Decision Tree

Definição

Uma **árvore de decisão** é um modelo preditivo que divide o espaço de decisão em regiões menores a partir de **perguntas hierárquicas** (nós de decisão).

- Cada **nó interno** representa uma condição (teste sobre um atributo).
- Cada **ramo** representa um resultado do teste.
- Cada **nó folha** representa uma classe (em classificação) ou um valor (em regressão).

2. Fundamentos Matemáticos

2.1. Critérios de Impureza

A escolha de qual atributo usar em cada divisão é guiada por medidas de **pureza** ou **impureza**.

- **Entropia (ID3, C4.5, C5.0 – J. Quinlan):**

$$H(S) = -\sum_{i=1}^{k} p_i \log_2(p_i)$$

Quanto menor a entropia, mais "puro" o nó.

onde p_i é a proporção de exemplos da classe i.

- **Índice de Gini (CART – Breiman):**

$$G(S) = 1 - \sum_{i=1}^{k} p_i^2$$

> Mede a probabilidade de um exemplo ser classificado erroneamente se fosse escolhido aleatoriamente conforme a distribuição do nó.

- **Ganho de Informação (Information Gain):**

$$IG(S, A) = H(S) - \sum_{v \in Valores(A)} \frac{|S_v|}{|S|} H(S_v)$$

> Quanto maior o ganho de informação, melhor o atributo para dividir.

- **Gain Ratio** (C4.5): Normaliza o ganho de informação para evitar viés em atributos com muitos valores distintos.

Algoritmos Clássicos de Indução de Árvores

- **ID3 (1986):** usa ganho de informação.
- **C4.5 (1993):** evolução do ID3, lida com atributos contínuos, valores faltantes e usa **gain ratio**.
- **CART (1984):** usa índice de Gini, gera árvores binárias, suporta classificação e regressão.

Overfitting e Generalização

Árvores podem se tornar **muito profundas**, ajustando-se demais ao conjunto de treino → **overfitting**.

Soluções:

- **Poda (Pruning):** cortar ramos que não trazem ganho significativo.
  - *Pré-poda*: limita profundidade, nº mínimo de exemplos por nó, etc.
  - *Pós-poda*: constrói árvore completa e depois remove ramos irrelevantes.
- **Regularização em sklearn (DecisionTreeClassifier):**
  - `max_depth`
  - `min_samples_split`
  - `min_samples_leaf`
  - `max_features`

Vantagens:

- Fácil de interpretar e visualizar.
- Não exige normalização dos dados.
- Capaz de lidar com atributos categóricos e contínuos.
- Explicável → útil em domínios regulados (saúde, finanças).

Desvantagens:

- Instáveis (pequena mudança nos dados → grande mudança na árvore).
- Tendência ao overfitting.
- Árvores muito profundas são difíceis de interpretar.
- Baixo desempenho comparado a modelos mais complexos.

Árvores de Decisão vs Ensemble

- Árvores **isoladas** = alto viés ou alta variância.
- **Random Forest** (Bagging de árvores): reduz variância, aumenta estabilidade.
- **Gradient Boosting**: combina árvores sequencialmente corrigindo erros anteriores.

Aplicações Comuns

- **Classificação**: detecção de fraude, diagnóstico médico, decisão de crédito.
- **Regressão**: previsão de preços de imóveis, tempo de vida útil de máquinas.
- **Extração de Regras**: análise interpretável de modelos de ML.

Resumo visual mental:

- Raiz = pergunta mais discriminativa.
- Nós = critérios sucessivos.
- Folhas = previsões finais.
- Crescimento da árvore = "dividir para conquistar" até que os dados sejam "quase puros".

Cálculo de Entropia e Ganho de Informação

Dataset de Treino (simplificado – 14 instâncias)

Problema clássico: **"Jogar Tênis?"** (Tomado de Quinlan – ID3).

Dia	Clima (Outlook)	Temperatura	Umidade	Vento	Jogar?
1	Ensolarado	Quente	Alta	Fraco	Não
2	Ensolarado	Quente	Alta	Forte	Não
3	Nublado	Quente	Alta	Fraco	Sim
4	Chuvoso	Ameno	Alta	Fraco	Sim
...	...	...	...	...	...
14	Chuvoso	Ameno	Alta	Forte	Não

Classe alvo: Jogar (Sim/Não).

Temos 9 "Sim" e 5 "Não".

Cálculo de Entropia e Ganho de Informação

Passo 1 – Entropia inicial (do conjunto completo)

$$H(S) = -p_{sim}\log_2(p_{sim}) - p_{não}\log_2(p_{não})$$

$$p_{sim} = \frac{9}{14}, \quad p_{não} = \frac{5}{14}$$

$$H(S) = -\frac{9}{14}\log_2\left(\frac{9}{14}\right) - \frac{5}{14}\log_2\left(\frac{5}{14}\right)$$

$$H(S) \approx 0.94$$

Cálculo de Entropia e Ganho de Informação

Passo 2 – Escolher atributo para dividir

Exemplo: Clima (Outlook)

- Valores possíveis: {Ensolarado, Nublado, Chuvoso}.
- Calculamos a entropia para cada subconjunto:

1. **Clima = Ensolarado**
   - 2 "Sim", 3 "Não".

$$H(S_{ens}) = -\frac{2}{5}\log_2\frac{2}{5} - \frac{3}{5}\log_2\frac{3}{5}$$

$$H(S_{ens}) \approx 0.97$$

2. **Clima = Nublado**
   - 4 "Sim", 0 "Não".

$$H(S_{nub}) = 0$$

   (conjunto puro)

3. **Clima = Chuvoso**
   - 3 "Sim", 2 "Não".

$$H(S_{chu}) = -\frac{3}{5}\log_2\frac{3}{5} - \frac{2}{5}\log_2\frac{2}{5}$$

$$H(S_{chu}) \approx 0.97$$

Cálculo de Entropia e Ganho de Informação

Passo 3 – Ganho de Informação

$$IG(S, Clima) = H(S) - \sum_{v \in \{ens,nub,chu\}} \frac{|S_v|}{|S|} H(S_v)$$

$$IG(S, Clima) = 0.94 - \left(\frac{5}{14} \cdot 0.97 + \frac{4}{14} \cdot 0 + \frac{5}{14} \cdot 0.97 \right)$$

$$IG(S, Clima) = 0.94 - 0.69 = 0.25$$

Cálculo de Entropia e Ganho de Informação

Passo 4 – Comparar com outros atributos

Você repete o cálculo para **Temperatura, Umidade, Vento**.

- O atributo com **maior IG** é escolhido como **raiz da árvore**.

No caso clássico do dataset “Jogar Tênis”, o primeiro nó escolhido é **Clima (Outlook)**.

Cálculo de Entropia e Ganho de Informação

A **entropia mede incerteza**: 0 = puro, máximo = balanceado.

O **ganho de informação mede a redução da incerteza** ao usar um atributo.

A árvore vai **crescendo de forma gulosa** (greedy), escolhendo o melhor atributo localmente em cada nó.