



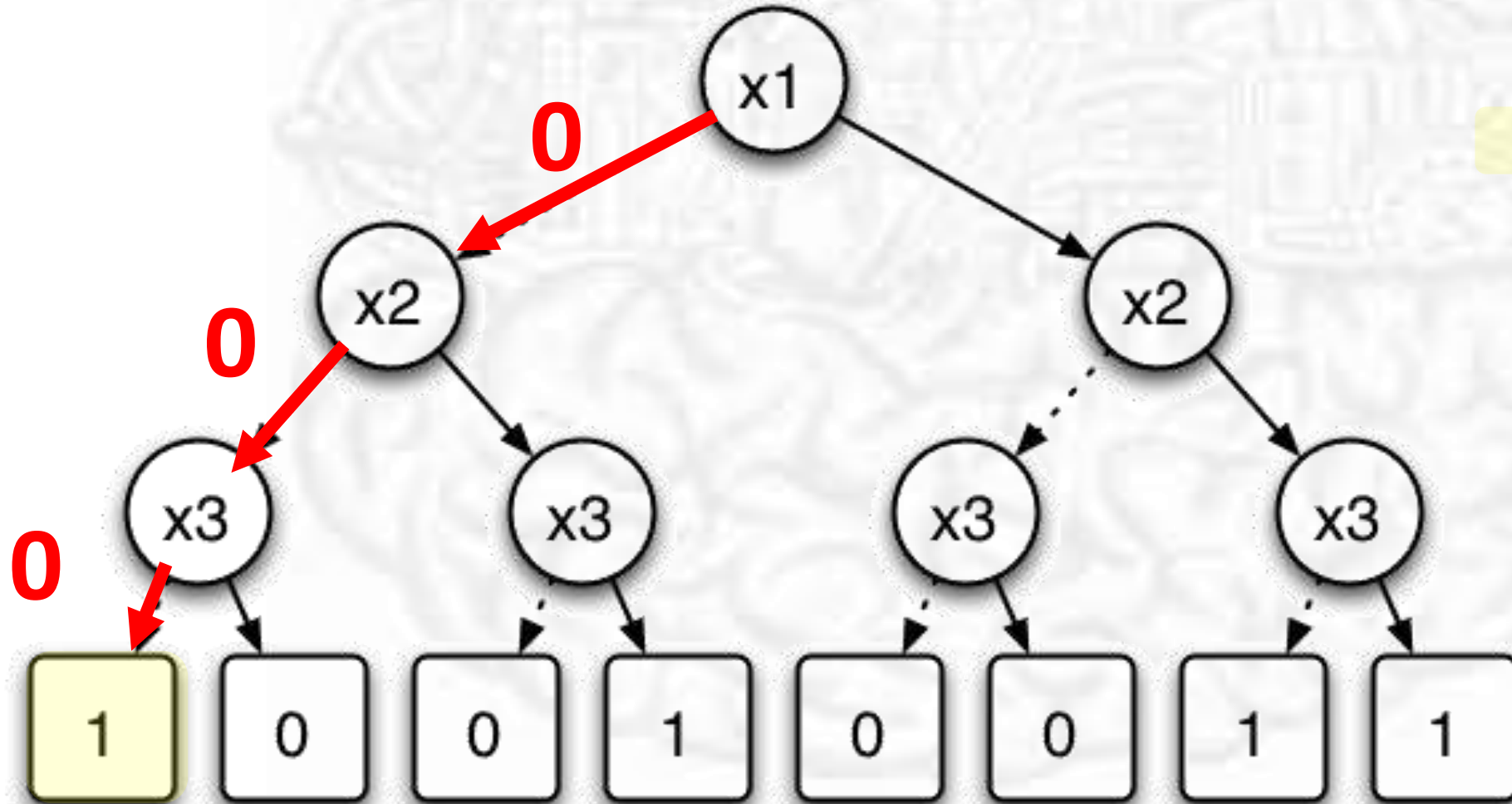
Machine Learning* com Python

Parte 4 – Árvores de Decisão

Árvores de decisão

- DT: Decision Tree
- Método de aprendizagem supervisionada não paramétrico usado para classificação e regressão.
- Objetivo:
 - Criar um modelo que prevê o valor de uma variável de destino, aprendendo regras de decisão simples inferidas a partir dos recursos de dados.

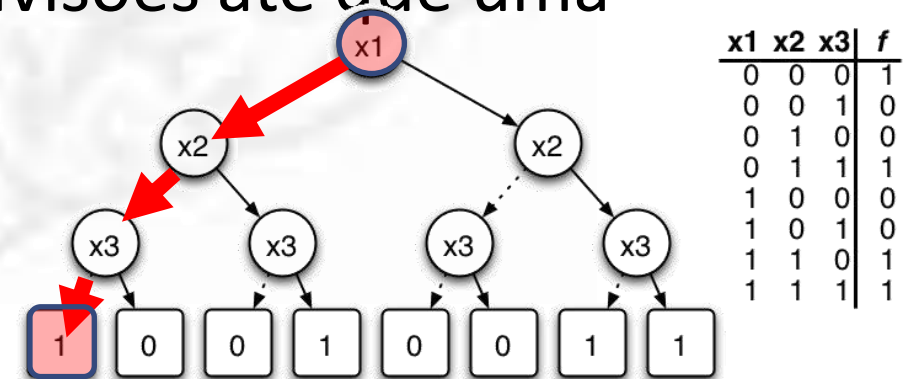
Exemplo de Árvore de Decisão Binária



x_1	x_2	x_3	f
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Árvore de Decisão binária

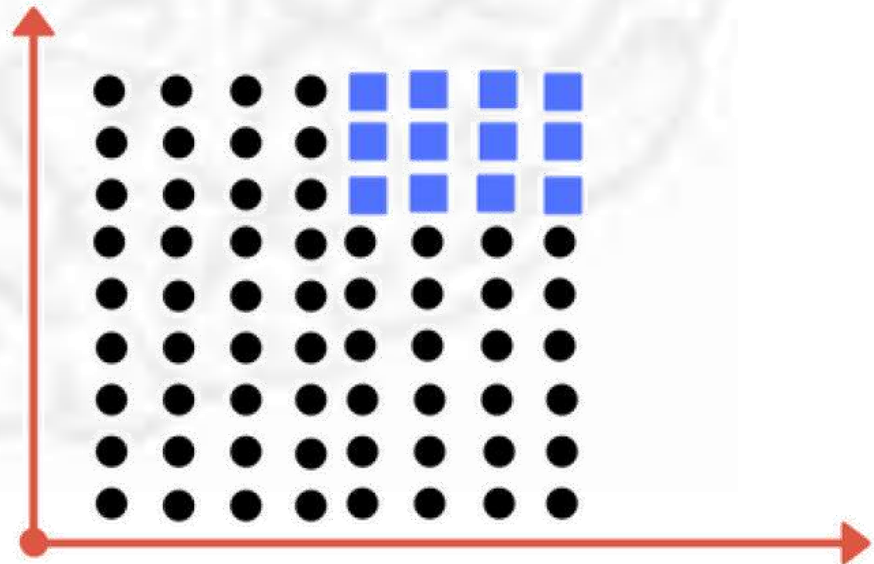
- A representação do modelo DT é uma árvore binária.
 - cada nó pode ter zero, um ou dois nós filhos
- Um nó representa uma variável de entrada única (X) e um ponto de divisão nessa variável, assumindo que a variável é numérica.
- Os nós da folha (também chamados nós terminais) da árvore contêm uma variável de saída (y) que é usada para fazer uma previsão (resposta).
- Uma vez criada, uma árvore pode ser navegada com uma nova linha de dados após cada ramificação com as divisões até que uma previsão final seja feita.



Motivação

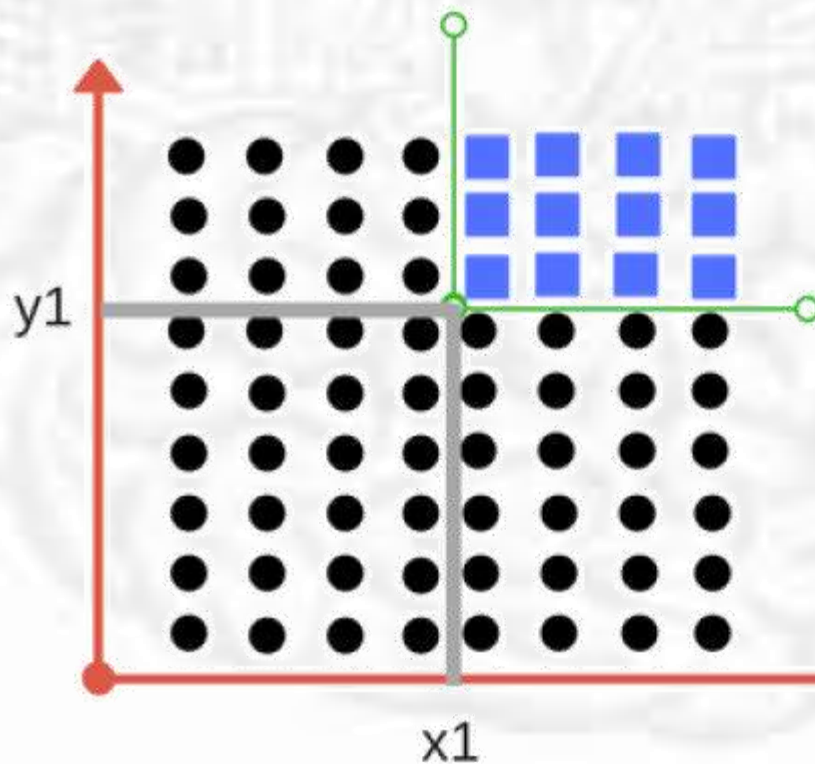
- Suponha que tenhamos uma distribuição de treino para duas classes representadas por círculos preto e quadrados azuis.
- É possível desenhar uma única linha de separação?

Talvez não!



Motivação

- Precisamos de mais de uma linha, para dividir em classes.

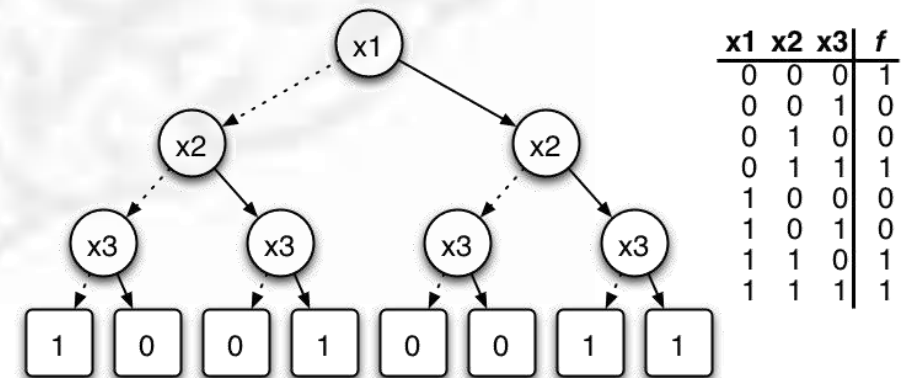


Precisamos de duas linhas aqui separando de acordo com o valor limiar de x e outro para o valor limiar de y .

Isto é o que a árvore de decisão tenta fazer.

Árvore de Decisão - 1

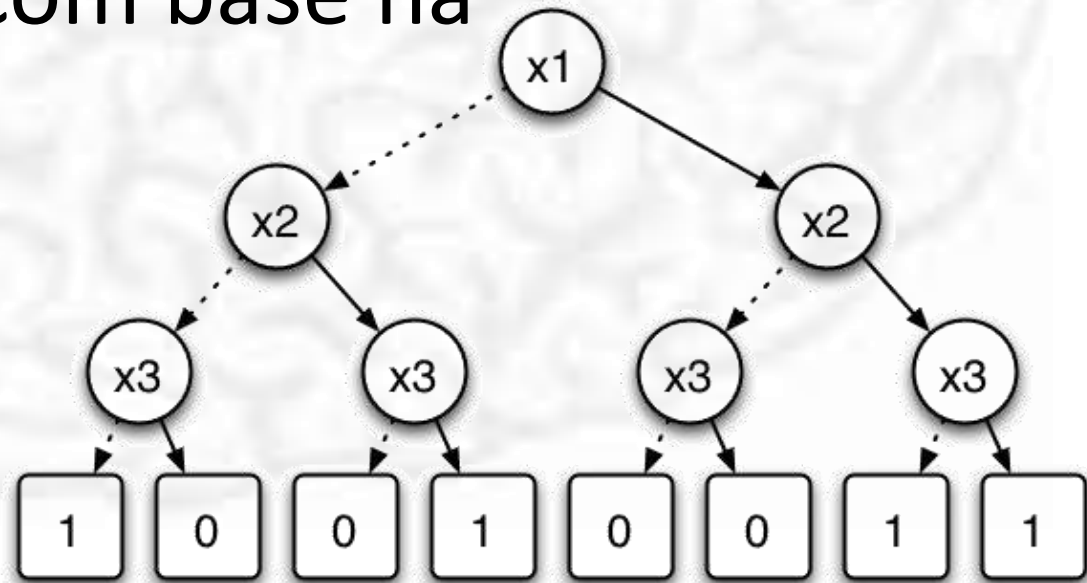
- Criar uma árvore de decisão é, na verdade, um processo de dividir o espaço de entrada.
- Uma abordagem gulosa é usada para dividir o espaço chamado divisão binária recursiva.
 - todos os valores são alinhados e diferentes pontos de divisão são testados usando uma função de custo.
 - A divisão com o melhor custo é selecionada.



Árvore de Decisão - 2

- Todas as variáveis de entrada e todos os possíveis pontos de divisão são avaliados e escolhidos de maneira gulosa com base na função de custo.

- Regressão
- Classificação

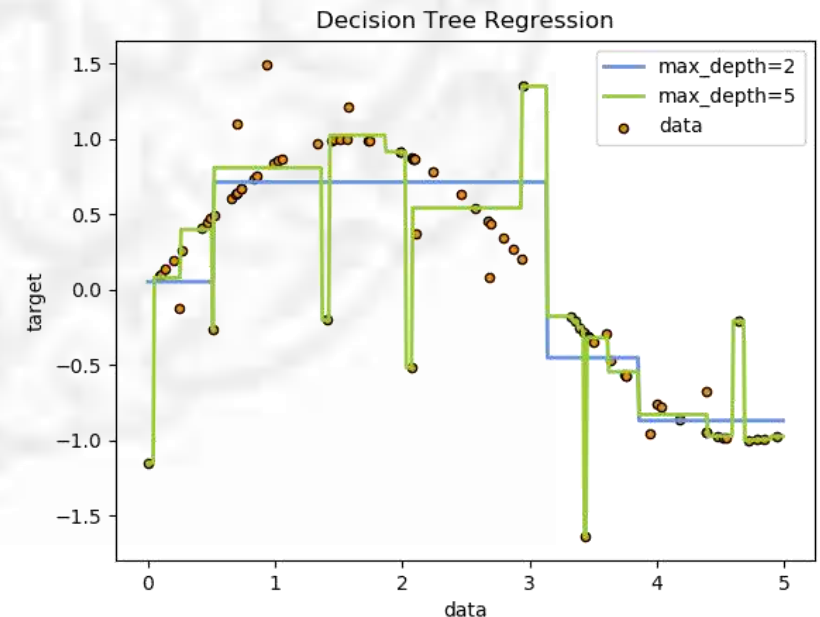


x1	x2	x3	f
0	0	0	1
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1

Árvore de Decisão - 3

Regressão:

- A função de custo que é minimizada para escolher pontos de divisão, onde o erro de soma quadrática em todas as amostras de treinamento que se enquadram no retângulo.



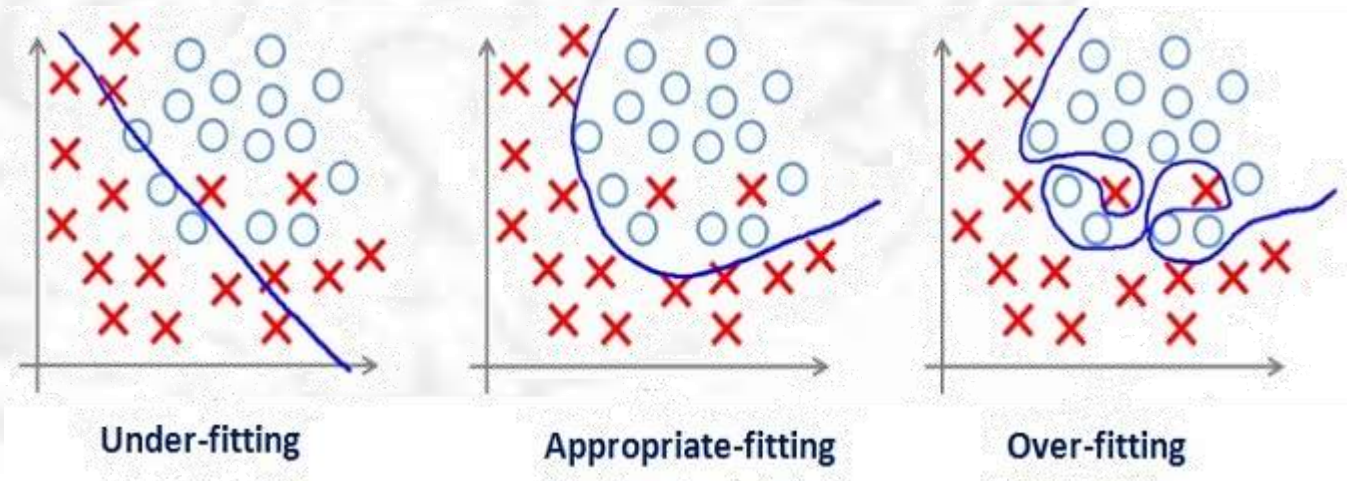
Árvore de Decisão - 4

Classificação

- Duas funções principais
 - A função de custo de Gini é usada, o que fornece uma indicação de quão puros são os nós, onde a pureza do nó se refere a quão misturados os dados de treinamento são atribuídos a cada nó.
 - O conceito de entropia pode ser utilizado também (algoritmo ID3).

Árvore de Decisão - 5

- Quanto parar?
- A divisão continua até que os nós contenham um número mínimo de exemplos de treinamento ou uma profundidade máxima de árvore seja atingida.
- Evitar overfitting





ALGORITMO ID3

Algoritmo ID3

Iterative Dichotomiser 3

- Vamos apenas pegar um conjunto de dados famoso no mundo de aprendizado de máquina, que é o problema de prever se jogamos ou não com base na condição climática.
 - Quatro valores X (outlook, temp, umidade e com vento) categóricos e um valor y (play Y ou N) categórico.
 - Precisamos aprender o mapeamento entre X e y.
- Problema de classificação binária → construir a árvore usando o algoritmo ID3

Algoritmo ID3

- Quatro valores X (outlook, temp, umidade e com vento) categóricos e um valor y (play Y ou N) categórico.

Outlook	Temp	Umidade	Vento	Jogar?
Ensolarado	Quente	Alta	Falso	não
Ensolarado	Quente	Alta	Verdadeiro	Não
Nublado	Quente	Alta	Falso	Sim
Chuvoso	Ameno	Alta	Falso	Sim
Chuvoso	Fresco	Normal	Falso	Sim
Chuvoso	Fresco	Normal	Verdadeiro	Não
Nublado	Fresco	Normal	Verdadeiro	Sim
Ensolarado	Ameno	Alta	Falso	Não
Ensolarado	Fresco	Normal	Falso	Sim
Chuvoso	Ameno	Normal	Falso	Sim
Ensolarado	Ameno	Normal	Verdadeiro	Sim
Nublado	Ameno	Alto	Verdadeiro	Sim
Nublado	Quente	Normal	Falso	Sim
Chuvoso	Ameno	Alto	Verdadeiro	Não

Algoritmo ID3

- Precisamos aprender o mapeamento entre X e y .
 - Problema de classificação binária → construir a árvore usando o algoritmo ID3
- Para criar uma árvore, precisamos ter um nó raiz primeiro e sabemos que nós são recursos / atributos (perspectiva, temperatura, umidade e ventoso),

Algoritmo ID3

- Determinar o atributo que melhor classifica os dados de treinamento; use este atributo na raiz da árvore. Repita este processo para cada ramo.
- Isso significa que estamos realizando uma pesquisa de cima para baixo e gulosa no espaço de possíveis árvores de decisão.

Outlook	Temp	Umidade	Vento	Jogar?
Ensolarado	Quente	Alta	Falso	não
Ensolarado	Quente	Alta	Verdadeiro	Não
Nublado	Quente	Alta	Falso	Sim
Chuvoso	Ameno	Alta	Falso	Sim
Chuvoso	Fresco	Normal	Falso	Sim

Algoritmo ID3

- Como escolhemos o melhor atributo?
 - Usamos o atributo com o maior ganho de informação em ID3
- Para definir precisamente o ganho de informação, começamos definindo uma medida comumente usada na teoria da informação, chamada entropia, que caracteriza a (im)pureza de uma coleção arbitrária de exemplos.

Outlook	Temp	Umidade	Vento	Jogar?
Ensolarado	Quente	Alta	Falso	não
Ensolarado	Quente	Alta	Verdadeiro	Não
Nublado	Quente	Alta	Falso	Sim
Chuvoso	Ameno	Alta	Falso	Sim
Chuvoso	Fresco	Normal	Falso	Sim

Algoritmo ID3

- Entropia
 - É uma medida da quantidade de incerteza no conjunto de dados D, isto é, caracteriza os dados

$$H(D) = - \sum_{c \in C} p(c) \log_2 p(c)$$

Onde

- D é o conjunto de dados sobre o qual calculamos a entropia
- C é o conjunto de classes da resposta
 - No nosso exemplo $C = \{\text{não}, \text{sim}\}$
- $p(c)$ é a proporção de elementos na classe c dentro do conjunto S

Em ID3, a entropia é calculada para cada atributo restante. O atributo com a menor entropia é utilizado para dividir o conjunto D.

Outlook	Temp	Umidade	Vento	Jogar?
Ensolarado	Quente	Alta	Falso	não
Ensolarado	Quente	Alta	Verdadeiro	Não
Nublado	Quente	Alta	Falso	Sim
Chuvoso	Ameno	Alta	Falso	Sim
Chuvoso	Fresco	Normal	Falso	Sim

Algoritmo ID3

- Para um problema de classificação binária
 - Se todos os exemplos forem positivos ou todos forem negativos, então a entropia será zero, ou seja, baixa.
 - Se metade dos exemplos são de classe positiva e metade são de classe negativa, então a entropia é um, ou seja, alta.

Outlook	Temp	Umidade	Vento	Jogar?
Ensolarado	Quente	Alta	Falso	não
Ensolarado	Quente	Alta	Verdadeiro	Não
Nublado	Quente	Alta	Falso	Sim
Chuvoso	Ameno	Alta	Falso	Sim
Chuvoso	Fresco	Normal	Falso	Sim

Algoritmo ID3

- Ganho de informação $IG(A)$
 - É uma medida da diferença da entropia antes do conjunto D ser dividido pelo atributo A , isto é, indica em quanto a incerteza de D é reduzida ao dividir D usando o atributo A

$$IG(A, D) = H(D) - \sum_{t \in T} p(t)H(t)$$

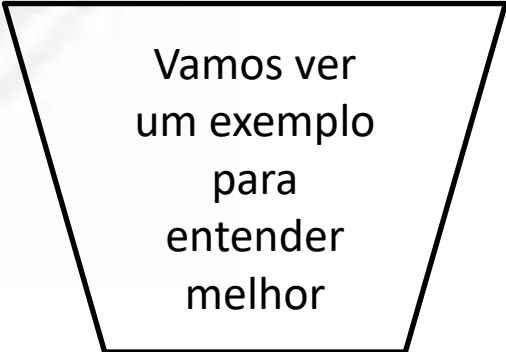
Onde

- $H(D)$ é a entropia do conjunto D
- T são os subconjuntos criados ao dividir D pelo atributo A
- $p(t)$ é a proporção entre o número de elementos no subconjunto t e o número de elementos em S
- $H(t)$ é a entropia do subconjunto t

Outlook	Temp	Umidade	Vento	Jogar?
Ensolarado	Quente	Alta	Falso	não
Ensolarado	Quente	Alta	Verdadeiro	Não
Nublado	Quente	Alta	Falso	Sim
Chuvoso	Ameno	Alta	Falso	Sim
Chuvoso	Fresco	Normal	Falso	Sim

Algoritmo ID3

- Vamos aplicar essas métricas ao nosso conjunto de dados para dividir os dados (obtendo o nó raiz)
- Passos:
 1. compute a entropia para o conjunto de dados
 2. para cada atributo / recurso:
 1. Entropia é calculada para todos os valores categóricos
 2. Obter entropia de informação média para o atributo atual
 3. O Ganho é calculado para o atributo atual
 3. Escolha o atributo de maior ganho.
 4. Repita até conseguirmos a árvore que desejamos.



Vamos ver
um exemplo
para
entender
melhor

Algoritmo ID3

$C = \{\text{Sim}, \text{Não}\}$

- Calcule a entropia para o conjunto de dados:

$$H(D) = - \sum_{c \in C} p(c) \log_2 p(c)$$

No conjunto D (ao lado) temos:

Total de elementos = 14

Total com Sim = 9 $\rightarrow p_{\text{sim}} = -\frac{9}{14} \log_2 \frac{9}{14} \approx 0.41$

Total com Não = 5 $\rightarrow p_{\text{não}} = -\frac{5}{14} \log_2 \frac{5}{14} \approx 0.53$

$$H(D) = p_{\text{sim}} + p_{\text{não}} = 0.94$$

Outlook	Temp	Umidade	Vento	Jogar?
Ensolarado	Quente	Alta	Falso	não
Ensolarado	Quente	Alta	Verdadeiro	Não
Nublado	Quente	Alta	Falso	Sim
Chuvoso	Ameno	Alta	Falso	Sim
Chuvoso	Fresco	Normal	Falso	Sim
Chuvoso	Fresco	Normal	Verdadeiro	Não
Nublado	Fresco	Normal	Verdadeiro	Sim
Ensolarado	Ameno	Alta	Falso	Não
Ensolarado	Fresco	Normal	Falso	Yes
Chuvoso	Ameno	Normal	Falso	Sim
Ensolarado	Ameno	Normal	Verdadeiro	Sim
Nublado	Ameno	Alto	Verdadeiro	Sim
Nublado	Quente	Normal	Falso	Sim
Chuvoso	Ameno	Alto	Verdadeiro	Não

Algoritmo ID3

- Calcular a entropia e ganho de informação para cada atributo

$$E_{outlook=Ensolarado} = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5} \approx 0.971$$

$$E_{outlook=Nublado} = -1\log_2 1 - 0\log_2 0 \approx 0$$

$$E_{outlook=Chuvoso} = -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5} \approx 0.971$$

Entropia média para Outlook

$$H_{D,outlook} = \frac{5}{14}0.971 + \frac{4}{14}0 + \frac{5}{14}0.971 = 0.693$$

Ganho de informação para Outlook

$$G(D, outlook) = H(D) - H_{D,outlook} = 0.94 - 0.693 = 0.247$$

Outlook	Temp	Umidade	Vento	Jogar?
Ensolarado	Quente	Alta	Falso	não
Ensolarado	Quente	Alta	Verdadeiro	Não
Nublado	Quente	Alta	Falso	Sim
Chuvoso	Ameno	Alta	Falso	Sim
Chuvoso	Fresco	Normal	Falso	Sim
Chuvoso	Fresco	Normal	Verdadeiro	Não
Nublado	Fresco	Normal	Verdadeiro	Sim
Ensolarado	Ameno	Alta	Falso	Não
Ensolarado	Fresco	Normal	Falso	Sim
Chuvoso	Ameno	Normal	Falso	Sim
Ensolarado	Ameno	Normal	Verdadeiro	Sim
Nublado	Ameno	Alto	Verdadeiro	Sim
Nublado	Quente	Normal	Falso	Sim
Chuvoso	Ameno	Alto	Verdadeiro	Não

Algoritmo ID3

- Calcular a entropia e ganho de informação para cada atributo

$$E_{temp=Quente} = -\frac{2}{4}\log_2\frac{2}{4} - \frac{2}{4}\log_2\frac{2}{4} \approx 1$$

$$E_{temp=Ameno} = -\frac{4}{6}\log_2\frac{4}{6} - \frac{2}{6}\log_2\frac{2}{6} \approx 0.918$$

$$E_{temp=Fresco} = -\frac{3}{4}\log_2\frac{3}{4} - \frac{1}{4}\log_2\frac{1}{4} \approx 0.811$$

Entropia média para *temp*

$$H_{D,temp} = \frac{4}{14} 1 + \frac{6}{14} 0.918 + \frac{4}{14} 0.231 = 0.911$$

Ganho de informação para Outlook

$$G(D, temp) = H(D) - H_{D,temp} = 0.94 - 0.911 = 0.029$$

Outlook	Temp	Umidade	Vento	Jogar?
Ensolarado	Quente	Alta	Falso	não
Ensolarado	Quente	Alta	Verdadeiro	Não
Nublado	Quente	Alta	Falso	Sim
Chuvoso	Ameno	Alta	Falso	Sim
Chuvoso	Fresco	Normal	Falso	Sim
Chuvoso	Fresco	Normal	Verdadeiro	Não
Nublado	Fresco	Normal	Verdadeiro	Sim
Ensolarado	Ameno	Alta	Falso	Não
Ensolarado	Fresco	Normal	Falso	Sim
Chuvoso	Ameno	Normal	Falso	Sim
Ensolarado	Ameno	Normal	Verdadeiro	Sim
Nublado	Ameno	Alto	Verdadeiro	Sim
Nublado	Quente	Normal	Falso	Sim
Chuvoso	Ameno	Alto	Verdadeiro	Não

Algoritmo ID3

- Calcular a entropia e ganho de informação para cada atributo

$$E_{umidade=Alta} = -\frac{3}{7}\log_2\frac{3}{7} - \frac{4}{7}\log_2\frac{4}{7} \approx 0.985$$

$$E_{umidade=Normal} = -\frac{6}{7}\log_2\frac{6}{7} - \frac{1}{7}\log_2\frac{1}{7} \approx 0.591$$

Entropia média para Umidade

$$H_{D,umidade} = \frac{7}{14}0.985 + \frac{7}{14}0.591 = 0.788$$

Ganho de informação para Umidade

$$G(D,umidade) = H(D) - H_{D,umidade} = 0.94 - 0.788 = 0.152$$

Outlook	Temp	Umidade	Vento	Jogar?
Ensolarado	Quente	Alta	Falso	não
Ensolarado	Quente	Alta	Verdadeiro	Não
Nublado	Quente	Alta	Falso	Sim
Chuvoso	Ameno	Alta	Falso	Sim
Chuvoso	Fresco	Normal	Falso	Sim
Chuvoso	Fresco	Normal	Verdadeiro	Não
Nublado	Fresco	Normal	Verdadeiro	Sim
Ensolarado	Ameno	Alta	Falso	Não
Ensolarado	Fresco	Normal	Falso	Sim
Chuvoso	Ameno	Normal	Falso	Sim
Ensolarado	Ameno	Normal	Verdadeiro	Sim
Nublado	Ameno	Alta	Verdadeiro	Sim
Nublado	Quente	Normal	Falso	Sim
Chuvoso	Ameno	Alta	Verdadeiro	Não

Algoritmo ID3

- Calcular a entropia e ganho de informação para cada atributo

$$E_{vento=Verdadeiro} = -\frac{6}{8}\log_2\frac{6}{8} - \frac{2}{8}\log_2\frac{2}{8} \approx 0.811$$

$$E_{vento=Falso} = -\frac{3}{6}\log_2\frac{3}{6} - \frac{3}{6}\log_2\frac{3}{6} \approx 0.1$$

Entropia média para Vento

$$H_{D,vento} = \frac{8}{14} 0.811 + \frac{6}{14} 1 = 0.892$$

Ganho de informação para Vento

$$G(D, vento) = H(D) - H_{D,vento} = 0.94 - 0.892 = 0.048$$

Outlook	Temp	Umidade	Vento	Jogar?
Ensolarado	Quente	Alta	Falso	não
Ensolarado	Quente	Alta	Verdadeiro	Não
Nublado	Quente	Alta	Falso	Sim
Chuvoso	Ameno	Alta	Falso	Sim
Chuvoso	Fresco	Normal	Falso	Sim
Chuvoso	Fresco	Normal	Verdadeiro	Não
Nublado	Fresco	Normal	Verdadeiro	Sim
Ensolarado	Ameno	Alta	Falso	Não
Ensolarado	Fresco	Normal	Falso	Yes
Chuvoso	Ameno	Normal	Falso	Sim
Ensolarado	Ameno	Normal	Verdadeiro	Sim
Nublado	Ameno	Alta	Verdadeiro	Sim
Nublado	Quente	Normal	Falso	Sim
Chuvoso	Ameno	Alta	Verdadeiro	Não

Algoritmo ID3

- Escolha o atributo de maior ganho.

OUTLOOK

Entropia _____: 0.693
Ganho de informação: 0.247

TEMPERATURA

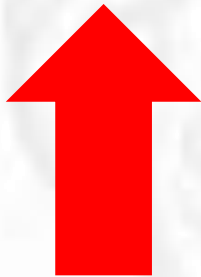
Entropia _____: 0.911
Ganho de informação: 0.029

UMIDADE

Entropia _____: 0.788
Ganho de informação: 0.152

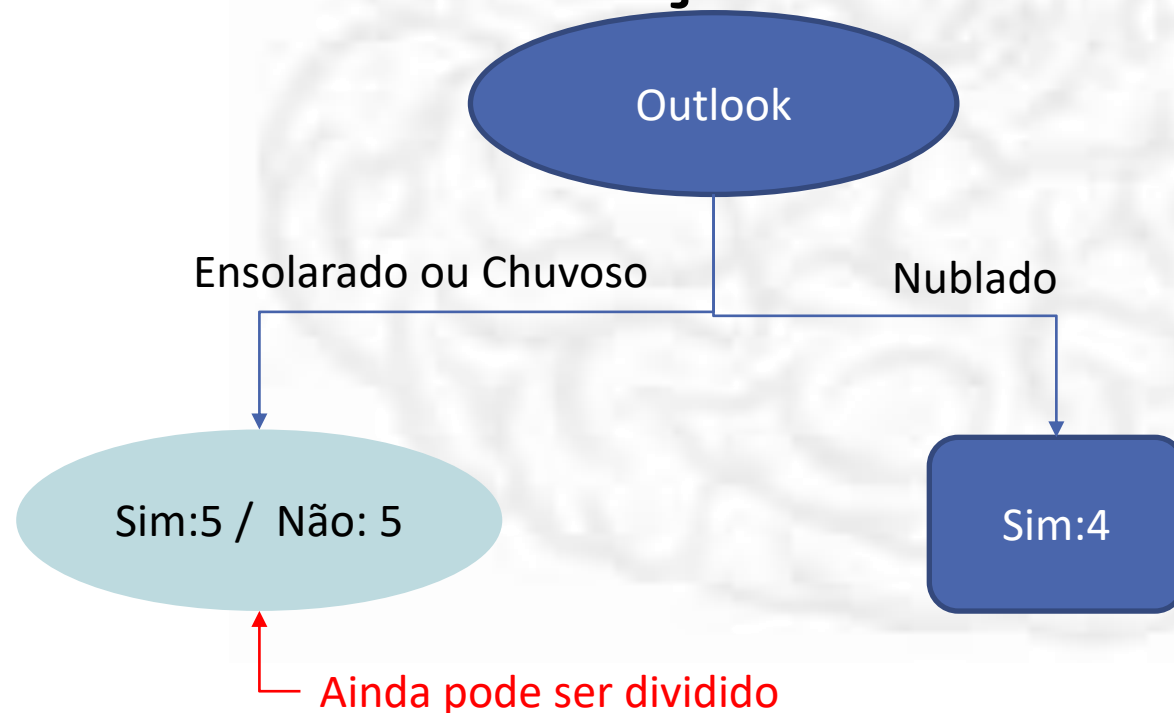
VENTO

Entropia _____: 0.892
Ganho de informação: 0.048



Algoritmo ID3

- Então, concluímos que nosso nó raiz é o Outlook e a árvore começa



Outlook	Temp	Umidade	Vento	Jogar?
Ensolarado	Quente	Alta	Falso	Não
Ensolarado	Quente	Alta	Verdadeiro	Não
Nublado	Quente	Alta	Falso	Sim
Chuvoso	Ameno	Alta	Falso	Sim
Chuvoso	Fresco	Normal	Falso	Sim
Chuvoso	Fresco	Normal	Verdadeiro	Não
Nublado	Fresco	Normal	Verdadeiro	Sim
Ensolarado	Ameno	Alta	Falso	Não
Ensolarado	Fresco	Normal	Falso	Sim
Chuvoso	Ameno	Normal	Falso	Sim
Ensolarado	Ameno	Normal	Verdadeiro	Sim
Nublado	Ameno	Alta	Verdadeiro	Sim
Nublado	Quente	Normal	Falso	Sim
Chuvoso	Ameno	Alta	Verdadeiro	Não

Algoritmo ID3

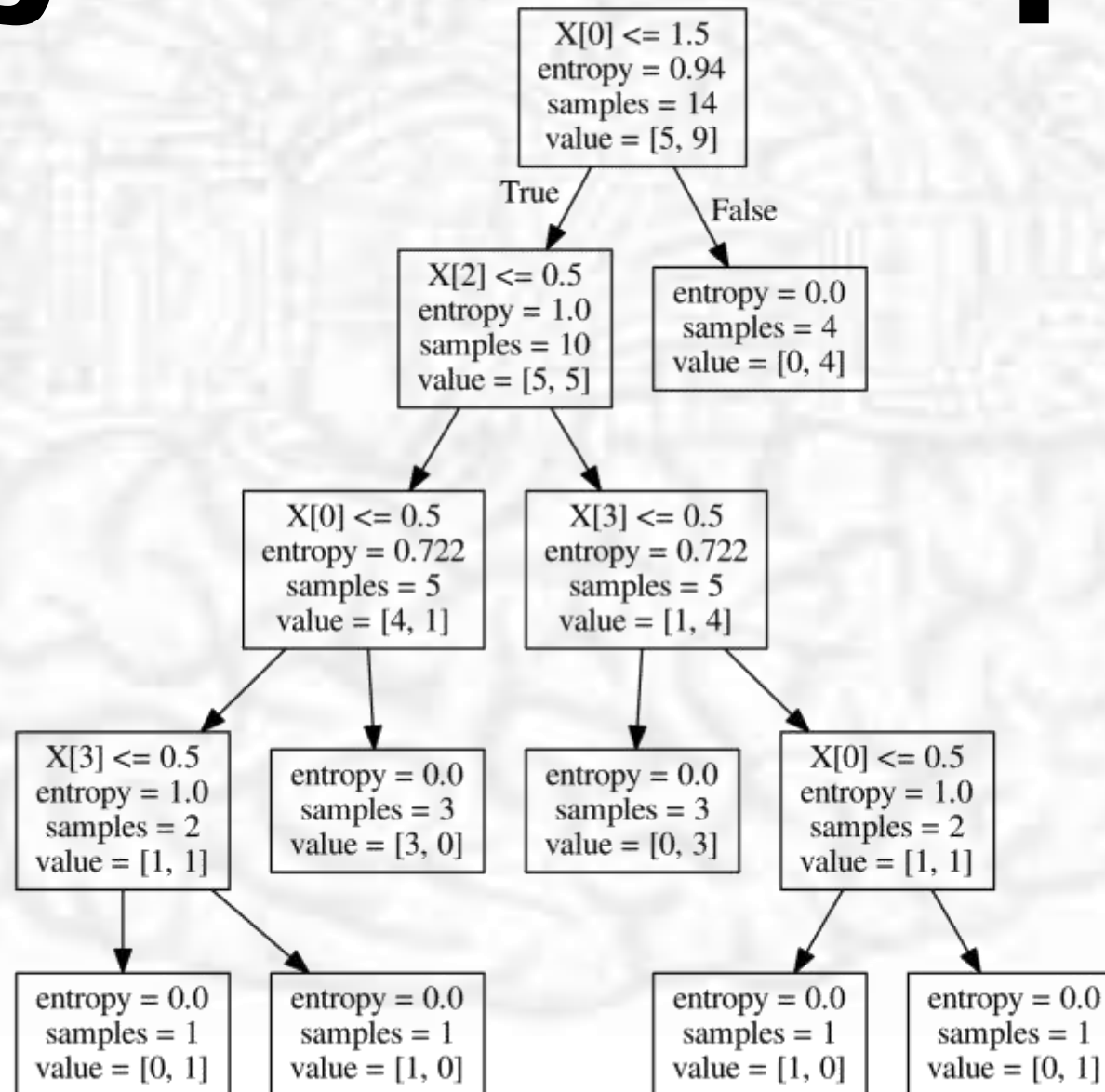
Outlook	Temp	Umidade	Vento	Jogar?
Ensolarado	Quente	Alta	Falso	não
Ensolarado	Quente	Alta	Verdadeiro	Não
Ensolarado	Ameno	Alta	Falso	Não
Ensolarado	Fresco	Normal	Falso	Sim
Ensolarado	Ameno	Normal	Verdadeiro	Sim
Chuvoso	Ameno	Alta	Falso	Sim
Chuvoso	Fresco	Normal	Falso	Sim
Chuvoso	Fresco	Normal	Verdadeiro	Não
Chuvoso	Ameno	Normal	Falso	Sim
Chuvoso	Ameno	Alta	Verdadeiro	Não

Python

- Biblioteca scikit-learn
 - `sklearn.tree.DecisionTreeClassifier`
 - `criterion=entropy`



Python - Entropia



ÍNDICE DE GINI

Índice de GINI para uma variável binária

- O índice de GINI dá uma ideia de quão boa é uma divisão pela forma como as classes são misturadas nos dois grupos criados pela divisão.

$$GINI = 1 - \sum_{i=0}^1 p_i^2$$

p_i é a frequência relativa de cada classe em cada nó

Índice de Gini para uma variável com k classes

- O índice de GINI para k classes é semelhante, basta somar das devidas proporções para as k classes

$$GINI = 1 - \sum_{i=0}^k p_i^2$$

p_i é a frequência relativa de cada classe em cada nó

Índice de GINI

- Quando o índice é igual a zero, o nó é puro.
- Quando ele se aproxima do valor um, o nó é impuro
 - aumenta o número de classes uniformemente distribuídas neste nó

Algoritmo

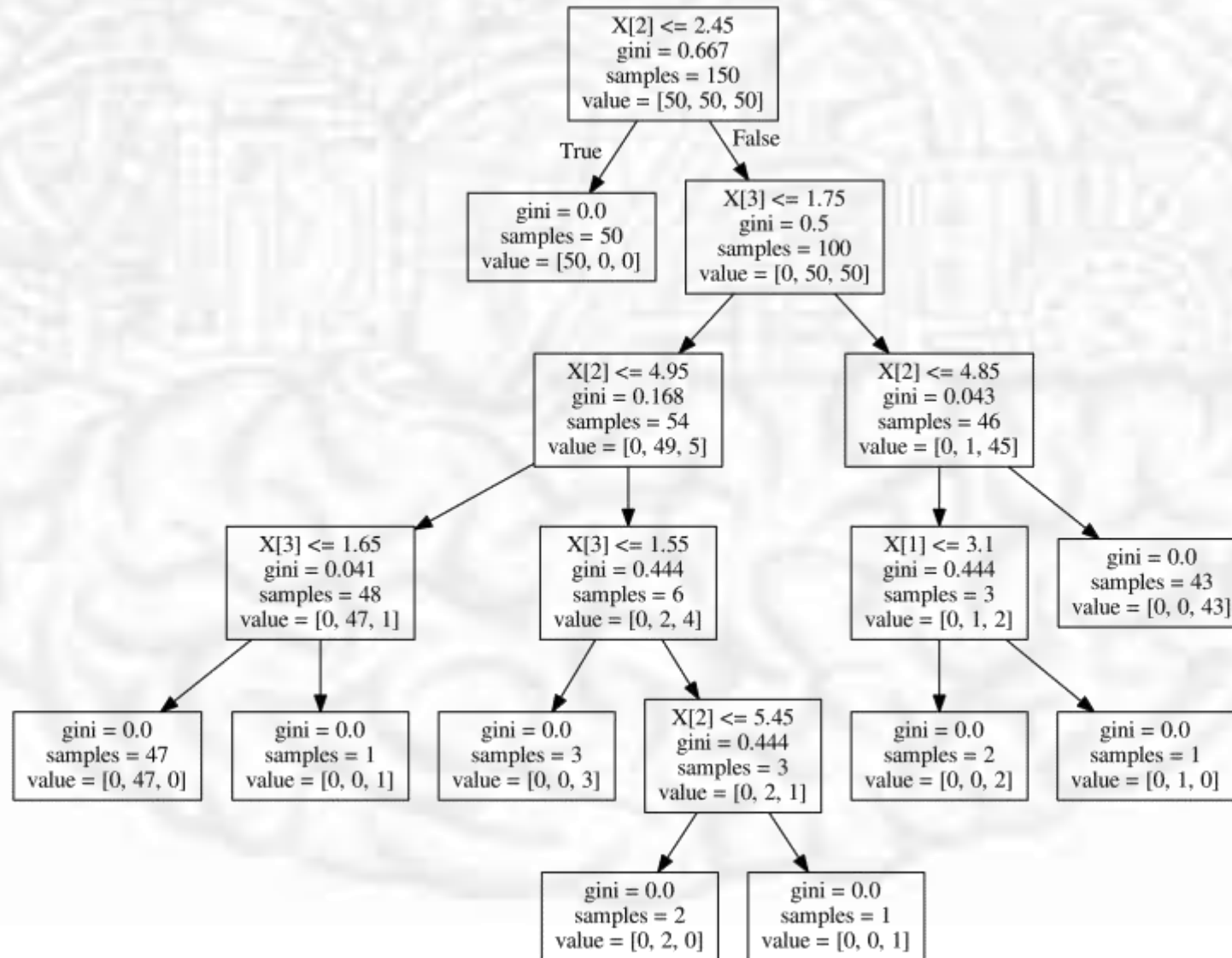
1. compute o índice de GINI para o conjunto de dados
2. para cada atributo:
 1. Calcule o índice de GINI para todos os valores categóricos
 2. Obter entropia média para o atributo atual
 3. Calcular o ganho de GINI
3. Escolha o atributo com o melhor de ganho do GINI
4. Repita até conseguirmos a árvore que desejamos.

Python

- Biblioteca scikit-learn
 - `sklearn.tree.DecisionTreeClassifier`
 - `criterion=gini` **# default**



Python - Gini



PRUNING



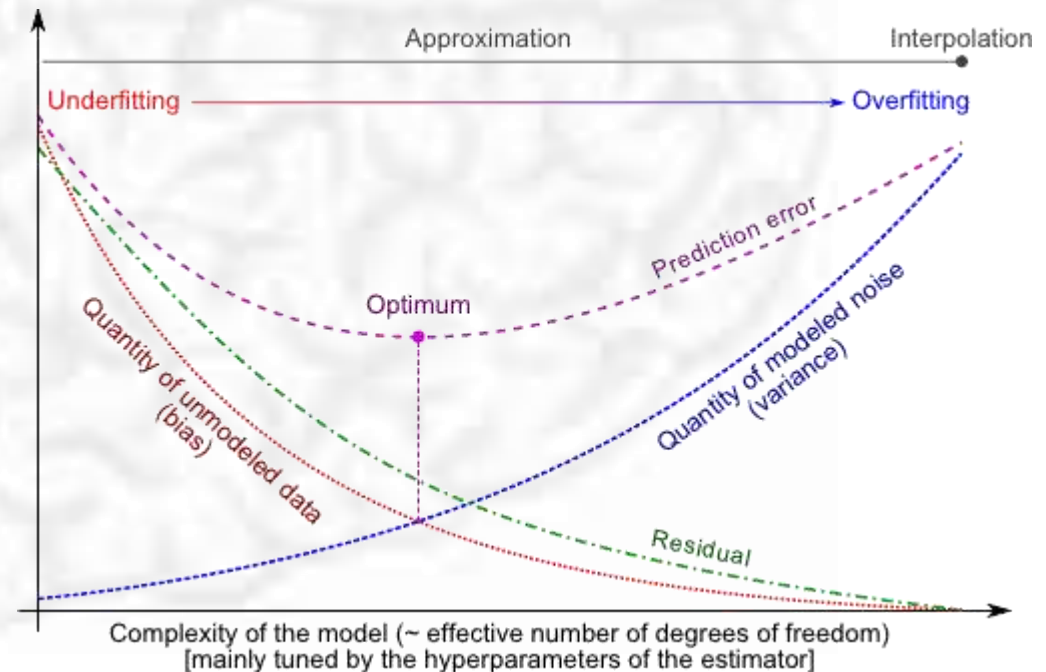
Overfitting

- A definição de overfitting é:
 - uma árvore de decisão d faz sobre-ajustamento aos dados se existir uma árvore d' tal que: d tem menor erro que d' no conjunto de treino mas d' tem menor erro na população”.

Gama (2004)

Overfitting

- Para saber qual é o número ótimo de nós, representa-se graficamente o percentual de erro no conjunto de treinamento e teste versus o número de nós da árvore.
- Quando o erro no conjunto de teste começar a crescer, verifica-se o número de nós nesse ponto.



Overfitting

- Solução:
 - pruning
- A podagem pode ser dada de duas circunstâncias:
 - Pode ser usada para parar o crescimento da árvore mais cedo, chamada de pré-podagem ou poda descendente ou
 - Pode acontecer com a árvore já completa, chamada de pós-podagem ou poda ascendente.

Processo de podagem

1. Percorre a árvore em profundidade.
2. Para cada nó de decisão calcular:
 1. erro no nó
 2. soma dos erros nos nós descendentes
3. Se o erro do nó é menor ou igual à soma dos erros dos nós descendentes então o nó é transformado em folha

Pós-podagem

- Na pós-podagem a árvore é gerada no tamanho máximo
- Então a árvore é podada aplicando métodos de evolução confiáveis.
- A sub-árvore com o melhor desempenho será a escolhida.
- Este processo pode ser computacionalmente ineficiente pelo fato de gerar uma árvore muito grande e depois esta mesma árvore é reduzida a uma árvore mínima.

Pré-podagem

- Para interromper o crescimento da árvore, verifica-se se a divisão é confiável ou não.
- Caso seja confiável, devemos parar o crescimento da árvore.
- É mais rápida porém menos eficiente que a pós-podagem
 - risco de interromper o crescimento da árvore ao selecionar uma árvore sub-ótima.

O que o sklearn oferece?

3 parâmetros de remoção

- **max_leaf_nodes**
 - Reduzir o número de nós de folhas
- **min_samples_leaf**
 - Restringir o tamanho da folha de amostra
 - O tamanho mínimo da amostra nos nós terminais pode ser fixado em 30, 100, 300 ou 5% do total
- Profundidade máxima (**max_depth**)
 - Reduza a profundidade da árvore para construir uma árvore generalizada
 - Defina a profundidade da árvore para 3, 5, 10, dependendo da verificação nos dados de teste

CONCLUSÃO

Vantagens

- Simples de entender e interpretar. Árvores podem ser visualizadas.
- Requer pouca preparação de dados. Outras técnicas geralmente requerem a normalização de dados, variáveis dummy precisam ser criadas e valores em branco a serem removidos. No entanto, observe que este módulo não suporta valores ausentes.
- O custo de usar a árvore (ou seja, prever dados) é logarítmico no número de pontos de dados usados para treinar a árvore.
- Capaz de lidar com dados numéricos e categóricos. Outras técnicas são geralmente especializadas na análise de conjuntos de dados que possuem apenas um tipo de variável. Veja algoritmos para mais informações.
- Capaz de lidar com problemas de várias saídas.
- Usa um modelo de caixa branca. Se uma determinada situação é observável em um modelo, a explicação para a condição é facilmente explicada pela lógica booleana. Em contraste, num modelo de caixa preta (por exemplo, numa rede neural artificial), os resultados podem ser mais difíceis de interpretar.
- Possível validar um modelo usando testes estatísticos. Isso torna possível explicar a confiabilidade do modelo.
- Funciona bem, mesmo que suas suposições sejam de algum modo violadas pelo verdadeiro modelo do qual os dados foram gerados.

Desvantagens

- As funções de aprendizado da árvore de decisão podem criar árvores super complexas que não generalizam bem os dados. Isso é chamado overfitting.
 - Mecanismos como poda, definição do número mínimo de amostras necessárias em um nó folha ou configuração da profundidade máxima da árvore são necessários para evitar esse problema.
- As árvores de decisão podem ficar instáveis porque pequenas variações nos dados podem resultar na geração de uma árvore completamente diferente. Esse problema é atenuado pelo uso de árvores de decisão em um *ensemble*.
- O problema de aprender uma árvore de decisão ótima é conhecido por ser NP-completo sob vários aspectos de otimalidade e até mesmo para conceitos simples. Consequentemente, os algoritmos práticos de aprendizagem da árvore de decisão são baseados em algoritmos heurísticos, como o algoritmo guloso, em que decisões locais ótimas são tomadas em cada nó. Tais algoritmos não podem garantir o retorno da árvore de decisão ótima globalmente. Isso pode ser atenuado pelo treinamento de várias árvores em um aprendiz conjunto, onde os recursos e amostras são amostrados aleatoriamente com a substituição.
- Existem conceitos que são difíceis de aprender porque as árvores de decisão não as expressam facilmente, como problemas de XOR, paridade ou multiplexadores.
- Os alunos da árvore de decisão criam árvores tendenciosas se algumas classes dominam. Portanto, recomenda-se equilibrar o conjunto de dados antes de se ajustar à árvore de decisão.