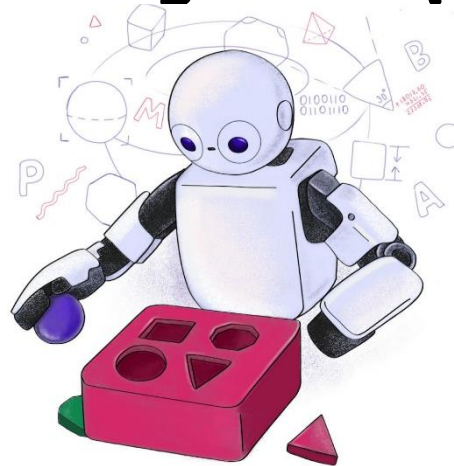


T320 - Introdução ao Aprendizado de Máquina II: *Classificação (Parte I)*



Inatel

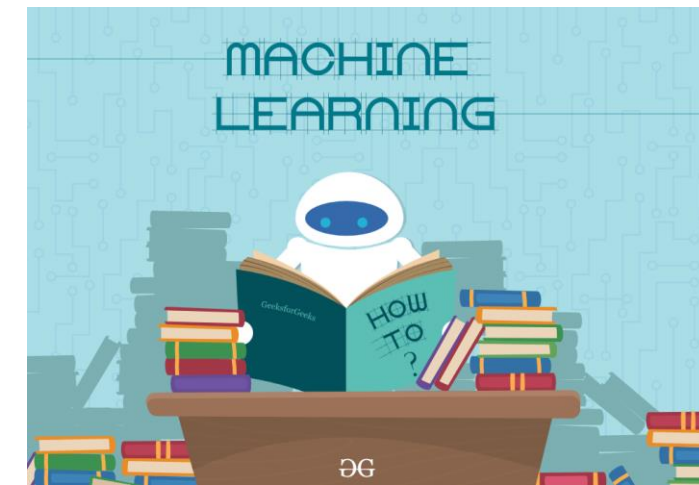
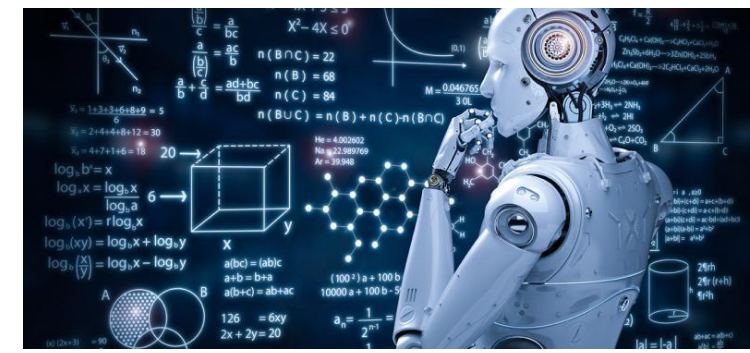
Felipe Augusto Pereira de Figueiredo
felipe.figueiredo@inatel.br

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
 - Redes Neurais
 - Clustering
 - k-Means
- O curso terá sempre uma parte ***expositiva*** e outra ***prática*** para fixação dos conceitos introduzidos.
 - Quizzes e exercícios envolvendo os conceitos discutidos.

Objetivo do curso

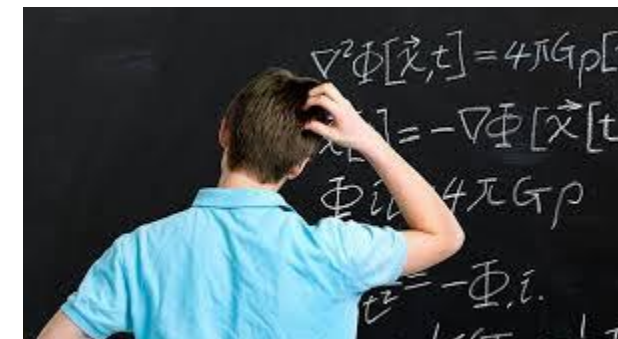
- O objetivo principal do curso é apresentar
 - os ***conceitos fundamentais*** da teoria do aprendizado de máquina.
 - um ***conjunto de ferramentas*** (ou seja, algoritmos) de aprendizado de máquina para solução de problemas.
- Ao final do curso vocês devem ser capazes de
 - Entender e discutir sobre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



Critérios de Avaliação



- **Dois (2) trabalhos** em grupo com peso de 85% cada.
 - Envolvendo questões práticas e/ou teóricas.
 - **Uma parte de cada trabalho será feita presencialmente.**
- Dois (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 Teams.
- Extra: 10% da nota da FETIN na segunda nota.
 - O trabalho precisa usar ML.
- **Frequência**
 - Gerada automaticamente pelo Teams.
 - Por favor, acompanhem suas frequências no portal.



Cronograma

Aula	Data	Dia	Horário	Atividade
1	5/8/2023	Sábado	10:00 às 11:40	Introdução ao Aprendizado de Máquina
2	12/8/2023			Introdução ao Aprendizado de Máquina
3	19/8/2023			Introdução ao Aprendizado de Máquina
4	26/8/2023			Introdução ao Aprendizado de Máquina
5	2/9/2023			Introdução ao Aprendizado de Máquina
6	9/9/2023			Introdução ao Aprendizado de Máquina
7	16/9/2023			Introdução ao Aprendizado de Máquina
8	23/9/2023			Avaliação Presencial I (Projeto I) (Sala ???)
9	30/9/2023			Introdução ao Aprendizado de Máquina
10	7/10/2023			Introdução ao Aprendizado de Máquina
11	14/10/2023			Introdução ao Aprendizado de Máquina
12	21/10/2023			Introdução ao Aprendizado de Máquina
13	28/10/2023			Introdução ao Aprendizado de Máquina
14	4/11/2023			Introdução ao Aprendizado de Máquina
15	11/11/2023			Introdução ao Aprendizado de Máquina
16	18/11/2023			Avaliação Presencial II (Projeto II) (Sala???)
17	25/11/2023			Introdução ao Aprendizado de Máquina
18	2/12/2023			Introdução ao Aprendizado de Máquina
19	9/12/2023			Introdução ao Aprendizado de Máquina
20	16/12/2023			Introdução ao Aprendizado de Máquina

Referências

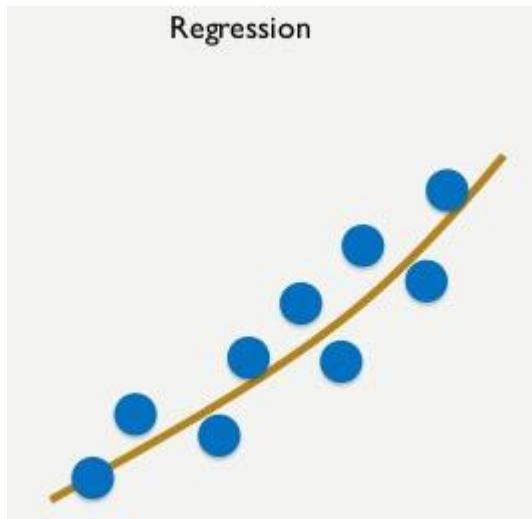
- [1] Stuart Russell and Peter Norvig, *“Artificial Intelligence: A Modern Approach,”* Prentice Hall Series in Artificial Intelligence, 3rd ed., 2015.
- [2] Aurélien Géron, *“Hands-On Machine Learning with Scikit-Learn and TensorFlow: Concepts, Tools, and Techniques to Build Intelligent Systems”*, 1st 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/1lylIMu1w6POBhrVnw11yqXXy6BjC439j?usp=sharing>

Avisos

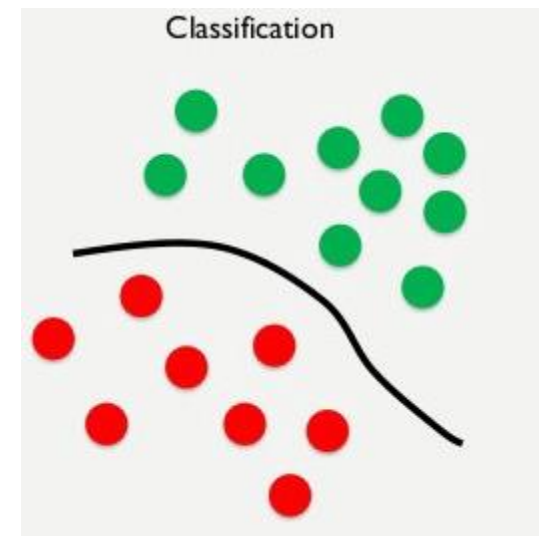
- Toda nossa comunicação (avisos, atendimentos e tarefas) será feita via Teams.
- Todas as aulas serão gravadas e os vídeos ficarão disponíveis na pasta “Recordings” dentro de “Arquivos”.
- Todo material do curso está disponível no GitHub:
 - https://github.com/zz4fap/t320_aprendizado_de_maquina
- Entregas de exercícios (laboratórios e quizzes) devem ser feitas através do Teams.
 - Se atentem às datas e horários de entrega das atividades.
- Vídeos do minicurso de Python e de como usar o Colab estão na pasta "Recordings" dentro de “Arquivos”.
- Horários de Atendimento
 - Professor: quartas-feiras das 17:30 às 18:30 e quintas-feiras das 16:00 às 17:00.
 - Monitora (**Arielli Ajudarte: arielli.a@get.inatel.br**): terças-feiras das 18:00 às 19:00.
 - Atendimento remoto via Teams.

Classificação

- Tarefa (ou problema) de aprendizado supervisionado.
 - As saídas esperadas (rótulos) são conhecidas.
- Envolve encontrar uma função, $f(x)$, que mapeie os atributos de entrada em **valores discretos**, ou seja, em classes.

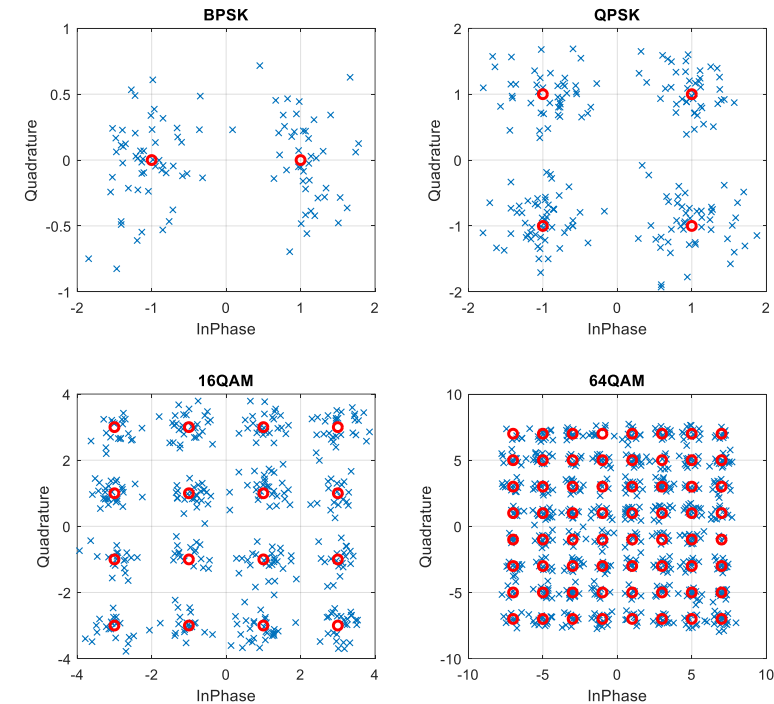
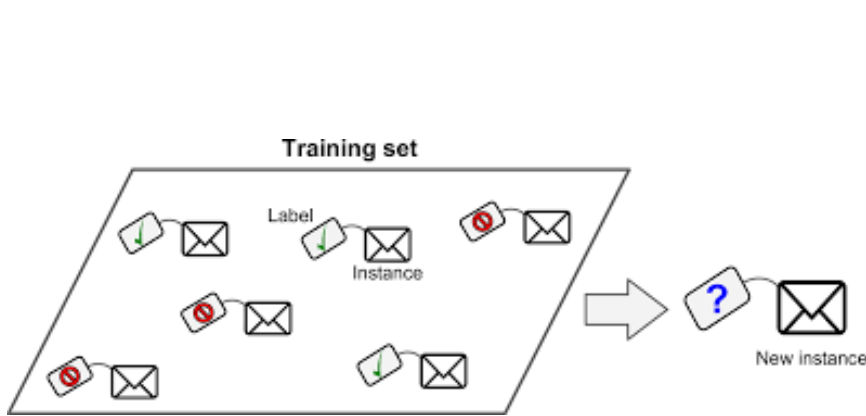


$f(x)$ **aproxima** o comportamento dos dados.



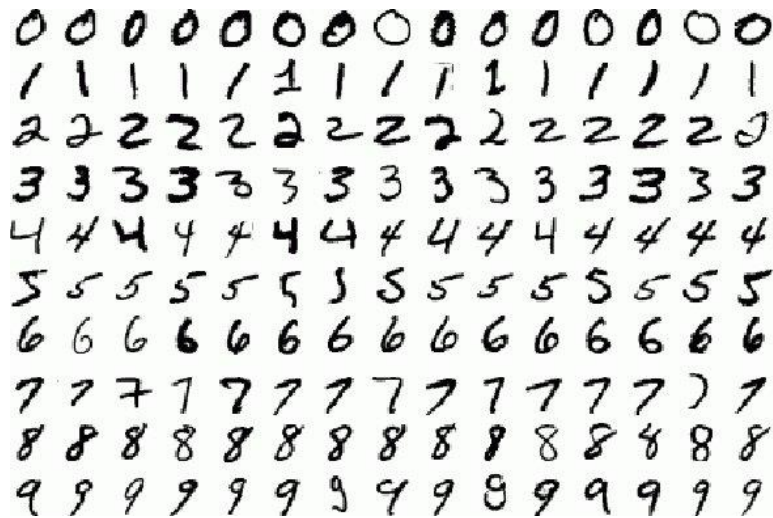
$f(x)$ **separa** os dados em classes.

Tarefas de classificação

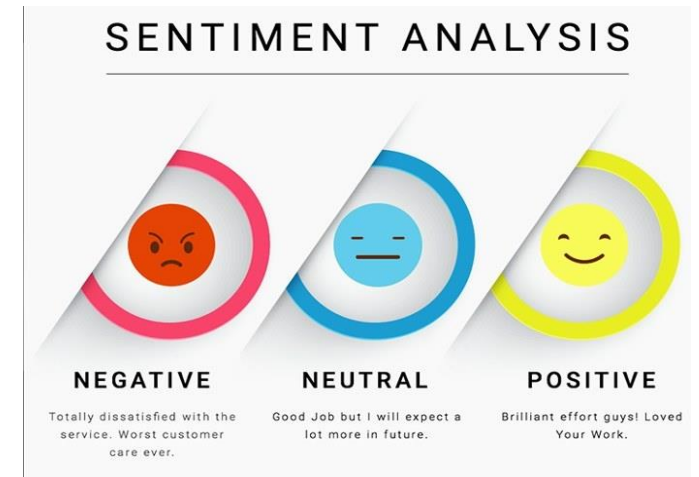


- Classificação de emails entre *spam* e *ham* (legítimo).
- Classificação de objetos em imagens ou vídeos.
- Detecção ou classificação de símbolos de modulações digitais.
- Classificação de modulações (QPSK, AM, FM, etc.).

Tarefas de classificação



- Reconhecimento de texto e dígitos.
- Classificação de texto.
- Classificação de sentimentos.
- Classificação do doenças pulmonares



Definição do problema de classificação

- **Problema:** encontrar uma função, $f(\mathbf{x})$, que atribua a um ***exemplo de entrada***, \mathbf{x} , uma de Q classes possíveis, as quais denotaremos como $C_q, q = 1, \dots, Q$.
 - Por exemplo, as classes podem ser
 - *Spam* e *ham* (legítimo): $Q = 2$.
 - Dígitos de 0 a 9: $Q = 10$.
 - Símbolos de uma modulação específica (e.g., QPSK: $Q = 4$).
 - Objetos (carros, barcos, cães, gatos, etc.)
- Semelhante ao problema da ***regressão linear***, existe um conjunto de treinamento com N pares de ***vetores de atributos*** e ***rótulos*** $\{\mathbf{x}(i); y(i)\}_{i=0}^{N-1}$ que é utilizado para treinar um ***classificador***, onde
 - $\mathbf{x}(i) = [x_1(i) \ \cdots \ x_K(i)]^T \in \mathbb{R}^{K \times 1}$ representa o i -ésimo vetor de atributos, o qual é composto por K atributos, $x_1(i), \dots, x_K(i)$;
 - e $y(i)$ representa o i -ésimo ***rótulo***.

Representação da saída desejada



- A **saída desejada** (i.e., **rótulo**) de um classificador para um **vetor de atributos**, $x(i)$, deve ser um valor que identifique a qual **classe** o vetor $x(i)$ pertence.
- Sendo assim, a **saída** de um **classificador** é uma variável **categórica** (i.e., **discreta**).
- Portanto, para realizarmos o treinamento do **modelo de classificação**, devemos escolher uma **representação numérica** para as **saídas desejadas**.
- 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** ($Q = 2$): 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 **vetor de atributos**:

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

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

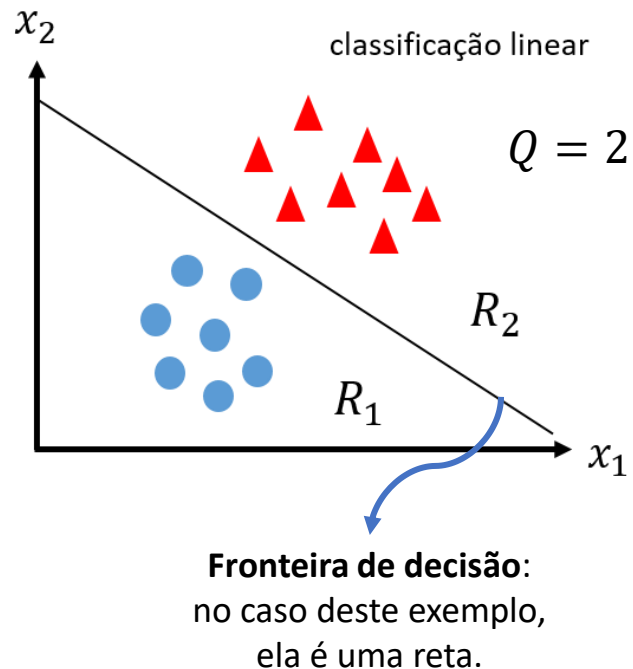
$$y(i) = \begin{cases} -1, & \mathbf{x}(i) \in C_1 \\ 1, & \mathbf{x}(i) \in C_2 \end{cases}.$$

Representação da saída desejada

- **Classificação multi-classes:** existem mais de 2 classes possíveis ($Q > 2$).
 - 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** (Q 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 a codificação **one-hot**?

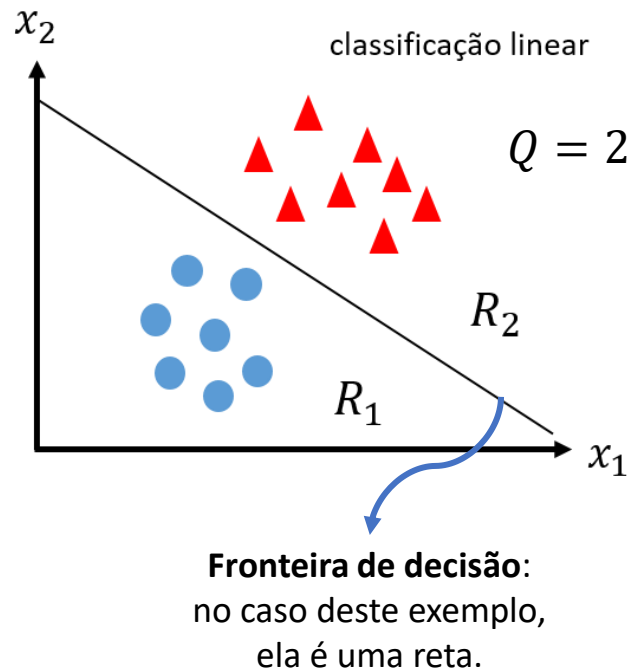
$$\left. \begin{array}{l} \text{esportes:} \quad [1 \quad 0 \quad 0 \quad 0]^T \\ \text{política:} \quad [0 \quad 1 \quad 0 \quad 0]^T \\ \text{ciências:} \quad [0 \quad 0 \quad 1 \quad 0]^T \\ \text{variedades:} [0 \quad 0 \quad 0 \quad 1]^T \end{array} \right\} \text{ Assim, } \mathbf{y}(i) \in \mathbb{R}^{Q \times 1}, \text{ de maneira} \\ \text{que o classificador realiza um} \\ \text{mapeamento } \mathbb{R}^{K \times 1} \rightarrow \mathbb{R}^{Q \times 1}.$$

Fronteiras de decisão de um classificador



- Antes, nós usávamos **funções hipótese** para **aproximar o comportamento de um conjunto de dados**, agora, as usaremos para **separar grupos de dados (i.e., classes)**.
- Para facilitar o entendimento, vamos imaginar o **espaço bi-dimensional**, \mathbb{R}^2 , criado pelos **atributos** x_1 e x_2 , mostrado na figura ao lado.
- Os **pares de atributos** pertencem a duas classes ($Q = 2$):
 - Círculos azuis, pertencentes à classe C_1 .
 - Triângulos vermelhos, pertencentes à classe C_2 .

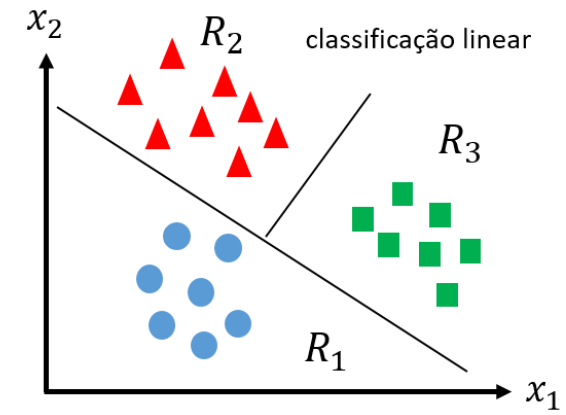
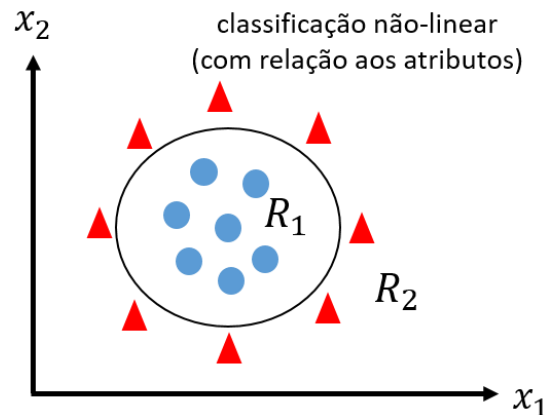
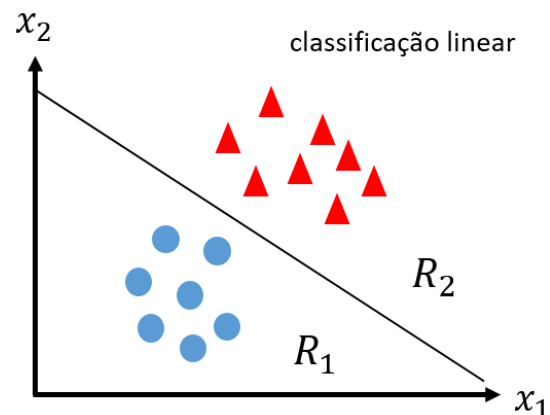
Fronteiras de decisão de um classificador



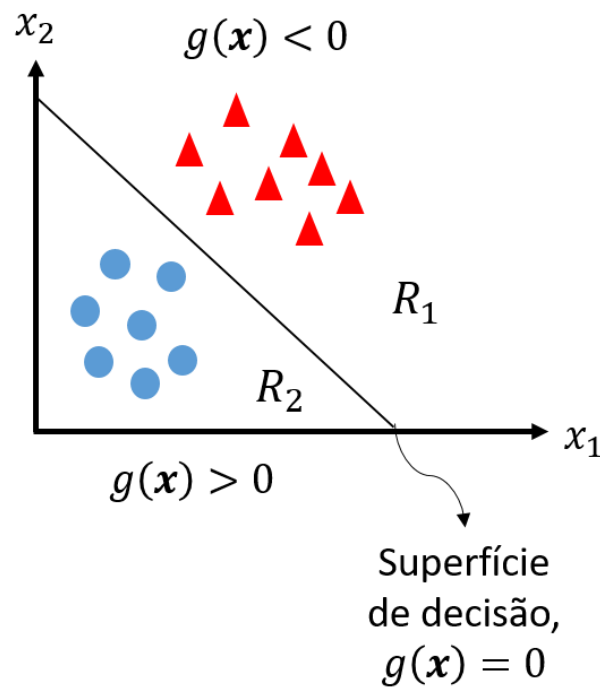
- Esse espaço pode ser dividido em **duas regiões de decisão**, R_1 e R_2 , onde cada **região** corresponde a uma classe.
- As regiões de decisão são separadas por **fronteiras de decisão**, que nada mais são do que **funções**.
- Na figura, como $Q = 2$, temos apenas uma fronteira de decisão.
- Uma **fronteira de decisão** corresponde a uma **superfície** (também chamada de **superfície de separação**) no **espaço de atributos** que separa 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 e elipses).
- 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**.

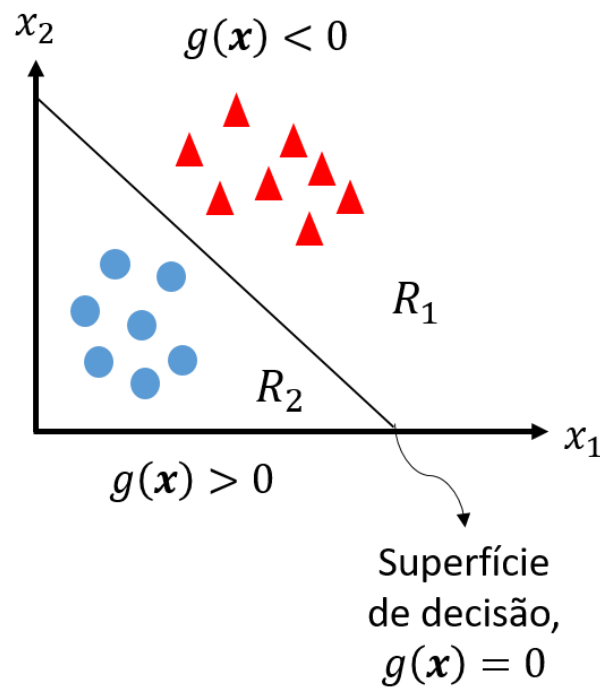


Funções discriminantes



- Em geral, uma **função discriminante** pode ser escrita da seguinte forma
$$g(\mathbf{x}) = a_0 + a_1x_1 + a_2x_2 + \dots + a_Kx_K = \mathbf{a}^T \mathbf{x},$$
que nada mais é do que uma **combinação linear dos atributos em relação aos pesos**, assim como nós vimos em regressão linear.
- $g(\mathbf{x})$ também pode ser interpretada 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**.

Funções discriminantes



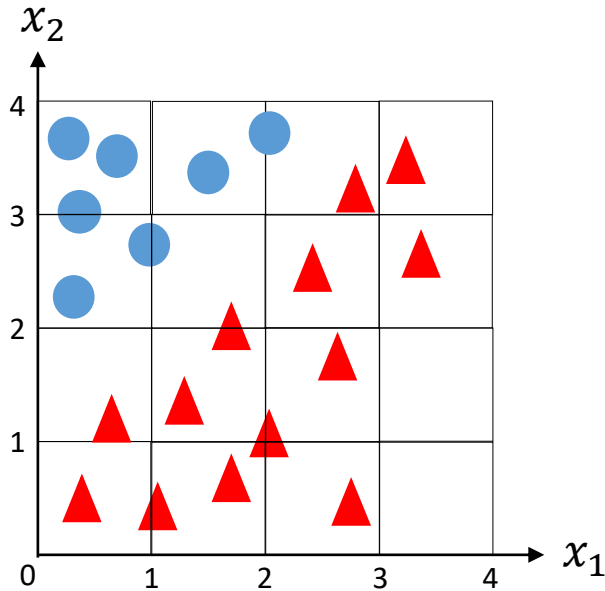
- O **objetivo é encontrar os pesos da função discriminante** de tal forma que a classe escolhida seja:

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

Indeterminação:
empate entre as
classes.

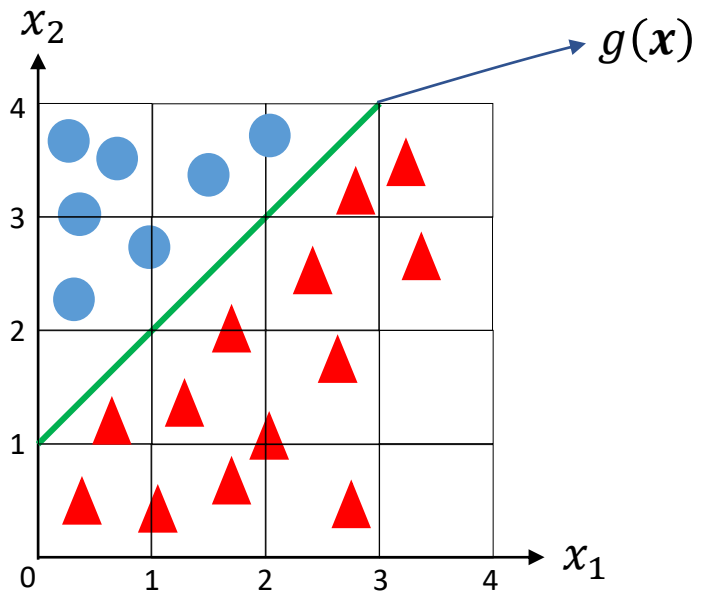
- OBS.: Podemos ter também **funções discriminantes não-lineares em relação aos atributos**, e.g., $g(\mathbf{x}) = a_0 + x_1^2 + x_2^2$ (eq. de um círculo centrado na origem, onde $a_0 = -r^2$).

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



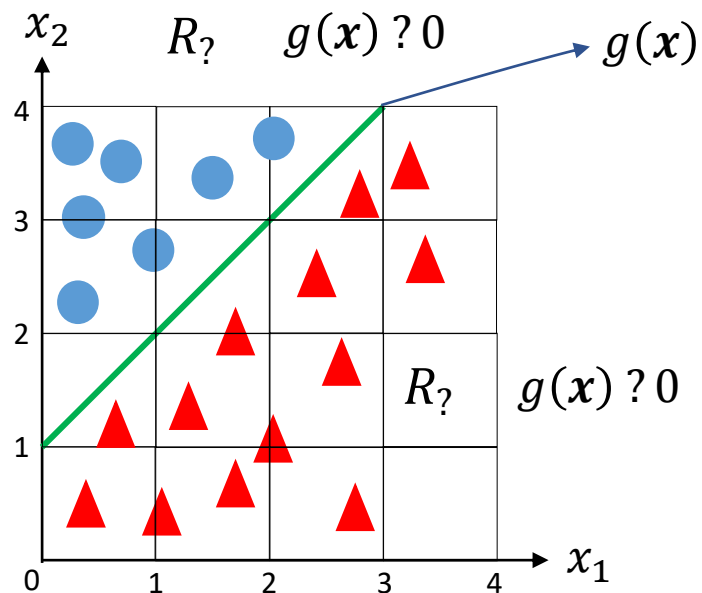
- Analisem a figura ao lado.
- Temos 2 classes, 2 atributos, x_1 e x_2 , e queremos encontrar uma **função discriminante**, $g(\mathbf{x})$, que as separe.
- Qual formato deve ter esta **função discriminante** para que ela tenha boa capacidade de generalização?
 - Lembrem-se do princípio da navalha de Occam: *a explicação mais simples (i.e., menos complexa) é geralmente a mais provável de estar correta.*

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



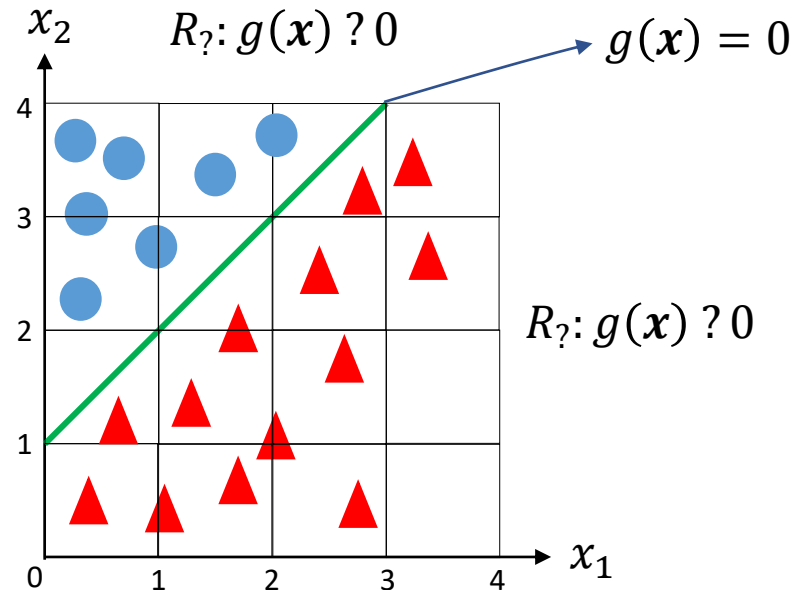
- Qual formato deve ter esta **função discriminante** para que ela tenha boa capacidade de generalização?
 - O formato mais simples, seguindo o princípio da navalha de Occam, é o de uma **reta**.

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



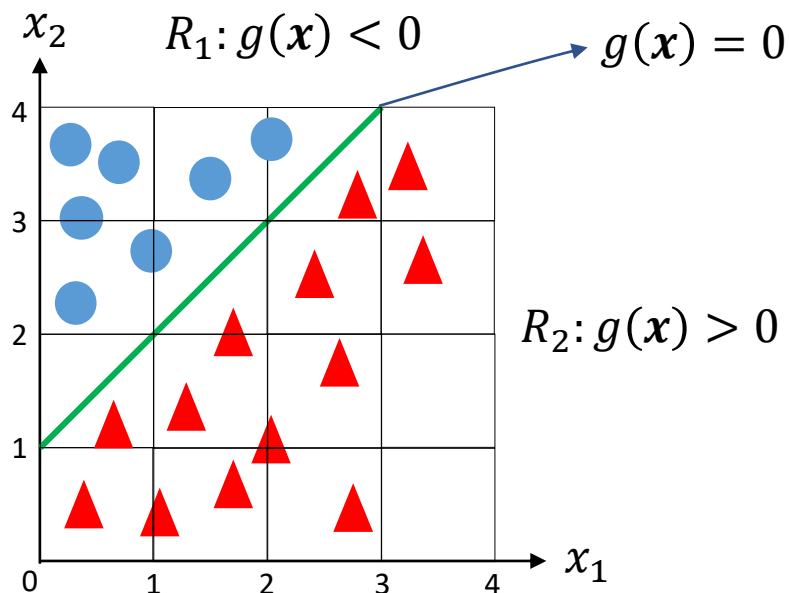
- Visualmente, nós traçamos uma reta em uma posição que separe as classes da melhor forma possível.
- A **função discriminante** que representa esta reta é definida como
$$g(\mathbf{x}) = a_0 + a_1x_1 + a_2x_2$$
- Agora que definimos uma função e sua posição no gráfico, precisamos encontrar os **pesos** e as **regiões de decisão**.
- Como podemos encontrar os pesos?

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



- Se temos 3 incógnitas, precisamos de um sistema com 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
 - $g(\mathbf{x}) = 1 + x_1 - x_2$

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



- Agora, vamos encontrar as **regiões de decisão** substituindo alguns valores em $g(\mathbf{x}) = 1 + x_1 - x_2$.
 - $x_1 = 1$ e $x_2 = 1$ resulta em $g(\mathbf{x}) > 0$.
 - ✓ Região da classe **positiva**, C_2 .
 - $x_1 = 1$ e $x_2 = 3$ resulta em $g(\mathbf{x}) < 0$.
 - ✓ Região da classe **negativa**, C_1 .
 - $x_1 = 1$ e $x_2 = 2$ resulta em $g(\mathbf{x}) = 0$.
 - ✓ **Indeterminação**: não podemos afirmar a qual classe o exemplo pertence.
 - ✓ Podemos atribuir arbitrariamente a uma das duas classes ou escolher a classe que possui maior número de exemplos.

Tarefas

- **Quiz:** “*T320 - Quiz - Classificação (Parte I)*” que se encontra no MS Teams.
- **Exercício Prático:** [Laboratório #1](#).
 - Pode ser acessado através do link acima (Google Colab) ou no GitHub.
 - Se atentem aos prazos de entrega.
 - [Instruções para resolução e entrega dos laboratórios](#).
 - **Laboratórios podem ser resolvidos em grupo, mas as entregas devem ser individuais.**

Obrigado!

