Decision Tree, Neural Network

อ. ปรัชญ์ ปิยะวงศ์วิศาล

Pratch Piyawongwisal

Today

- TPQI
- Decision Tree
- Neural Network

Recap: Supervised Learning

Decision Tree

Neural Net

kNN

Logistic Regression

SVM

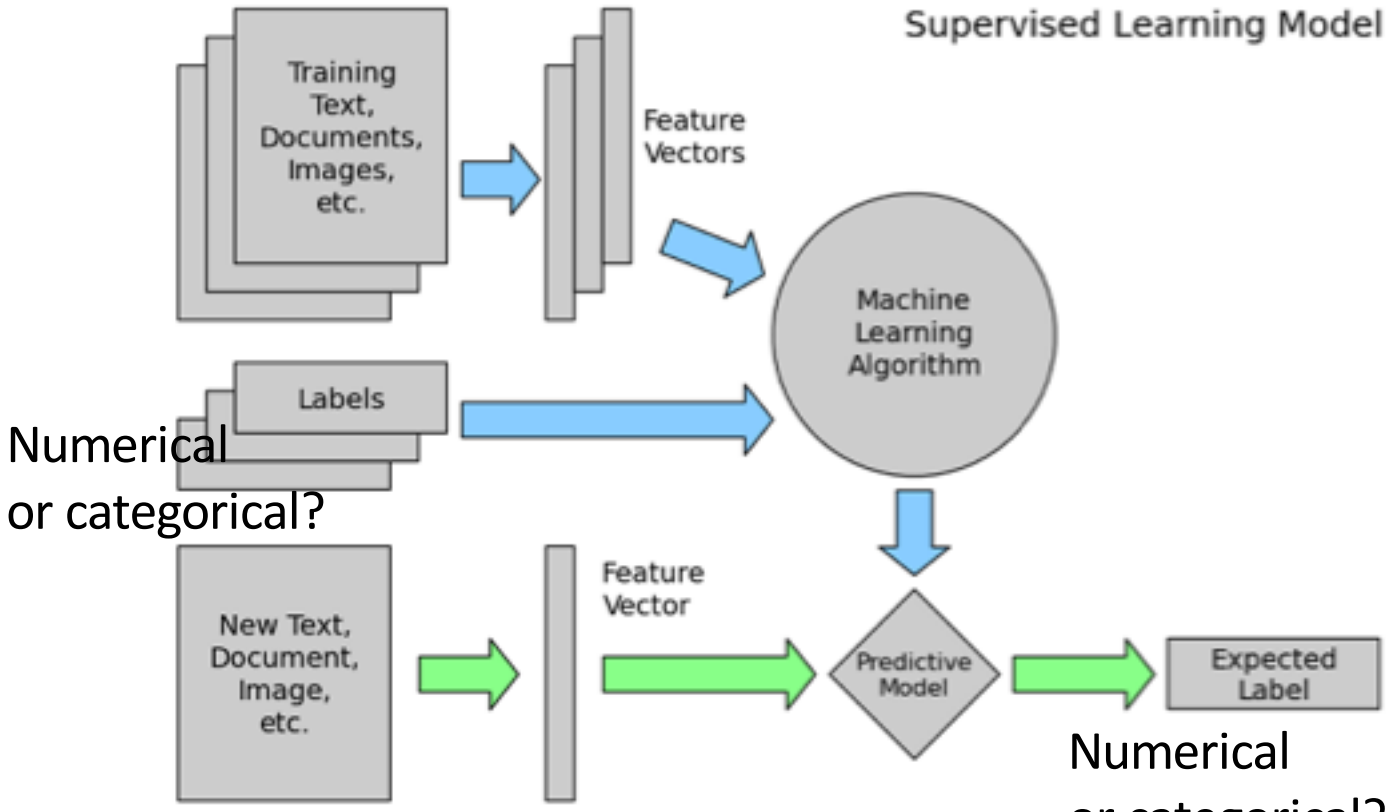
Classification

- Predicts class labels/categories
- ทำนายค่าที่เป็นหมวดหมู่ = จำแนกประเภท
- อาจมองเป็นการหา **boundary** ที่แบ่งข้อมูลในแต่ละหมวดหมู่ ออกจากกัน

Regression

- Predicts continuous values
- ทำนายค่าที่เป็นจำนวนจริง
- อาจมองเป็นการหา **hyperplane** ที่ **fit** กับข้อมูลที่มีมากที่สุด

Linear Regression



Decision Tree

Decision Tree

- เป็น supervised learning ใช้ทำ classification เป็นหลัก
- ข้อดี
  - เข้าใจง่าย
  - สามารถตีความและอธิบายที่มาที่ไปได้ง่าย (white box model) -- ต่างจาก black box model อย่าง Neural Network
  - ใช้กับข้อมูลที่เป็น numerical หรือ categorical ก็ได้
  - สามารถประมาณความน่าจะเป็นที่จะเป็นแต่ละคลาส (class probability) ได้
- ข้อเสีย
  - decision tree เป็นปัญหา NP-complete ไม่สามารถหาคำตอบที่ดีที่สุดใน poly time ได้ จึงต้องใช้อัลกอริทึมแบบ heuristic เช่น greedy ในการหาคำตอบแบบ local optima

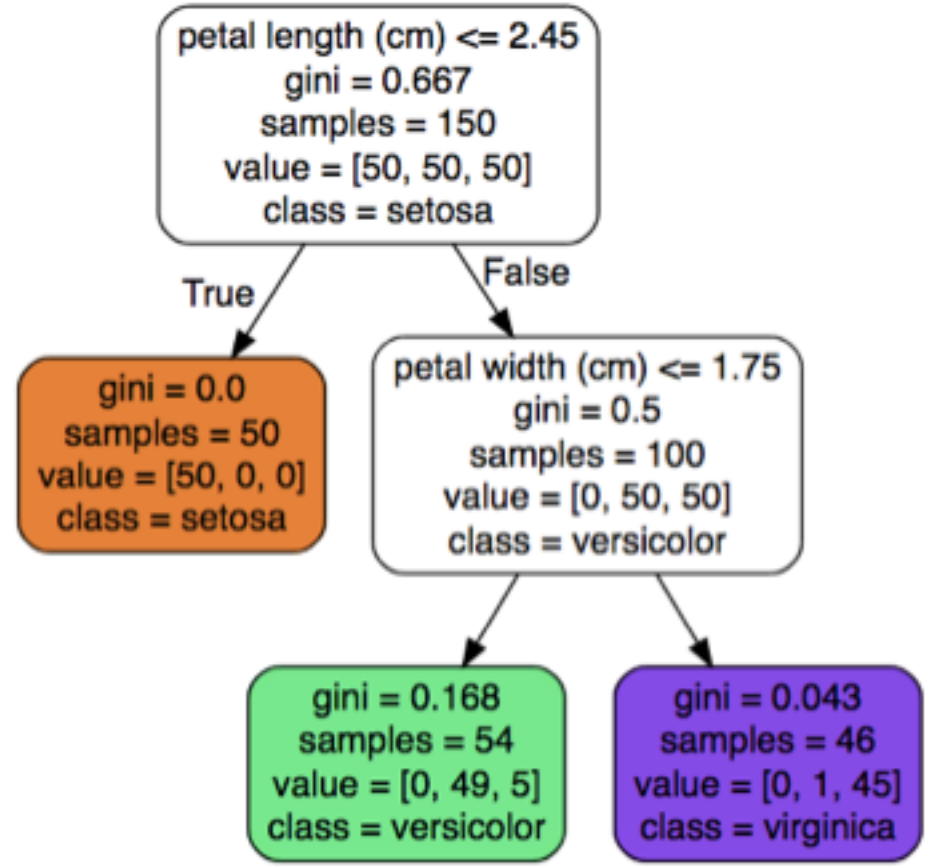
ลอง Implement Decision Tree

```
from sklearn.datasets import load_iris
from sklearn.tree import DecisionTreeClassifier
import numpy as np

iris = load_iris()
X = iris.data[:, 2:]        # feature: petal length/width
y = iris.target             # label: 0, 1, 2
np.random.seed(42)
tree_clf = DecisionTreeClassifier(max_depth=2)
tree_clf.fit(X, y)
```

Visualizing the Decision Tree

```
from sklearn.tree import export_graphviz

tree_dot = export_graphviz(
    tree_clf,
    out_file=None,   # or out_file="iris_tree.dot"
    feature_names=iris.feature_names[2:],
    class_names=iris.target_names,
    rounded=True,
    filled=True
)
print(tree_dot)
```

จากนั้นนำผลลัพธ์ไปแปลงเป็นแผนภาพได้ที่ http://www.webgraphviz.com/



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

Alternative: using python-graphviz

Once trained, we can export the tree in Graphviz format using the `export_graphviz` exporter. If you use the conda package manager, the graphviz binaries and the python package can be installed with

```
conda install python-graphviz
```

Alternatively binaries for graphviz can be downloaded from the graphviz project homepage, and the Python wrapper installed from pypi with pip install graphviz.

Below is an example graphviz export of the above tree trained on the entire iris dataset; the results are saved in an output file iris.pdf:

```
>>> import graphviz
>>> dot_data = tree.export_graphviz(clf, out_file=None)
>>> graph = graphviz.Source(dot_data)
>>> graph.render("iris")
```

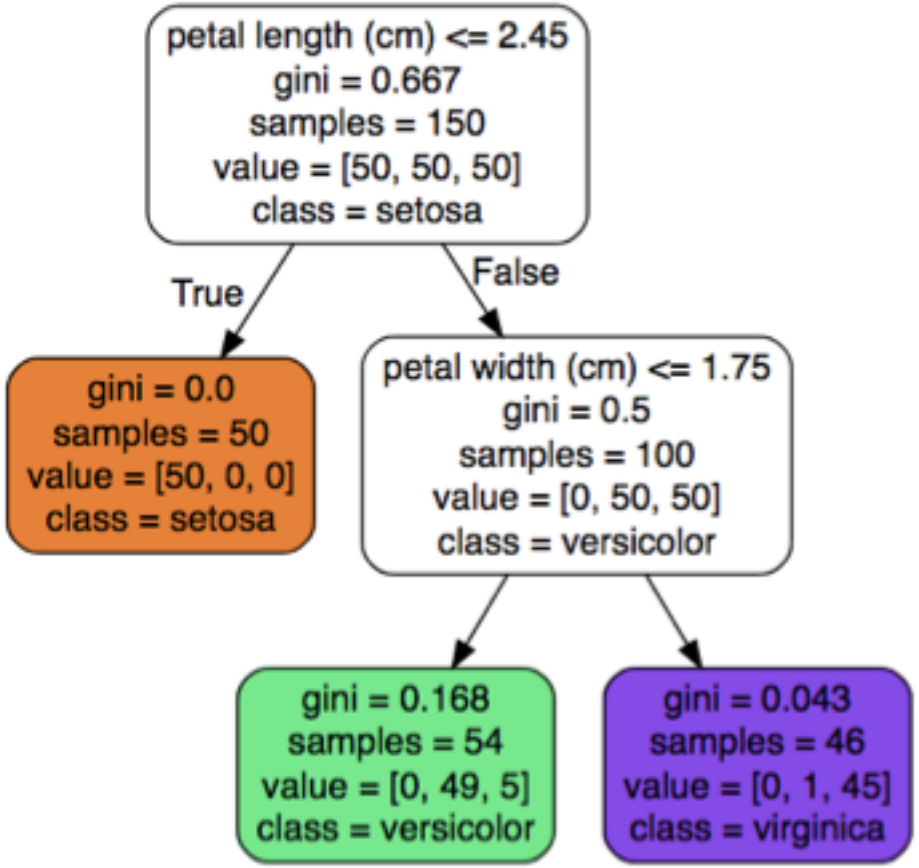
The `export_graphviz` exporter also supports a variety of aesthetic options, including coloring nodes by their class (or value for regression) and using explicit variable and class names if desired. Jupyter notebooks also render these plots inline automatically:

```
>>> dot_data = tree.export_graphviz(clf, out_file=None,
                                    feature_names=iris.feature_names,
                                    class_names=iris.target_names,
                                    filled=True, rounded=True,
                                    special_characters=True)
>>> graph = graphviz.Source(dot_data)
>>> graph
```

ตีความภาพ decision tree

- เปรียบเสมือน flowchart ของกฎที่อยู่ในรูป if-else ดังนี้

```
if petal_length <= 2.45:
      class = "setosa"
elif petal <= 1.75:
      class = "versicolor"
else:
      class = "virginica"
```



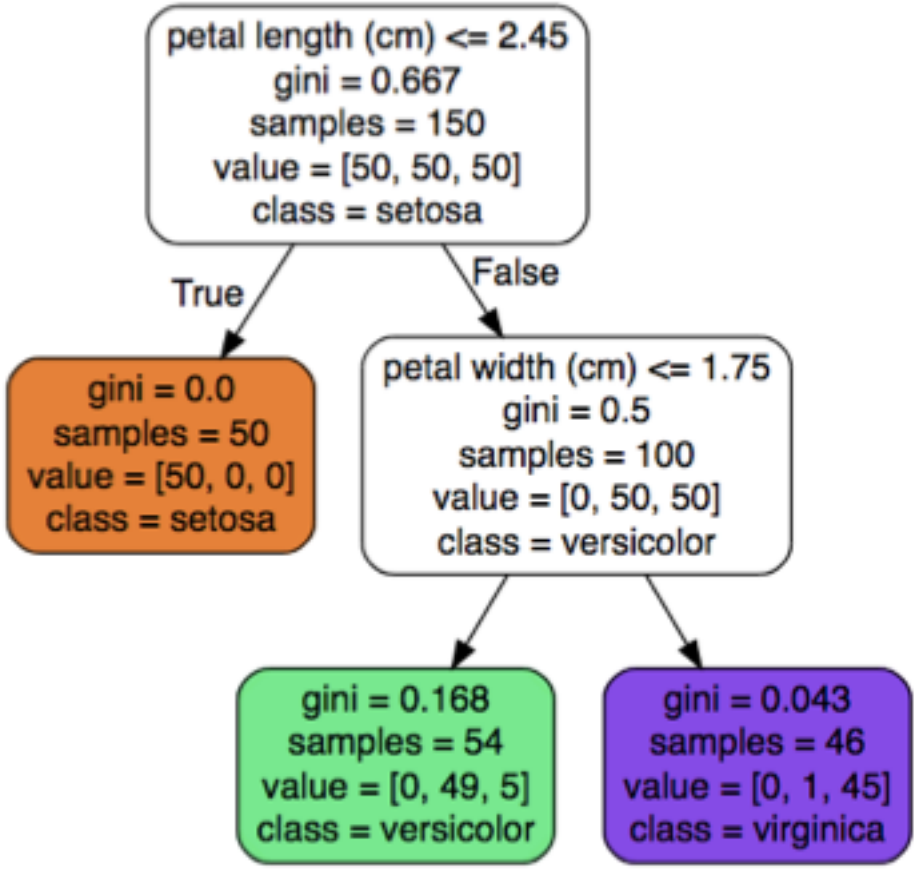
ตีความภาพ decision tree

- node สีขาว คือ ตัวแบ่ง (splitting attribute)
- ค่า gini คือค่าความปะปน (Impurity) ของ node
  - gini = 0.0 แสดงว่าข้อมูล train ใน node นั้นเป็นคลาสเดียวกันหมด
  - gini มาก แสดงว่า node มีข้อมูลหลายคลาสปะปนกัน

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

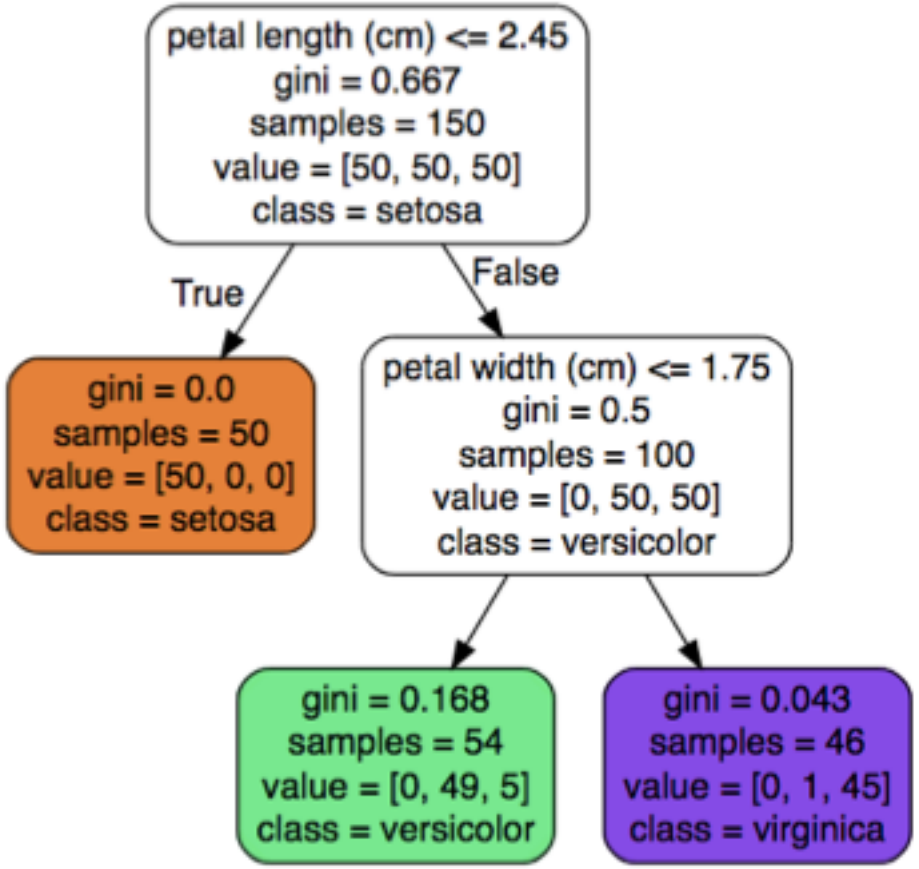
$p_{i,k}$ คืออัตราส่วนของข้อมูลคลาส k ต่อข้อมูลทั้งหมด ณ node ที่ I

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



การจำแนกด้วย decision tree

- ตัวอย่าง: จงใช้ decision tree นี้ classify ดอกกล้วยไม้ที่มีความยาวกลีบ 5cm และมีความกว้างกลีบ 1.5cm

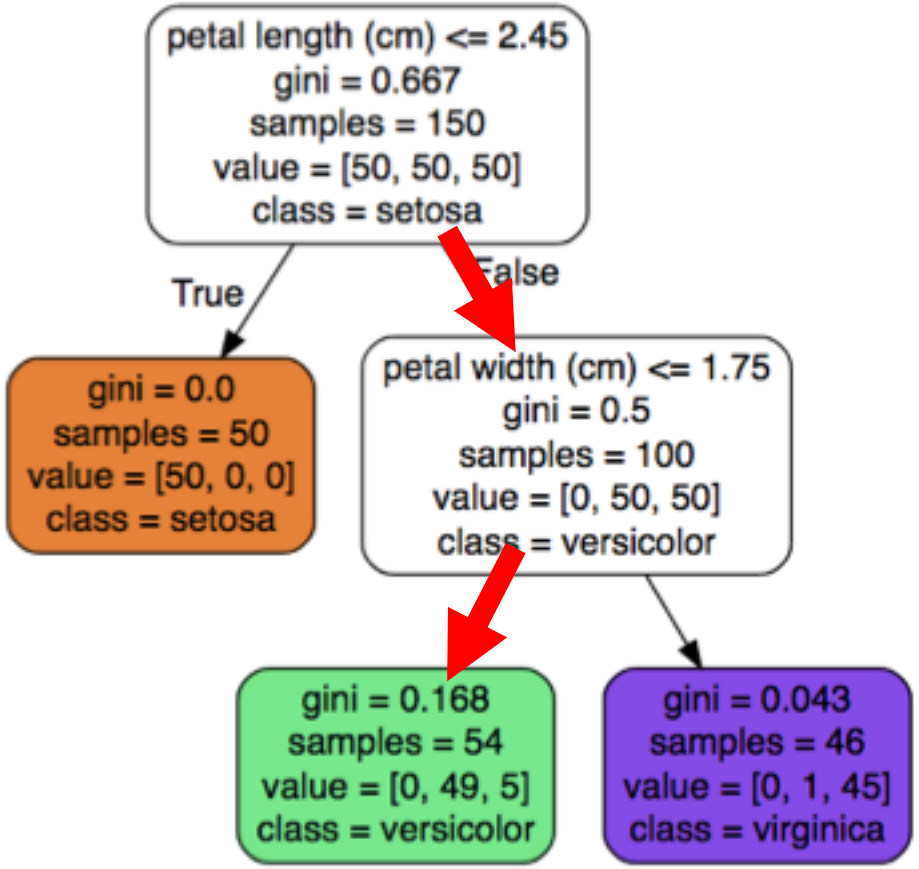


การจำแนกด้วย decision tree

- ตัวอย่าง: จงใช้ decision tree นี้ classify ดอกกล้วยไม้ที่มีความยาวกลีบ 5cm และมีความกว้างกลีบ 1.5cm

- คำตอบ: predict class="versicolor"

- Q: จงหา class probability ต่อไปนี้
  - ความน่าจะเป็นที่ดอกนี้จะเป็น setosa?
  - ความน่าจะเป็นที่ดอกนี้จะเป็น virginica?



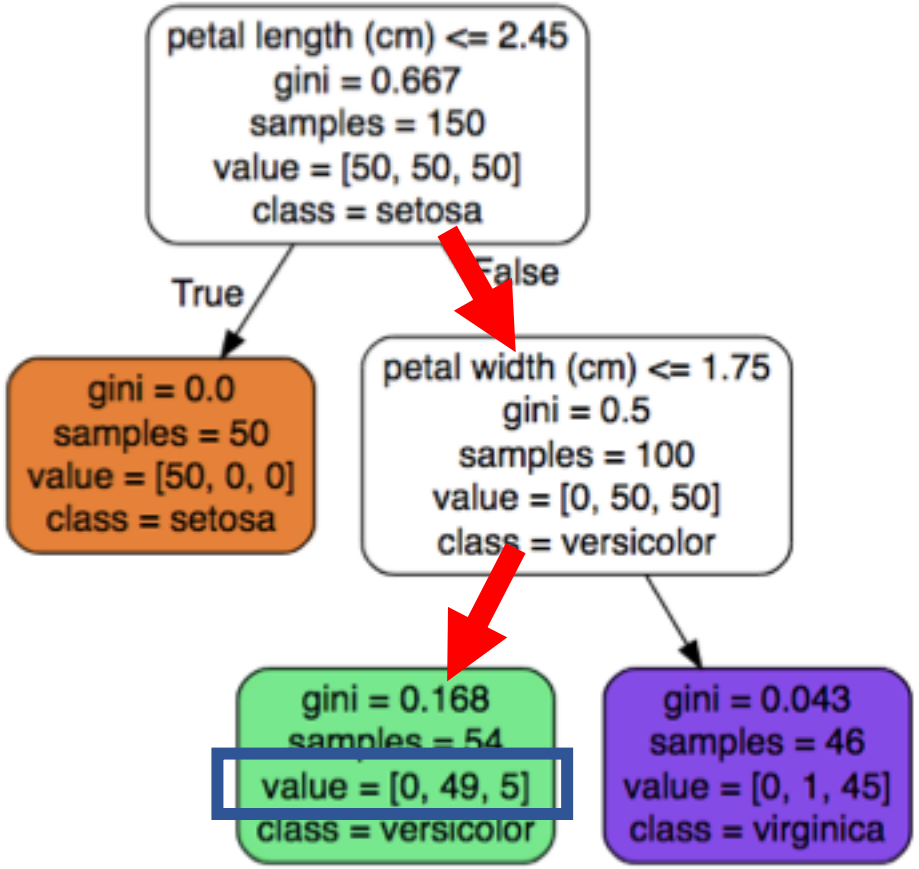
การจำแนกด้วย decision tree

- ตัวอย่าง: จงใช้ decision tree นี้ classify ดอกกล้วยไม้ที่มีความยากลีบ 5cm และมีความกว้างกลีบ 1.5cm

- คำตอบ: predict class="versicolor"

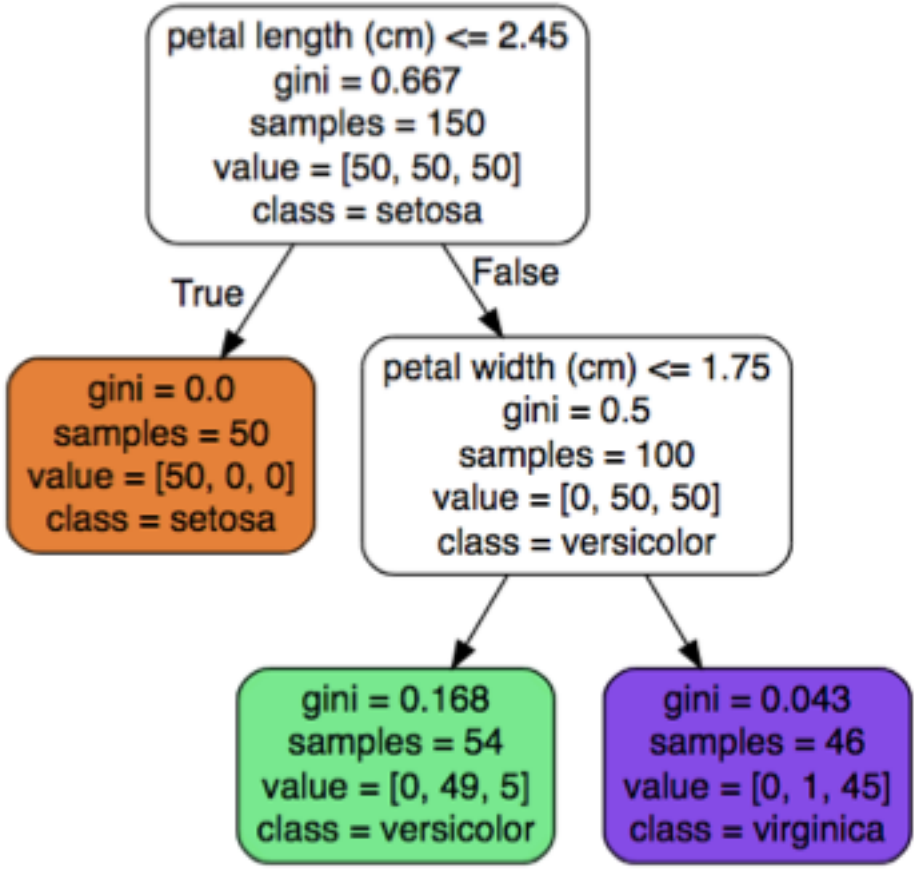
- Q: จงหา class probability ต่อไปนี้
  - ความน่าจะเป็นที่ดอกนี้จะเป็น setosa?
  - P(setosa) = 0/54 = 0%

  - ความน่าจะเป็นที่ดอกนี้จะเป็น virginica?
  - P(virginica) = 5/54 = 9.3%



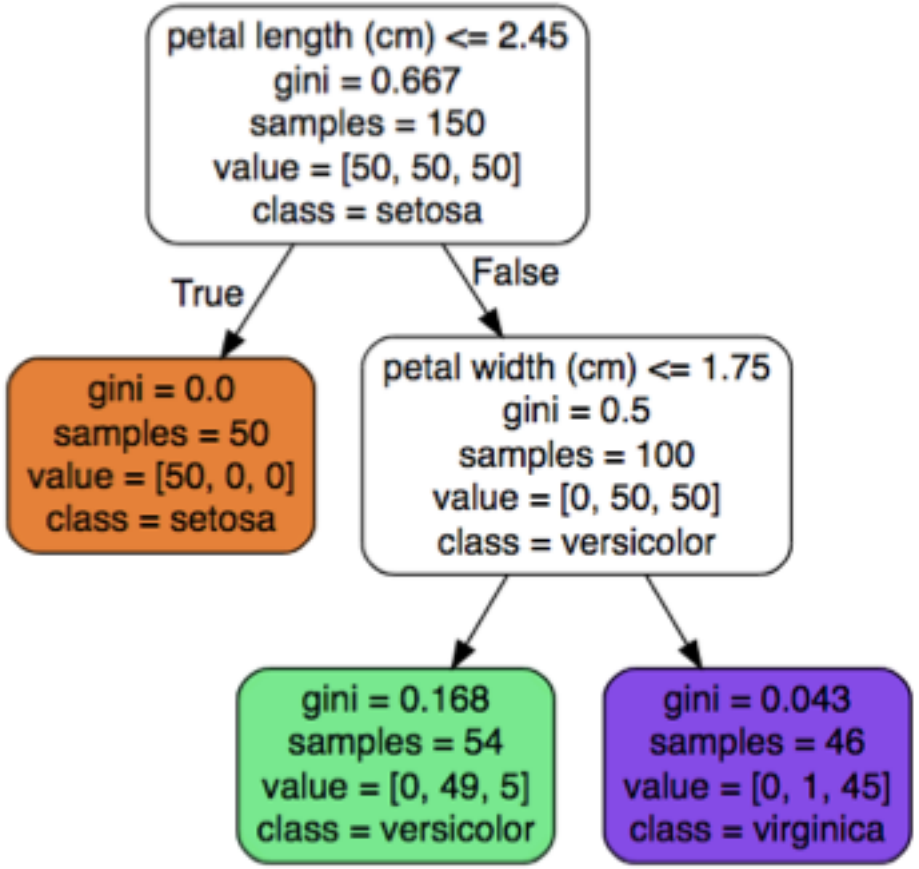
CART Algorithm สำหรับสร้าง decision tree

- การสร้าง **decision tree** จะเริ่มจาก **root node**
- ในแต่ละ **level** เราจะเลือก **(k, tk)** ที่ดีที่สุดมาเป็นตัว **split**
  - attribute k
  - threshold tk
  - กฎ: k <= tk
- จะเลือก **(k, tk)** ที่ดีที่สุดได้อย่างไร?



CART Algorithm สำหรับสร้าง decision tree

- จะเลือก (k, tk) ที่ดีที่สุดได้อย่างไร?
- เป้าหมาย: ต้องการ (k, tk) ที่ทำให้ subset ข้อมูลหลังการ split บริสุทธิ์ (pure) ที่สุด
  - = Gini ต่ำที่สุด
- ดังนั้น Cost function ที่เราต้องการ minimize คือ:
  - $J(k, t_k) = \frac{m_{left}}{m} G_{left} + \frac{m_{right}}{m} G_{right}$
- ในแต่ละชั้นต้นไม้ อัลกอริทึม CART จะหา (k, tk) ที่ทำให้ J ต่ำสุด

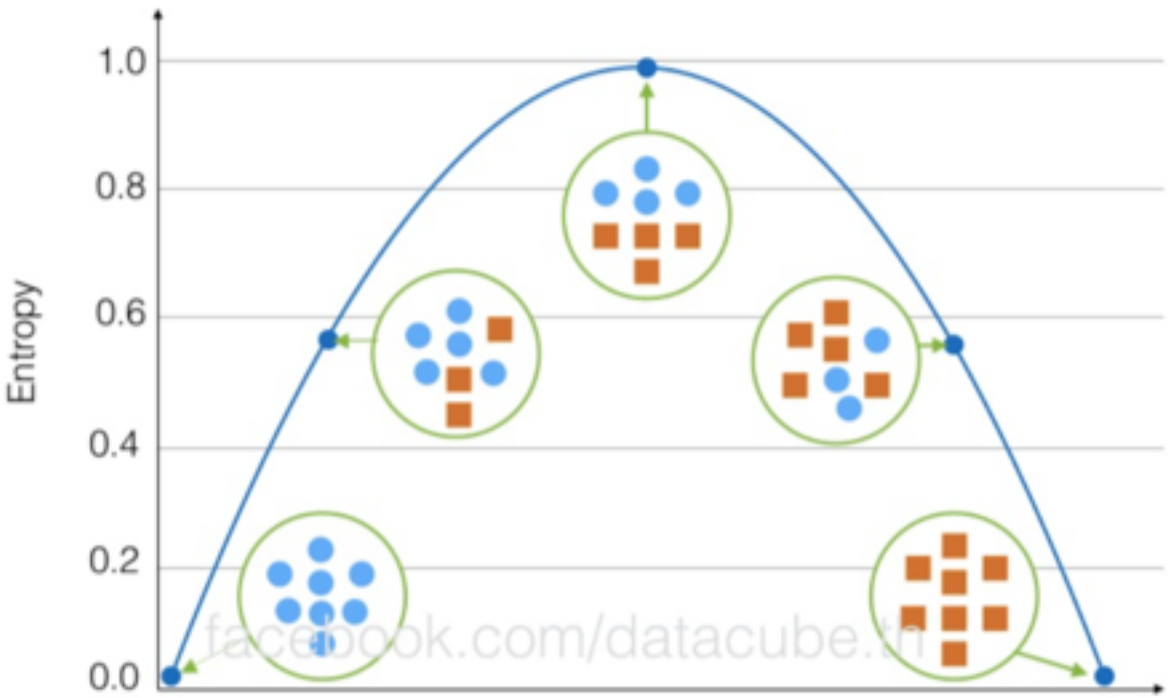


*ปัญหาการหา optimal tree เป็น NP-complete ไม่สามารถหาใน poly time ได้
จึงต้องใช้วิธีแบบ greedy คือหา (k, tk) ที่ทำให้ cost น้อยสุด ทีละชั้น แม้ว่าลงไปชั้นล่างๆ แล้วอาจทำให้ cost กลับมาสูงขึ้นได้ก็ตาม

Choosing Splitting Criteria

- นอกจาก **Gini Impurity** แล้ว ยังสามารถใช้ **Entropy/Information Gain** เป็นเกณฑ์ในการวัดประสิทธิภาพของ **splitting attribute** ได้ด้วย
  - ID3 algorithm

- **splitting criterion** ถือเป็น **hyperparameter** หนึ่งของ **decision tree**



$$H(T) = I_E(p_1, p_2, \ldots, p_J) = -\sum_{i=1}^{J} p_i \log_2 p_i$$

$$\overbrace{IG(T,a)}^{\text{Information Gain}} = \overbrace{H(T)}^{\text{Entropy(parent)}} - \overbrace{H(T|a)}^{\text{Weighted Sum of Entropy(Children)}}$$

Neural Networks

Midterm & TPQI

- AI
  - Strong vs Weak
  - Turing Test
- Machine Learning
  - supervised vs unsupervised
  - classification vs regression
  - model, train, test, feature, class, cross-validation, overfitting
- Supervised Learning
  - kNN (ข้อเสียคืออะไร, ค่า k ส่งผลกับ boundary อย่างไร)
  - Linear Regression (วิธี gradient descent ดีกว่า normal eq อย่างไร, regularize ทำเพื่ออะไร)
    - prediction model, MSE cost function, normal equation, gradient descent, regularization
  - Logistic Regression (ใช้ทำอะไรได้, นึกถึง iris)
    - prediction model, sigmoid cost function
  - SVM (ใช้ทำอะไรได้, ดีอย่างไร, kernel มีประโยชน์อย่างไร)
    - prediction model, hard vs soft margin, kernel