

# Modelagem Preditiva Avançada

Rafael Lychowski

## MAPA CONCEITUAL DA DISCIPLINA

### PRIMEIRO DIA

### SEGUNDO DIA

MANHÃ

- Revisão Modelagem Preditiva
- Estudo de Caso

- SVM
- Estudo de Caso
- Redes Neurais
- Estudo de Caso

TARDE

- Combinação
- Estudo de Caso

- Algoritmos Genéticos
- Estudo de Caso
- Trabalho Final

## MAPA CONCEITUAL DA DISCIPLINA

### PRIMEIRO DIA

- Revisão Modelagem Preditiva
- Estudo de Caso

### TERCEIRO DIA

- SVM
- Estudo de Caso
- Redes Neurais
- Estudo de Caso

### SEGUNDO DIA

- Combinação
- Estudo de Caso

### QUARTO DIA

- Algoritmos Genéticos
- Estudo de Caso

### QUINTO DIA

- Trabalho Final

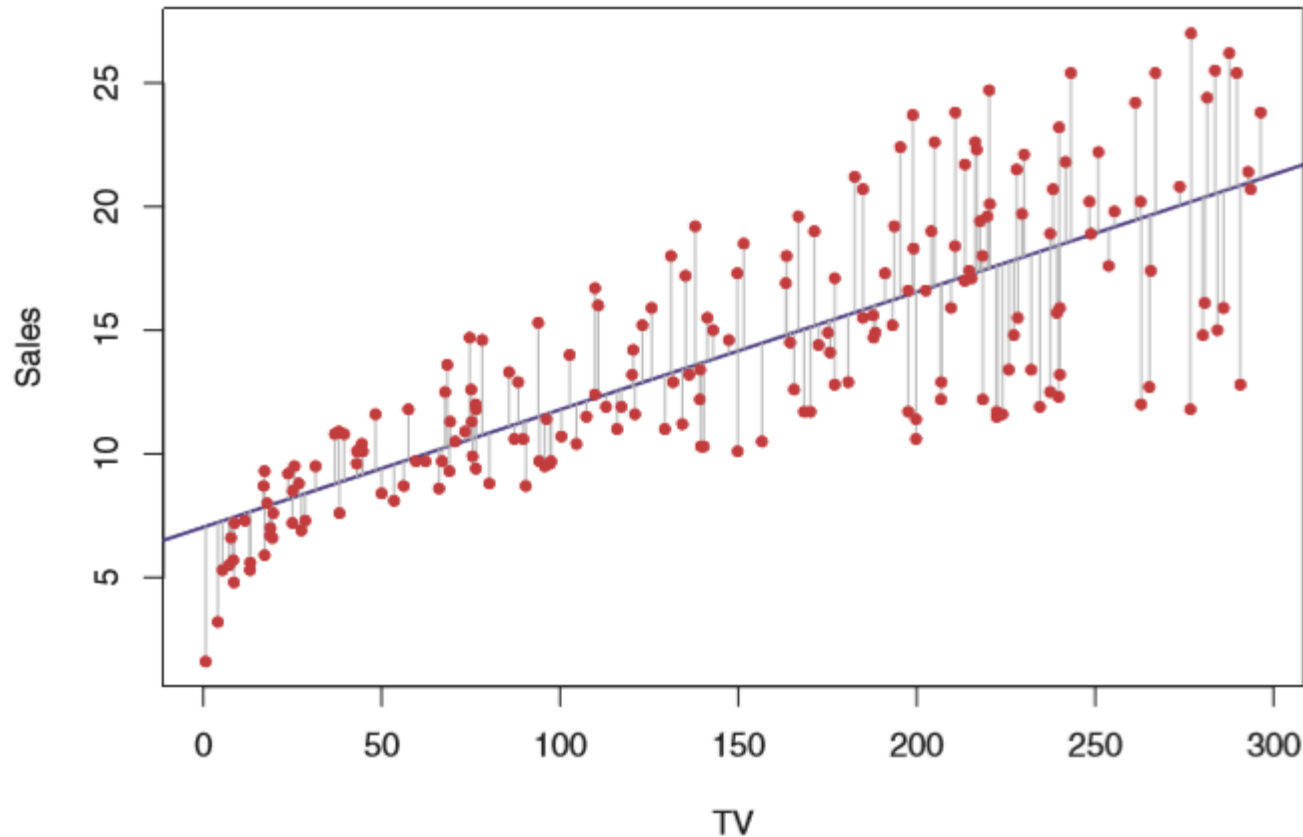
## CRISP – DM (Cross Industry Standard Process for Data Mining)



Método	Sub Método	Objetivo	Caso de Uso	Algoritmos
<b>Supervisionados</b>  Para cada conjunto de entrada existe um valor alvo correspondente	Regressão	Estimar uma variável contínua. Forecast, Time series	<ul style="list-style-type: none"> <li>• Forecast da demanda de compras</li> <li>• Predição da quantidade de chuva</li> </ul>	<ul style="list-style-type: none"> <li>• Linear Regression</li> <li>• Neural networks</li> <li>• Decision trees</li> </ul>
	Classificação	Estimar uma variável discreta	<ul style="list-style-type: none"> <li>• Prever a quebra de equipamentos</li> <li>• Risco de crédito</li> </ul>	<ul style="list-style-type: none"> <li>• Logistic Regression</li> <li>• SVMs</li> <li>• Neural Networks</li> <li>• Decision Trees</li> </ul>
<b>Não Supervisionados</b>  Encontrar as relações entre diferentes entradas sem uma variável alvo definida	Clustering	Identificar objetos similares	<ul style="list-style-type: none"> <li>• Segmentação de clientes (marketing)</li> <li>• Análise de Redes Sociais</li> </ul>	<ul style="list-style-type: none"> <li>• K-Means</li> <li>• DBSCAN</li> <li>• HDBSCAN</li> <li>• Hierarchical Clustering</li> </ul>
	Redução de Dimensionalidade	Reduzir a complexidade dos dados	<ul style="list-style-type: none"> <li>• Sistemas de Recomendação (Netflix, Amazon)</li> <li>• Processamento de Linguagens Naturais</li> </ul>	<ul style="list-style-type: none"> <li>• PCA, SVD, ALS</li> <li>• Latent dirichlet allocation</li> <li>• t-SNE, MDS</li> </ul>

## Regressão Linear

$$Y = \beta_0 + \beta_1 X + \epsilon.$$



## Regressão Linear

1. Existe alguma relação entre a variável de input e de output ?
2. O quão essa relação é forte ?
3. Qual variável contribui mais ? (importância)
4. Com qual acurácia podemos estimar o efeito de cada variável de input na variável de output ?
5. Com qual acurácia podemos estimar a variável de output ?
6. A relação das variáveis é linear ?
7. Existe sinergia entre as variáveis de input ?

## Regressão Linear

$$h_{\theta}(x) = \theta_0 + \theta_1 x$$

E se tivermos várias features ?

$$h_{\theta}(x) = \theta_0 + \theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n$$



## Bias - Variance Tradeoff

$$\text{Error}(x) = \underbrace{\left( \underbrace{E[\hat{f}(x)]}_{\text{predicted}} - \underbrace{f(x)}_{\text{true}} \right)^2}_{\text{Bias}^2} + \underbrace{E \left[ \underbrace{\hat{f}(x)}_{\text{predicted}} - \underbrace{E[\hat{f}(x)]}_{\text{average predicted value}} \right]^2}_{\text{Variance}} + \underbrace{\sigma_e^2}_{\text{irreducible error}}$$

Bias<sup>2</sup>

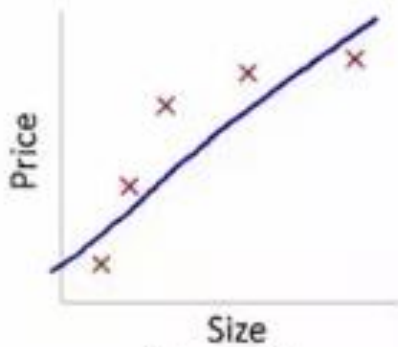
How much predicted values differ from true values.

Variance

How predictions made on the same value vary on different realizations of the model

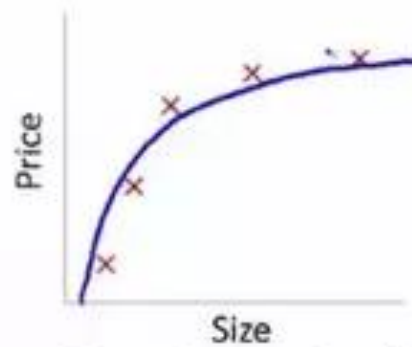
## Revisão

### Bias x Variance



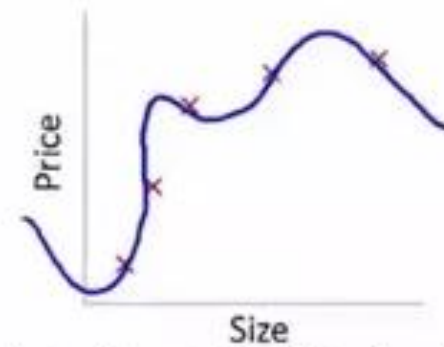
$$\theta_0 + \theta_1 x$$

High bias  
(underfit)



$$\theta_0 + \theta_1 x + \theta_2 x^2$$

"Just right"



$$\theta_0 + \theta_1 x + \theta_2 x^2 + \theta_3 x^3 + \theta_4 x^4$$

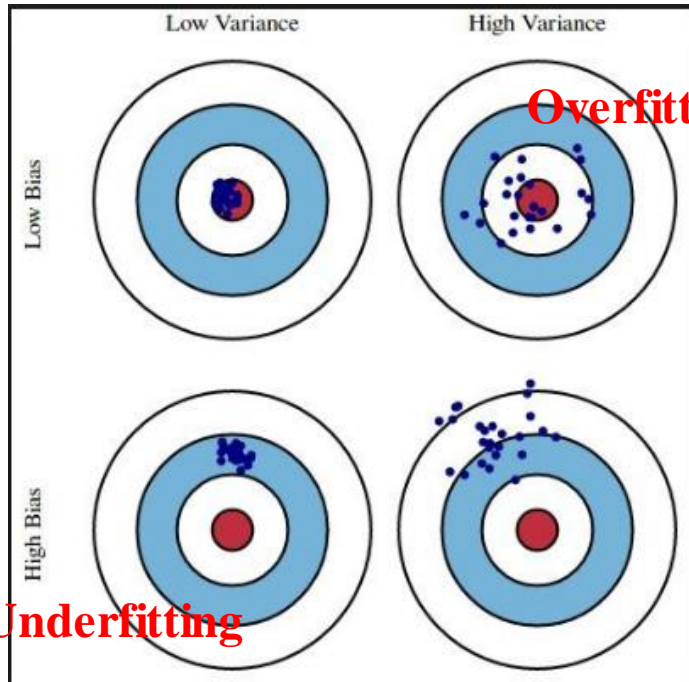
High variance  
(overfit)

- Adicionar variáveis (features)
- Aumentar complexidade do modelo

- Adicionar mais dados (base de treino)
- Diminuir variáveis (features)

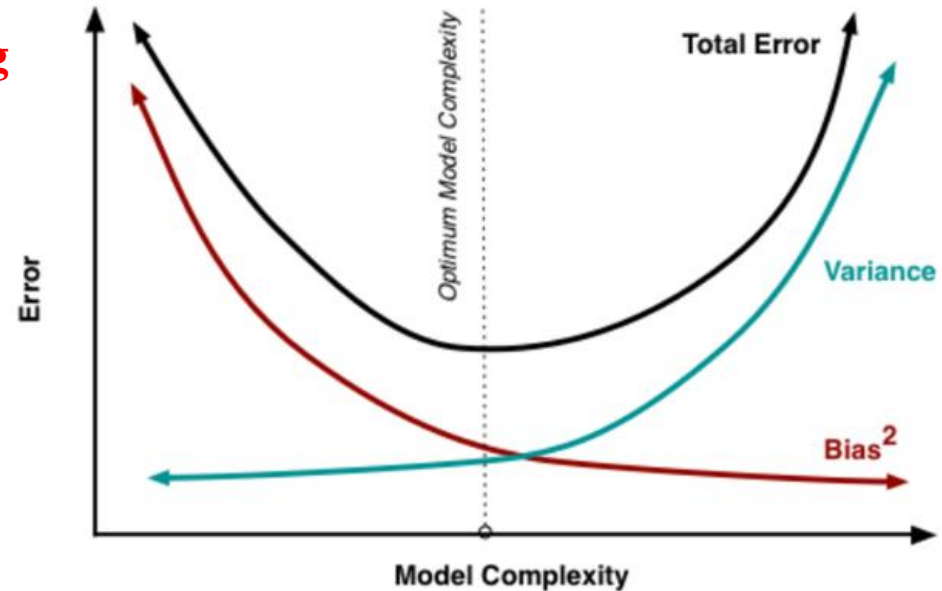
## Revisão

### Bias x Variance

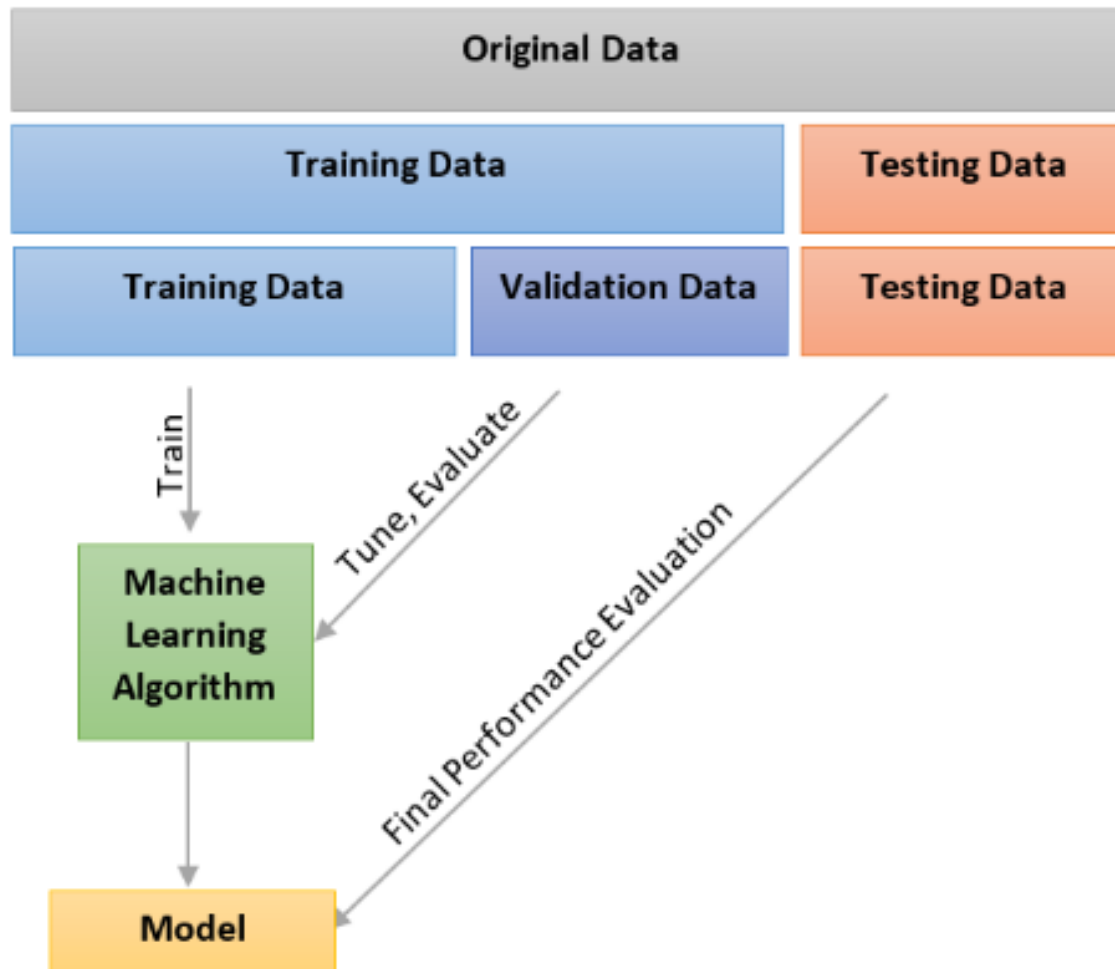


Overfitting

Underfitting



## Partições

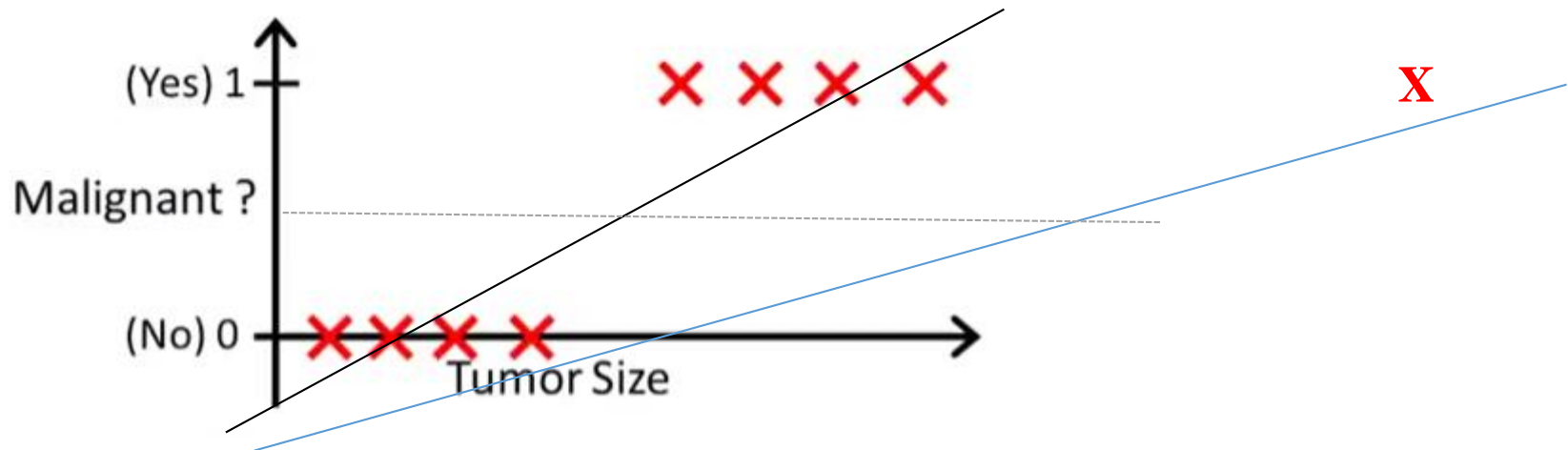


## Regressão Linear

- Parabéns! Você acaba de ser contratado como o mais novo Data Scientist de uma empresa global de Real Estate. Com o crescimento acelerado da cidade de Boston, devido sua proximidade a centros de excelência como Harvard e MIT, o mercado imobiliário da região apresenta uma oportunidade única. Para comprar os melhores imóveis aos melhores preços, você deve desenvolver um modelo capaz de receber os dados de um imóvel qualquer e dizer qual deve ser seu preço aproximado. Assim ao buscar por oportunidades na região poderá filtrar o que está caro demais e o que está barato.
- Para desenvolver seu modelo você irá realizar as seguintes tarefas:
  - 1) Importar o dataset em [https://lirielly.github.io/LinearRegression\\_index.html](https://lirielly.github.io/LinearRegression_index.html)
  - 2) Importar o dataset para o ambiente R.
  - 3) Fazer uma rápida exploração dos dados.
  - 4) Dividir a base em 70% para treino e 30% para teste do modelo.
  - 5) Treinar o modelo de regressão linear na base de treino.
  - 6) Validar a performance do modelo e determinar os principais preditores.
  - 7) Fazer o “scoring” do modelo para os dados na base de teste.
  - 8) Usar o modelo para determinar o preço de um imóvel com as informações no exercício 3

Feature	Descrição	Tipo
CRIM	Taxa de Crimes	Real
NROOMS	Número de Quartos	Real
AGE	Idade do Imóvel	Real
DISC	Distância do Centro	Real
BATH	Número de Banheiros	Real
TAX	Taxa de IPTU	Real
MEDV	Preço do Imóvel	Real

## Classificação: por que não regressão ?



**Um único valor pode distorcer completamente o resultado**

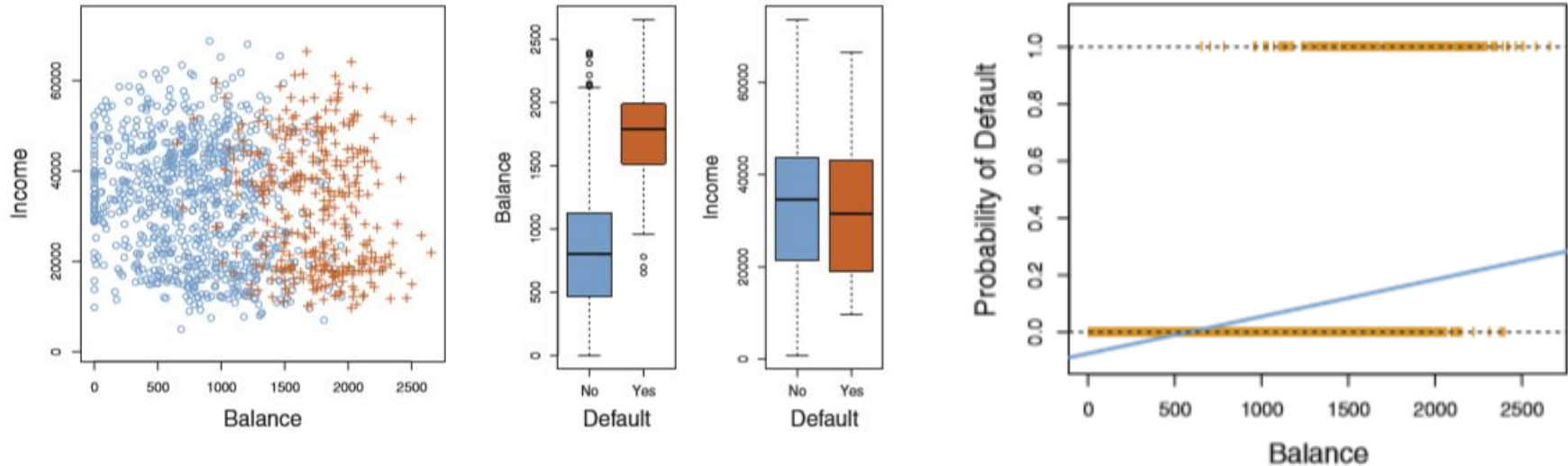
## Classificação: por que não regressão ?

$$Y = \begin{cases} 1 & \text{if stroke;} \\ 2 & \text{if drug overdose;} \\ 3 & \text{if epileptic seizure.} \end{cases}$$

$$Y = \begin{cases} 0 & \text{if stroke;} \\ 1 & \text{if drug overdose.} \end{cases}$$

**Ordem e valores relativos não correspondem a realidade**

## “Regressão” Logística

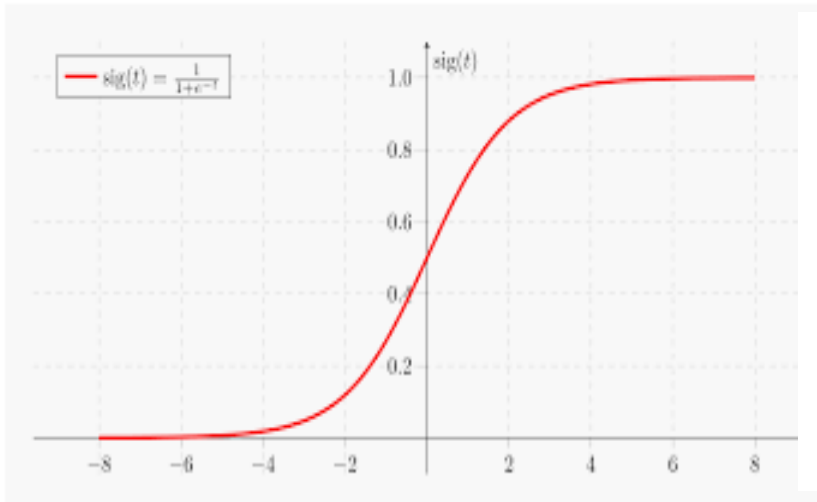


**Resultado da regressão pode exceder o intervalo (0 a 1)**

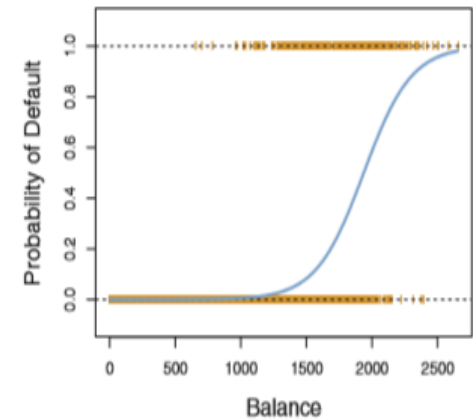
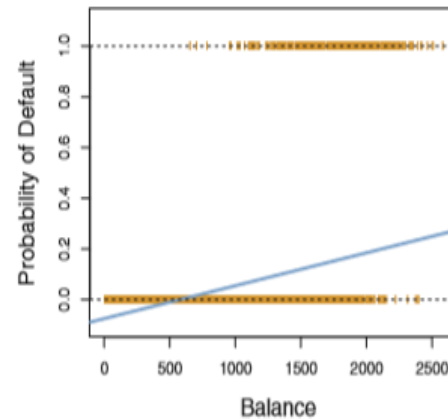


## “Regressão” Logística

$$p(x) = \frac{e^{\beta_0 + \beta_1 x_1 + \dots + \beta_p x_p}}{1 + e^{\beta_0 + \beta_1 x_1 + \dots + \beta_p x_p}}$$



Sigmoid Function ou  
Logistics Function

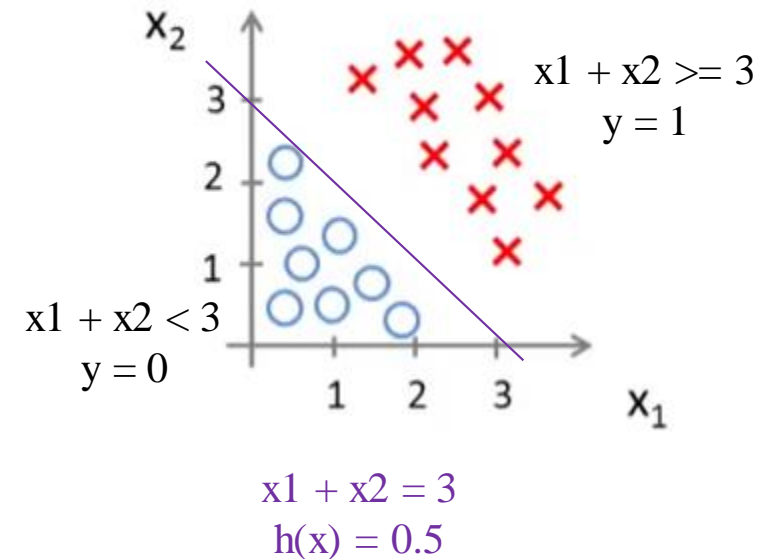


Logistic Regression:  $0 \leq h_{\theta}(x) \leq 1$

$$\begin{aligned} h_{\theta}(x) &\geq 0.5 \rightarrow y = 1 \\ h_{\theta}(x) &< 0.5 \rightarrow y = 0 \end{aligned}$$

$$\begin{aligned} g(z) &\geq 0.5 \\ \text{when } z &\geq 0 \end{aligned}$$

## “Regressão” Logística



$$h_{\theta}(x) \geq 0.5 \rightarrow y = 1$$

$$h_{\theta}(x) < 0.5 \rightarrow y = 0$$

$$g(z) \geq 0.5$$

$$\text{when } z \geq 0$$

$$h_{\theta}(x) = g(\theta_0 + \theta_1 x_1 + \theta_2 x_2)$$

$$\theta_0 = -3$$

$$\theta_1 = 1$$

$$\theta_2 = 1$$

Predict “ $y = 1$ ” if  $-3 + x_1 + x_2 \geq 0$

$$x_1 + x_2 \geq 3$$

## “Regressão” Logística

- Simples x Múltipla: uma mesma variável pode ter um efeito sozinha e outra quando combinada (sessão 4.3.3 página 148)

	Coefficient
<b>Intercept</b>	−3.5041
<b>student [Yes]</b>	0.4049

	Coefficient
<b>Intercept</b>	−10.8690
<b>balance</b>	0.0057
<b>income</b>	0.0030
<b>student [Yes]</b>	−0.6468

This is an important distinction for a credit card company that is trying to determine to whom they should offer credit. A student is riskier than a non-student if no information about the student's credit card balance is available. However, that student is less risky than a non-student with the same credit card balance!

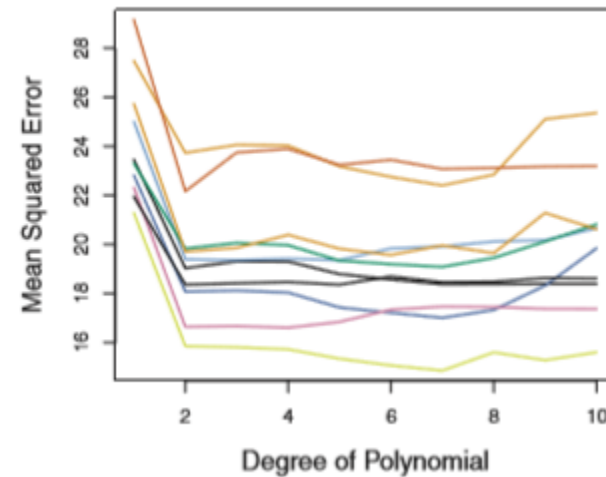
## Regressão Logística

- Mais um dia no MLBB (Machine Learning Bank of Boston) e devido ao aumento na demanda por imóveis na região o mercado de crédito está em alta e é preciso alocar seus empréstimos da melhor maneira possível. O problema é que os pedidos são tantos que estão sobrecarregando os analistas de crédito do banco. Para resolver essa situação você decide desenvolver um modelo para automatizar o processo de aprovação de crédito. Sua mais nova oportunidade de apresentar o modelo é no pedido de crédito feito por um grupo brasileiro de Real Estate que está se expandindo na região.
- Para desenvolver seu modelo você irá realizar as seguintes tarefas:
  - 1) Importar o dataset em [https://lirielly.github.io/LogisticRegression\\_index.html](https://lirielly.github.io/LogisticRegression_index.html)
  - 2) Importar o dataset para o ambiente R.
  - 3) Fazer uma rápida exploração dos dados.
  - 4) Dividir a base em 70% para treino e 30% para teste do modelo.
  - 5) Treinar o modelo de regressão logística na base de treino.
  - 6) Validar a performance do modelo e determinar os principais preditores.
  - 7) Fazer o “scoring” do modelo para os dados na base de teste.
  - 8) Determinar a aprovação ou não de crédito para o cliente no exercício 3

Feature	Descrição	Tipo
GENDER	Gênero	Flag
AGE	Idade	Real
DEBT	Dívidas	Real
MARRIED	Estado Civil	Flag
BANK_CUSTOMER	Cliente do Banco	Flag
EDUCATION_LEVEL	Nível de Educação (Médio, Superior, etc..)	Categorical
ETHNICITY	Etnia	Categorical
YEARS_EMPLOYED	Anos de Trabalho	Real
PRIOR_DEFAULT	Histórico de Calote/Atraso	Flag
EMPLOYED	Situação Empregatícia	Flag
CITIZEN	Cidadão USA	Flag
ZIPCODE	Localidade	Categorical
INCOME	Renda	Real
APPROVED	Crédito Aprovado ou Não	Target

## Resampling

- Aumenta confiabilidade do modelo
- Exige maior poder computacional
- Cross validation



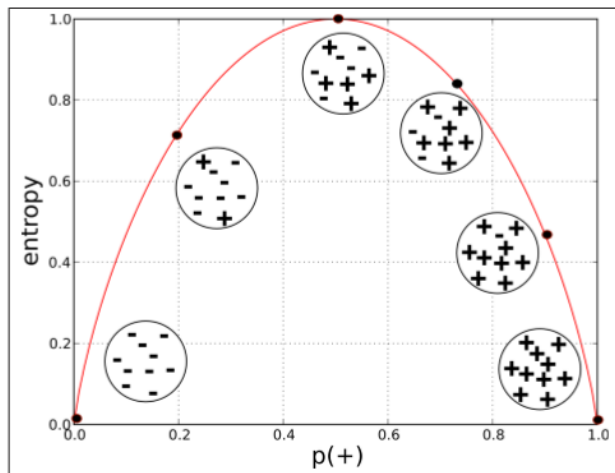
Amostras diferentes geram resultados bem diferentes !!

## Cross Validation



## Árvore de Decisão

- Fácil interpretação, fácil visualização
- Não precisa de variável *dummy* para preditores qualitativos
- Critério de divisão: Ganho de informação - IG (Shannon, 1948)
- Baseado em Entropia como uma medida de desordem
- Desordem: quanto mais mesclado, maior a entropia

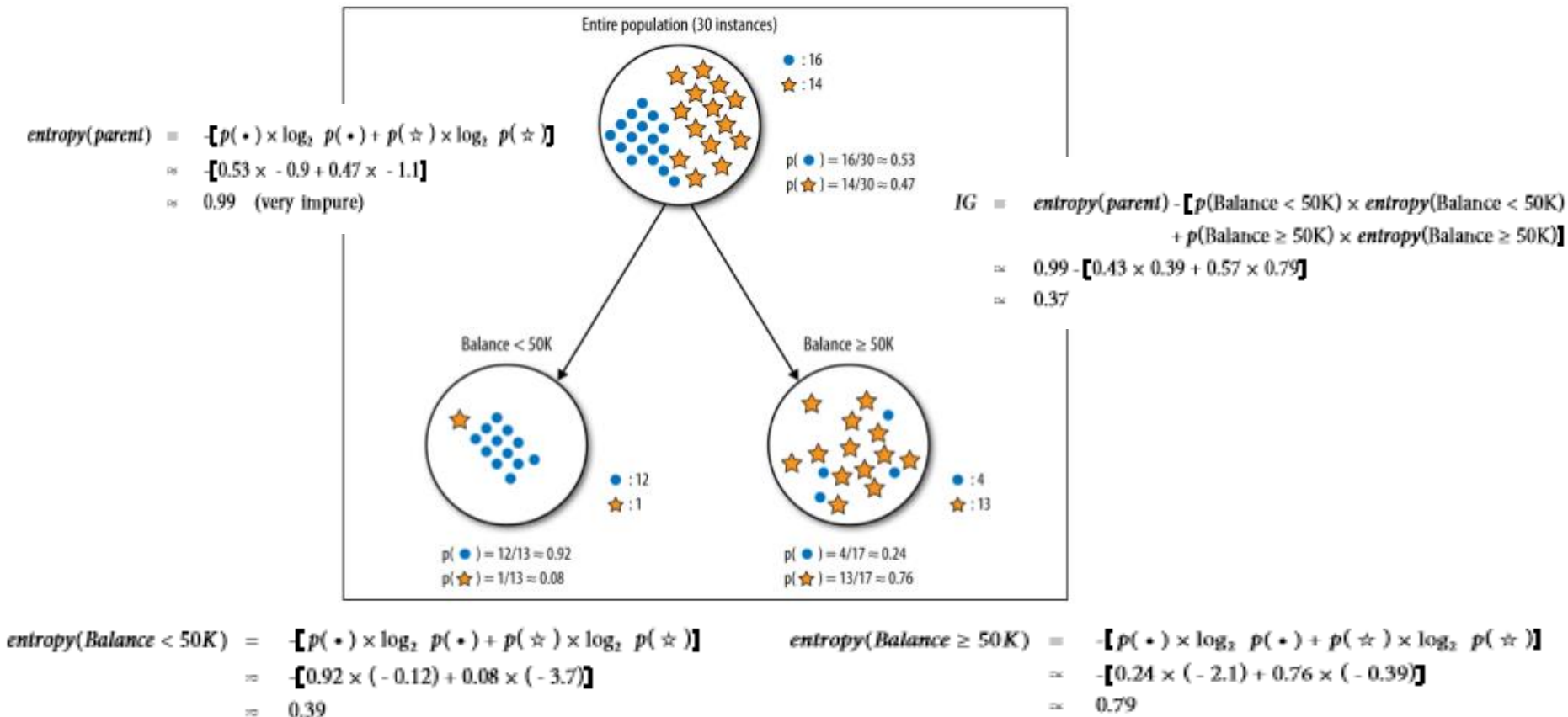


$$entropy = - p_1 \log (p_1) - p_2 \log (p_2) - \dots$$

$$IG(parent, children) = entropy(parent) - [p(c_1) \times entropy(c_1) + p(c_2) \times entropy(c_2) + \dots]$$

## Árvore de Decisão

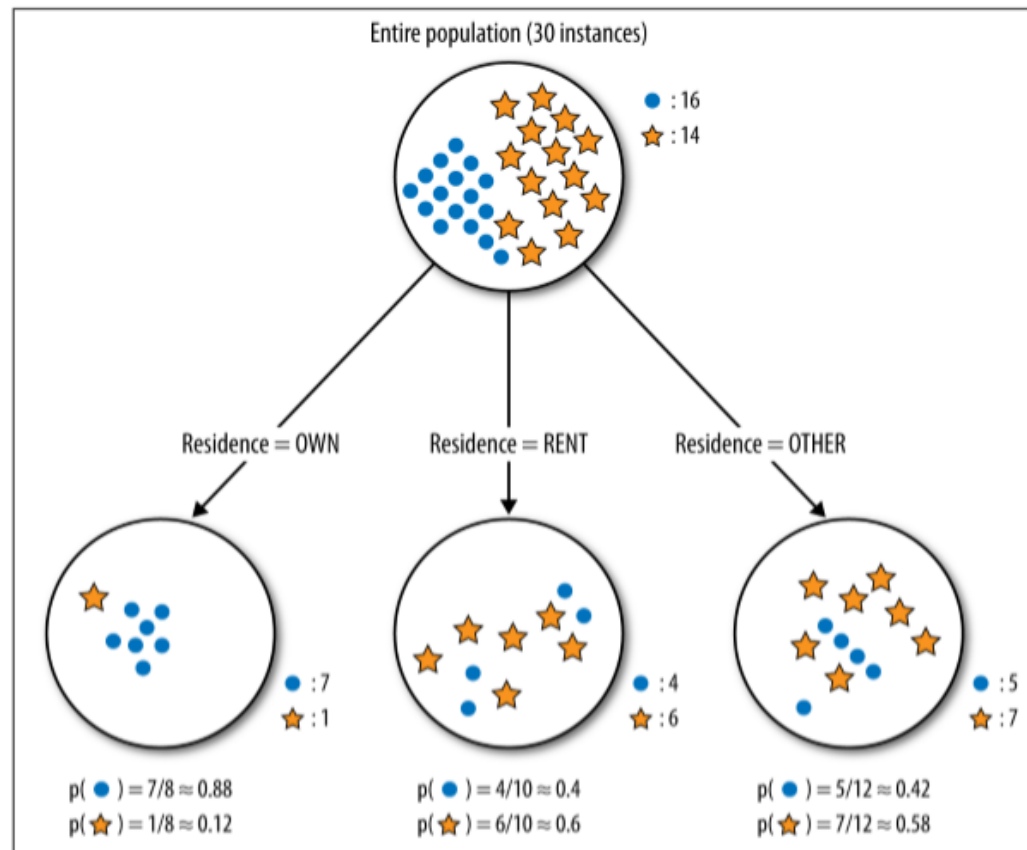
- Algoritmo: Se a divisão reduzir a entropia (ou aumenta o ganho de informação) então siga com a divisão





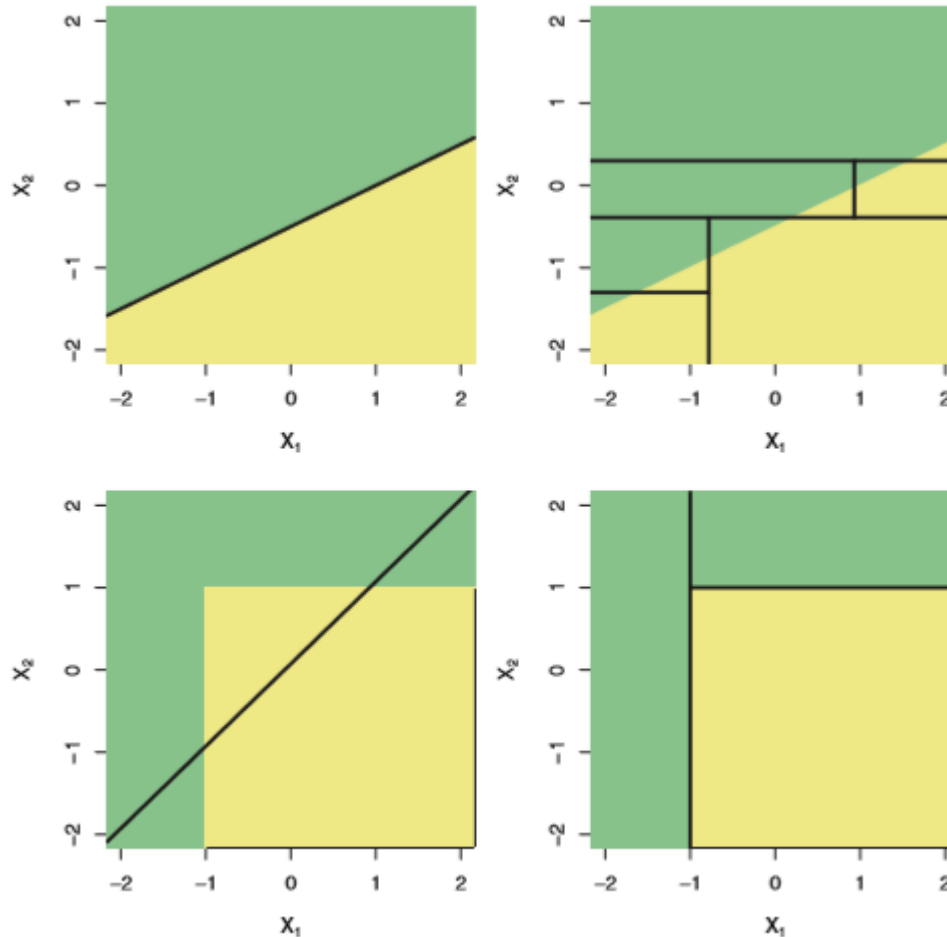
## Árvore de Decisão

- Algoritmo: Se a divisão reduzir a entropia (ou aumenta o ganho de informação) então siga com a divisão



$$\begin{aligned} \text{entropy}(\text{parent}) &\approx 0.99 \\ \text{entropy}(\text{Residence=OWN}) &\approx 0.54 \\ \text{entropy}(\text{Residence=RENT}) &\approx 0.97 \\ \text{entropy}(\text{Residence=OTHER}) &\approx 0.98 \\ IG &\approx 0.13 \end{aligned}$$

## Árvore de Decisão x Modelo Linear



## Árvore de Decisão

- Não costumam ter alta acurácia
- Muito voláteis -> uma mudança simples no dado pode mudar completamente a árvore

**Solução:** Combinação

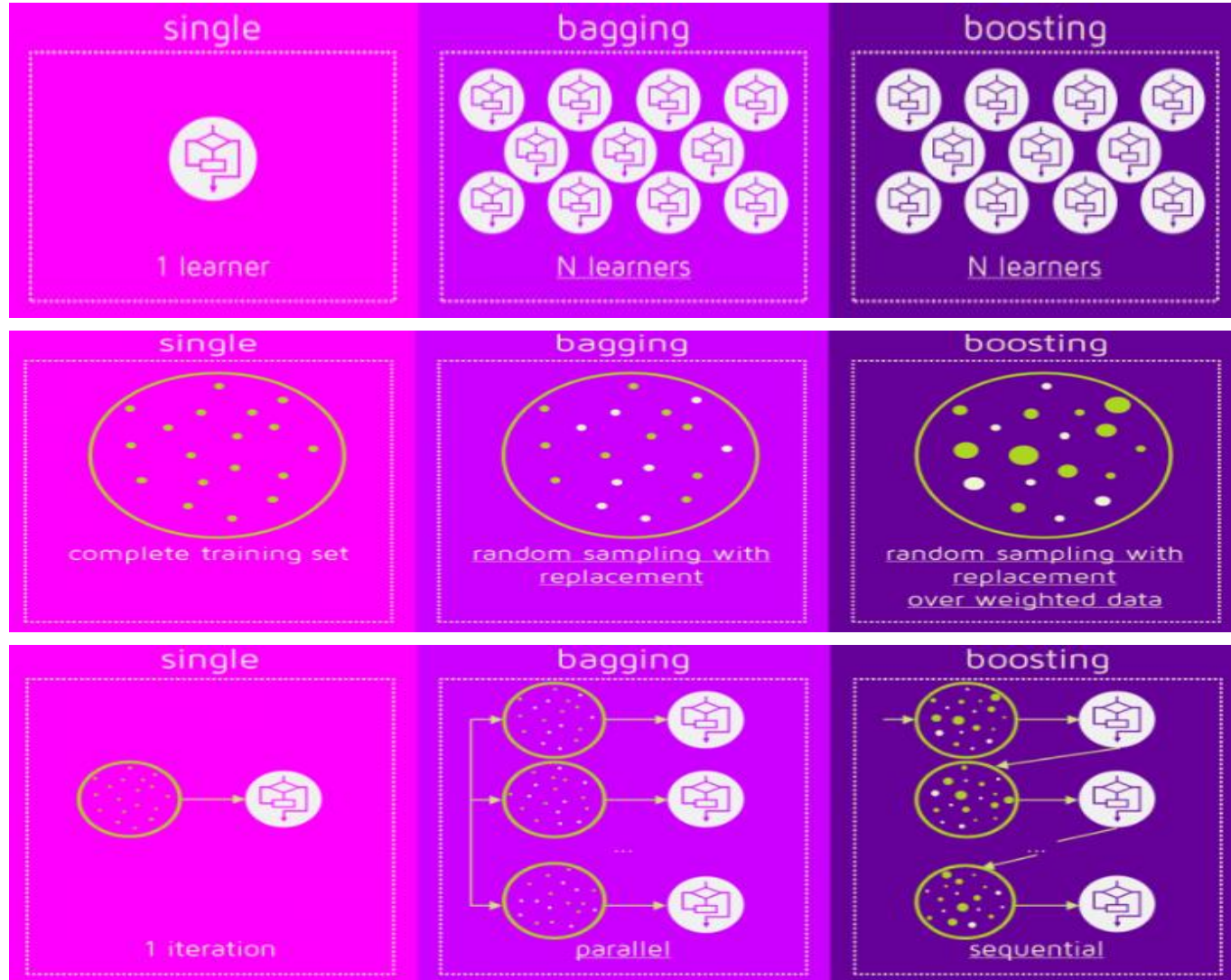
“Set of weak learners are combined to create a strong learner”

**Boosting** -> cria 1 árvore por vez, vai melhorando a próxima utilizando os erros da árvore anterior

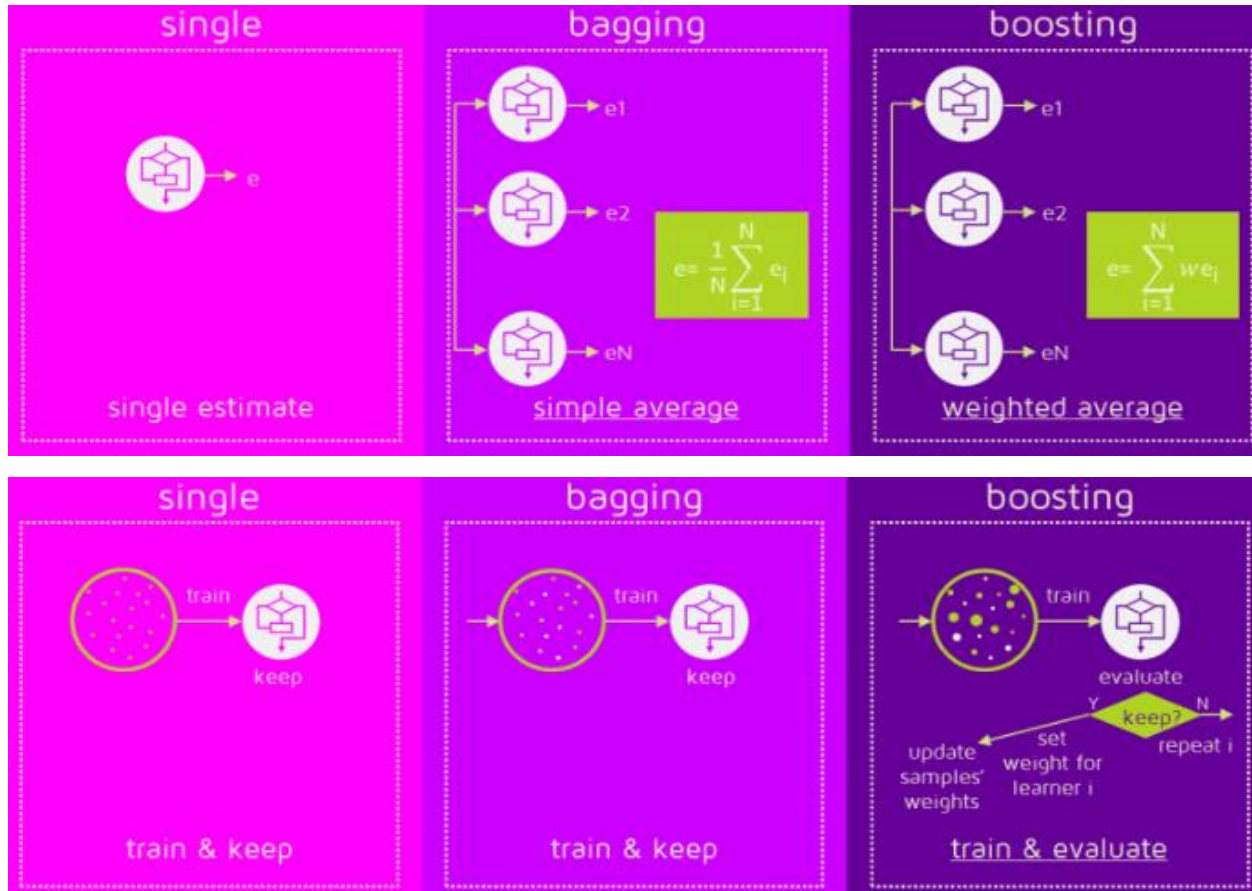
**Bagging (Bootstrap AGGregatING)** -> utiliza n amostras aleatórias de treinamento. Considera a média das árvores

**Random Forest** -> igual Bagging, mas varia a quantidade de features

## Combinação



## Combinação



Variações: AdaBoost, LPBoost, XGBoost, GradientBoost, BrownBoost.

## Árvore de Decisão

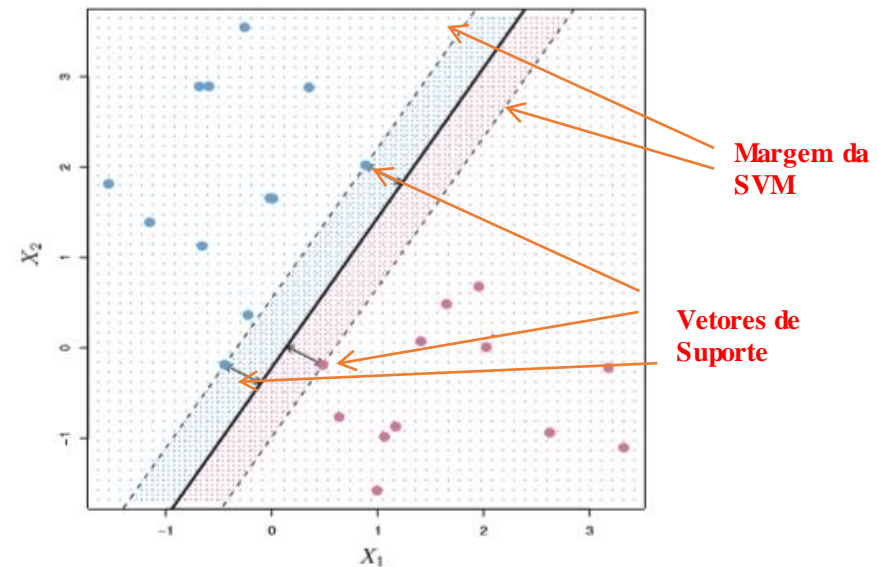
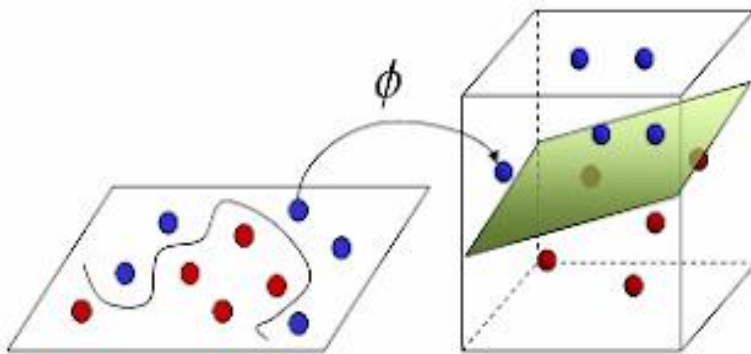
- Com o crescimento urbano acelerado da cidade de Boston e o aquecimento da economia local os serviços públicos estão sobrecarregados. O hospital Boston D'Or decidiu investir em automatizar o processo de direcionamento de pacientes para os especialistas corretos, para isso será desenvolvido um modelo de árvore de decisão baseado em um formulário padrão preenchido pelos paciente.
- Para desenvolver seu modelo você irá realizar as seguintes tarefas:
  - 1) Importar o dataset BostonHospital disponível em [https://lirielly.github.io/DecisionTree\\_index.html](https://lirielly.github.io/DecisionTree_index.html)
  - 2) Importar o dataset para o ambiente R.
  - 3) Fazer uma rápida exploração dos dados.
  - 4) Dividir a base em 70% para treino e 30% para teste do modelo.
  - 5) Treinar 3 árvores de decisão (boosting, bagging, random forest) na base de treino.
  - 6) Validar a performance dos modelos e determinar os principais preditores.
  - 7) Fazer o “scoring” do modelo para os dados na base de teste.
  - 8) Determinar o tipode de patologia do indivíduo do exercício 3



## SVM – Máquina de Vetores de Suporte

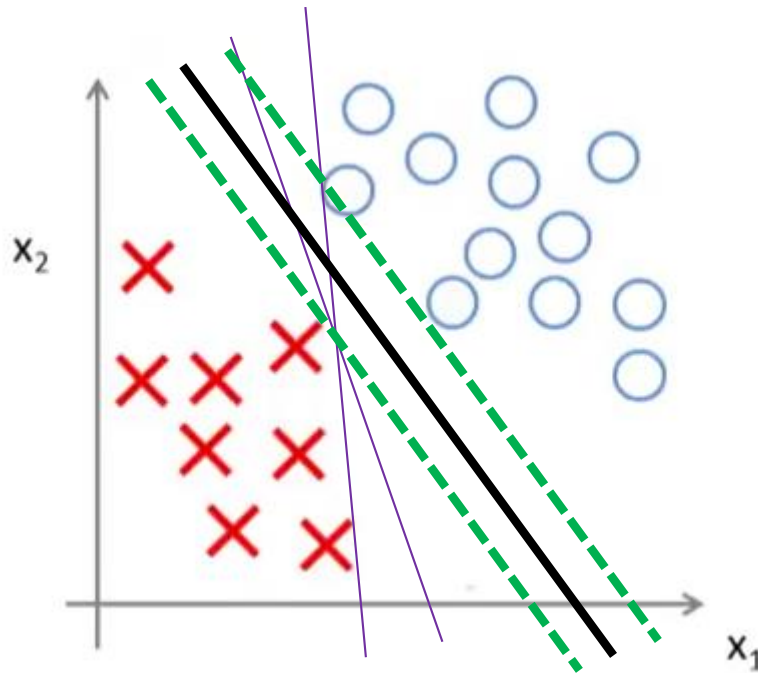
- Hiperplano

*figura geométrica de curvatura nula em um espaço euclidiano  $n$ -dimensional e cuja equação em coordenadas cartesianas é linear.*



O hiperplano depende diretamente dos vetores de suporte, mas não das outras observações: um movimento para qualquer uma das outras observações não afetaria o hiperplano de separação, desde que o movimento da observação não o faça cruzar o limite definido pela margem.

## SVM – Máquina de Vetores de Suporte

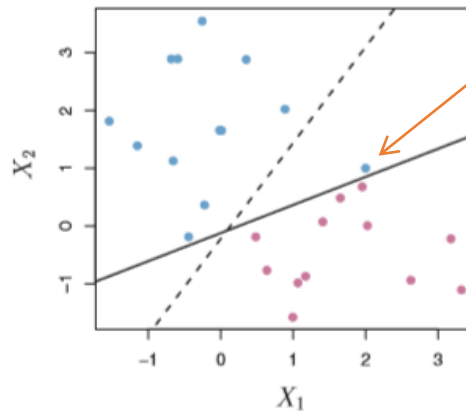
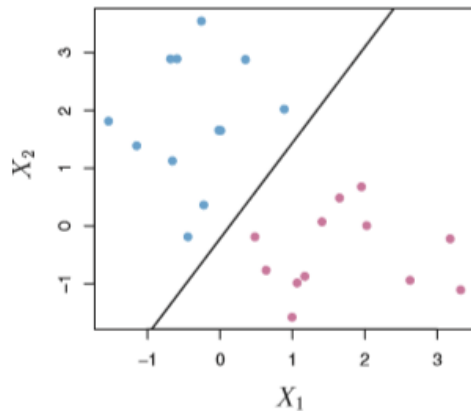


Modelo mais robusto

Tenta separar os conjuntos o máximo possível através da margem

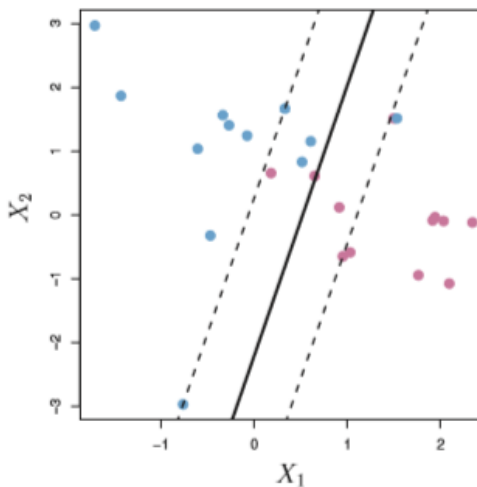
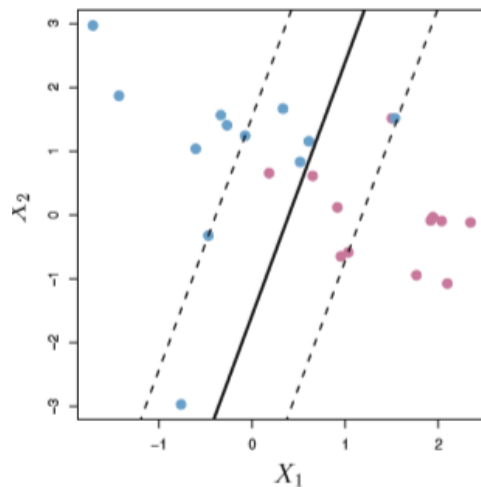


## SVM – Máquina de Vetores de Suporte



Nova observação

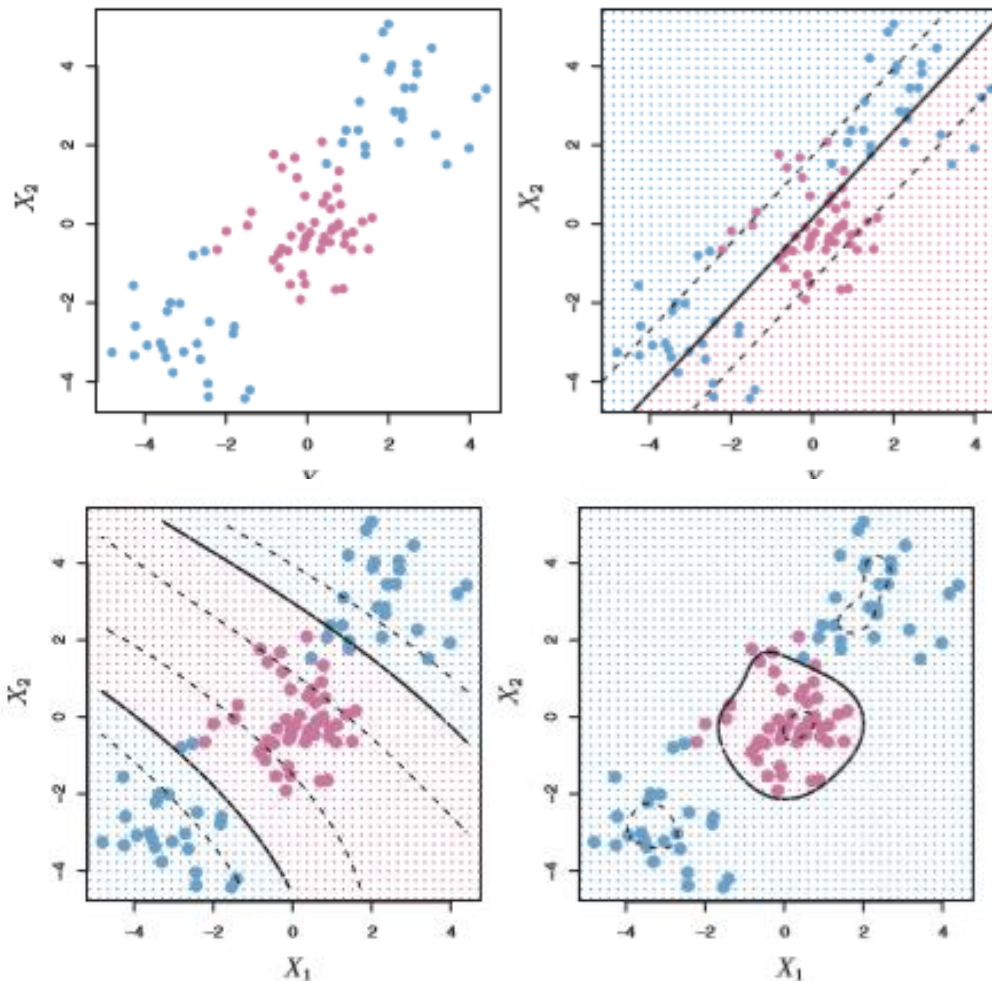
Uma única adição da amostra altera significativamente a reta: overfitting



Modelo mais robusto

Tunings diferentes de SVMs  
Balanceamento de Bias x  
Variance

## SVM – Máquina de Vetores de Suporte



SVM utilizando  
Kernel Polinomial

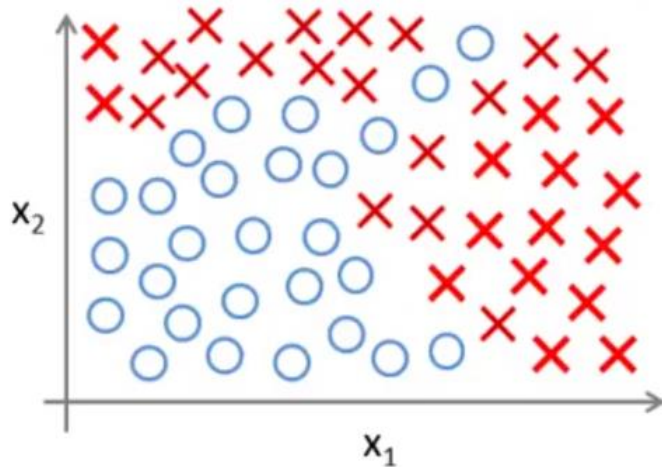
SVM utilizando  
Kernel Radial

## SVM

- O fundo de investimento Bettinas está investindo em uma nova solução baseada em machine learning para automatizar e aperfeiçoar a classificação do “credit rating” de companhias públicas baseado em um conjunto de métricas fundamentalistas. Assim o fundo pretende assumir o mínimo risco necessário para bater os fundos concorrentes e seu benchmark o CDI.
- Para desenvolver seu modelo você irá realizar as seguintes tarefas:
  - 1) Importar o dataset em [https://lirielly.github.io/SVM\\_index.html](https://lirielly.github.io/SVM_index.html)
  - 2) Importar o dataset para o ambiente R.
  - 3) Fazer uma rápida exploração dos dados.
  - 4) Dividir a base em 70% para treino e 30% para teste do modelo.
  - 5) Treinar o modelo de regressão linear na base de treino.
  - 6) Validar a performance do modelo e determinar os principais preditores.
  - 7) Fazer o “scoring” do modelo para os dados na base de teste.
  - 8) Usar o modelo para determinar o credit rating com as informações no exercício 3

## Redes Neurais – Por que usar ?

### Non-linear Classification



Regressão Logística multi polinomial

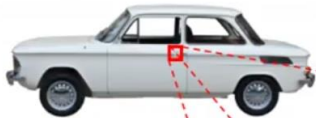
$$g(\theta_0 + \theta_1 x_1 + \theta_2 x_2 + \theta_3 x_1 x_2 + \theta_4 x_1^2 x_2 + \theta_5 x_1^3 x_2 + \theta_6 x_1 x_2^2 + \dots)$$

E se tivermos mais do que somente X1 e X2 ? (mundo real)

Para 100 deles, chegaríamos a mais de 5000 termos !!!

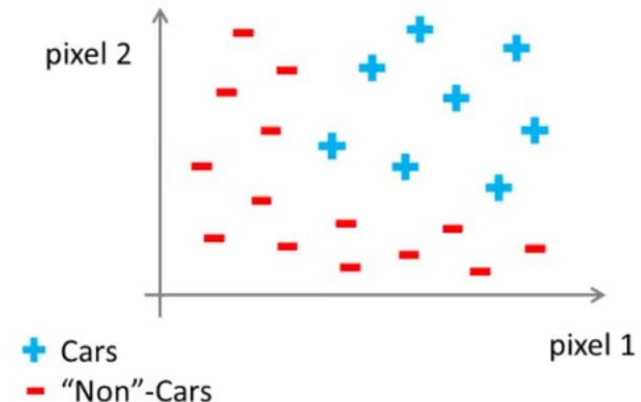
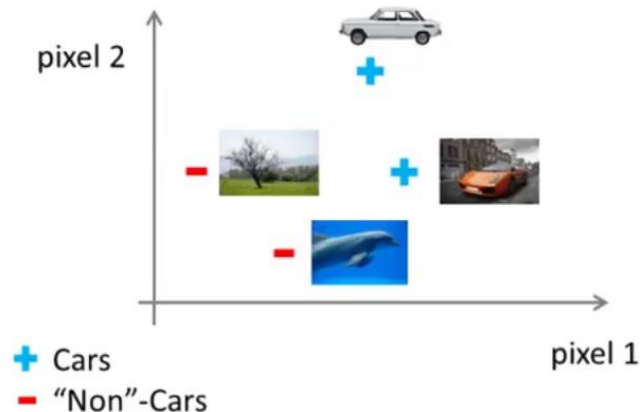
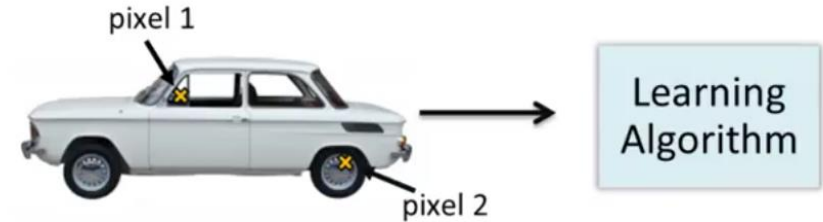
## Redes Neurais – Por que usar ?

You see this:

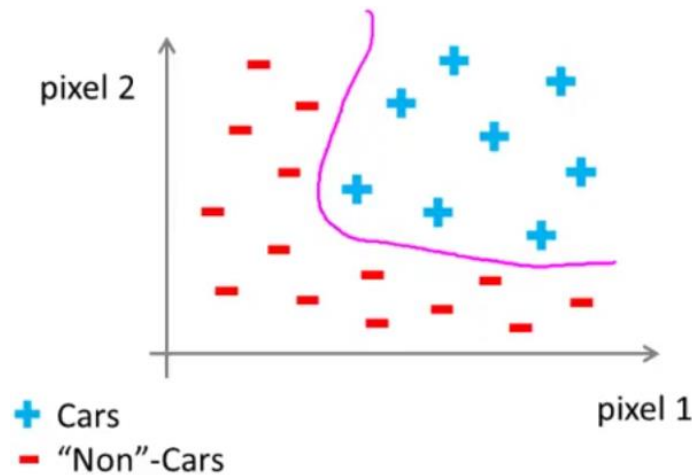


But the camera sees this:

194	210	201	212	199	213	215	195	178	158	182	209
180	189	190	221	209	205	191	167	147	115	129	163
114	126	140	188	176	165	152	140	170	106	78	88
87	103	115	154	143	142	149	153	173	101	57	57
102	112	106	131	122	138	152	147	128	84	58	66
94	95	79	104	105	124	129	113	107	87	69	67
68	71	69	98	89	92	98	95	89	88	76	67
41	56	68	99	63	45	60	82	58	76	75	65
20	43	69	75	56	41	51	73	55	70	63	44
50	50	57	69	75	75	73	74	53	68	59	37
72	59	53	66	84	92	84	74	57	72	63	42
67	61	58	65	75	78	76	73	59	75	69	50



## Redes Neurais – Por que usar ?

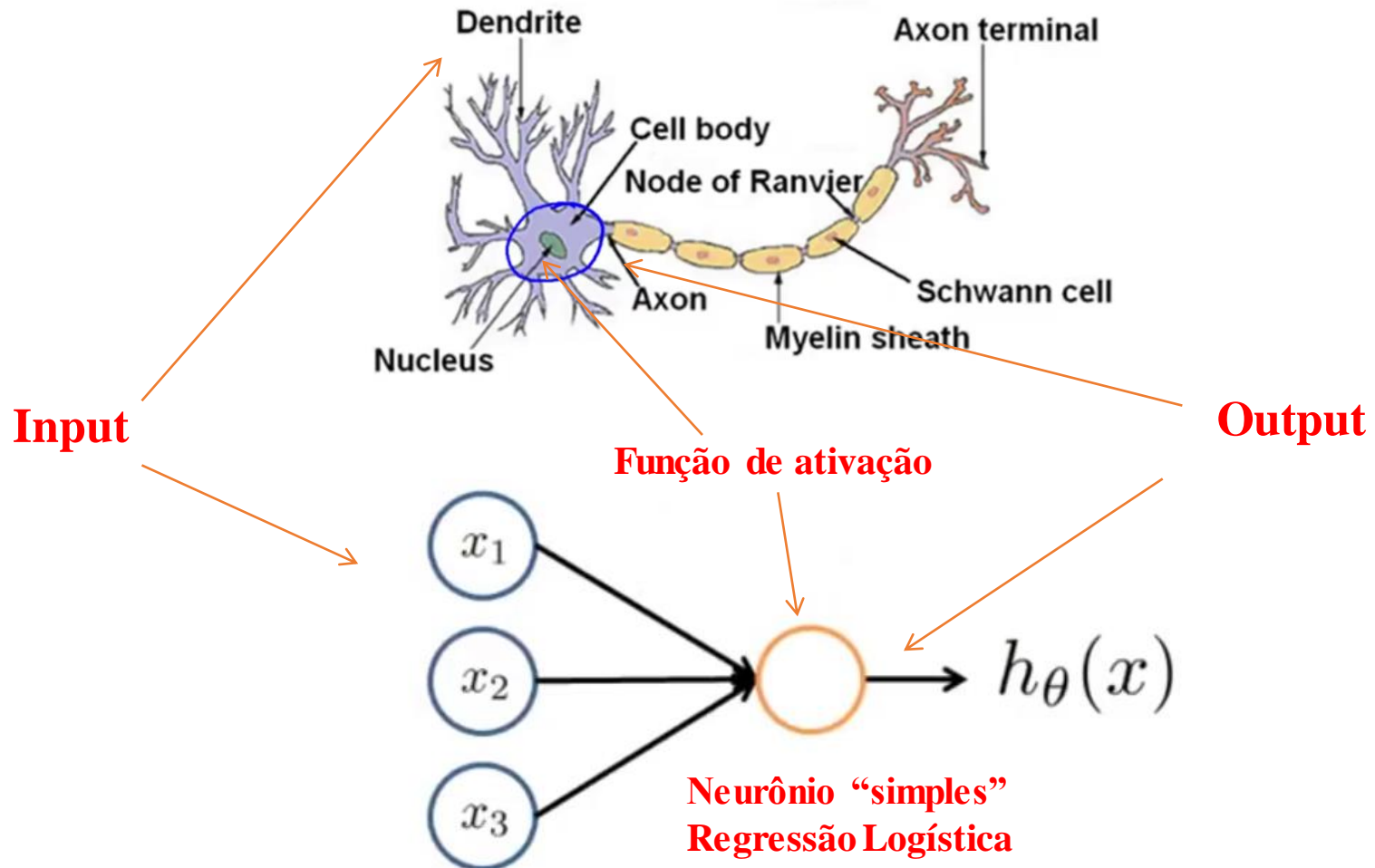


50 x 50 pixel images  $\rightarrow$  2500 pixels  
 $n = 2500$  (7500 if RGB)

$$x = \begin{bmatrix} \text{pixel 1 intensity} \\ \text{pixel 2 intensity} \\ \vdots \\ \text{pixel 2500 intensity} \end{bmatrix}$$

Quadratic features ( $x_i \times x_j$ ):  $\approx 3$  million  
features

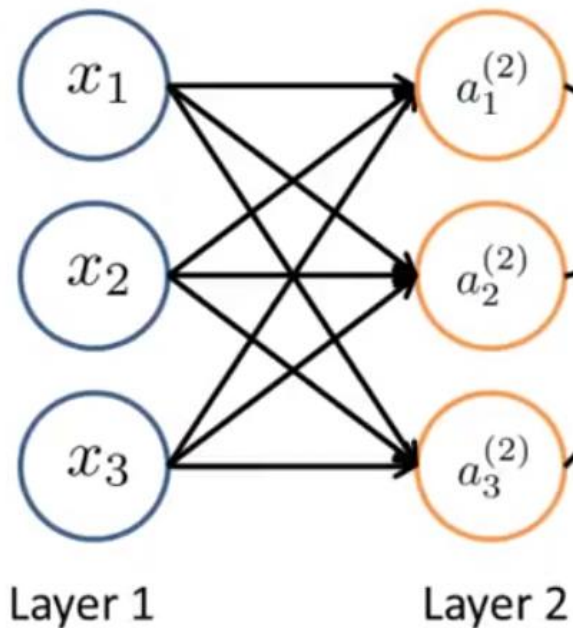
## Redes Neurais





## Redes Neurais

### Rede de Neurônios

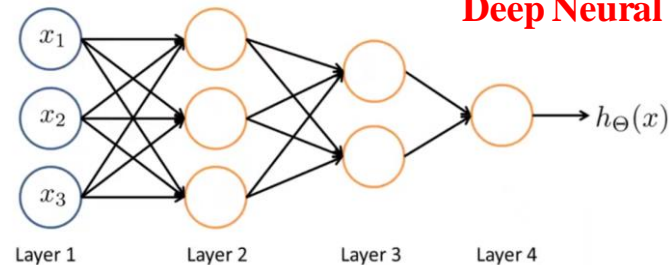


**Input  
Layer**

**Hidden  
Layer**

**Output  
Layer**

### Deep Neural Networks



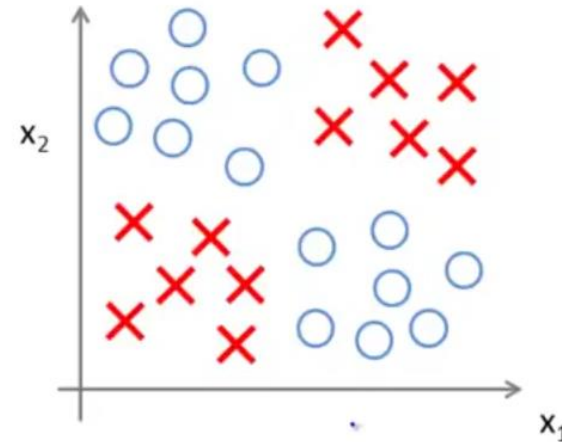
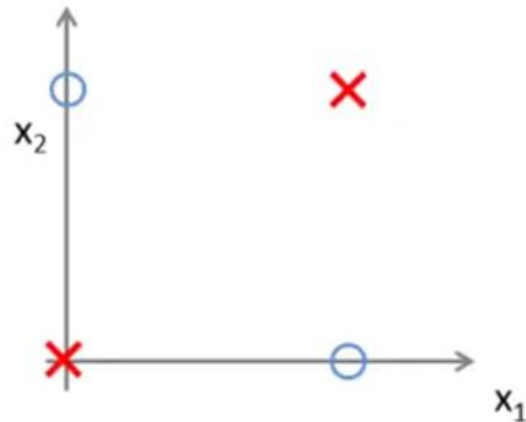
$$h_{\Theta}(x)$$
$$a_1^{(2)} = g(\Theta_{10}^{(1)} x_0 + \Theta_{11}^{(1)} x_1 + \Theta_{12}^{(1)} x_2 + \Theta_{13}^{(1)} x_3)$$
$$a_2^{(2)} = g(\Theta_{20}^{(1)} x_0 + \Theta_{21}^{(1)} x_1 + \Theta_{22}^{(1)} x_2 + \Theta_{23}^{(1)} x_3)$$
$$a_3^{(2)} = g(\Theta_{30}^{(1)} x_0 + \Theta_{31}^{(1)} x_1 + \Theta_{32}^{(1)} x_2 + \Theta_{33}^{(1)} x_3)$$
$$h_{\Theta}(x) = a_1^{(3)} = g(\Theta_{10}^{(2)} a_0^{(2)} + \Theta_{11}^{(2)} a_1^{(2)} + \Theta_{12}^{(2)} a_2^{(2)} + \Theta_{13}^{(2)} a_3^{(2)})$$

### Forward Propagation

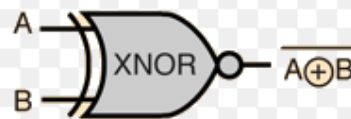


## Redes Neurais

$x_1, x_2$  are binary (0 or 1)



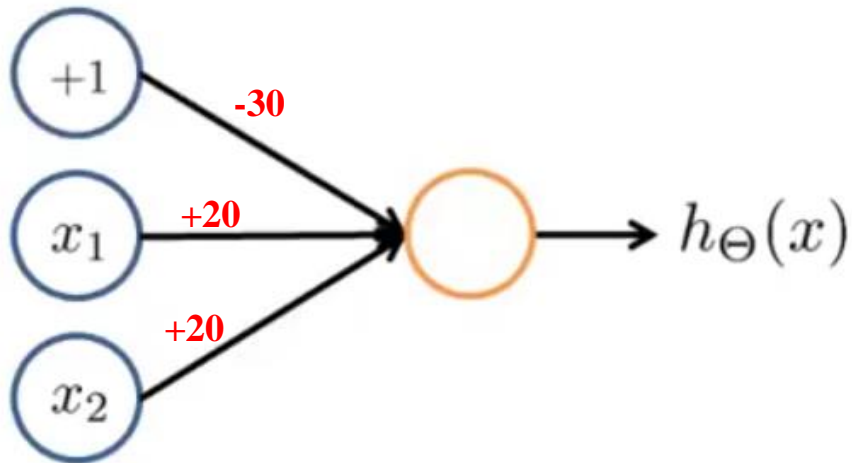
**$x_1 \text{ XNOR } x_2$**



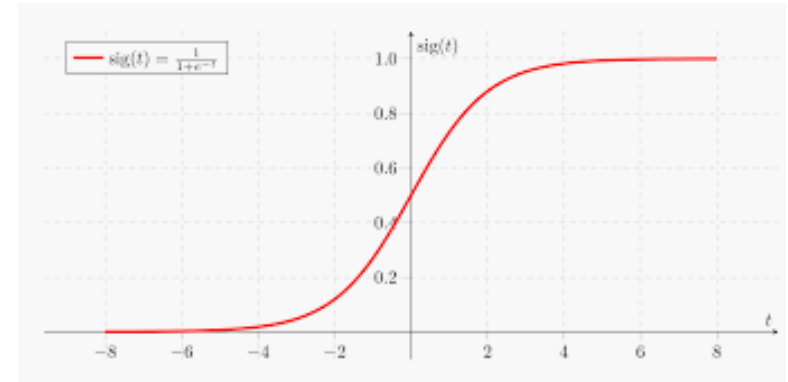
A	B	Out
0	0	1
0	1	0
1	0	0
1	1	1

## Redes Neurais

$$y = x_1 \text{ AND } x_2$$



$$h(x) = g(-30 + 20 \cdot x_1 + 20 \cdot x_2)$$

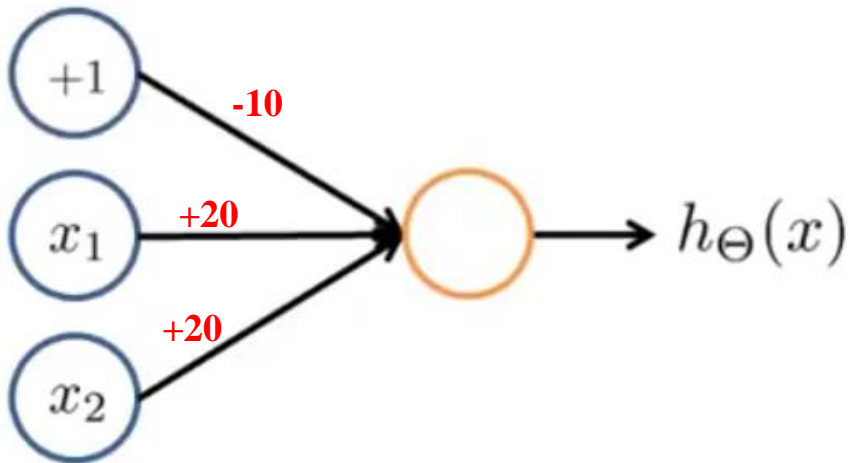


$x_1$	$x_2$	$h(x)$
0	0	$g(-30) \rightarrow 0$
0	1	$g(-10) \rightarrow 0$
1	0	$g(-10) \rightarrow 0$
1	1	$g(10) \rightarrow 1$

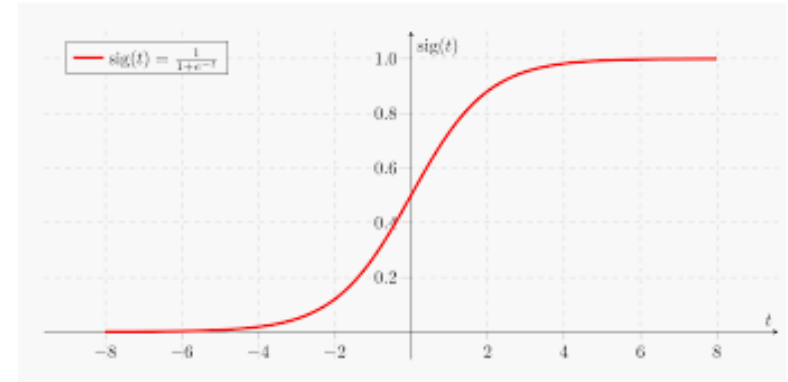
$$h(x) = x_1 \text{ AND } x_2$$

## Redes Neurais

$$y = x1 \text{ OR } x2$$



$$h(x) = g(-10 + 20 \cdot x_1 + 20 \cdot x_2)$$

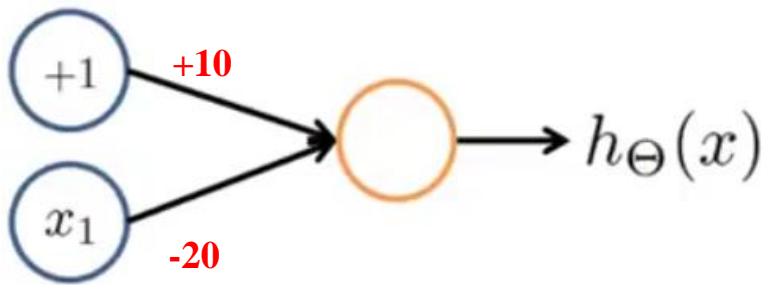


$x_1$	$x_2$	$h(x)$
0	0	$g(-10) \rightarrow 0$
0	1	$g(10) \rightarrow 1$
1	0	$g(10) \rightarrow 1$
1	1	$g(30) \rightarrow 1$

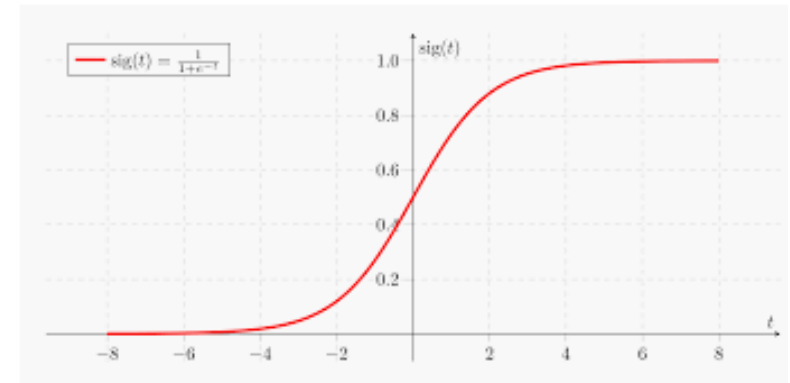
$$h(x) = x1 \text{ OR } x2$$

## Redes Neurais

NOT X1



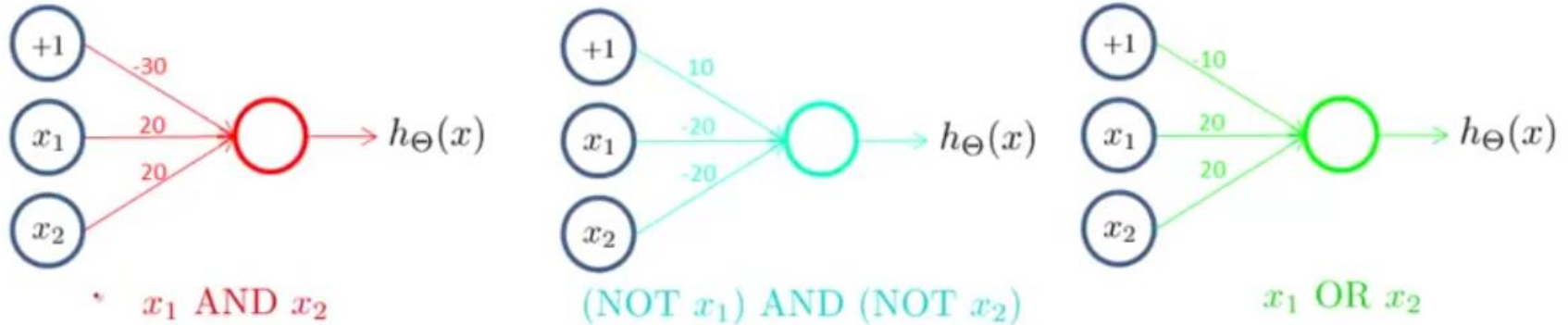
$$h(x) = g(10 + 20 \cdot x_1)$$



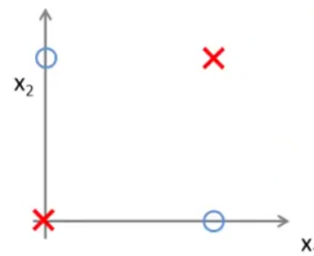
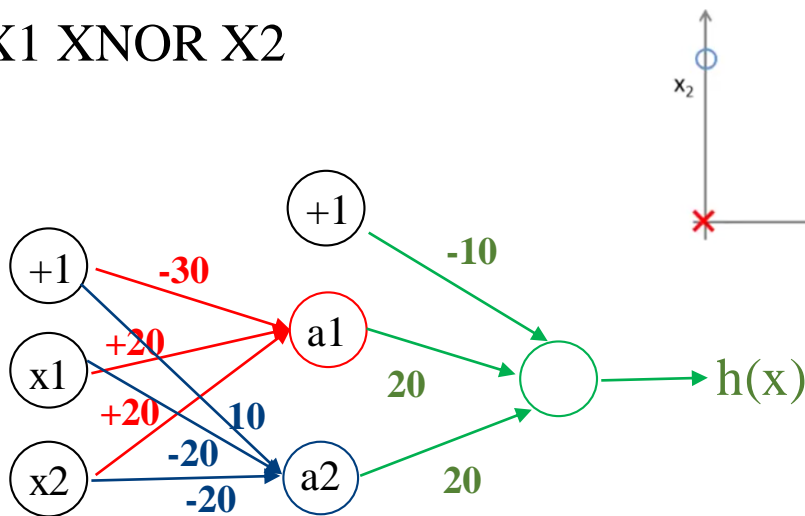
$x_1$	$h(x)$
0	$g(10) \rightarrow 1$
1	$g(-10) \rightarrow 0$

$$h(x) = \text{NOT } x_1$$

## Redes Neurais



X1 XNOR X2



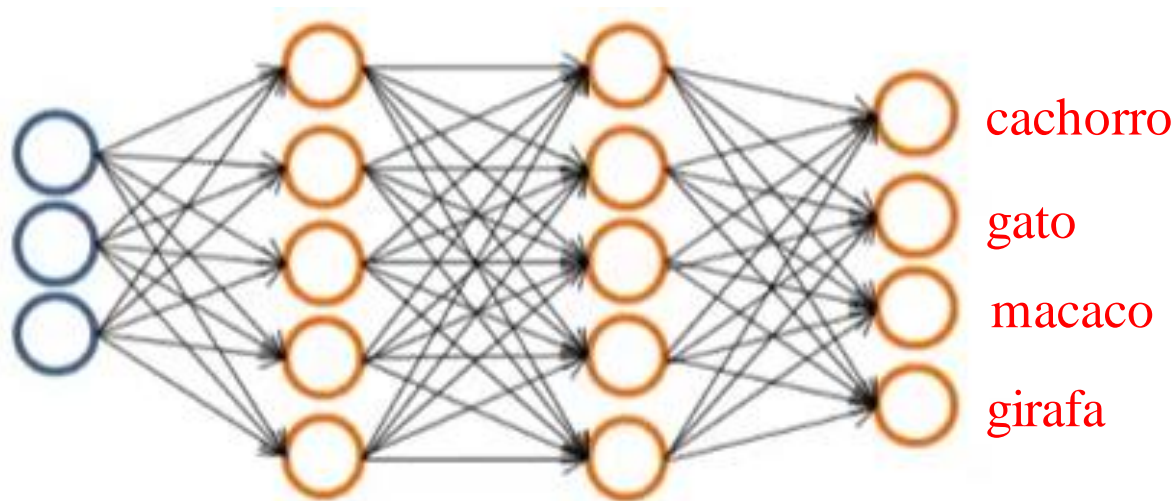
$x_1$	$x_2$	$a_1$	$a_2$	$h(x)$
0	0	0	1	1
0	1	0	0	0
1	0	0	0	0
1	1	1	0	1

$$h(x) = X1 \text{ XNOR } X2$$

## Redes Neurais

E se objetivo da predição for uma classe multi-variada ?

Ex: definir se a imagem é um cachorro, gato, macaco ou girafa ?



<https://playground.tensorflow.org/>

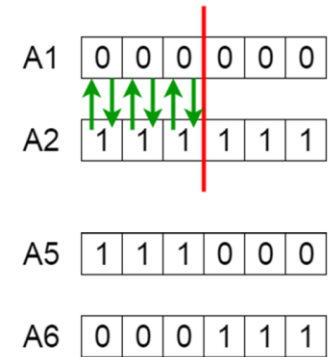
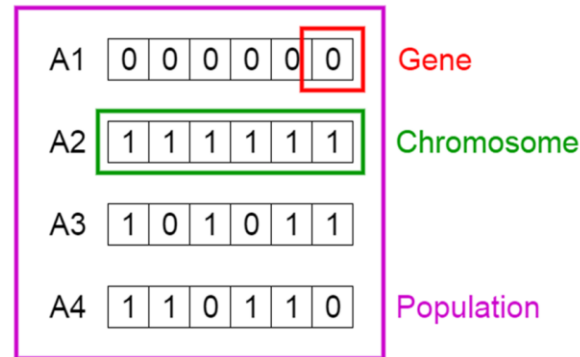
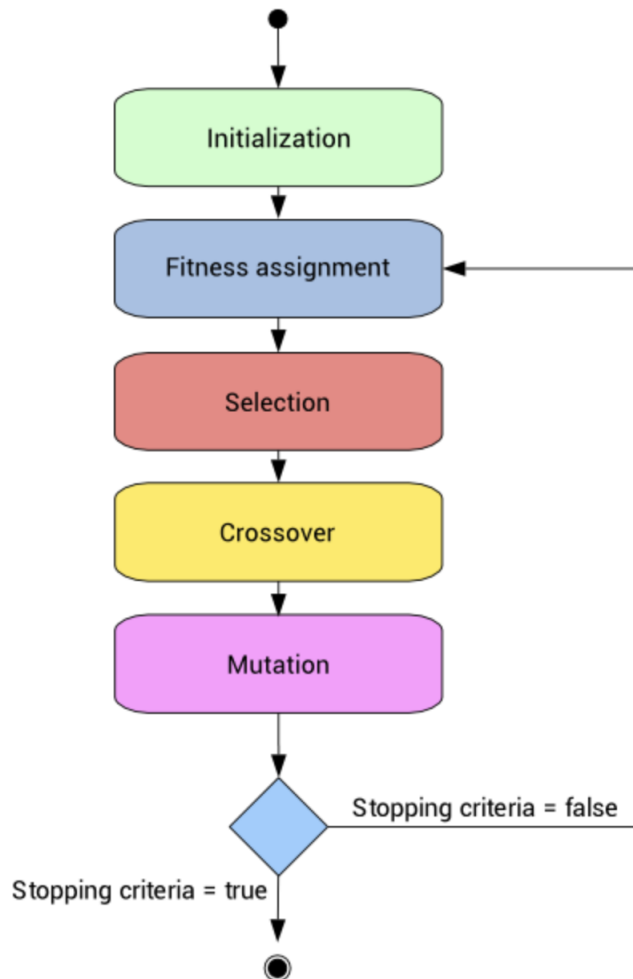
## Redes Neurais

- Com o recente sucesso do Machine Learning Bank of Boston no setor de crédito, o seu velho concorrente Goldman Data decide focar em captar novos clientes para se capitalizar. Para isso é preciso que o processo de abertura de conta seja o mais breve possível, assim surge a idéia de que o cadastramento de todos os dados seja feito baseado apenas no envio de uma foto do RG. Para isso você recebe a responsabilidade de desenvolver um OCR (Optical Character Recognition) para reconhecer dígitos e adicioná-los no formulário de cadastro do cliente.
- Para desenvolver seu modelo você irá realizar as seguintes tarefas:
  - 1) Importar o dataset BostonGoldman disponível em [https://lirielly.github.io/NeuralNetworks\\_index.html](https://lirielly.github.io/NeuralNetworks_index.html)
  - 2) Importar o dataset para o ambiente R.
  - 3) Fazer uma rápida exploração dos dados.
  - 4) Dividir a base em 70% para treino e 30% para teste do modelo.
  - 5) Treinar uma rede neural reconhecimento de dígitos.
  - 6) Validar a performance dos modelos.
  - 7) Fazer o “scoring” do modelo para os dados na base de teste.
  - 8) Determinar o dígito dado no exercício 3 e verificar se o modelo acertou.



0000000000000000  
1111111111111111  
2222222222222222  
3333333333333333  
4444444444444444  
5555555555555555  
6666666666666666  
7777777777777777  
8888888888888888  
9999999999999999

## Algoritmo Genético (GA)



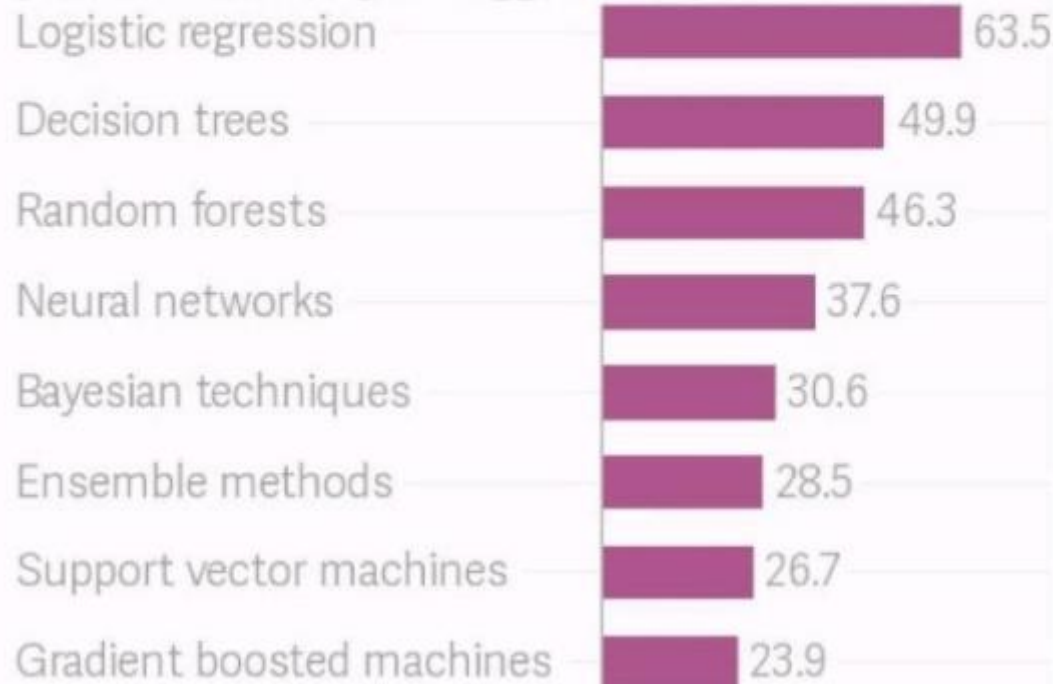


## Algoritmo Genético (GA)

- O grupo de e-commerce Amazonia está investido em soluções de data-driven marketing para customizar a experiência dos usuários em seu site. Para ser o mais eficiente possível o grupo de Data Scientist's da Amazonia decide desenvolver um modelo junto com a equipe de marketing para classificar a faixa de renda de um usuário qualquer. Assim na página inicial irão aparecer produtos de uma natureza mais premium ou genéricos.
- Para desenvolver seu modelo você irá realizar as seguintes tarefas:
  - 1) Importar o dataset AmazonGA disponível em [https://lirielly.github.io/GA\\_index.html](https://lirielly.github.io/GA_index.html)
  - 2) Importar o dataset para o ambiente R.
  - 3) Fazer uma rápida exploração dos dados.
  - 4) Dividir a base em 70% para treino e 30% para teste do modelo.
  - 5) Utilizar a técnica de algoritmo genético para reduzir o número de features do modelo
  - 6) Validar e comparar a performance dos modelos.
  - 7) Fazer o “scoring” do melhor para os dados nas bases de teste.

## Popular Predictive Methods Used By Data Scientists

(Based on a survey of Kaggle users)





## Case Final

- A multinacional de varejo Waldata está querendo expandir a sua presença na América Latina e por isso decide firmar uma parceria com a FGV para desenvolver um modelo preditivo do valor de vendas. Além disso a companhia decide investir em uma solução de “Targeted Advertising” que utilizará um modelo de “Machine Learning” para dizer se um cliente irá aderir ou não ao serviço premium.
- Para desenvolver seu modelo você irá realizar as seguintes tarefas:
  - 1) Importar os datasets RETAIL e MARKETING disponíveis em [https://lirielly.github.io/TCC\\_index.html](https://lirielly.github.io/TCC_index.html)
  - 2) Importar os datasets para o ambiente R.
  - 3) Fazer uma exploração detalhada dos dados. (Distribuições, valores faltantes etc ..)
  - 4) Dividir as bases em 70% para treino e 30% para teste do modelo. (Utilize sempre seed(314))
  - 5) Testar modelos de classificação para verificar se o cliente irá aderir ao serviço:
    - 1) Regressão Logística, Árvores de Decisão, SVM e Redes Neurais
  - 6) Testar modelos de regressão para o valor de vendas das lojas:
    - 1) Regressão Linear, Árvore de Decisão, e Redes Neurais
  - 7) Validar a performance dos modelos ( $R^2$  & Matriz de Confusão)
  - 8) Fazer o “scoring” dos modelos para os dados nas respectivas bases de teste