Lecture 8
Random Forests

GEOL 4397: Data analytics and machine learning for geoscientists

Jiajia Sun, Ph.D.

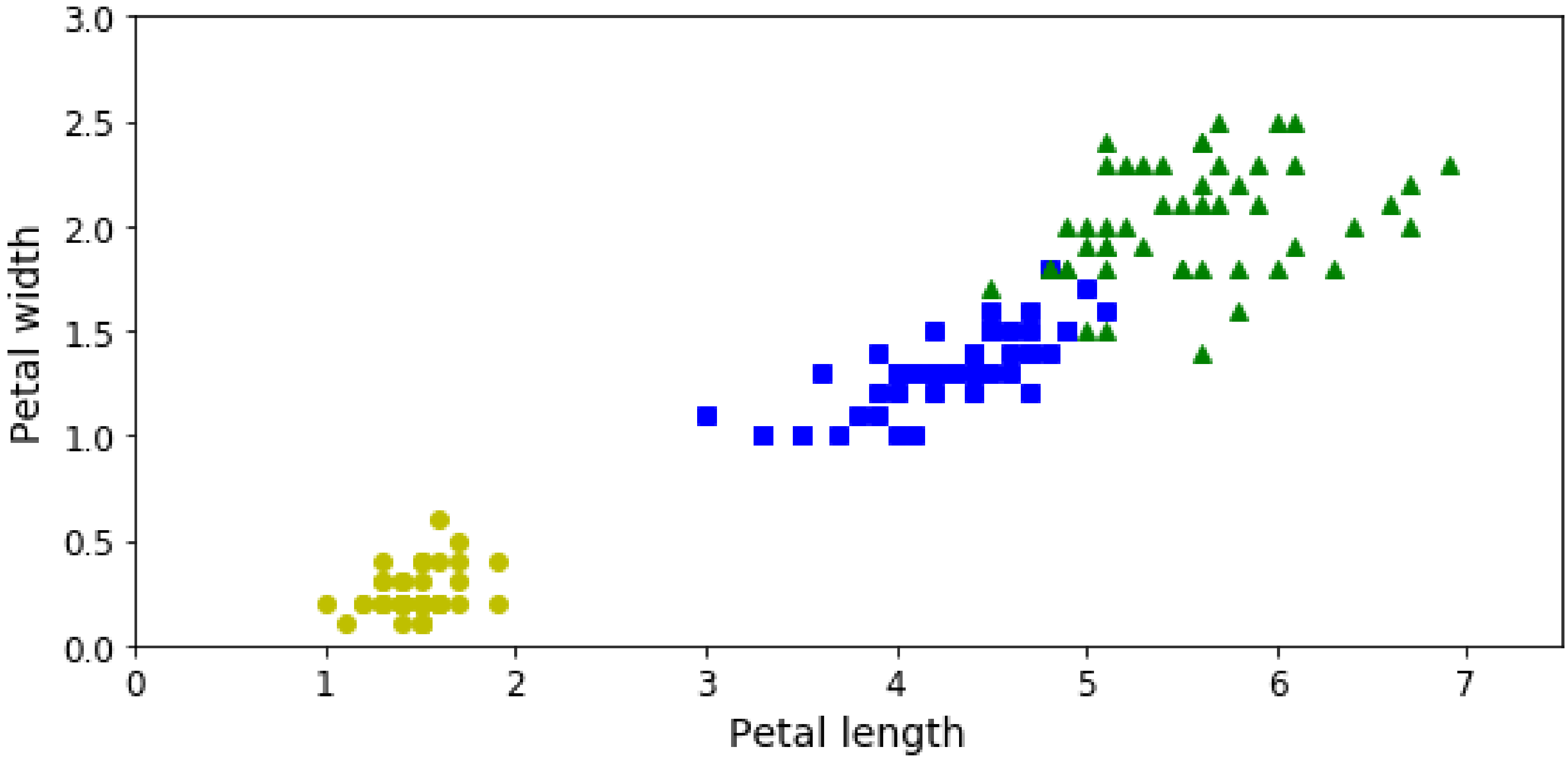
Feb 26th, 2019

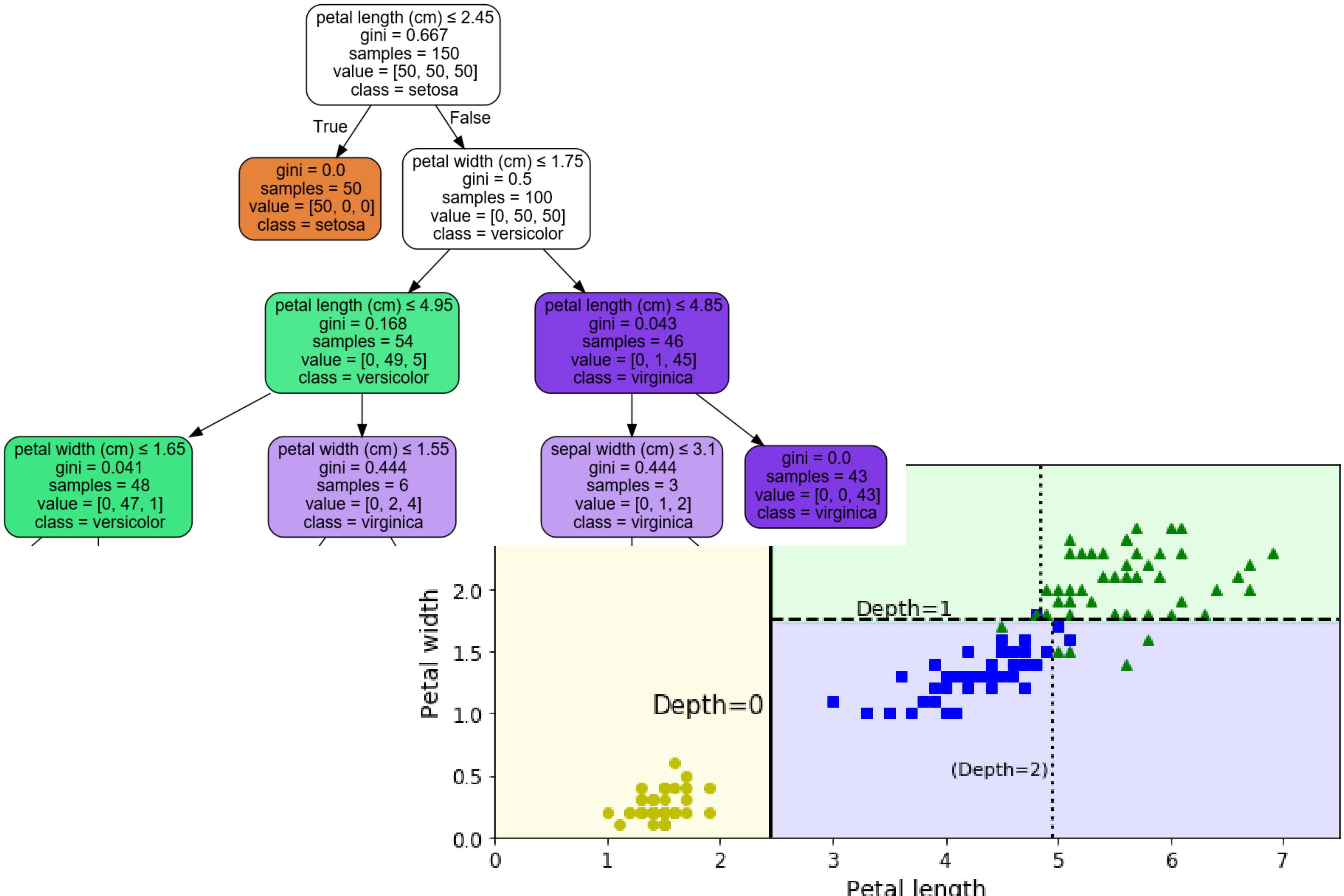
UNIVERSITY of HOUSTON
YOU ARE THE PRIDE
EARTH AND ATMOSPHERIC SCIENCES

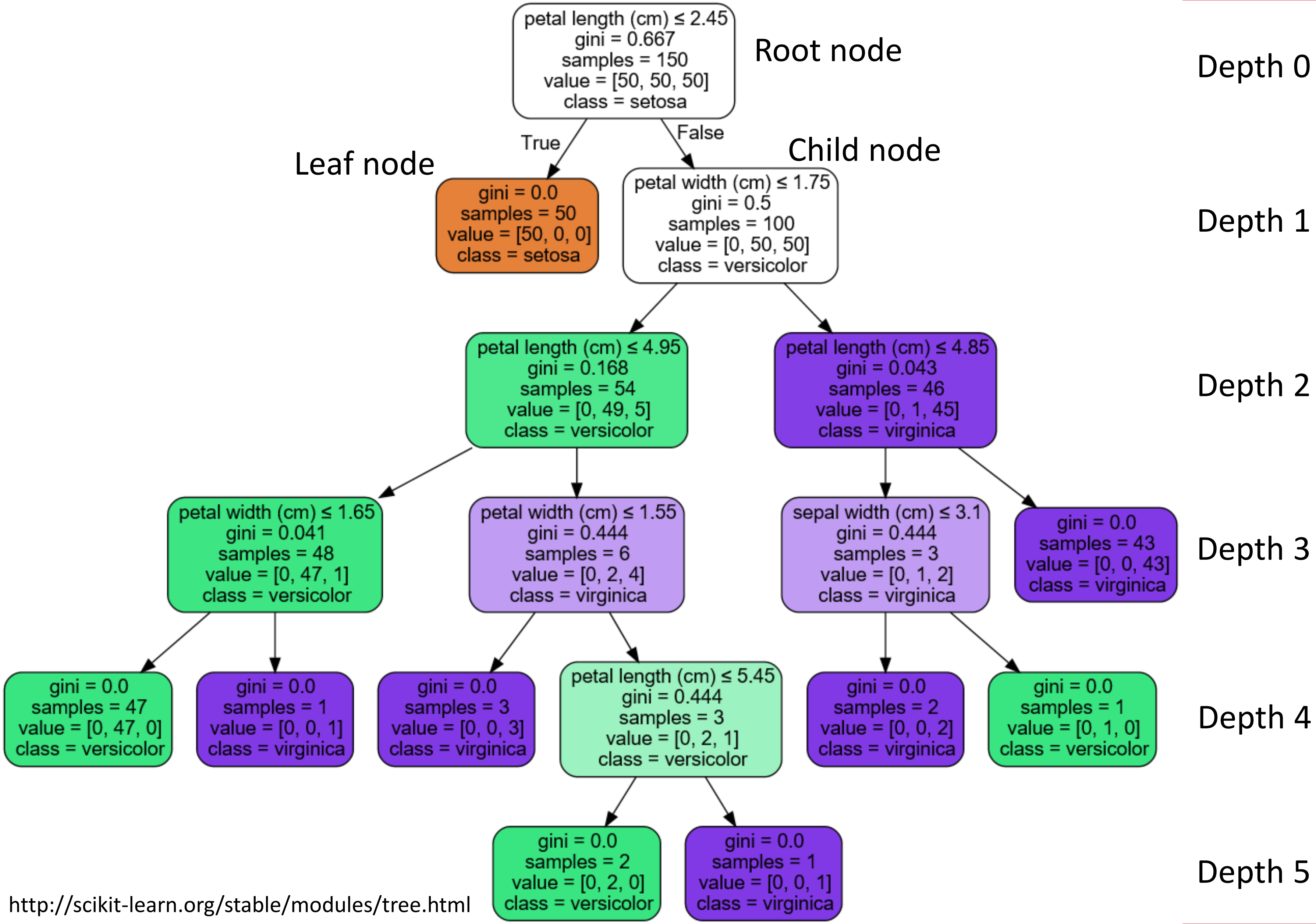
Agenda

- Decision Trees: review
- Random Forests: motivation
- Random Forests: concepts
- Random Forests: implementation

Iris data



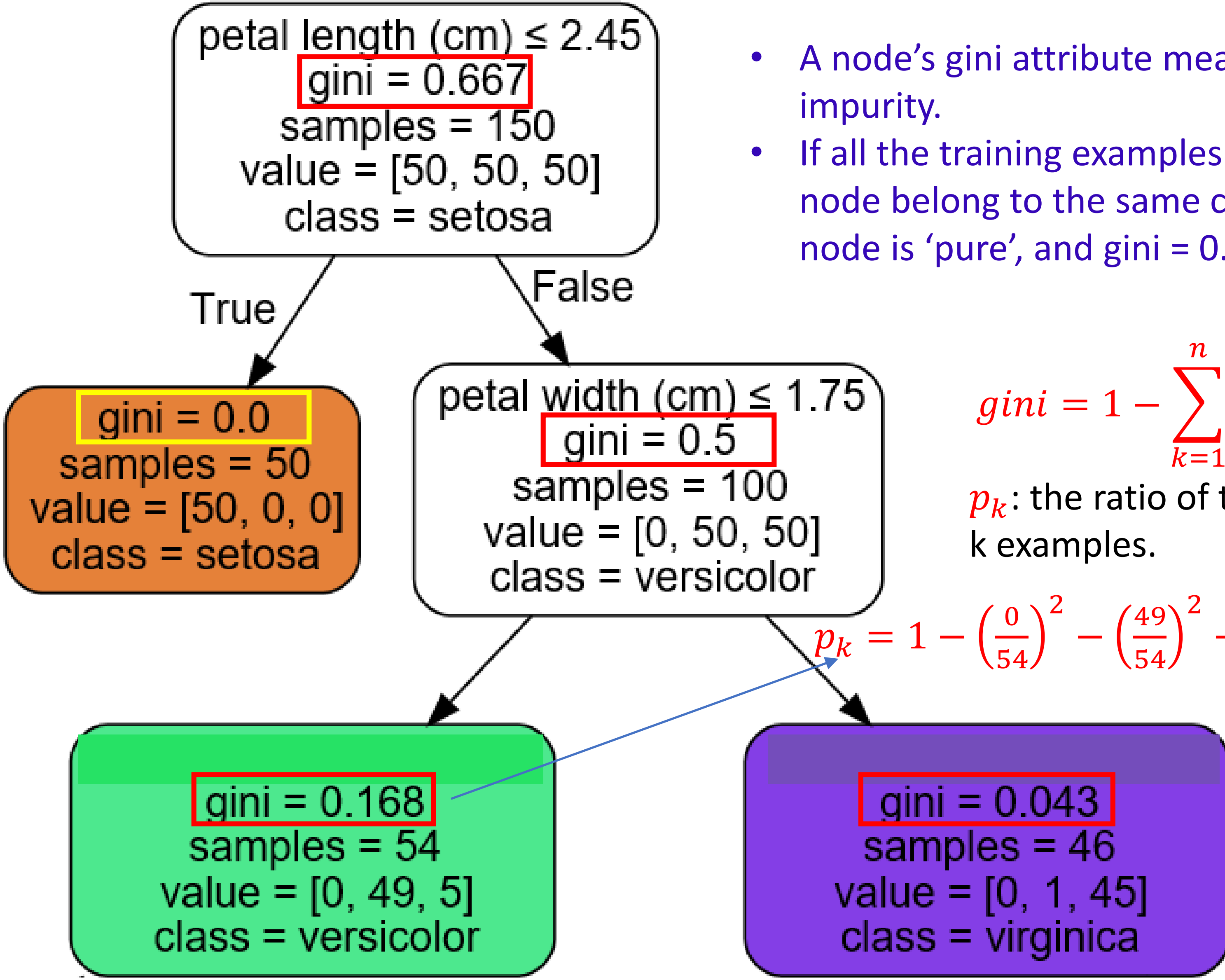

petal length (cm) ≤ 2.45
gini = 0.667
samples = 150
value = [50, 50, 50]
class = setosa
True
False
gini = 0.0
samples = 50
value = [50, 0, 0]
class = setosa
petal width (cm) ≤ 1.75
gini = 0.5
samples = 100
value = [0, 50, 50]
class = versicolor
petal length (cm) ≤ 4.95
gini = 0.168
samples = 54
value = [0, 49, 5]
class = versicolor
petal length (cm) ≤ 4.85
gini = 0.043
samples = 46
value = [0, 1, 45]
class = virginica
petal width (cm) ≤ 1.65
gini = 0.041
samples = 48
value = [0, 47, 1]
class = versicolor
petal width (cm) ≤ 1.55
gini = 0.444
samples = 6
value = [0, 2, 4]
class = virginica
sepal width (cm) ≤ 3.1
gini = 0.444
samples = 3
value = [0, 1, 2]
class = virginica
gini = 0.0
samples = 43
value = [0, 0, 43]
class = virginica
Depth=1
Depth=0
(Depth=2)
Petal width
Petal length
0.0
0.5
1.0
1.5
2.0
0
1
2
3
4
5
6
7



http://scikit-learn.org/stable/modules/tree.html

Splitting data and growing trees

- You need to know what question to ask at each node.
- How to choose which feature and which threshold value to use?



- A node's gini attribute measures its impurity.
- If all the training examples in a node belong to the same class, this node is 'pure', and gini = 0.

$$gini = 1 - \sum_{k=1}^{n} p_k^2$$

p_k: the ratio of the class k examples.

$$p_k = 1 - \left(\frac{0}{54}\right)^2 - \left(\frac{49}{54}\right)^2 - \left(\frac{5}{54}\right)^2 = 0.168$$

CART algorithm

- Scikit-Learn uses Classification And Regression Tree (CART) to grow (or train) decision trees.
- The idea is simple: the algorithm splits the training set in two subsets using a single feature k and a threshold t_k (e.g., petal length <= 2.45 cm)
- It searches for the pair (k, t_k) that produces the *purest* subsets
- by minimizing a cost function …

CART: cost function

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

g_{left}: the impurity of the left subset

g_{right}: the impurity of the right subset

The tree stops growing once it reaches the maximum depth (max_depth), or if it cannot find a split that will reduce impurity.

Decision Trees

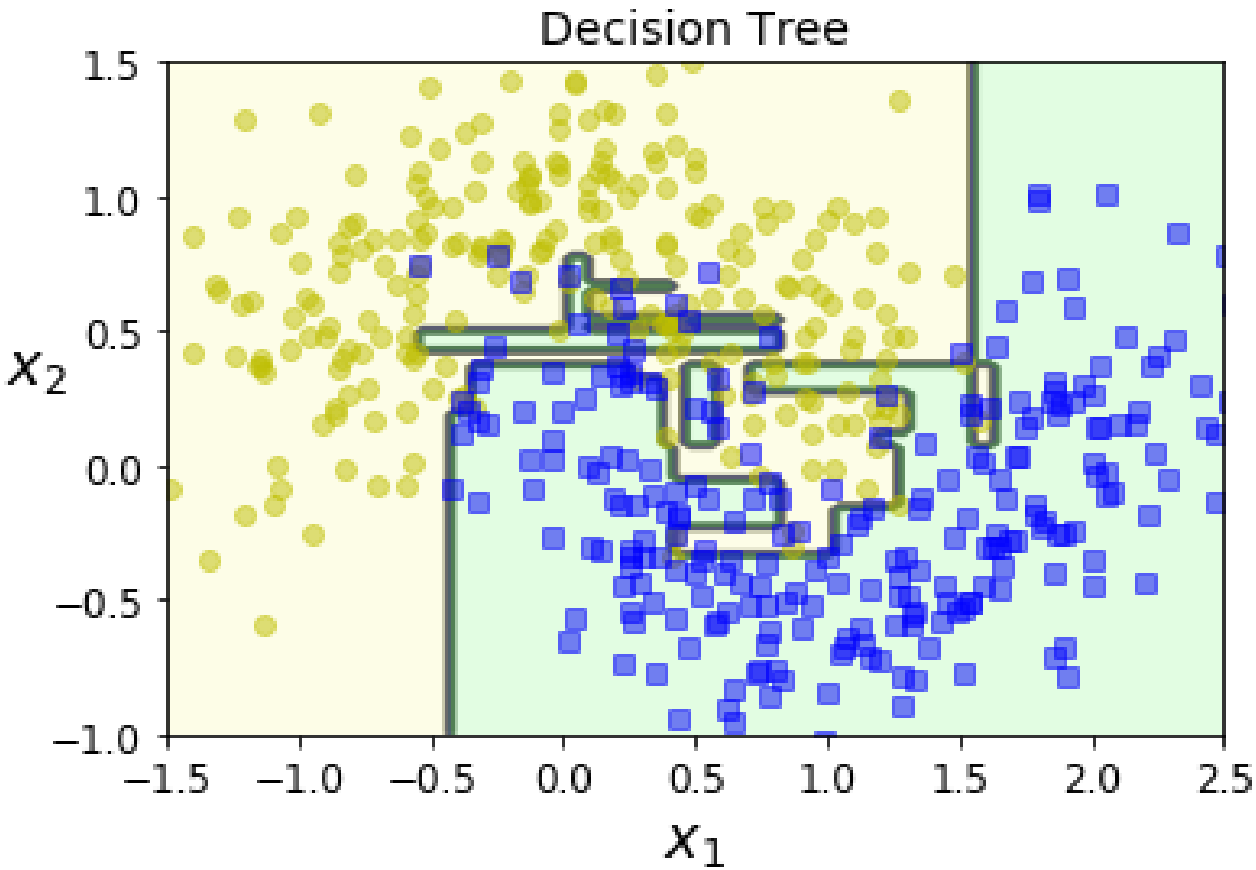
- Iteratively split the data by asking and answering a question
- Very intuitive
- Independent of scaling (no scaling needed)

Decision Trees

- Iteratively split the data by asking and answering a question
- Very intuitive
- Independent of scaling (no scaling needed)

- However, they tend to overfit the data (if the trees grow very deep) ----> high variance
- Very sensitive to small variations in the training data

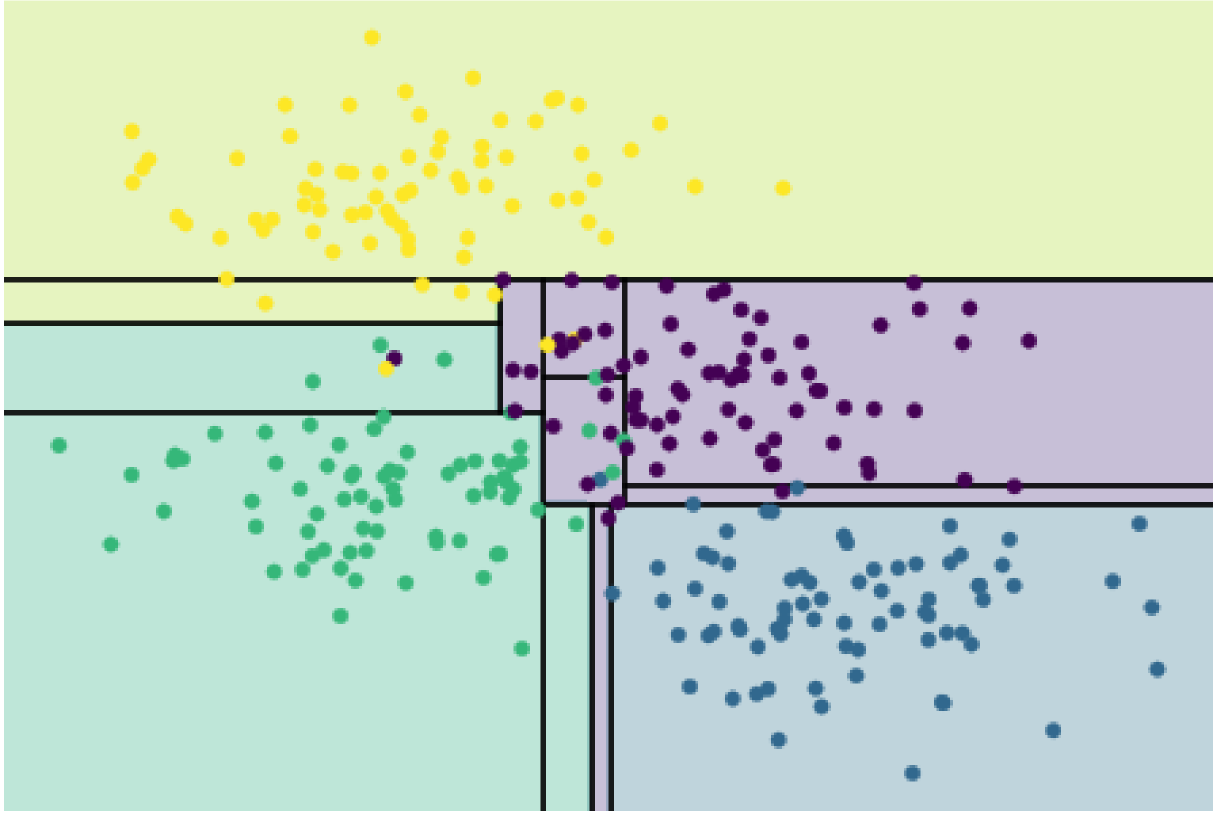
overfitting



Decision boundary from a single decision tree

Aurelien Geron, 2017, Hands-on Machine Learning with Scikit-Learn & TensorFlow, pp 187

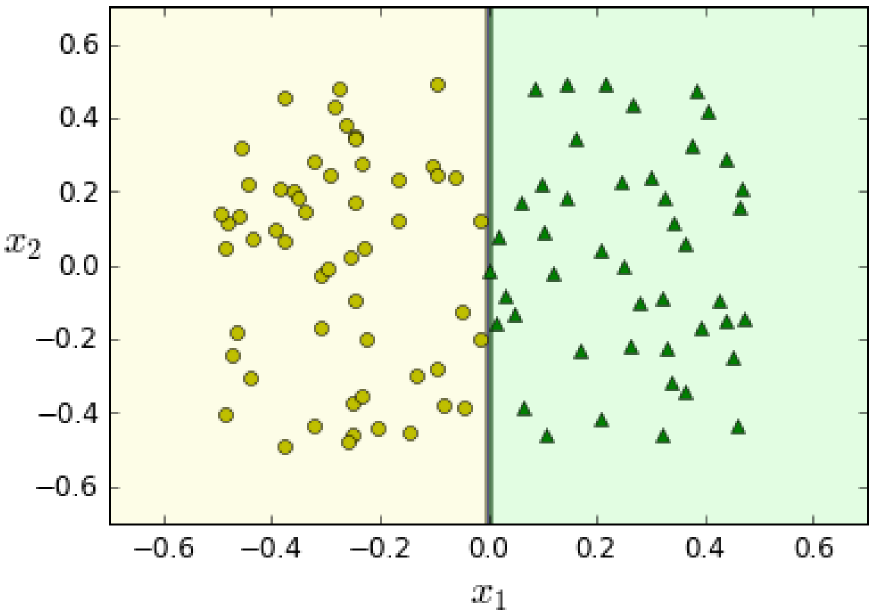
overfitting



Decision boundaries of a decision tree with max_depth = 5

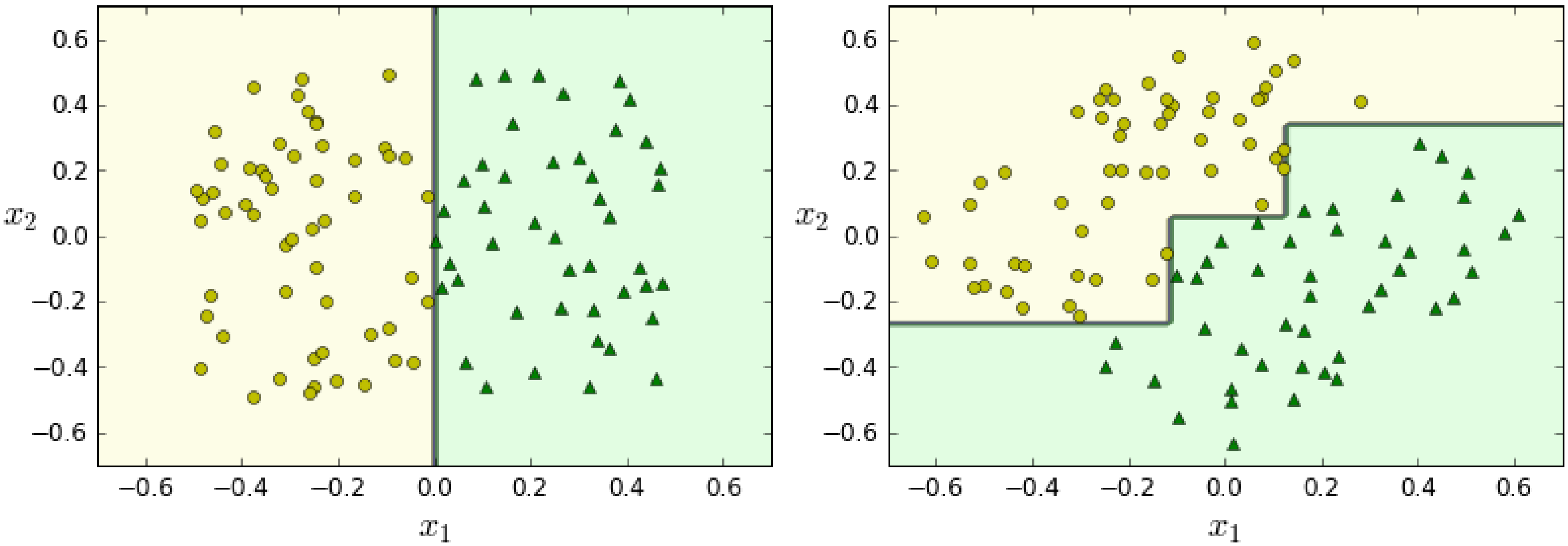
Jake VanderPlas, 2016, Python Data Science Handbook, pp 424

Sensitivity to training set rotation



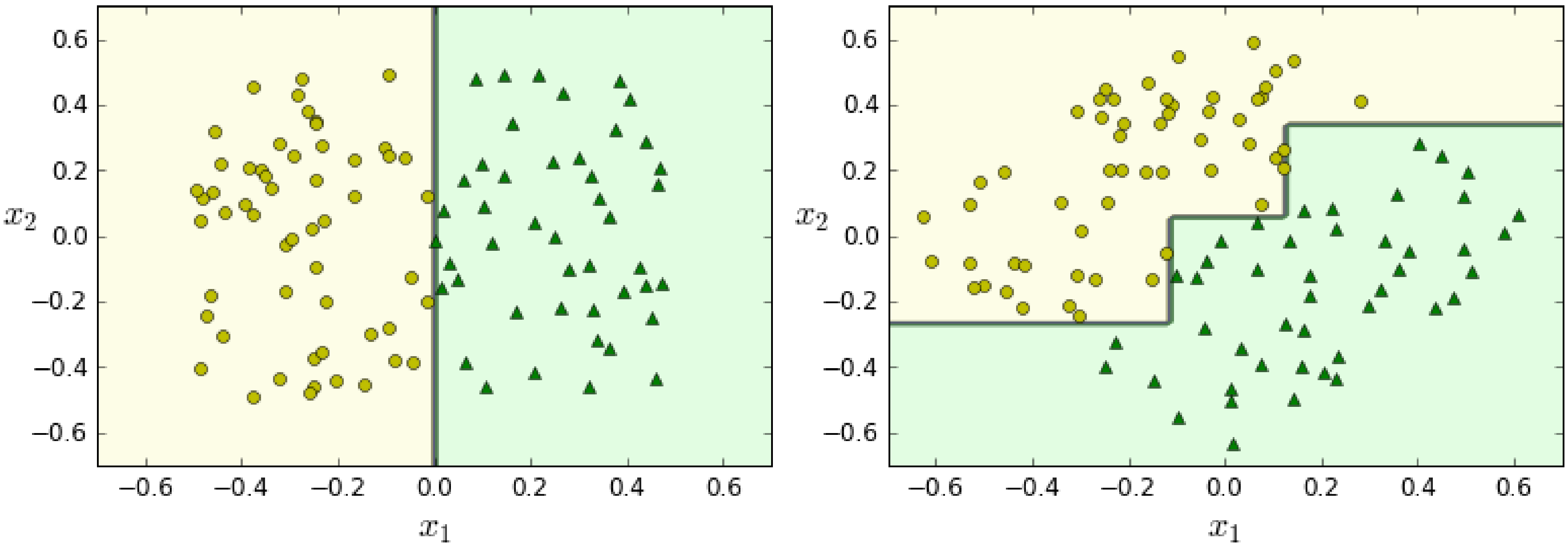
Aurelien Geron, 2017, Hands-on Machine Learning with Scikit-Learn & TensorFlow, pp 177

Sensitivity to training set rotation



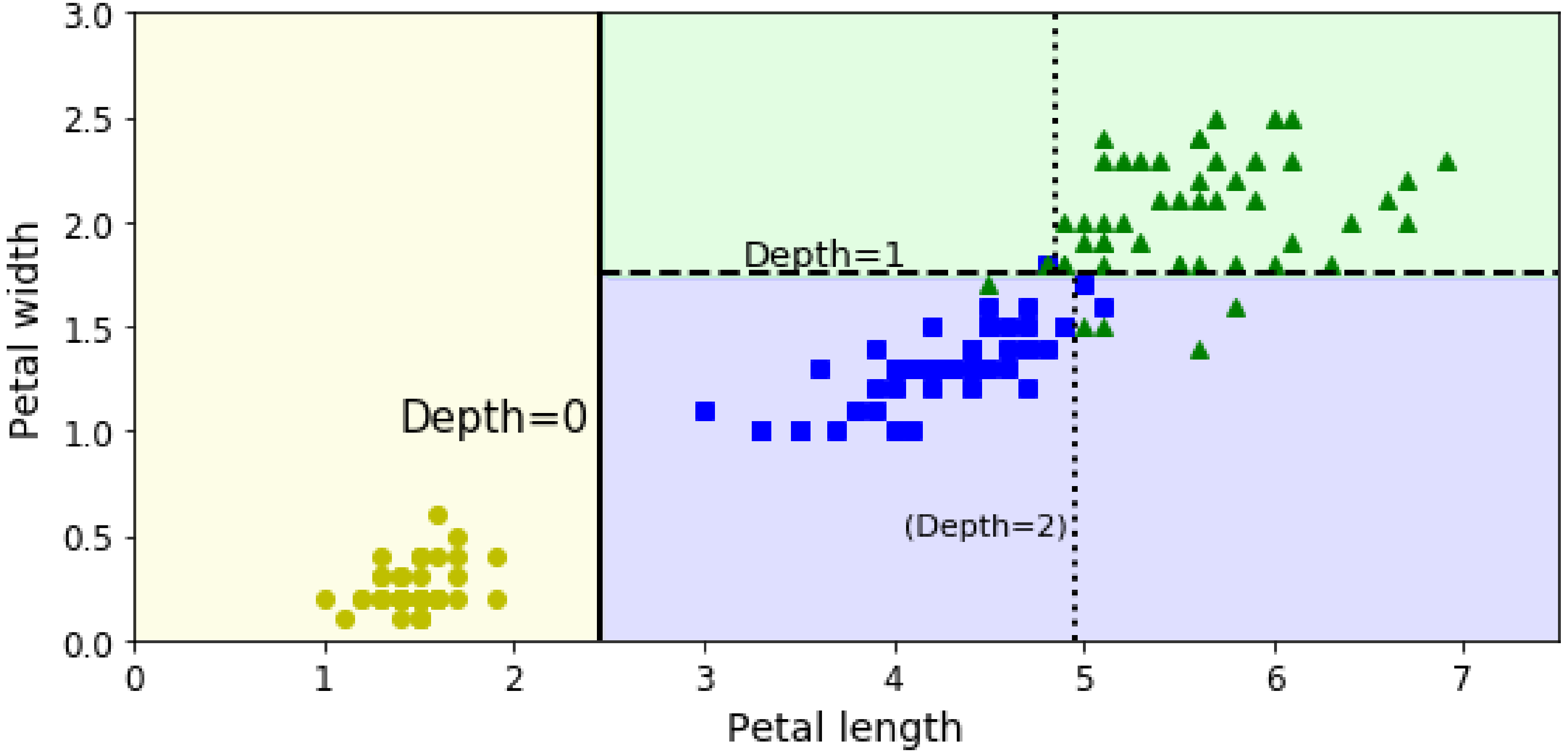
Aurelien Geron, 2017, Hands-on Machine Learning with Scikit-Learn & TensorFlow, pp 177

Sensitivity to training set rotation



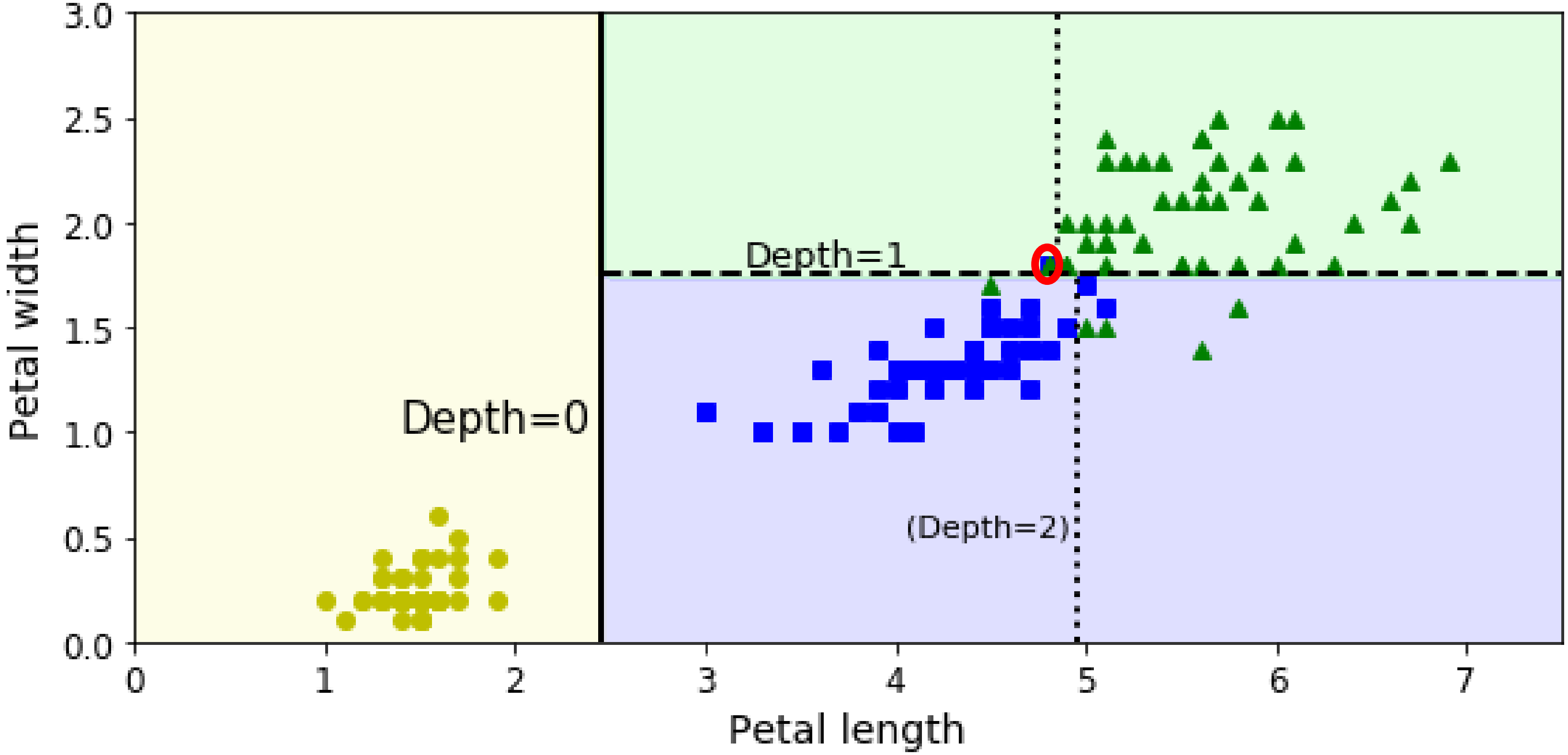
Decision boundary looks unnecessarily convoluted. Will not generalize well.

Aurelien Geron, 2017, Hands-on Machine Learning with Scikit-Learn & TensorFlow, pp 177



Aurelien Geron, 2017, Hands-on Machine Learning with Scikit-Learn & TensorFlow, pp 170

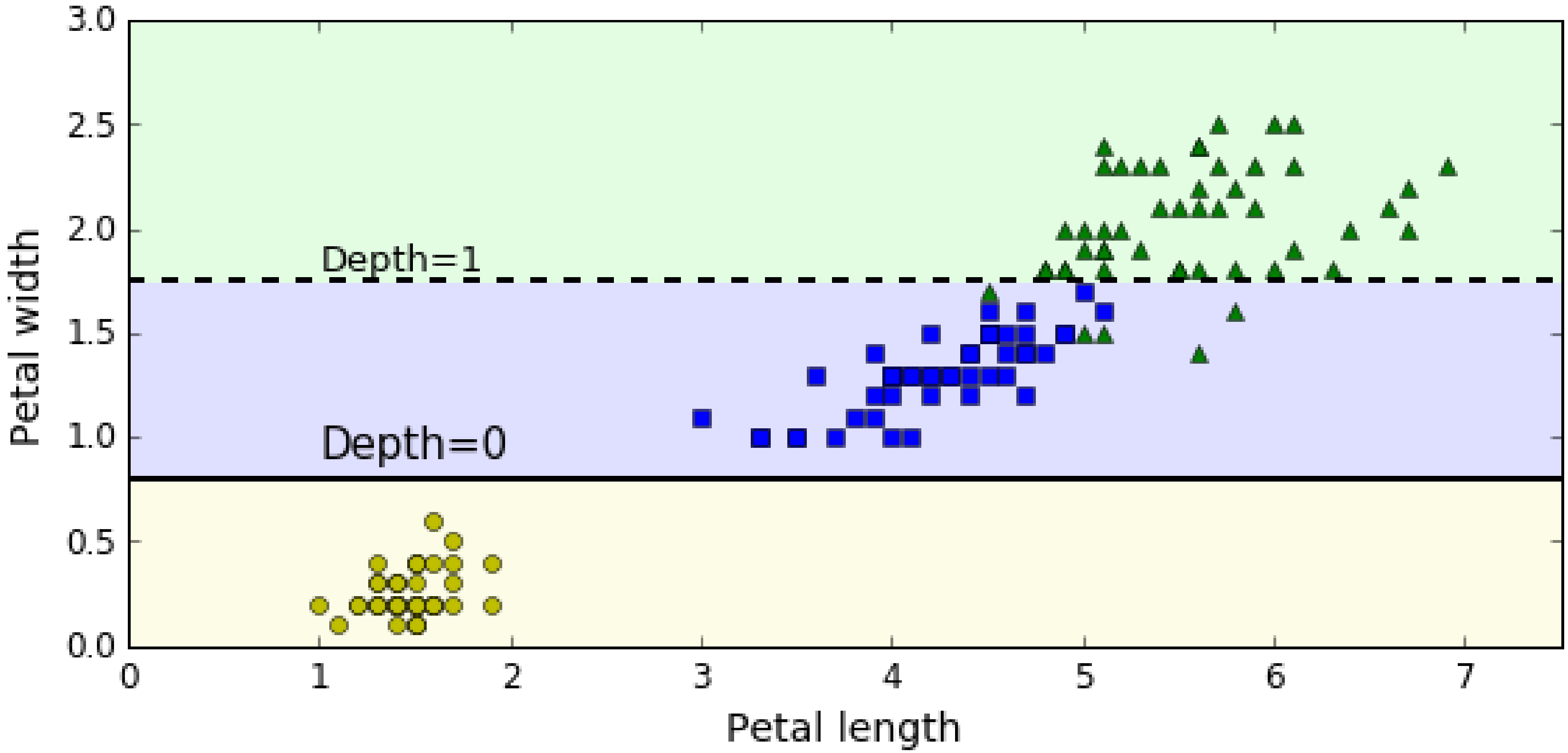
Sensitivity to small variations



Remove the widest Iris-Versicolor

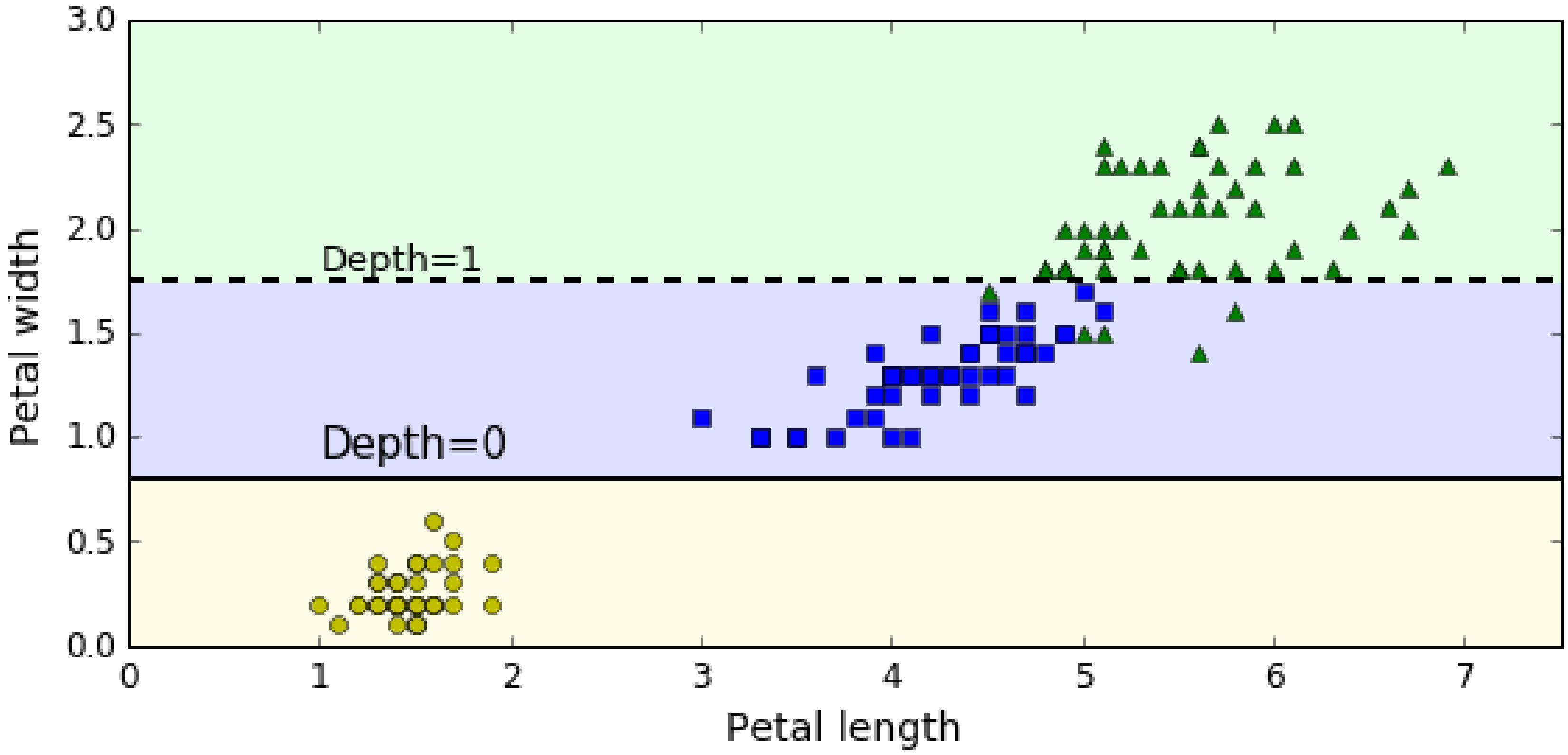
Aurelien Geron, 2017, Hands-on Machine Learning with Scikit-Learn & TensorFlow, pp 170

Sensitivity to small variations



Aurelien Geron, 2017, Hands-on Machine Learning with Scikit-Learn & TensorFlow, pp 178

Sensitivity to small variations



Random Forests can limit the instability by averaging predictions over many trees.

Aurelien Geron, 2017, Hands-on Machine Learning with Scikit-Learn & TensorFlow, pp 178

Random Forests

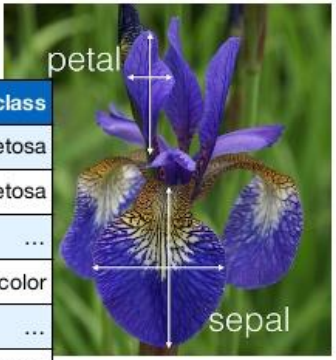
- An ensemble of decision trees
- Trained on random subsets of the original dataset
- Use averaging (or aggregating) to improve the prediction accuracy and control overfitting

http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

Nomenclature

IRIS

https://archive.ics.uci.edu/ml/datasets/Iris



Instances (samples, observations)

	sepal_length	sepal_width	petal_length	petal_width	class
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
...	...	...	...	...	...
50	6.4	3.2	4.5	1.5	veriscolor
...	...	...	...	...	...
150	5.9	3.0	5.1	1.8	virginica

Features (attributes, dimensions)

Classes (targets)

$$\mathbf{X} = \begin{bmatrix} x_{11} & x_{12} & \cdots & x_{1D} \\ x_{21} & x_{22} & \cdots & x_{2D} \\ x_{31} & x_{32} & \cdots & x_{3D} \\ \vdots & \vdots & \ddots & \vdots \\ \vdots & \vdots & \ddots & \vdots \\ x_{N1} & x_{N2} & \cdots & x_{ND} \end{bmatrix}$$

$$\mathbf{y} = [y_1, y_2, y_3, \cdots y_N]$$

https://www.slideshare.net/SebastianRaschka/nextgen-talk-022015

Random Sampling

- Randomly sampling features

Random Sampling

- Randomly sampling features

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa

Random Sampling

- Randomly sampling features

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa

Random Sampling

- Randomly sampling features

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa

Random Sampling

- Randomly sampling features
- Randomly sampling instances

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa

Random Sampling

- Randomly sampling features
- Randomly sampling instances

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa

Random Sampling

- Randomly sampling features
- Randomly sampling instances

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa

Random Sampling

- Randomly sampling features
- Randomly sampling instances

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa

Random Sampling

- Randomly sampling features
- Randomly sampling instances

Bootstrap samples

- random sampling with replacement

Random Sampling

- Randomly sampling features
- Randomly sampling instances

Bootstrap samples

- random sampling with replacement

Random Patches

- Sampling both training instances and features

Geron, A., 2017, Hands-on Machine Learning with Scikit-Learn & TensorFlow, pp188

Random Sampling

- Randomly sampling features
- Randomly sampling instances

Bootstrap samples

- rando

Random

- Samp

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa

Geron, A., 2017, Hands-on Machine Learning with Scikit-Learn & TensorFlow, pp188

Random Sampling

- Randomly sampling features
- Randomly sampling instances

Bootstrap samples

- rand[...]

Random [...]

- Samp[...]

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa

Geron, A., 2017, Hands-on Machine Learning with Scikit-Learn & TensorFlow, pp188

Random Sampling

- Randomly sampling features
- Randomly sampling instances

Bootstrap samples

- rand

Random

- Samp

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6		1.5		setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4		1.7		setosa
7	4.6		1.4		setosa
8	5.0	3.4	1.5	0.2	setosa

Geron, A., 2017, Hands-on Machine Learning with Scikit-Learn & TensorFlow, pp188

Random Sampling

- Randomly sampling features
- Randomly sampling instances

Bootstrap samples

- random sampling with replacement

Random Patches

- Sampling both training instances and features

Random Subspaces

- Keeping all the training instances but sampling features

Geron, A., 2017, Hands-on Machine Learning with Scikit-Learn & TensorFlow, pp188

Random Sampling

- Randomly sampling features
- Randomly sampling instances

Bootstrap samples

- rand

Random

- Samp

Random

- Keepi es

	Sepal.Length	Sepal.Width	Petal.Length	Petal.Width	Species
1	5.1	3.5	1.4	0.2	setosa
2	4.9	3.0	1.4	0.2	setosa
3	4.7	3.2	1.3	0.2	setosa
4	4.6	3.1	1.5	0.2	setosa
5	5.0	3.6	1.4	0.2	setosa
6	5.4	3.9	1.7	0.4	setosa
7	4.6	3.4	1.4	0.3	setosa
8	5.0	3.4	1.5	0.2	setosa

Geron, A., 2017, Hands-on Machine Learning with Scikit-Learn & TensorFlow, pp188

How does RF work?

http://scikit-learn.org/stable/modules/ensemble.html#forest

How does RF work?

- Select the number of decision trees, M

http://scikit-learn.org/stable/modules/ensemble.html#forest

How does RF work?

- Select the number of decision trees, M
- For each decision tree:
- create bootstrap samples of training instances

http://scikit-learn.org/stable/modules/ensemble.html#forest

How does RF work?

- Select the number of decision trees, M
- For each decision tree:
- create bootstrap samples of training instances
- grow a decision tree

http://scikit-learn.org/stable/modules/ensemble.html#forest

How does RF work?

- Select the number of decision trees, M
- For each decision tree:
- create bootstrap samples of training instances
- grow a decision tree

when splitting a node, instead of searching for the best feature among all features, just search for the best feature among a random subset of the features

http://scikit-learn.org/stable/modules/ensemble.html#forest

How does RF work?

- Select the number of decision trees, M
- For each decision tree:
- create bootstrap samples of training instances
- grow a decision tree

  when splitting a node, instead of searching for the best feature among all features, just search for the best feature among a random subset of the features
- Average predictions from all M decision trees

http://scikit-learn.org/stable/modules/ensemble.html#forest

How does RF work?

- Select the number of decision trees, M
- For each decision tree:
-     create bootstrap samples of training instances
-     grow a decision tree

    when splitting a node, instead of searching for the best feature among all features, just search for the best feature among a random subset of the features

- Average predictions from all M decision trees
- In Scikit-Learn, there is a RandomForestClassifier.

http://scikit-learn.org/stable/modules/ensemble.html#forest

History

- The first algorithm for random forests was created by *Tin Kam Ho* using random subspace method (i.e., using random samples of features instead of the entire feature set)

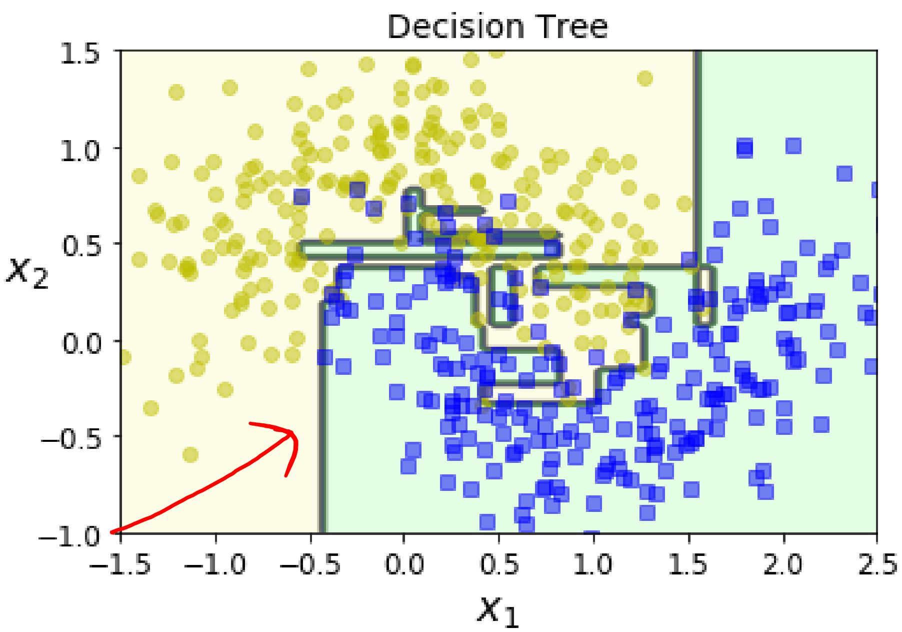
https://en.wikipedia.org/wiki/Random_forest
https://en.wikipedia.org/wiki/Random_subspace_method

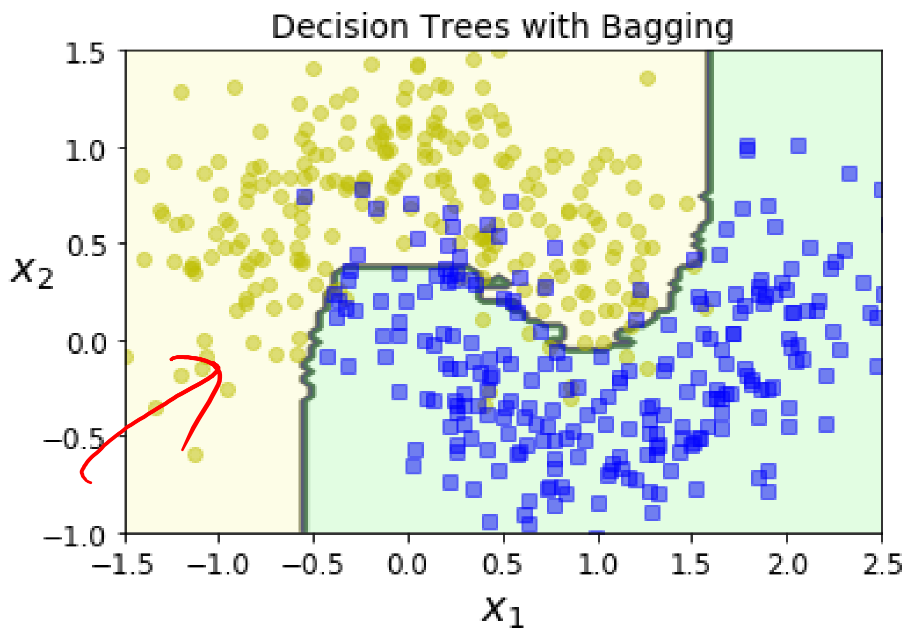
History

- The first algorithm for random forests was created by *Tin Kam Ho* using random subspace method (i.e., using random samples of features instead of the entire feature set)

- An extension developed by *Leo Breiman* and *Adele Cutler* using random patches (i.e., using random samples of training instances and features)

https://en.wikipedia.org/wiki/Random_forest
https://en.wikipedia.org/wiki/Random_subspace_method





Left: Decision boundary of a single Decision Tree with unlimited depths using the moons data set.

Right: Average decision boundary of an ensemble of 500 decision trees

Observation: Averaging over 500 decision trees results in a smaller variance, and better prediction accuracy on new data.

Aurelien Geron, 2017, Hands-on Machine Learning with Scikit-Learn & TensorFlow, pp 187

Extremely Randomized Trees

- For a random forest, at each node only a random subset of the features is considered for splitting

http://scikit-learn.org/stable/modules/ensemble.html#forest

Extremely Randomized Trees

- For a random forest, at each node only a random subset of the features is considered for splitting
- Instead of looking for the most discriminative thresholds, thresholds are drawn at random for each candidate feature and the best of these randomly-generated thresholds is picked for splitting.

http://scikit-learn.org/stable/modules/ensemble.html#forest

Extremely Randomized Trees

- For a random forest, at each node only a random subset of the features is considered for splitting
- Instead of looking for the most discriminative thresholds, thresholds are drawn at random for each candidate feature and the best of these randomly-generated thresholds is picked for splitting.
- Extra-Trees for short (ExtraTreeClassifier in Scikit-Learn)

http://scikit-learn.org/stable/modules/ensemble.html#forest

Extremely Randomized Trees

- For a random forest, at each node only a random subset of the features is considered for splitting
- Instead of looking for the most discriminative thresholds, thresholds are drawn at random for each candidate feature and the best of these randomly-generated thresholds is picked for splitting.
- Extra-Trees for short (ExtraTreeClassifier in Scikit-Learn)
- It is hard to tell in advance whether a RandomForestClassifier will perform better or worse than an ExtraTreeClassifier.

http://scikit-learn.org/stable/modules/ensemble.html#forest

Feature importance

- With Random Forests (or, Decision Trees), it is fairly straightforward to measure the relative importance of each feature.

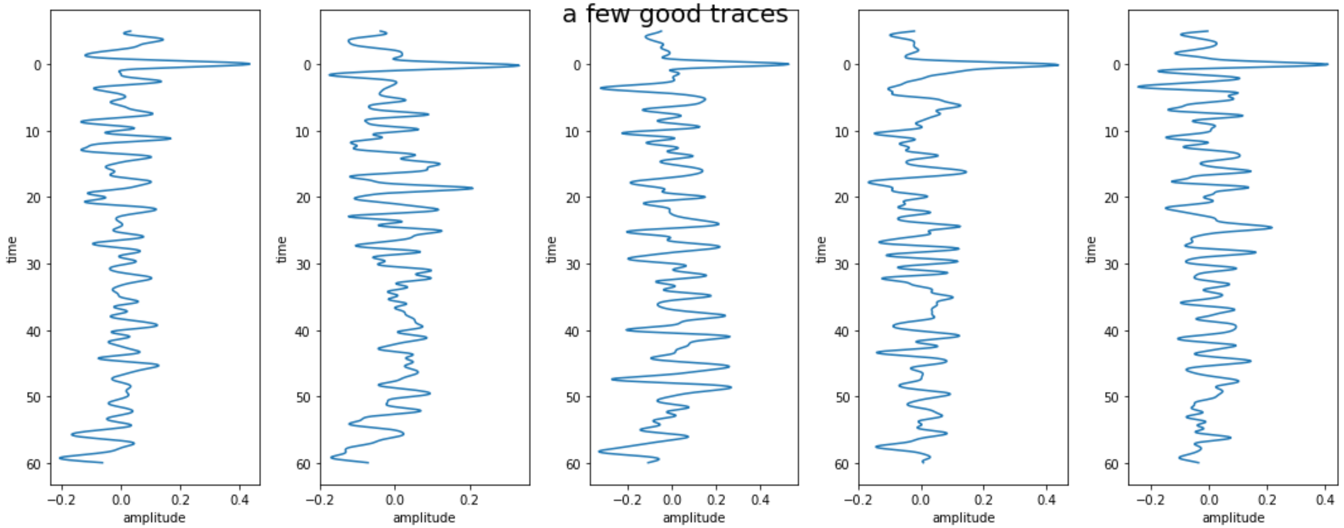
Feature importance

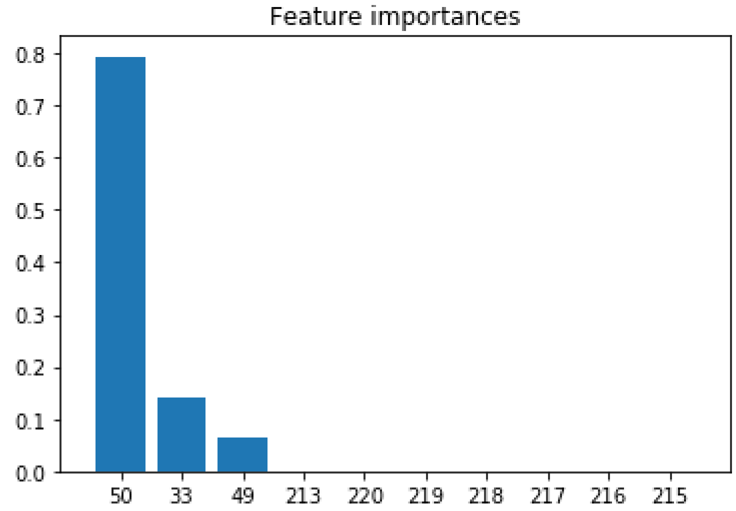
- With Random Forests (or, Decision Trees), it is fairly straightforward to measure the relative importance of each feature.
- Scikit-Learn measures a feature's importance by looking at how much the tree nodes that use that feature reduces impurity on average.

Feature importance

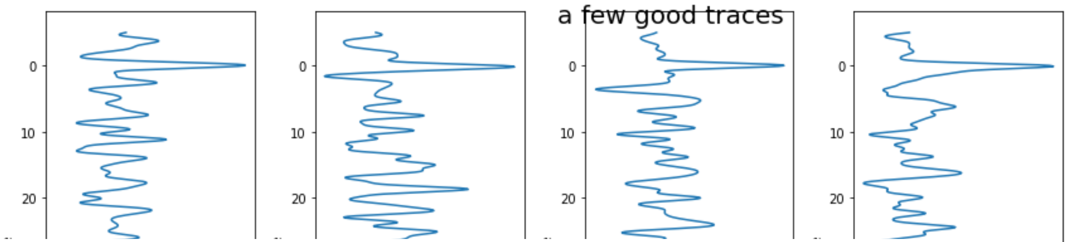
- With Random Forests (or, Decision Trees), it is fairly straightforward to measure the relative importance of each feature.
- Scikit-Learn measures a feature's importance by looking at how much the tree nodes that use that feature reduces impurity on average.
- Scikit-Learn computes the feature importance automatically.
- You can access the result using the feature_importances_ variable.

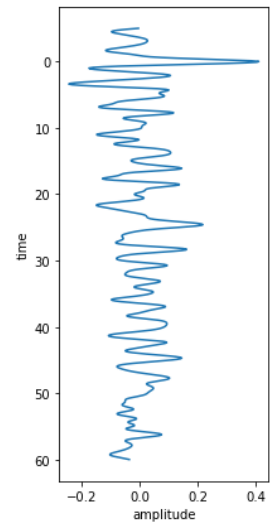
Example: Classifying seismic P-wave receiver functions





Example: Classifying seismic P-wave receiver functions





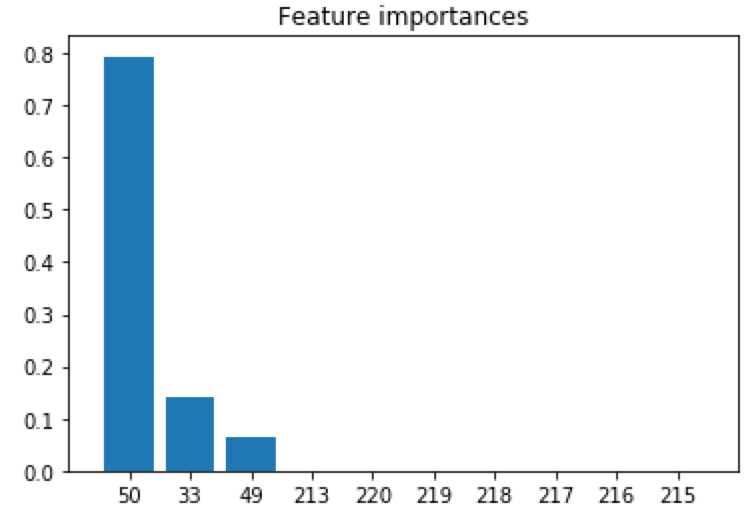

```python
# The following code is based on a modification of the codes in this webpage
# http://scikit-learn.org/stable/auto_examples/ensemble/plot_forest_importances.html
importances = tree_clf_2.feature_importances_
indices = np.argsort(importances)[::-1]

# Print the feature ranking
print("Feature ranking:")

for f in range(10):
    print("%d. feature %d (%f)" % (f + 1, indices[f], importances[indices[f]]))

# Plot the feature importances of the forest
plt.figure()
plt.title("Feature importances")
plt.bar(range(10), importances[indices][:10])
plt.xticks(range(10), indices)
plt.xlim([-1, 10])
plt.show()
```

```
Feature ranking:
1. feature 50 (0.793811)
2. feature 33 (0.141697)
3. feature 49 (0.064491)
4. feature 213 (0.000000)
5. feature 220 (0.000000)
6. feature 219 (0.000000)
7. feature 218 (0.000000)
8. feature 217 (0.000000)
9. feature 216 (0.000000)
10. feature 215 (0.000000)
```



Feature importance

- Random Forests are very handy to get a quick understanding of what features are important
- Very useful for feature selection

Understanding Random Forests

- Forest: a collection of trees
- Random: trees are trained on random subsets of training instances and features

Understanding Random Forests

- Forest: a collection of trees
- Random: trees are trained on random subsets of training instances and features
- Therefore, a random forest refers to a collection (or, an ensemble) of decision trees trained on random subsets of the original data set.

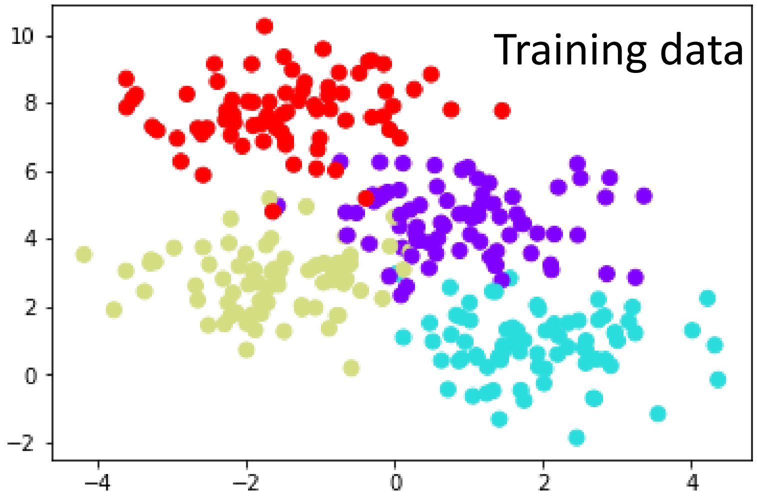
Understanding Random Forests

- Forest: a collection of trees
- Random: trees are trained on random subsets of training instances and features
- Therefore, a random forest refers to a collection (or, an ensemble) of decision trees trained on random subsets of the original data set.
- Prediction is made by aggregating the votes from all the trees for a classification task (or averaging the predicted values for a regression task)

Implementation in Scikit-Learn

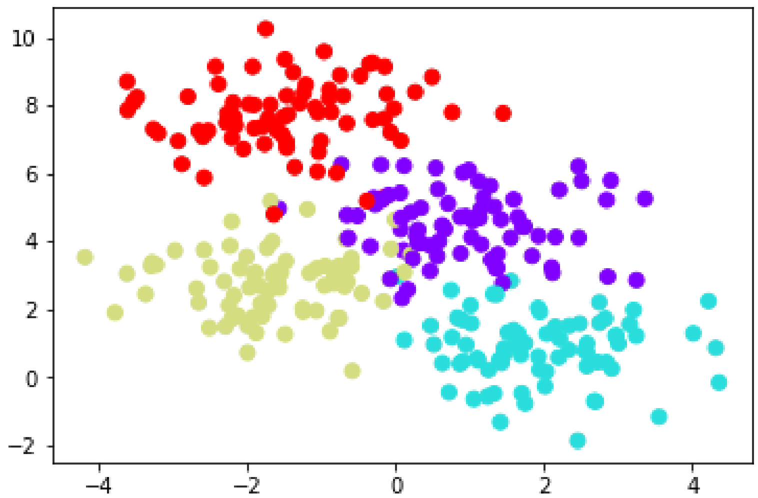
```
import numpy as np
import matplotlib.pyplot as plt
from sklearn.datasets import make_blobs

X, y = make_blobs(n_samples=300, centers=4,
                  random_state=0, cluster_std=1.0)
plt.scatter(X[:, 0], X[:, 1], c=y, s=50, cmap='rainbow');
```

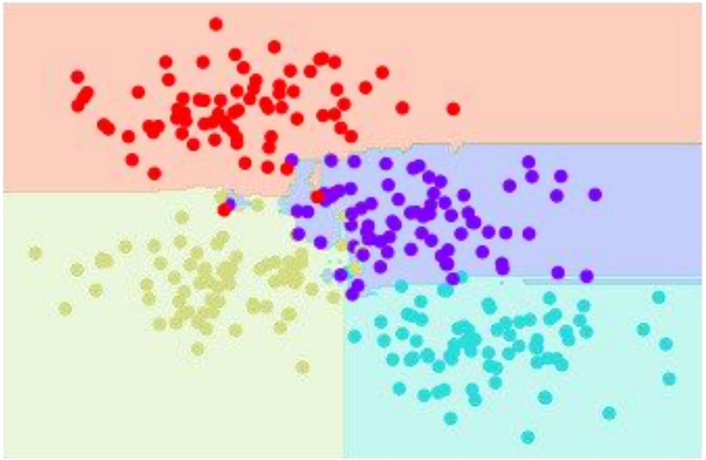


Implementation in Scikit-Learn

```
from sklearn.ensemble import RandomForestClassifier
model = RandomForestClassifier(n_estimators=100, random_state=0)
model.fit(X,y)
```



Training data



Decision boundary learned from a random forest comprising 100 trees