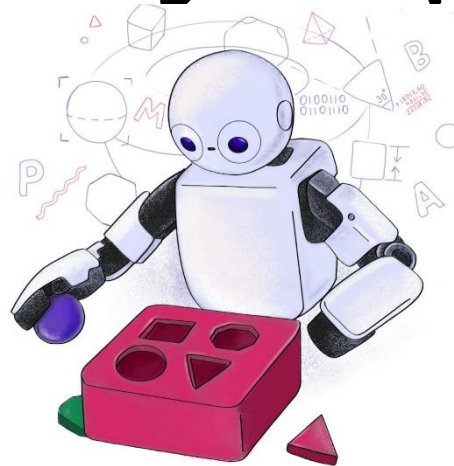


# T320 - Introdução ao Aprendizado de Máquina II: *Classificação (Parte I)*



**Inatel**

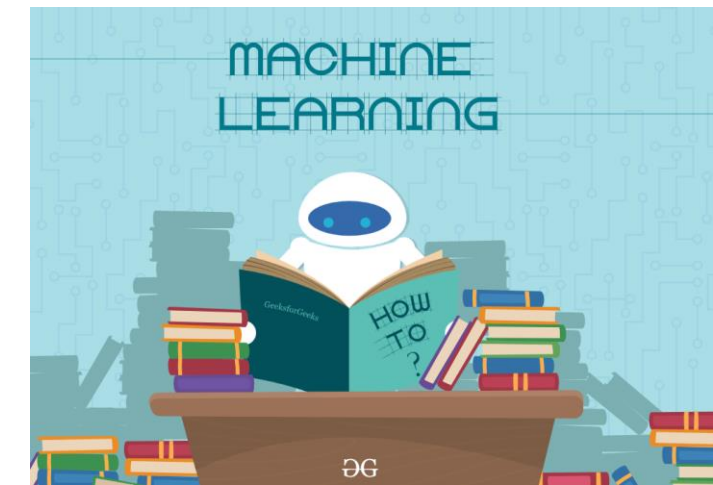
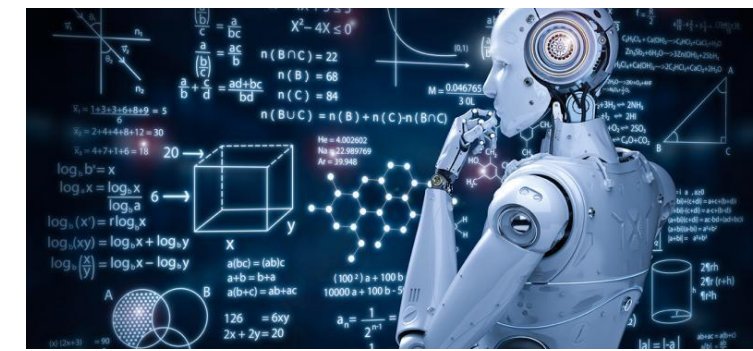
Felipe Augusto Pereira de Figueiredo  
felipe.figueiredo@inatel.br

# 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
- O curso será o mais prático possível, com vários exercícios envolvendo o uso dos algoritmos discutidos.

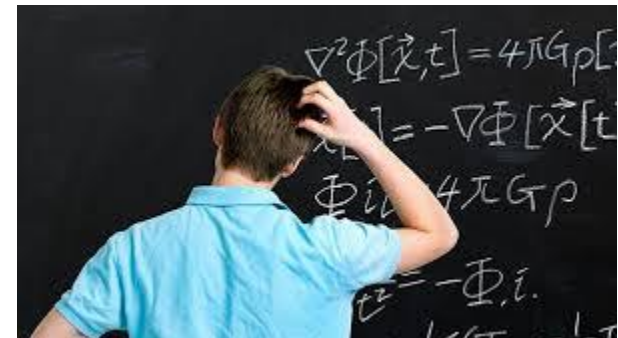
# Objetivo do curso

- O objetivo principal do curso é apresentar à vocês
  - os conceitos fundamentais da teoria do aprendizado de máquina.
  - um conjunto de ferramentas (ou seja, algoritmos) de aprendizado de máquina.
- 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.



# Avaliação do curso

- Avaliações
  - Dois (2) trabalhos valendo 85% da nota.
  - Envolvendo questões teóricas e/ou práticas.
- Atividades
  - Exercícios e quizzes valendo 15% da nota.
  - Ao longo das aulas e para casa.
  - [Entregues no MS Teams.](#)



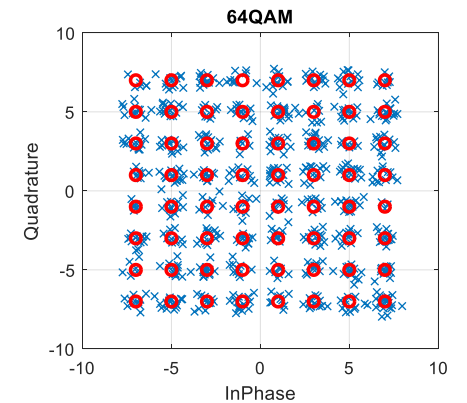
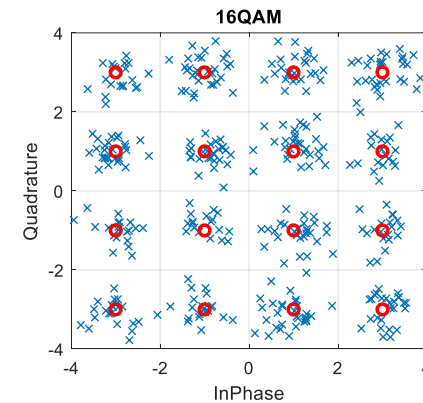
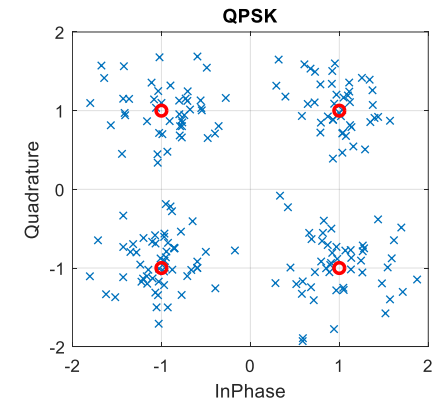
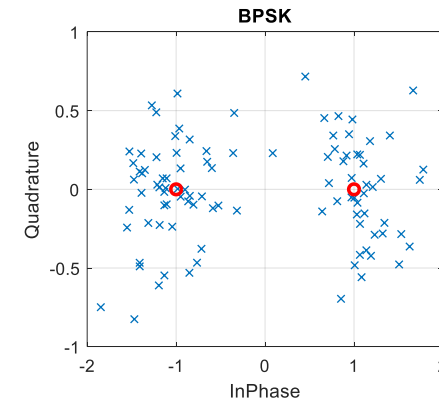
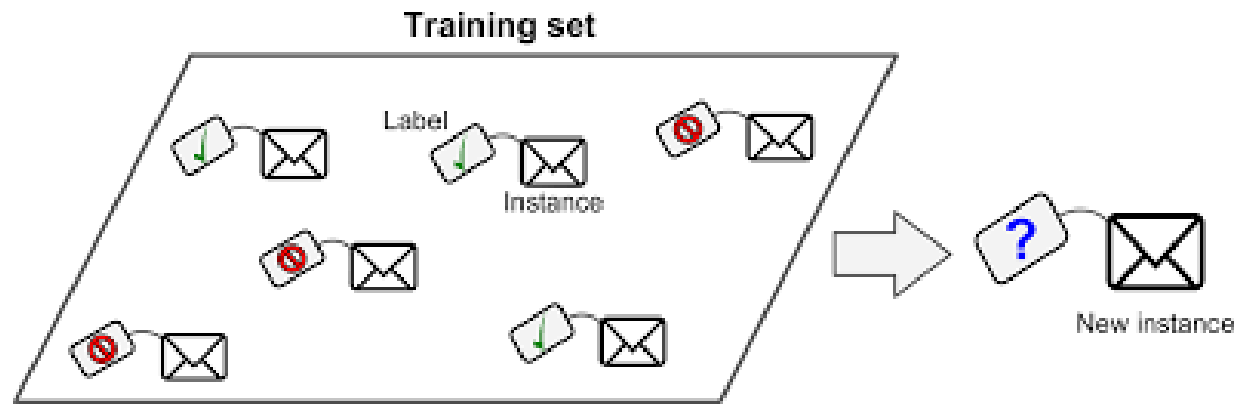
# Referências

- [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

- Entregas de exercícios (laboratórios e quizzes) devem ser feitas no MS Teams.
  - Se atentem às datas/horários de entrega no MS Teams.
- Todo material do curso será disponibilizado no MS Teams e no GitHub:
  - <https://github.com/zz4fap/t320> *aprendizado de maquina*
- Horários de Atendimento
  - Professor: Segundas-feiras das 18:30 às 19:30 e Quartas-feiras das 15:30 às 16:30 via MS Teams.
  - Monitora (Bruna de Souza: ***bruna.br@gea.inatel.br***): Todas as Segundas-feiras das 17:30 às 18:30.

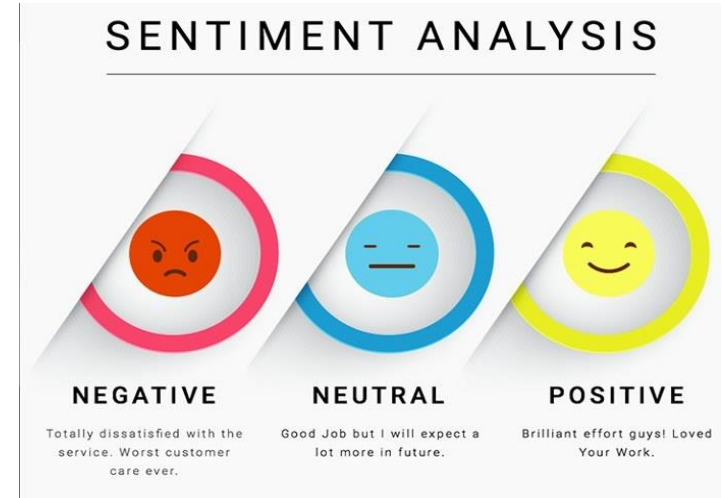
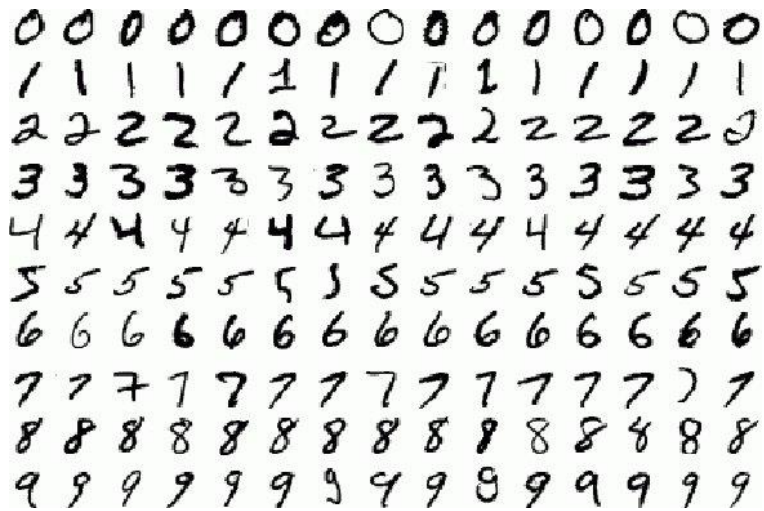
# Motivação



- Classificação de emails entre SPAM e pessoal (HAM).
- Detecção de símbolos (classificação de símbolos).
- Classificação de modulações (QPSK, AM, FM, etc.)



# Motivação



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



# Definição do problema de classificação

- **Problema:** atribuir a cada **exemplo de entrada** o **rótulo** correspondente a uma das  $Q$  classes existentes,  $C_q, q = 1, \dots, Q$ , à 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, cachorro, gato, etc.)
- Semelhante ao problema da regressão linear, existe um conjunto de treinamento com exemplos 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 exemplo de entrada, o qual é caracterizado por  $K$  atributos,  $x_1, \dots, x_K$
  - e  $y(i) \in \mathbb{R}$  representa o  $i$ -ésimo **rótulo**. Como veremos a seguir,  $y$  pode ser um escalar  $\mathbb{R}^1$  ou um vetor  $\mathbb{R}^{Q \times 1}$ .

# Representação da saída desejada

- Como vocês devem ter percebido, classificadores são algoritmos com ***treinamento supervisionado***.
- A saída desejada para um dado ***exemplo de entrada***,  $x$ , deve ser o ***rótulo***,  $y$ , da classe à qual ele pertence.
- Sendo assim, a saída  $y$  de um ***classificador***, é uma variável ***categórica*** (ou seja, ***discreta***).
- Portanto, para realizarmos o treinamento do modelo, é necessário escolher uma ***representação numérica*** para a saída desejada, ou seja,  $y$ .
- Assim, como veremos a seguir, duas opções podem ser adotadas, dependendo do tipo de classificação a ser feita.

# Representação da saída desejada

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

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

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

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

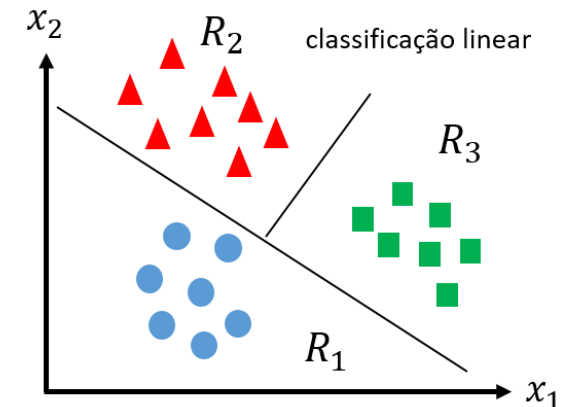
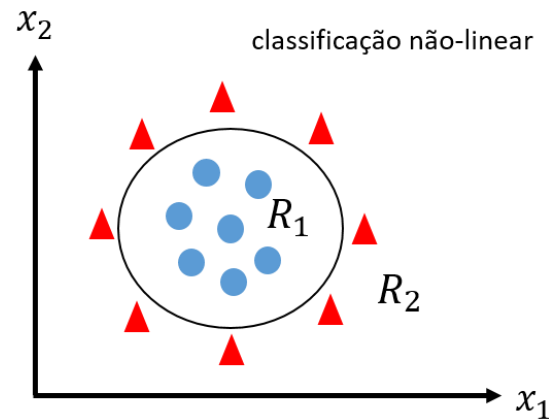
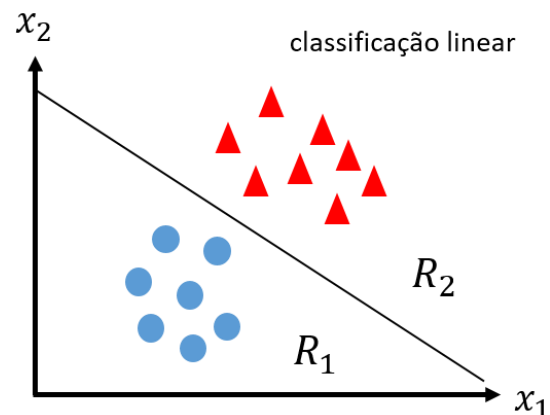
# Representação da saída desejada

- **Classificação multi-classes:** existem mais de 2 classes possíveis ( $Q > 2$ ).
  - Uma estratégia bastante utilizada para representar estas classes é conhecida como ***one-hot encoding***.
- ***One-hot encoding***: utiliza uma representação binária para cada uma das variáveis categóricas.
  - Neste caso, o ***classificador*** produz múltiplas saídas, cada uma representando a ***possibilidade*** como veremos mais tarde, a probabilidade) do exemplo de entrada pertencer a uma classe específica.
  - **Exemplo:** imaginemos um classificador de notícias com quatro classes possíveis: *esportes*, *política*, *ciências* e *variedades*. Como vocês as representariam com o ***one-hot encoding***?

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

# Fronteiras de decisão de um classificador

- Antes, usávamos funções para aproximar um modelo gerador, agora, as usaremos para separar classes.
- O espaço  $K$  dimensional (i.e.,  $\mathbb{R}^K$ ) criado pelos **atributos** é dividido em **regiões de decisão**,  $R_i, i = 1, \dots, Q$ , as quais são separadas pelas **fonteiras de decisão**.
- **Fonteiras de decisão** correspondem a **superfícies de decisão** no **espaço de atributos** onde ocorre uma indeterminação, ou seja, um empate entre diferentes classes possíveis.
- As **fonteiras de decisão** podem ser **lineares** (e.g., retas e planos) ou **não-lineares** (e.g., círculos).
- As **fonteiras 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.
- Figuras mostram **regiões de decisão** em problemas de classificação **binária** e **multi-classes**.



# Funções discriminates lineares

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

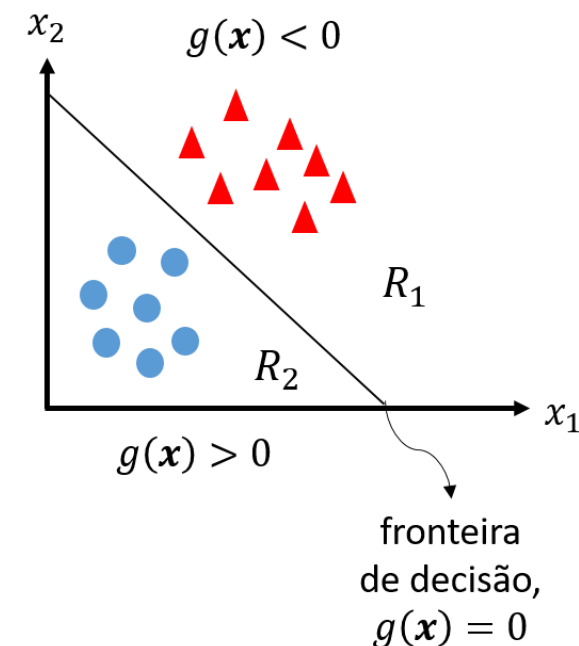
$$g(\mathbf{x}) = a_0 + a_1x_1 + a_2x_2 + \dots + a_Kx_K = \mathbf{a}^T \mathbf{x},$$

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

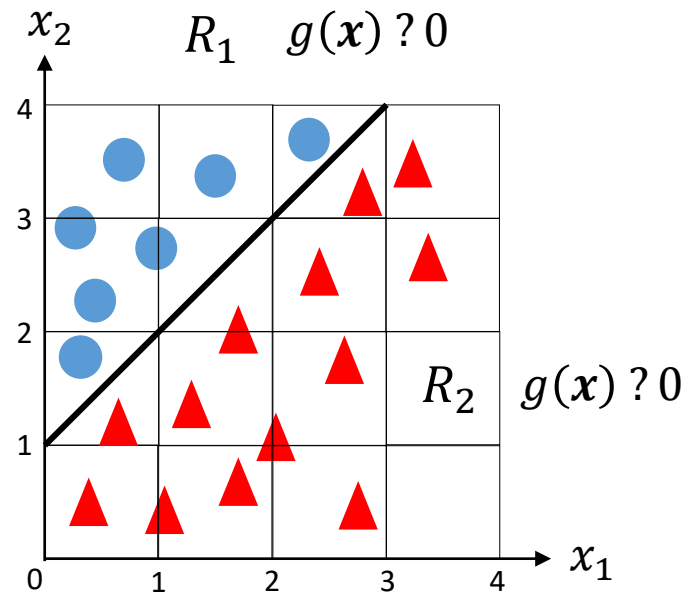
- $g(\mathbf{x})$  também pode ser visto como um **hiperplano** que separa as classes. Um **hiperplano** pode ser 1 ponto em 1D, uma reta em 2D e um plano em 3D.
  - O bias,  $a_0$ , dá o deslocamento com relação à origem.
  - E os restantes dos pesos determinam a orientação do **hiperplano**.
- A ideia aqui é encontrar os pesos da **função discriminante** de tal forma que

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

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



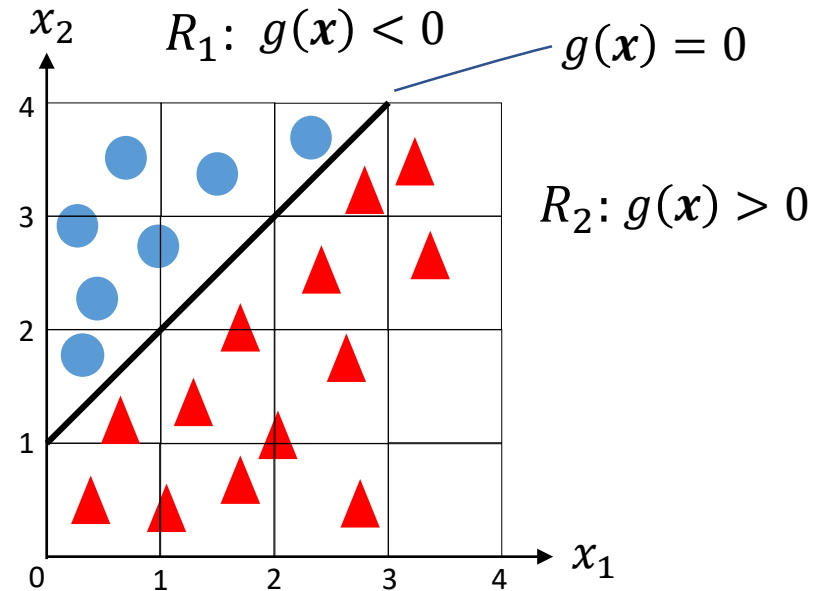
# Exemplo: Encontrar a função discriminante, $g(\mathbf{x})$



- Dada a seguinte função discriminante:  $g(\mathbf{x}) = a_0 + a_1x_1 + a_2x_2$
- Encontre os pesos e as regiões de decisão.



# Exemplo: Encontrar a função discriminante, $g(\mathbf{x})$



- 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
  - $g(\mathbf{x}) = 1 + x_1 - x_2$

# Tarefas

- **Quiz:** “*T320 - Quiz – xxxx*” 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!

