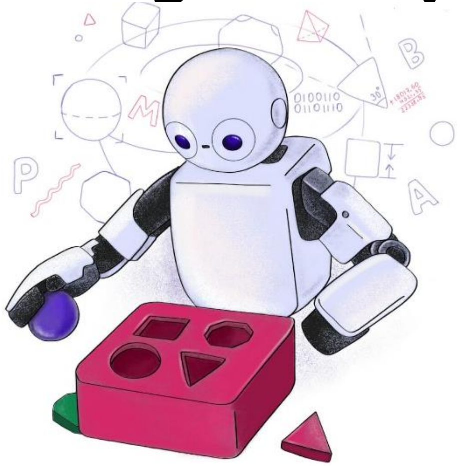
T320 - Introdução ao Aprendizado de Máquina II: *Classificação (Parte I)*



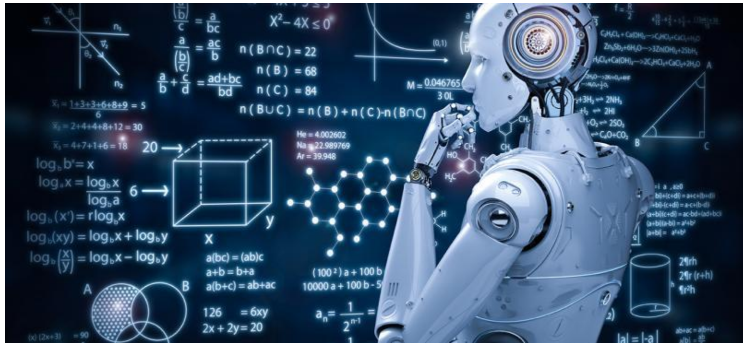
Inatel

Felipe Augusto Pereira de Figueiredo
felipe.figueiredo@inatel.br

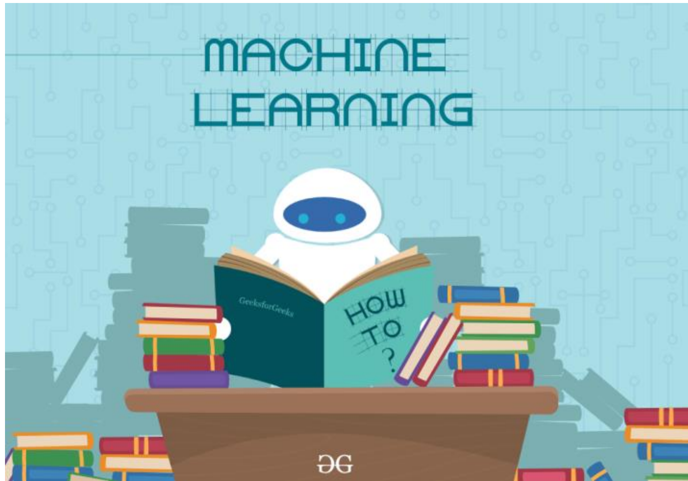
A disciplina

- Continuação de ***T319 - Introdução ao Aprendizado de Máquina I***.
- ***Curso introdutório*** onde veremos os conceitos básicos de funcionamento dos seguintes algoritmos de ***machine learning*** (ML):
  - Classificadores
    - Regressão Logística
    - Regressão Softmax
    - k-Vizinhos mais Próximos
  - Redes Neurais
  - Clustering
    - k-Means
- O curso terá sempre uma parte ***expositiva*** e outra ***prática*** para fixação dos conceitos introduzidos.
  - Quizzes e exercícios envolvendo o uso dos algoritmos discutidos.

Objetivo do curso



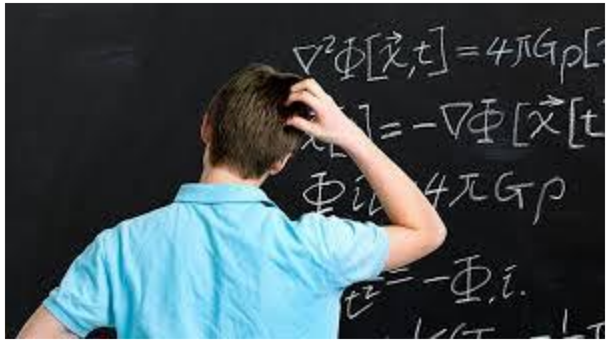
- O objetivo principal do curso é apresentar
  - os conceitos fundamentais da teoria do aprendizado de máquina.
  - um conjunto de ferramentas (ou seja, algoritmos) de aprendizado de máquina para solução de problemas.
- Ao final do curso vocês devem ser capazes de
  - Entender e discutir sobre os principais algoritmos de ML.
  - Compreender a terminologia utilizada na área.
  - Aplicar algoritmos de ML para a resolução de problemas.
  - Analisar e entender novos algoritmos de ML.
  - Criar seus próprios projetos.



Critérios de Avaliação

- 2 trabalhos com peso de 85% cada.
  - Envolvendo questões teóricas e/ou práticas.
- 2 conjuntos de exercícios (quizzes e laboratórios) com peso de 15% cada.
  - Podem sempre ser entregues até a próxima aula.
  - Podem ser resolvidos em grupo, mas entregas devem ser individuais.
  - Exercícios serão atribuídos através de tarefas do MS Teams.

- **Frequência**
  - Gerada automaticamente pelo Teams.
  - Por favor, acompanhem suas presenças no portal.



Referências

[1] Stuart Russell and Peter Norvig, “*Artificial Intelligence: A Modern Approach*,” Prentice Hall Series in Artificial Intelligence, 3rd ed., 2015.

[2] Aurélien Géron, “*Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems*”, 1st ed., O'Reilly Media, 2017.

[3] Joseph Misiti, “*Awesome Machine-Learning*,” on-line data base with several free and/or open-source books (https://github.com/josephmisiti/awesome-machine-learning).

[4] Andriy Burkov, “*The Hundred-Page Machine-Learning Book*,” Andriy Burkov 2019.

[5] C. M. Bishop, “*Pattern Recognition and Machine Learning*,” Springer, 1st ed., 2006.

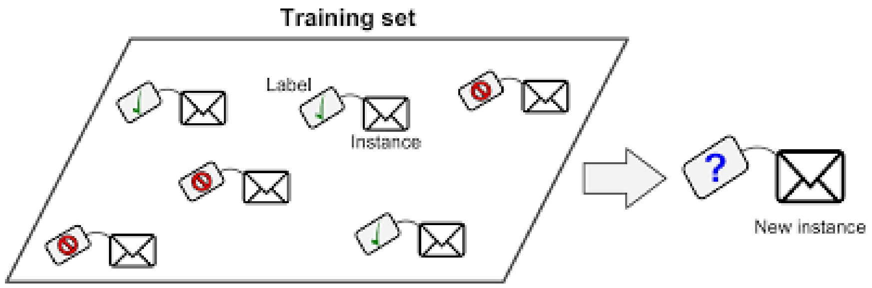
[6] S. Haykin, “*Neural Networks and Learning Machines*,” Prentice Hall, 3ª ed., 2008.

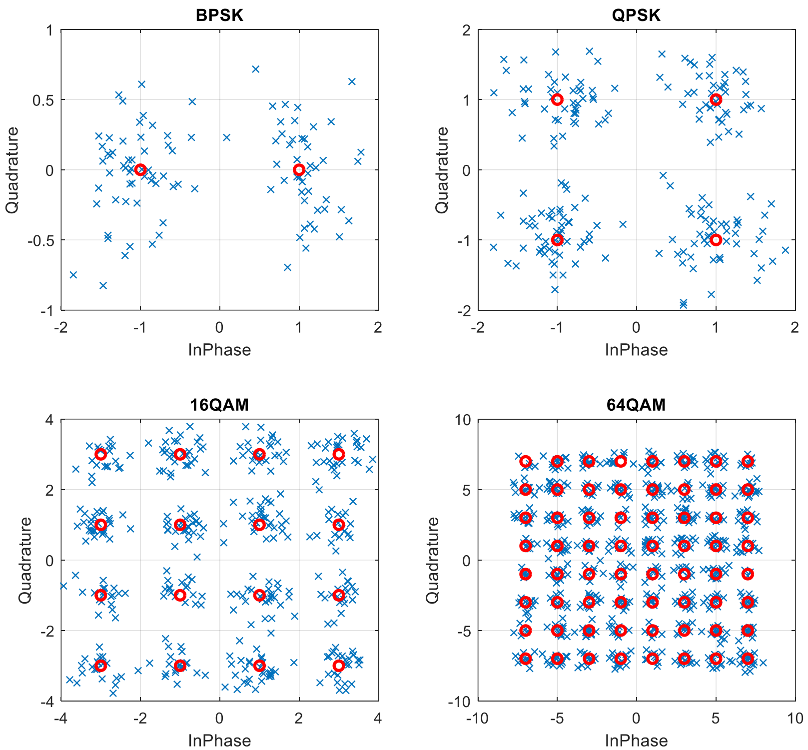
[7] Coleção de livros: https://drive.google.com/drive/folders/1IyIIMu1w6POBhrVnw11yqXXy6BjC439j?usp=sharing

Avisos

- Entregas de exercícios (laboratórios e quizzes) devem ser feitas no MS Teams.
  - Se atentem às datas/horários de entrega.
- Todo material do curso será disponibilizado no MS Teams e no GitHub:
  - https://github.com/zz4fap/t320_aprendizado_de_maquina
- Horários de Atendimento
  - Professor: quintas-feiras das 18:30 às 19:30 e sextas-feiras das 15:30 às 16:30.
  - Monitor (Pedro Rezende: ***pedro_rafael@get.inatel.br***): Todas as sextas-feiras das 17:30 às 18:30.
  - Atendimento via MS Teams.

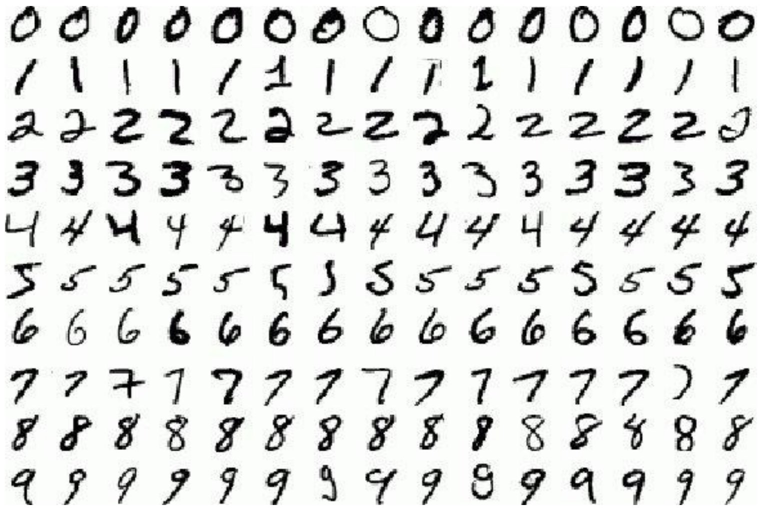
Motivação para tarefas de classificação

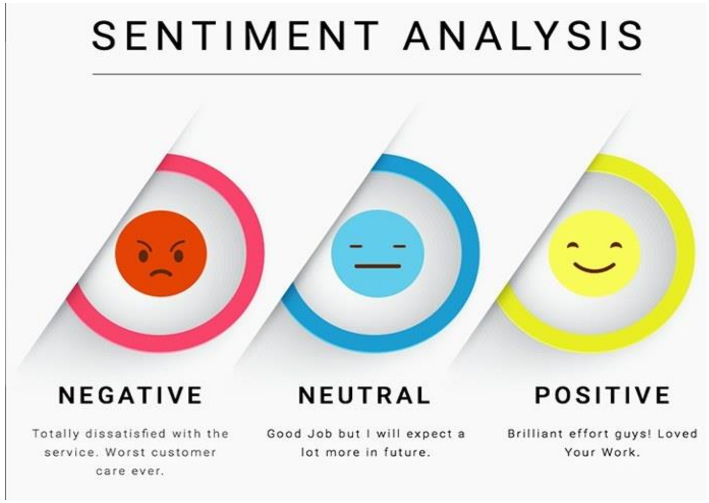




- Em alguns casos, nós precisamos encontrar funções, $f(x)$, que mapeiem os atributos de entrada em determinados valores discretos, ou seja, em classes. Por exemplo
  - Classificação de emails entre SPAM e HAM (legítimo).
  - Classificação de objetos.
  - Detecção ou classificação de símbolos.
  - Classificação de modulações (QPSK, AM, FM, etc.)

Motivação para tarefas de classificação





- Reconhecimento de dígitos escritos à mão.
- Classificação de texto.
- Classificação de sentimentos.

Definição do problema de classificação

- **Problema**: encontrar uma função $f(\boldsymbol{x})$ que atribua a cada ***exemplo de entrada***, $\boldsymbol{x}$, uma das Q classes possíveis, $C_q, q = 1, \dots, Q$ e a qual o exemplo pertence.
  - As classes podem ser
    - Spam e not spam (ham).
    - Dígitos de 0 a 9.
    - Símbolos de uma modulação específica.
    - Objetos (carros, cães, gatos, etc.)
- Semelhante ao problema da ***regressão linear***, existe um conjunto de treinamento com N **exemplos** e **rótulos** $\{\boldsymbol{x}(i); y(i)\}_{i=0}^{N-1}$ que é utilizado para treinar um ***classificador***, onde
  - $\boldsymbol{x}(i) = [x_1(i) \quad \cdots \quad x_K(i)]^T \in \mathbb{R}^{K\times 1}$ representa o *i*-ésimo vetor exemplo de entrada, o qual é caraterizado por K atributos, $x_1(i), \dots, x_K(i)$
  - e $y(i)$ representa o *i*-ésimo ***rótulo***.
- Portanto, como vocês já devem ter percebido, ***classificadores*** são algoritmos de ***treinamento supervisionado***.

Representação da saída desejada

- A ***saída desejada*** de um classificador para um ***exemplo de entrada***, $\boldsymbol{x}(i)$ deve ser um valor que identifique a ***classe*** à qual $\boldsymbol{x}(i)$ pertence.
- Sendo assim, a saída $\hat{y}(i)$ de um ***classificador***, é uma variável ***categórica*** (ou seja, ***discreta***).
- Portanto, para realizarmos o treinamento do ***modelo de classificação***, devemos escolher uma ***representação numérica*** para a ***saída desejada***, y.
- Assim, como veremos a seguir, duas opções podem ser adotadas, dependendo se a classificação é ***binária*** ou ***multi-classes***.

Representação da saída desejada

- **Classificação binária**: existem apenas duas classes possíveis, C_1 e C_2, onde C_1 é chamada de ***classe negativa*** e C_2 a ***classe positiva***.
- Portanto, nesse caso, podemos utilizar ***uma única saída escalar binária*** para indicar a ***classe*** correspondente ao ***exemplo de entrada***:

$$y(i) = \begin{cases} 0, & \boldsymbol{x}(i) \in C_1 \\ 1, & \boldsymbol{x}(i) \in C_2 \end{cases}.$$

- Assim, $y(i) \in \mathbb{R}^1$, de maneira que o classificador realiza um mapeamento $\mathbb{R}^K \rightarrow \mathbb{R}^1$, ou seja, $y = f(\boldsymbol{x})$, onde $\boldsymbol{x} \in \mathbb{R}^K$ e $y \in \mathbb{R}^1$.
- Também é possível utilizar $y(i) = -1$ para $\boldsymbol{x}(i) \in C_1$, ou seja

$$y(i) = \begin{cases} -1, & \boldsymbol{x}(i) \in C_1 \\ 1, & \boldsymbol{x}(i) \in C_2 \end{cases}.$$

Representação da saída desejada

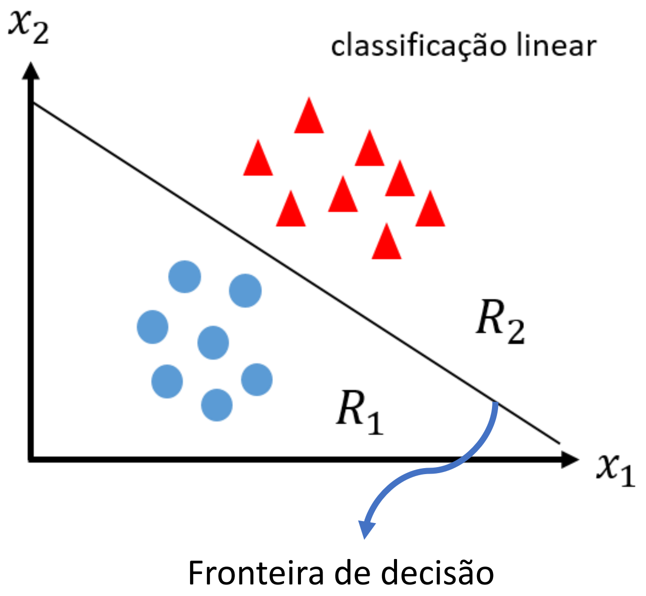
- **Classificação multi-classes**: existem mais de 2 classes possíveis ($Q > 2$).
  - Geralmente, nesse caso, o classificador terá Q saídas.
  - Uma estratégia bastante utilizada para representar estas classes é conhecida como codificação ***one-hot***.
- ***Codificação one-hot***: utiliza uma representação ***vetorial binária*** para as saídas.
  - Ou seja, as saídas são vetores com o valor 1 no elemento representando a classe do exemplo de entrada e 0 nos demais elementos.
  - Nesse caso, o ***classificador*** possui ***múltiplas saídas***, cada uma representando uma classe específica.
  - **Exemplo**: imaginemos um classificador de notícias com quatro classes possíveis: *esportes*, *política*, *ciências* e *variedades*. Como seria a representação com codificação ***one-hot***?

$$\left.\begin{array}{ll} \textit{esportes:} & [1 \quad 0 \quad 0 \quad 0]^T \\ \textit{política:} & [0 \quad 1 \quad 0 \quad 0]^T \\ \textit{ciências:} & [0 \quad 0 \quad 1 \quad 0]^T \\ \textit{variedades:} & [0 \quad 0 \quad 0 \quad 1]^T \end{array}\right\}$$

Assim, $\boldsymbol{y}(i) \in \mathbb{R}^{Q\times 1}$, de maneira que o classificador realiza um mapeamento $\mathbb{R}^K \to \mathbb{R}^Q$.

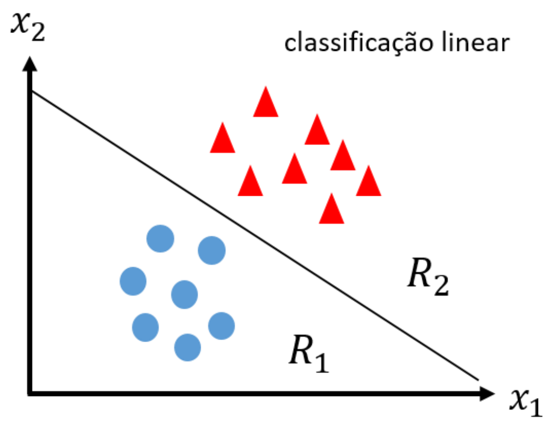
Fronteiras de decisão de um classificador

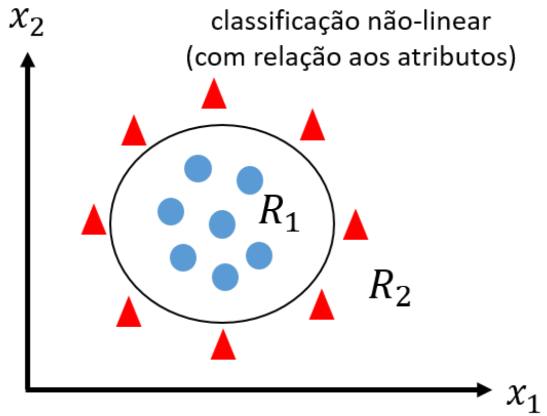
- Antes, usávamos ***funções hipótese*** para aproximar um ***modelo gerador***, agora, as usaremos para separar classes.
- Para facilitar o entendimento, vamos imaginar o ***espaço bi-dimensional***, $\mathbb{R}^2$, criado pelos ***atributos*** x_1 e x_2.
- Esse espaço pode ser dividido em ***regiões de decisão***, $R_i, i = 1, \dots, Q$, onde cada ***região*** corresponde a uma classe.
- As regiões de decisão são separadas por ***fronteiras de decisão***.
- Uma ***fronteira de decisão*** corresponde a uma ***superfície*** (também chamada de ***superfície de separação***) no ***espaço de atributos*** que separe as classes de forma ótima.

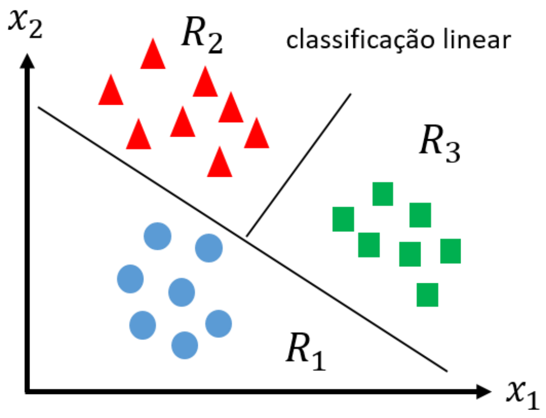


Fronteiras de decisão de um classificador

- As ***superfícies de decisão*** podem ser ***lineares*** (e.g., retas e planos) ou ***não-lineares*** (e.g., círculos).
- As ***superfícies de decisão*** são definidas por ***funções*** (lineares ou não) que separam as classes.
- Essas funções são normalmente chamadas de ***funções discriminantes***, pois separam as classes.
- As figuras mostram ***regiões de decisão*** em problemas de classificação ***binária*** e ***multi-classes***.







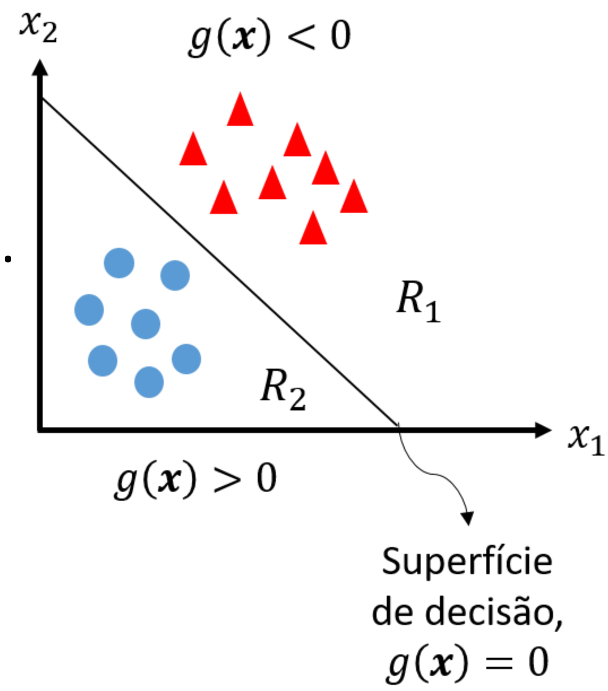
Funções discriminantes lineares

- Em geral, uma ***função discriminante linear*** pode ser escrita da seguinte forma

$$g(\boldsymbol{x}) = a_0 + a_1 x_1 + a_2 x_2 + \ \ldots + a_K x_K = \boldsymbol{a}^T \boldsymbol{x},$$

que nada mais é do que uma ***combinação linear dos pesos***, assim como nós vimos na regressão linear.

- $g(\boldsymbol{x})$ também pode ser vista como um ***hiperplano*** que separa as classes. Um ***hiperplano*** pode ser 1 ponto em 1D, uma reta em 2D, um plano em 3D, etc.
  - O coeficiente a_0 (***bias***) dá o deslocamento com relação à origem.
  - E o restante dos pesos determina a orientação do ***hiperplano***.



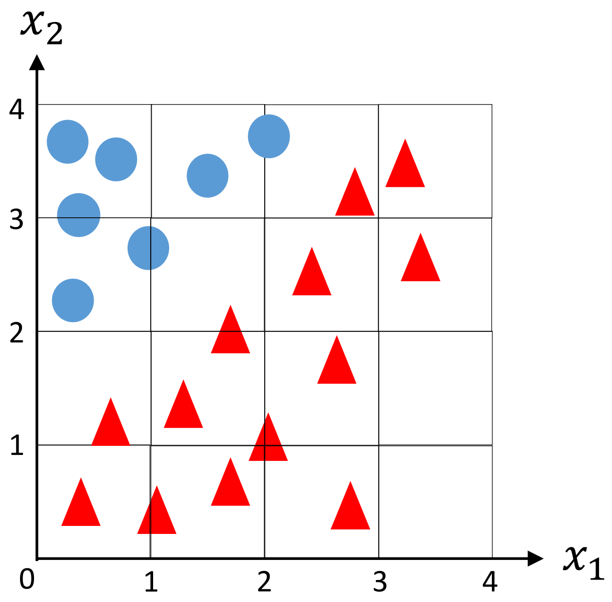
- A ideia aqui é encontrar os pesos da ***função discriminante*** de tal forma que

$$C_q = \begin{cases} 1, & g(\boldsymbol{x}) < 0 \\ 2, & g(\boldsymbol{x}) > 0 \\ \text{uma ou outra,} & g(\boldsymbol{x}) = 0 \end{cases}$$

Indeterminação: empate entre as classes.

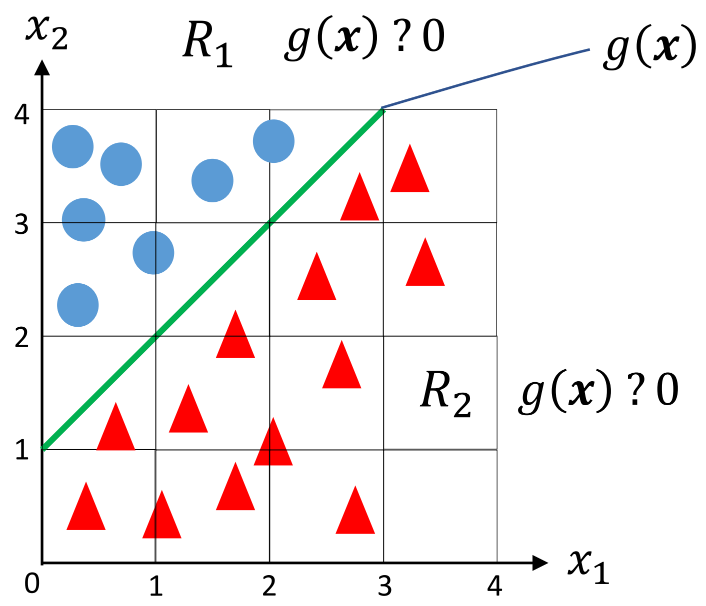
- OBS.: Como vimos anteriormente, podemos ter também ***funções discriminates não-lineares em relação aos atributos***, e.g., $g(\boldsymbol{x}) = a_0 + x_1^2 + x_2^2$ (eq. de um círculo centrado na origem).

Exemplo: Encontrando os pesos da função discriminante, $g(\boldsymbol{x})$



- Analisem a figura.
- Temos 2 classes, 2 atributos, x_1 e x_2, e queremos encontrar uma ***função discriminate***, $g(\boldsymbol{x})$, que as separe.
- Qual formato deve ter esta ***função discriminante***?
  - O formato mais simples (navalha de Occam) é o de uma reta.

Exemplo: Encontrando os pesos da função discriminante, $g(\boldsymbol{x})$

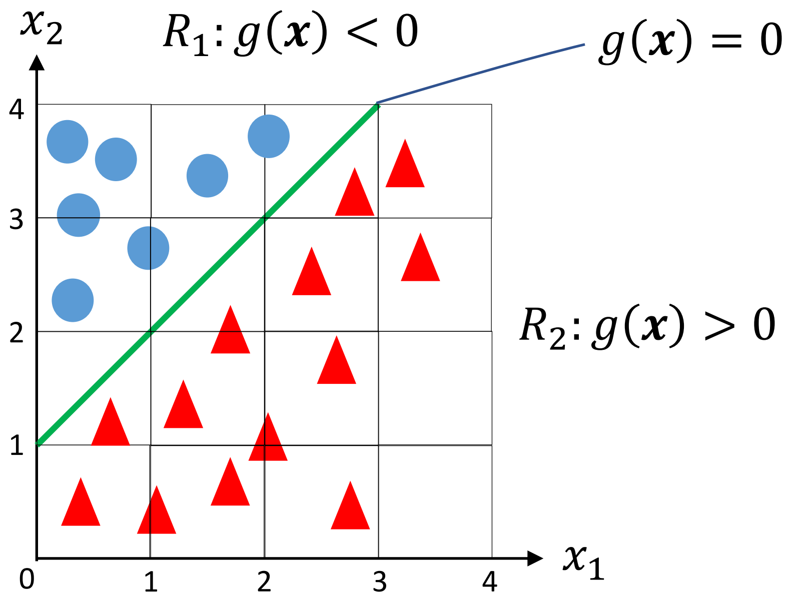


- Visualmente, traçamos uma reta em uma posição que separe as classes da melhor forma possível.
- A ***função discriminante*** que representa esta reta é definida como

$$g(\boldsymbol{x}) = a_0 + a_1 x_1 + a_2 x_2$$

- Agora que definimos uma função e sua posição no gráfico, precisamos encontrar os ***pesos*** e as ***regiões de decisão***.

Exemplo: Encontrando os pesos da função discriminante, $\boldsymbol{g(x)}$



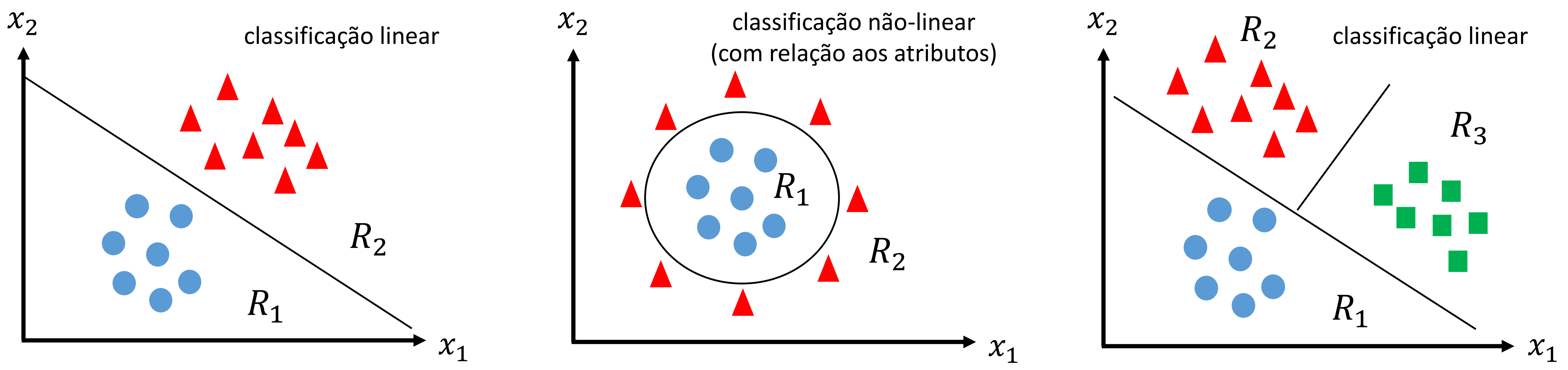
- Temos 3 incógnitas e 3 equações:
  - $(x_1 = 0, x_2 = 1) \rightarrow 0 = a_0 + a_2 \therefore a_0 = -a_2$
  - $(x_1 = 1, x_2 = 2) \rightarrow 0 = a_0 + a_1 + 2a_2 \therefore a_1 = -(a_0 + 2a_2)$
  - $(x_1 = 2, x_2 = 3) \rightarrow 0 = a_0 + 2a_1 + 3a_2 \therefore a_1 = -(a_0 + 3a_2)/2$
- Resolvendo o sistema, encontramos $a_0 = 1$, $a_1 = 1$, $a_2 = -1$, então
  - $g(\boldsymbol{x}) = 1 + x_1 - x_2$

[Exemplo: encontrando_pesos_da_função_discriminante.ipynb]

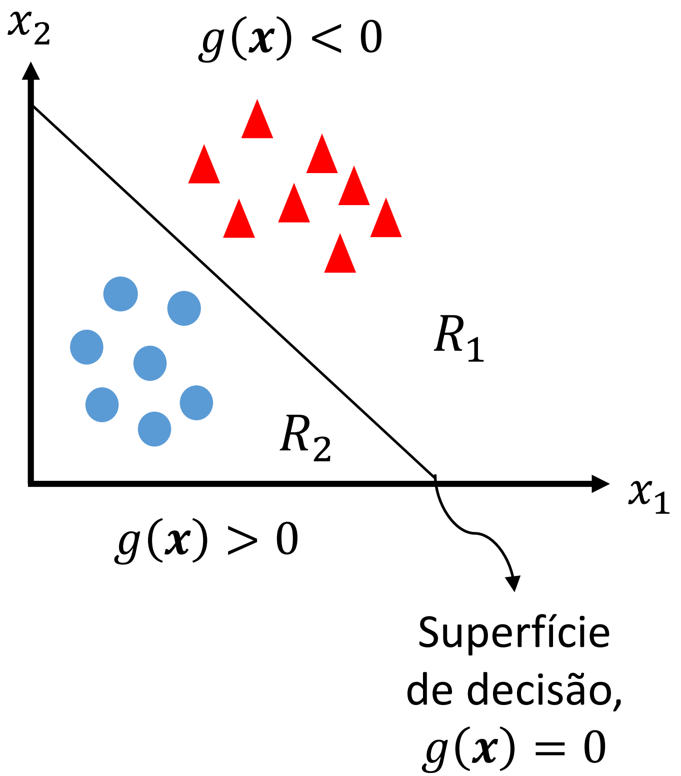
Tarefas

- **Quiz**: *"T320 - Quiz - Classificação (Parte I)"* que se encontra no MS Teams.
- **Exercício Prático**: **Laboratório #1**.
  - Pode ser baixado do MS Teams ou do GitHub.
  - Pode ser respondido através do link acima (na nuvem) ou localmente.
  - Instruções para resolução e entrega dos laboratórios.
  - **Atividades podem ser feitas em grupo, mas as entregas devem ser individuais.**

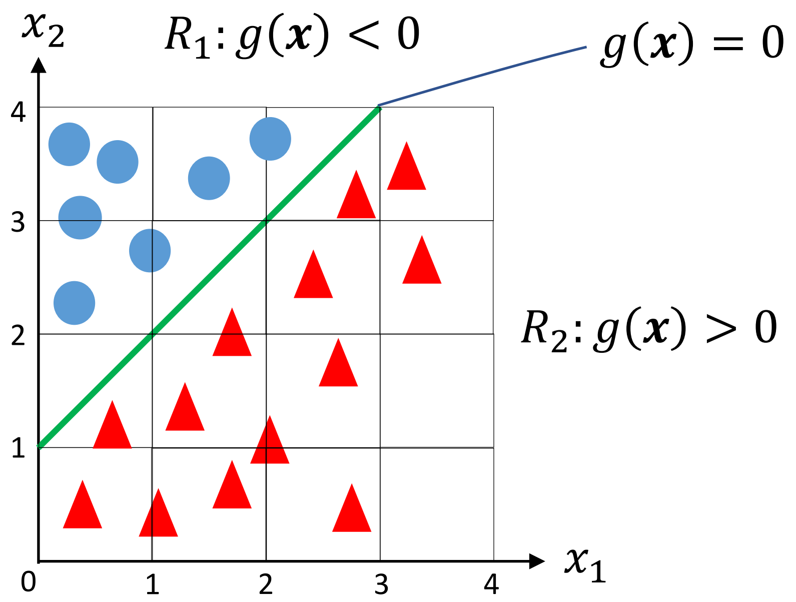
Obrigado!



x_2
classificação linear
R_2
R_1
x_1
x_2
classificação não-linear
(com relação aos atributos)
R_1
R_2
x_1
x_2
R_2
classificação linear
R_3
R_1
x_1



x_2
$g(\boldsymbol{x}) < 0$
R_1
R_2
x_1
$g(\boldsymbol{x}) > 0$
Superfície
de decisão,
$g(\boldsymbol{x}) = 0$



x_2
$R_1: g(\boldsymbol{x}) < 0$
$g(\boldsymbol{x}) = 0$
$R_2: g(\boldsymbol{x}) > 0$
x_1
0
1
2
3
4