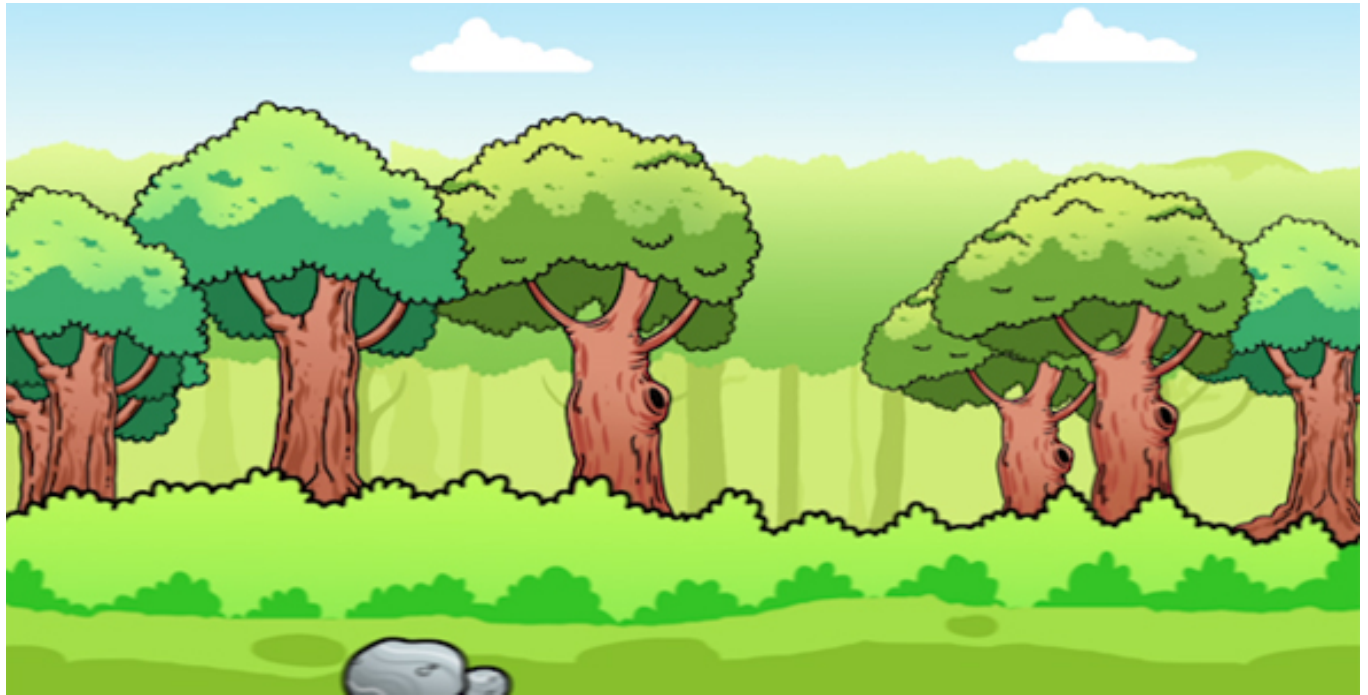
A Deep Dive into Random Forests

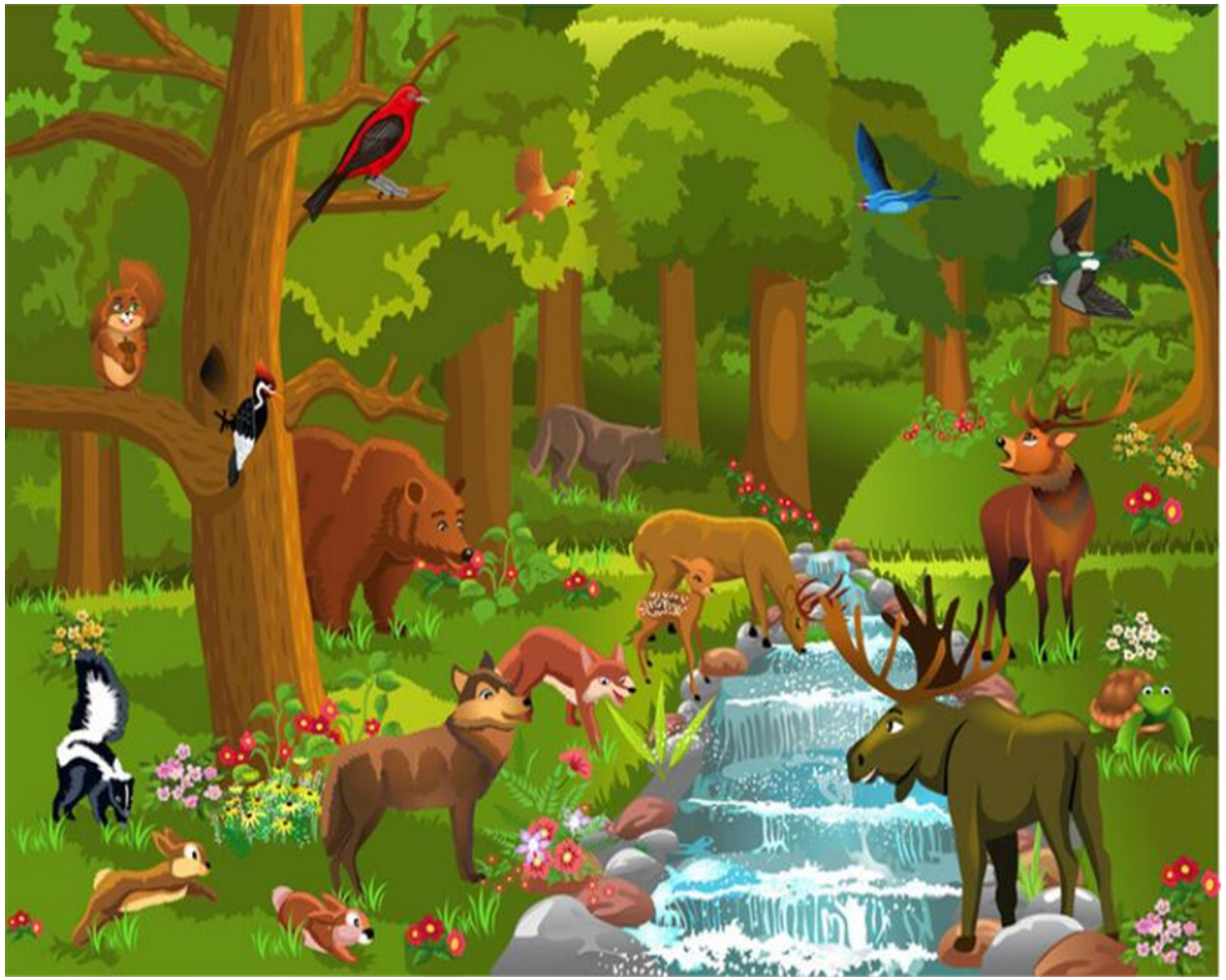
By: Ari Silburt

https://github.com/silburt/RF_DeepDive



Random Forests May Seem Scary…

But They’re Actually Not Too Bad!



Plan

- Decision Tree
- Random Forests (+ Bagging)
- Tree Optimization and Feature Importance (Gini Criterion)
- Model Regularization
- Closing Notes

Decision Tree

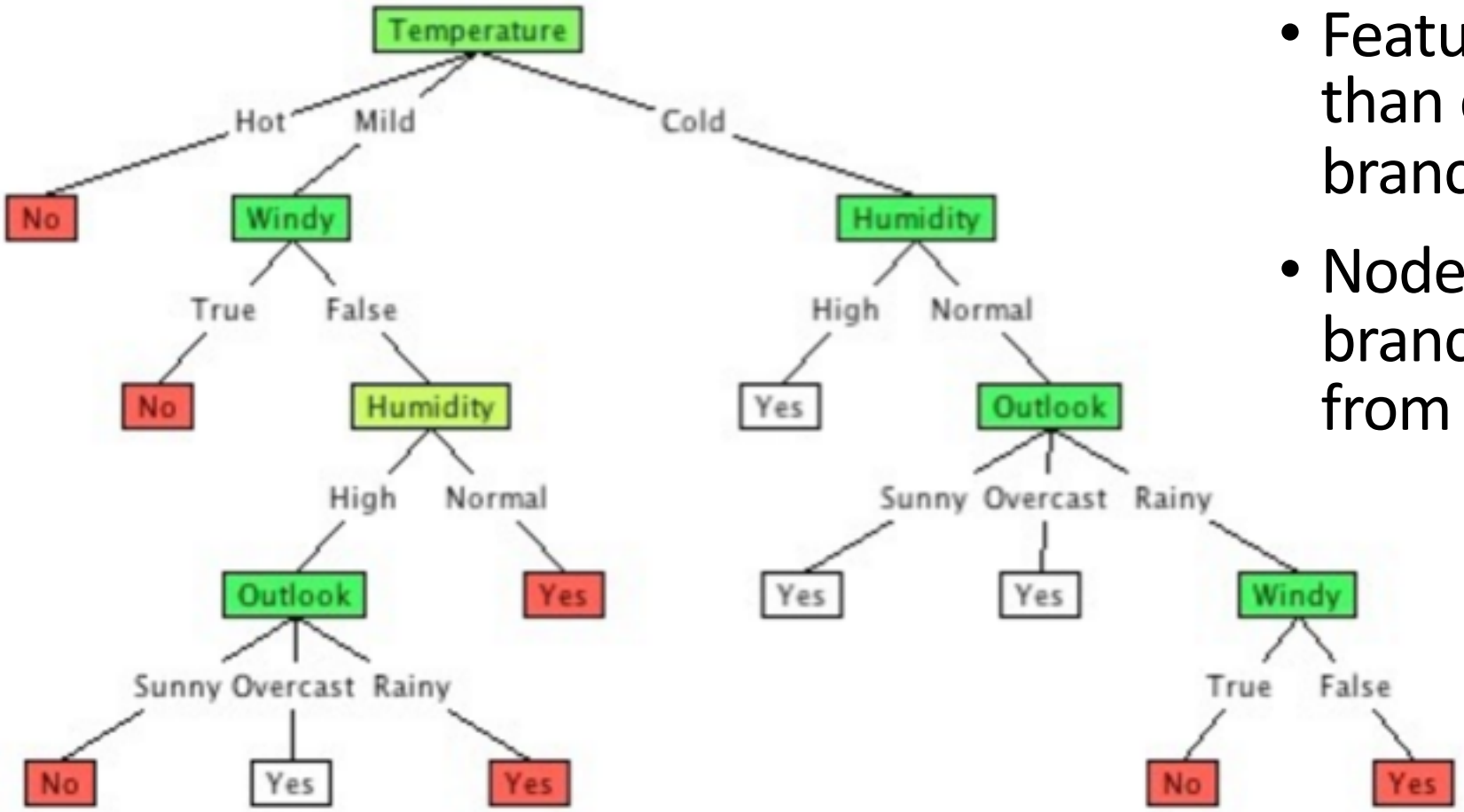
Decision Tree

Example: Should We Play Tennis?

Play Tennis	Outlook	Temperature	Humidity	Windy
No	Sunny	Hot	High	No
No	Sunny	Hot	High	Yes
Yes	Overcast	Hot	High	No
Yes	Rainy	Mild	High	No
Yes	Rainy	Cold	Normal	No

- If temperature is not hot
  - Play
- If outlook is overcast
  - Play tennis
- Otherwise
  - Don't play tennis

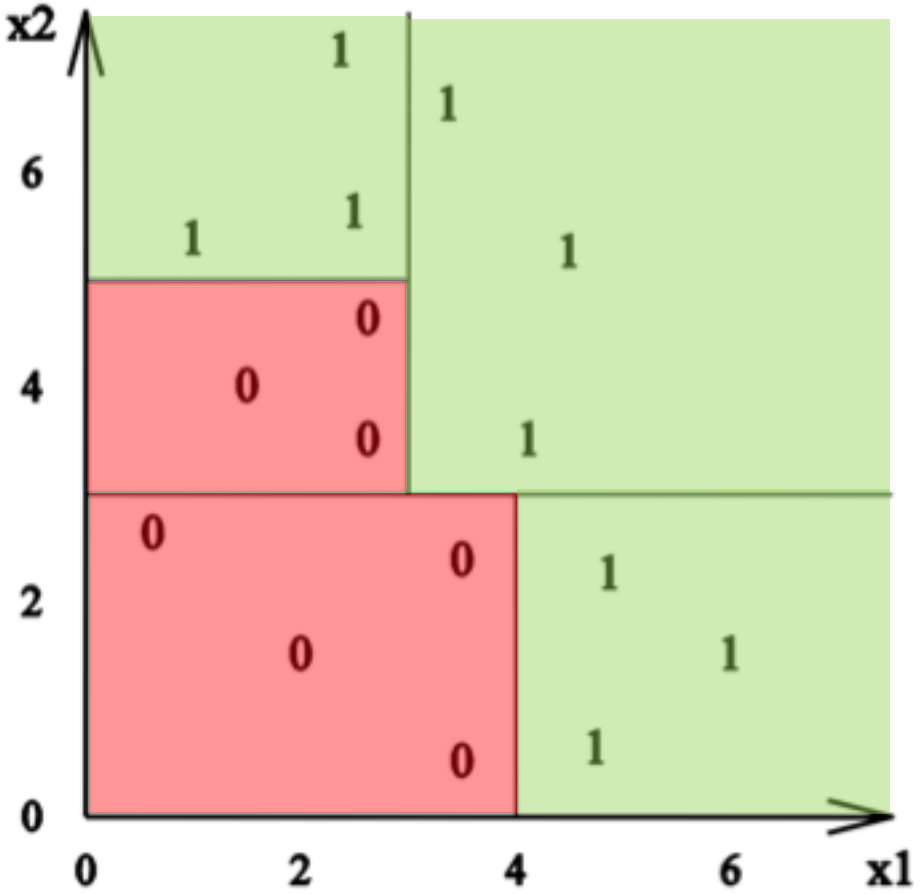
Decision Tree

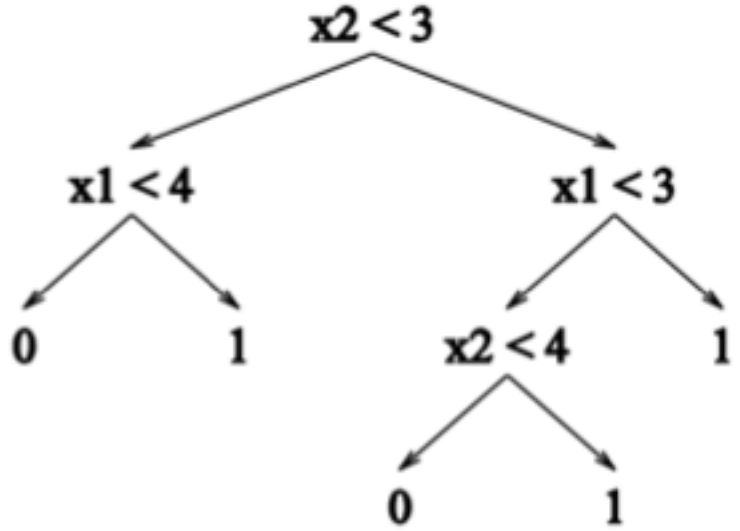


Properties

- Feature can show up more than once in different branches (e.g. windy).
- Node can have both a branch and leaf stemming from it.

Decision Tree – Pros and Cons

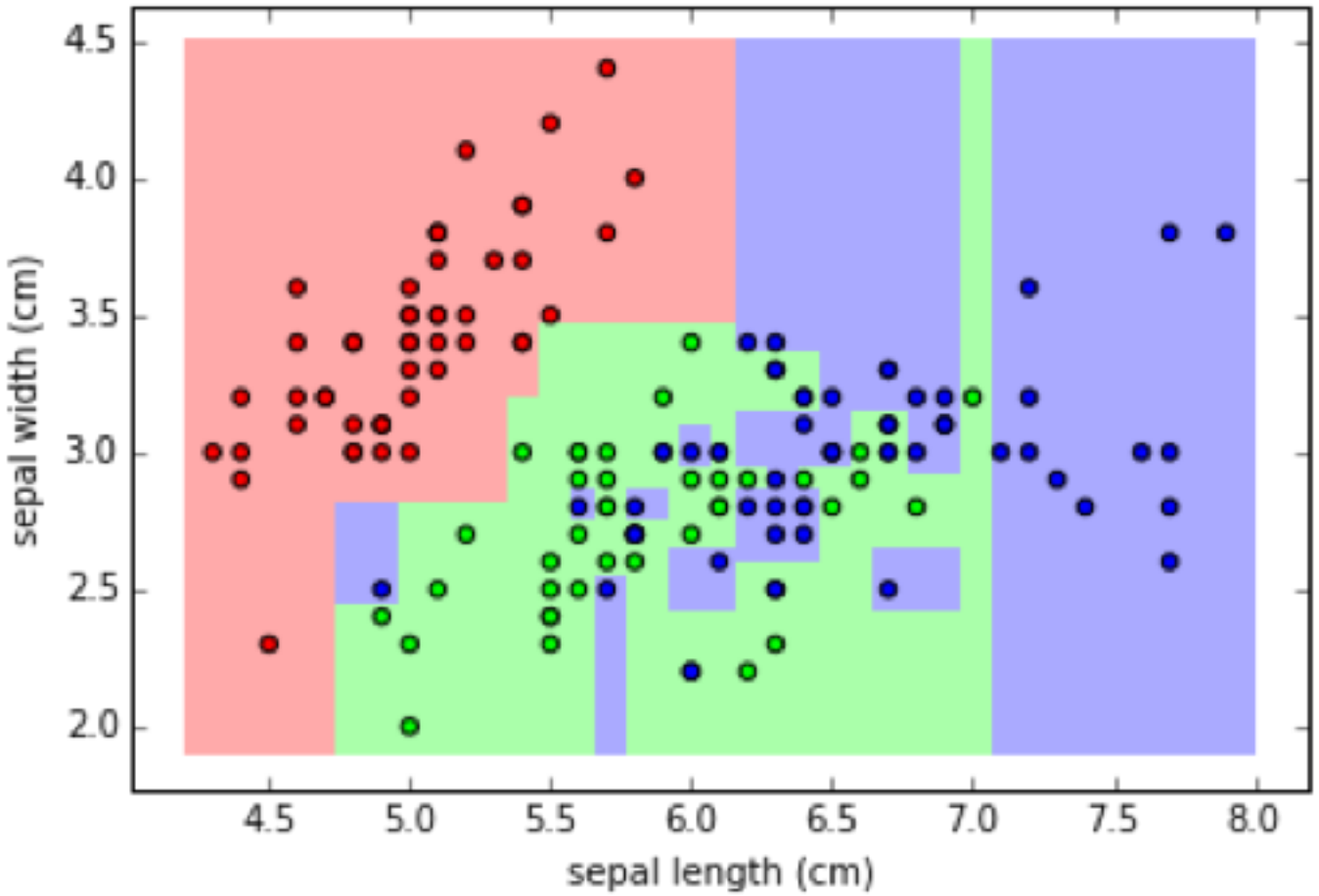




Pros:

- Non-linear decision boundaries
- Easy to interpret
- Numerical & Categorical Data

Decision Tree – Pros and Cons



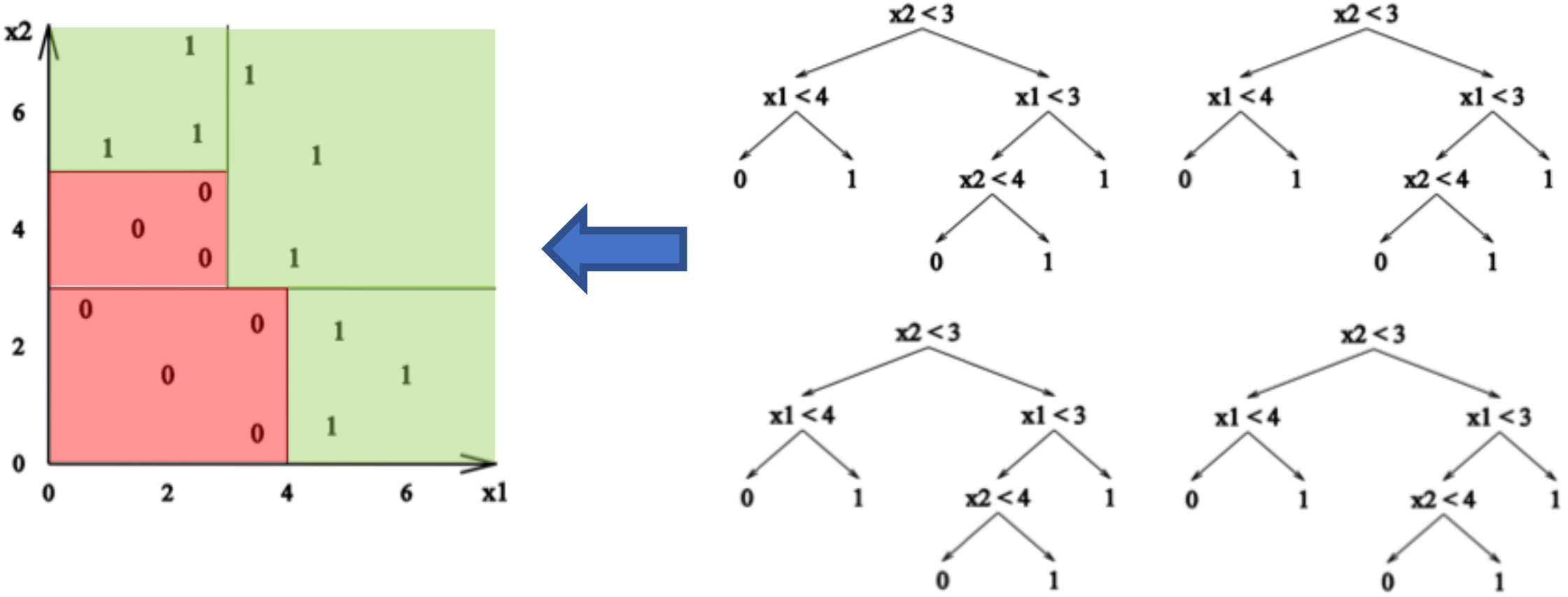
Cons:

- Easy to Overfit
- High Variance (i.e. unstable).

Random Forests

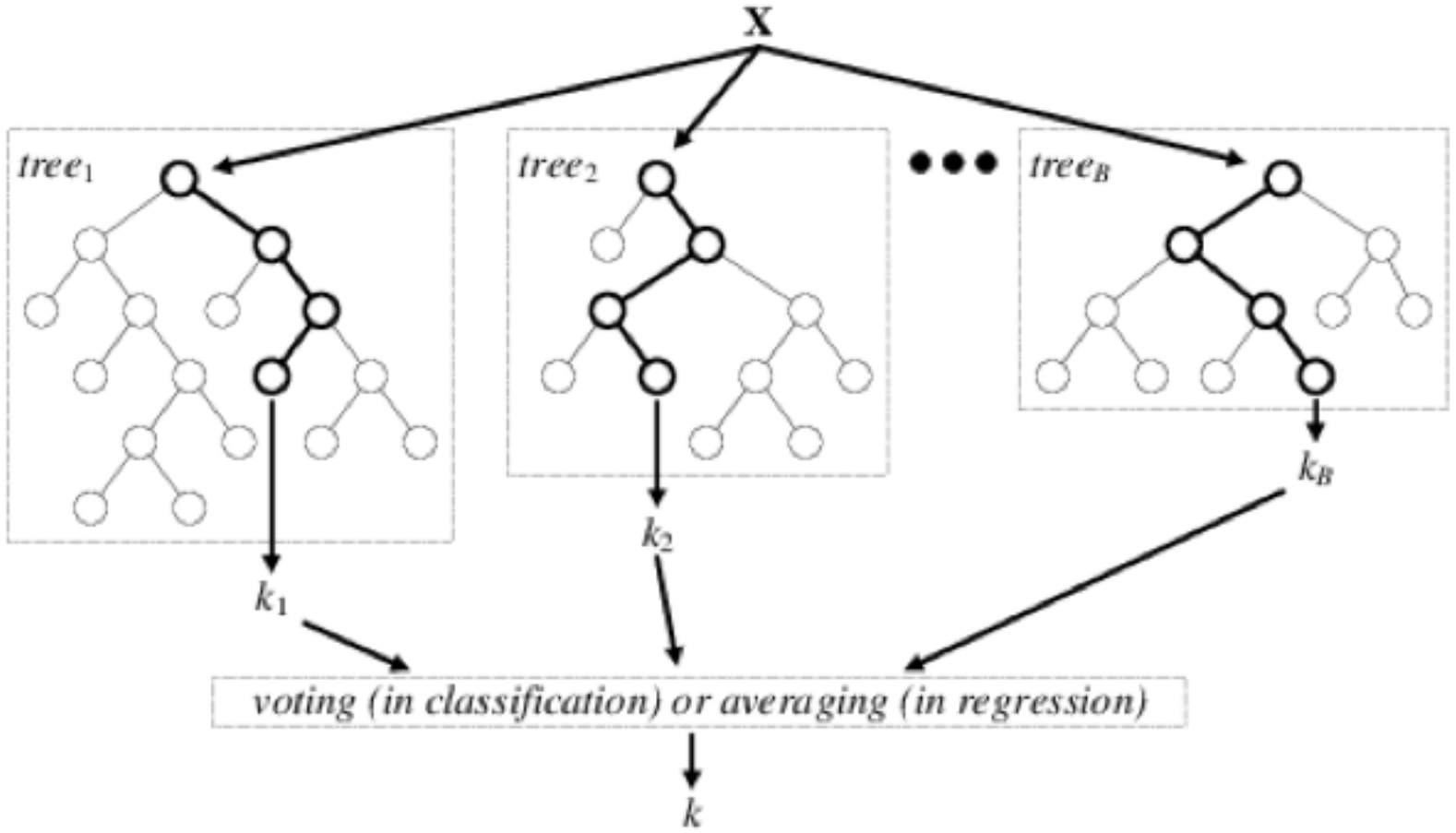
Random Forests – Many Decision Trees

- We can guess that a Random Forest = many decision trees. But how?
- Many copies of the exact same tree is useless…



Random Forests – Many Decision Trees

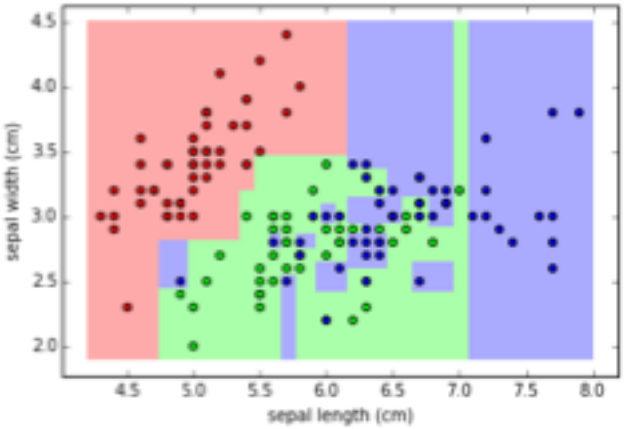
- OK, so we want some tree variation, but how…



Random Forests – Many Decision Trees

- We want tree variation, but how…
- **Vary trees such that overall variance is reduced:**

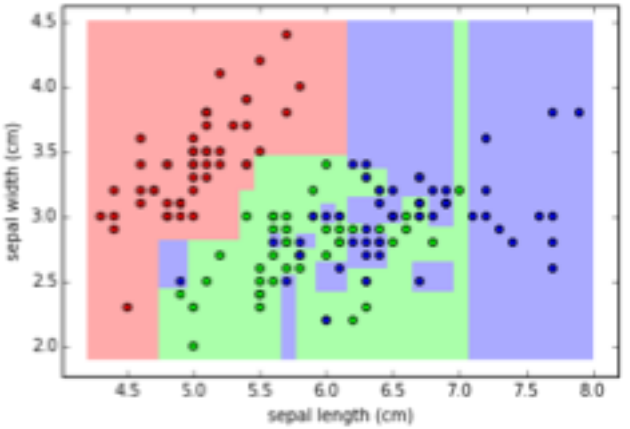
Remember: Decision Tree



Random Forests – Many Decision Trees

- We want tree variation, but how…
- **Vary trees such that overall variance is reduced:**
- STATS101:

Remember: Decision Tree

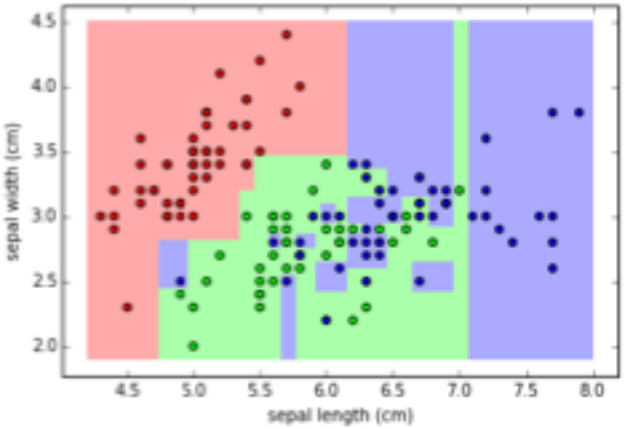


Given a set of **independent, uncorrelated observations** $Z_1, Z_2, \ldots, Z_n$ each with variance σ^2, the variance of Z is $\frac{\sigma^2}{n}$.

Random Forests – Many Decision Trees

- We want tree variation, but how…
- **Vary trees such that overall variance is reduced:**
- STATS101:

Remember: Decision Tree



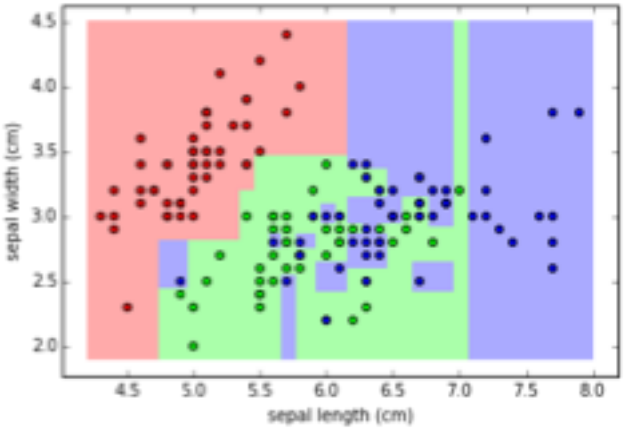
Given a set of **independent, uncorrelated observations** $Z_1, Z_2, \ldots, Z_n$ each with variance σ^2, the variance of Z is $\frac{\sigma^2}{n}$.

This is why a forest of identical trees is useless.

Random Forests – Many Decision Trees

- We want tree variation, but how…
- **Vary trees such that overall variance is reduced:**
- STATS101:

Remember: Decision Tree



Given a set of **independent, uncorrelated observations** $Z_1, Z_2, \ldots, Z_n$ each with variance σ^2, the variance of Z is $\frac{\sigma^2}{n}$.

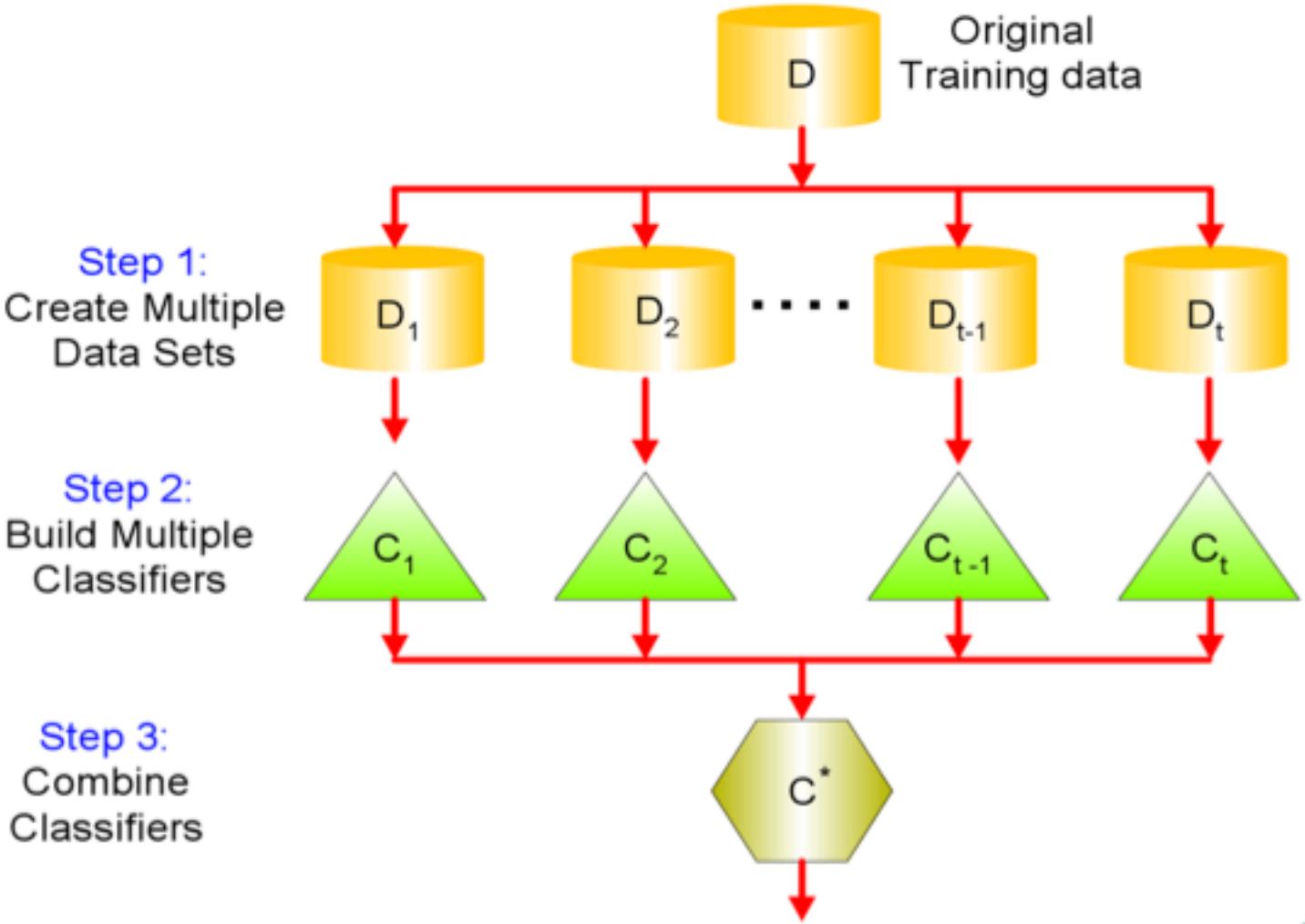
This is why a forest of identical trees is useless.

This is why ensembling many models together always improves results.

Random Forests – Randomize Data

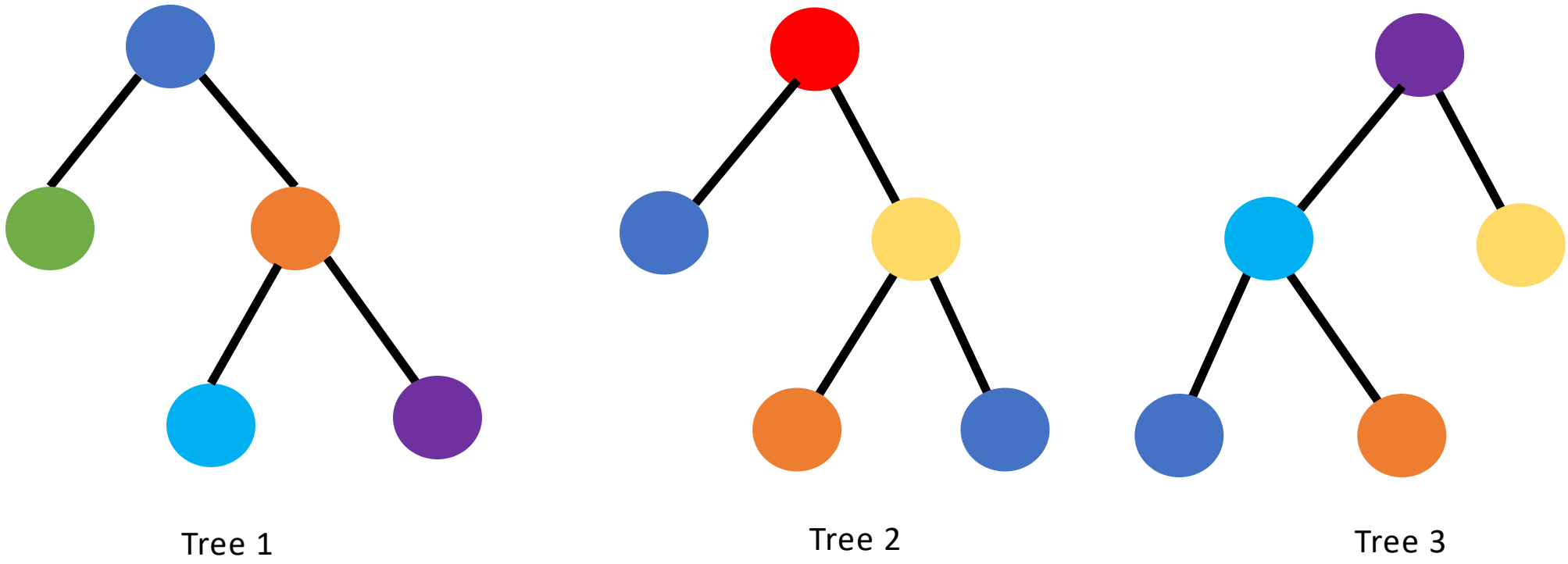
- **Bootstrap sampling**: Given set D containing N training examples, create D' by drawing N examples at random **with replacement** from D
- Bagging
  - Create k bootstrap samples $D_1, \ldots, D_k$
  - Train distinct classifier on each D_i
  - Classify new instance by majority vote / average

Random Forests – Randomize Data



[image from the slides of David Sontag]

Random Forests – Randomize Features



Tree 1 Tree 2 Tree 3

Feature 1 Feature 3 Feature 5 Feature 7

Feature 2 Feature 4 Feature 6

Random Forests – Intuition Check

- What happens if you assign more/less data per tree?

- What happens if you select more/less of the total features per tree:

Random Forests – Intuition Check

- What happens if you assign more/less data per tree?

Less: Trees more uncorrelated, but at some point too little data hurts training.

More: Trees become more correlated, but training of each tree improved.

- What happens if you select more/less of the total features per tree:

Less: Trees more uncorrelated, but at some point many trees become “dead”, i.e. fitting entire trees on unimportant features.

More: Trees become more correlated, but training of each tree improved.

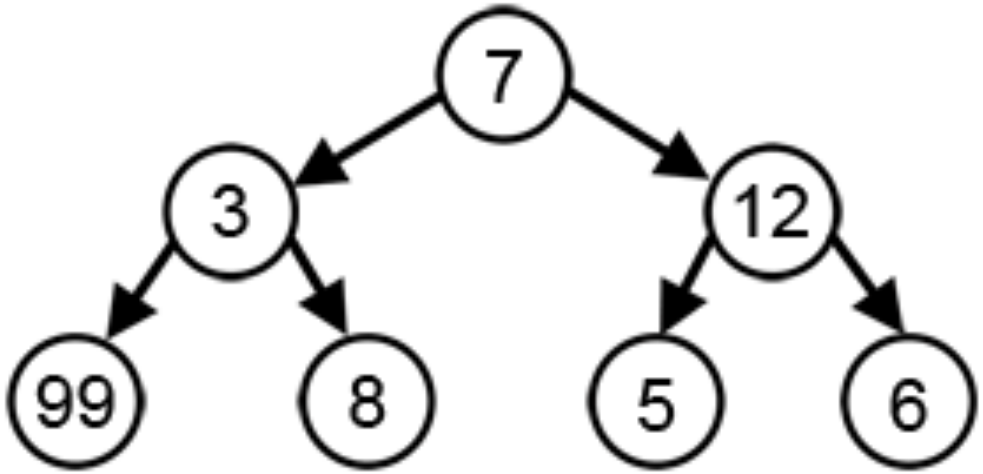
Tree Optimization and Feature Importances

Tree Optimization – Greedy Criterion

- Trees grown according to *what the local best option is*.
- Criterion: Gini, Information Gain.

Short Aside - Greedy Algorithm Example

Example: Find largest path.



Tree Optimization – Greedy Criterion

- Note: The criterion governing tree growth is *different* than your global cost function (e.g. precision-recall, accuracy, etc.), which determines how well your entire model is doing.

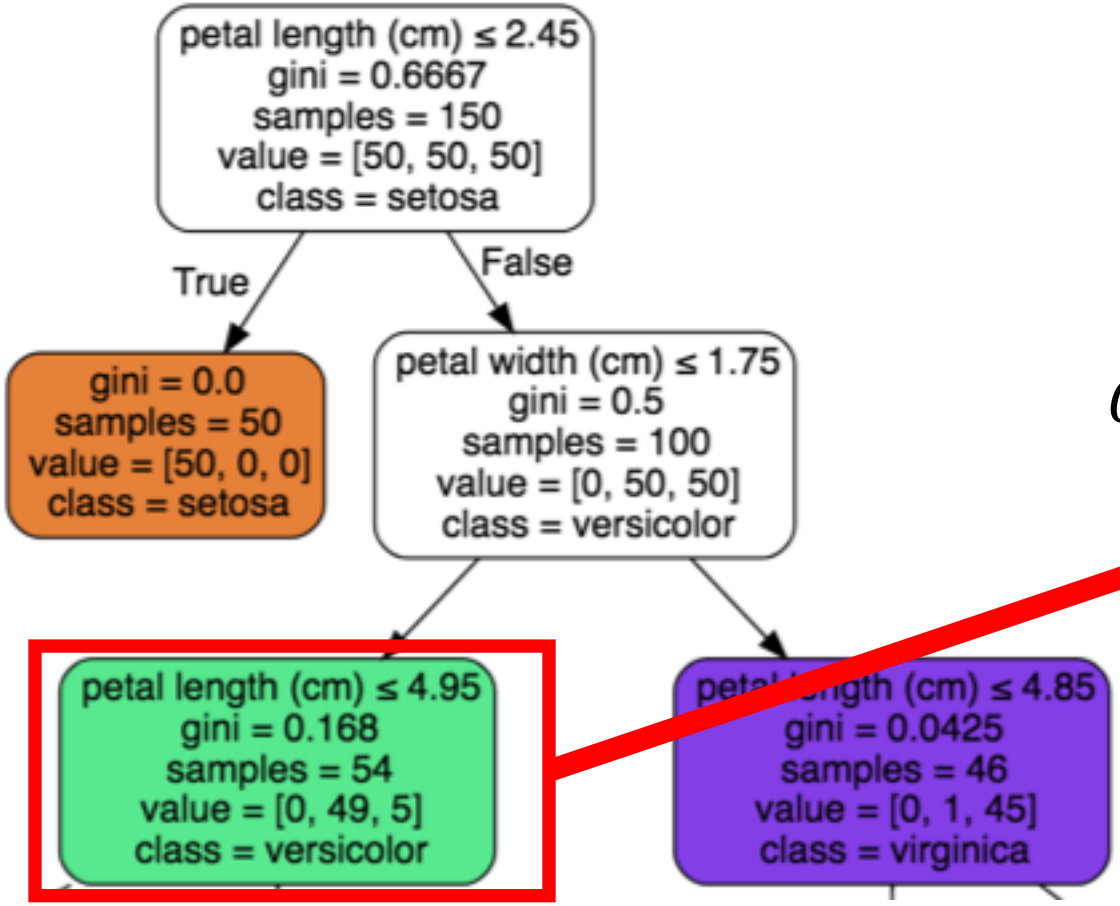
Tree Optimization – Gini Impurity

"Gini impurity is a measure of how often a randomly chosen element from the set would be incorrectly labeled if it was randomly labeled according to the distribution of labels in the subset."

$$G_n = 1 - \sum_{i=0}^{C_n} \left(\frac{N_{n,i}}{N_n}\right)^2$$

G_n = Gini Impurity of node n
C_n = total number of classes in node n
$N_{n,i}$ = Number of samples in node n from class i
N_n = Number of samples in node n

Tree Optimization – Gini Impurity Example



$$G_n = 1 - \sum_{i=0}^{C_n} \left(\frac{N_{n,i}}{N_n}\right)^2$$

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

$$G_n = 0.168$$

Splits decided such that the gini impurity is minimized

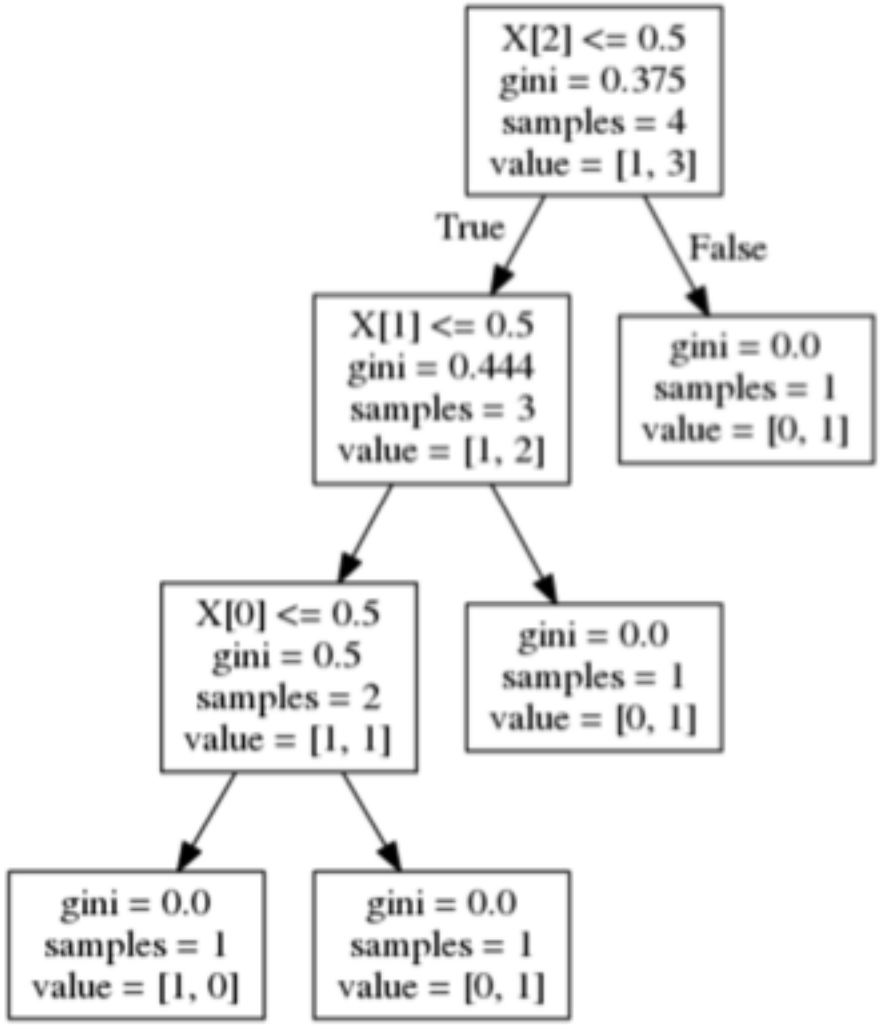
Feature Importance – Gini Impurity

"The importance of a feature is computed as the (normalized) total reduction of the criterion brought by that feature. It is also known as the Gini importance."

$$\tilde{I}_f = \sum_{j=0}^{O_f} \frac{N_{n_j}}{T} \left(G_{n_j} - \frac{N_{LC_j}}{N_{n_j}} G_{LC_j} - \frac{N_{RC_j}}{N_{n_j}} G_{RC_j} \right), \quad I_f = \tilde{I}_f / \sum \tilde{I}_f$$

$I_f, \tilde{I}_f =$ Normalized/Unnormalized Importance of feature f
$O_f =$ Occurrences of feature f in tree
$N_{n_j} =$ number of samples in node n (for feature j)
$T =$ total number of samples
$G_{n_j} =$ Gini Impurity of node n (for feature j)
$N_{LC_j}, N_{RC_j} =$ number of samples in left/right child node (for feature j)
$G_{LC_j}, G_{RC_j} =$ Gini Impurity of left/right child node (for feature j)

Feature Importance – Gini Impurity Example



$$I_f = \sum_{j=0}^{o_f} \frac{N_{n_j}}{T} \left(G_{n_j} - \frac{N_{LC_j}}{N_{n_j}} G_{LC_j} - \frac{N_{RC_j}}{N_{n_j}} G_{RC_j} \right)$$

$$I_{X[0]} = \frac{2}{4}(0.5 - 0 - 0) = 0.25$$

$$I_{X[1]} = \frac{3}{4}\left(0.444 - \left(\frac{2}{3}\right)0.5 - 0\right) = 0.083$$

$$I_{X[2]} = \frac{4}{4}\left(0.375 - \left(\frac{3}{4}\right)0.444 - 0\right) = 0.042$$

$$\boldsymbol{I_{X[0]}} = \frac{0.25}{0.25 + 0.083 + 0.042} = \mathbf{0.67}$$

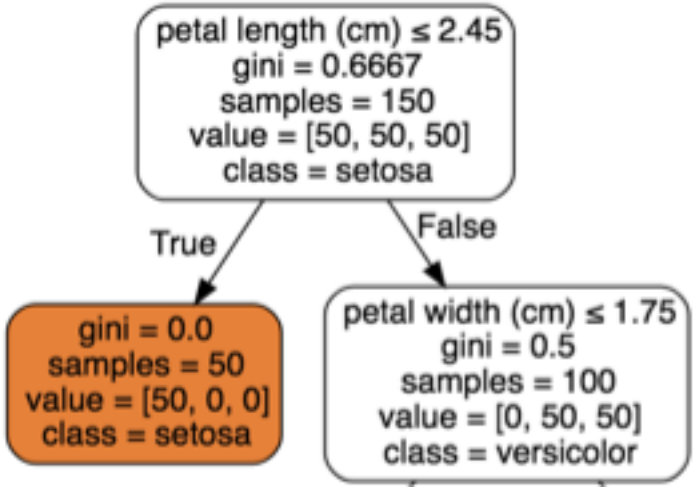
$$\boldsymbol{I_{X[1]}} = \frac{0.083}{0.25 + 0.083 + 0.042} = \mathbf{0.22}$$

$$\boldsymbol{I_{X[2]}} = \frac{0.042}{0.25 + 0.083 + 0.042} = \mathbf{0.11}$$

Feature Importance – Intuition Check

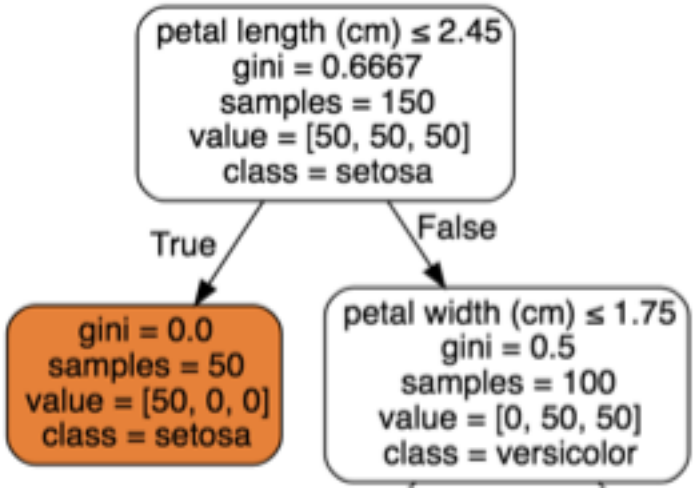
$$\tilde{I}_f = \sum_{j=0}^{O_f} \frac{N_{n_j}}{T} \left(G_{n_j} - \frac{N_{LC_j}}{N_{n_j}} G_{LC_j} - \frac{N_{RC_j}}{N_{n_j}} G_{RC_j} \right)$$

What factors maximize feature importance?



Feature Importance – Intuition Check

$$\tilde{I}_f = \sum_{j=0}^{o_f} \frac{N_{n_j}}{T} \left(G_{n_j} - \frac{N_{LC_j}}{N_{n_j}} G_{LC_j} - \frac{N_{RC_j}}{N_{n_j}} G_{RC_j} \right)$$



What factors maximize feature importance?

Features used at the top of the tree contribute to the final prediction decision of a larger fraction of the input samples. The **expected fraction of the samples** they contribute to can thus be used as an estimate of the **relative importance of the features**.

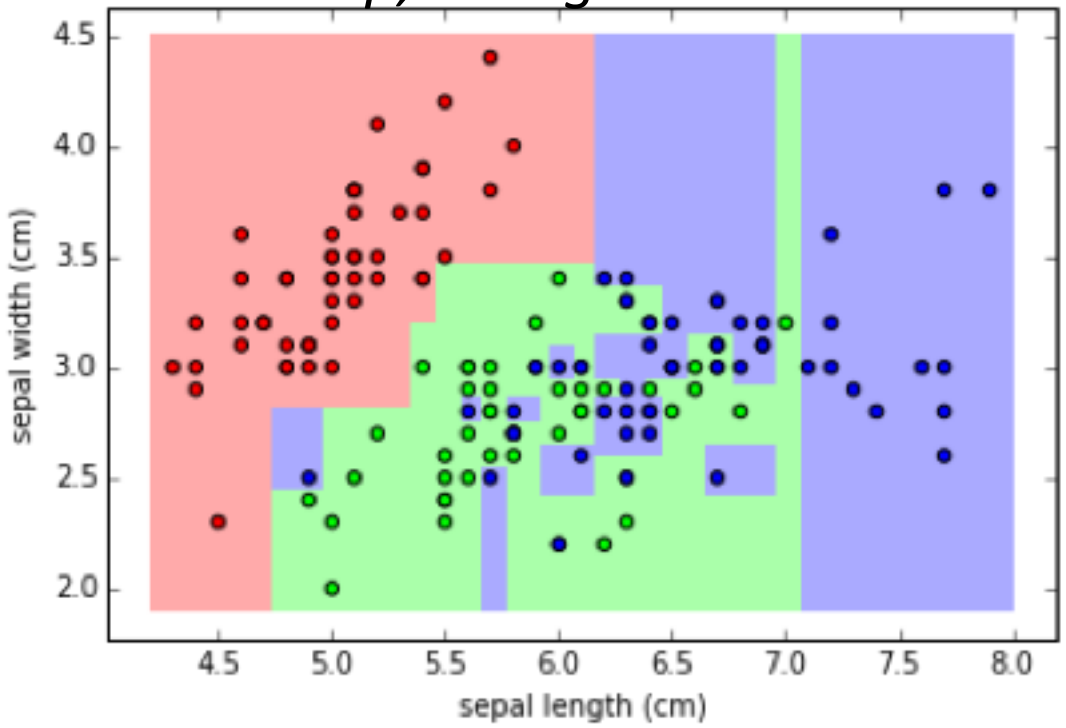
Model Regularization

Model Regularization

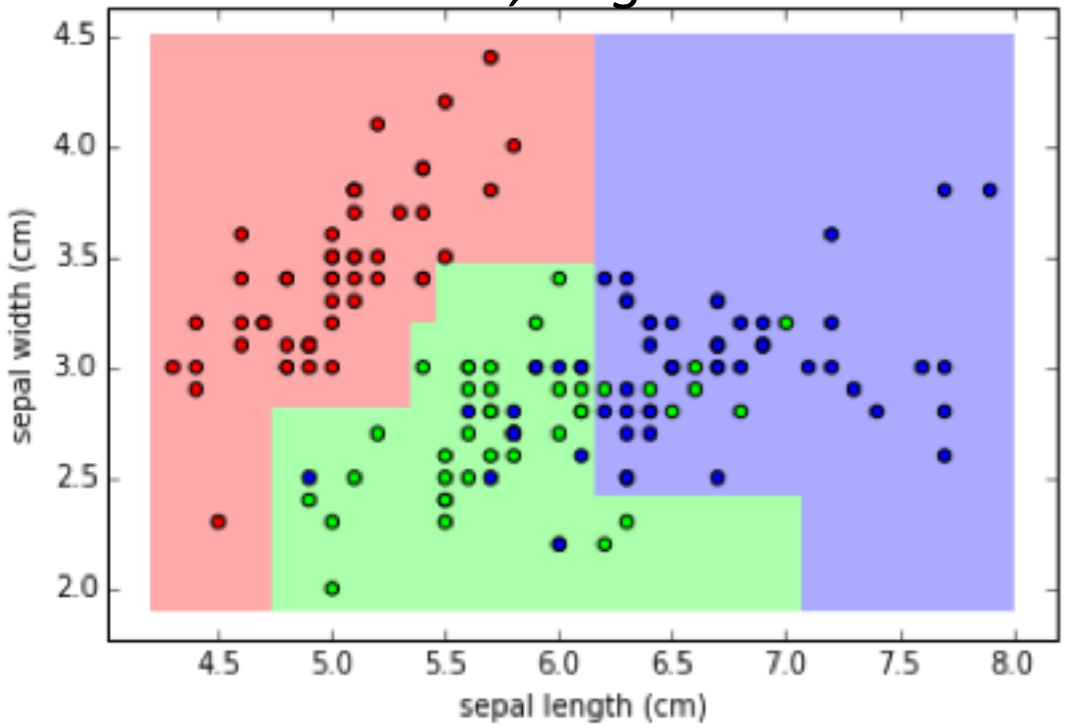
- Main complexity parameter is *max_depth* of the tree.
- Deep trees can split the data up more, leading to overfitting.
- Some nodes here have a single sample in it!

Model Regularization

Deep, Unregularized Tree



Shallower, Regularized Tree



Closing Notes

Great Demo

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

Final Nice Attributes - Proof Left as an Exercise to the Student ;)

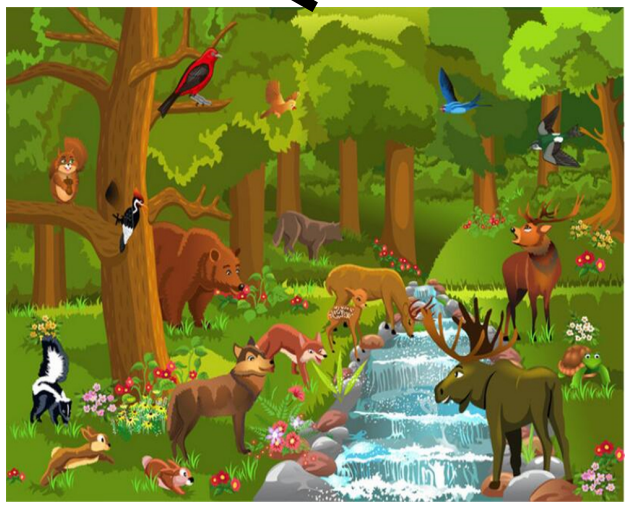
- Get Out Of Bag (OOB) error rate for free, which is equivalent to leave-one-out cross validation.
- In theory, Random Forest cannot overfit on the number of trees, i.e. you can’t have too many trees, it’s just more expensive to train more.
- Deals with missing values through “surrogate splits”.
- Efficiently ignores redundant/irrelevant variables.

The End

$\mathrm{Insight\_Fellow}_i$

Watched the clock/dozed off

Paid attention the entire time



Links

- https://www2.isye.gatech.edu/~tzhao80/Lectures/Lecture_6.pdf
- http://scikit-learn.org/stable/modules/tree.html
- https://stackoverflow.com/questions/20224526/how-to-extract-the-decision-rules-from-scikit-learn-decision-tree
- http://www.utdallas.edu/~nrr150130/cs7301/2016fa/lects/Lecture_10_Ensemble.pdf
- http://www2.stat.duke.edu/~rcs46/lectures_2017/08-trees/08-tree-advanced.pdf
- https://stackoverflow.com/questions/49170296/scikit-learn-feature-importance-calculation-in-decision-trees
- https://github.com/scikit-learn/scikit-learn/blob/18cdaa69c14a5c84ab03fce4fb5dc6cd77619e35/sklearn/tree/_tree.pyx#L1056
- https://github.com/scikit-learn/scikit-learn/blob/master/sklearn/tree/_criterion.pyx
- https://medium.com/@srnghn/the-mathematics-of-decision-trees-random-forest-and-feature-importance-in-scikit-learn-and-spark-f2861df67e3