
INSIGHT
Data Science Laboratory
Federal University of Ceará

PRIMEIROS PASSOS NO MUNDO DE DATA SCIENCE & MACHINE LEARNING

INSIGHT

DEPARTAMENTO DE COMPUTAÇÃO

UNIVERSIDADE FEDERAL DO CEARÁ

AGENDA

1. REVIEW
2. K-NN
3. ÁRVORE DE DECISÃO

1. REVIEW

REVIEW

- Parâmetros?
  - “Concentram” o que foi aprendido a partir dos dados.
- Hiperparâmetros?
  - Definem o “comportamento geral” do modelo e não são aprendidos pelos dados.

2. KNN

“Diga-me com quem tu andas que eu direi quem tu és”,

- vovó feat mamãe, since me entendo por gente

Você já escutou esse ditado?

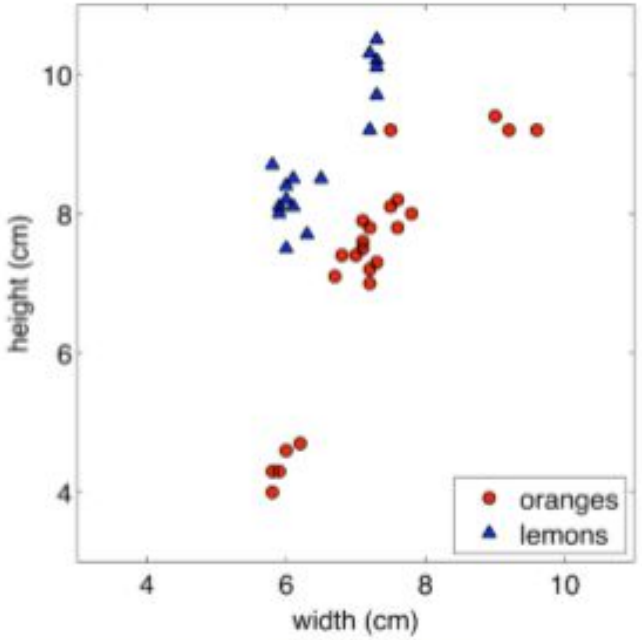
Um modelo pode não ter parâmetros?

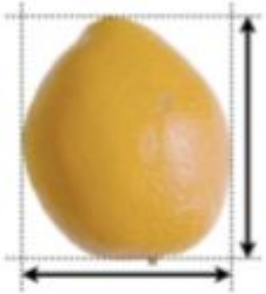
APRENDIZADO BASEADO EM INSTÂNCIAS

- Bem, como diferenciar laranjas de limões?

APRENDIZADO BASEADO EM INSTÂNCIAS

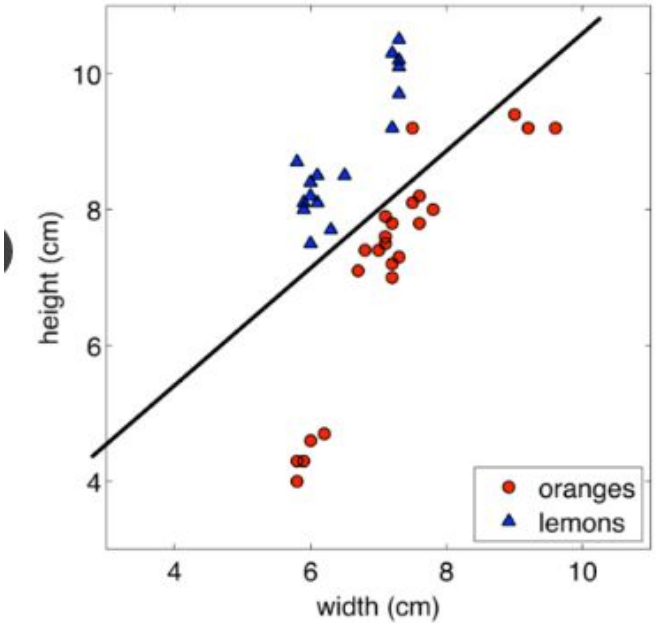
- **IDEIA**: Mapear largura e altura das frutas

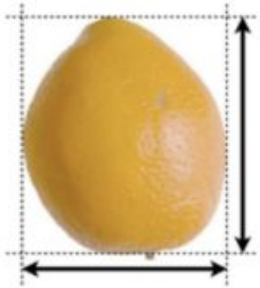




APRENDIZADO BASEADO EM INSTÂNCIAS

- A gente já viu que...

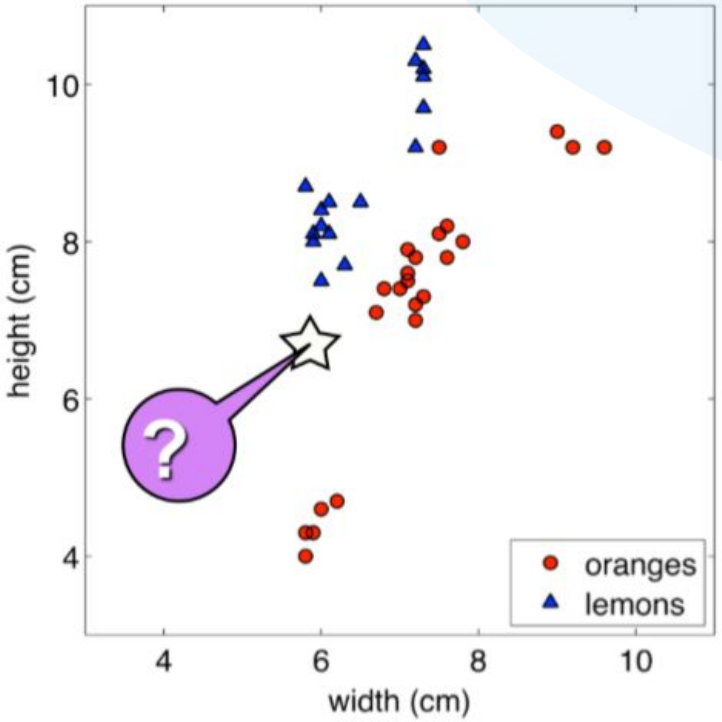




APRENDIZADO BASEADO EM INSTÂNCIAS

E é aqui que jaz a essência do KNN:

- Classificar um novo padrão a partir dos mais próximos!



APRENDIZADO BASEADO EM INSTÂNCIAS

- Modelos **não-paramétricos**
- **Não possuem uma etapa de treinamento**
- Predições são baseadas nas instâncias de treinamento mais próximas do padrão de teste.
- Precisam armazenar os dados de treinamento para realizar predições.

Mas como eu sei quem tá próximo de quem?

DISTÂNCIAS

→ Distância Euclidiana:

$$\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_2 = \sqrt{\sum_{d=1}^{D}(x_{id} - x_{jd})^2}.$$

→ Distância de Manhattan:

$$\|\boldsymbol{x}_i - \boldsymbol{x}_j\|_1 = \sum_{d=1}^{D}|x_{id} - x_{jd}|.$$

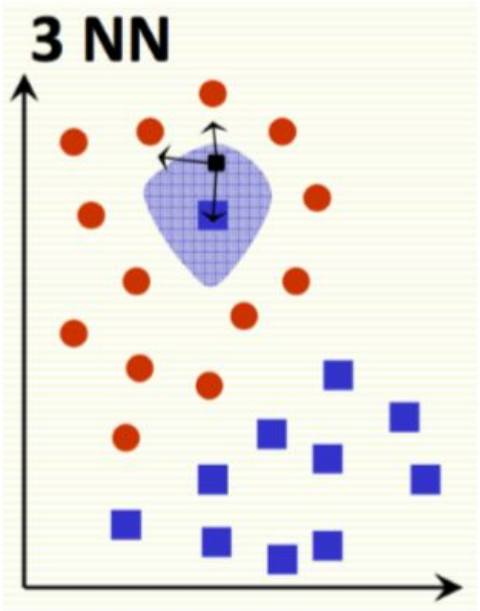
→ Distância de Mahalanobis:

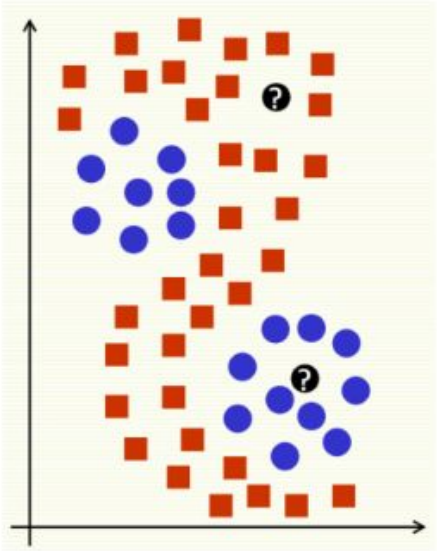
$$d_M(\boldsymbol{x}_i, \boldsymbol{x}_j) = \sqrt{(\boldsymbol{x}_i - \boldsymbol{x}_j)^\top \boldsymbol{\Sigma}^{-1}(\boldsymbol{x}_i - \boldsymbol{x}_j)},$$

em que $\boldsymbol{\Sigma}$ é matriz de covariância dos dados de treinamento.

K-NN: K NEAREST NEIGHBORS

- **IDEIA**: Posso ver a classificação dos vizinhos mais próximos para obter a classificação do novo padrão!





K-NN: K NEAREST NEIGHBORS

K Nearest Neighbors (KNN) para classificação

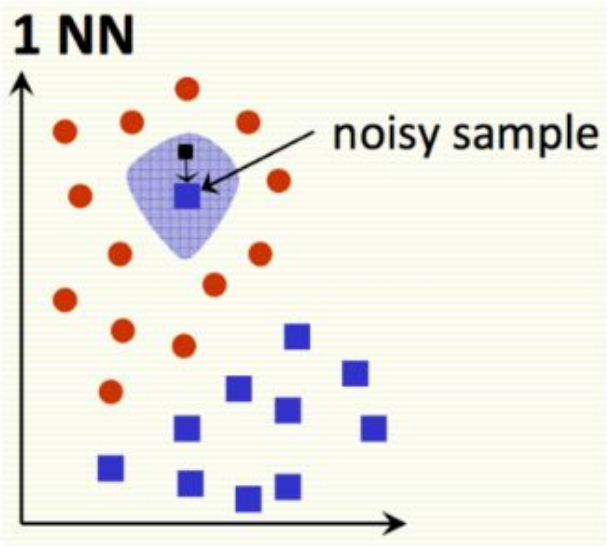
1. Encontre os K padrões $\boldsymbol{x}_k, k \in \{1, \cdots, K\}$ mais próximo do padrão de teste $\boldsymbol{x}_*$:

$$\boldsymbol{x}_{\text{NN}} = \arg \min_{\boldsymbol{x}_i \in \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N\}} d(\boldsymbol{x}_i, \boldsymbol{x}_*).$$

2. Retorne a classe mais comum entre os K padrões encontrados.

K-NN OBSERVAÇÕES

- Valores **muito baixos de K** podem ser sensíveis a ruído e tornam a região de decisão mais complexa.



K-NN OBSERVAÇÕES

- Valores **muito baixos de K** podem ser sensíveis a ruído e tornam a região de decisão mais complexa.
- Valores **muito altos de K** podem incluir informação de dados muito distantes e simplificam a região de decisão.

K-NN OBSERVAÇÕES

- **PROBLEMA**: Se alguns atributos tiverem magnitude muuuuuito maior que outros, eles serão tratados como mais importantes!
  - Normalize os dados!

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

$$x_{scaled} = \frac{x - mean}{sd}$$

K-NN OBSERVAÇÕES

- **PROBLEMA**: Dados com muitos atributos/features
  - Maldição da dimensionalidade
  - Custo computacional aumenta demais!
- Selecionar/combinar os atributos mais relevantes

$$x_{scaled} = \frac{x - x_{min}}{x_{max} - x_{min}}$$

$$x_{scaled} = \frac{x - mean}{sd}$$

GRID SEARCH PARA K-NN

Grid search para valor de K no modelo KNN

1. Separe o conjunto dados em treino, validação e teste;
2. Escolha um valor candidato para K;
3. Calcule o erro do KNN no conjunto de validação usando os dados de treino;
4. Repita os dois passos anteriores para todos os candidatos para K;
5. Escolha o valor de K com menor erro na validação;
6. Forme um novo conjunto de treino a partir do treino e validação anteriores;
7. Calcule o erro nos dados de teste usando o novo conjunto de treinamento e o melhor valor de K encontrado.

REGRESSÃO COM K-NN!

K Nearest Neighbors (KNN) para regressão

1. Encontre os K padrões $\boldsymbol{x}_k, k \in \{1, \cdots, K\}$ mais próximo do padrão de teste $\boldsymbol{x}_*$:

$$\boldsymbol{x}_{\text{KNN}} = \arg \min_{\boldsymbol{x}_i \in \{\boldsymbol{x}_1, \cdots, \boldsymbol{x}_N\}} d(\boldsymbol{x}_i, \boldsymbol{x}_*).$$

2. Retorne uma ponderação das saídas dos K padrões encontrados:

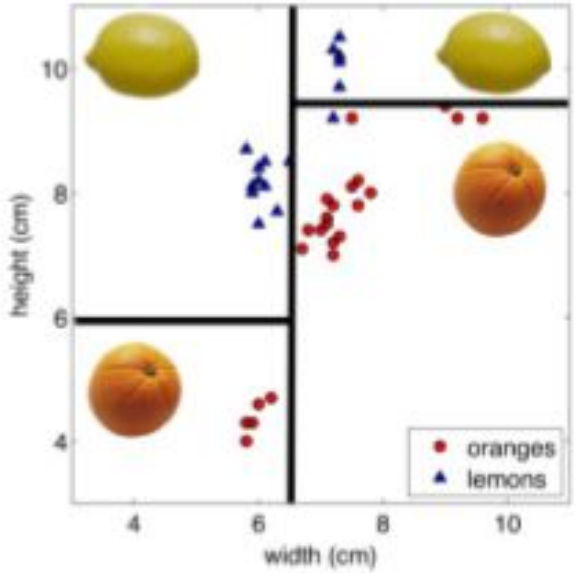
$$y_* = \frac{\sum_{k=1}^{K} \gamma_k y_k}{\sum_{k=1}^{K} \gamma_k}.$$

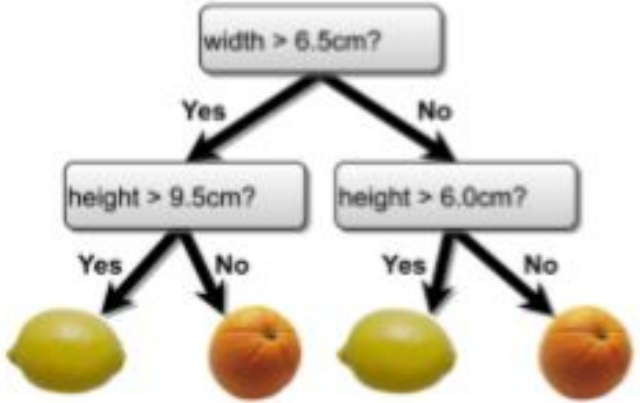
- Abordagens comuns para a ponderação das saídas:
  - → Uniforme: $\gamma_k = 1, \forall k$.
  - → Inversamente proporcional às distâncias: $\gamma_k = \frac{1}{d(\boldsymbol{x}_k, \boldsymbol{x}_*)}, \forall k$.

3. ÁRVORE DE DECISÃO

ÁRVORE DE DECISÃO

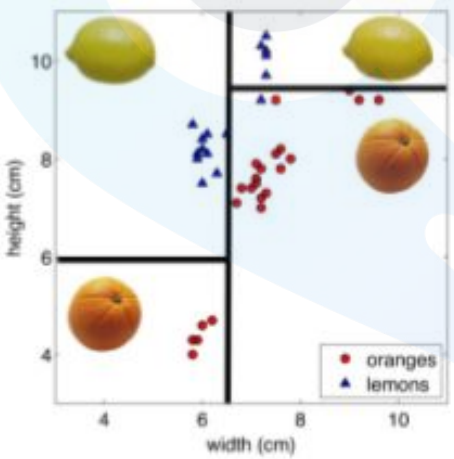
- **IDEIA**: Usar regras lógicas (se-então) para separar as frutas!

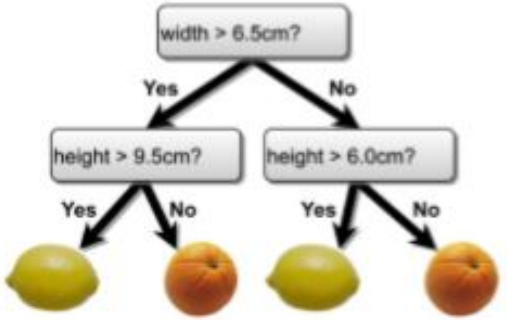




ÁRVORE DE DECISÃO

- Nós internos verificam os valores dos atributos
- Ramificação é feita de acordo com o limiar (threshold) escolhido.
- Nós terminais (folhas) estão associados a uma classe específica.





ÁRVORE DE DECISÃO

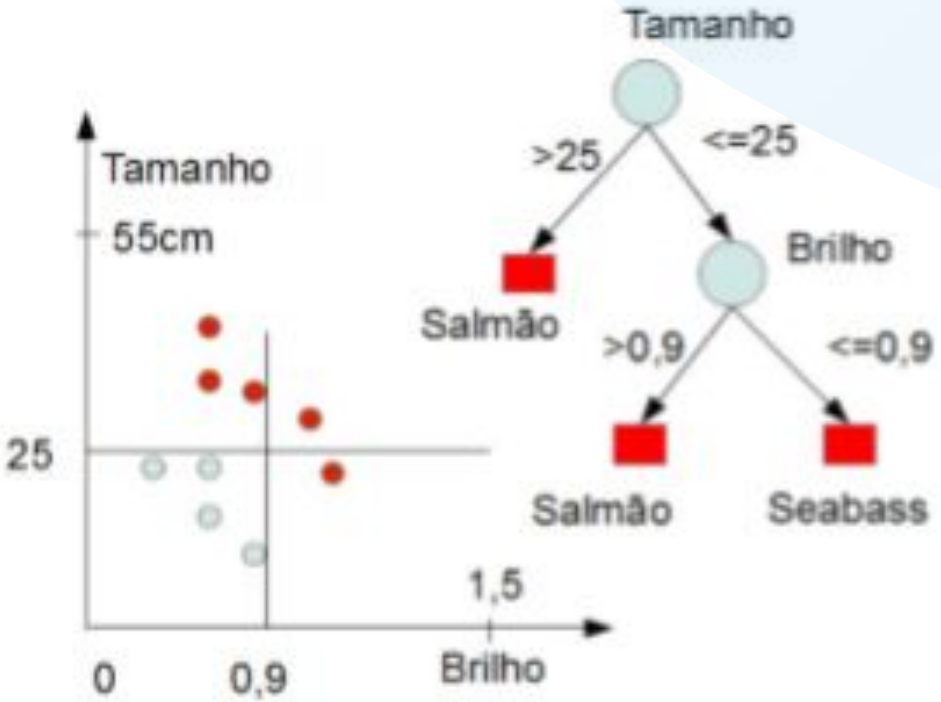
Predições usando árvores de decisão

Dada uma árvore de decisão já existente e um padrão de teste:

1. Inicie no nó mais superior (raiz da árvore).
2. Considere o atributo do nó em questão.
3. Verifique o limiar do nó atual e siga um dos ramos existentes.
4. Caso chegue em um nó terminal (folha), retorne a saída associada. Caso contrário, desça para o próximo nó interno e continue.

ÁRVORE DE DECISÃO

Brilho	Tamanho	Classe
1.2	23	Salmão
1.1	30	Salmão
0.9	36	Salmão
0.8	45	Salmão
0.8	38	Salmão
0.9	15	Seabass
0.8	20	Seabass
0.8	25	Seabass
0.7	25	Seabass



ÁRVORE DE DECISÃO

- **IDEIA**: Usar regras lógicas (se-então) para separar as frutas!
- **Problema:** Como obter a árvore de decisão automaticamente a partir dos dados de treinamento?
- **Problema:** Construir a menor árvore (mais concisa) é um problema NP completo.

ÁRVORE DE DECISÃO

- **Ideia:** Seguir uma abordagem heurística gulosa (greedy):
  - Comece de uma árvore vazia;
  - Encontre o melhor atributo para realizar uma divisão;
  - Repita recursivamente o passo anterior para o próximo nó até encontrar uma folha.

ÁRVORE DE DECISÃO

- **Problema:** Como encontrar o melhor atributo para realizar a divisão?

ÁRVORE DE DECISÃO

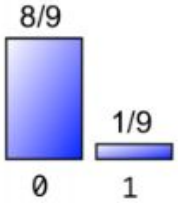
- **Problema:** Como encontrar o melhor atributo para realizar a divisão?
- **Ideia:** Usar índices de pureza.
  - **Pureza máxima:** Somente exemplos de uma mesma classe em uma folha.
  - **Pureza mínima:** Quantidades iguais de todas as classes em uma folha.
  - Distribuições intermediárias devem ser quantificadas por um índice.

ÁRVORE DE DECISÃO

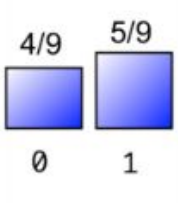
Entropia (teoria da informação)

- Taxa de informação gerada por uma fonte de dados.
- Dados improváveis fornecem mais informação (mais "surpresa").
- Maior a pureza, menor a entropia, sendo quantificada por:

$$H = -\sum_{k} P(C_k) \log_2 P(C_k)$$



$$-\frac{8}{9}\log_2\frac{8}{9} - \frac{1}{9}\log_2\frac{1}{9} \approx 0.5$$



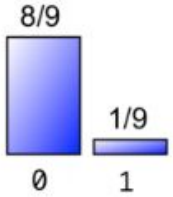
$$-\frac{4}{9}\log_2\frac{4}{9} - \frac{5}{9}\log_2\frac{5}{9} \approx 0.99$$

ÁRVORE DE DECISÃO

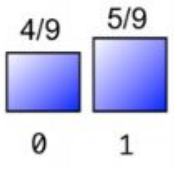
Índice (ou impureza de) Gini

- Frequência em que um exemplo aleatório é incorretamente classificado.
- Pode ser quantificado por:

$$G = \sum_k P(C_k)(1 - P(C_k)) = 1 - \sum_k P(C_k)^2$$



$$1 - \left(\frac{8}{9}\right)^2 - \left(\frac{1}{9}\right)^2 \approx 0.2$$



$$1 - \left(\frac{4}{9}\right)^2 - \left(\frac{5}{9}\right)^2 \approx 0.49$$

ÁRVORE DE DECISÃO

- Vamos aplicar o índice Gini na divisão dos exemplos de peixes.

Brilho	Tamanho	Classe
1.2	23	Salmão
1.1	30	Salmão
0.9	36	Salmão
0.8	45	Salmão
0.8	38	Salmão
0.9	15	Seabass
0.8	20	Seabass
0.8	25	Seabass
0.7	25	Seabass

- Gini original (5 Salmão e 4 Seabass):

$$G = 1 - \left(\frac{5}{9}\right)^2 - \left(\frac{4}{9}\right)^2 \approx 0.49$$

- Escolhendo Brilho > 0.7:
  - $\rightarrow$ 1 Seabass e 0 Salmão: $G_1 = 1 - \left(\frac{1}{1}\right)^2 - \left(\frac{0}{1}\right)^2 = 0$
  - $\rightarrow$ 3 Seabass e 5 Salmão: $G_2 = 1 - \left(\frac{3}{8}\right)^2 - \left(\frac{5}{8}\right)^2 \approx 0.47$
  - $\rightarrow$ Gini médio das ramificações: $G_B = \frac{1}{9} G_1 + \frac{8}{9} G_2 \approx 0.42$

ÁRVORE DE DECISÃO

- Vamos aplicar o índice Gini na divisão dos exemplos de peixes.

Brilho	Tamanho	Classe
1.2	23	Salmão
1.1	30	Salmão
0.9	36	Salmão
0.8	45	Salmão
0.8	38	Salmão
0.9	15	Seabass
0.8	20	Seabass
0.8	25	Seabass
0.7	25	Seabass

- Gini original (5 Salmão e 4 Seabass):

$$G = 1 - \left(\frac{5}{9}\right)^2 - \left(\frac{4}{9}\right)^2 \approx 0.49$$

- Escolhendo Tamanho > 25:
  - $\rightarrow$ 4 Seabass e 1 Salmão: $G_1 = 1 - \left(\frac{4}{5}\right)^2 - \left(\frac{1}{5}\right)^2 \approx 0.32$
  - $\rightarrow$ 0 Seabass e 4 Salmão: $G_2 = 1 - \left(\frac{0}{4}\right)^2 - \left(\frac{4}{4}\right)^2 = 0$
  - $\rightarrow$ Gini médio das ramificações: $G_T = \frac{5}{9}G_1 + \frac{4}{9}G_2 \approx 0.18$

ÁRVORE DE DECISÃO

- Vamos aplicar o índice Gini na divisão dos exemplos de peixes.

Brilho	Tamanho	Classe
1.2	23	Salmão
1.1	30	Salmão
0.9	36	Salmão
0.8	45	Salmão
0.8	38	Salmão
0.9	15	Seabass
0.8	20	Seabass
0.8	25	Seabass
0.7	25	Seabass

- Gini original (5 Salmão e 4 Seabass):

$$G = 1 - \left(\frac{5}{9}\right)^2 - \left(\frac{4}{9}\right)^2 \approx 0.49$$

- Opções de ramificação:

$$\text{Brilho} > 0.7 \rightarrow G_B \approx 0.42$$
$$\text{Tamanho} > 25 \rightarrow G_T \approx 0.18$$

- Escolhemos a opção que apresenta a maior queda de impureza Gini em relação ao nó pai (Tamanho > 25).

ÁRVORE DE DECISÃO

Treinamento guloso (*greedy*) de árvores de decisão

1. Calcule o índice de pureza/impureza do nó atual (nó pai);
2. Crie ramificações a partir de um atributo e um limiar candidatos;
3. Escolha a ramificação com maior queda de impureza (maior pureza) em relação ao nó pai;
4. Para cada nó criado pela ramificação escolhida:
   - Se não houver exemplos de treinamento, retorne a classe mais comum no nó pai.
   - Se todos os exemplos são de uma mesma classe, retorne-a.
   - Caso contrário, retorne ao primeiro passo.

ÁRVORE DE DECISÃO

- **Vantagens**
  - Facilmente interpretáveis, pois geram regras de decisão
  - Seleção automática de atributos importantes
  - Podem lidar com dados faltosos
- **Desvantagens**
  - Tendência ao overfitting
  - Pequenas variações no conjunto de treinamento resultam em árvores diferentes


LOONEY TUNES