INF623

2024/1

UFV

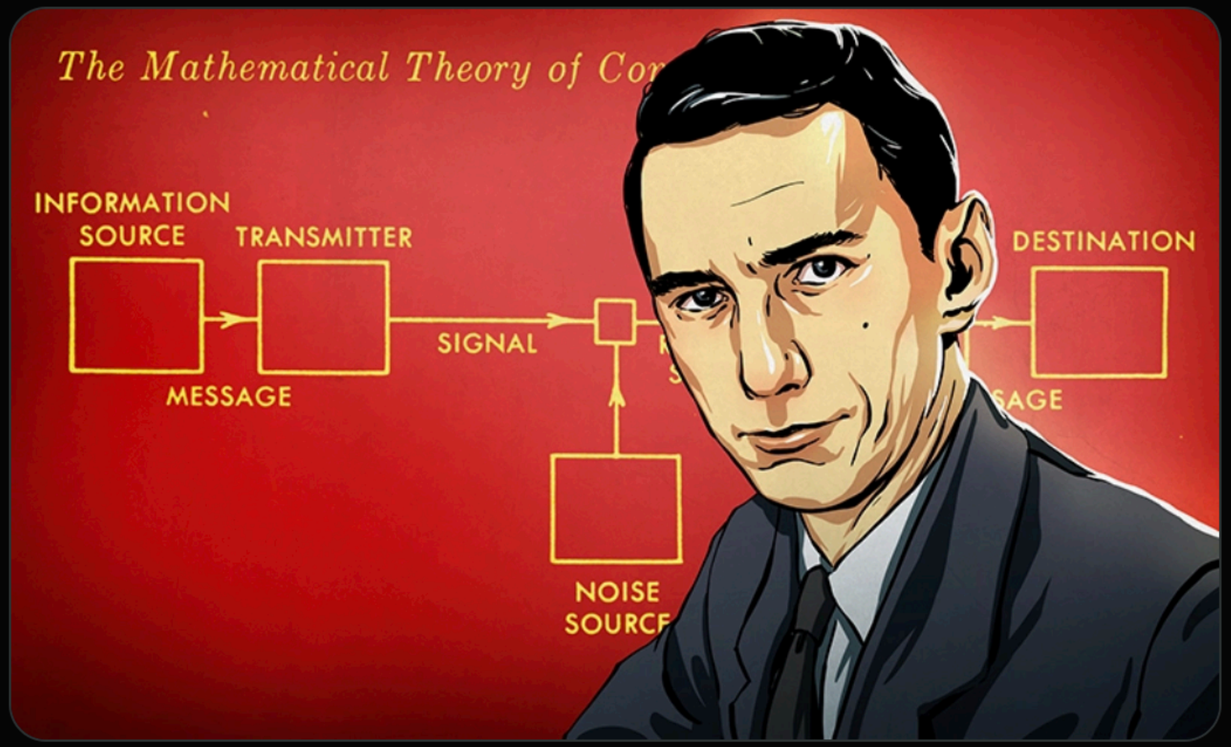
Inteligência Artificial

A26: Aprendizado supervisionado III

MIT CSAIL @MIT_CSAIL · 23h

Over 75 years ago Claude Shannon ushered in the field of information theory with his paper "A Mathematical Theory of Communication", which has been cited over 100,000 times: bit.ly/2H0ZxvR

Image v/@hackaday



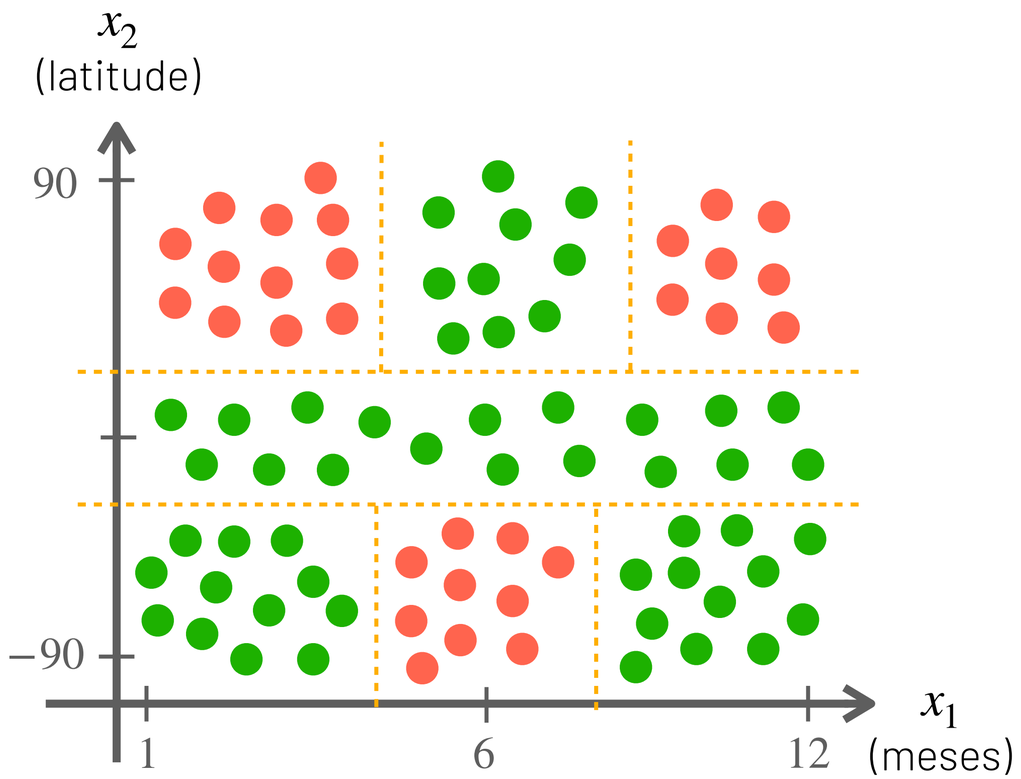
9 265 857 38K

Plano de aula

- Árvore de decisão
- Funções de impureza
  - Gini
  - Entropia
- Implementação
- Características contínuas
- Árvores de regressão
- Problemas comuns com árvores de decisão

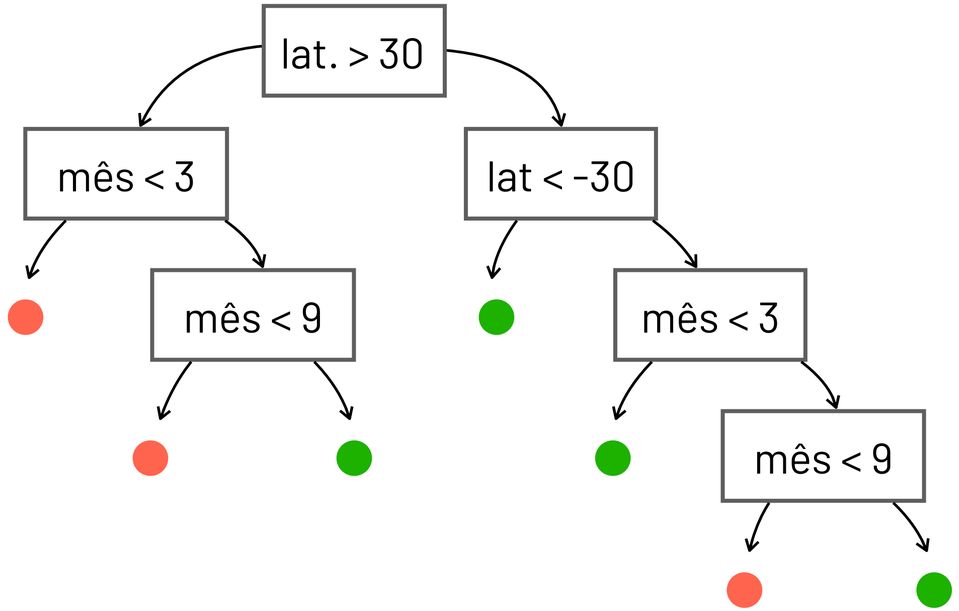
Problemas com o k-NN

O k-NN tem que armazenar o conjunto de dados D inteiro para fazer previsões e isso pode ser um problema (memória e tempo de previsão) quando D cresce.



Como fazer previsões sem armazenar D?

- Armazenar apenas os pontos de corte que separam as classes!



UFV

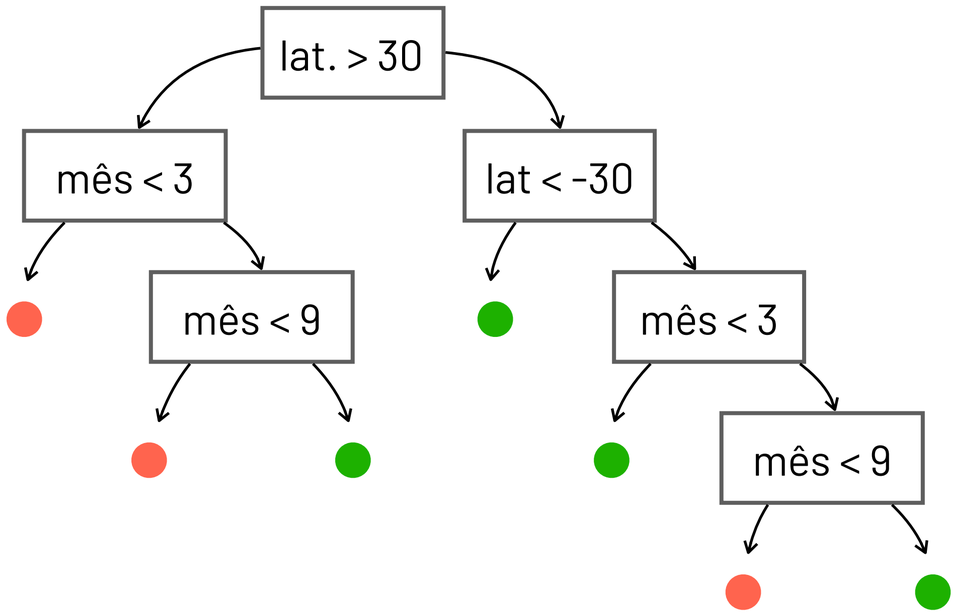
Árvores de decisão

Árvores de decisão são um algoritmo não-paramétrico de classificação e regressão onde o espaço de hipóteses H é definido por árvores:

- Nós representam características;
- Arestas representam os valores das características dos nós onde elas originam.

Objetivo é encontrar uma árvore de decisão:

- Com **menor profundidade** possível;
- Que tenha apenas folhas **puras**
  (i.e. todos os exemplos tenham o mesmo rótulo)



Funções de impureza

Uma função de impureza mede o quão puro um conjunto de dados é em relação aos seus rótulos

- **Impureza Gini**

Seja $D = \{(x_1, y_1), \ldots, (x_n, y_n)\}, y_i \in \{1,...,c\}$, onde c é um número de classes;

e $D_k = \{(x, y) \in D \,|\, y = k\}$, o subconjunto D_k onde os exemplos tem rótulo k

e $p_k = \frac{|D_k|}{|D|}$ a probabilidade de um exemplo qualquer ter o rótulo k em D

A impureza Gini de D é definida por $G(D) = \sum_{1}^{k} p_k(1 - p_k)$

Impureza Gini

Visualização da impureza gini para duas classes:

$$G(D) = \sum_{1}^{k} p_k(1 - p_k)$$

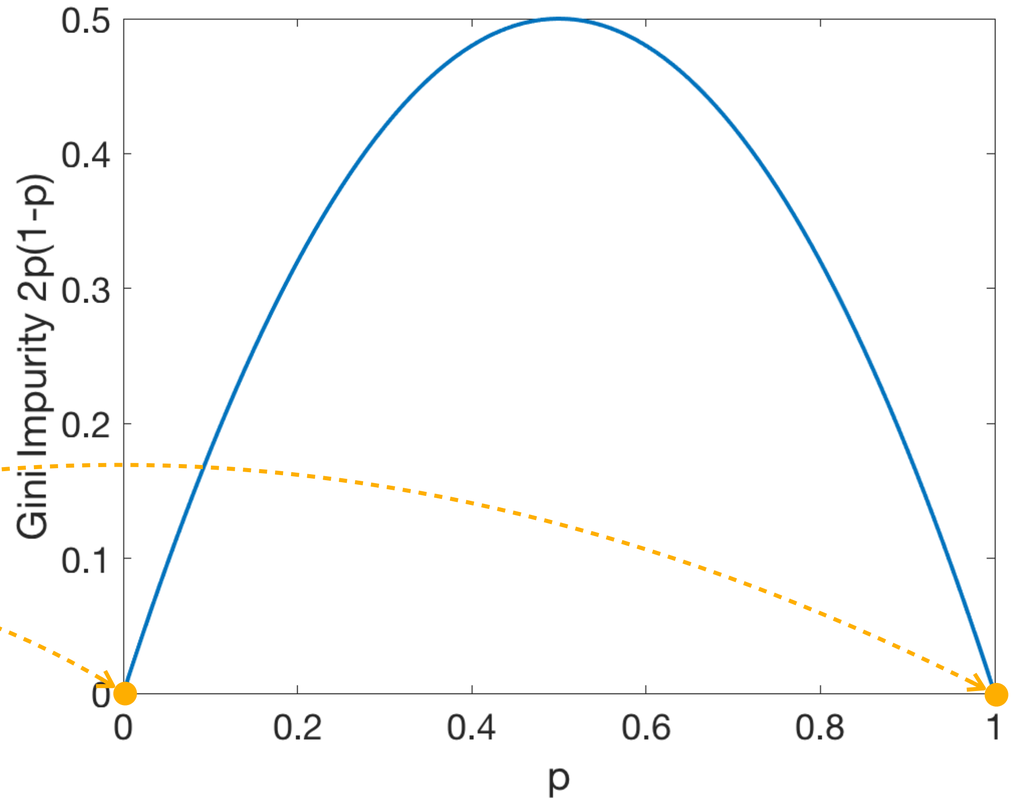
$$G(D) = p_0(1 - p_0) + p_1(1 - p_1)$$

$$G(D) = p_0(1 - p_0) + (1 - p_0)p_0$$

$$G(D) = p(1 - p) + (1 - p)p$$

$$G(D) = 2p(1 - p)$$

Objetivo: $min \sum_{1}^{k} p_k(1 - p_k)$

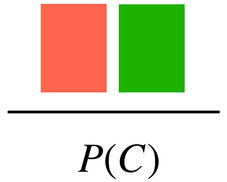


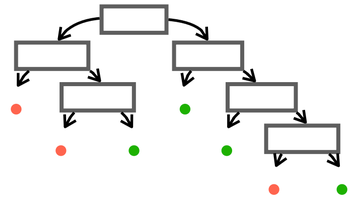
Queremos encontrar árvores onde as folhas tem impureza 0!
(i.e. todos os exemplos tenham o mesmo rótulo 1 ou 0)

Impureza por entropia

Qual é a pior distribuição de rótulos que nós não queremos ter uma folha?

$$P(C) = \{r = 0{,}5,\ g = 0{,}5\}$$





Se tivermos uma folha com uma distribuição qualquer Q, como saber se ela está distante de P?

$$KL(P,Q) = \sum_{1}^{k} p_k log \frac{p_k}{q_k}$$ (divergência de Kullback-leibler)

$$= \sum_{1}^{k} p_k log \frac{p_k}{\frac{1}{|C|}} = \sum_{1}^{k} p_k log(p_k\ |C|) = \sum_{1}^{k} p_k\,(log(p_k) + log(|C|)) = \sum_{1}^{k} p_k\, log(p_k) + p_k\, log(|C|)$$

$$= \sum_{1}^{k} p_k\, log(p_k) + \sum_{1}^{k} p_k\, log(|C|) = \sum_{1}^{k} p_k\, log(p_k) + log(|C|) \cancel{\sum_{1}^{k} p_k} = \sum_{1}^{k} p_k\, log(p_k) + log(|C|)$$

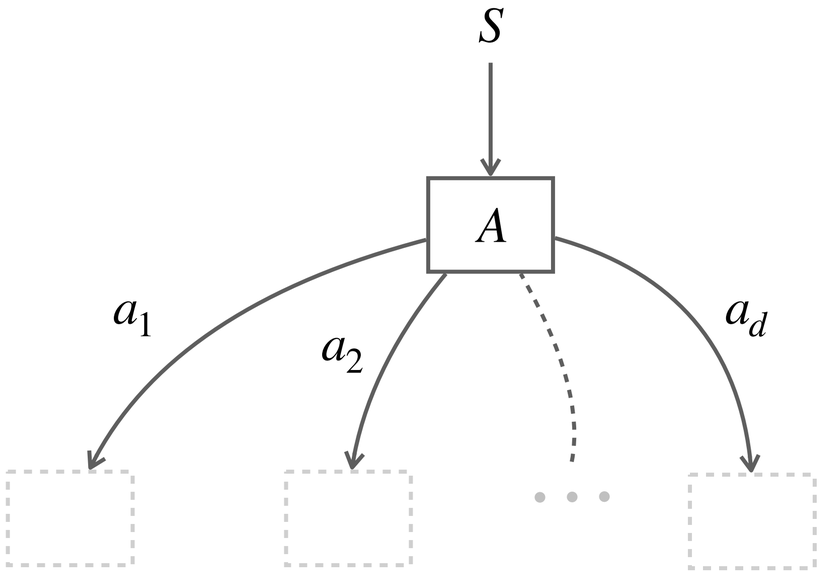
Podemos usar essa distância como métrica de impureza:

$$= max \sum_{1}^{k} p_k\, log(p_k) + \cancel{log(|C|)} = min - \sum_{1}^{k} p_k\, log(p_k)$$

→ Esse valor é chamado de **entropia**!

Entropia de um nó

A entropia $H(A)$ de um nó A é a soma das entropias $H(a_i)$ dos diferentes valores a_i que o atributo A possui no conjunto de dados, ponderadas pelas probabilidades p_{a_i} dos valores a_i:



$$H(A) = \sum_i P_{a_i} H(a_i) = P_{a_1} H(a_1) + P_{a_2} H(a_2) + \ldots + P_{a_d} H(a_d)$$

Onde a entropia $H(a_i)$ de um valor a_i é:

$$H(a_i) = -\sum_1^k p_k \, log(p_k)$$

E a probabilidade p_{a_i} de um valor a_i é:

$$p_{a_i} = \frac{|S_{a_i}|}{|S|}$$

S_{a_i} é o conjunto de dados onde o atributo $A = a_i$ e S é o conjunto de dados sem esse filtro

Exercício 1: calcular entropia

Tempo	Temperatura	Umidade	Vento	Jogar
Ensolarado	Alta	Alta	Fraco	Não
Ensolarado	Alta	Alta	Forte	Não
Nublado	Alta	Alta	Fraco	Sim
Chuvoso	Média	Alta	Fraco	Sim
Chuvoso	Baixa	Normal	Fraco	Sim
Chuvoso	Baixa	Normal	Forte	Não
Nublado	Baixa	Normal	Forte	Sim
Ensolarado	Média	Alta	Fraco	Não
Ensolarado	Baixa	Normal	Fraco	Sim
Chuvoso	Média	Normal	Fraco	Sim
Ensolarado	Média	Normal	Forte	Sim
Nublado	Média	Alta	Forte	Sim
Nublado	Alta	Normal	Fraco	Sim
Chuvoso	Média	Alta	Forte	Não

Esse conjunto de dados contém exemplos de dias adequados para jogar tênis de acordo com as condições climáticas. Qual é a entropia $H(A = Tempo)$?

$$H(A) = \sum_{i} P_{a_i} H(a_i) \quad H(a_i) = -\sum_{1}^{k} p_k \log(p_k) \quad p_{a_i} = \frac{|S_{a_i}|}{|S|}$$

$$H(a_1 = Ensolarado) = -(\frac{2}{5}\ log_2\frac{2}{5}) - (\frac{3}{5}\ log_2\frac{3}{5}) \approx 0{,}97$$

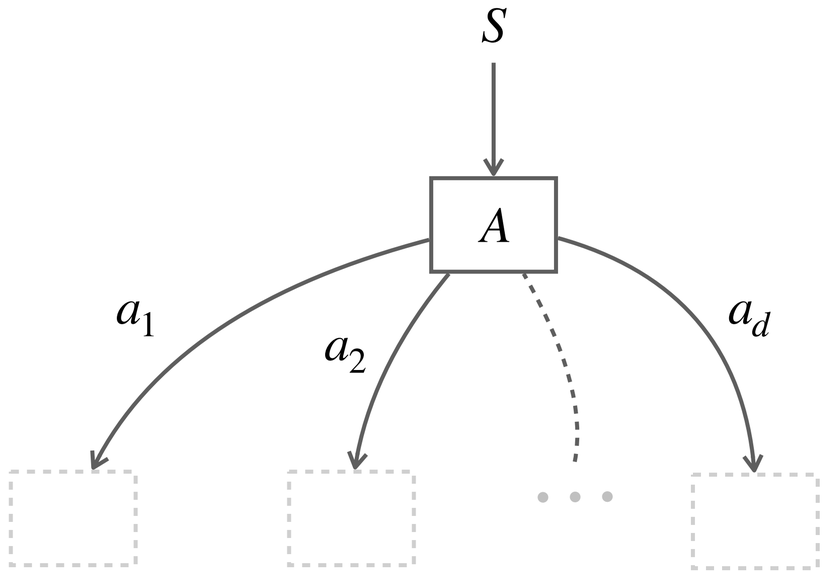
$$H(a_2 = Nublado) = -(\frac{4}{4}\ log_2\frac{4}{4}) - (\frac{0}{4}\ log_2\frac{0}{4}) = 0$$

$$H(a_3 = Chuvoso) = -(\frac{3}{5}\ log_2\frac{3}{5}) - (\frac{2}{5}\ log_2\frac{2}{5}) \approx 0{,}97$$

$$H(A = Tempo) = \frac{5}{14} \times 0{,}97 + \frac{4}{14} \times 0 + \frac{5}{14} \times 0{,}97 = 0{,}69$$

Ganho de informação

Agora que temos uma medida de impureza, qual atributo A escolher para um determinado nó? Queremos escolher o atributo A que maximiza o **ganho de informação**, isto é, a redução esperada da entropia de $H(A)$ na entropia de $H(S)$:



$$H(A) = \sum_i P_{a_i} H(a_i) = P_{a_1} H(a_1) + P_{a_2} H(a_2) + \ldots + P_{a_d} H(a_d)$$

$$H(a_i) = -\sum_1^k p_k \, log(p_k)$$

$$p_{a_i} = \frac{|S_{a_i}|}{|S|}$$

Ganho de informação de A:

$$G(A) = H(S) - H(A)$$

Exercício 2: calcular ganho de informação

Tempo	Temperatura	Umidade	Vento	Jogar
Ensolarado	Alta	Alta	Fraco	Não
Ensolarado	Alta	Alta	Forte	Não
Nublado	Alta	Alta	Fraco	Sim
Chuvoso	Média	Alta	Fraco	Sim
Chuvoso	Baixa	Normal	Fraco	Sim
Chuvoso	Baixa	Normal	Forte	Não
Nublado	Baixa	Normal	Forte	Sim
Ensolarado	Média	Alta	Fraco	Não
Ensolarado	Baixa	Normal	Fraco	Sim
Chuvoso	Média	Normal	Fraco	Sim
Ensolarado	Média	Normal	Forte	Sim
Nublado	Média	Alta	Forte	Sim
Nublado	Alta	Normal	Fraco	Sim
Chuvoso	Média	Alta	Forte	Não

Esse conjunto de dados contém exemplos de dias adequados para jogar tênis de acordo com as condições climáticas. Qual é o ganho de informação $G(A = Tempo)$?

$$G(A = Tempo) = H(S) - H(A = Tempo)$$

$$H(A = Tempo) = \frac{5}{14} \times 0{,}97 + \frac{4}{14} \times 0 + \frac{5}{14} \times 0{,}97 = 0{,}69$$

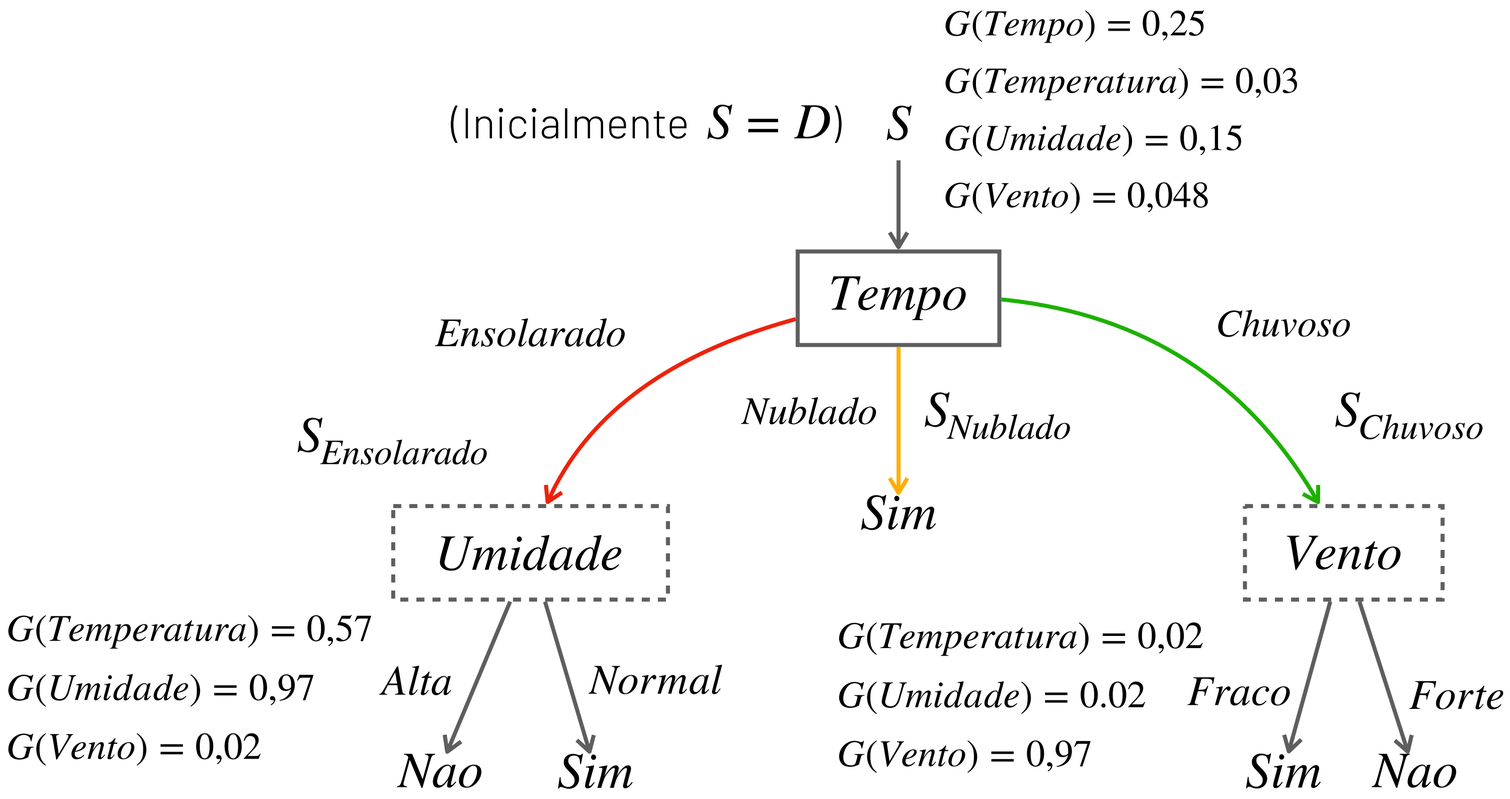
$$H(S) = - p_{sim} log(p_{sim}) - p_{nao} log(p_{nao})$$

$$H(S) = - (\frac{9}{14} \ log \frac{9}{14}) - (\frac{5}{14} \ log \frac{5}{14}) \approx 0{,}94$$

$$G(A = Tempo) = 0{,}94 - 0{,}69 = 0{,}25$$

Aprendendo árvores de decisão

Construir a menor árvore de decisão com folhas puras é um problema NP-difícil. No entanto, podemos aproximar uma boa solução com um algoritmo recursivo e guloso que, a cada passo da recursão, cria um nó com o atributo A que divide S maximizando o ganho de informação, até que todos os exemplos tenham a mesma classe ou não existam mais atributos ou exemplos para dividir.



Tempo	Temperatura	Umidade	Vento	Jogar
Ensolarado	Alta	Alta	Fraco	Não
Ensolarado	Alta	Alta	Forte	Não
Nublado	Alta	Alta	Fraco	Sim
Chuvoso	Média	Alta	Fraco	Sim
Chuvoso	Baixa	Normal	Fraco	Sim
Chuvoso	Baixa	Normal	Forte	Não
Nublado	Baixa	Normal	Forte	Sim
Ensolarado	Média	Alta	Fraco	Não
Ensolarado	Baixa	Normal	Fraco	Sim
Chuvoso	Média	Normal	Fraco	Sim
Ensolarado	Média	Normal	Forte	Sim
Nublado	Média	Alta	Forte	Sim
Nublado	Alta	Normal	Fraco	Sim
Chuvoso	Média	Alta	Forte	Não

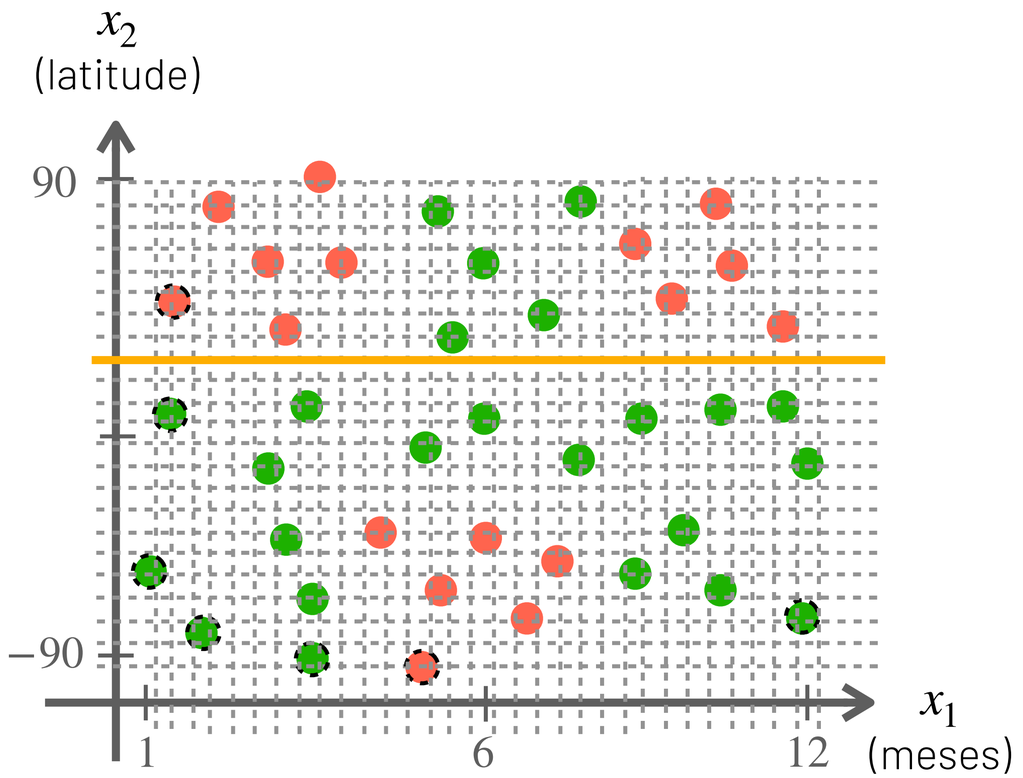
Implementação de árvores de decisão

```
def learn-decision-tree(examples, attributes, parent_examples):
1.  if len(examples) == 0:
2.      return plurality-value(parent_examples)
3.  elif all examples have the same classification:
4.      return the classification
5.  elif len(attributes) == 0:
6.      return plurality-value(examples)
7.  else:
8.      A ← argmax_{a∈attributes} information-gain(a, examples)
9.      tree ← a new decision tree with root test A
10.     for each value v of A do
11.         exs ← {e : e ∈ examples and e.A = v}
12.         subtree ← learn-decision-tree(exs, attributes - A, examples)
13.         add a branch to tree with label (A = v) and subtree subtree
14. return tree
```

Árvores de decisão para características contínuas

Para atributos com valores contínuos o algoritmo é o mesmo, mas com três diferenças:

- O atributo do nó é escolhido:
  1. Ordenando os exemplos para cada atributo
  2. Calculando o ponto médio entre cada dois exemplos consecutivo
  3. O atributo com ponto de corte de maior ganho de informação é escolhido
- Cada nó tem apenas dois filhos < e >=
- O atributo escolhido não é eliminado para o próximo passo recursivo (porque ele pode ser reutilizado com outro ponto de corte)



Demonstração de árvores de decisão

Abrir o seguinte colab:

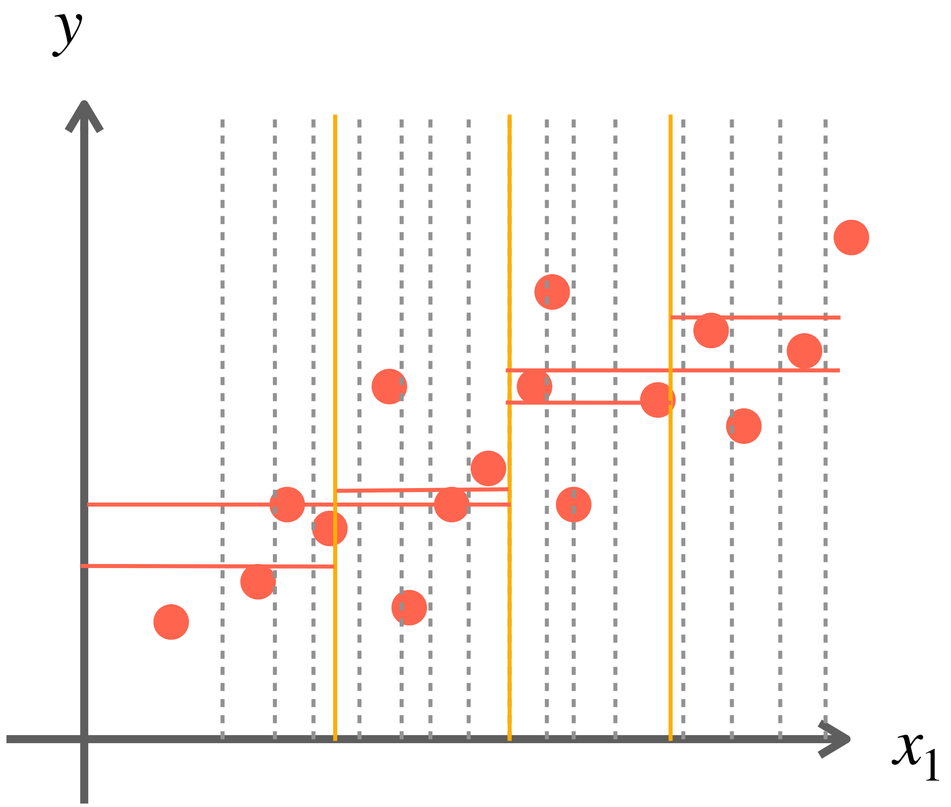
https://colab.research.google.com/drive/10sf18leU7bW0ICDUfyPTqy3Xr_68iviV?usp=sharing

Visualizar:

- Árvores de decisão para dados categóricos
- Árvores de decisão para dados contínuos

Árvores de regressão

Árvores de decisão podem ser utilizadas para regressão, basta alterar a função de impureza:



- Ao invés de ganho de informação, a árvore de regressão usa a redução de variância para determinar a melhor divisão

$$Var(A) = \sum_i P_{a_i} Var(a_i)$$

$$Var(a_i) = \frac{1}{|S_{a_i}|} \sum_{(x,y) \in S_{a_i}} (y - \mu)^2$$

$$p_{a_i} = \frac{|S_{a_i}|}{|S|}$$

$$G(A) = Var(S) - Var(A)$$

- **O rótulo y é previsto** como a média dos exemplos de um nó folha

Problemas com árvores de decisão

Árvores de decisão são algoritmos bastante flexíveis e interpretáveis, mas têm os seguintes problemas:

- Altamente sensíveis aos dados de treinamento. Pequenas variações nos dados podem resultar em árvores completamente diferentes.
- Tendem a criar modelos complexos que se ajustam muito bem aos dados de treinamento (sobreajuste)
- Dificuldade em capturar relações lineares simples entre variáveis, pois elas são baseadas em divisões ortogonais.

Próxima aula

A26: Aprendizado supervisionado IV

Classificadores lineares