

data

# Decision Trees / Árvores de decisão

João Pedro (Dora) Mattos • 17/03/2021

*@joaopedromattos*

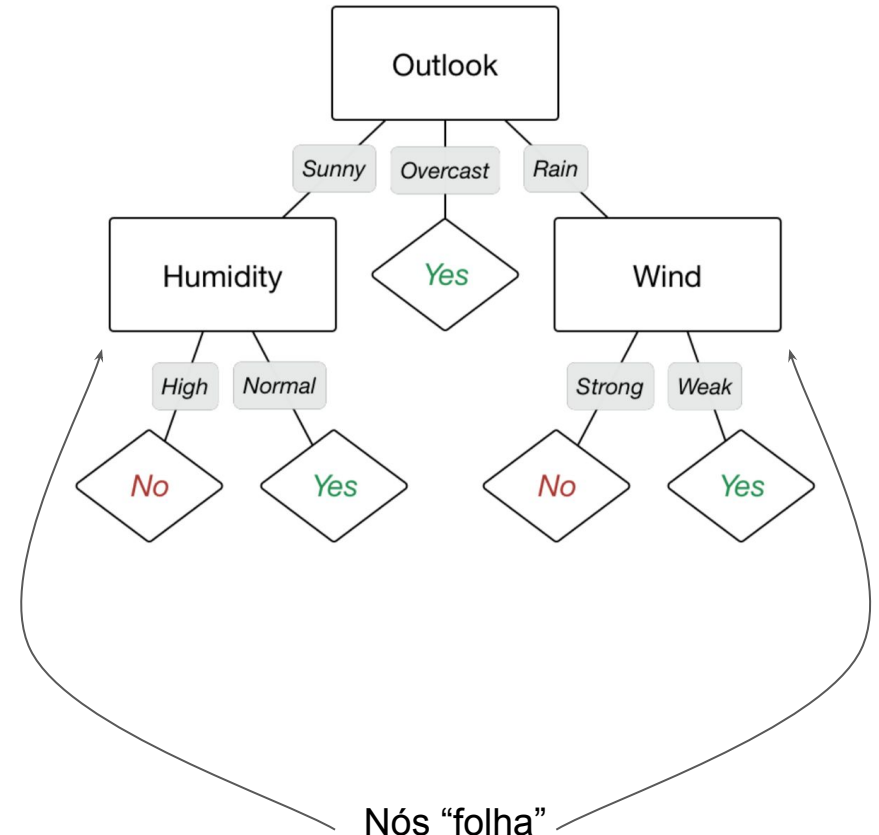


Motivação



Day	Outlook	Temp.	Humidity	Wind	Play?
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

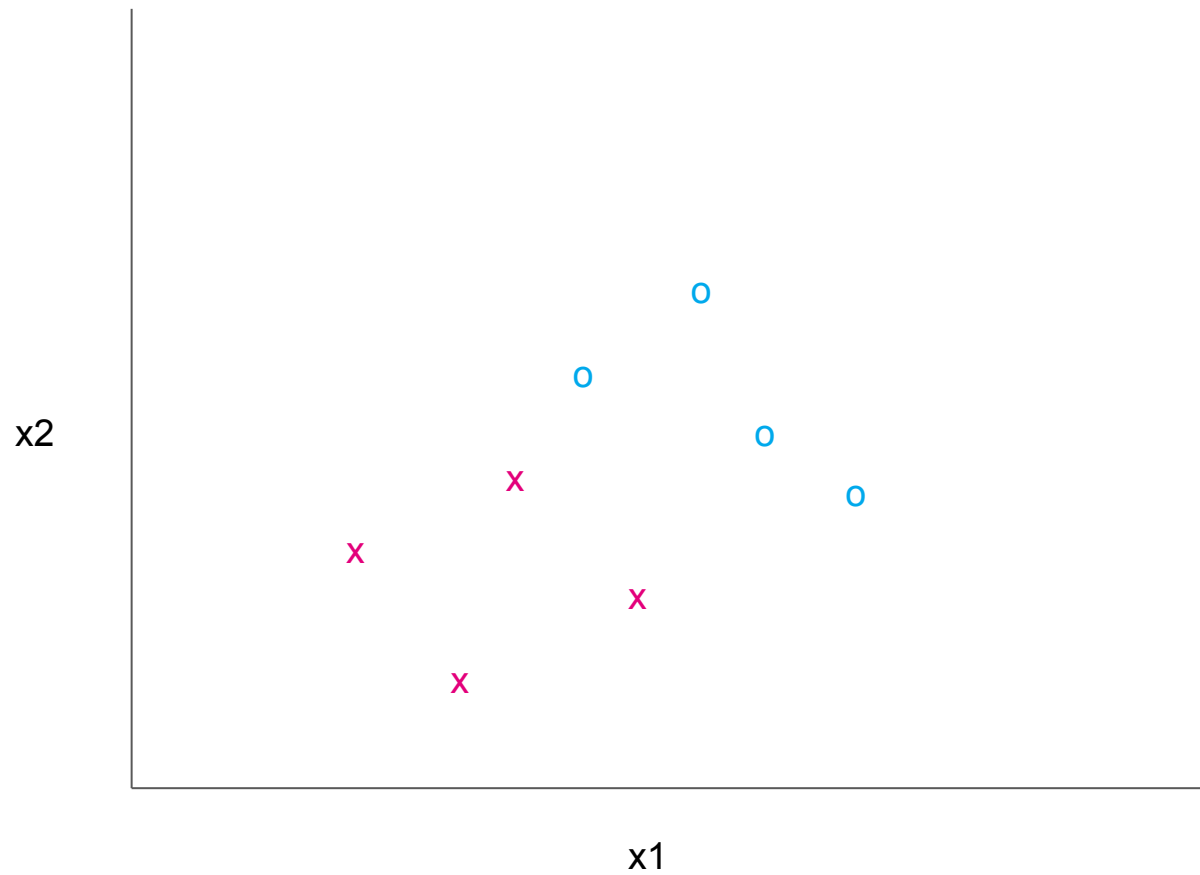
Day	Outlook	Temp.	Humidity	Wind	Play?
1	Sunny	Hot	High	Weak	No
2	Sunny	Hot	High	Strong	No
3	Overcast	Hot	High	Weak	Yes
4	Rain	Mild	High	Weak	Yes
5	Rain	Cool	Normal	Weak	Yes
6	Rain	Cool	Normal	Strong	No
7	Overcast	Cool	Normal	Strong	Yes
8	Sunny	Mild	High	Weak	No
9	Sunny	Cool	Normal	Weak	Yes
10	Rain	Mild	Normal	Weak	Yes
11	Sunny	Mild	Normal	Strong	Yes
12	Overcast	Mild	High	Strong	Yes
13	Overcast	Hot	Normal	Weak	Yes
14	Rain	Mild	High	Strong	No

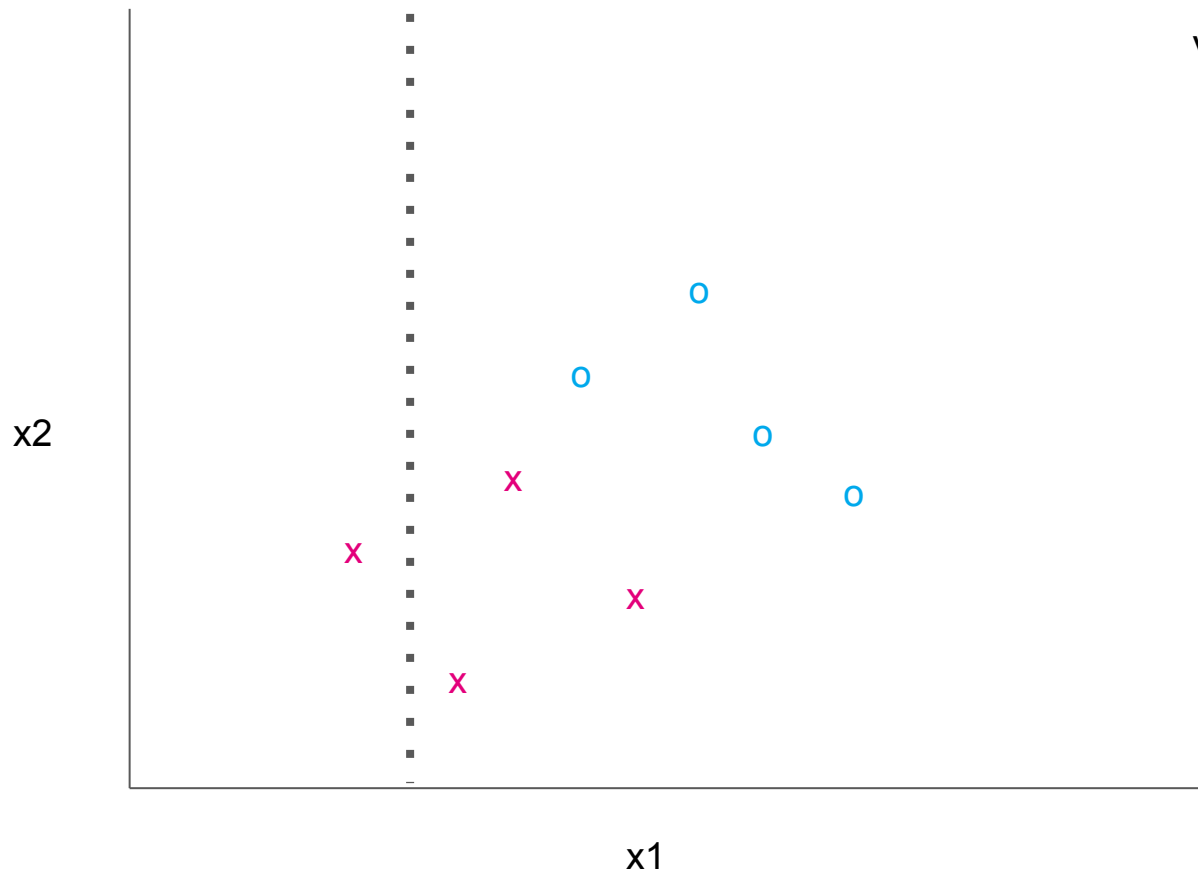




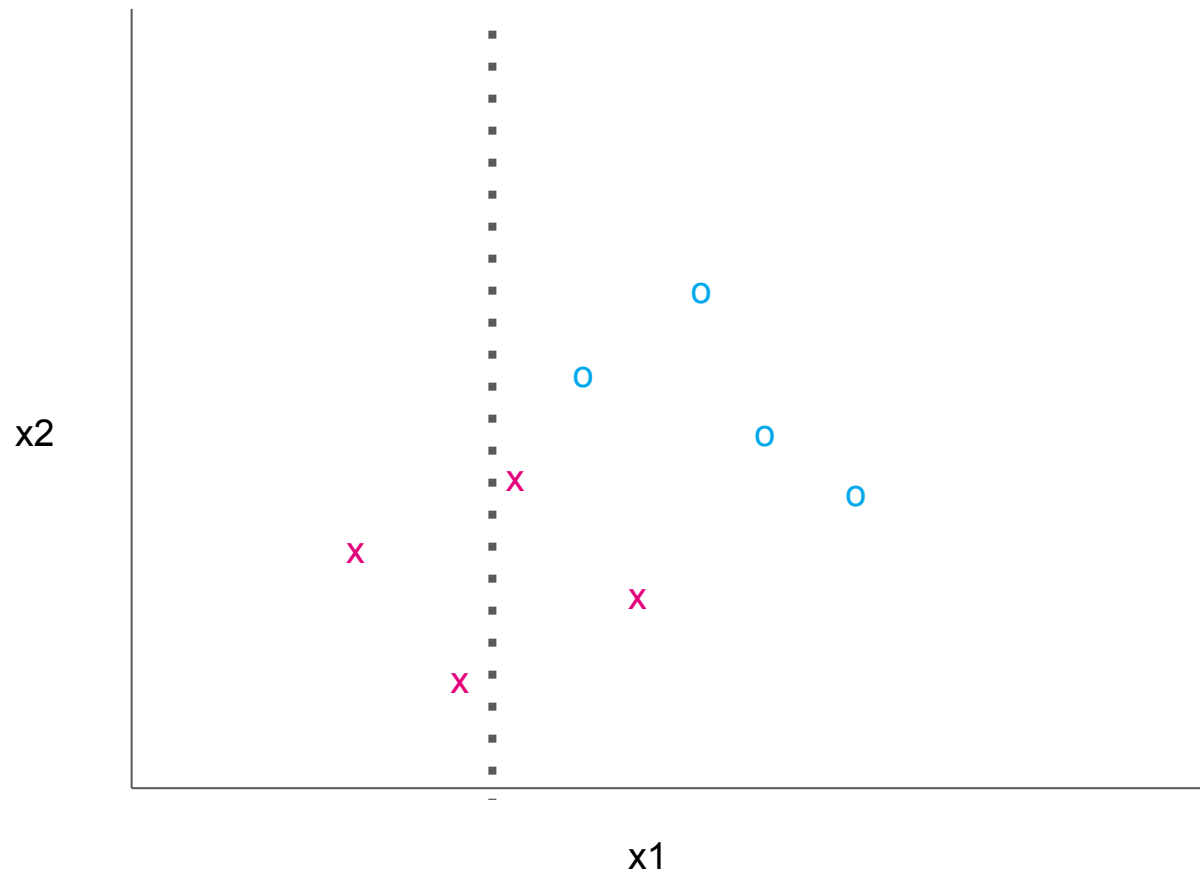
Construindo  
uma árvore



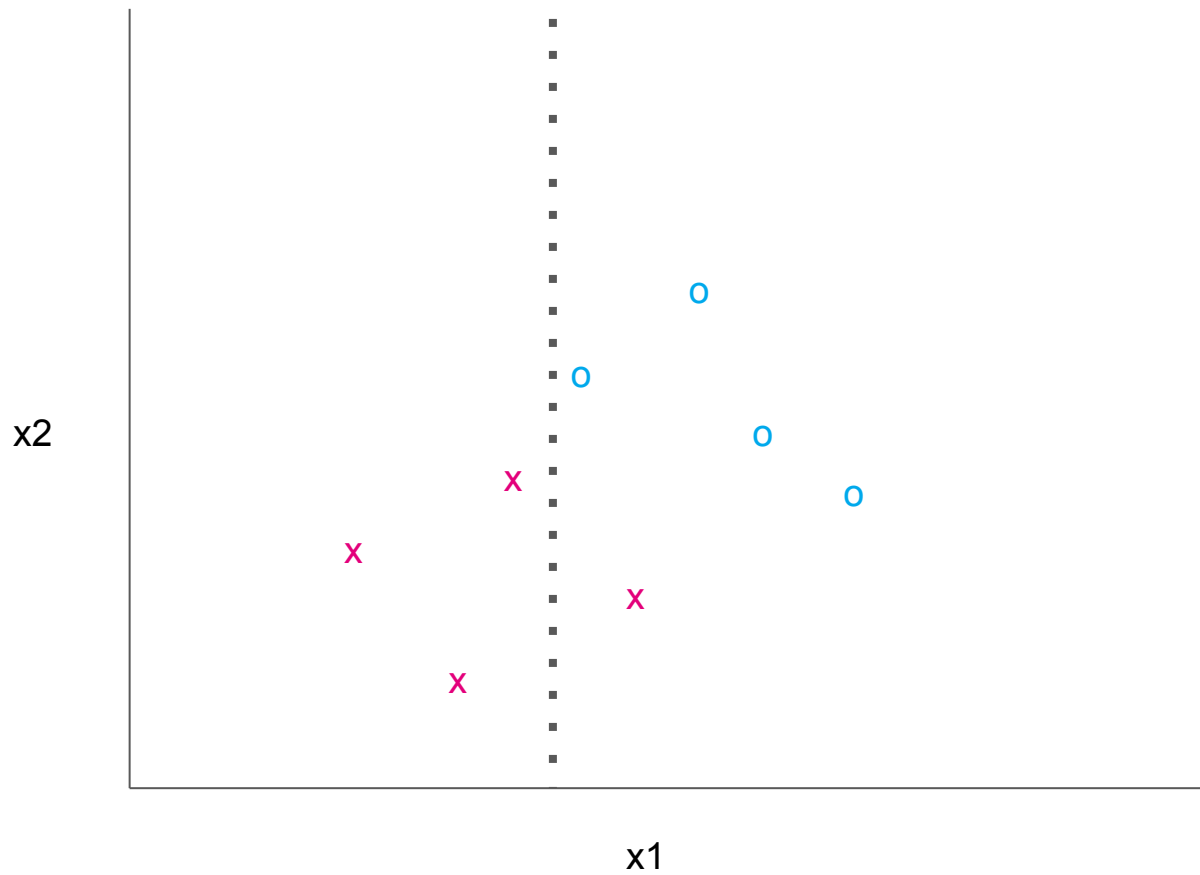


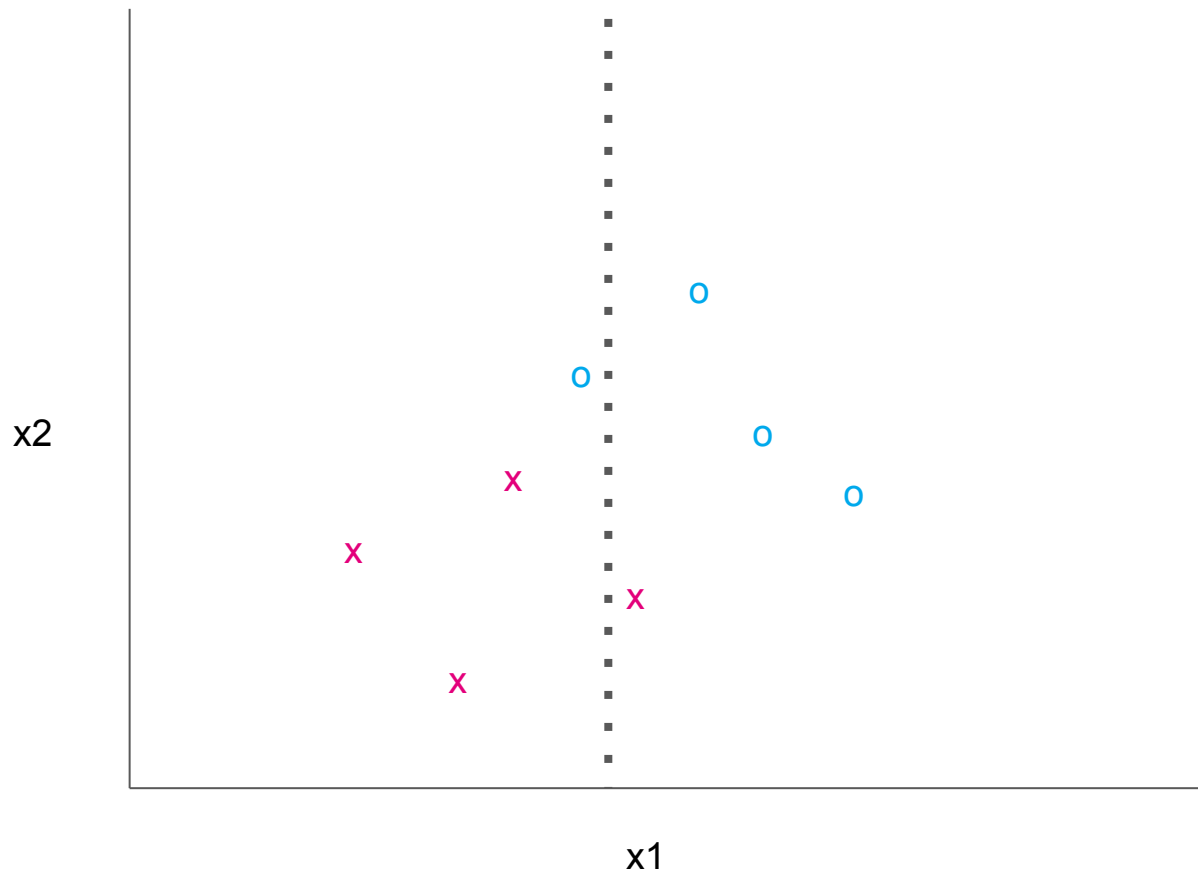


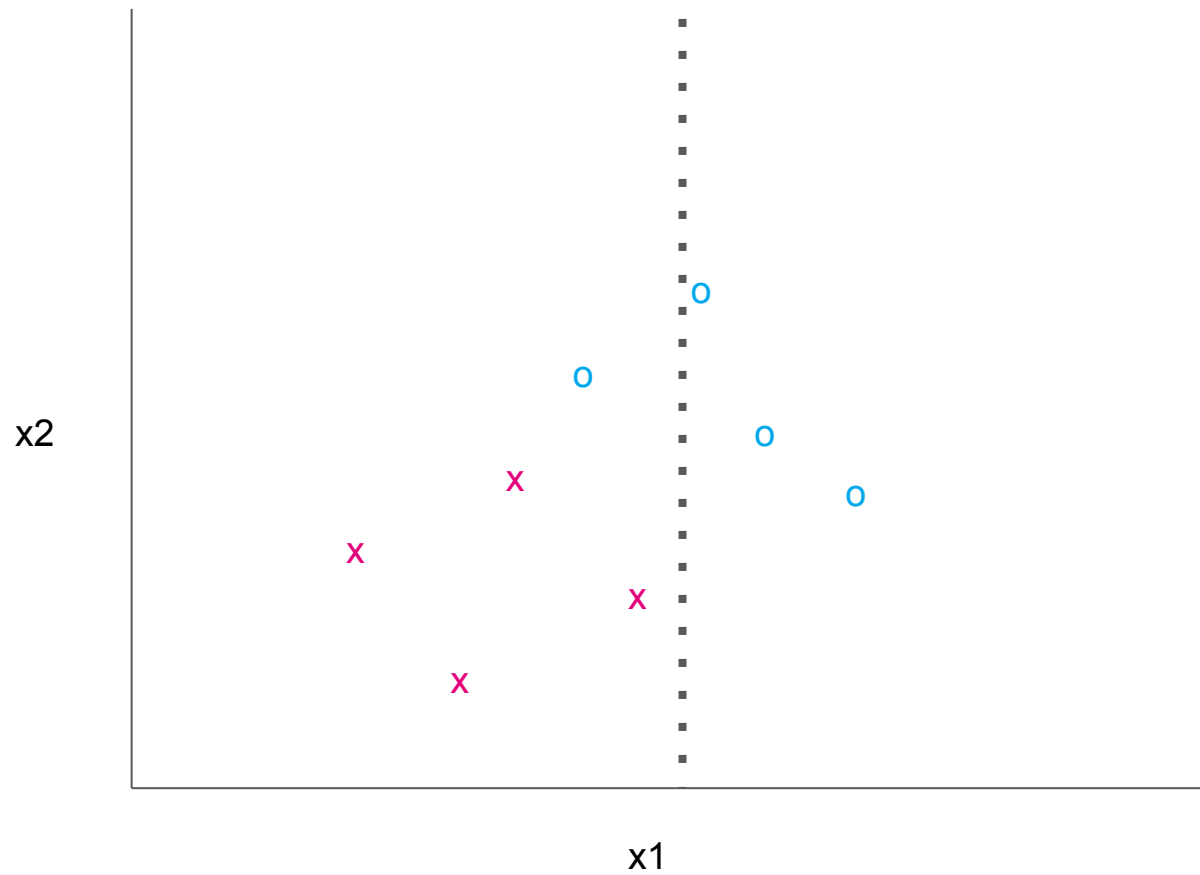
Vamos tentar separar  
nossos dados pela  
*feature*  $x_1$

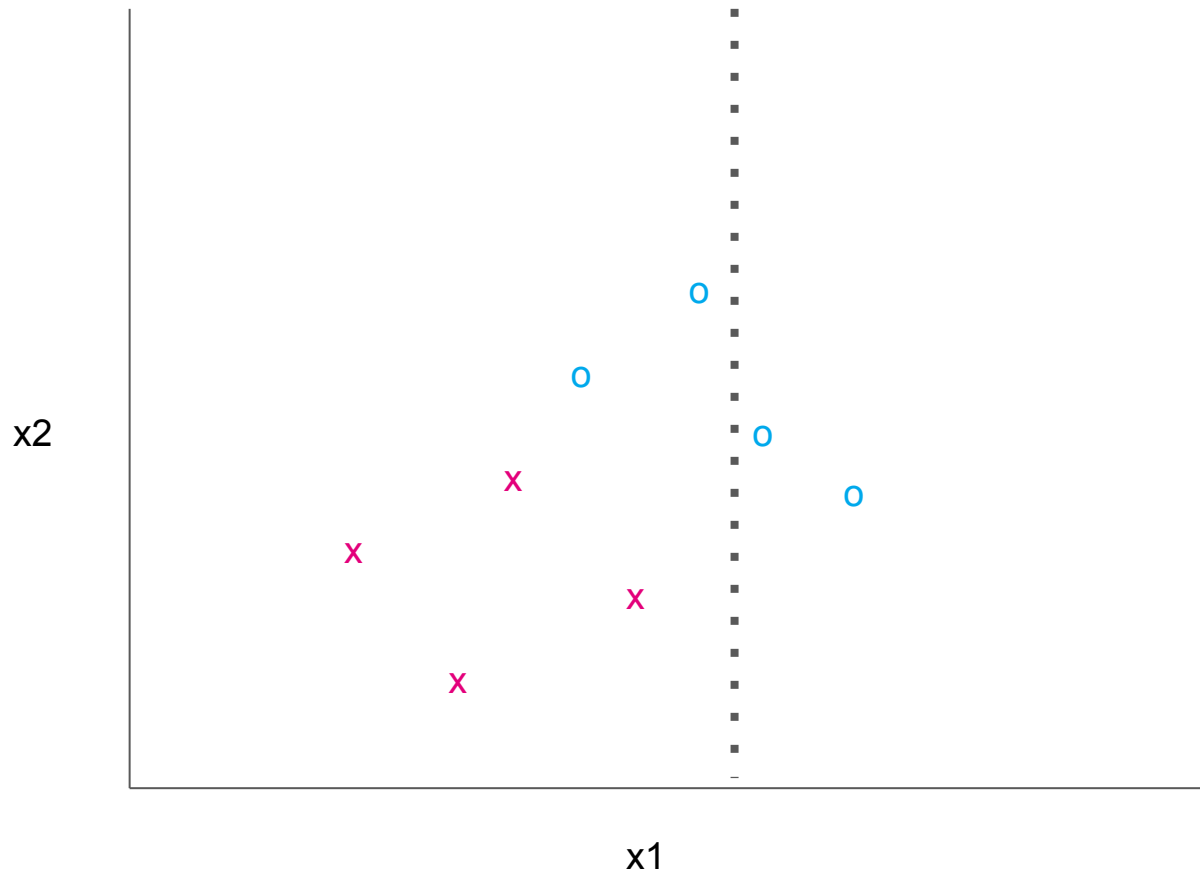


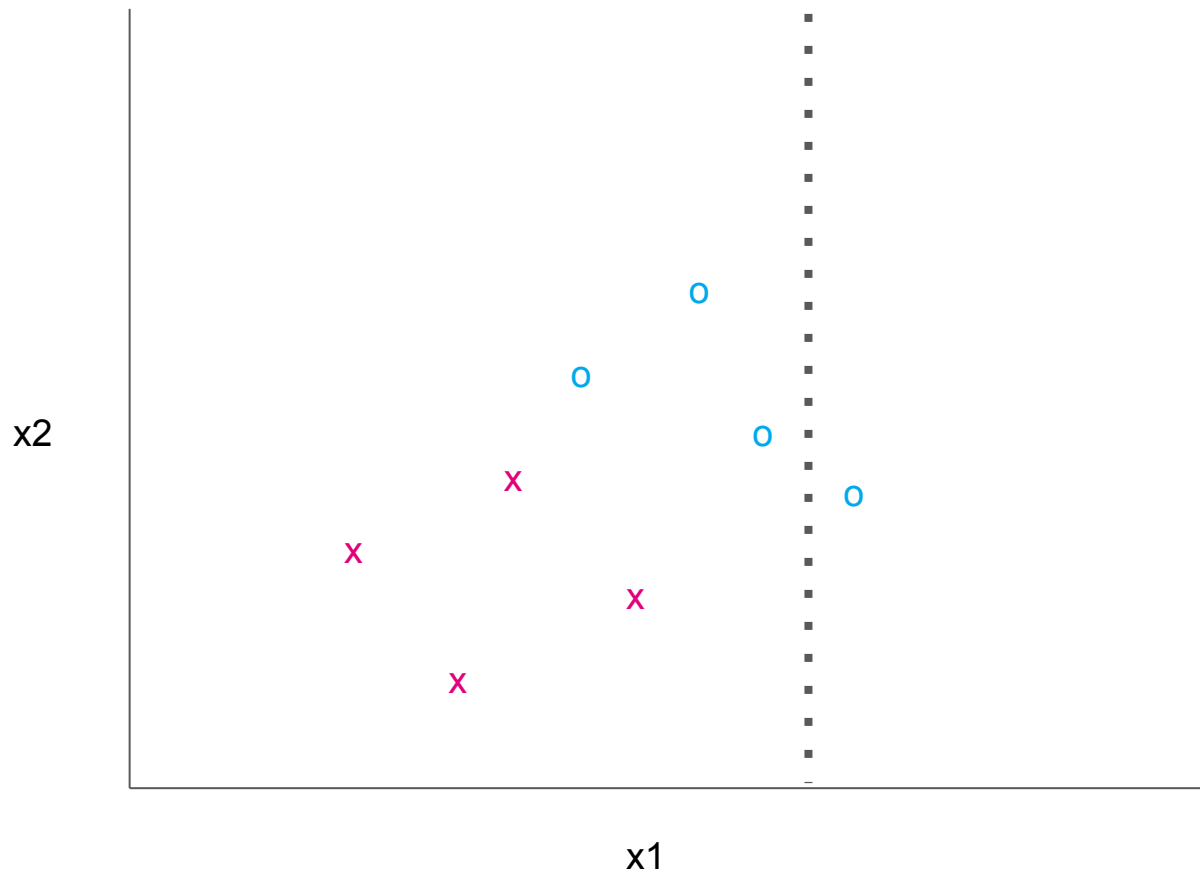


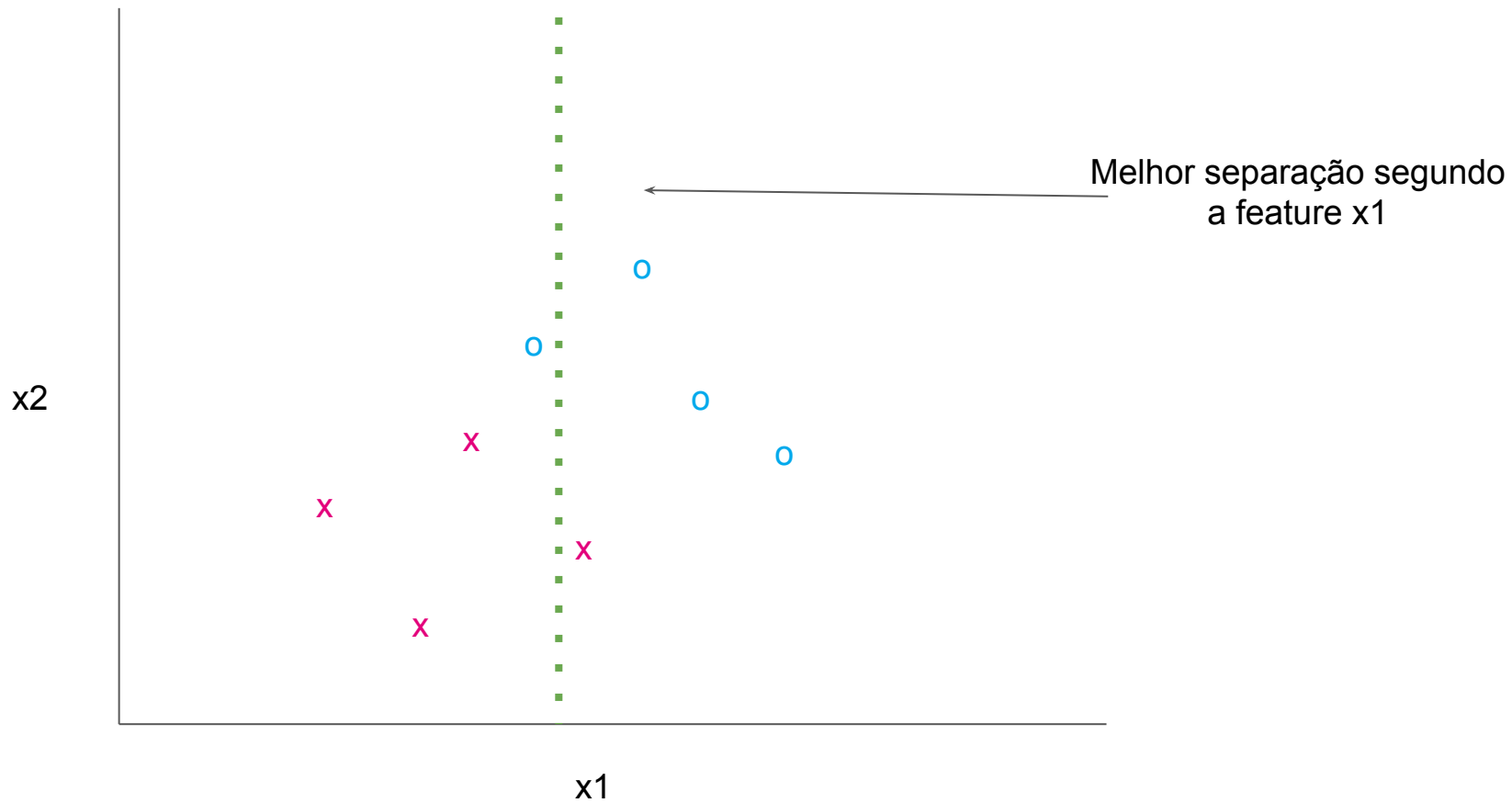


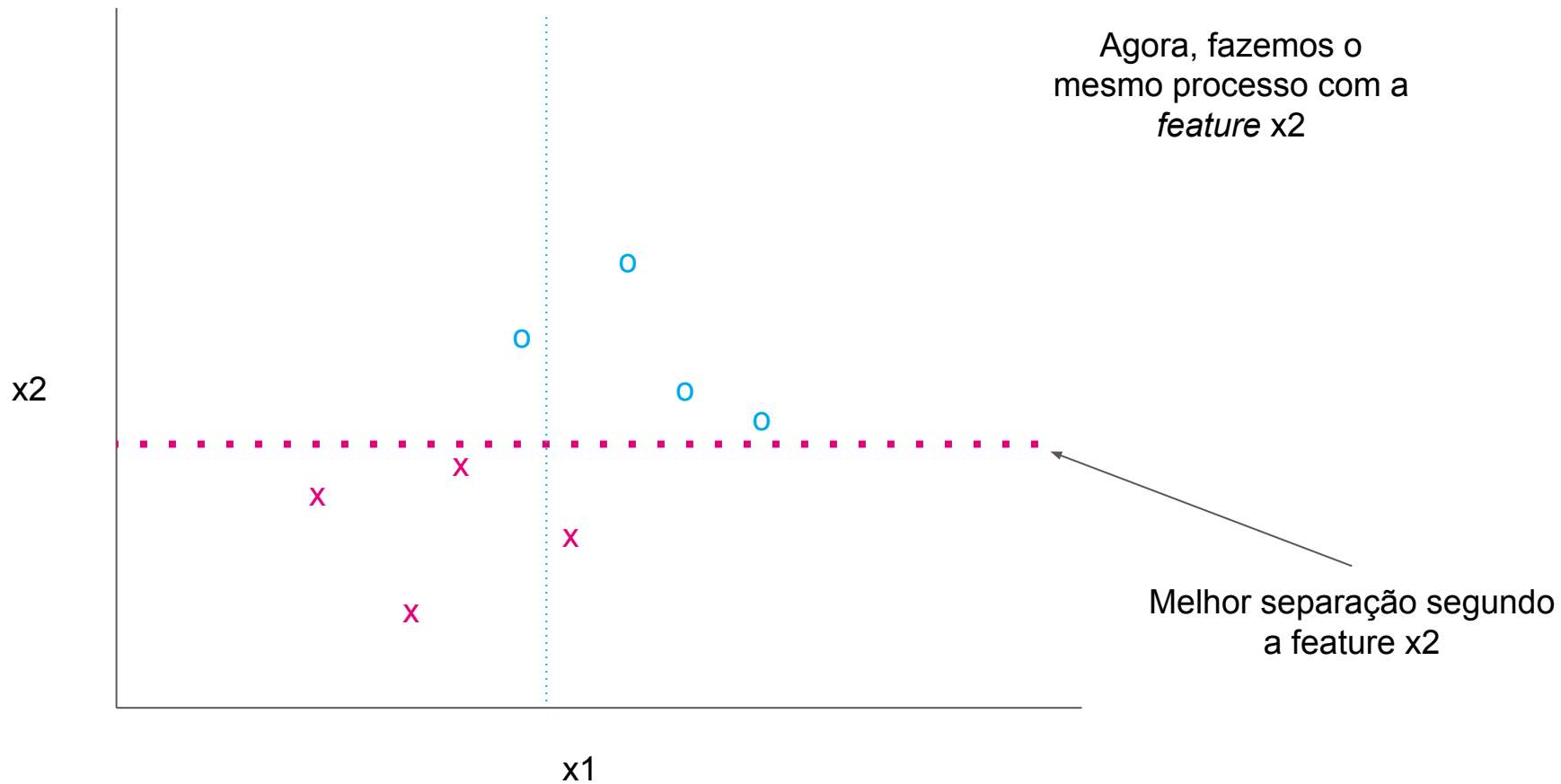


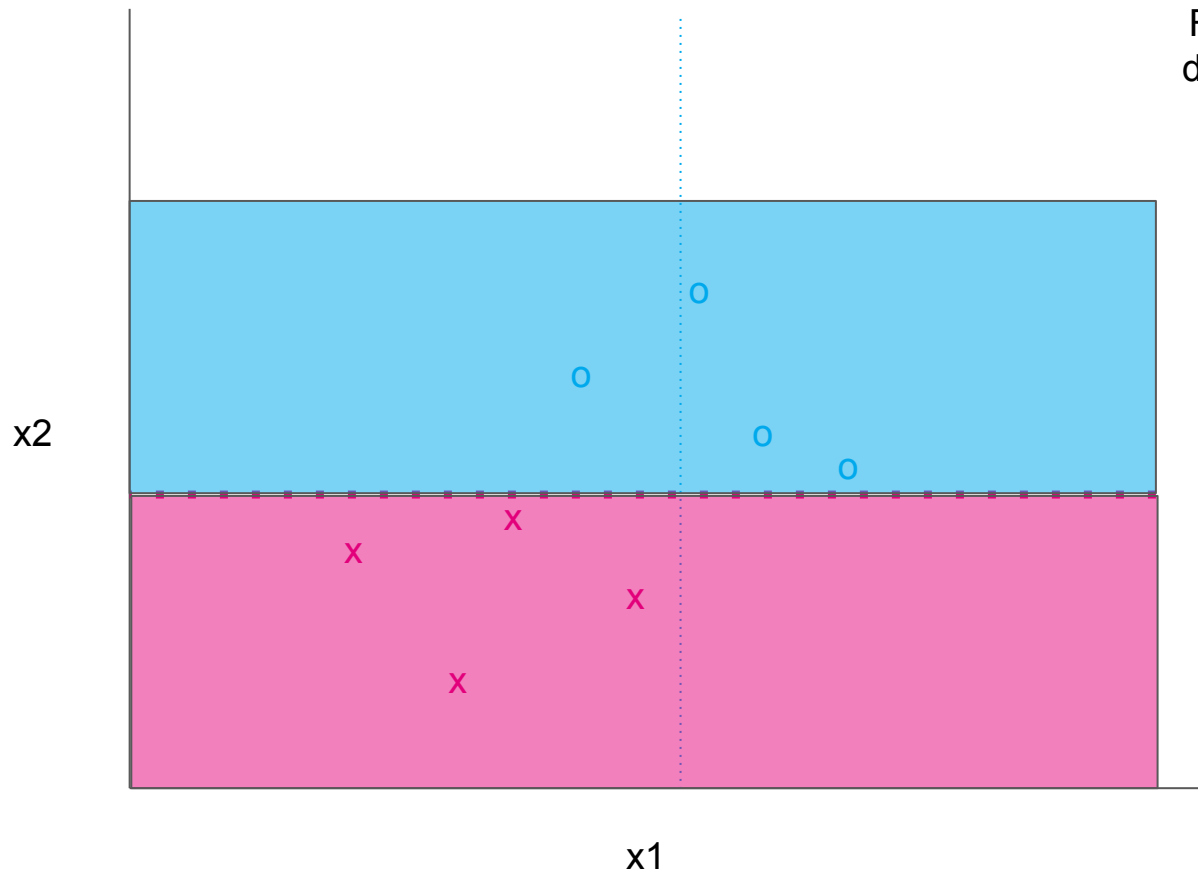






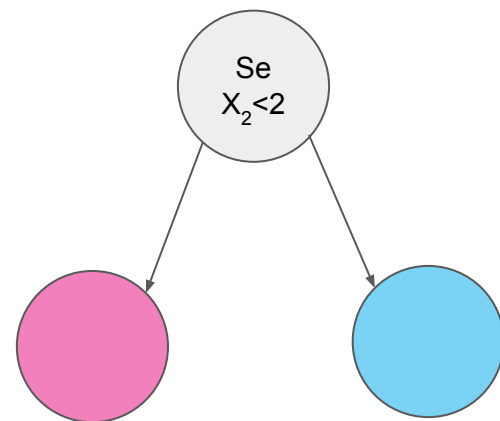
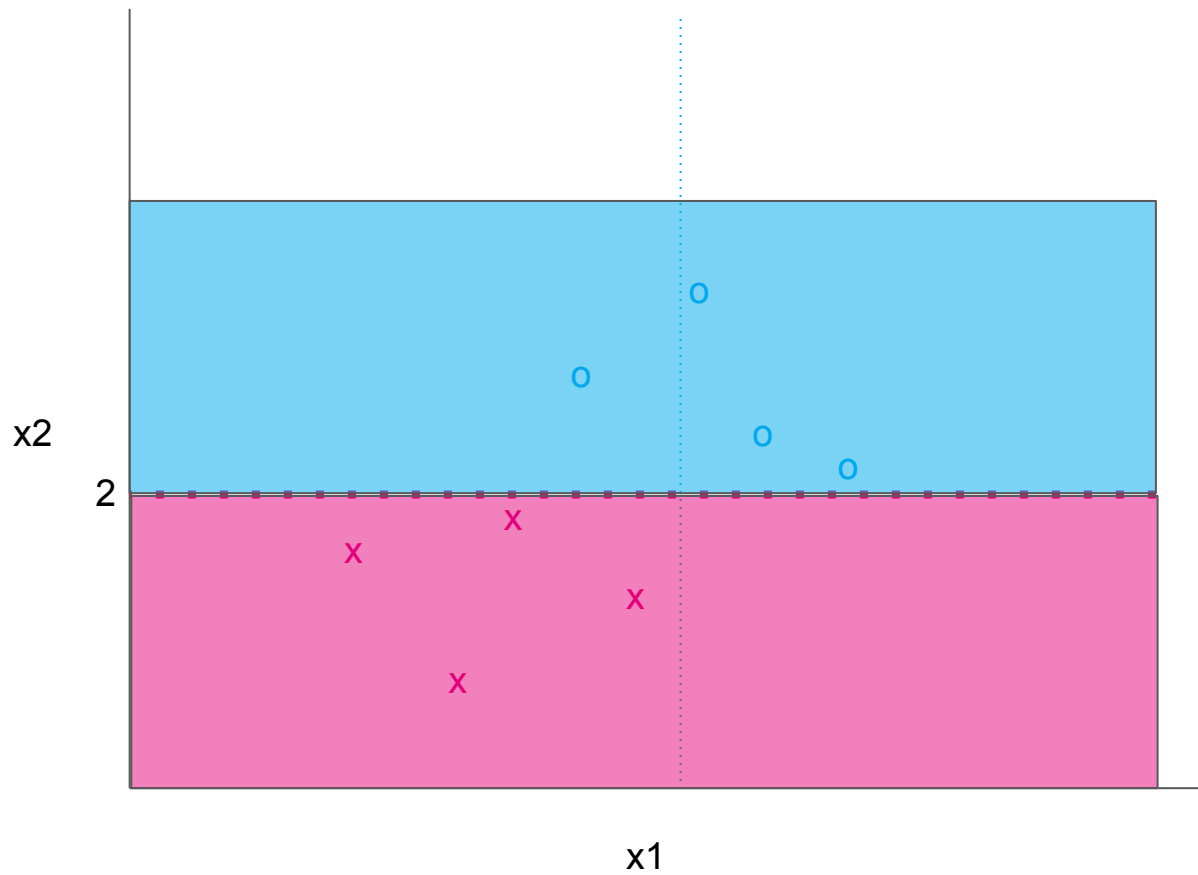




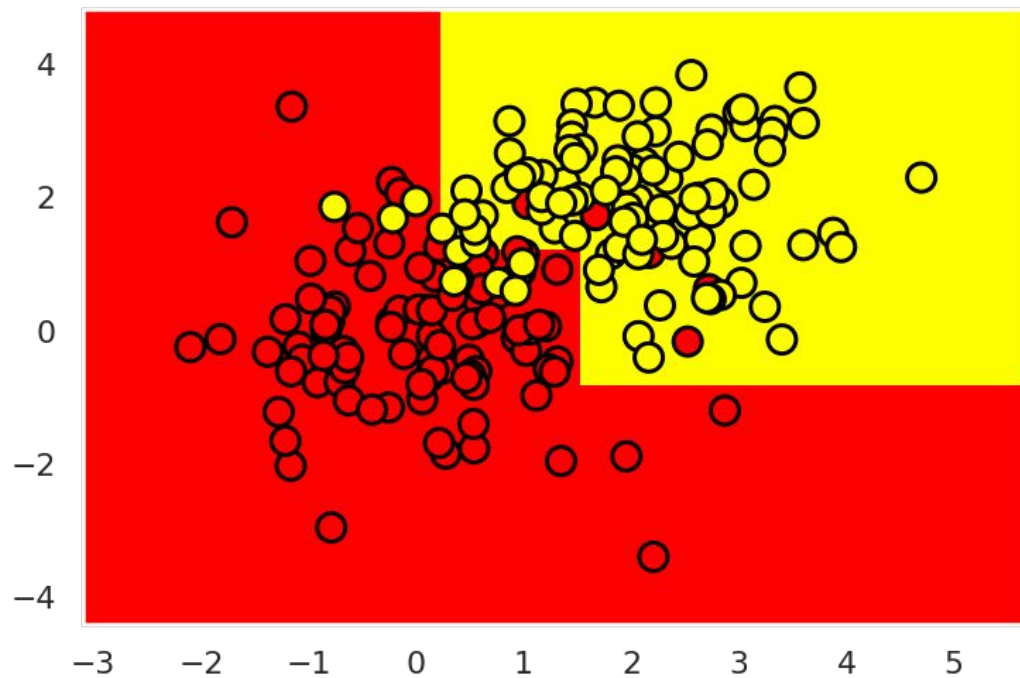


Feature  $x_2$  separa melhor os dados, logo decidimos usá-la para dividir





# Na prática





Dúvidas



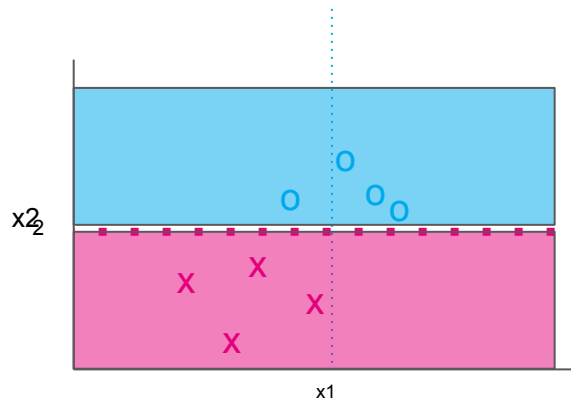
# Encontrando a árvore ideal...

- É a menor árvore possível que vai bem em nosso conjunto de treino
- Problema NP-Hard (computacionalmente inviável)
- Solução: **Heurísticas** greedy (Solução aproximada do que queremos)



# Heurísticas para cortar

- Passar por todos os pontos em todas as features e pegar o melhor (Estratégia *Greedy* / Gulosa).
- Nossa melhor divisão é aquela que no passo atual separa melhor nossos dados.
  - Impureza de Gini
  - Entropia
  - Variance Reduction -> Regressão



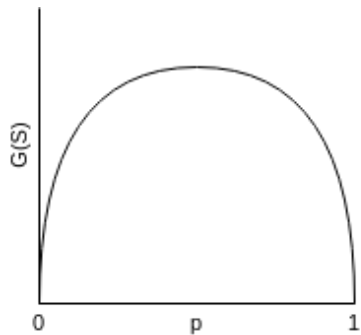
# Métricas de Pureza para Classificação

Impureza de Gini

Entropia

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

$$H(p_k) = - \sum_{k=1}^K p_k \log(p_k)$$



Número de pontos do corte pertencentes à classe K

Número de pontos do corte

# Métrica de Pureza de para Regressão

Variance Reduction

$$L(S) = \frac{1}{|S|} \sum_i^S (y_i - \mu_S)^2$$

Número de  
pontos dentro  
do nó

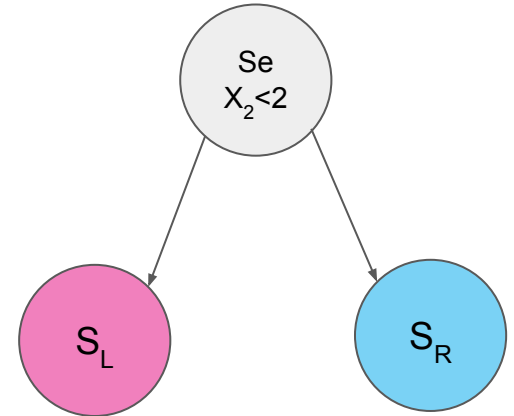
Variância

Média dos  
pontos do nó

# Métrica de Pureza para a divisão

$$Impurity = \frac{|S_L|}{|S|} M(S_L) + \frac{|S_R|}{|S|} M(S_R)$$

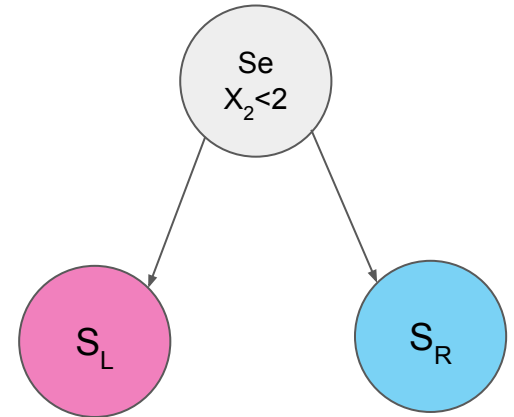
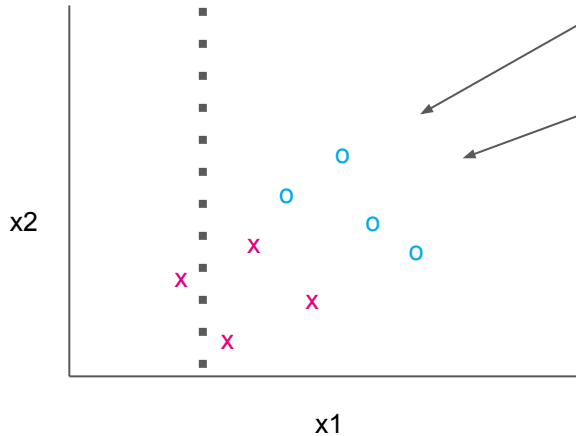
Impureza de Gini, Entropia ou Variance Reduction





# Métrica de Pureza para a divisão

$$Impurity = \frac{|S_L|}{|S|} M(S_L) + \frac{|S_R|}{|S|} M(S_R)$$





dúvidas?



# Custo de treino

$$O(\underbrace{ND}_{\text{Custo de cada divisão}} \cdot \underbrace{KN}_{\text{Custo de avaliar cada divisão}}) = \underbrace{O(KDN^2)}$$

Custo de cada  
divisão

Custo de  
avaliar cada  
divisão



## Custo de treino

$$O(\underbrace{ND}_{\text{Custo de cada divisão}} \cdot \underbrace{KN}_{\text{Custo de avaliar cada divisão}}) = \underbrace{O(KDN^2)}_{\text{Um custo maior que a fase de teste do k-NN?}}$$

Custo de cada  
divisão

Custo de  
avaliar cada  
divisão

Um custo maior que a fase  
de teste do k-NN?

Podemos aproveitar os cálculos  
de avaliar uma quebra para todas  
as subsequentes - [Referência](#)

$$O(KDN)$$

## Custo de teste (Inferência)

Basta descer a árvore, o que é

uma operação logarítmica

**(MUITO RÁPIDO)!**



# Custo em memória

Com as Decision Trees armazenamos apenas os nós da árvore, o que faz com que sejam **muito** leves =D





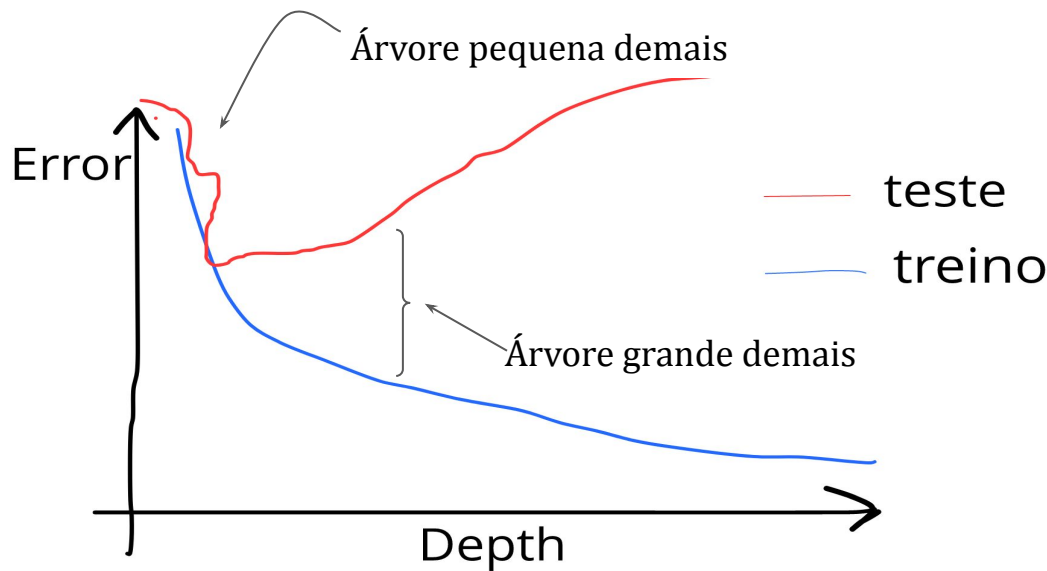
dúvidas?



O que acontece se fizermos uma árvore  
muito profunda? E muito rasa?

# Desvantagens de árvores de decisão ?

- Na prática, árvores de decisão **não são muito boas sozinhas**





# Então... porque Decision Trees?

- Fácil interpretabilidade
- Funcionam “out of the box”, pois usam qualquer tipo de dado
- Não são sensíveis à escala das *features*
- **Muito** rápidas e otimizadas

Na prática, Árvores de Decisão **são péssimas**, mas são a base do *Bagging* e do *Boosting*



# Referências

- Aulas [29](#) e [30](#) - Curso Machine Learning Cornell - Kilian Weinberger





dúvidas?

