Aprendizado de Máquinas
Árvore de decisão

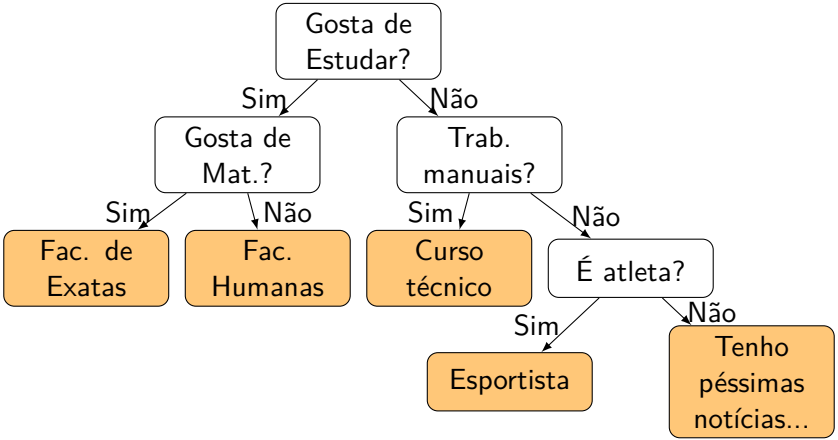
Douglas Rodrigues

Universidade Federal Fluminense

Árvore de decisão

- Ideia: criar uma árvore de decisão para classificar indivíduos, baseado nas suas características.
- Não há a ideia de distribuição: utilizamos apenas os dados de cada indivíduo para criar uma regra de separação.
- A separação deve envolver apenas duas respostas: sim ou não.

Árvore de decisão



Algoritmo

1. Coloque todas variáveis em um grupo.
2. Encontrar a variável que melhor separa os dados (com menos impurezas). Em geral, utilizamos o índice Gini, que varia entre 0 (mais puro possível) e 0.5 (mais impuro possível). No caso binário:

   Gini(nó)= 1 - (prop. do "tipo 1")^2 - (prop. de "tipo 2")^2
3. Escolhemos a variável com menor índice Gini para separar os dados.
4. Repetimos o processo.

Exemplo

- Vamos carregar o banco de dados `golf`. Ele apresenta observações sobre as condições climáticas e se o indivíduo jogou golfe no dia. Descarte a coluna `Day`.
- A ideia é tentar predizer se o indivíduo foi jogar golfe, baseado nas condições climáticas.
- Nossas variáveis de interesse são “Outlook”, “Temperature”, “Humidity”, “Wind” e “Play”.

Exemplo

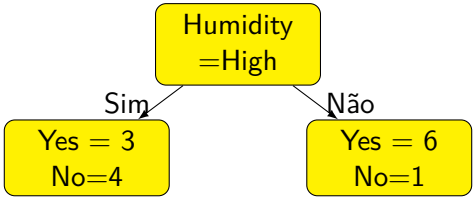
- Primeiro, precisamos decidir qual variável ficará no topo da árvore. Para isso, vamos calcular o índice de impureza Gini para cada variável.
- Vamos começar com a variável `Humidity`.

$$\texttt{Humidity:} \begin{cases} 7 \ \texttt{High} \\ 7 \ \texttt{Normal} \end{cases}$$

- Como a resposta deve ser sim ou não, vou transformar essa variável em uma cuja resposta seja `sim` ou `não`.

$$\texttt{Humidity=High:} \begin{cases} 7 \ \texttt{Sim} \\ 7 \ \texttt{Não} \end{cases}$$

Exemplo

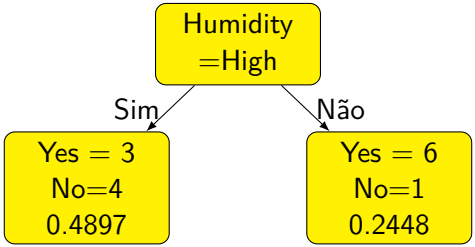


- Calculando índice de impureza Gini para o nó `[Humidity=High Não]`:

$$Gini(\text{H=H Não}) = 1 - \left(\frac{6}{7}\right)^2 - \left(\frac{1}{7}\right)^2 = 0.2448$$

- Calculando índice de impureza Gini para o nó `[Humidity=High Sim]`:

$$Gini(\text{H=H Sim}) = 1 - \left(\frac{3}{7}\right)^2 - \left(\frac{4}{7}\right)^2 = 0.4897$$

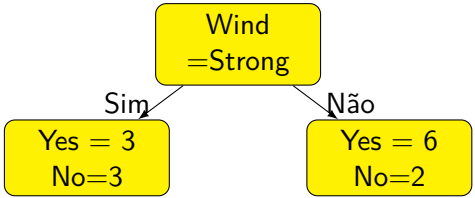


- Agora, calculamos o índice Gini da variável `[Humidity=High]`, que será a média do índice para as respostas `Sime Não`, ponderado pelo frequência de elementos em cada nó.

$$\begin{aligned} Gini(\text{Humidity=High}) &= 0.2448 \cdot \frac{7}{14} + 0.4897 \frac{7}{14} \\ &= \mathbf{0.3673} \end{aligned}$$

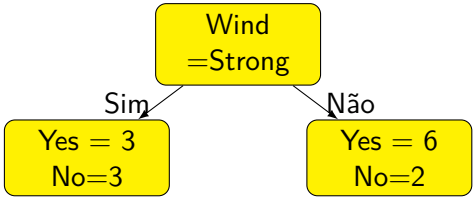
Exercício

1. Calcule o índice de impureza Gini para a variável `Wind`.



Exercício

1. Calcule o índice de impureza Gini para a variável `Wind`.



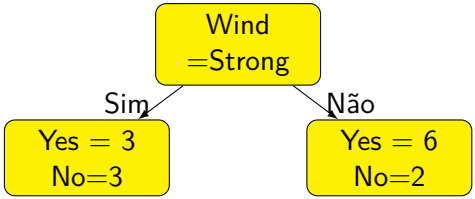
2. Calculando índice de impureza Gini para o nó `[Wind=Strong Não]` e `[Wind=Strong Sim]`:

$$Gini(\text{W=S Não}) = 1 - \left(\frac{6}{8}\right)^2 - \left(\frac{2}{8}\right)^2 = 0.375$$

$$Gini(\text{W=S Sim}) = 1 - \left(\frac{3}{6}\right)^2 - \left(\frac{3}{6}\right)^2 = 0.5$$

Exercício

1. Calcule o índice de impureza Gini para a variável `Wind`.

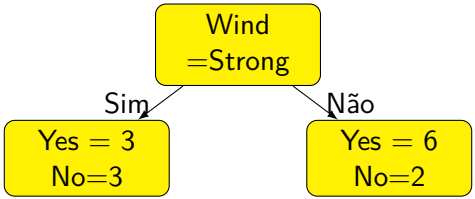


2. Calculando índice de impureza Gini para o nó `[Wind=Strong Não]` e `[Wind=Strong Sim]`:

$$Gini(\text{W=S Não}) = 1 - \left(\frac{6}{8}\right)^2 - \left(\frac{2}{8}\right)^2 = 0.375$$

$$Gini(\text{W=S Sim}) = 1 - \left(\frac{3}{6}\right)^2 - \left(\frac{3}{6}\right)^2 = 0.5$$

Exercício



- Por fim, calculamos o índice de impureza Gini para o nó `[Wind=Strong]`

$$\begin{aligned} Gini(\text{Wind=Strong}) &= 0.375 \cdot \frac{8}{14} + 0.5 \cdot \frac{6}{14} \\ &= \mathbf{0.4285} \end{aligned}$$

Exemplo

- Agora, devemos passar para a variável `Outlook`. Observe que ela possui três fatores:

$$\texttt{Outlook} = \begin{cases} 4 & \texttt{Overcast} \\ 5 & \texttt{Rain} \\ 5 & \texttt{Sunny} \end{cases}$$

Exemplo

- Nesse caso, vamos ter que calcular o índice Gini para todas combinações possíveis:

$$\texttt{Outlook} = \texttt{Overcast} \begin{cases} \texttt{Sim} \\ \texttt{Não} \end{cases}$$

$$\texttt{Outlook} = \texttt{Rain} \begin{cases} \texttt{Sim} \\ \texttt{Não} \end{cases}$$

$$\texttt{Outlook} = \texttt{Sunny} \begin{cases} \texttt{Sim} \\ \texttt{Não} \end{cases}$$

$$\texttt{Outlook} = \texttt{Overcast ou Rain} \begin{cases} \texttt{Sim} \\ \texttt{Não} \end{cases}$$

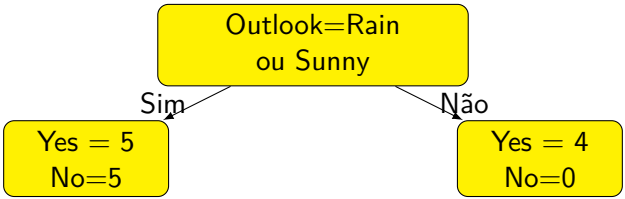
$$\texttt{Outlook} = \texttt{Overcast ou Sunny} \begin{cases} \texttt{Sim} \\ \texttt{Não} \end{cases}$$

$$\texttt{Outlook} = \texttt{Rain ou Sunny} \begin{cases} \texttt{Sim} \\ \texttt{Não} \end{cases}$$

Exemplo

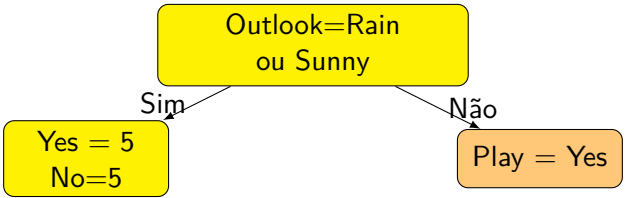
- OBS: Como temos que `[Outlook = Overcast Não] = [Outlook = Rain ou Sunny]`
  conseguimos economizar algumas contas.
- Conforme o número de fatores, o processo se torna mais lento.
- Fazendo todas as contas, para todas as variáveis, vamos obter que `[Outlook=Rain ou Sunny]` tem a menor impureza. Então, ficará no topo da árvore.

Exemplo



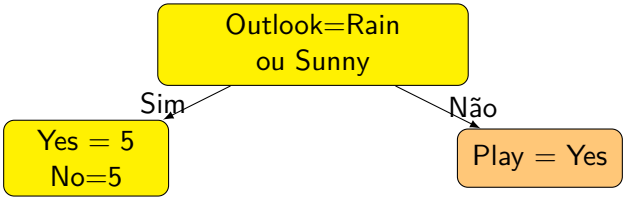
- Observe que à direita o índice Gini do nó é ZERO, ou seja, não há nada mais puro que isso. Logo, vamos atribuir àquele nó o status `Play=Yes`.

Exemplo



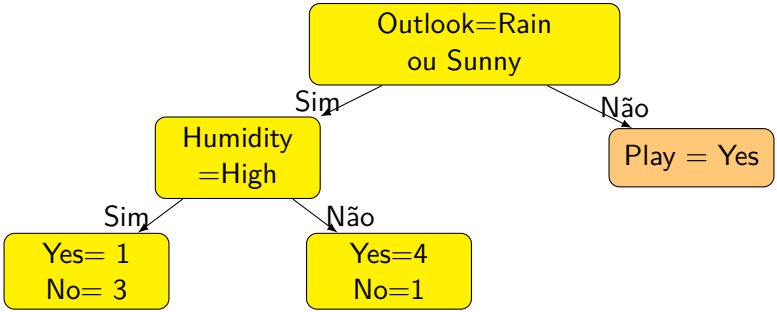
- Observe que à direita o índice Gini do nó é ZERO, ou seja, não há nada mais puro que isso. Logo, vamos atribuir àquele nó o status `Play=Yes`.

Exemplo



- No lado esquerdo, calculamos o índice Gini de todas as variáveis para os indivíduos `[Outlook=Rain ou Sunny Sim]`. Vamos obter que o menor índice de impureza Gini será é do `[Humidity=High]`.

Exemplo



- No lado esquerdo, calculamos o índice Gini de todas as variáveis para os indivíduos `[Outlook=Rain ou Sunny Sim]`. Vamos obter que o menor índice de impureza Gini é do `[Humidity=High]`.

Árvores com rpart()

- O processo de construção pode terminar
  1. Quando a pureza do nó é maior do que o de qualquer variável que adicionamos.
  2. Quando atingimos folhas 100% puras.
  3. Quando o ganho ao aumentar a árvore é muito pequeno.
- Pergunta: quando vale a pena aumentar a árvore? Vamos construir árvores de decisão usando o comando `rpart()`.

```
> library(rpart)
> library(rpart.plot) #Para desenhar a árvore
> playTree<-rpart(Play~.,data=golf)
> rpart.plot(playTree) #Para desenhar a árvore
```

Árvores com `rpart()`

- Observe que a árvore ficou "vazia": assuma **Yes** sempre, e acerte com precisão de 64%.
- Isso ocorre devido aos valores iniciais do comando `rpart.control()`, que ajusta os parâmetros da função `rpart()`.
  `> rpart.control`

rpart.control()

Principais parâmetros:

- **`minsplit:`** o número mínimo de observações que devem existir em um nó para que uma divisão seja tentada. Padrão: minsplit=20.
- **`minbucket:`** o número mínimo de observações em qualquer nó terminal (folhas). Padrão: minbucket=minsplit/3.
- **`cp (complexity parameter):`** O mínimo de ganho de ajuste que devemos ter em cada divisão. O principal papel desse parâmetro é economizar tempo de computação removendo as divisões que obviamente não valem a pena. Padrão: cp=0.01
- **`maxdepth:`** Profundidade máxima da árvore (a profundidade da raiz é zero). Não pode ser maior que 30.

- Exemplo 1

```
> ctrl=rpart.control(minsplit=0)
>playTree<-rpart(Play∼.,data=golf,control=ctrl)
> rpart.plot(playTree)
```

- Exemplo 2

```
> ctrl=rpart.control(minsplit=0,cp=0.1)
> playTree<-rpart(Play∼.,data=golf,control=ctrl)
> rpart.plot(playTree)
```

- Exemplo 3

```
> ctrl=rpart.control(minsplit=0,maxdepth=3)
> playTree<-rpart(Play∼.,data=golf,control=ctrl)
> rpart.plot(playTree)
```