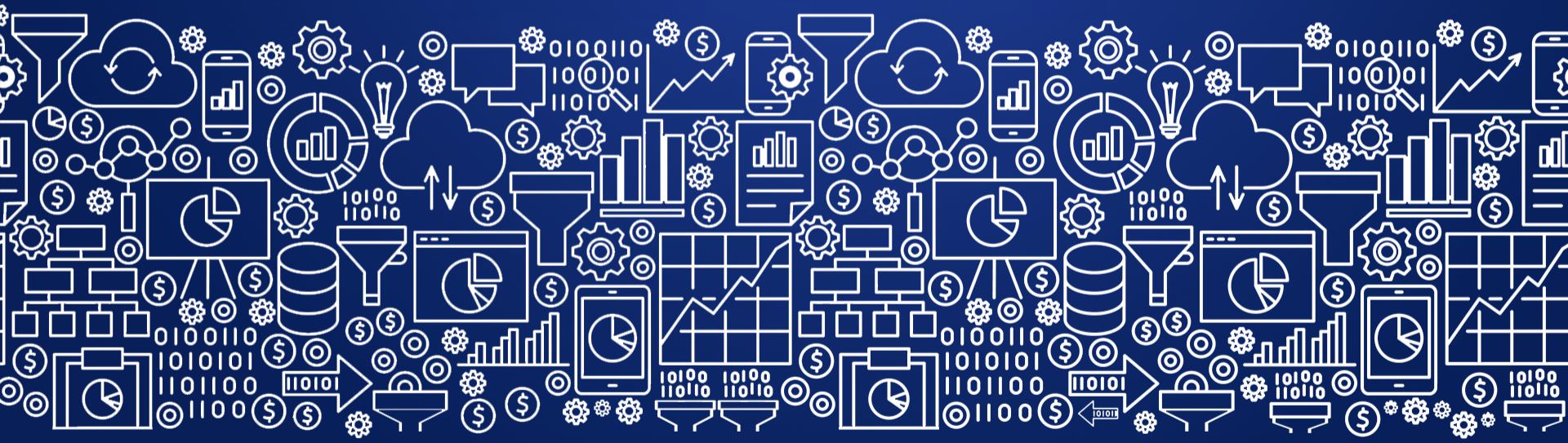
Minicurso – Machine Learning (ML)

Igor Braga, Igor Martinelli e Mateus Pedrino

QUEM SOMOS



QUEM SOMOS

Big Data

Empresa líder em **Data Science** para Business Performance no Brasil

Inteligência Artificial
+
visão de business
+
massa enorme de dados
=
resultados mensuráveis na ponta

SEM GUESSING,
SEM SUBJETIVIDADE:
100% QUANTI

Equipe de PhDs e mestres gerando inovação na fronteira do conhecimento

Portfolio de produtos com foco em KPIs de resultado em vendas e marketing

“Walk the walk”: relacionamentos de longo prazo em que só ganhamos se o cliente tem ganho comprovado

ALGUNS DE NOSSOS CLIENTES

Big Data

Anheuser-Busch

ambev

ALPARGATAS

BR

Unilever

Estácio

DYNAMO Capital LLP

extrafarma

oBoticário

Ipiranga

Quilmes

magazineluiza vem ser feliz

brf

Petz Seu pet center de estimação

CASES

Previsão de venda de revistas em bancas

- Venda consignada: não vendeu, editora recolhe, gerando prejuízo
- Criamos modelo para prever a quantidade de revistas que seriam vendidas em cada banca
- **Modelo vivo**: a cada semana os resultados da semana anterior eram alimentados ao modelo
- Resultado: redução de **64%** no encalhe, sem perda significativa de vendas

CASES

Recomendação de produtos no mundo real

Problema I: alguns produtos-chave incentivados em toda a base de PDVs*, outros a critério do vendedor

Problema II: vendedor possui a inteligência sobre o PDV. O que acontece se ele sai?

Criamos modelo para sugerir produtos e quantidades a cada visita, baseado em PDVs semelhantes

Resultado: aumento de lucro de **12%** nas áreas em que o projeto foi empregado

*pontos de venda

CASES

Precificação em rede de farmácias.
Ganhos apurados na casa de 7 dígitos.

Objetivo: encontrar o **preço ótimo** de 14.000 SKUs em cada uma das 400+ lojas da rede de farmácias

Precificamos tanto SKUs **Farma** quanto **não-Farma**: linha capilar, higiene bucal, desodorantes, fraldas etc.

Último mês: grupo de tratamento (Big Data precifica) obteve **lucro bruto 4,4% maior** que grupo de controle

O QUE FAZEMOS

Big Data

DADOS DA BIG DATA

DADOS DO CLIENTE

CRIAMOS ALGORITMOS DE INTELIGÊNCIA ARTIFICIAL

ACTIONABLE INSIGHT PARA O CLIENTE

TEMOS MAIS DE 17.000 VARIÁVEIS

Big Data

ALGUNS DOS DADOS GEORREFERENCIADOS PARA ANÁLISE

- Renda (algoritmo propr. → atualização mensal)
- Emprego
- Clima
- Educação Básica e Superior
- Tweets (> 2 bilhões), Facebook
- Saúde
- Frota e emplacamento de carros
- Religião
- Transporte coletivo
- 27 milhões de CNPJs
- Antenas de telefonia celular
- Criminalidade
- Perfil ideológico
- Densidade demográfica
- Padrões de trânsito
- Preços de 20.000+ SKUs em supermercados

NOSSOS PRODUTOS

PRICEO

- Encontrar o preço ideal por SKU em cada PDV, que maximiza lucro ou vendas
- Big data analytics + behavioral economics = ferramenta viva, constantemente melhorada

ONE-CLICK ORDER

- Determinar o sortimento ideal por PDV, com SKUs e quantidades
- Auto-calibrado constantemente
- Descobre e antecipa demanda do cliente final

OPTIMARC

- Marketing Mix Optimization
- Incorpora budget inteiro: mídia e trade
- Dashboard online com simulações real-time para praça, mídia e veículo

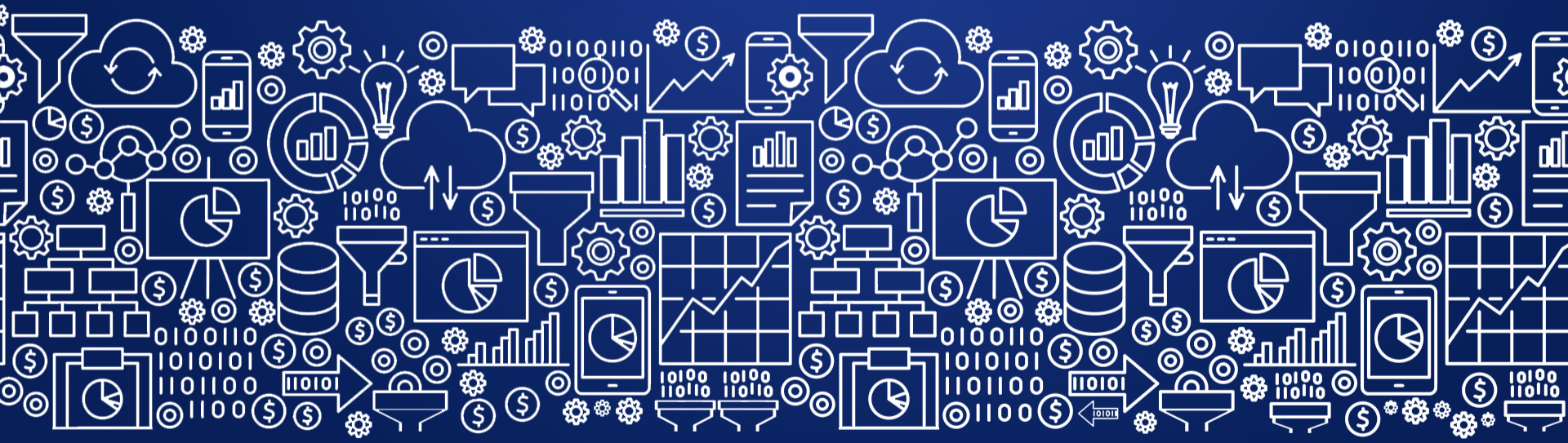
TARGATOM

- Definição de meta de faturamento ou lucro por PDV
- Levando em consideração variáveis do local em que PDV está inserido

AGENDA

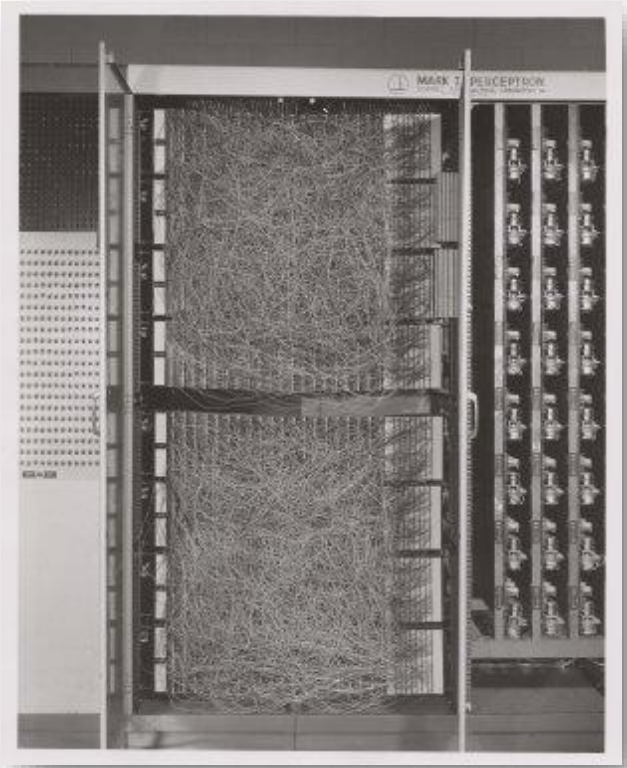
1. Histórico
2. Tipos de Machine Learning
3. Aprendizado Supervisionado
4. Overfitting e Underfitting
5. Perceptron Multicamadas
6. Árvores de Decisão

HISTÓRICO



PERCEPTRON

- Surgiu em 1958
- Desenvolvido por Frank Rosenblatt
- Finalidade: reconhecimento de padrões
- Baseado no modelo neural de McCulloch & Pitts
- Simulado em software, primeiramente
- Depois, virou hardware: *Mark I Perceptron*



PERCEPTRON

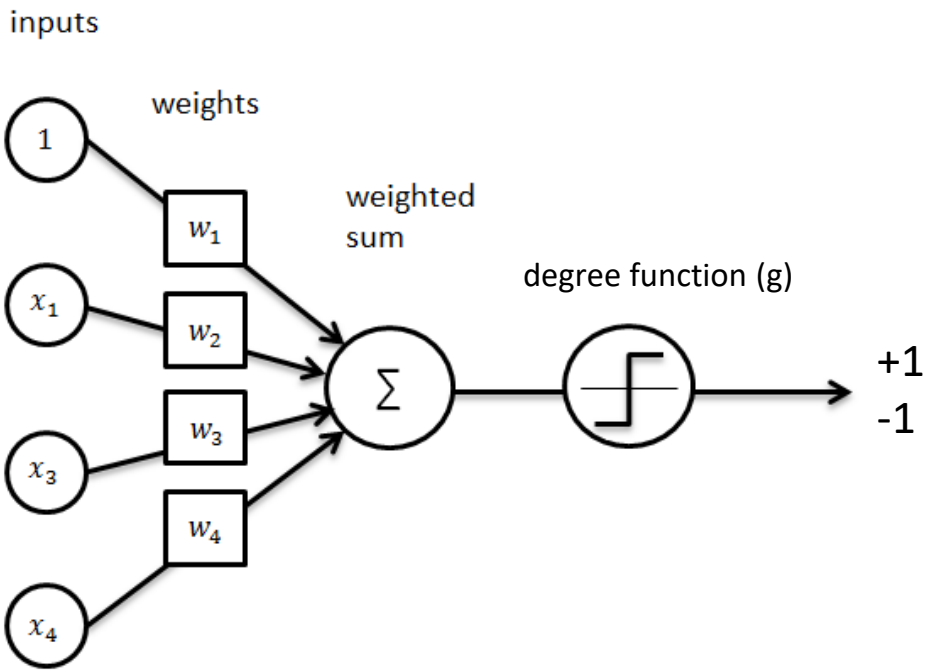
ARCHIVES | 1958

NEW NAVY DEVICE LEARNS BY DOING; Psychologist Shows Embryo of Computer Designed to Read and Grow Wiser

SPECIAL TO THE NEW YORK TIMES JULY 8, 1958

WASHINGTON, July 7 (UPI) -- The Navy revealed the embryo of an electronic computer today that it expects will be able to walk, talk, see, write, reproduce itself and be conscious of its existence.

PERCEPTRON

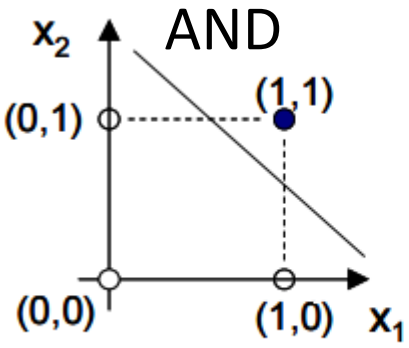


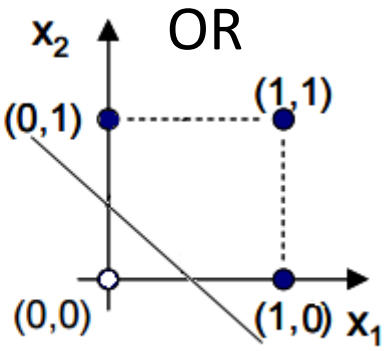
$$y = g\left(\sum_{n=1}^{d} x_n w_n\right)$$

Implementa superfície de separação linear

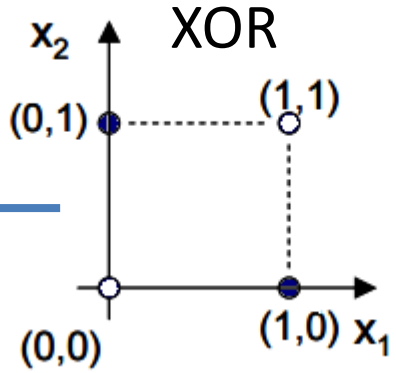
PERCEPTRON

- Limitado à separação linear
- Não reconhece padrões como o do XOR
- Hype causou o AI Winter





não é linearmente separável !



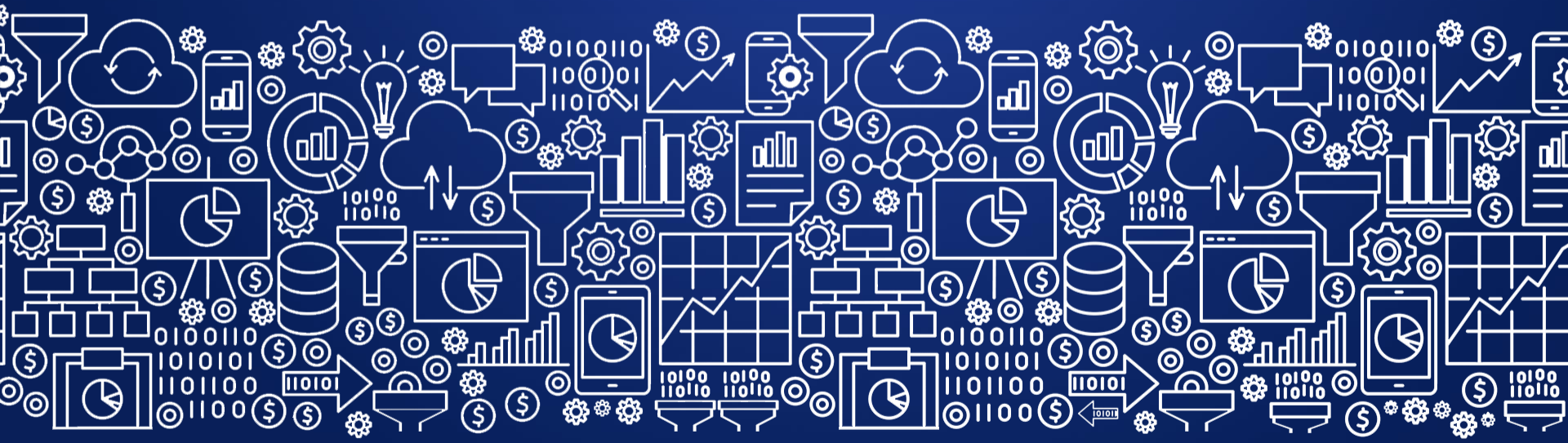
LINHA DO TEMPO

1958	Perceptron
1960-1965	Teoria da Complexidade Algorítmica (Solomonoff, Chaitin, Kolmogorov)
1962	Análise teórica do Perceptron
1965-1980	Teoria do Aprendizado Estatístico (Vapnik e Chervonenkis)
1967	Aparição do algoritmo k-Means
1967	Consistência do algoritmo k-NN

AI WINTER!

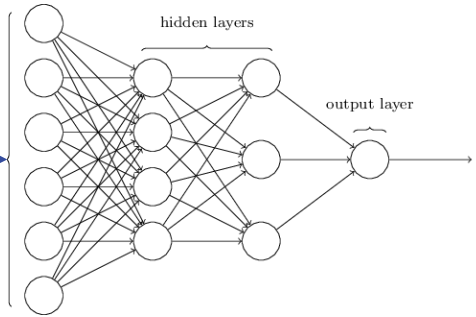
1984	Árvores de Decisão e Regressão (Breiman, Friedman)
1986	Ressureição das Redes Neurais: algoritmo Backpropagation
1989	Redes Neurais para reconhecer Zip Codes (CEPs)
1992-1995	Support Vector Machines (SVM)
1995	Boosting (ensemble)
1995	Aplicações: text mining, genômica, detecção de fraudes…
1998	Filtragem de Spam
2002	Análise de sentimento
2005	Carros autônomos
2008	Deep Learning e aplicações (imagem, voz, tradução)

TIPOS DE ML

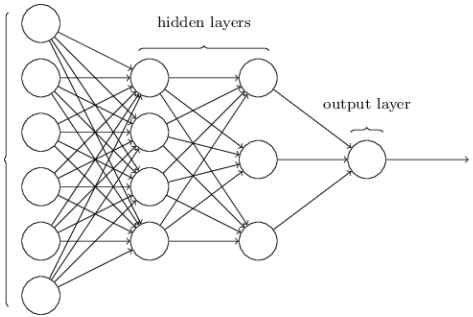


APRENDIZADO SUPERVISIONADO

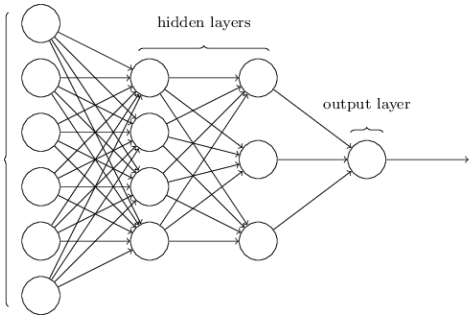
Algoritmo de aprendizado



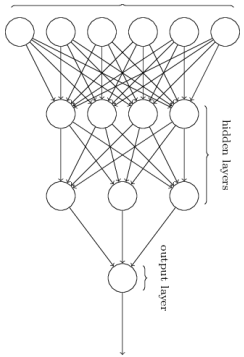
APRENDIZADO SUPERVISIONADO



gato



cachorro



gato e cachorro

APRENDIZADO SUPERVISIONADO

- Exemplos de treinamento $(x_1,y_1),\ldots,(x_n,y_n)$,…
- Encontrar $f(x)$ tal que $y_i \sim f(x_i)$
- Classificação: y é discreto
- Regressão: y é um número real

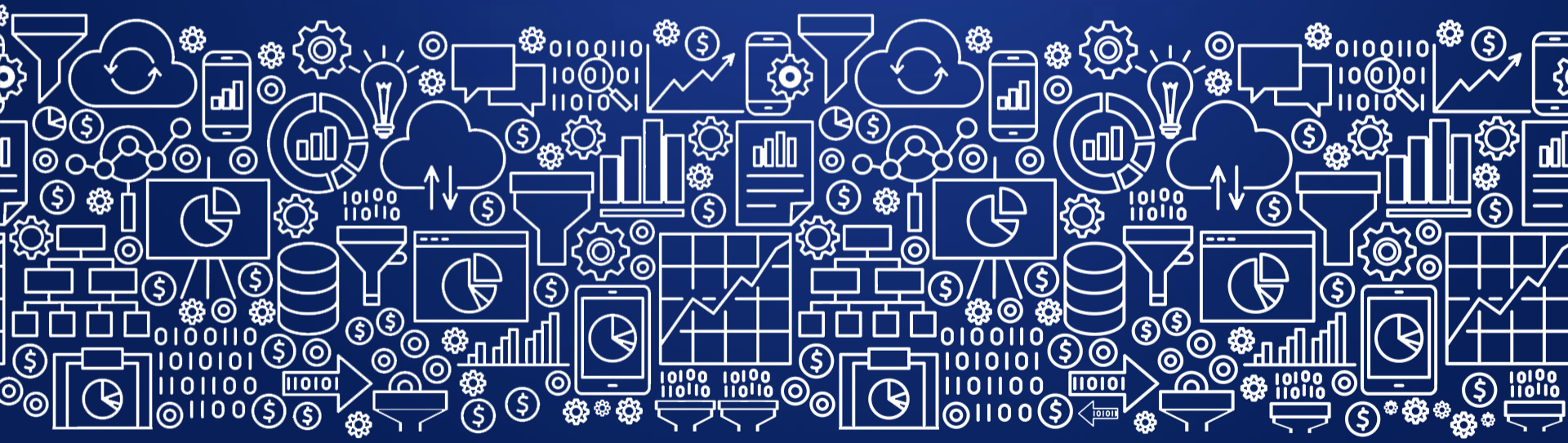
APRENDIZADO NÃO-SUPERVISIONADO

- Exemplos $x_1,\ldots,x_n$
- Não-categoria de aprendizado : complemento de aprendizado supervisionado
- Exemplos : clustering, redução de dimensionalidade, aprendizado de representações
- Descoberta de padrões interessantes nos dados

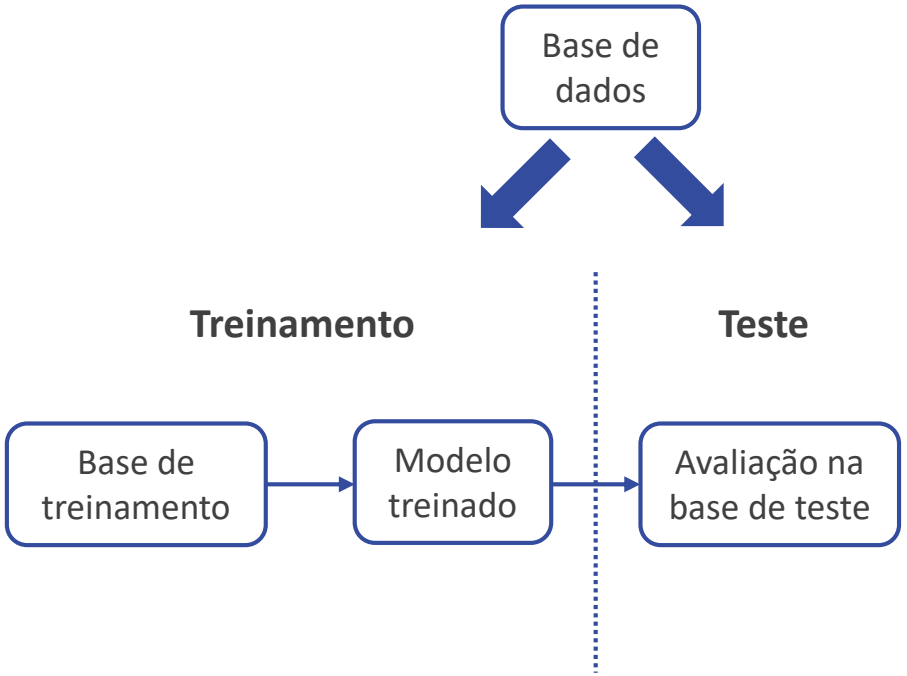
APRENDIZADO POR REFORÇO

- Encontrar estratégia (ações) que maximiza retorno (*reward*) e minimiza punição
- Delayed feedback: retorno de uma ação não é conhecido
- *Exploration* vs. *exploitation*
- Problema típico: multi-armed bandits;
- Aplicações: robótica, jogos

APRENDIZADO SUPERVISIONADO



ESTRUTURA


Base de dados
Treinamento
Teste
Base de treinamento
Modelo treinado
Avaliação na base de teste

EXEMPLO : PERCEPTRON

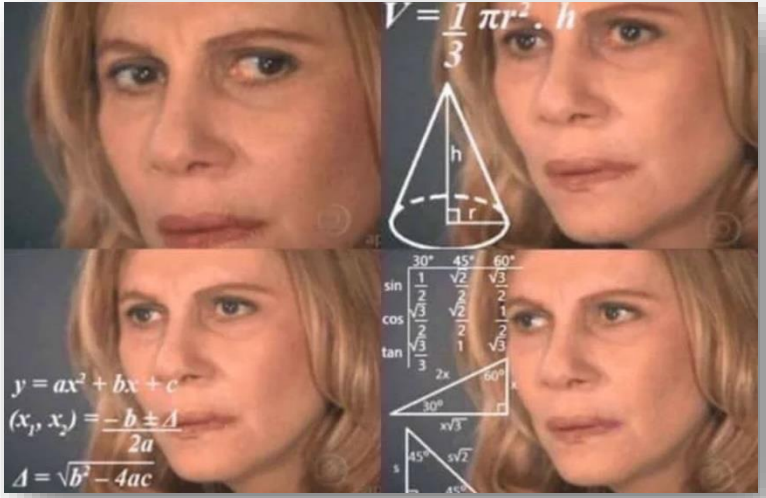
Algoritmo de treinamento começa com *w aleatório* e percorre os exemplos $(x_1,d_1),\ldots,(d_k,d_k)$ – x_i pertence a R^k e d_k a {-1,+1}:

1. Se o exemplo x_i é corretamente classificado, ou seja, o valor predito y_i é igual ao valor esperado d_i , o vetor de pesos w não sofre alterações

2. Se o exemplo x_i é incorretamente classificado, então w sofre alterações proporcionalmente a uma taxa de aprendizado η :

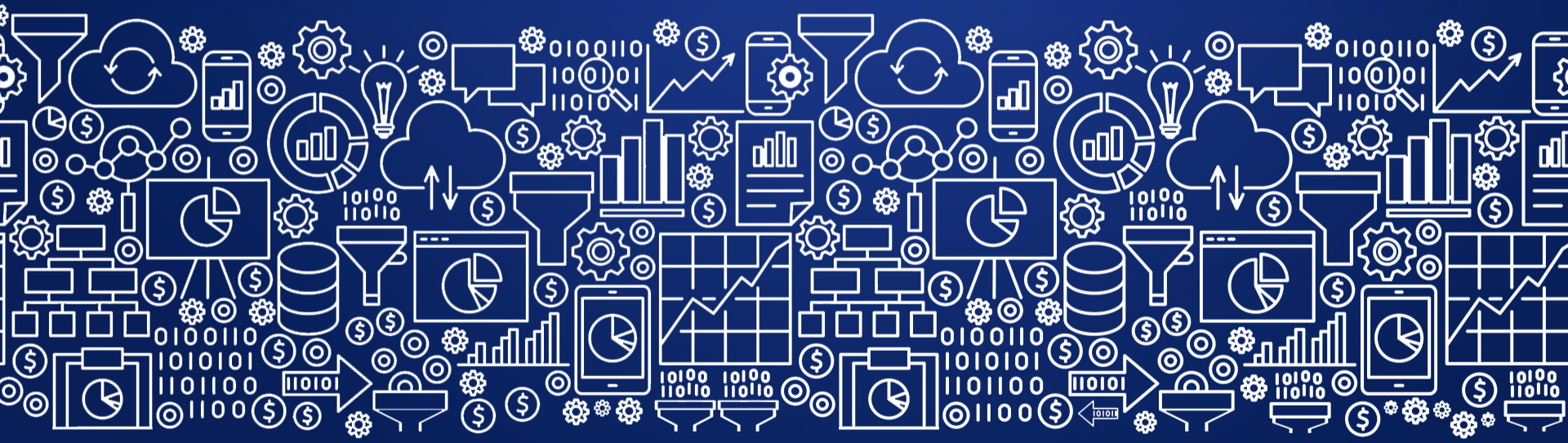
$$w \leftarrow w + \eta(d_i - y_i).\, x_i$$

REFLEXÃO...

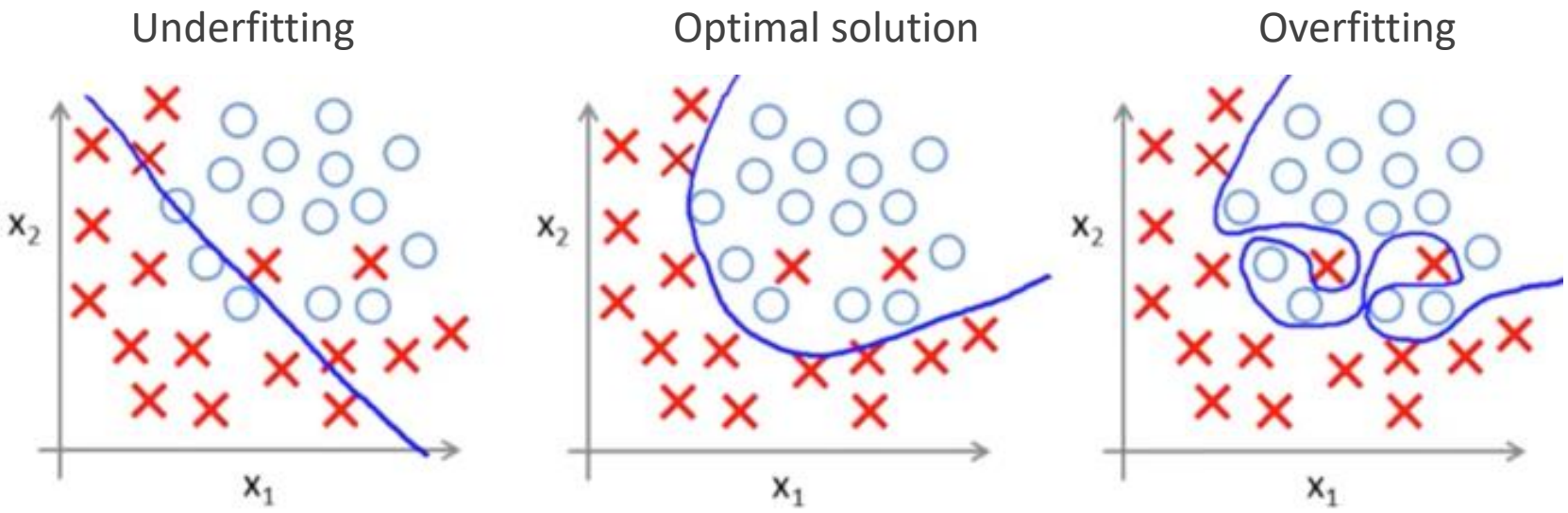


Bom desempenho durante o treinamento reflete um bom modelo ?

OVERFITTING E UNDERFITTING

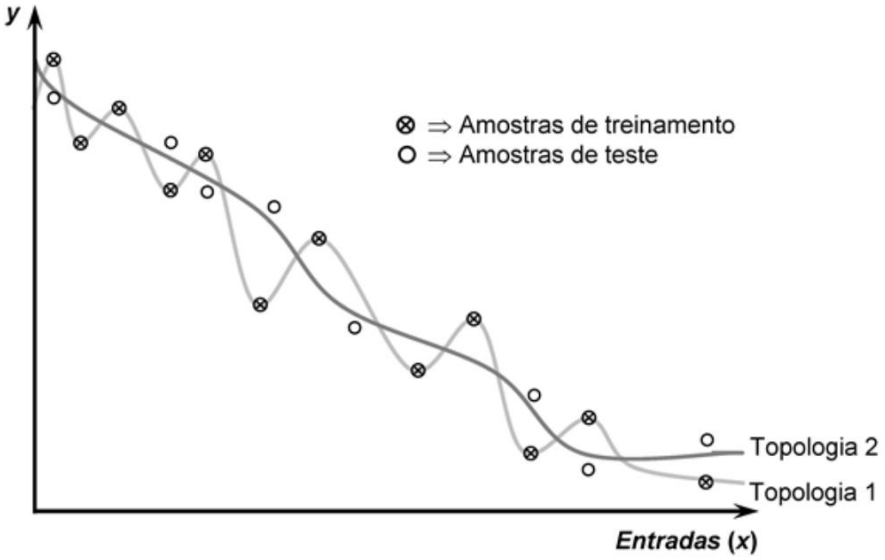


OVERFITTING E UNDERFITTING NA CLASSIFICAÇÃO



OVERFITTING

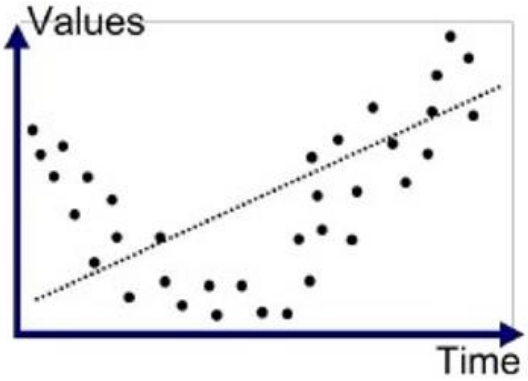
- Super ajuste de parâmetros do modelo : memorização vs generalização
- Em redes neurais por exemplo : aumento indiscriminado de camadas e neurônios
- Durante treinamento o erro tende a ser baixo e durante o teste o erro tende a ser alto



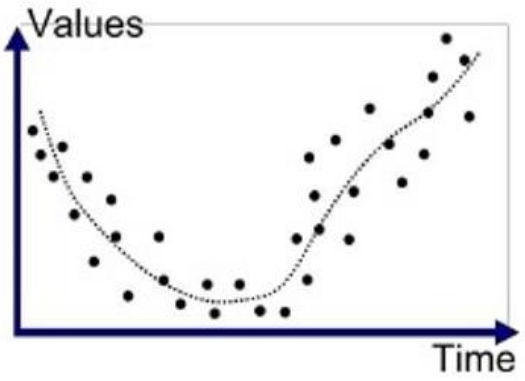
Topologia 1 : com overfitting
Topologia 2 : sem overfitting

UNDERFITTING

- Sub ajuste parâmetros do modelo e o mesmo não é capaz de aprender adequadamente
- Durante treinamento e teste o erro tende a ser baixo

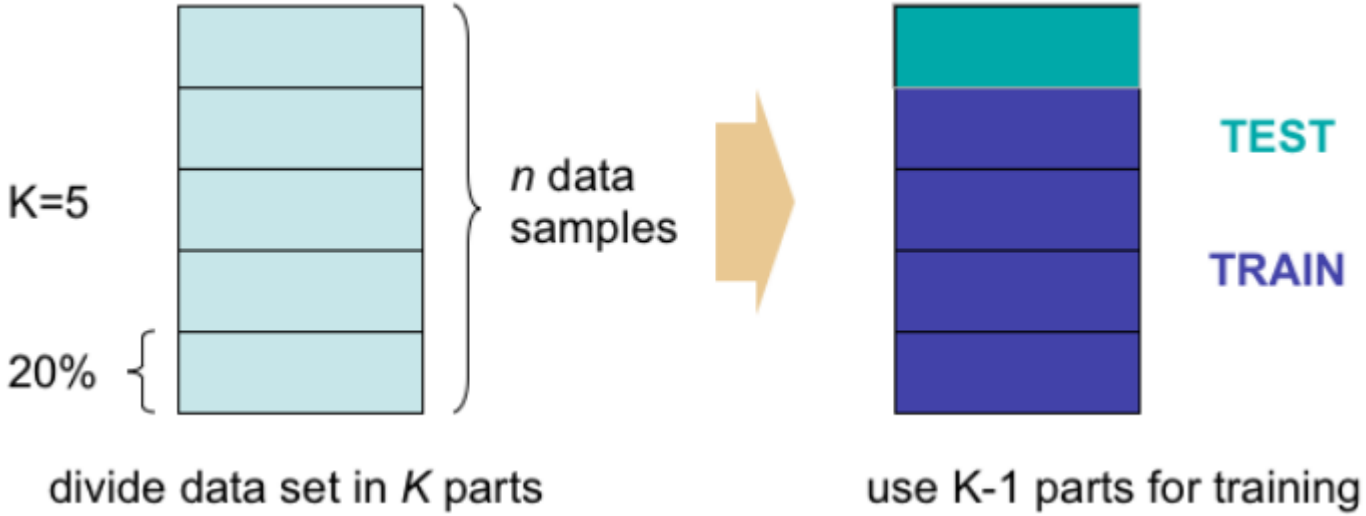


Underfitted

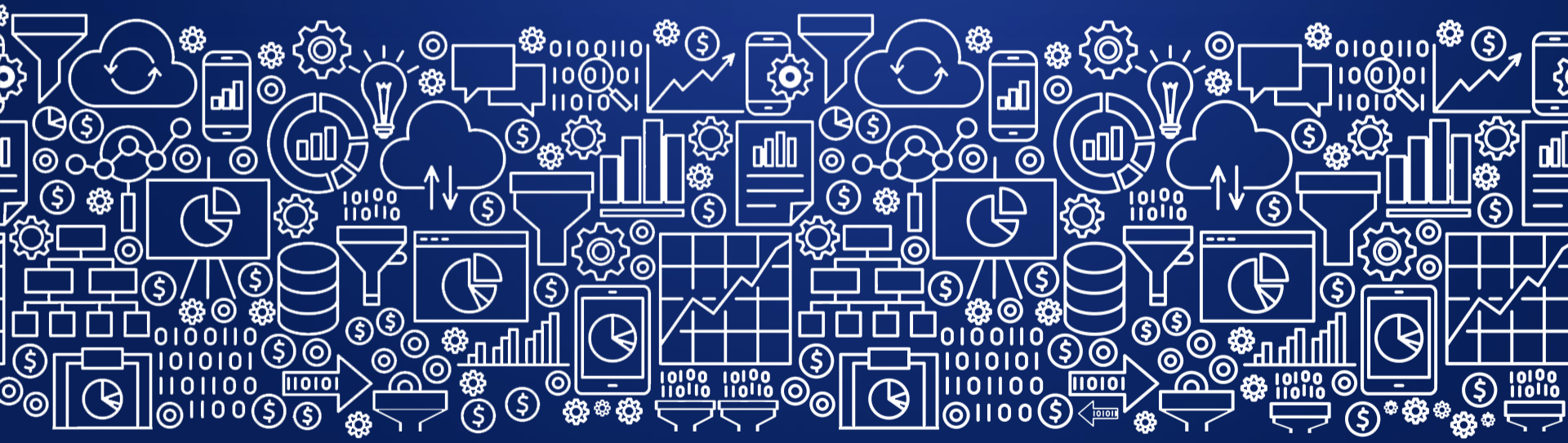


Good Fit/Robust

VALIDAÇÃO CRUZADA : K-FOLD

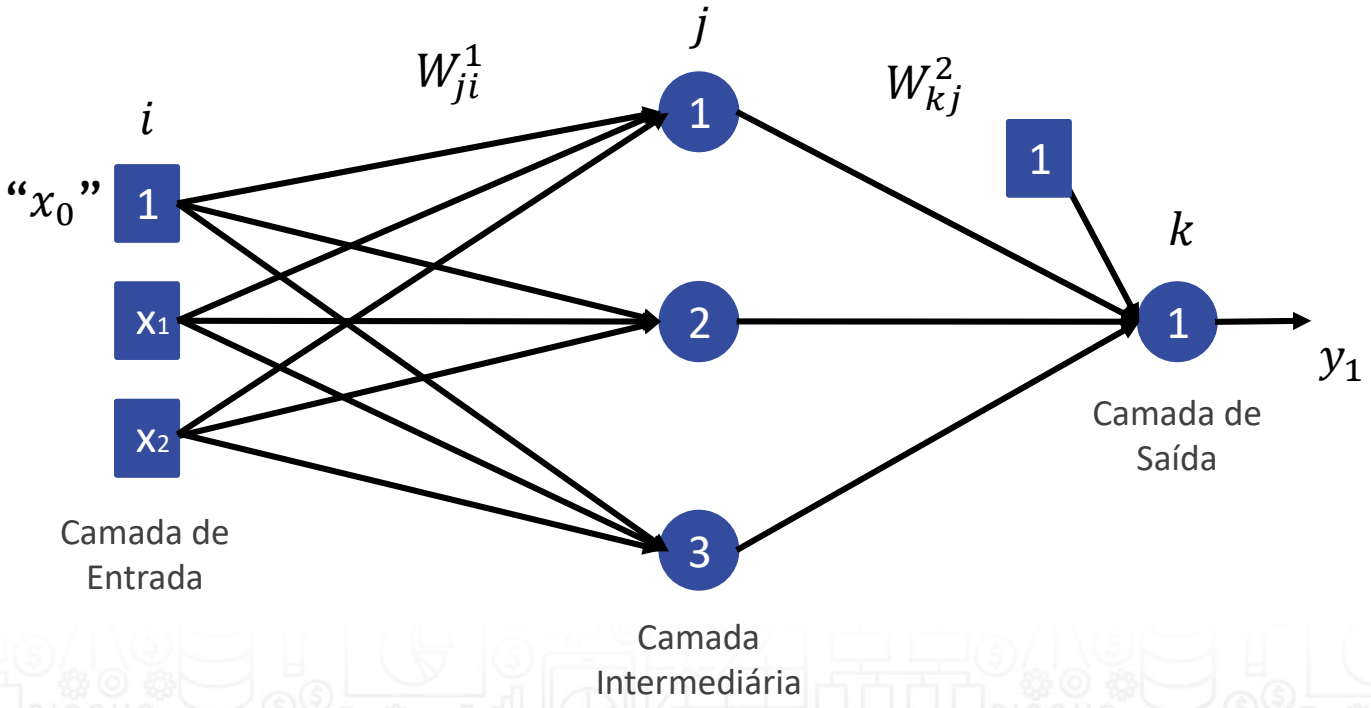


PERCEPTRON MULTICAMADAS



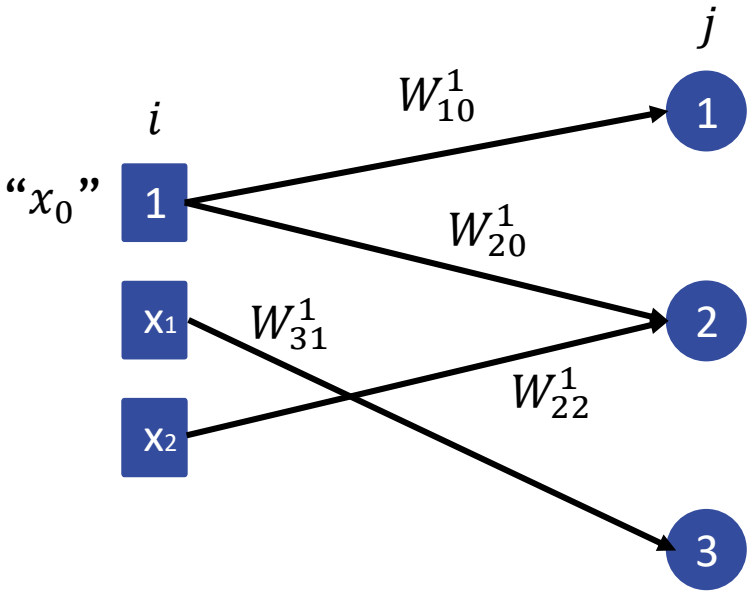
ESTRUTURA

- Dataset com duas entradas e uma saída (regressão)
- Topologia com uma camada escondida e 3 neurônios
- Wji é a matriz de pesos entre um elementos j e elementos i



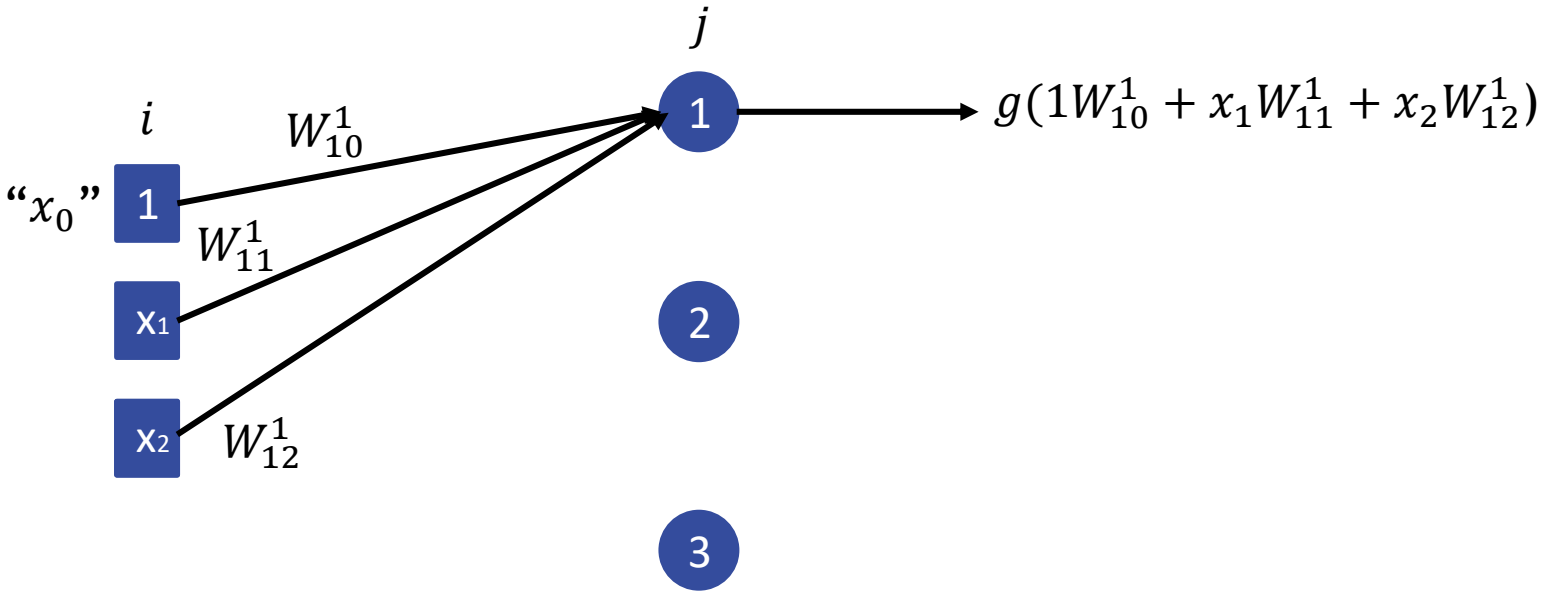
ESTRUTURA

- Exemplo de pesos da matriz W



ESTRUTURA

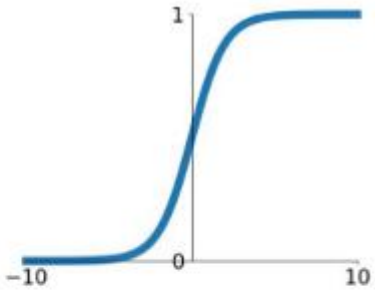
- Saída dos neurônios



FUNÇÕES DE ATIVAÇÃO

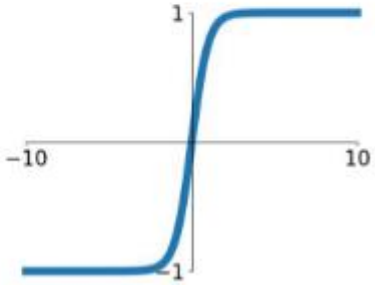
Sigmoid

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



tanh

$$\tanh(x)$$



NORMALIZAÇÃO E PADRONIZAÇÃO

- Motivação :
  - Ter a mesma escala no processo de treinamento
  - Saturação das funções de ativação

Normalização

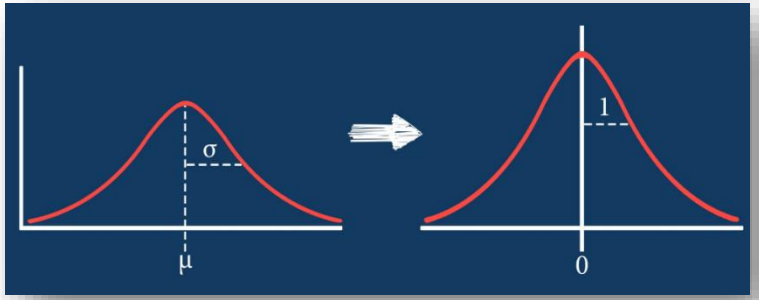
$$x_i^{max} \qquad 1$$

$$x_i \Longrightarrow x_{inorm}$$

$$x_i^{min} \qquad 0$$

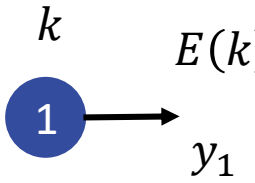
Padronização

$$x_{ipad} = \frac{x_i - \mu}{\sigma}$$



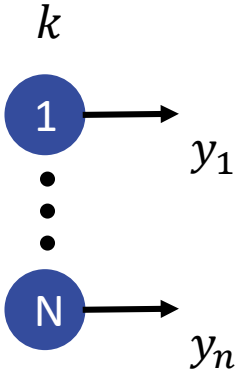
ATUALIZAÇÃO DOS PESOS

- E(k) : erro associado a uma amostra k : (x1, x2)
- d(k) : a saída esperada em relação à amostra k
- Pesos são atualizados em função do gradiente negativo do erro



$$E(k) = \frac{1}{2}(y_1 - d_1)^{\,2}$$

Para 1 saída

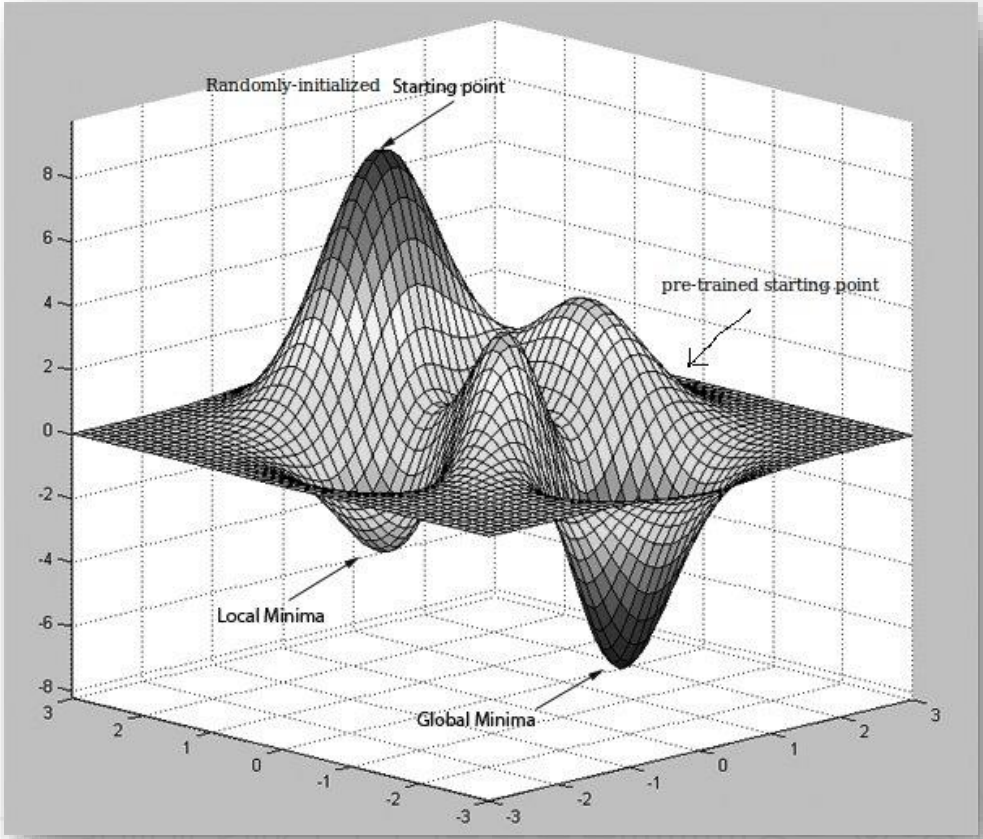


$$E(k) = \frac{1}{2}\sum_{n=1}^{N}(y_n(k) - d_n(k))^{\,2}$$

Para N saídas

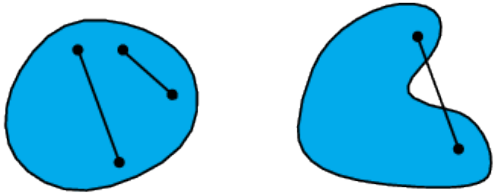
$$\boxed{W'_{ij} \propto -\nabla E}$$

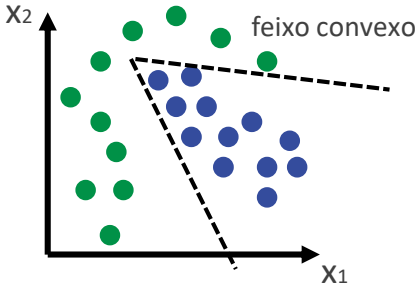
SUPERFÍCIE DE ERRO



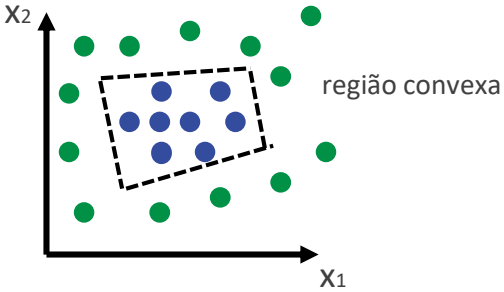
CLASSIFICAÇÃO

- Uma camada escondida é capaz de classificar problemas convexos (neurônios adicionam arestas)
- Problemas não convexos podem ser classificados com mais camadas escondidas
- Resolve o problema do XOR



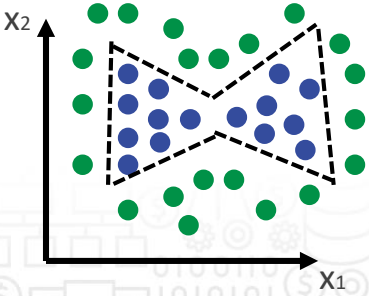


2 neurônios na camada escondida

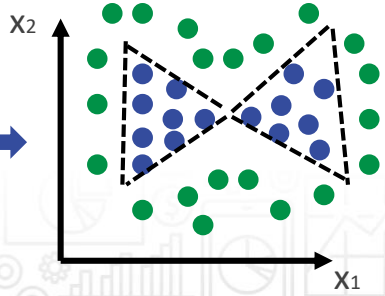


4 neurônios na camada escondida

região não convexa

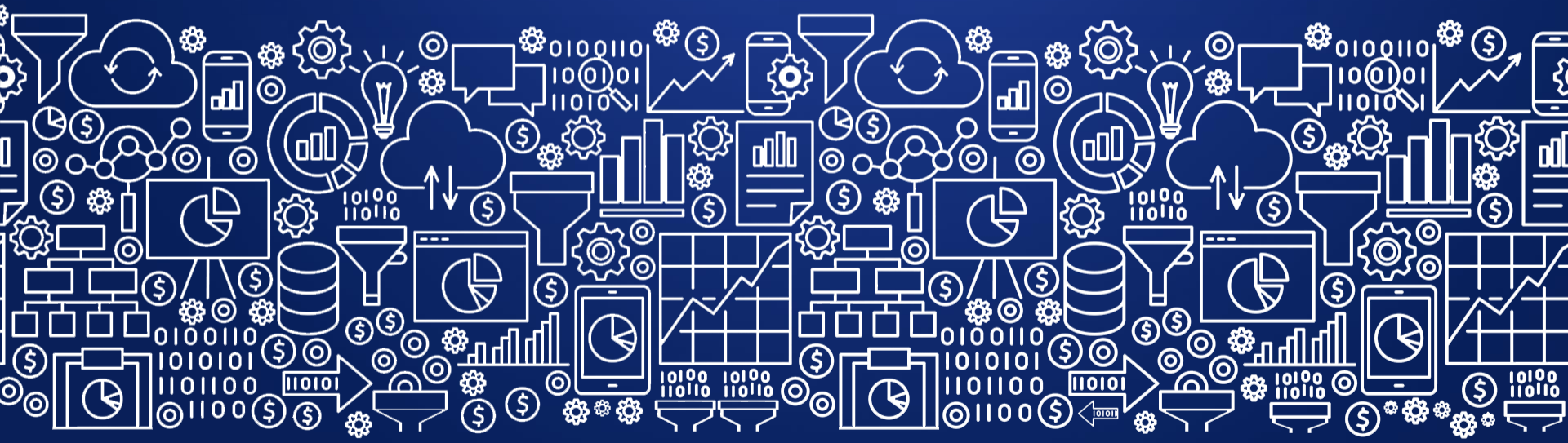


2 regiões convexas



2 camadas escondidas
(3 neurônios cada)

ÁRVORES DE DECISÃO

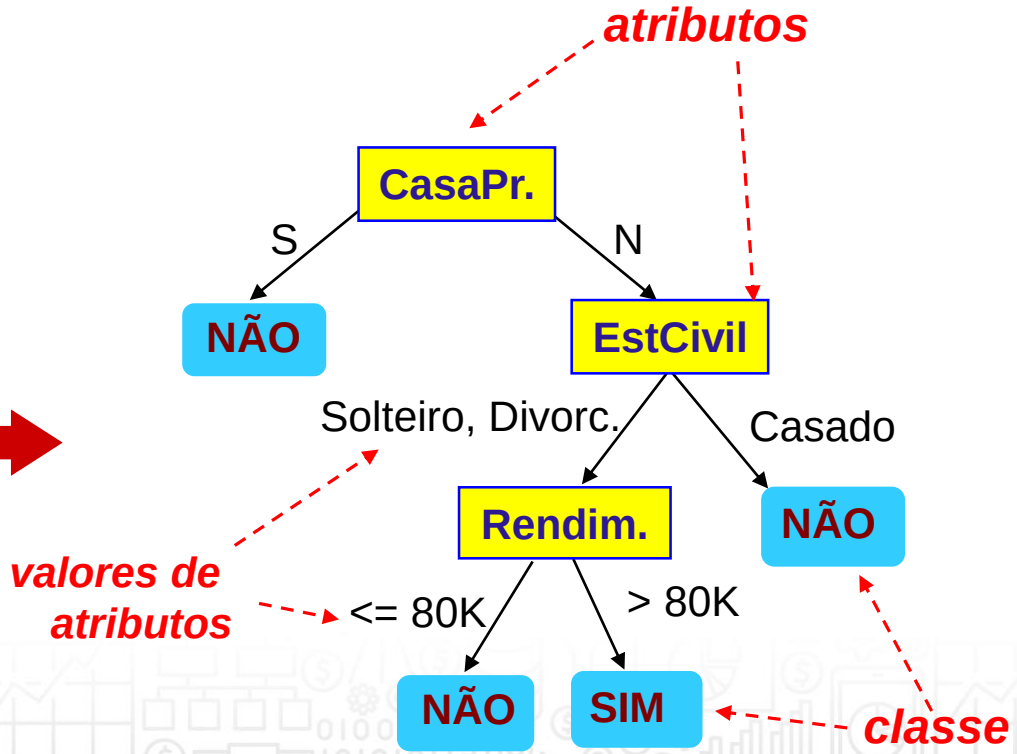


CARACTERÍSTICAS

- Sistema orientado a conhecimento (interpretável)
- Dividir para conquistar
- Nós representam atributos e arcos representam valores dos atributos

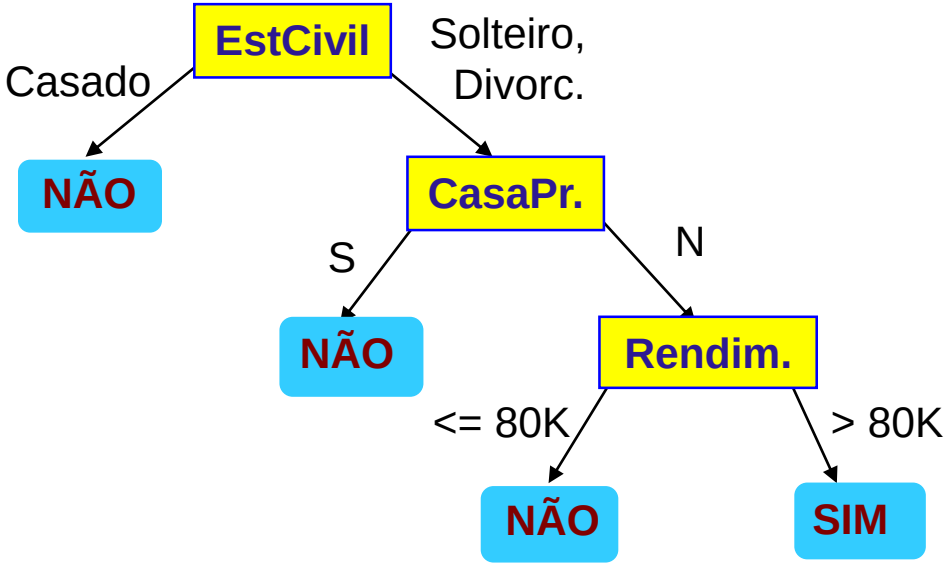
categórico categórico contínuo classe

Id	Casa própria	EstCivil	Rendim.	Mau Pagador
1	S	Solteiro	125K	NÃO
2	N	Casado	100K	NÃO
3	N	Solteiro	70K	NÃO
4	S	Casado	120K	NÃO
5	N	Divorc.	95K	SIM
6	N	Casado	60K	NÃO
7	S	Divorc.	220K	NÃO
8	N	Solteiro	85K	SIM
9	N	Casado	75K	NÃO
10	N	Solteiro	90K	SIM



CARACTERÍSTICAS

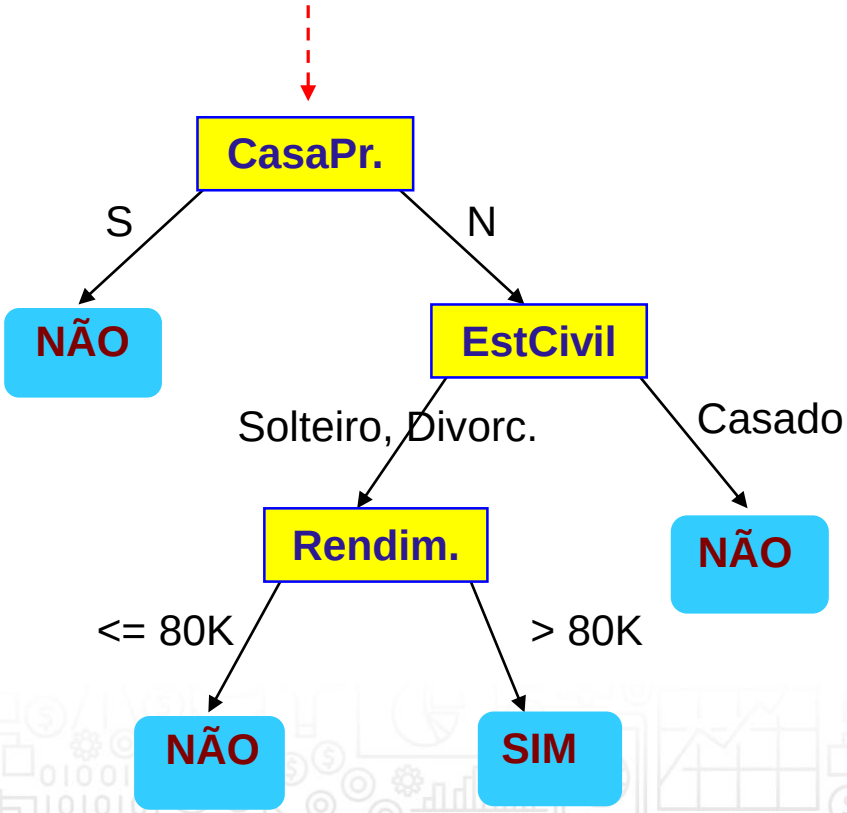
Id	Casa própria	EstCivil	Rendim.	Mau Pagador
	categórico	categórico	continuo	classe
1	S	Solteiro	125K	NÃO
2	N	Casado	100K	NÃO
3	N	Solteiro	70K	NÃO
4	S	Casado	120K	NÃO
5	N	Divorc.	95K	SIM
6	N	Casado	60K	NÃO
7	S	Divorc.	220K	NÃO
8	N	Solteiro	85K	SIM
9	N	Casado	75K	NÃO
10	N	Solteiro	90K	SIM



Pode haver mais de uma árvore para o mesmo conjunto de dados!!!

PREVISÃO COM O MODELO

Comece pela raíz da árvore!



Dado para teste

Casa Própria	Estado Civil	Rendim.	Mau pagador
N	Casado	80K	?

ALGORITMOS

- Há diversos algoritmos para construção de árvores que fazem uso de diferentes funções objetivas
- Ex de algoritmos : ID3, C3.5, CART, SPRINT
- Ex de funções objetivas : entropia, índice de gini

Entropia

$$E(S) = -\sum_{i=1}^{n} p_i log_2 p_i$$

- S : número de registros
- n : número de valores possíveis de cada classe
- pi : proporção das amostras de cada classe i em relação ao total de amostras

Ganho de informação

$$G(S,A) = E(S) - \sum_{i=1}^{n} \frac{|S_V|}{|S|} E(s)$$

- G(S,A) : ganho do atributo a sobre o conjunto S
- Sv : subconjunto S para um valor do atributo A
- |Sv| : número de elementros de Sv
- |S| : número de elementos de S

ID3 : NÓ RAIZ

caso	montante	idade	salário	conta	empréstimo
1	médio	sênior	baixo	sim	***não***
2	médio	sênior	baixo	não	***não***
3	baixo	sênior	baixo	sim	***sim***
4	alto	média	baixo	sim	***sim***
5	alto	jovem	alto	sim	***sim***
6	alto	jovem	alto	não	***não***
7	baixo	jovem	alto	não	***sim***
8	médio	média	baixo	sim	***não***
9	médio	jovem	alto	sim	***sim***
10	alto	média	alto	sim	***sim***
11	médio	média	alto	não	***sim***
12	baixo	jovem	baixo	não	***sim***
13	baixo	sênior	alto	sim	***sim***
14	alto	média	baixo	não	***não***

ID3 : NÓ RAIZ

variável dependente

caso	montante	idade	salário	conta	empréstimo
1	médio	sênior	baixo	sim	***não***
2	médio	sênior	baixo	não	***não***
3	baixo	sênior	baixo	sim	***sim***
4	alto	média	baixo	sim	**sim**
5	alto	jovem	alto	sim	**sim**
6	alto	jovem	alto	não	***não***
7	baixo	jovem	alto	não	***sim***
8	médio	média	baixo	sim	***não***
9	médio	jovem	alto	sim	***sim***
10	alto	média	alto	sim	***sim***
11	médio	média	alto	não	***sim***
12	baixo	jovem	baixo	não	***sim***
13	baixo	sênior	alto	sim	***sim***
14	alto	média	baixo	não	***não***

$$\boxed{E(S) = -\sum_{i=1}^{n} p_i log_2 p_i}$$

- Total de 14 amostras : S = 14
- $p_{não} = 5/14$
- $p_{sim} = 9/14$

$$E(S) = \left(\frac{5}{14} log_2 \frac{5}{14} + \frac{9}{14} log_2 \frac{9}{14}\right)$$

$$\boxed{E(S) = 0{,}94}$$

MONTANTE

- Montante = médio

montante	empréstimo
médio	*não*
médio	*não*
baixo	*sim*
alto	sim
alto	sim
alto	*não*
baixo	*sim*
médio	*não*
médio	*sim*
alto	*sim*
médio	*sim*
baixo	*sim*
baixo	*sim*
alto	*não*

$$E(S) = -\sum_{i=1}^{n} p_i log_2 p_i$$

- Total de 5 amostras : S = 5
- $p_{não} = 3/5$
- $p_{sim} = 2/5$

$$E(S) = -\left(\frac{3}{5} log_2 \frac{3}{5} + \frac{2}{5} log_2 \frac{3}{5}\right) = 0{,}971$$

MONTANTE

- Montante = baixo

montante	empréstimo
médio	*não*
médio	*não*
baixo	*sim*
alto	sim
alto	sim
alto	*não*
baixo	*sim*
médio	*não*
médio	*sim*
alto	*sim*
médio	*sim*
baixo	*sim*
baixo	*sim*
alto	*não*

$$E(S) = -\sum_{i=1}^{n} p_i log_2 p_i$$

- Total de 4 amostras : S = 4
- $p_{não} = 0/4$
- $p_{sim} = 4/4$

Entropia 0 : fácil separar as classes

$$E(S) = -\left(\frac{0}{4} log_2 \frac{0}{4} + \frac{4}{4} log_2 \frac{4}{4}\right) = 0$$

MONTANTE

- Montante = alto

montante	empréstimo
médio	*não*
médio	*não*
baixo	*sim*
alto	sim
alto	sim
alto	*não*
baixo	*sim*
médio	*não*
médio	*sim*
alto	*sim*
médio	*sim*
baixo	*sim*
baixo	*sim*
alto	*não*

$$E(S) = -\sum_{i=1}^{n} p_i log_2 p_i$$

- Total de 5 amostras : S = 5
- $p_{não} = 2/5$
- $p_{sim} = 3/5$

$$E(S) = -\left(\frac{2}{5} log_2 \frac{2}{5} + \frac{3}{5} log_2 \frac{3}{5}\right) = 0{,}971$$

MONTANTE

- Ganho de informação

$$G(S,A) = E(S) - \sum_{i=1}^{n} \frac{|S_V|}{|S|} E(s)$$

E(Montante = Médio)	E(Montante = Baixo)	E(Montante = Alto)	E(S)
0,971	0	0,971	0,94

$$G(S, Montante) = 0{,}94 - \left(\frac{5}{14}0{,}971 + \frac{4}{14}0 + \frac{5}{14}0{,}971\right) = 0{,}246$$

DEMAIS VARIÁVEIS

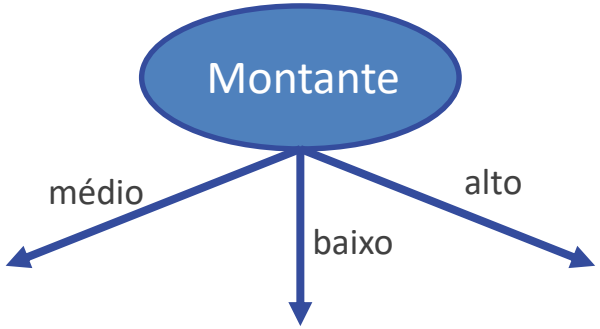
Maior ganho de informação!

$$\boxed{G(S, Montante) = 0{,}246}$$

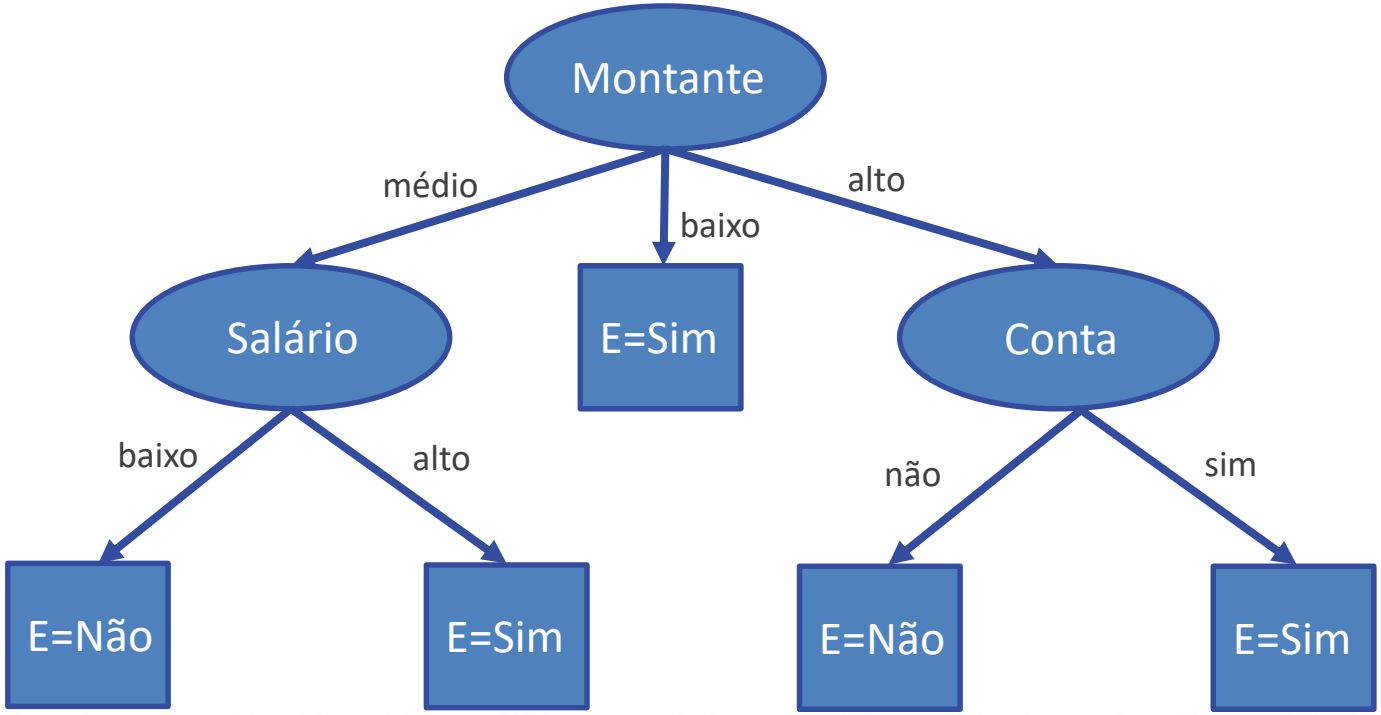
$$G(S, Idade) = 0{,}049$$

$$G(S, Salário) = 0{,}151$$

$$G(S, Conta) = 0{,}047$$

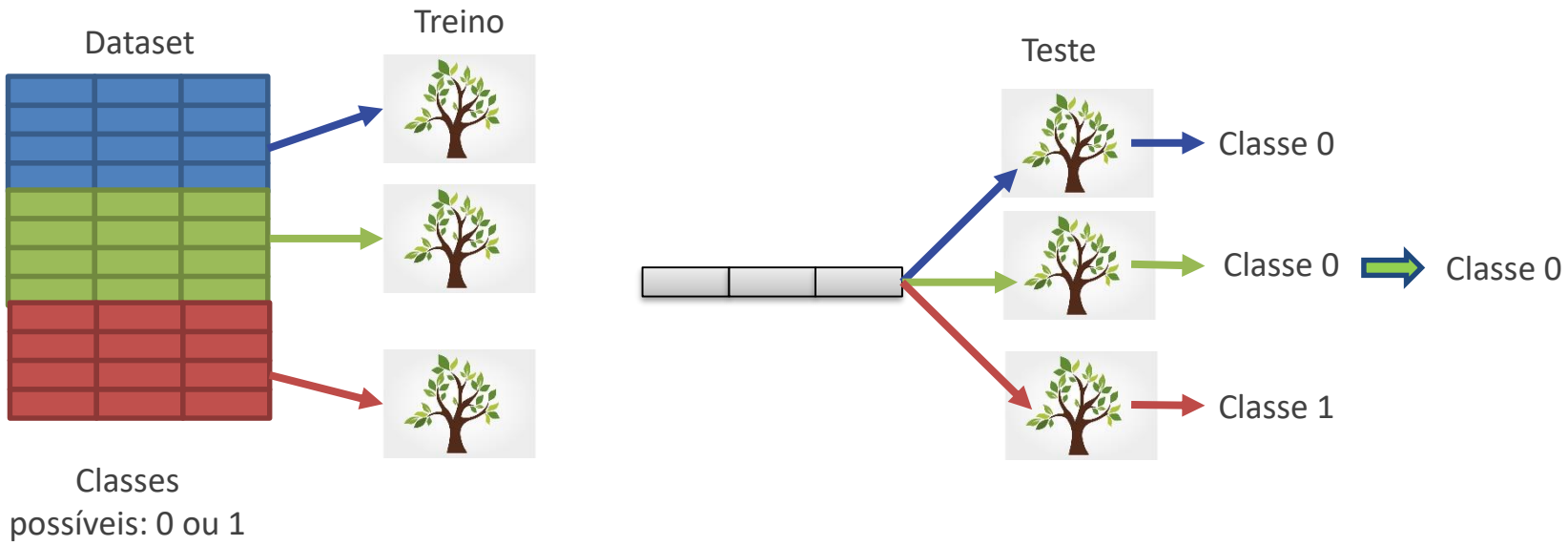


CONTINUANDO...



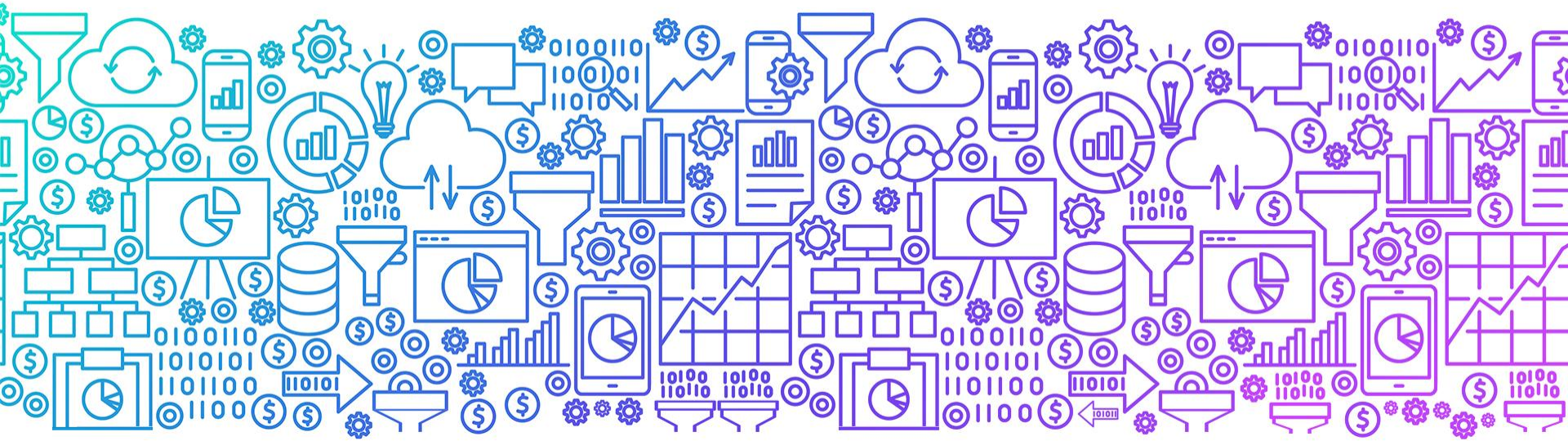
RANDOM FOREST

- Algoritmo de Ensemble
- Utiliza bagging de árvores de decisão : faz uma combinação das saídas de cada árvore treinada



PRA SABER MAIS...

- Lista dataelixir.com
- Lista datahackers.com.br
- Kaggle
- Coursera



WE’RE HIRING !

E-mails:

- igor.braga@bigdata.com.br
- igor.martinelli@bidata.com.br
- mateus.pedrino@bigdata.com.br

Big Data

Processo Seletivo:

- tinyurl.com/estagiobigdata
- tinyurl.com/dsbigdata

WWW.BIGDATA.COM.BR