

Algoritmos de Classificação

Árvores de Decisão

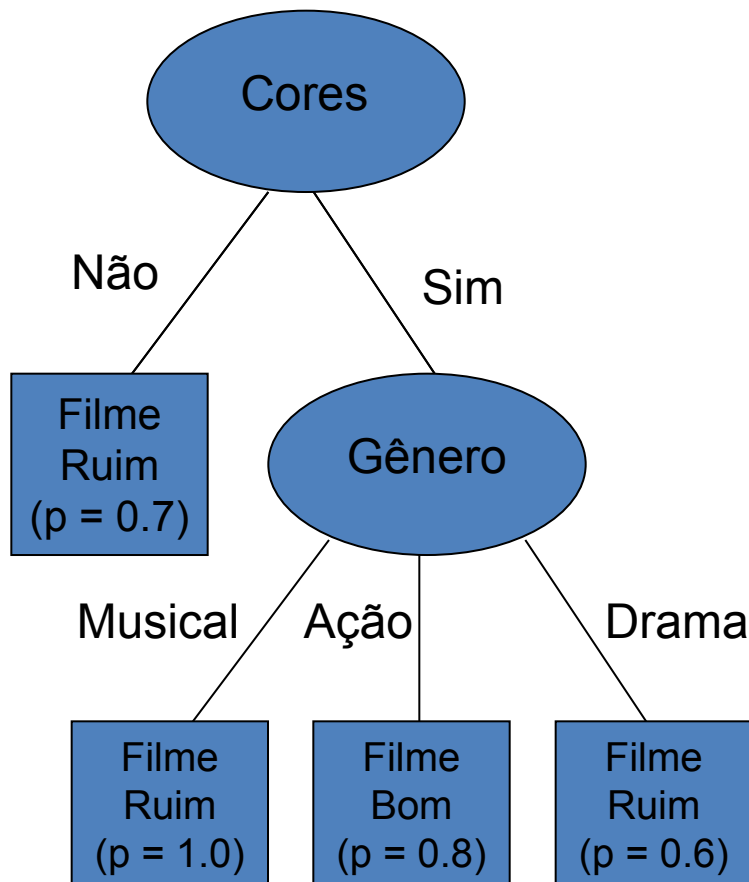
Árvores de Decisão

- Conhecimento adquirido pode ser representado em *linguagens* de alto nível
 - De forma legível e interpretável por humanos
- Motivações
 - Compreender um problema (mais do que obter modelos precisos)
 - Justificar decisões
 - Incorporar novo conhecimento

Árvores de Decisão

- Ampla classe de algoritmos de aprendizado
 - Exemplo: ID3, C4.5, CART,...
- Conhecimento representado em uma *árvore de decisão*, em geral, na linguagem da lógica de atributos

Árvores de Decisão



- Cada *nó interno* (*não-terminal*) contém um *teste* sobre os valores de um dado atributo
- *Folhas* da árvore (*nós terminais*) são associadas às classes
 - Comumente, acompanhadas com *graus de confiança*
- Novas instâncias classificadas percorrendo a árvore a partir da *raiz* até as folhas

Árvores de Decisão

- Construção

- Árvore de decisão construída de forma *recursiva* da raiz para as folhas (*top-down*)
 - A cada nó, é escolhido um *teste* que separe melhor os exemplos de classes diferentes
 - Maximização de *critério de separação*
 - Nós terminais são criados ao atingir um *critério de parada*
 - Ex.: todos os exemplos do nó pertencem à uma só classe

Árvores de Decisão

- Construção

- AD(Exemplos:D; Atributos:A; Alvo:C)
 - Crie nó_raiz
 - SE *Critério_de_Parada*
ENTÃO Crie nó terminal associada à classe mais freqüente
 - SENÃO Encontre *atributo aj* cujo *teste de decisão* maximize a *separação* dos exemplos que atingem o nó
 - PARA CADA *valor v* do teste adicione nova sub-árvore
 - Sub_arvore = AD($D_{[a_j = v]}$, A, C)

Árvores de Decisão

– Critérios de Separação

- Ganho de Informação (*Information Gain*)
 - *Impureza* ou incerteza de um nó pode ser medida através da *Entropia*

$$Ent(C, D) = - \sum_{c_i} \frac{|D_{[C=c_i]}|}{|D|} * \log_2 \frac{|D_{[C=c_i]}|}{|D|}$$

–

– Propriedades da Entropia:

- Se todos os exemplos de D são da mesma classe então entropia assume valor **mínimo**
 - $Ent(C, D) = 0$
- Se todas as classes têm o mesmo número de exemplos em D então entropia assume valor **máximo**

Árvores de Decisão

– Critérios de Separação

- Ganho de Informação (*Information Gain*)
 - O ganho de um atributo/teste é definido pela *redução de Entropia* proporcionada após a separação dos exemplos do nó

$$InfoGain(a, C, D) = Ent(C, D) - \sum_{v_i} \frac{|D_{[a=v_i]}|}{|D|} * Ent(C, D_{[a=v_i]})$$

↓
Entropia do nó pai

↓
Entropia do nó filho
ponderada pelo número
de exemplos do nó

Árvores de Decisão

– Critérios de Parada

- Totalidade (ou alternativamente, a maioria) do exemplos do nó pertencem a mesma classe
- Profundidade máxima para o nó
- Número mínimo de exemplos no nó
- Ganho pouco significativo no critério de separação
 - Obs.: valores definidos como *parâmetros* do aprendizado

Árvores de Decisão - Exemplo

- | Day | Outlook | Temp. | Humit. | Wind | Play |
|-----|----------|-------|--------|--------|------|
| D1 | Sunny | Hot | High | Weak | No |
| D2 | Sunny | Hot | High | Strong | No |
| D3 | Overcast | Hot | High | Weak | Yes |
| D4 | Rain | Mild | High | Weak | Yes |
| D5 | Rain | Cool | Normal | Weak | Yes |
| D6 | Rain | Cool | Normal | Strong | No |
| D7 | Overcast | Cool | Normal | Strong | Yes |
| D8 | Sunny | Mild | High | Weak | No |
| D9 | Sunny | Cool | Normal | Weak | Yes |
| D10 | Rain | Mild | Normal | Weak | Yes |
| D11 | Sunny | Mild | Normal | Strong | Yes |
| D12 | Overcast | Mild | High | Strong | Yes |
| D13 | Overcast | Hot | Normal | Weak | Yes |
| D14 | Rain | Mild | High | Strong | No |

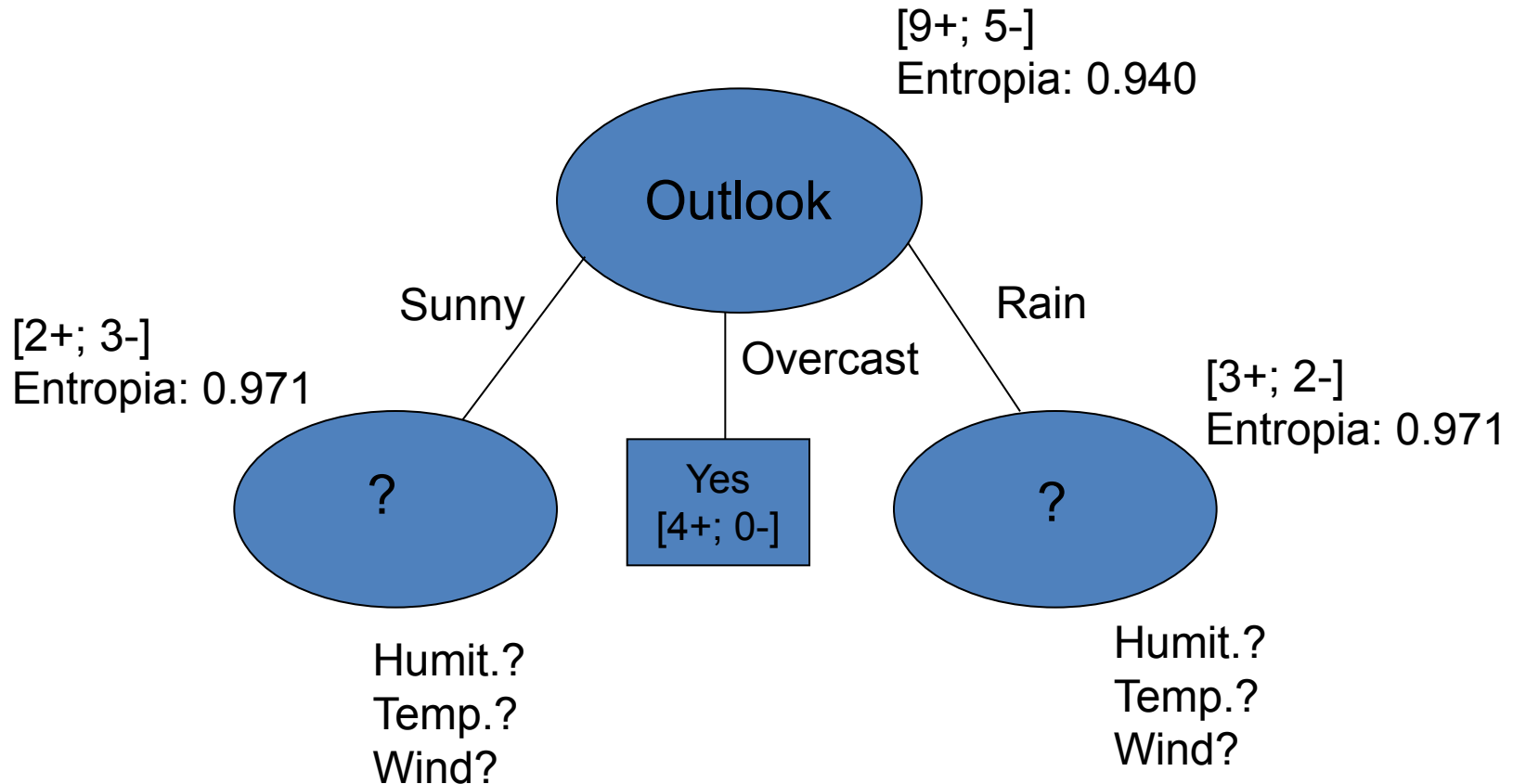
Árvores de Decisão - Exemplo

- Raíz: [9+; 5-]
 - Entropia = $- 9/14 \cdot \log_2(9/14) - 5/14 \cdot \log_2(5/14) = 0.940$
- Considerando teste com atributo Outlook
 - Outlook = Sunny: [2+;3-]
 - Entropia = $- 2/5 \cdot \log_2(2/5) - 3/5 \cdot \log_2(3/5) = 0.971$
 - Outlook = Overcast: [4+;0-]
 - Entropia = $- 4/4 \cdot \log_2(4/4) - 0/4 \cdot \log_2(0/4) = 0.0$
 - Outlook = Rain: [3+;2-]
 - Entropia = $- 3/5 \cdot \log_2(3/5) - 2/5 \cdot \log_2(2/5) = 0.971$
 - Média: $5/14 \cdot 0.971 + 4/14 \cdot 0.0 + 5/14 \cdot 0.971 = 0.694$
 - Ganho de Informação: $0.940 - 0.694 = 0.246$

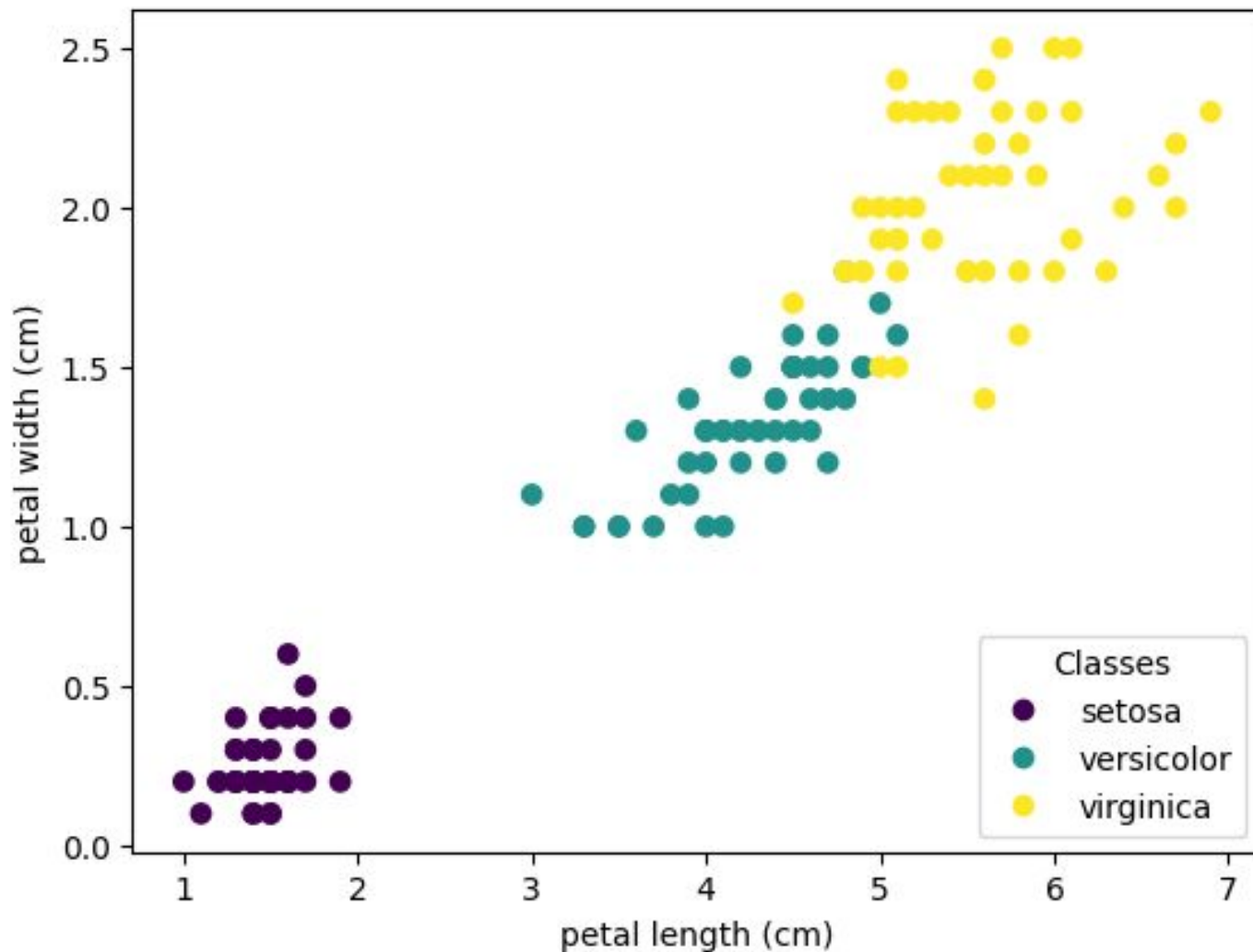
Árvores de Decisão - Exemplo

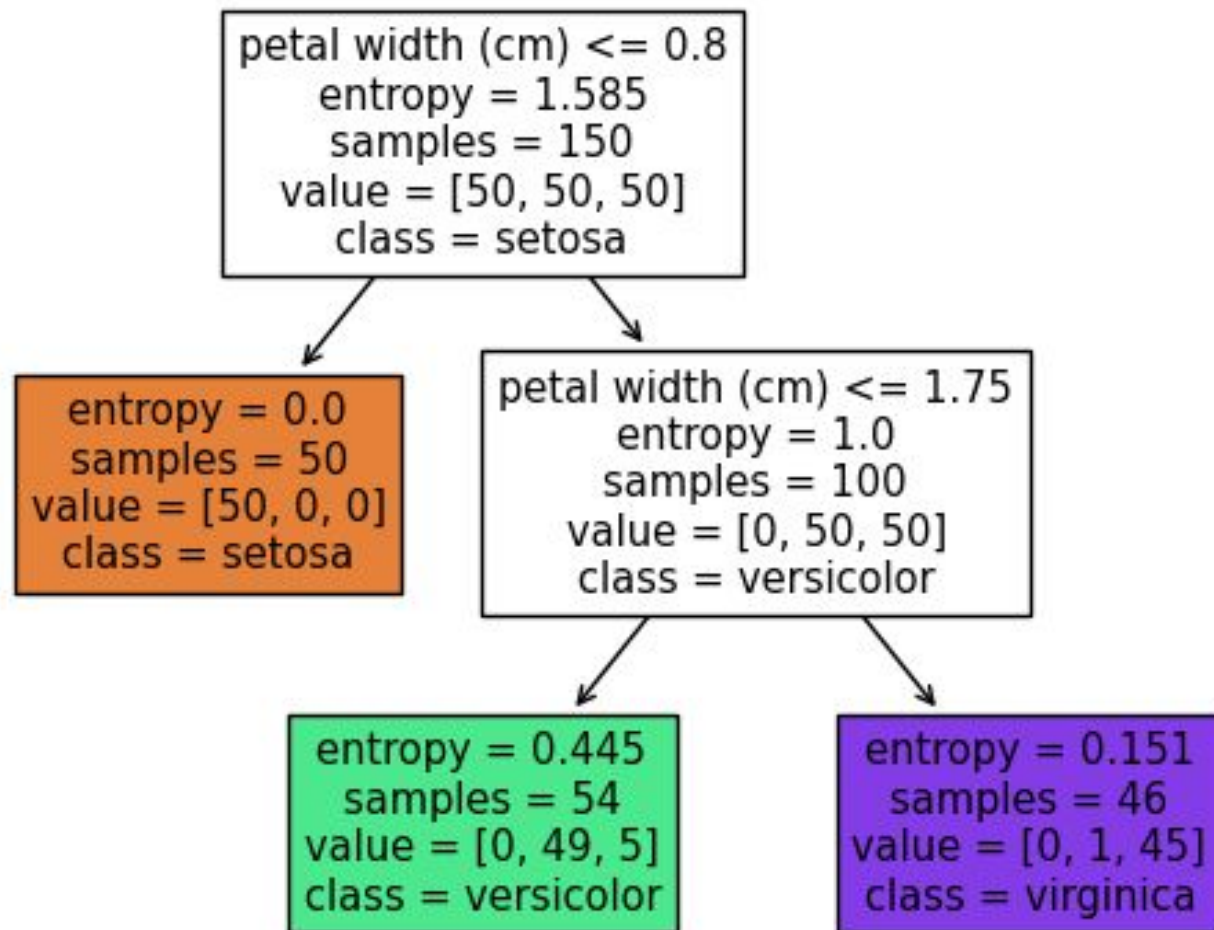
- Considerando os outros atributos:
 - $\text{Ganho}(\text{Outlook}, D) = 0.246$
 - $\text{Ganho}(\text{Humit.}, D) = 0.151$
 - $\text{Ganho}(\text{Wind}, D) = 0.048$
 - $\text{Ganho}(\text{Temp.}, D) = 0.029$
 - Atributo **Outlook** é o escolhido na raiz

Árvores de Decisão - Exemplo



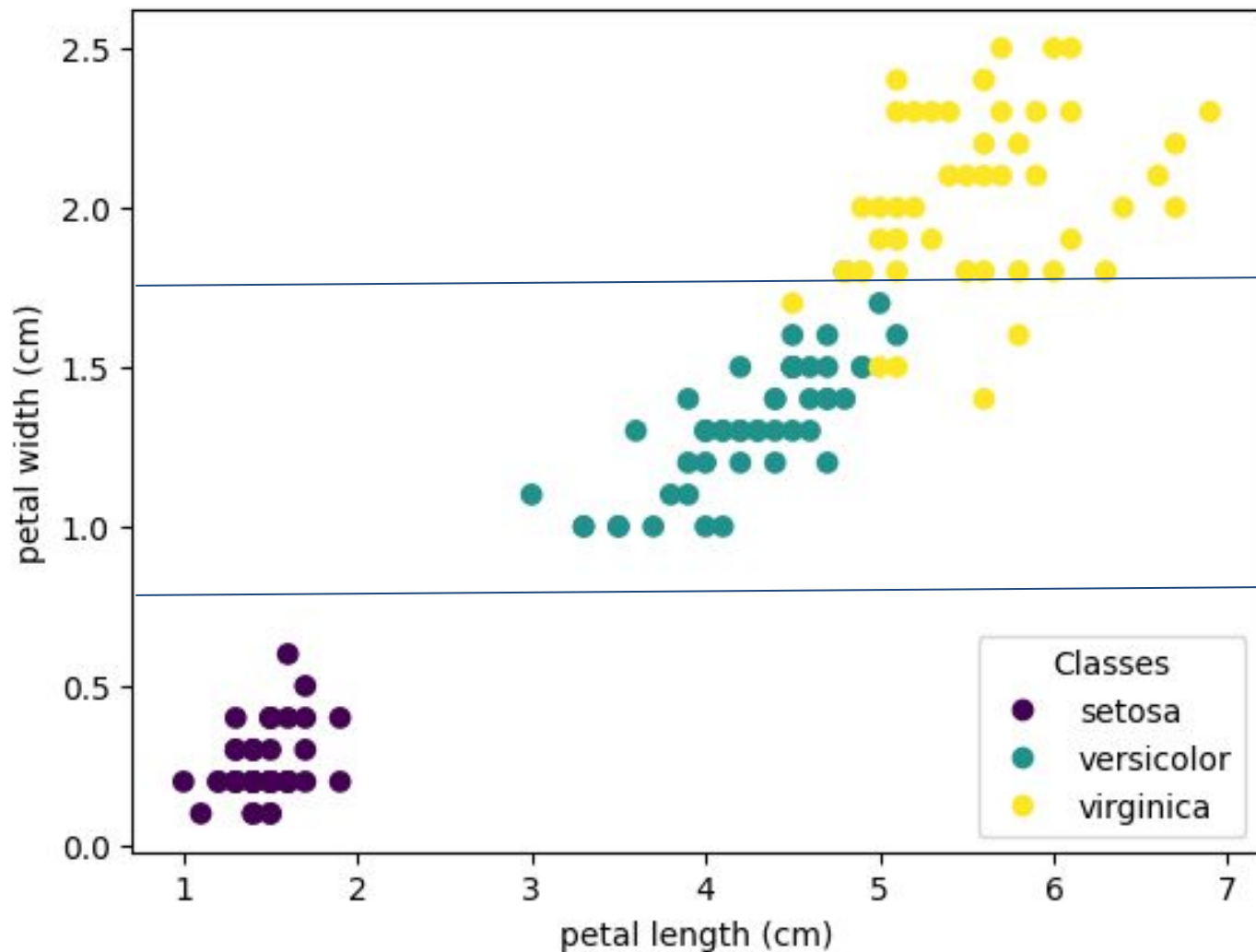
Outro exemplo: Iris Dataset

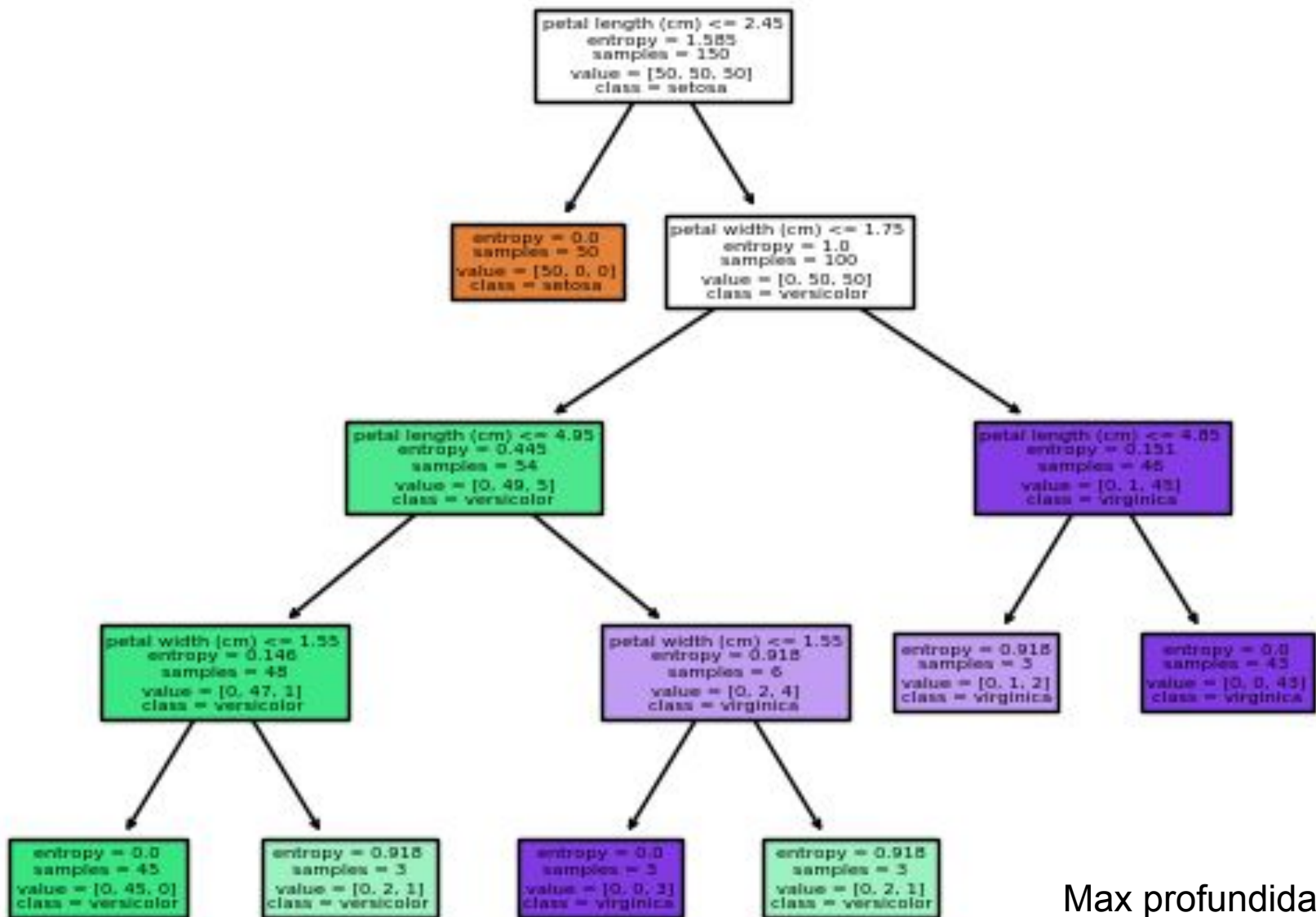




Max profundidade = 2

Outro exemplo: Iris Dataset





Max profundidade = 4

Overfitting!

Árvores de Decisão

- Discussão

- Vantagens:
 - Geram modelos dos dados (i.e., método *eager*)
 - Conhecimento interpretável
 - Pouca sensibilidade a atributos irrelevantes
 - Uma vez que implementam seleção de atributos
- Desvantagens:
 - Em geral, menos precisos comparados com algoritmos como redes neurais e SVMs
 - Superfícies muito simples de decisão

Random Forests

- Variação de árvores de decisões muito mais poderosa
- **Comitê** de árvores de decisão é treinado
 - Pequenas árvores geradas com **conjuntos de atributos** e **conjuntos de exemplos** diferentes

Random Forests - Treinamento

- Construa N árvores de decisão:
 - Passo 1: Selecione aleatoriamente um subconjunto pequeno de atributos
 - Passo 2: Faça uma **amostragem bootstrap** do conjunto de treinamento
 - I.e.: reamostragem com reposição
 - Passo 3: Gerar árvore de decisão
- Resultado: N árvores de decisão (o **comitê**)
 - Em geral bem **fracas** globalmente

Random Forests - Uso

- Dado um exemplo de teste
 - Colete as respostas das N árvores de decisão
 - Contar os votos dados a cada classe
- Resposta final:
 - Classe **majoritária** entre as respostas do comitê

Ver RandomForest no
WEKA

Referências

- T. Mitchell, *Machine Learning*, Cap. 3, 1997.
- I. Witten, E. Frank, 2000. *Data Mining – Practical Machine Learning Tools and Techniques with Java Implementations*.
- M. Monard, J. Baranauskas, Indução de Regras e Árvores de Decisão, *Sistemas Inteligentes*, Cap. 5, 2005.
- J. R. Quinlan, Induction of Decision Trees, *Machine Learning*, Vol.1, N.1, 1986.