Aula06 Árvores de Decisão

Aprendizado de Máquina Supervisionado - Problemas de Classificação
Inteligência Artificial

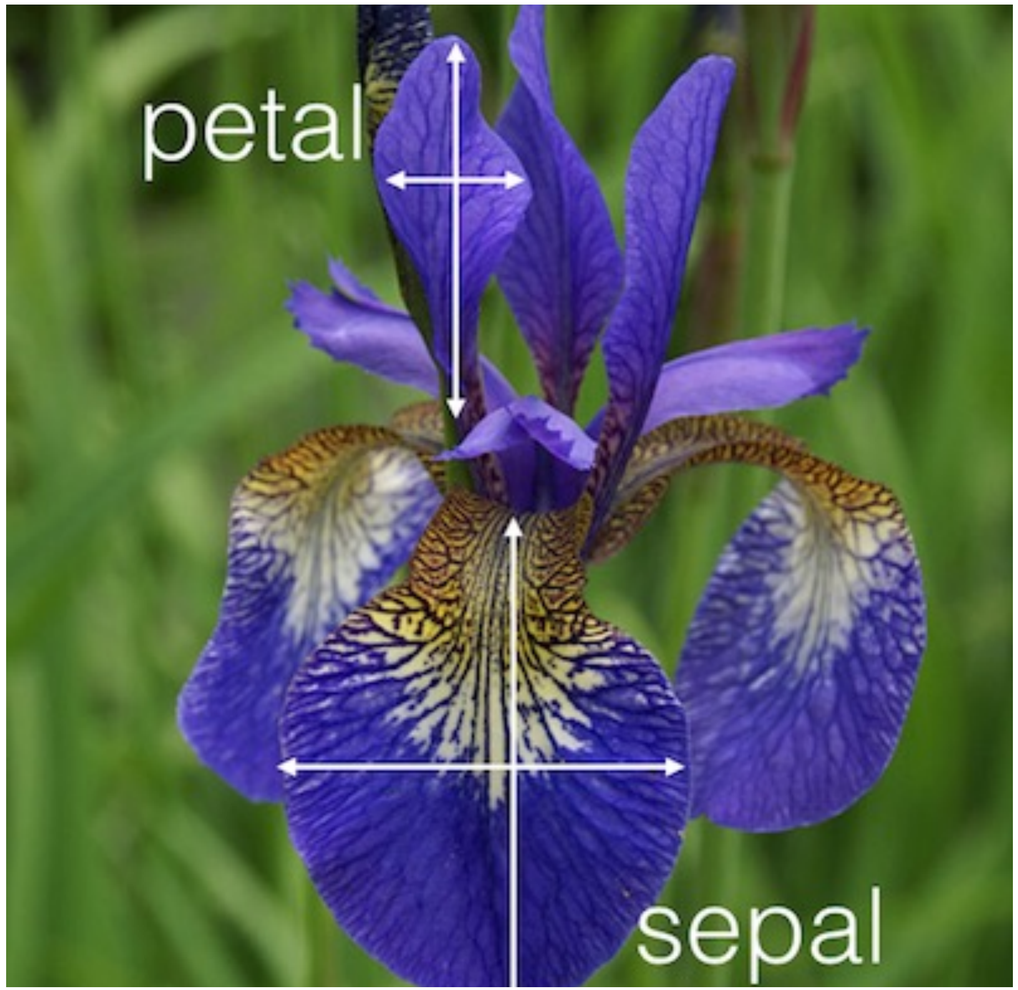
Prof. Sergio Bonato

Introdução

- Árvores de Decisão são algoritmos muito poderosos capazes de moldar conjuntos completos de dados.
- São bastante versáteis, podendo executar tarefas de classificação, regressão e até é mesmo classificação com múltiplas saídas/respostas.
- Iremos ver nesta disciplina o uso de árvores somente para classificação.
- A implementação que iremos estudar é a do Scikit-Learn que utiliza o algoritmo de treinamento CART.

Iris Dataset

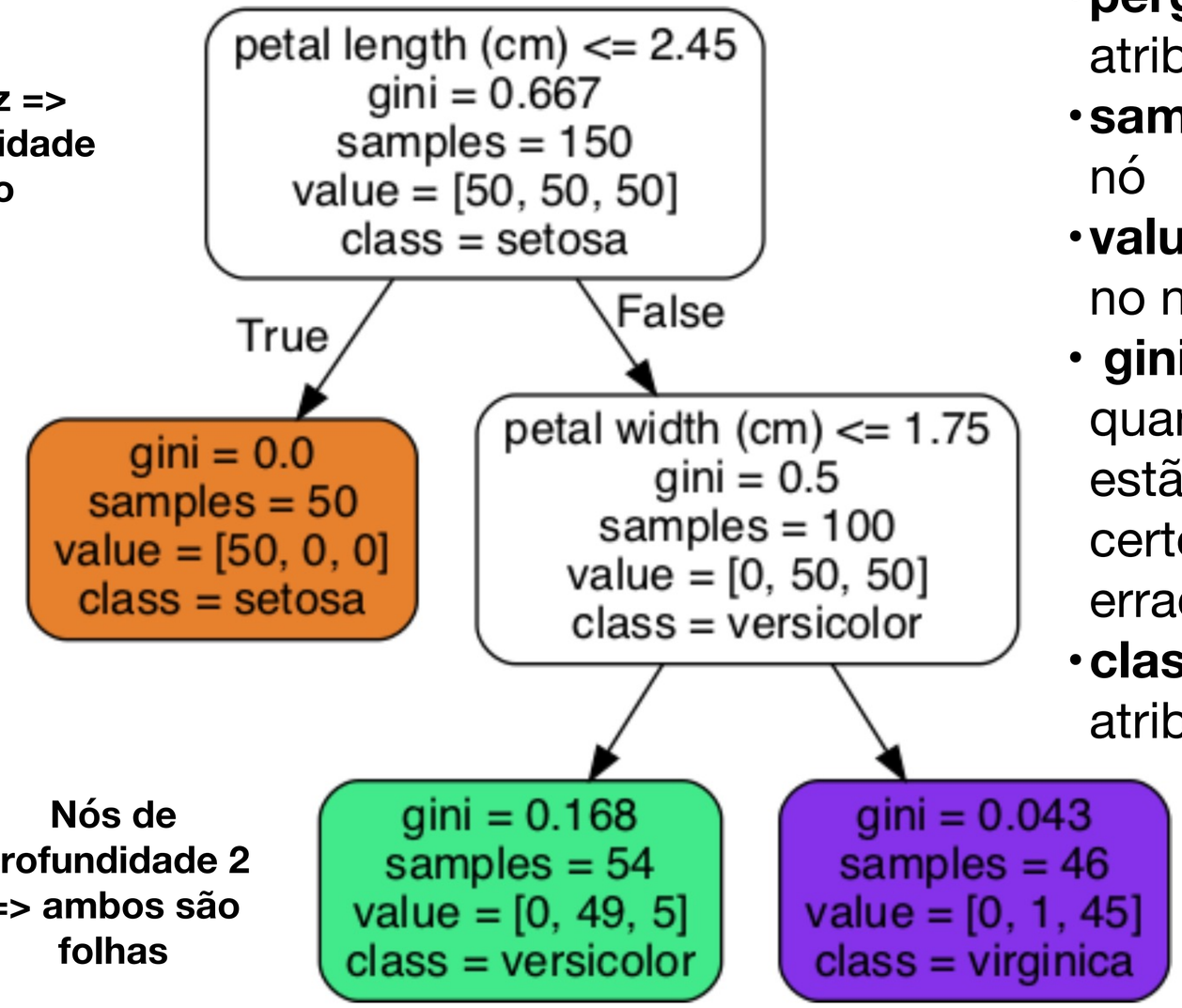
- 150 flores de iris de três espécies diferentes.
- As três classes do dataset são:
    1. Iris-setosa (*n*=50)
    2. Iris-versicolor (*n*=50)
    3. Iris-virginica (*n*=50)
- Os quatro atributos do dataset são:
    1. sepal length em cm
    2. sepal width em cm
    3. petal length em cm
    4. petal width em cm



http://sebastianraschka.com/Articles/2014_python_lda.html

Árvore de Decisão do Iris Dataset

Nó raiz => profundidade zero



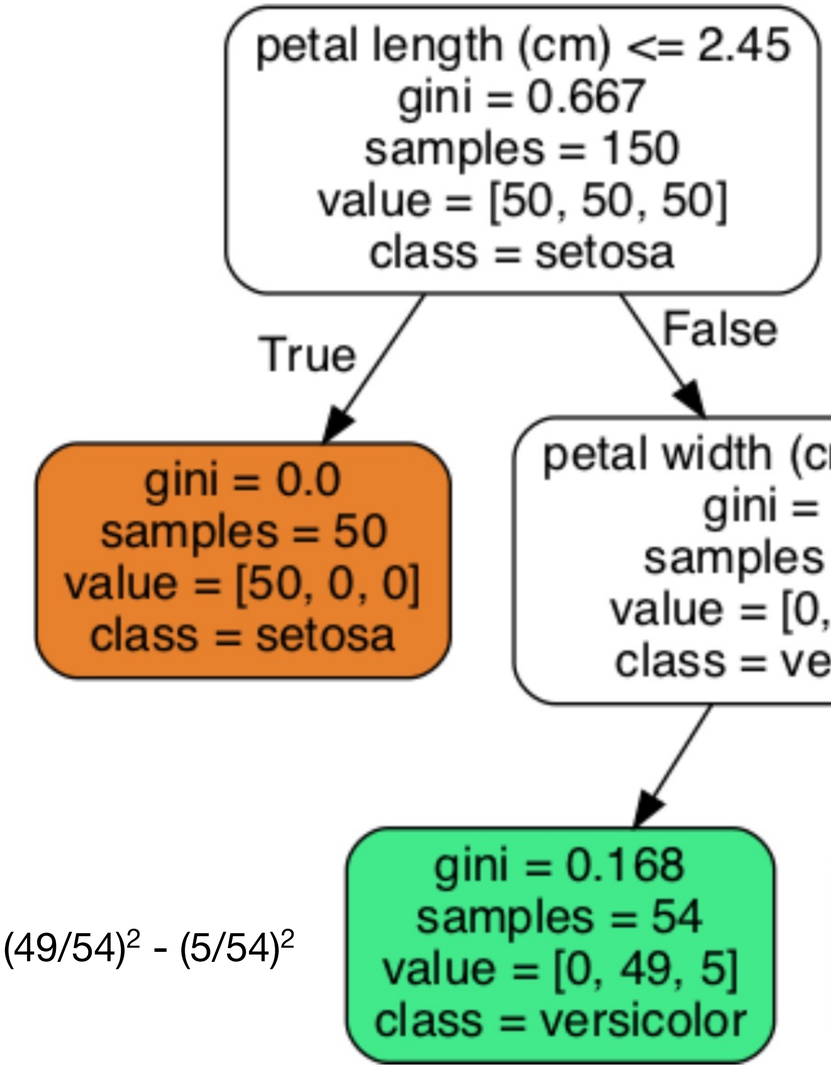
Nós de profundidade 1 => o laranja é folha

Nós de profundidade 2 => ambos são folhas

- **pergunta** leva em conta o melhor atributo com o menor gini
- **samples** é o número de amostras do nó
- **value** é a distribuição de cada classe no nó: **[setosa, versicolor, virginica]**
- **gini** é o índice de pureza, isto é, quantas amostras da classe correta estão no nó; 0 é pureza total (tudo certo) e 1 é impureza total (todas erradas).
- **class** é a classe que o algoritmo iria atribuir à amostra se parasse neste nó

Gini

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



gini = 1 - (50/150)^2 - (50/150)^2 - (50/150)^2

gini = 1 - (50/50)^2 - (0/50)^2 - (0/50)^2

gini = 1 - (0/100)^2 - (50/100)^2 - (50/100)^2

gini = 1 - (0/54)^2 - (49/54)^2 - (5/54)^2

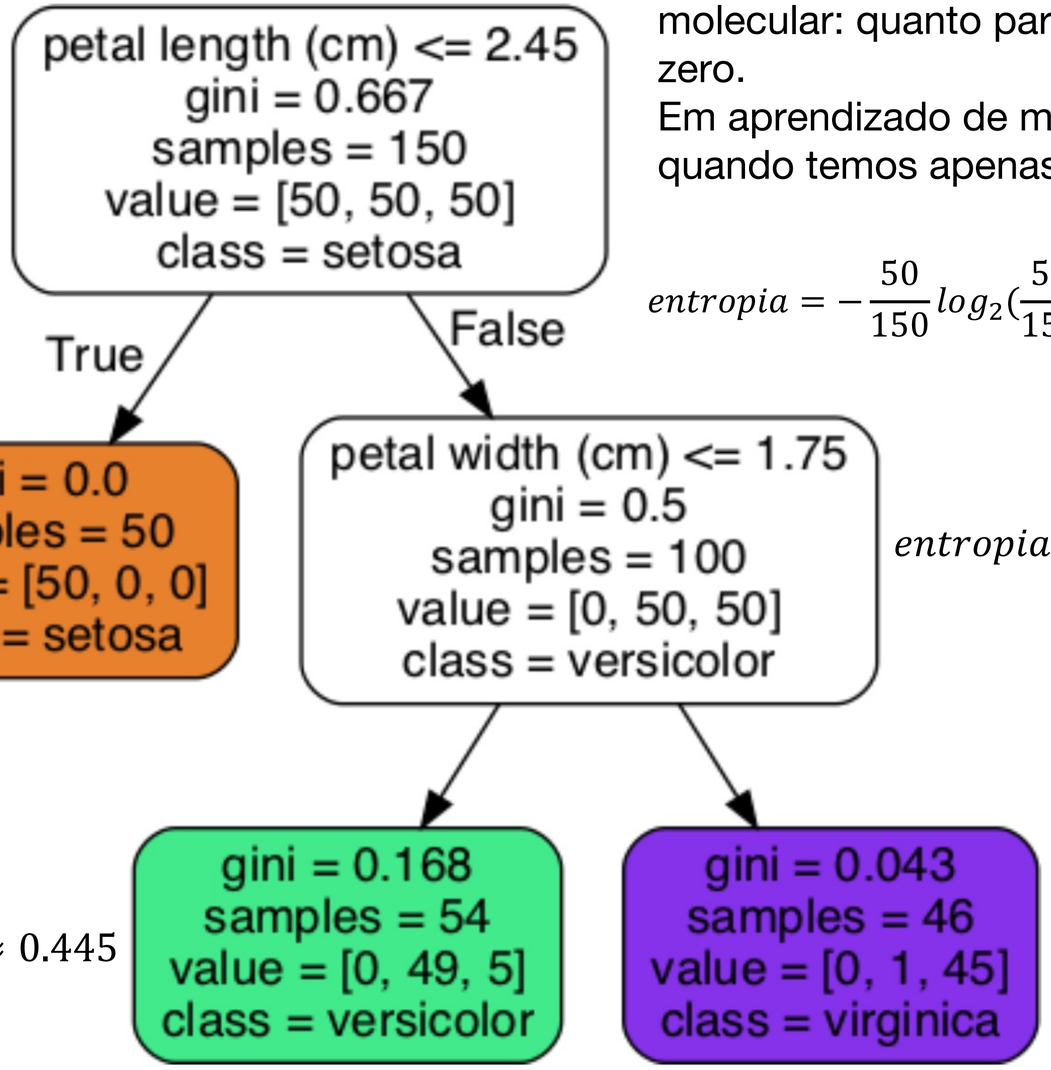
gini = 1 - (0/46)^2 - (1/46)^2 - (45/46)^2

Entropia

$$H_i = \sum_{k=1}^{n} P_{i,k}\, log_2(P_{i,k})$$

$$P_{i,k} \neq 0$$

Entropia vem da termodinâmica e mede a desordem molecular: quanto paradas e bem ordenadas ela tende a zero.
Em aprendizado de máquina, a entropia é igual a zero quando temos apenas uma classe no nó.



$$entropia = -\frac{50}{150} log_2(\frac{50}{150}) - \frac{50}{150} log_2(\frac{50}{150}) - \frac{50}{150} log_2(\frac{50}{150}) \approx 1.585$$

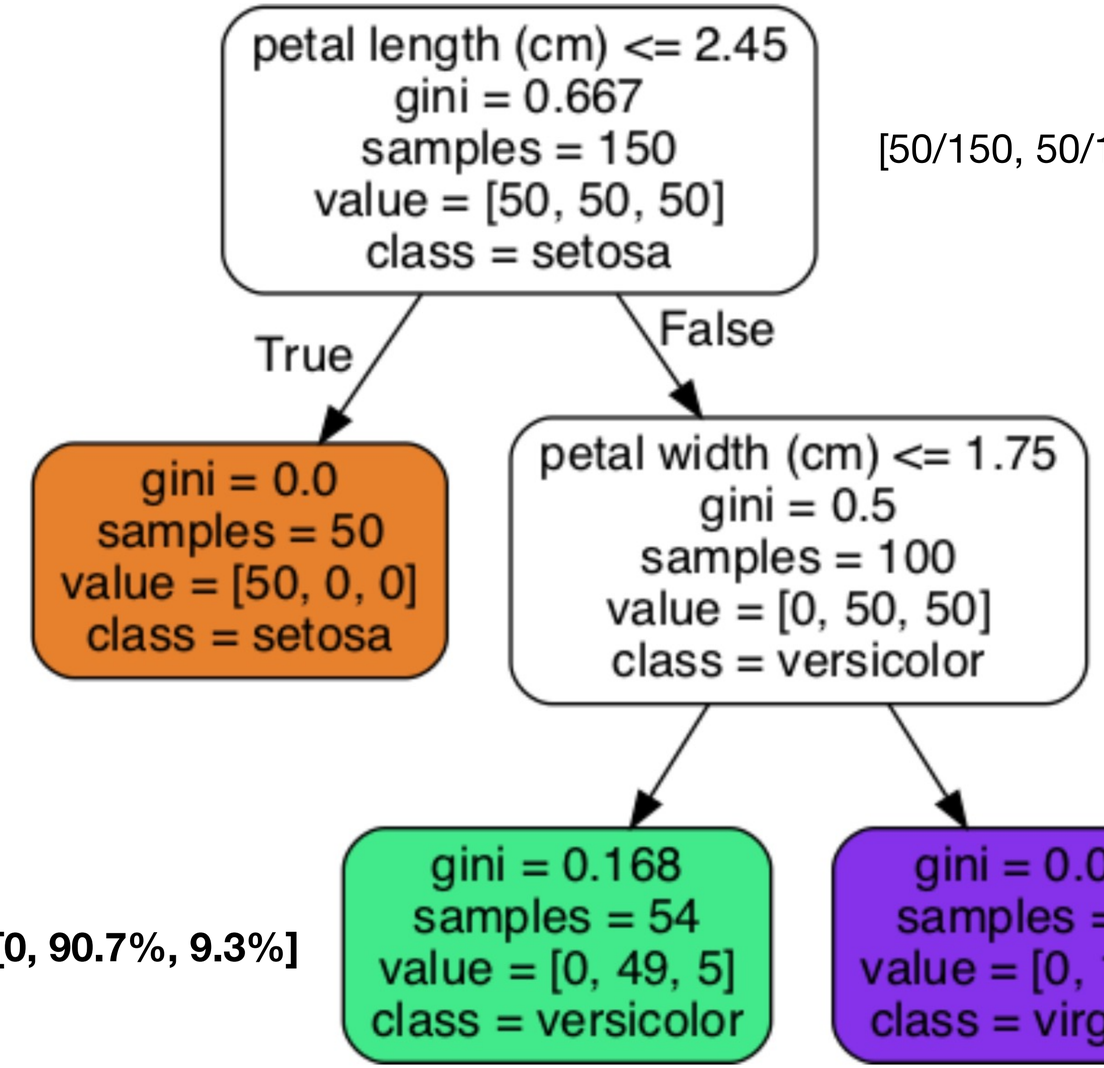
$$entropia = -\frac{50}{50} log_2(\frac{50}{50}) = 0$$

$$entropia = -\frac{50}{100} log_2(\frac{50}{100}) - \frac{50}{100} log_2(\frac{50}{100}) = 1.5$$

$$entropia = -\frac{49}{54} log_2(\frac{49}{54}) - \frac{5}{54} log_2(\frac{5}{54}) \approx 0.445$$

$$entropia = -\frac{1}{46} log_2(\frac{1}{46}) - \frac{45}{46} log_2(\frac{45}{46}) \approx 0.1$$

Probabilidades



[50/150, 50/150, 50/150] = **[33%, 33%, 33%]**

50/50, 0/50, 0/50] = **[100%, 0, 0]**

[0/100, 50/100, 50/100] = **[0, 50%, 50%]**

[0/54, 49/54, 5/54] = **[0, 90.7%, 9.3%]**

[0/46, 1/46, 45/46] = **[0, 2.2%, 98.8%]**

Gini ou Entropia

- Usar Gini ou Entropia, na maioria das vezes tanto faz, pois leva a árvores semelhantes.

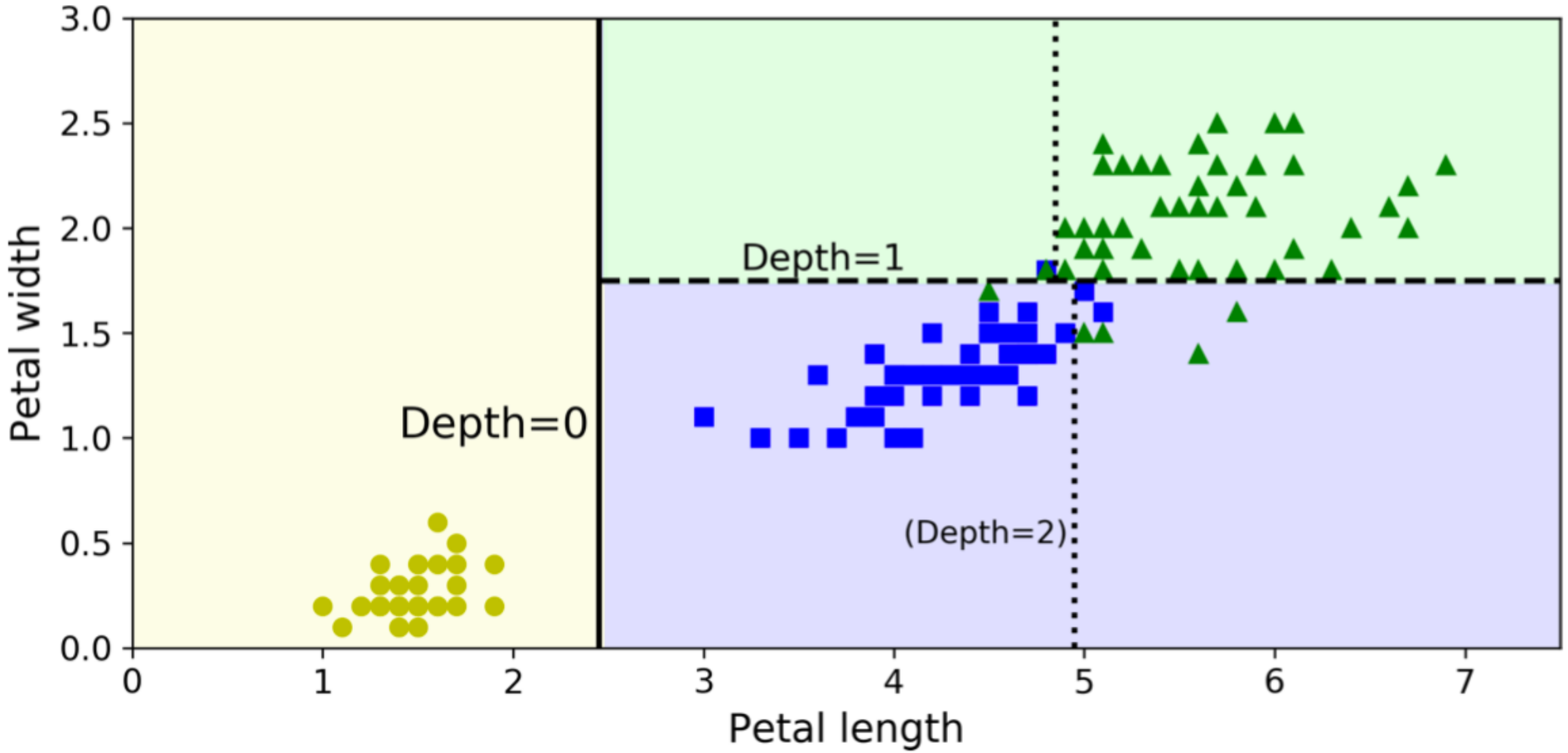
- O coeficiente de Gini é um pouco mais rápido para calcular, então é um bom padrão.

- No entanto, quando as árvores diferem, o coeficiente de Gini tende a isolar a classe mais frequente em seu próprio ramo da árvore, enquanto a entropia tende a produzir árvores ligeiramente mais equilibradas.

Algoritmo CART

- O *Scikit-Learn* usa o algoritmo Árvore de Classificação e Regressão, ou CART, em inglês.
- A ideia é bem simples: o algoritmo primeiro divide o conjunto de treinamento em dois subconjuntos utilizando uma única característica *k* e um limiar t_k (por exemplo, o comprimento da pétala ≤ 2.45 cm).
- Ele busca pelo par (k, t_k) que produz os subconjuntos mais puros.
- Depois de dividir com sucesso o conjunto de treinamento em dois, ele divide os subconjuntos utilizando a mesma lógica, depois os subsubconjuntos e assim por diante, recursivamente.
- Ele para quando atinge uma profundidade máxima (*max_depth*) ou não consegue encontrar uma divisão que reduza a impureza.

Fronteira de decisão da Árvore de Decisão



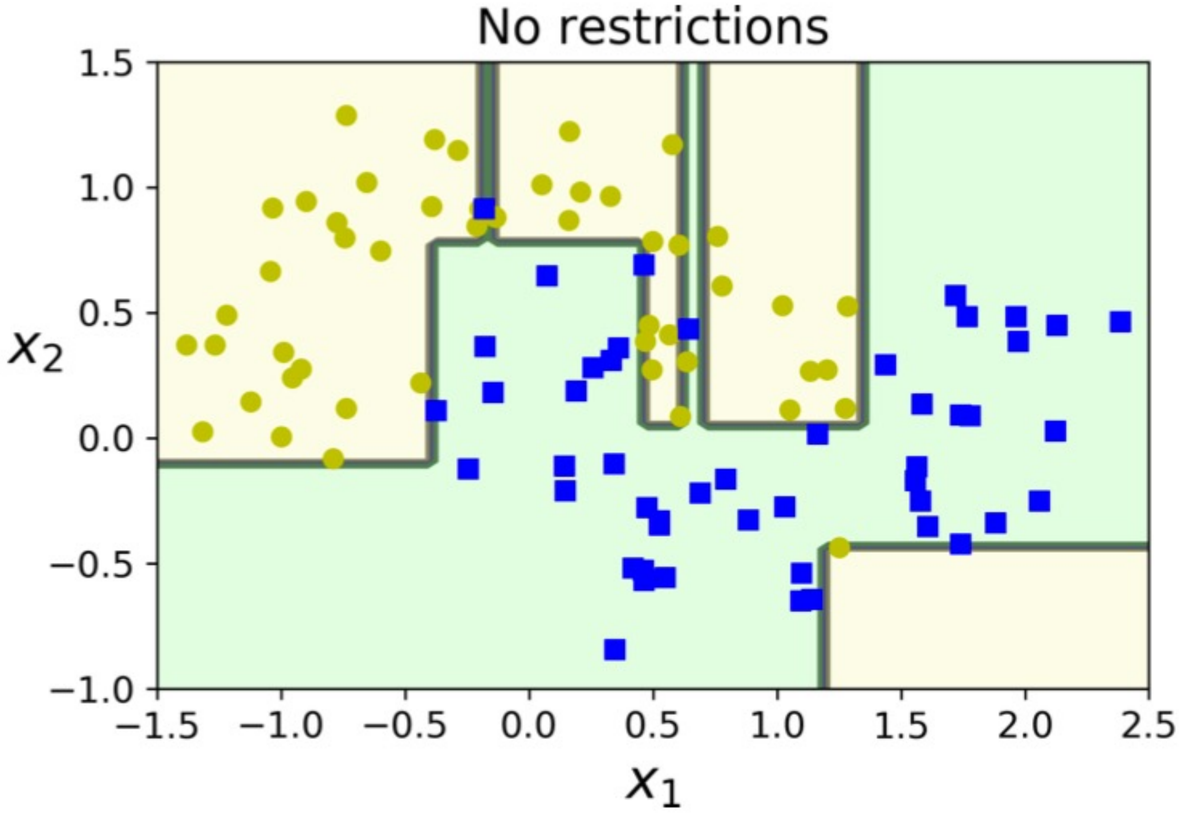
Hiperparâmetros e Regularização

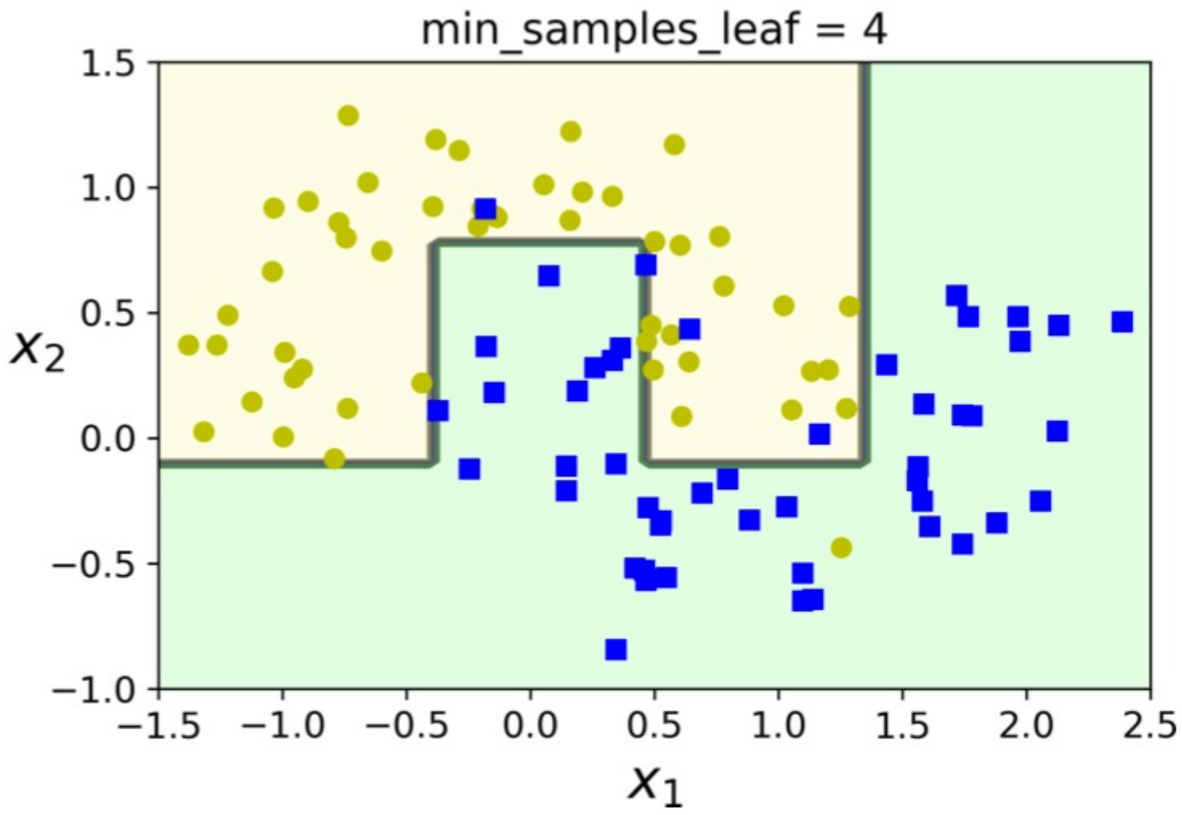
- Se o CART vai aprofundando a árvore até não conseguir mais reduzir a impureza, a tendência é que ele se sobreajuste ao conjunto de dados causando *overfitting.*

- Para evitar o overfitting é necessário restringir a liberdade do algoritmo. Isso se chama **regularização**.

- Para regularizar o modelo, usamos **hiperparâmetros.**

Hiperparâmetros e Regularização





Hiperparâmetros `DecisionTreeClassifier`

- `criterion`: gini ou entropy
- `max_depth`: profundidade máxima da árvore; None deixa livre
- `min_samples_split`: o número de amostras que um nó deve ter até que possa ser dividido
- `min_samples_leaf`: o mínimo de amostras que um nó da folha deve ter
- `min_weight_fraction`: o mesmo que min_samples_leaf, mas expressa como uma fração do número total de instâncias ponderadas
- `max_leaf_nodes`: número máximo de características que são avaliadas para divisão em cada nó

Dica: aumentar os hiperparâmetros min_* ou reduzir os hipermarâmetros max_* irá regularizar o modelo