"Prevejo um tempo em que os humanos serão para as máquinas, o que os cães são para nós. E eu estou torcendo pelas máquinas" (Claude Shannon).

Árvores de Decisão

Paulo Ricardo Lisboa de Almeida

DSBD

UFPR

Classificação Supervisionada

Em aprendizado de máquina, um modelo de classificação prediz a classe de determinada instância.

A instância pode ser uma foto, um sinal de áudio, um vetor com dados, ...

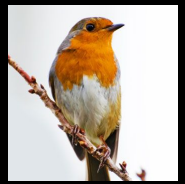
A classe pode ser um item que desejamos reconhecer (ex.: essa é uma foto de um cachorro, pessoa, pássaro, ...).

Classificação Supervisionada

Em aprendizado de máquina, um modelo de classificação prediz a classe de determinada instância.

A instância pode ser uma foto, um sinal de áudio, um vetor com dados, ...

A classe pode ser um item que desejamos reconhecer (ex.: essa é uma foto de um cachorro, pessoa, pássaro, ...).

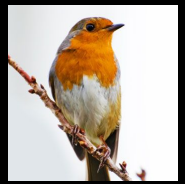


Classificação Supervisionada

Em aprendizado de máquina, um modelo de classificação prediz a classe de determinada instância.

A instância pode ser uma foto, um sinal de áudio, um vetor com dados, ...

A classe pode ser um item que desejamos reconhecer (ex.: essa é uma foto de um cachorro, pessoa, pássaro, ...).



Classificador

Cachorro

Pessoa

Pássaro

Classificação Supervisionada

Modelo Supervisionado.

Um modelo supervisionado aprende automaticamente com **dados rotulados de treino**.

Exemplo: são dadas imagens juntamente com seus rótulos criados por humanos para que o modelo possa gerar suas representações internas.

Problema

Considere um problema onde os dados de pressão arterial e nível de glicose são medidos para uma pessoa.

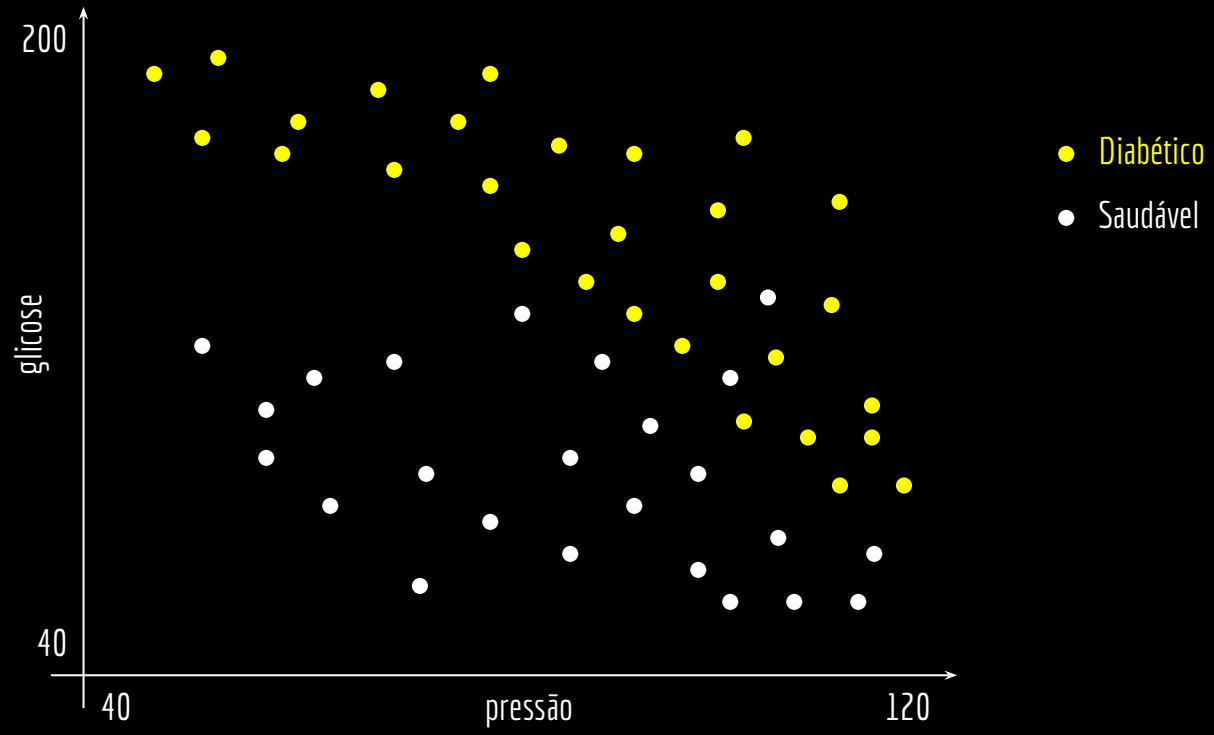
Esses dados podem ser lidos, por exemplo, via um *smart-watch*.

Desejamos classificar as pessoas entre “com diabetes” (classe 1) e “saudáveis” (classe 0).

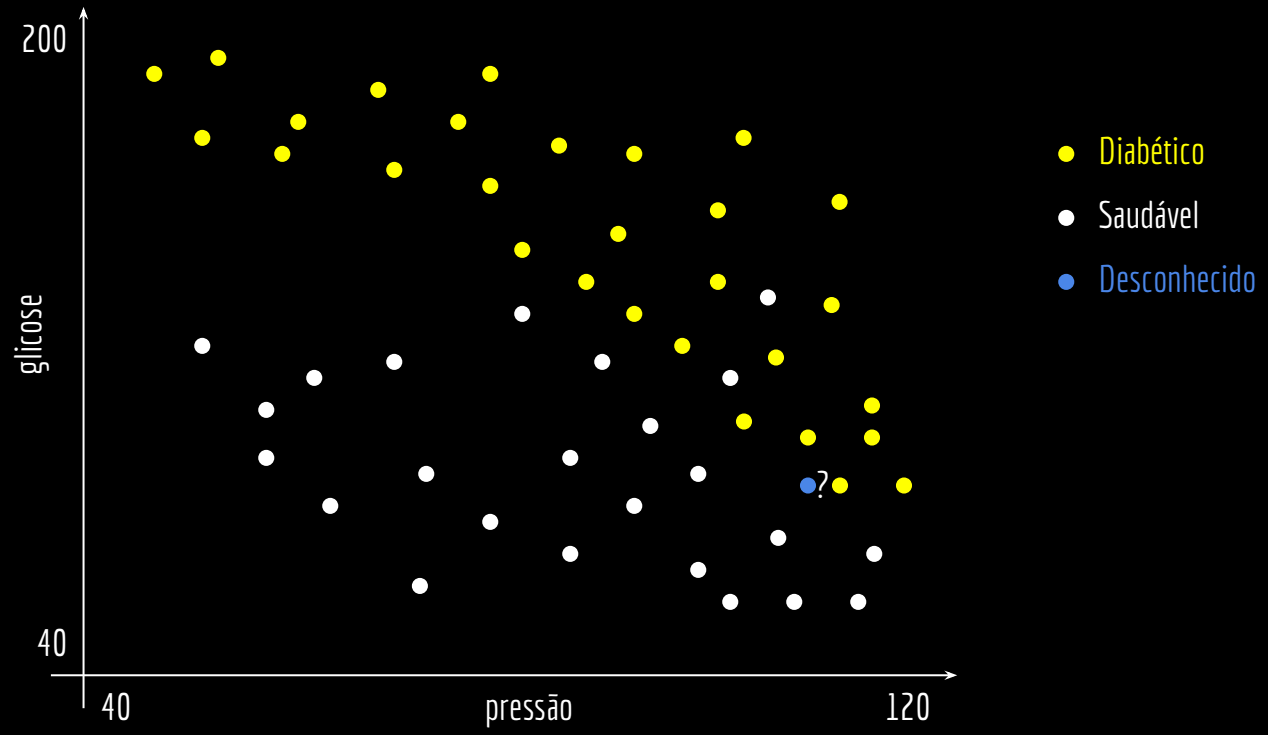
Temos então um problema binário.

Atenção: essa é uma simplificação da realidade. Classificar pessoas entre diabéticas e saudáveis não é nem de longe tão simples.

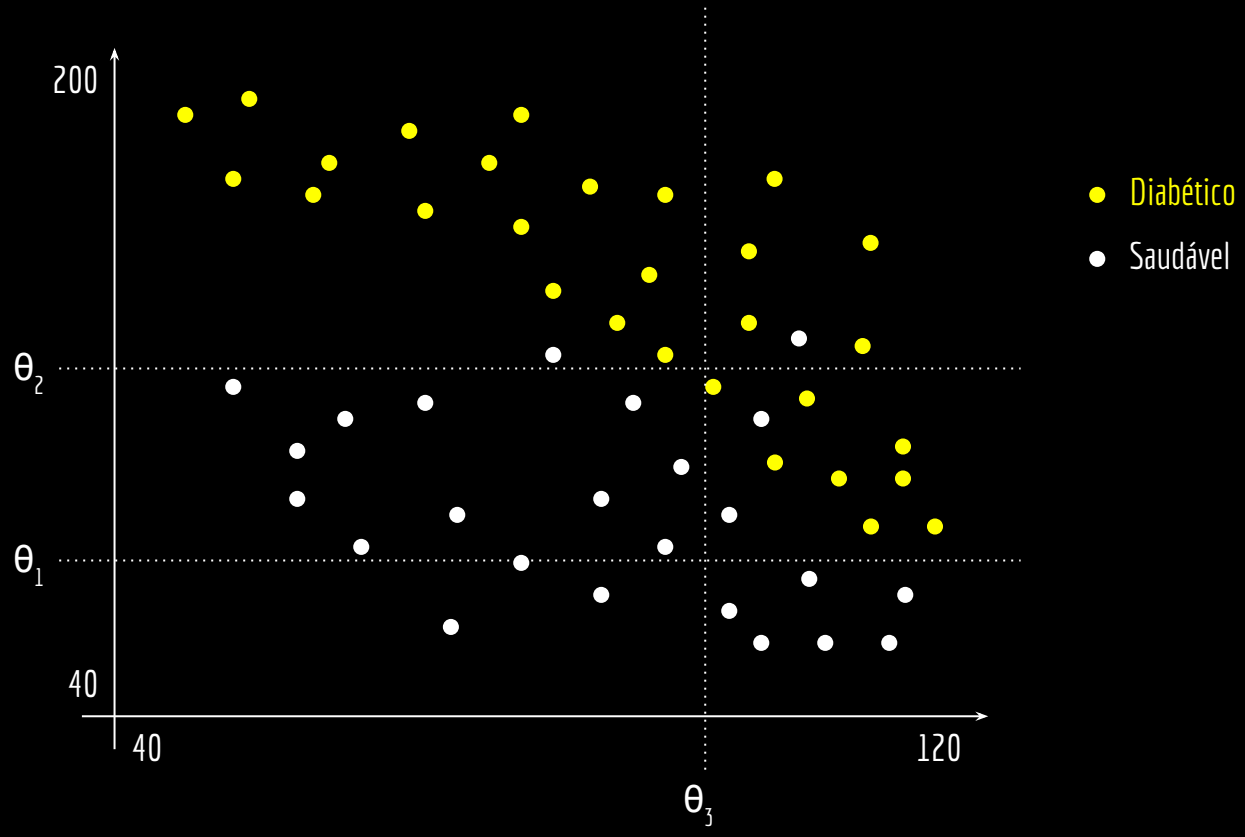
Dataset de Treinamento



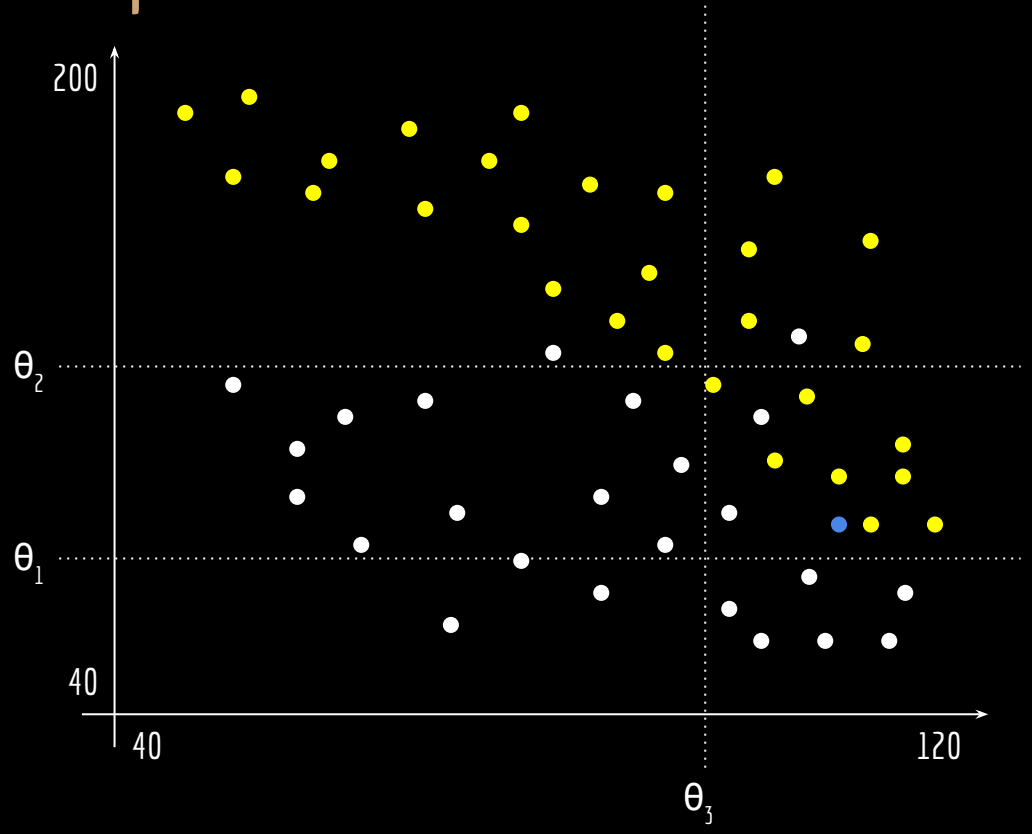
Dataset de Treinamento



Dividindo os dados



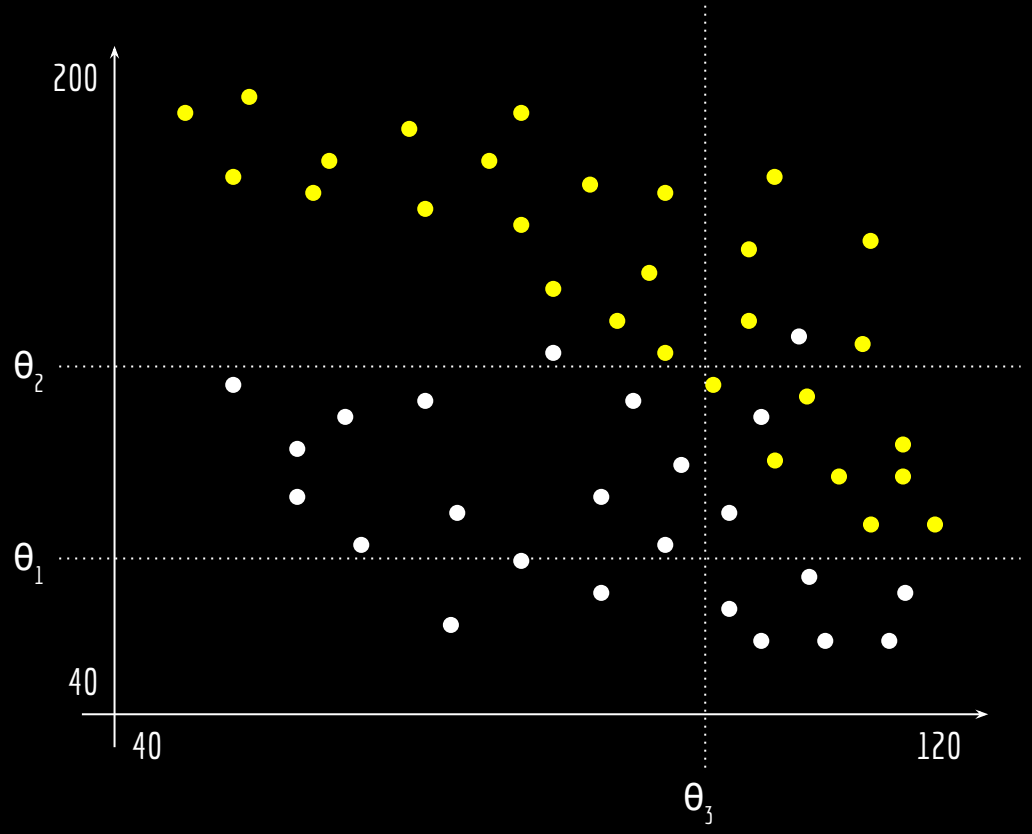
Classificando



- Diabético
- Saudável
- Classificado como Diabético

É maior que θ_3 no eixo x, e está entre θ_1 e θ_2 no eixo y, então está na área considerada como diabético.

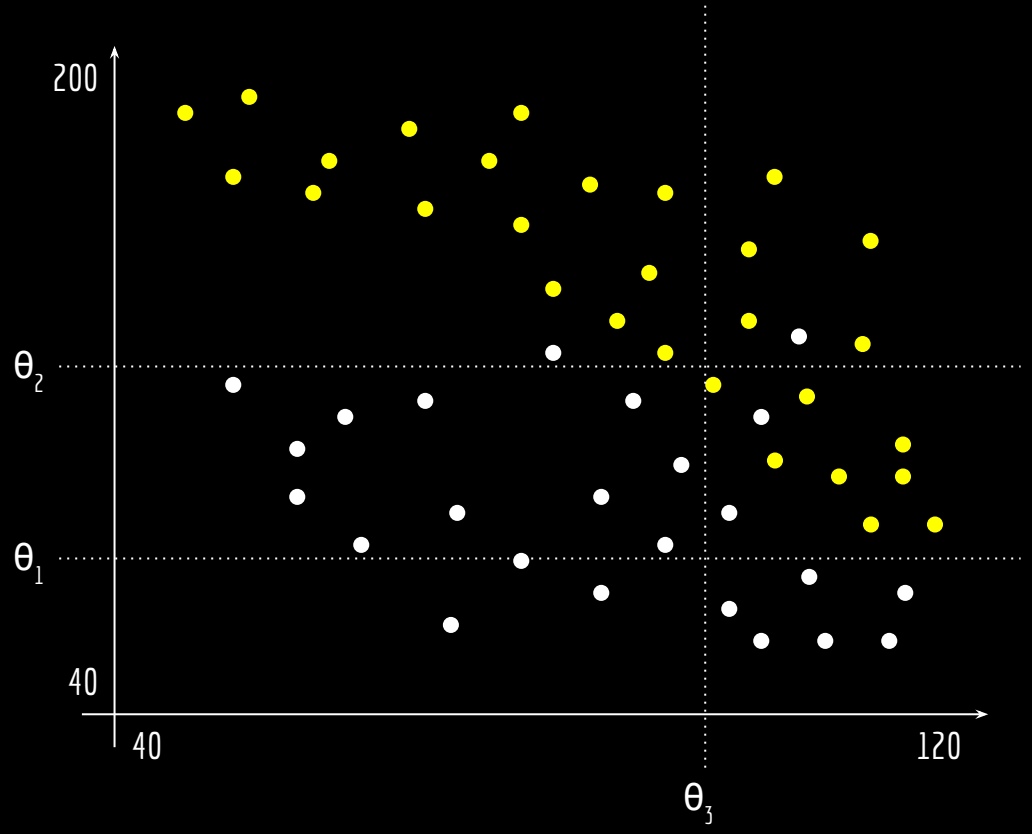
Árvore de Decisão

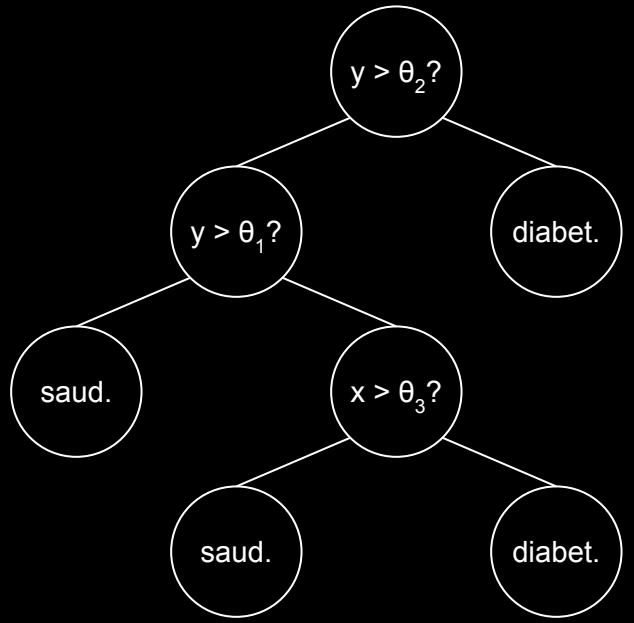


O problema pode ser modelado como uma árvore binária.

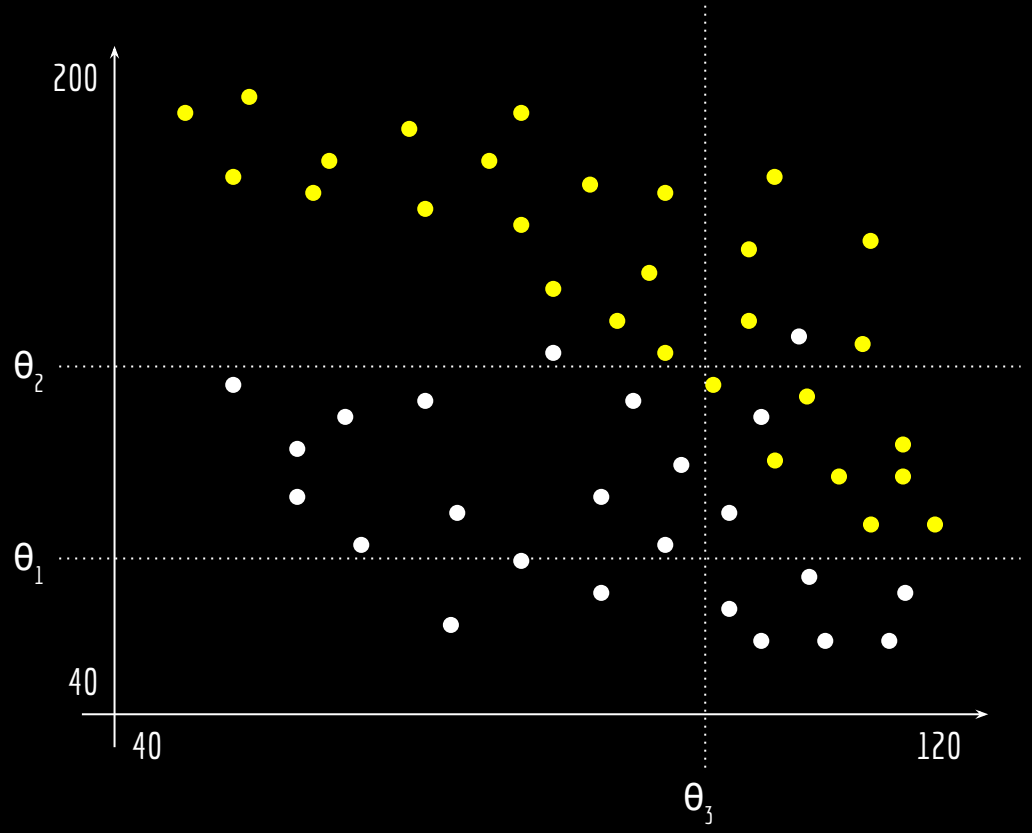
Algoritmo de **Árvore de Decisão**.

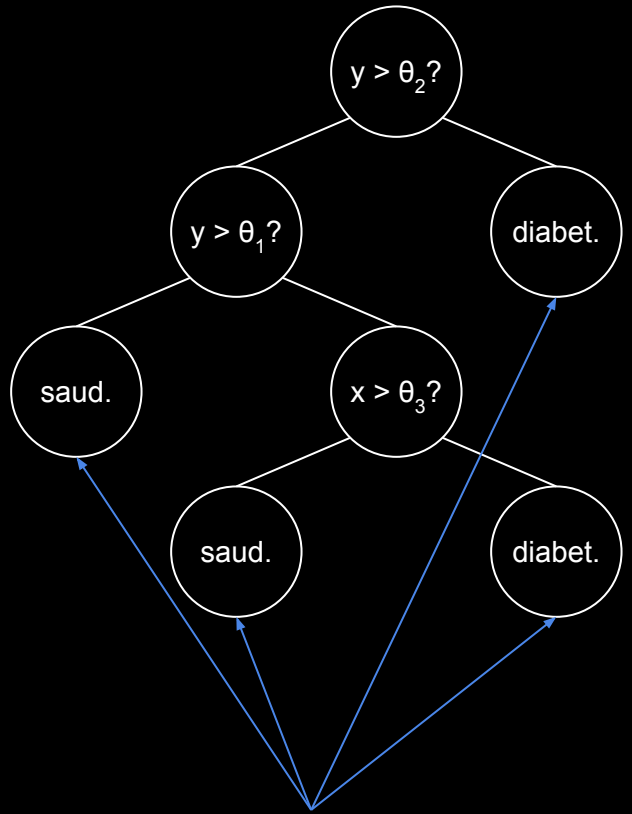
Árvore de Decisão





Árvore de Decisão



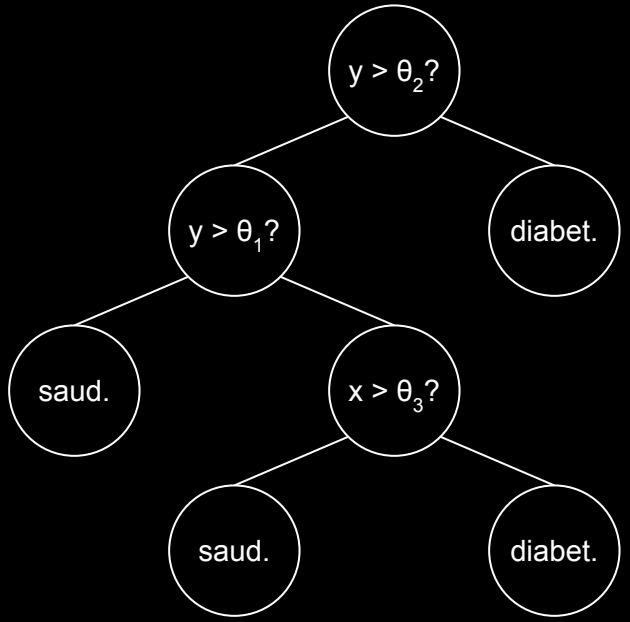


Folhas dão as classes.

Criando a árvore

Precisamos criar a árvore.

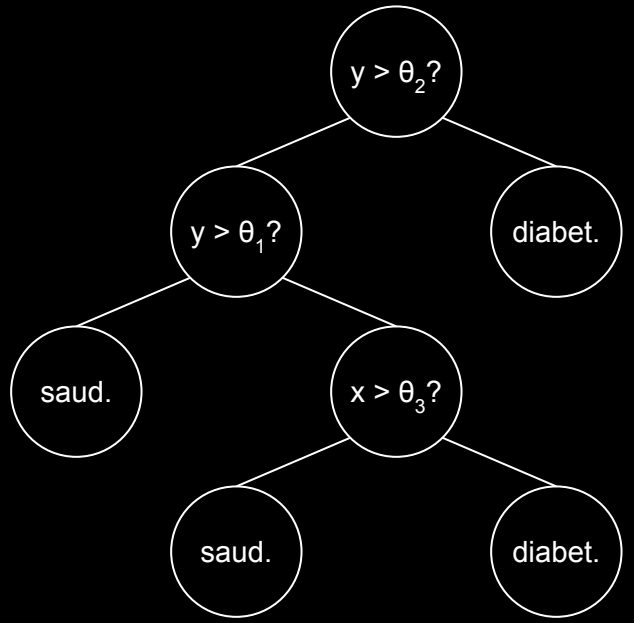
O que precisamos definir?



Criando a árvore

Precisamos criar a árvore.

- Definir a estrutura da árvore.
  - Ex.: por qual θ precisamos começar a busca?
- Definir os limiares (valores para cada θ).



Algoritmo

Definir a estrutura ótima da árvore é computacionalmente inviável mesmo para problemas pequenos.

Geralmente usamos algum algoritmo guloso.

Exemplos de algoritmos clássicos.

- CART
- ID.3
- C4.5 (Implementação Open Source no Weka se chama J48)

Vetores de característica

O exemplo anterior possuía apenas duas características.

Glicose e pressão.

Problema bidimensional.

No mundo real, geralmente encontramos problemas com mais dimensões.

Os dados que descrevem as instâncias de um problema são chamados de características.

Uma instância tem um vetor de características $x = [x_1, x_2, ..., x_n]$, e uma classe $y \in [y_1, y_2, ..., y_k]$.

Onde n é o número de características que descrevem o problema, e k é o número de classes possíveis.

Vetores de característica

Suponha que as características do problema são: *numGravidez*, *Glicose*, *Pressão*, *espessuraPele*, *Insulina* ,*BMI*, *DPF* e *Idade*.

A classe *y* pode ser *0 = saudável*, e *1 = diabetes*.

Um vetor de características para treinamento pode ser:

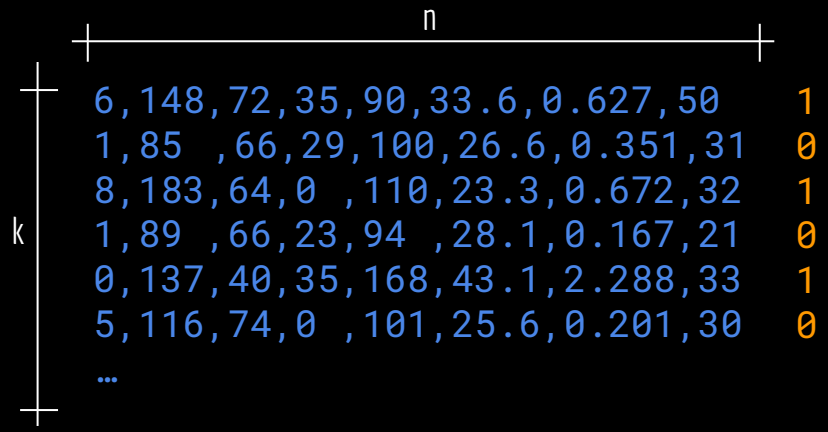
x = [6,148,72,35,0,33.6,0.627,50,1] e *y = 1*.

Temos agora um problema com 9 dimensões.

Dados de treinamento

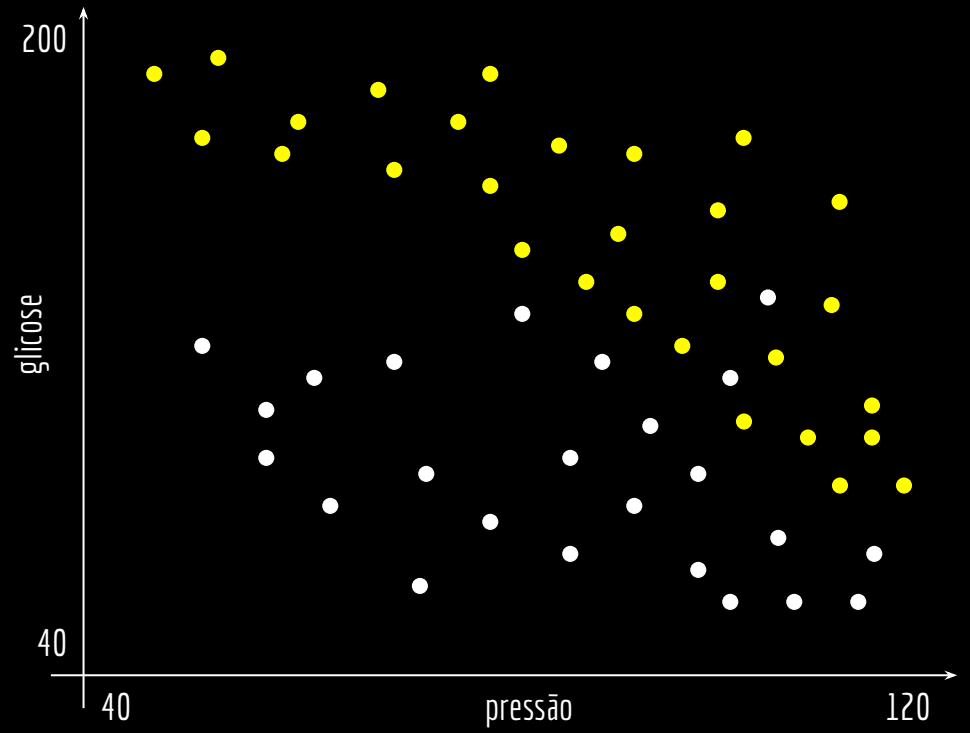
Supondo que temos *k* dados para treinamento, temos uma matriz de *k x n*, onde cada linha da matriz representa um vetor de treinamento. E um vetor y^T de tamanho *k*.

Matriz de *k x n* onde cada linha é um vetor de características.



Vetor y^T com os rótulos das instâncias.

Impureza

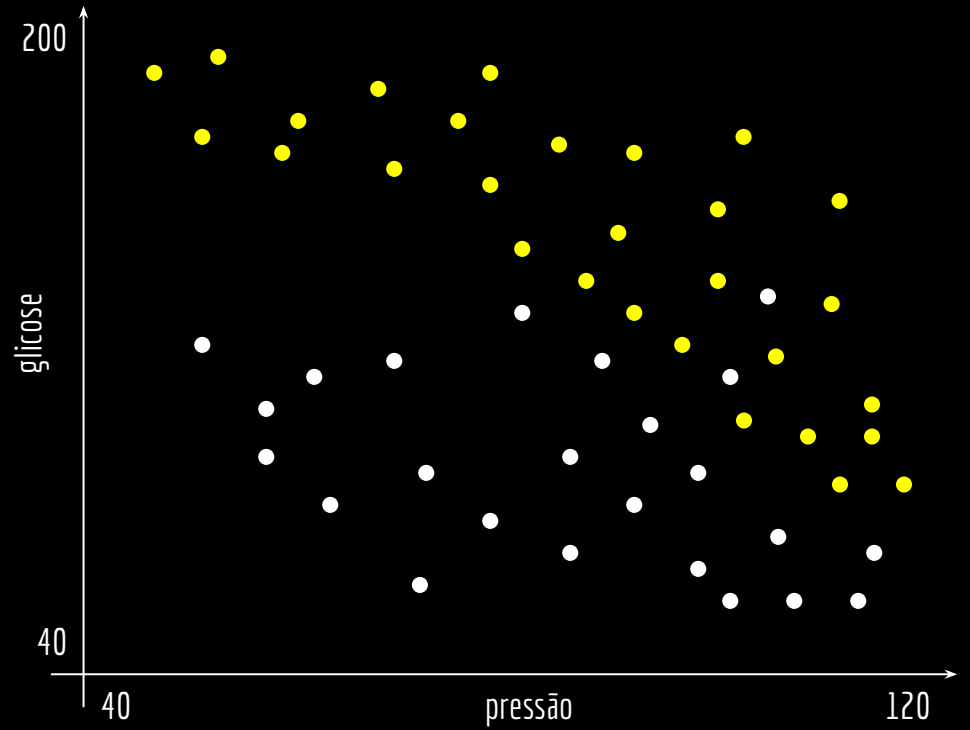


Em um problema **binário**, podemos calcular a impureza de determinada região como:

$$i(N) = P(y_1)P(y_2)$$

Onde $P(y_x)$ é a proporção de itens da classe 0 ou 1 na região.

Impureza



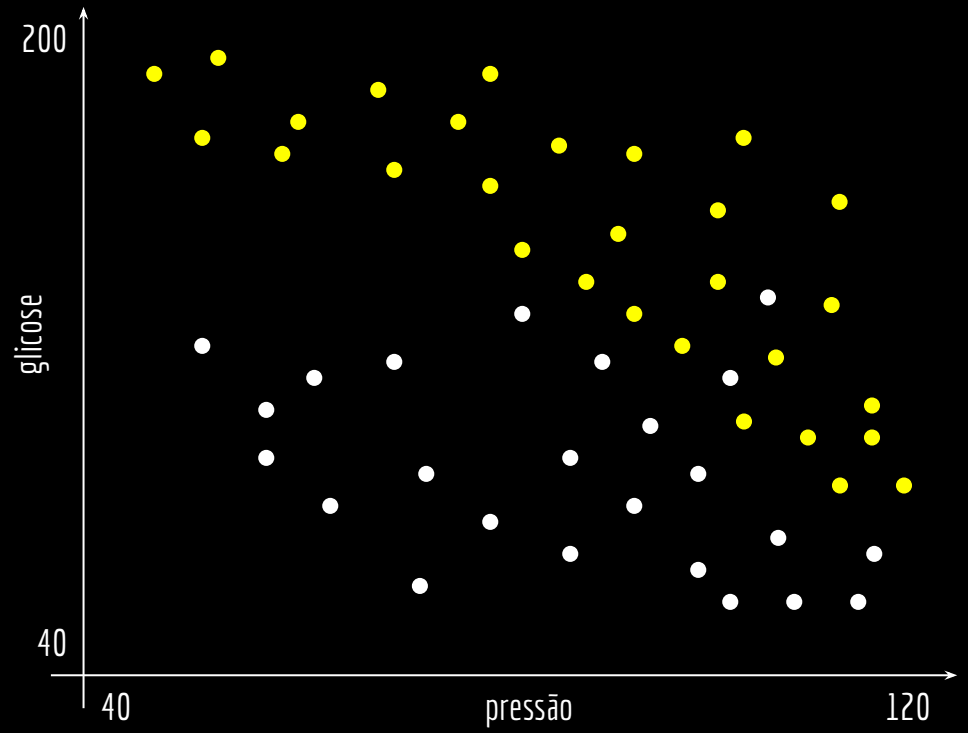
Em um problema **binário**, podemos calcular a impureza de determinada região como:

$$i(N) = P(y_1)P(y_2)$$

Onde $P(y_x)$ é a proporção de itens da classe 0 ou 1 na região.

Uma região pura terá i(N) = 0.

Impureza

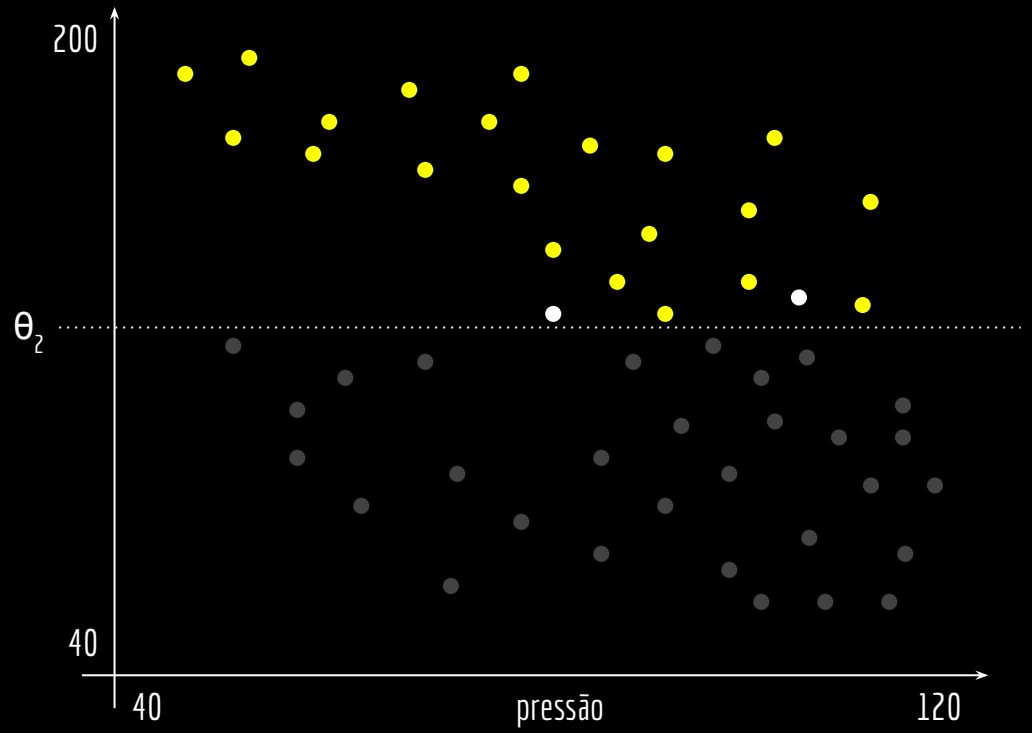


Contagens

- 28
- 24

i(N) = (28:52)×(24:52) = 0,25

Impureza



Contagens

- 21
- 2

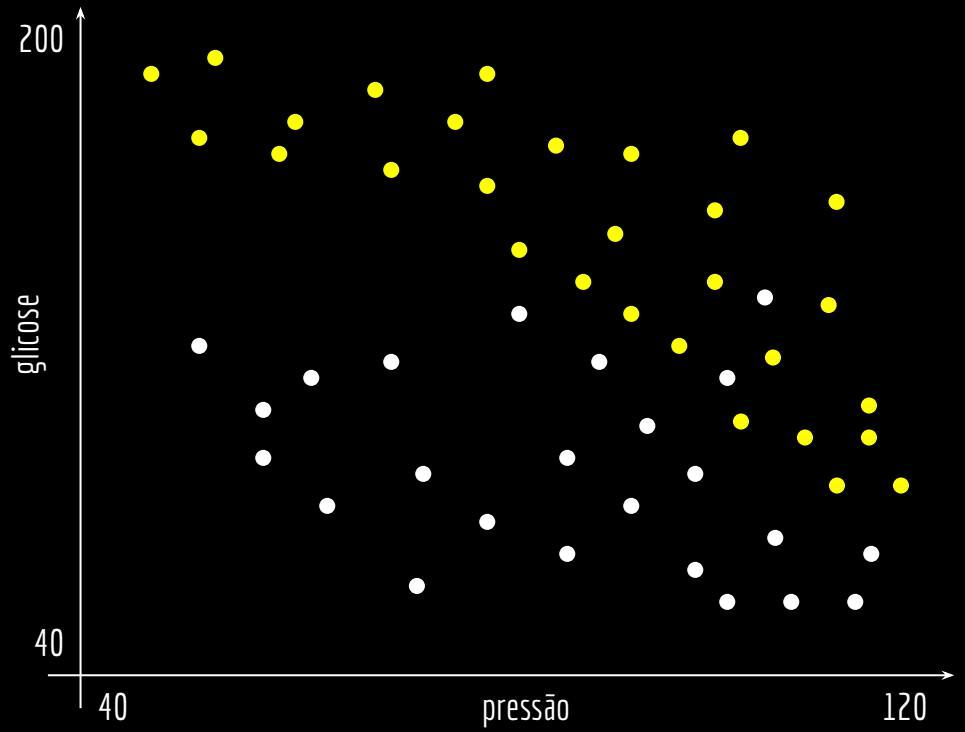
$i(N) = (21 \div 23) \times (2 \div 23) = 0{,}08$

Índice Gini

A impureza pode ser generalizada para casos onde temos duas ou mais classes, criando o **Índice Gini**:

$$i(N) = \sum_{i \neq j} P(y_i)P(y_j) = 1 - \sum_{j} P^2(y_j)$$

Algoritmo - Ideia



Adicionar um novo nodo na árvore.

Para cada característica c

Para cada limiar de decisão possível

Calcular o Índice Gini

Escolher como a característica e limiar que levam ao menor Índice Gini.

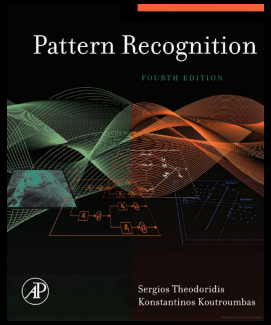
Continuar criando nodos enquanto reduções no Índice Gini sejam significativas.

Atenção

Atenção: essa é uma simplificação de um dos muitos algoritmos possíveis que podem ser usados para se criar a árvore de decisão.

Veja uma versão completa de um algoritmo, e algumas provas em Theodoridis e Koutroumbas (2008).

Quarta Edição, Seção 4.20.



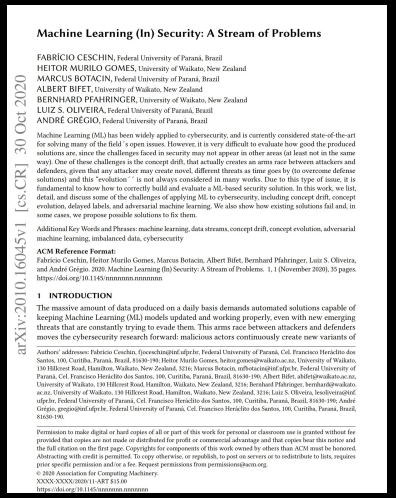
Árvores de Decisão

Árvores de Decisão são comumente usadas em ambientes como apoio à decisão de especialistas (médicos, economistas, ...) e em ambientes críticos, onde precisamos saber o motivo de determinada amostra ser classificada com determinada classe.

Por exemplo, em um ambiente de detecção de malwares, precisamos saber por que determinado programa foi classificado como malicioso.

Ex.: na árvore de decisão, podemos ver que
o programa chama determinada função maliciosa.

Ceschin, F., Gomes, H. M., Botacin, M., Bifet, A., Pfahringer, B., Oliveira, L. S., Grégio, A. Machine learning (in) security: A stream of problems. 2020.



Machine Learning (In) Security: A Stream of Problems

FABRÍCIO CESCHIN, Federal University of Paraná, Brazil
HEITOR MURILO GOMES, University of Waikato, New Zealand
MARCUS BOTACIN, Federal University of Paraná, Brazil
ALBERT BIFET, University of Waikato, New Zealand
BERNHARD PFAHRINGER, University of Waikato, New Zealand
LUIZ S. OLIVEIRA, Federal University of Paraná, Brazil
ANDRÉ GRÉGIO, Federal University of Paraná, Brazil

Machine Learning (ML) has been widely applied to cybersecurity, and is currently considered state-of-the-art for solving many of the field 's open issues. However, it is very difficult to evaluate how good the produced solutions are, since the challenges faced in security may not appear in other areas (at least not in the same way). One of these challenges is the concept drift, that actually creates an arms race between attackers and defenders, given that any attacker may create novel, different threats as time goes by (to overcome defense solutions) and this "evolution'' is not always considered in many works. Due to this type of issue, it is fundamental to know how to correctly build and evaluate a ML-based security solution. In this work, we list, detail, and discuss some of the challenges of applying ML to cybersecurity, including concept drift, concept evolution, delayed labels, and adversarial machine learning. We also show how existing solutions fail and, in some cases, we propose possible solutions to fix them.

Additional Key Words and Phrases: machine learning, data streams, concept drift, concept evolution, adversarial machine learning, imbalanced data, cybersecurity

ACM Reference Format:
Fabrício Ceschin, Heitor Murilo Gomes, Marcus Botacin, Albert Bifet, Bernhard Pfahringer, Luiz S. Oliveira, and André Grégio. 2020. Machine Learning (In) Security: A Stream of Problems. 1, 1 (November 2020), 35 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

The massive amount of data produced on a daily basis demands automated solutions capable of keeping Machine Learning (ML) models updated and working properly, even with new emerging threats that are constantly trying to evade them. This arms race between attackers and defenders moves the cybersecurity research forward: malicious actors continuously create new variants of

Authors' addresses: Fabrício Ceschin, fjoceschin@inf.ufpr.br, Federal University of Paraná, Cel. Francisco Heráclito dos Santos, 100, Curitiba, Paraná, Brazil, 81630-190; Heitor Murilo Gomes, heitor.gomes@waikato.ac.nz, University of Waikato, 130 Hillcrest Road, Hamilton, Waikato, New Zealand, 3216; Marcus Botacin, mfbotacin@inf.ufpr.br, Federal University of Paraná, Cel. Francisco Heráclito dos Santos, 100, Curitiba, Paraná, Brazil, 81630-190; Albert Bifet, abifet@waikato.ac.nz, University of Waikato, 130 Hillcrest Road, Hamilton, Waikato, New Zealand, 3216; Bernhard Pfahringer, bernhard@waikato.ac.nz, University of Waikato, 130 Hillcrest Road, Hamilton, Waikato, New Zealand, 3216; Luiz S. Oliveira, lesoliveira@inf.ufpr.br, Federal University of Paraná, Cel. Francisco Heráclito dos Santos, 100, Curitiba, Paraná, Brazil, 81630-190; André Grégio, gregio@inf.ufpr.br, Federal University of Paraná, Cel. Francisco Heráclito dos Santos, 100, Curitiba, Paraná, Brazil, 81630-190.

XXXX-XXXX/2020/11-ART $15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

Árvores de Decisão

Árvores de decisão são propensas a overfitting (sobreajuste).

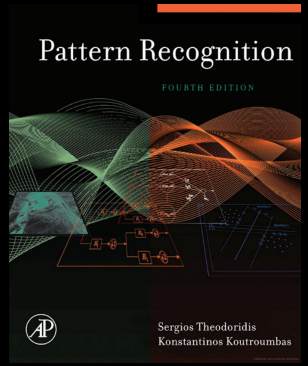
- O modelo pode se tornar muito complexo, com muitas fronteiras.
- No pior caso, cada região contém uma única instância.
- Problemas de generalização.

Exercícios

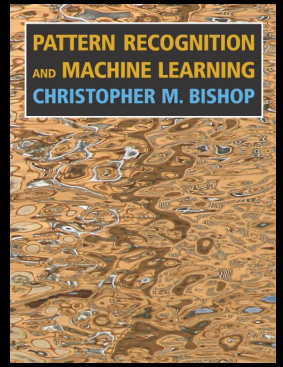
1. Faça testes com o programa de Árvore de Decisão disponibilizado no Moodle.
    a. Use o dataset para previsão de Diabetes disponibilizado.
        i. O dataset é do mundo real, e está disponível em www.kaggle.com/datasets/uciml/pima-indians-diabetes-database
        ii. Cuidado ao analisar os resultados! O dataset é desbalanceado (possui mais itens da classe saudável do que da classe diabetes). Isso pode enviesar a acurácia.
    b. Baixe outros datasets, e faça testes.

Referências

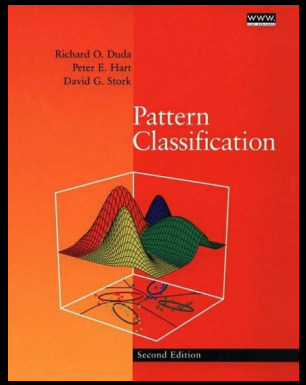
Theodoridis, S., Koutroumbas, K. Pattern Recognition. 4a ed. Elsevier Science. 2008.



Bishop, C. M. Pattern Recognition and Machine Learning. Springer. 2007.



Stork, D. G., Hart, P. E., Duda, R. O. Pattern Classification. Wiley. 2012.



scikit-learn.org/stable/modules/tree.html

www.kaggle.com/datasets/uciml/pima-indians-diabetes-database

www.datacamp.com/tutorial/decision-tree-classification-python

Licença