i :
 - $x^{(i)} < \tau$ goes to the left child node
 - $x^{(i)} > \tau$ goes to the right child node
- Ex: variable is *price*, split value is *500'000* \$:
 - An apartment costing 300'000\$ goes to the left node
 - An apartment costing 750'000\$ goes to the right node
- How to chose variables and split value??

How to split?

- In the website, decisions are taken according to the “best split”.
- If you looked into the footnote, they also recommended to look for “Gini index” or “Cross entropy”. These are two node purity indices. For a given node m :

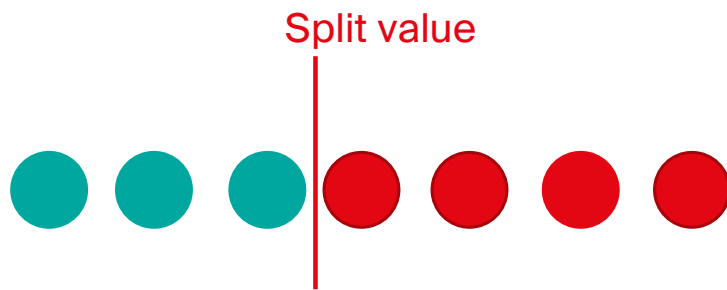
- Gini index G :
$$G = \sum_{k=1}^K \hat{p}_{mk} (1 - \hat{p}_{mk})$$

- Cross Entropy D :
$$D = \sum_{k=1}^K \hat{p}_{mk} \log \hat{p}_{mk}$$

- \hat{p}_{mk} is the proportion of samples in node m from class k

How to split (example)

- Let's consider two variables leading to very different splits



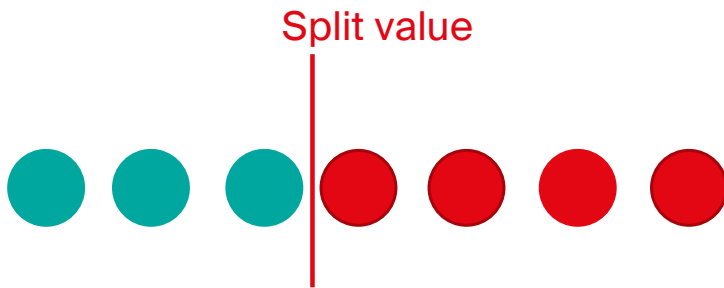
$$\text{Gini (left)} = (0 * (1-0)) + (1 * (1-1)) = 0 + 0 = 0$$

$$\text{Gini (right)} = (1 * (1-1)) + (0 * (1-0)) = 0 + 0 = 0$$

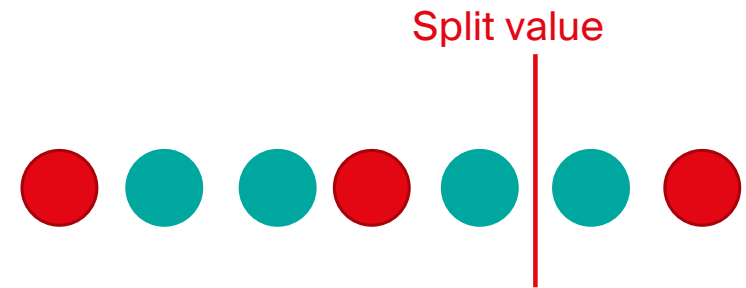
$$\text{Gini(split)} = 0$$

How to split (example)

- Let's consider two variables leading to very different splits



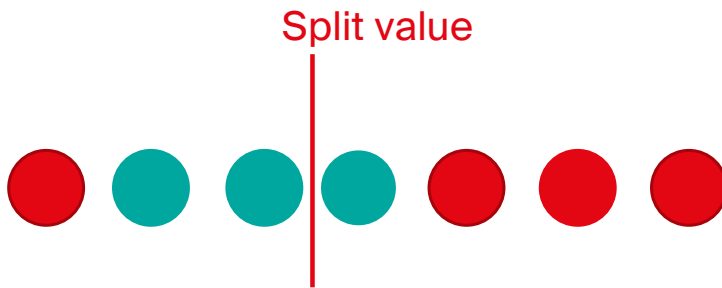
$$\begin{aligned} \text{Gini (left)} &= (0 * (1-0)) + (1 * (1-1)) = 0 + 0 = 0 \\ \text{Gini (right)} &= (1 * (1-1)) + (0 * (1-0)) = 0 + 0 = 0 \\ \text{Gini(split)} &= 0 \end{aligned}$$



$$\begin{aligned} \text{Gini(left)} &= (2/5 * 3/5) + (3/5 + 2/5) = 0.24 + 0.24 = 0.48 \\ \text{Gini(right)} &= (.5 * .5) + (.5 * .5) = 0.25 + 0.25 = 0.5 \\ \text{Gini(split)} &= 0.98 \end{aligned}$$

How to split (example)

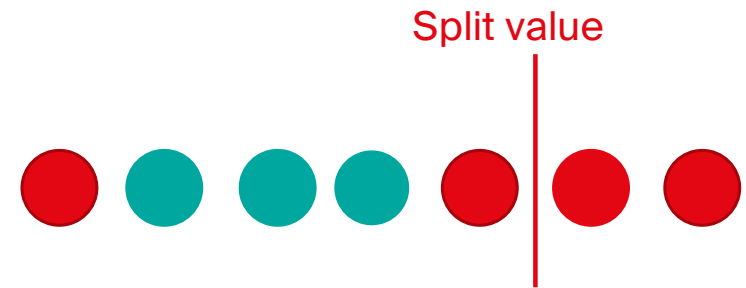
- Let's consider one fix variable and two different split values



$$\text{Gini(left)} = (1/3 * (2/3)) + (2/3 * (1/3)) = 0.22 + 0.22 = 0.44$$

$$\text{Gini(right)} = (3/4 * (1/4)) + (1/4 * (3/4)) = 0.38$$

$$\text{Gini(split)} = 0.82$$



$$\text{Gini(left)} = (2/5 * (3/5)) + (3/5 * (2/5)) = 0.24 + 0.24 = 0.48$$

$$\text{Gini(right)} = (1 * (1-1)) + (0 * (1-0)) = 0$$

$$\text{Gini(split)} = 0.48$$

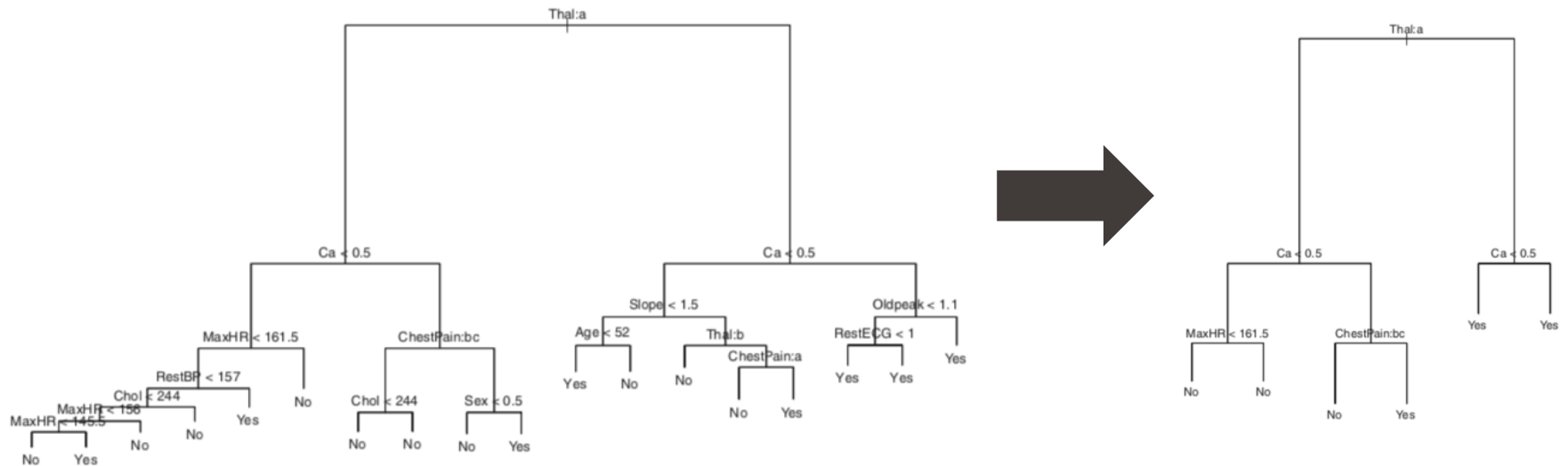
How do we construct the partitioning?

- Top-down, greedy approach
 - Top-down: start at top of tree, successively split predictor space
 - Greedy: at each step, best split is made at that step only. Stop when a criterion is met.
- This will lead to overfit and overcomplex trees.

How do we construct the partitioning?

■ Solution

- A. Stop early (e.g. set a minimum depth)
- B. Prune the tree



How do we construct the partitioning?

- Solution

- A. Stop early (e.g. set a minimum depth)
- B. Prune the tree using cost:

$$\sum_{m=1}^{|T|} Gini(m) + \alpha |T|$$

- First grow a very large (deep) tree T_0 . You now have the solution for $\alpha = 0$.
- For each α , there is a subtree $T \subset T_0$
- Increase α and re-run. The **regularizer** is the price to pay for increasing the number of terminal nodes
- Use a k -fold cross-validation approach to evaluate the error function above. Use the average error as final measure.
- Once minimized, grow an optimal tree using all data.

From the tree to the forest

- A single decision tree can overfit (see above)
- A single decision tree can suffer from **high variance**: if we use different training sets, decisions can be quite different
- The concept of **bagging** is meant to reduce such variance by building a **committee of models**.
- Random forests (RF) use it.



Source: [Udacity.com/course/ud501](https://udacity.com/course/ud501)

From the tree to the forest

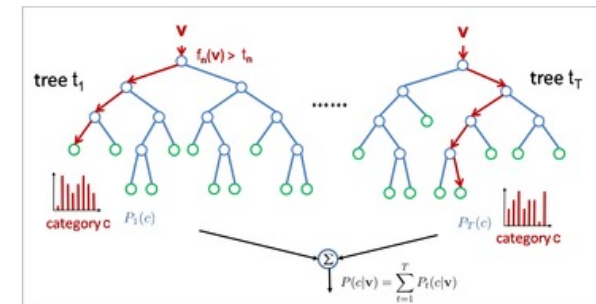


- Let's say a single decision tree has an output Z with variance σ^2
- If we repeat the modeling with n independent trials, we get n models with variances $Z_1, Z_2, Z_3, \dots, Z_n$, each one with variance σ^2 .
- According to the **central limit theorem**, the variance of their average is σ^2/n .
- In other words: averaging independent models reduces variance.

Forest through randomization

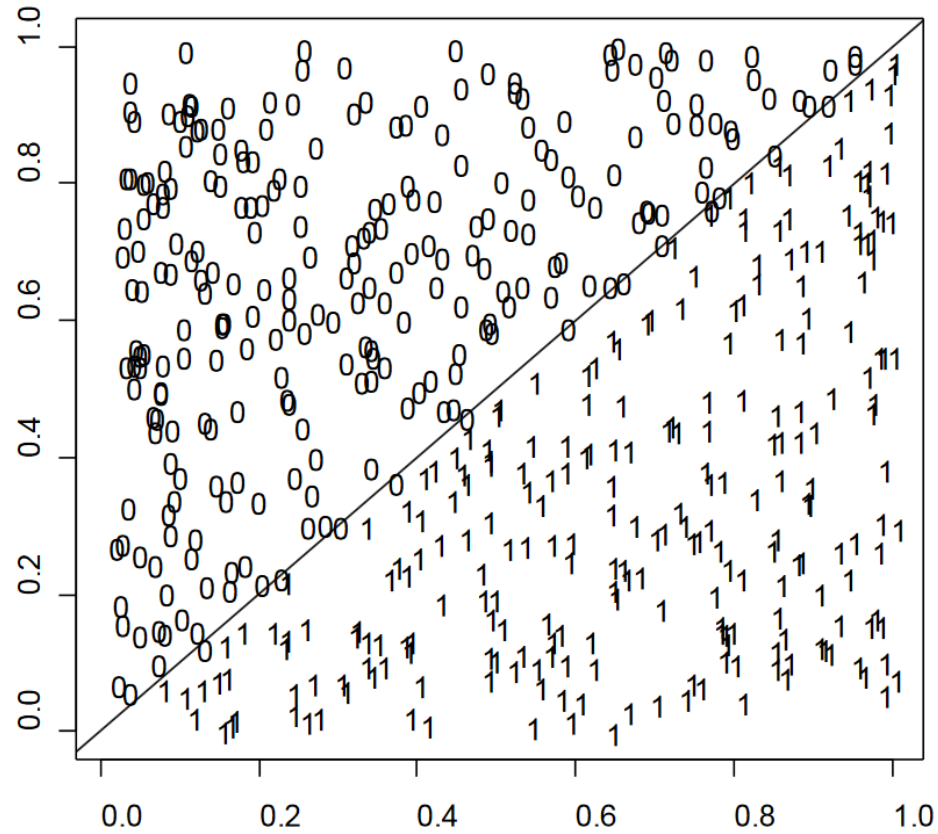
- In practice we train B different methods (f^*) with subsets of the data and features

- Then take a majority vote (classification)



- Even more in practice, we can't have truly independent subsets, so we resample parts of the data and use them in each model training.

Decision Trees & Random Forest

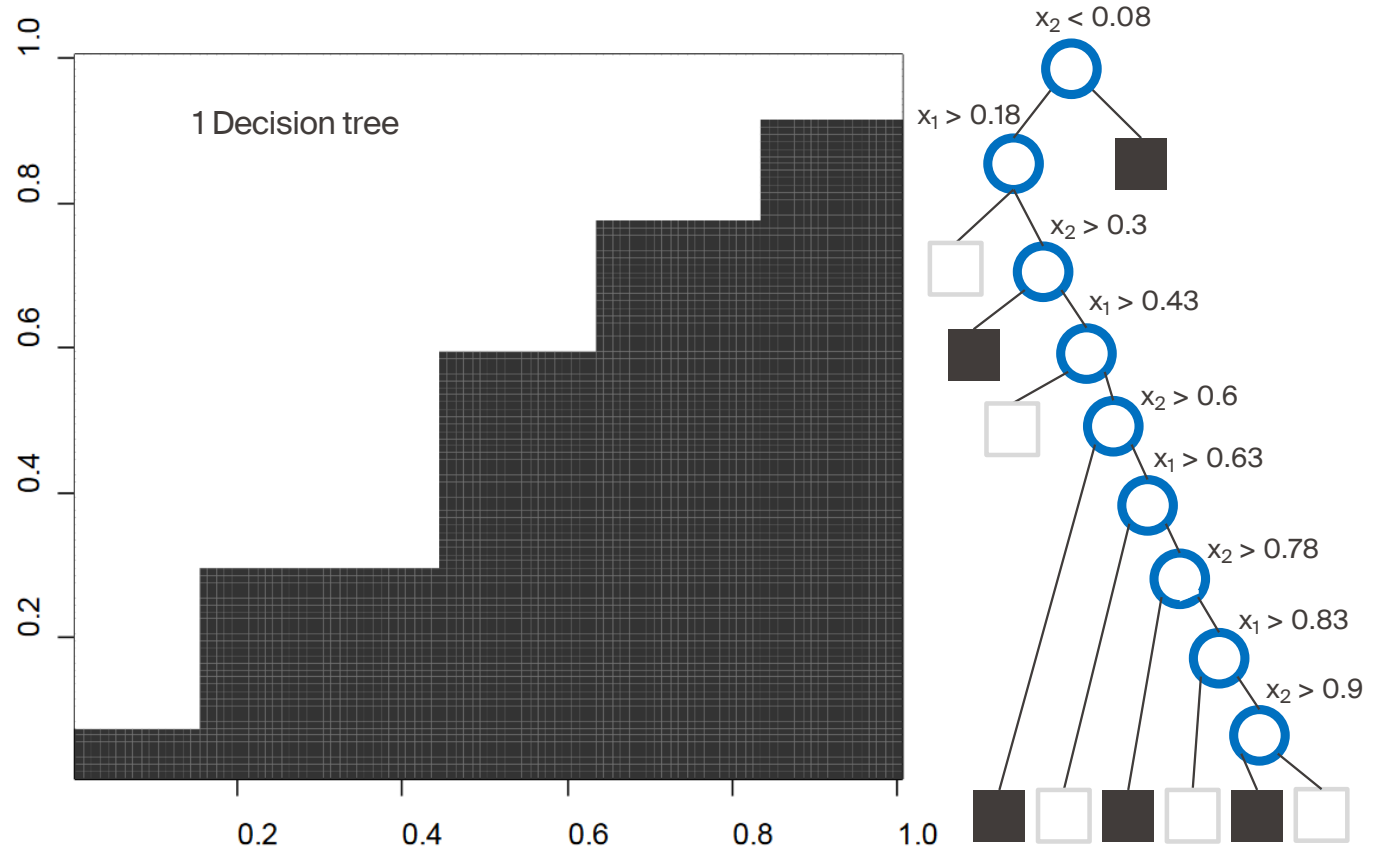


From <http://www.math.usu.edu/adele/RandomForests/ENAR.pdf>

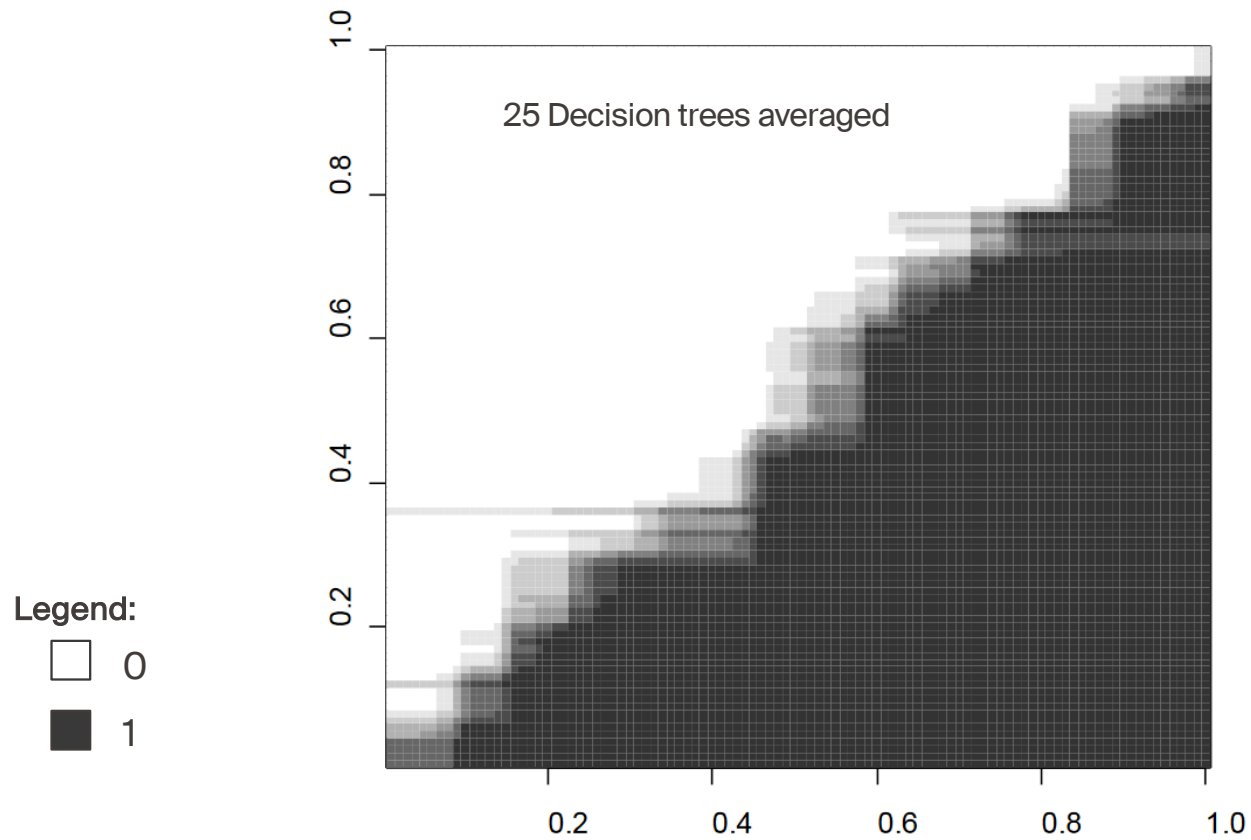
Decision Trees & Random Forest

IPEO course - 3 image classification (RF)
30 September 2025

Legend:
 0
 1

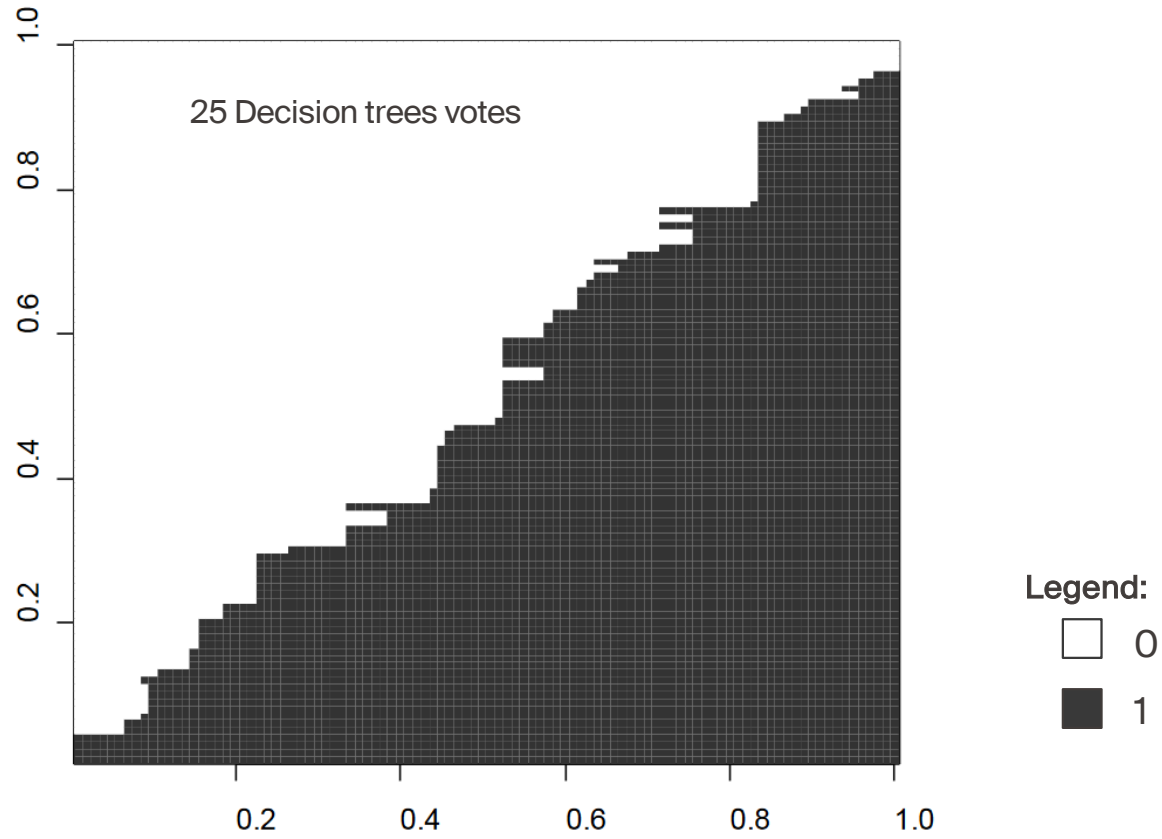


Decision Trees & Random Forest



From <http://www.math.usu.edu/adele/RandomForests/ENAR.pdf>

Decision Trees & Random Forest



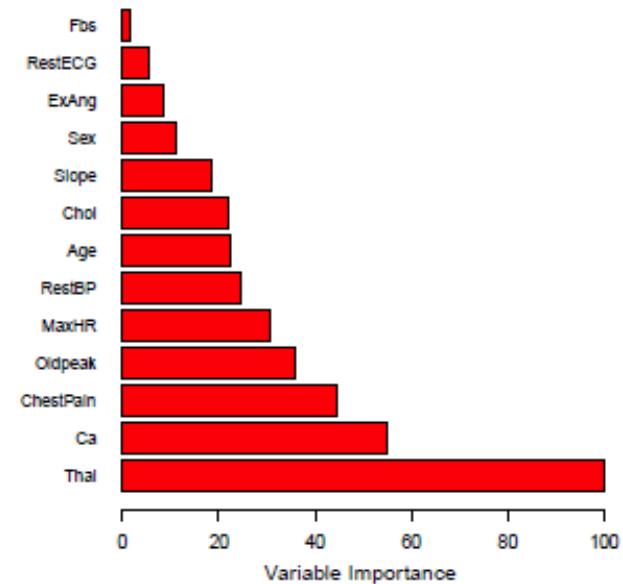
From <http://www.math.usu.edu/adele/RandomForests/ENAR.pdf>

And we get for free

Error estimates.

- **Out-of-bag (OOB):** observations (samples) not used in a given tree of the forest
- **OOB prediction:** majority vote of the predictions over these unused samples
- **OOB classification error** is an estimate of the test error for a bagged model

Variable importance.



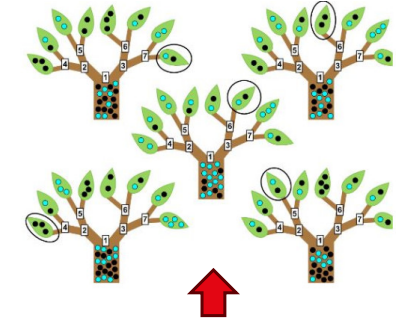
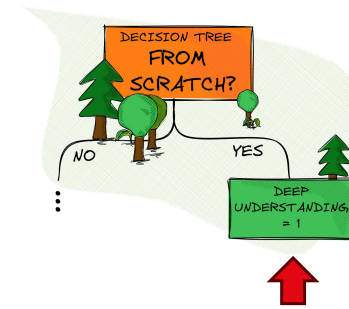
Random forests.

- Basically you repeat the decision tree generation over and over
- Each time you take a different subset of your labeled data
- Each time you generate different variables and split values
- So each tree will be different, but consider the same problem

A different look on the same problem

- Some will be good in some classes
- Others will be good at searching some parts of the feature space (those they have seen)
- But together they will be a strong robust, nonlinear model.

Random forest is cool because



- It is simple
- You basically have 2 parameters: **depth of the trees** and **number of trees**
- It can handle high dimensional data (= many variables)
- It gives you a non linear solution
- BUT still you need to pre-compute your features before training the model.