Lesson 9: Decision trees

- Decision trees intro (I do)
- Independent exercise (We do)
  - Tree induction with *python* and *scikit-learn*
- Debrief (You do)
  - Visualizing decision trees
  - When to use decision trees
    - Pitfalls and things to avoid
  - Next steps

Decision trees

Advantages:

- **Elegantly** provides **intuitive** insights into model function
  - Neural networks, SVM's, etc. ≈ black box
- Very useful for **feature selection**
  - Crucial for the *accuracy* and *efficiency* of your model

Decision trees

Advantages:

- **Elegantly** provides **intuitive** insights into model function
  - Neural networks, SVM's, etc. ≈ black box
- Very useful for **feature selection**
  - Crucial for the *accuracy* and *efficiency* of your model
- Included in most ML libraries (including scikit-learn)

So… *Let's try it out!*

Decision tree demo

Iris setosa

Iris versicolor

Iris virginica

	sepal length (cm)	sepal width (cm)	petal length (cm)	petal width (cm)	species
115	6.4	3.2	5.3	2.3	virginica
121	5.6	2.8	4.9	2.0	virginica
73	6.1	2.8	4.7	1.2	versicolor
58	6.6	2.9	4.6	1.3	versicolor
147	6.5	3.0	5.2	2.0	virginica
112	6.8	3.0	5.5	2.1	virginica
70	5.9	3.2	4.8	1.8	versicolor

Decision tree demo

```
 1 #!/usr/bin/env python
 2
 3 ## Import dataset
 4 import numpy as np
 5 from sklearn import tree
 6 from sklearn.datasets import load_iris
 7 iris = load_iris()
 8
 9 ## Split training/test data
10 test_idx = [0,50,100]
11
12 # training data
13 train_target = np.delete(iris.target, test_idx)
14 train_data = np.delete(iris.data, test_idx, axis=0)
15
16 # testing data
17 test_target = iris.target[test_idx]
18 test_data = iris.data[test_idx]
19
20 ## Tree induction
21 clf = tree.DecisionTreeClassifier()
22 clf.fit(train_data, train_target)
23
24 ## Predict label for new flower
25 print test_target
26 print clf.predict(test_data)
```

goo.gl/KWjsvv

```
1 #!/usr/bin/env python
2
3 ## Import dataset
4 import numpy as np
5 from sklearn import tree
6 from sklearn.datasets import load_iris
7 iris = load_iris()
8
9 ## Split training/test data
10 test_idx = [0,50,100]
11
12 # training data
13 train_target = np.delete(iris.target, test_idx)
14 train_data = np.delete(iris.data, test_idx, axis=0)
15
16 # testing data
17 test_target = iris.target[test_idx]
18 test_data = iris.data[test_idx]
19
20 ## Tree induction
21 clf = tree.DecisionTreeClassifier()
22 clf.fit(train_data, train_target)
23
24 ## Predict label for new flower
25 print test_target
26 print clf.predict(test_data)
```

requires Sklearn!

goo.gl/KWjsvv

```
1 #!/usr/bin/env python
2
3 ## Import dataset
4 import numpy as np
5 from sklearn import tree
6 from sklearn.datasets import load_iris
7 iris = load_iris()
8
9 ## Split training/test data
10 test_idx = [0,50,100]
11
12 # training data
13 train_target = np.delete(iris.target, test_idx)
14 train_data = np.delete(iris.data, test_idx, axis=0)
15
16 # testing data
17 test_target = iris.target[test_idx]
18 test_data = iris.data[test_idx]
19
20 ## Tree induction
21 clf = tree.DecisionTreeClassifier()
22 clf.fit(train_data, train_target)
23
24 ## Predict label for new flower
25 print test_target
26 print clf.predict(test_data)
```

Tweak these values!

goo.gl/KWjsvv

```
python decision_tree_example.py
[0 1 2]
[0 1 2]
```

```
 9 ## Split training/test data
10 test_idx = [0,50,100]
11
12 # training data
13 train_target = np.delete(iris.target, test_idx)
14 train_data = np.delete(iris.data, test_idx, axis=0)
15
16 # testing data
17 test_target = iris.target[test_idx]
18 test_data = iris.data[test_idx]
19
20 ## Tree induction
21 clf = tree.DecisionTreeClassifier()
22 clf.fit(train_data, train_target)
23
24 ## Predict label for new flower
25 print test_target
26 print clf.predict(test_data)
```

goo.gl/KWjsvv

Decision tree demo

Part I:

- **Download** and **run** *“decision_tree_example.py”* from:
  goo.gl/Kwjsvv

Decision tree demo

Part I:

- **Download** and **run** *“decision_tree_example.py”* from:
  goo.gl/Kwjsvv

Part II:

- Go to:
  dreampuf.github.io/GraphvizOnline/
- **Copy** contents of *“iris_classifier.txt”*
- **Paste** into text field into GraphvizOnline
- **Observe** figure and **ask** yourself:

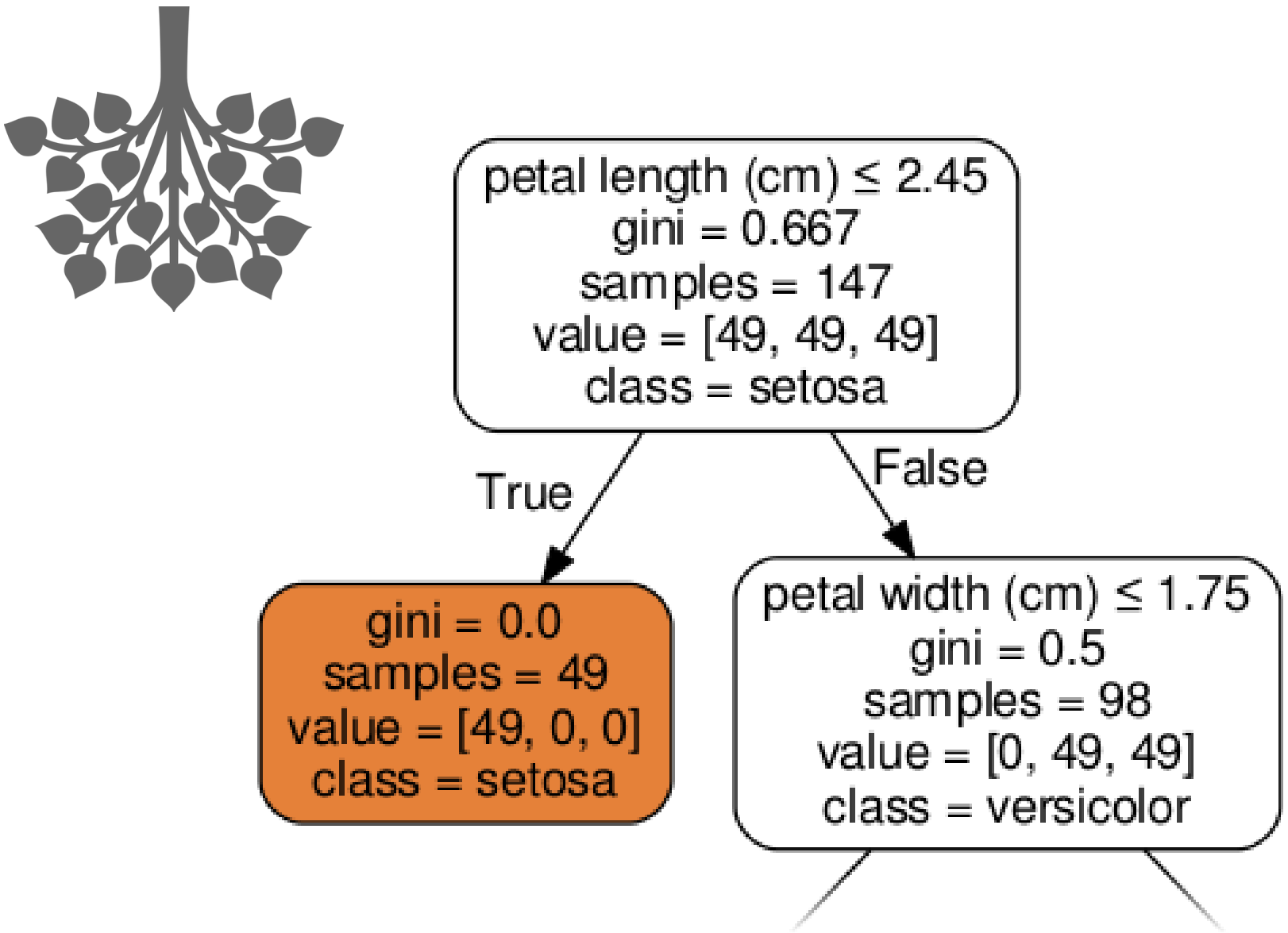
“Which features are most helpful in making predictions about my target variable?”

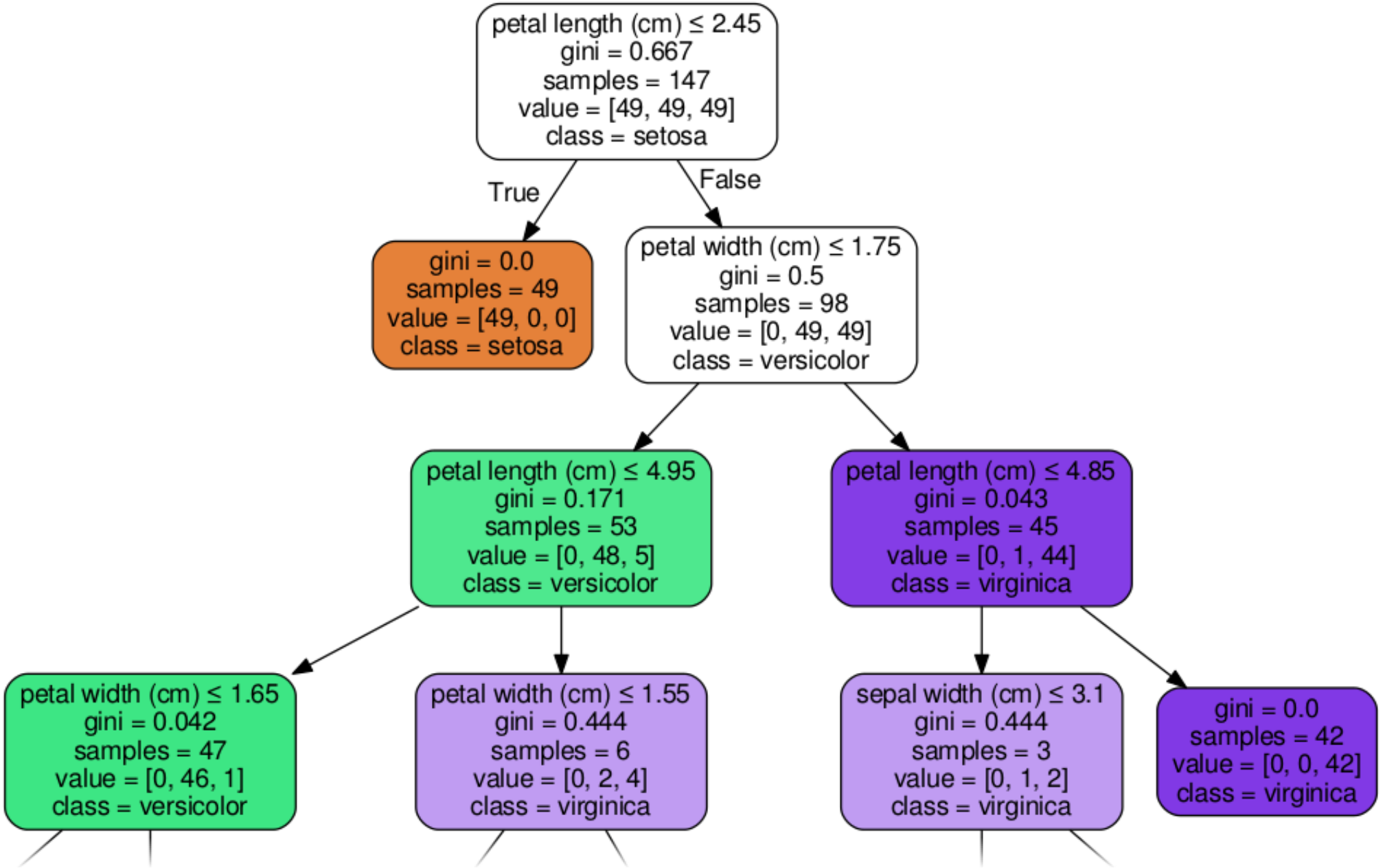
Debrief

Visualizing trees

- Helps us ask:

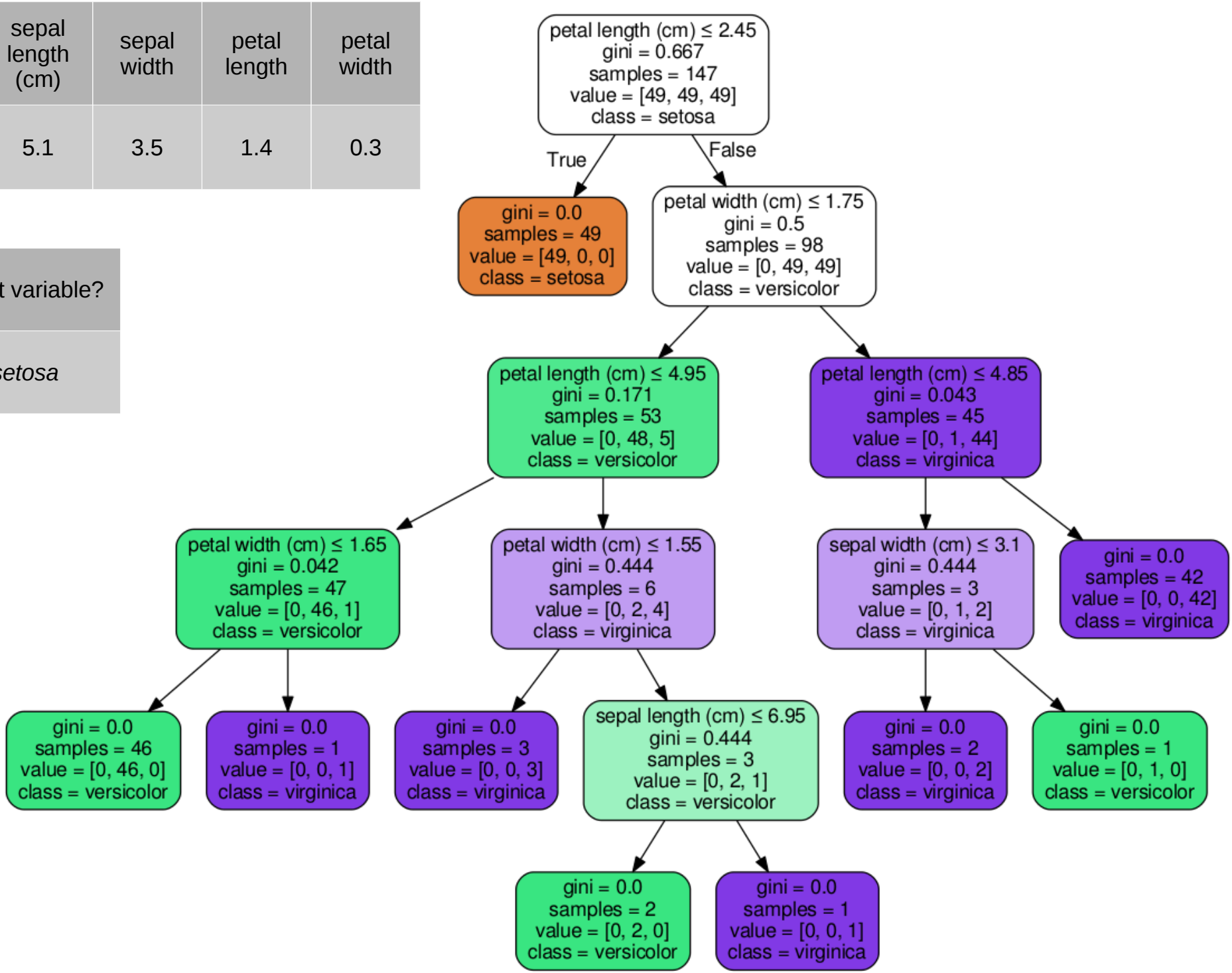
  “Which of my *features* are the *most helpful* in making predictions about my target variable?”


petal length (cm) ≤ 2.45
gini = 0.667
samples = 147
value = [49, 49, 49]
class = setosa
True
False
gini = 0.0
samples = 49
value = [49, 0, 0]
class = setosa
petal width (cm) ≤ 1.75
gini = 0.5
samples = 98
value = [0, 49, 49]
class = versicolor


petal length (cm) ≤ 2.45
gini = 0.667
samples = 147
value = [49, 49, 49]
class = setosa
True
False
gini = 0.0
samples = 49
value = [49, 0, 0]
class = setosa
petal width (cm) ≤ 1.75
gini = 0.5
samples = 98
value = [0, 49, 49]
class = versicolor
petal length (cm) ≤ 4.95
gini = 0.171
samples = 53
value = [0, 48, 5]
class = versicolor
petal length (cm) ≤ 4.85
gini = 0.043
samples = 45
value = [0, 1, 44]
class = virginica
petal width (cm) ≤ 1.65
gini = 0.042
samples = 47
value = [0, 46, 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 = 42
value = [0, 0, 42]
class = virginica

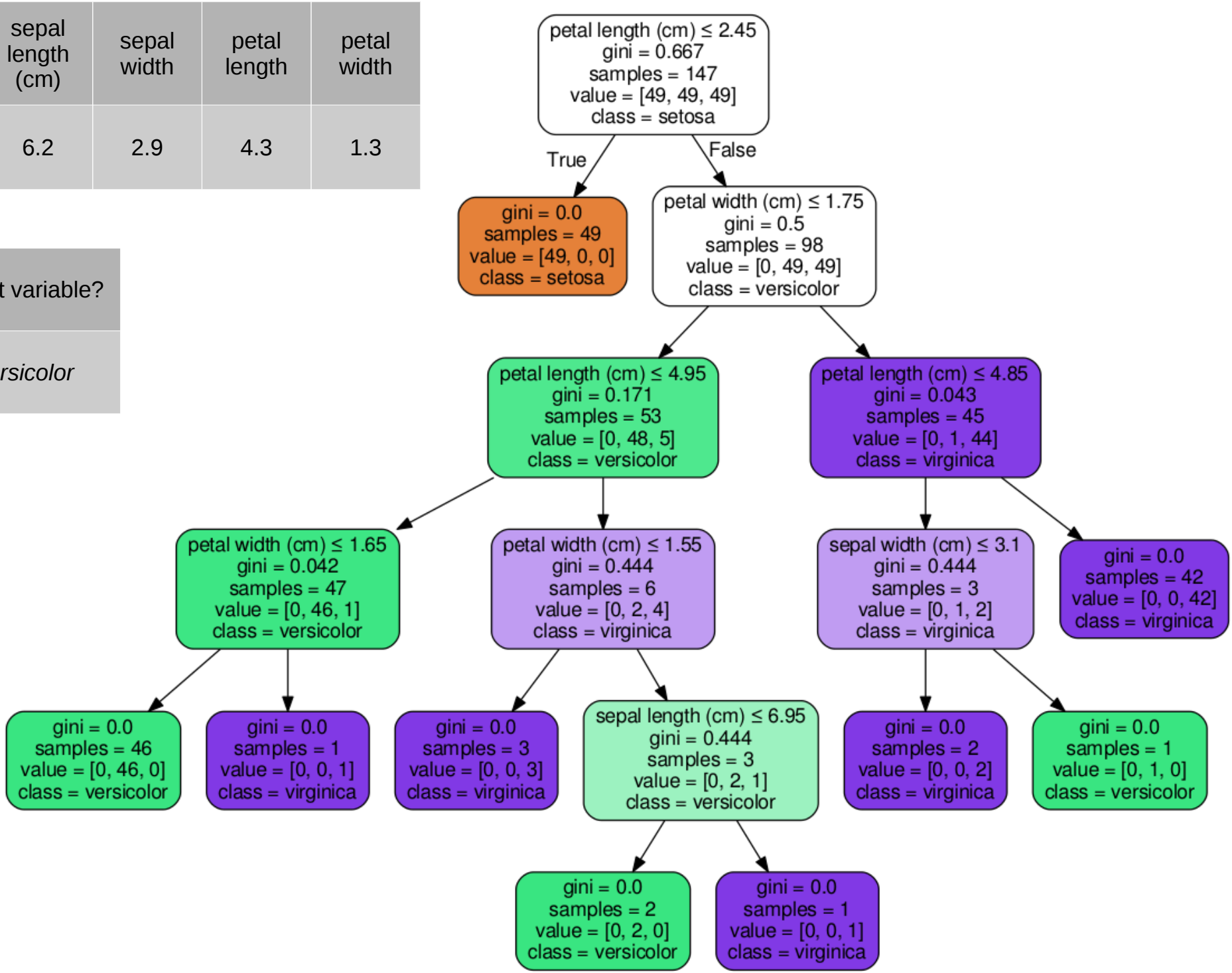
sample #	sepal length (cm)	sepal width	petal length	petal width
17	5.1	3.5	1.4	0.3

target variable?
setosa



sample #	sepal length (cm)	sepal width	petal length	petal width
97	6.2	2.9	4.3	1.3

target variable?
versicolor



Debrief: Hazards and pitfalls

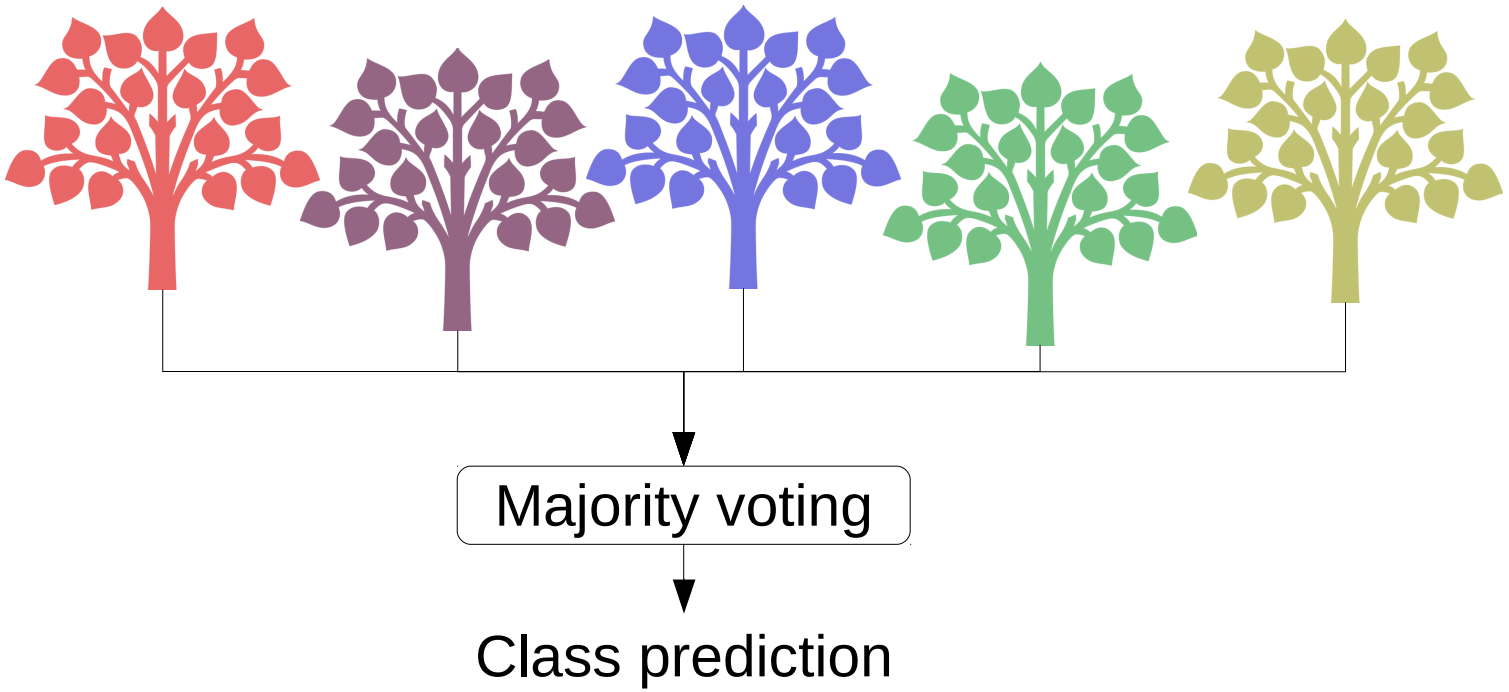
- Decision trees are *efficient*, but not *powerful*

Debrief: Hazards and pitfalls

- Decision trees are *efficient*, but not *powerful*
- As always, overfitting is a concern
  - Algorithm will always find some solution

Debrief: Next steps

- Random forests



Lesson 9: Decision trees

- Decision trees intro (I do)
- Independent exercise (We do)
  - Tree induction with *python* and *scikit-learn*
- Debrief (You do)
  - Visualizing decision trees
  - When to use decision trees
    - Pitfalls and things to avoid
  - Next steps