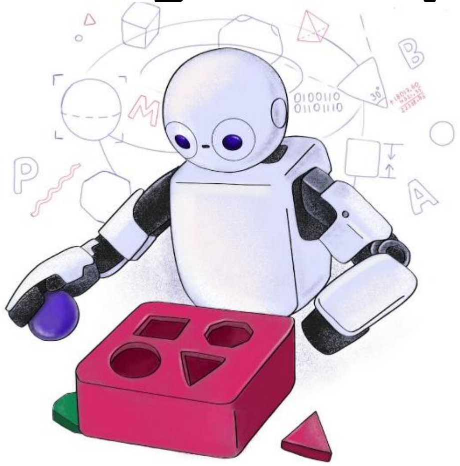
T320 - Introdução ao Aprendizado de Máquina II: *Classificação (Parte I)*



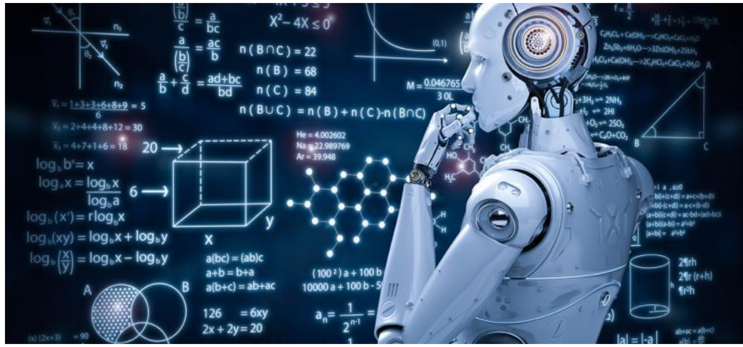
Inatel

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

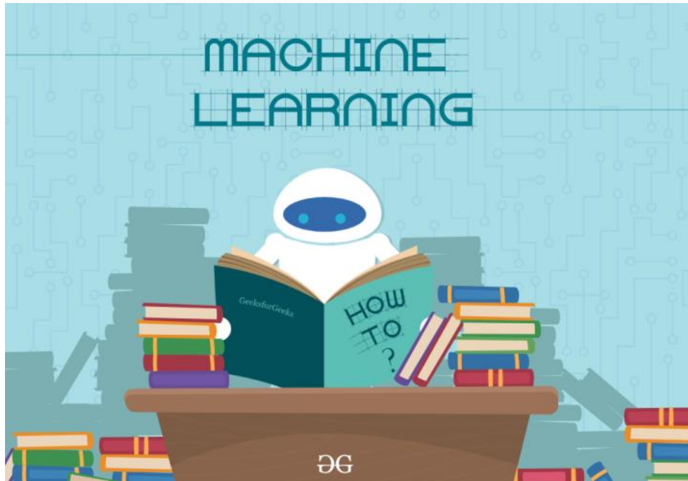
A disciplina

- 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 laboratórios envolvendo os conceitos discutidos.

Objetivo do curso



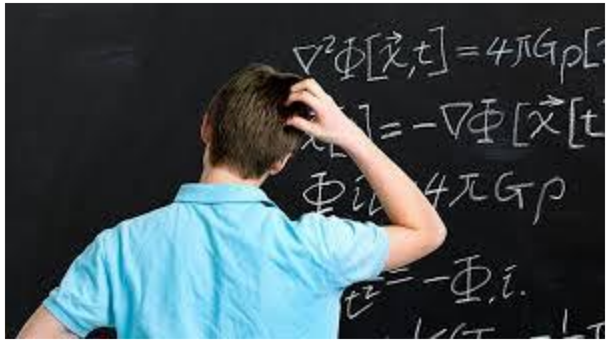
- O objetivo principal do curso é apresentar
  - os ***conceitos fundamentais*** da teoria do aprendizado de máquina.
  - um ***conjunto de ferramentas*** (i.e., algoritmos, métricas, técnicas) de aprendizado de máquina para solução de problemas.
- Ao final do curso vocês devem ser capazes de
  - Entender 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.
  - Envolvem 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 ser resolvidos em grupos de no máximo 3 alunos.
  - Exercícios serão atribuídos e entregues através do MS Teams.

- Extra: 10% da nota da FETIN na segunda nota.
  - O trabalho precisa usar IA.
- **Frequência**
  - Gerada automaticamente pelo Teams.
  - Por favor, acompanhem suas frequências no portal.



Cronograma

Aula	Data	Dia	Horário	Atividade
1	14/2/2026	Sábado	10:00 às 11:40	Introdução ao Aprendizado de Máquina
2	21/2/2026			Introdução ao Aprendizado de Máquina
3	28/2/2026			Introdução ao Aprendizado de Máquina
4	7/3/2026			Introdução ao Aprendizado de Máquina
5	14/3/2026			Introdução ao Aprendizado de Máquina
6	21/3/2026			Introdução ao Aprendizado de Máquina
7	28/3/2026			Introdução ao Aprendizado de Máquina
8	**4/4/2026***			Introdução ao Aprendizado de Máquina
9	11/4/2026			Introdução ao Aprendizado de Máquina
10	18/4/2026			Introdução ao Aprendizado de Máquina
11	25/4/2026			Introdução ao Aprendizado de Máquina
12	**2/5/2026**			**Avaliação Presencial I (Projeto I) (Sala I-??)**
13	9/5/2026			Introdução ao Aprendizado de Máquina
14	16/5/2026			Introdução ao Aprendizado de Máquina
15	**23/5/2026***			Introdução ao Aprendizado de Máquina
16	30/5/2026			Introdução ao Aprendizado de Máquina
17	**6/6/2026**			**Avaliação Presencial II (Projeto II) (Sala I-??)**
18	13/6/2026			Introdução ao Aprendizado de Máquina
19	20/6/2026			Introdução ao Aprendizado de Máquina
20	27/6/2026			Introdução ao Aprendizado de Máquina

***Feriados (as reposições serão assíncronas)**

Referências

[1] Stuart Russell e 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] Levy Boccato, “Notas de aula do curso Tópicos em Sistemas Inteligentes II - Aprendizado de Máquina” (IA006), disponíveis em https://www.dca.fee.unicamp.br/~lboccato/ia006_2s2019.html (2019).

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

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

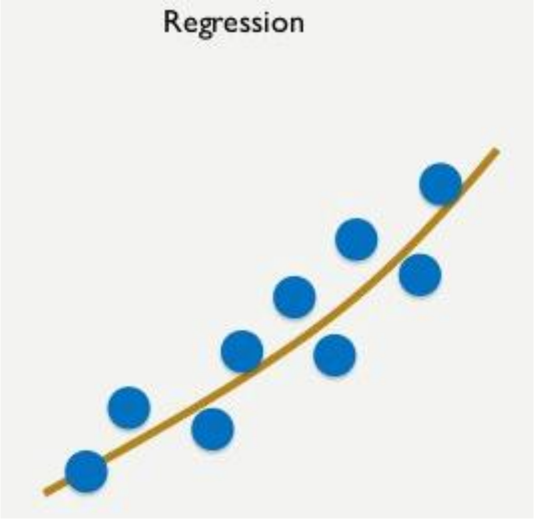
[6] Coleção de livros, https://tinyurl.com/mp64ksye

Avisos

- Toda nossa comunicação será feita via Teams.
- Todas as aulas serão gravadas e ficarão disponíveis na pasta “Recordings”.
- Todo material do curso está disponível no GitHub:
  - https://github.com/zz4fap/t320_aprendizado_de_maquina
- Entregas de aboratórios 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".
- Horários de atendimento
  - Professor: sextas-feiras das 18:30 às 19:30.
  - Monitor (Marcus Wilians Gomes Chagas: marcuswilians@gea.inatel.br): todas as terças-feiras das 17:30 às 19:30.
  - **Atendimento remoto via Teams.**

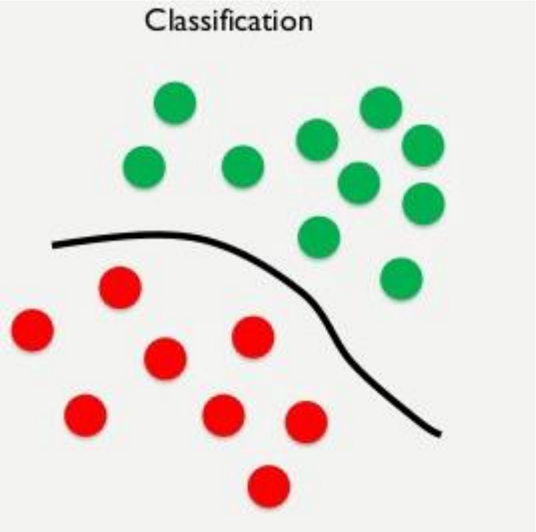
Classificação

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



$f(\boldsymbol{x})$ captura o ***comportamento geral*** dos dados.

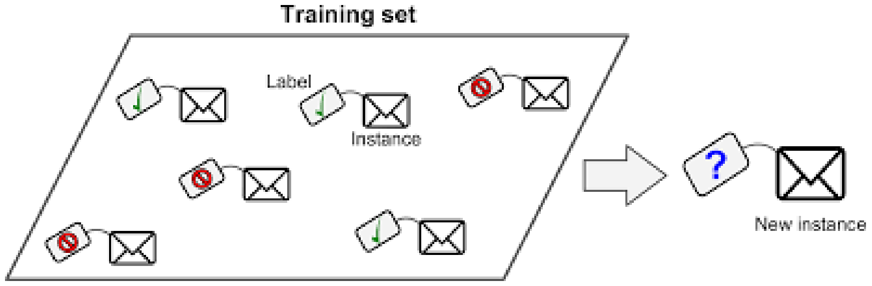
$f(\boldsymbol{x})$ ***aproxima*** o comportamento dos dados.

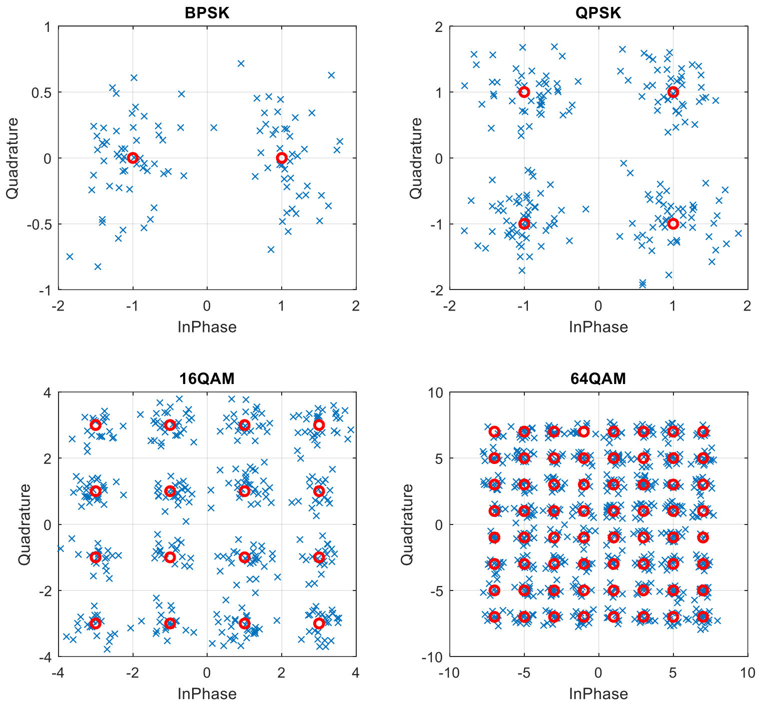


$f(\boldsymbol{x})$ forma uma ***fronteira de decisão***.

$f(\boldsymbol{x})$ ***classifica*** os dados.

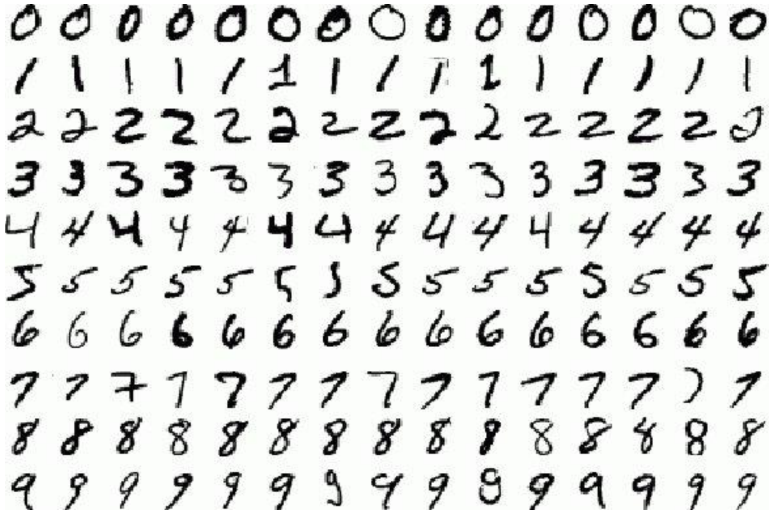
Tarefas de classificação

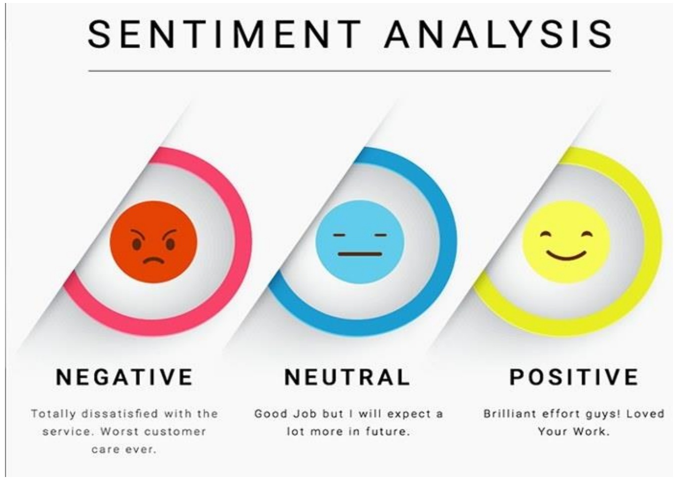




- Classificação de emails entre *spam* e *ham* (legítimo).
- Classificação de objetos em imagens ou vídeos.
- Detecção ou classificação de símbolos de modulações digitais.
- Classificação de modulações (QPSK, AM, FM, etc.).

Tarefas de classificação





- Reconhecimento de texto.
- Classificação de texto (e.g., notícias).
- Classificação de sentimentos.
- Classificação do doenças (e.g., pulmonares).

Definição do problema de classificação

- **Problema**: encontrar uma função, $f(\boldsymbol{x})$, que atribua a um ***exemplo de entrada***, $\boldsymbol{x}$, 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*** $\{\boldsymbol{x}(i); y(i)\}_{i=0}^{N-1}$ 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 de atributos, o qual é composto por K atributos, $x_1(i), \dots, x_K(i)$;
  - e $y(i)$ representa o *i*-ésimo ***rótulo***.

Como representar a saída desejada?

- A ***saída desejada*** (i.e., ***rótulo***) de um classificador para um ***vetor de atributos***, $\boldsymbol{x}(i)$, deve ser um ***valor que identifique*** a qual ***classe*** o vetor $\boldsymbol{x}(i)$ pertence.
- Sendo assim, a ***saída*** de um ***classificador*** é uma ***variável categórica*** (i.e., ***valor discreto pertencente a um conjunto finito de valores***).

Como representar a saída desejada?

- Portanto, para realizarmos o treinamento do ***modelo de classificação***, nós 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*** ($Q = 2$) ou ***multi-classes*** ($Q > 2$).

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 de ***classe positiva***.
- Portanto, nesse caso, o classificador possui ***uma única saída escalar binária*** para indicar a ***classe*** correspondente ao ***vetor de atributos***:

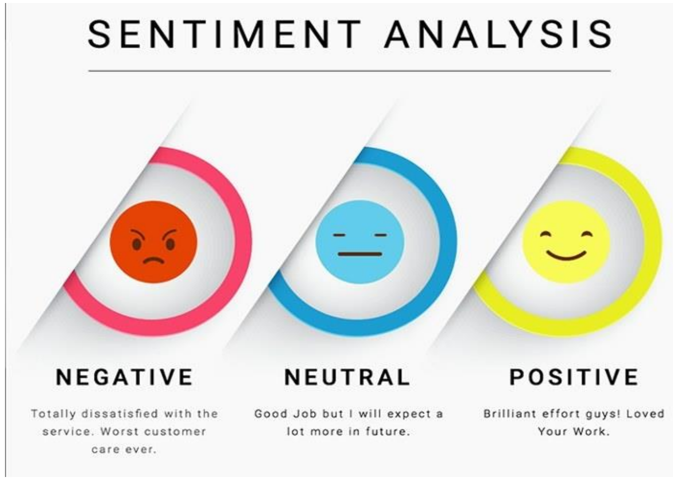
$$y(i) = \begin{cases} 0, & \boldsymbol{x}(i) \in C_1 \\ 1, & \boldsymbol{x}(i) \in C_2 \end{cases}.$$

- Assim, $y(i) \in \mathbb{R}^1$ (***escalar***), de maneira que ***o classificador realiza um mapeamento*** $\mathbb{R}^{K \times 1} \rightarrow \mathbb{R}^1$, i.e., $y(i) = f(\boldsymbol{x}(i))$, onde $\boldsymbol{x}(i) \in \mathbb{R}^{K \times 1}$.
- 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 ($Q > 2$).
- Uma estratégia bastante utilizada para representar estas classes é conhecida como ***codificação one-hot***.



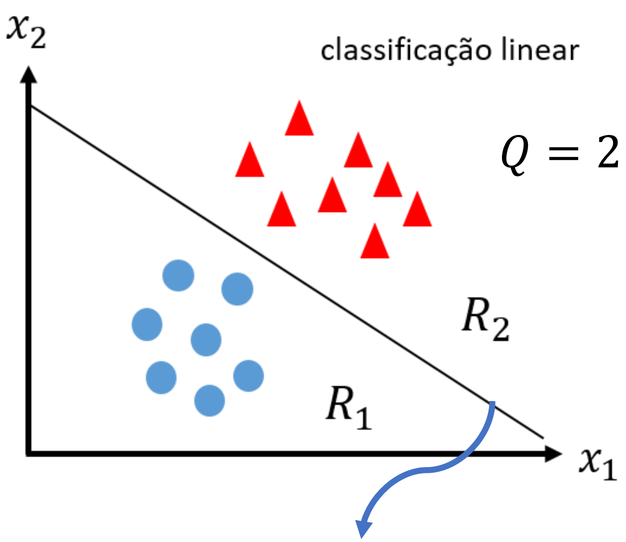
Representação da saída desejada

- ***Codificação one-hot***: saídas são representadas por ***vetores binários.***
  - As saídas são vetores com comprimento Q com o valor 1 no elemento que representa 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***?

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

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

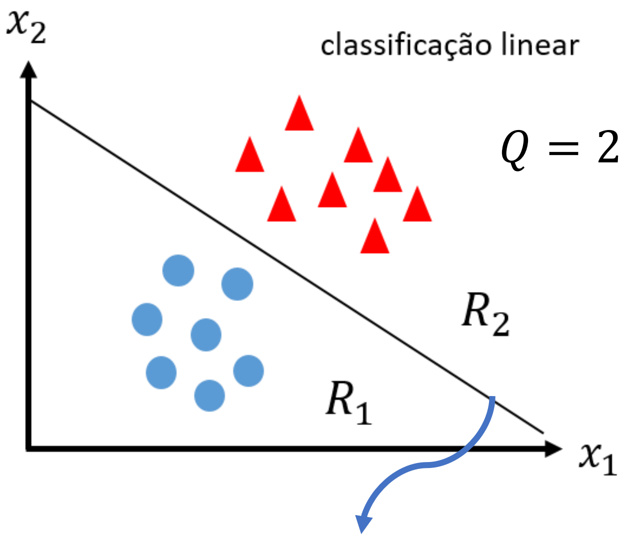
Fronteiras de decisão de um classificador



Fronteira de decisão: no caso deste exemplo, ela é uma reta.

- 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 pertencem à classe C_1.
  - Triângulos vermelhos pertencem à classe C_2.

Fronteiras de decisão de um classificador

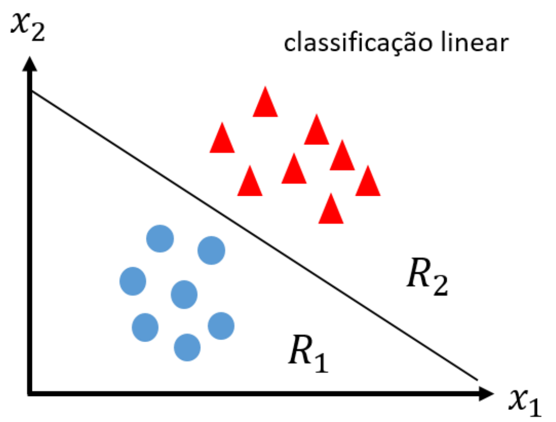


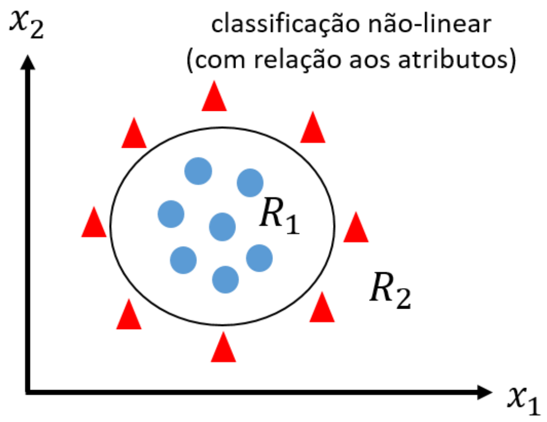
Fronteira de decisão:
no caso deste exemplo,
ela é uma reta.

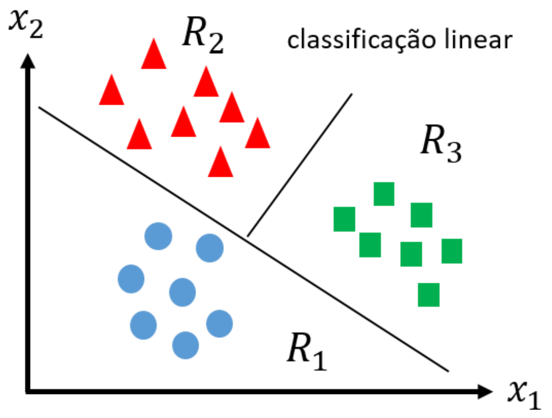
- Esse espaço pode ser dividido em ***duas regiões de decisão***, R_1 e R_2, onde ***cada região*** corresponde a ***uma classe***.
- As regiões de decisão são separadas por ***fronteiras de decisão***, que nada mais são do que ***funções***.
- Na figura, como $Q = 2$, temos apenas uma fronteira de decisão.
- Uma ***fronteira de decisão*** corresponde a uma ***superfície de separação*** (1D, 2D, 3D, etc.) no ***espaço de atributos*** que separa as classes.

Fronteiras de decisão de um classificador

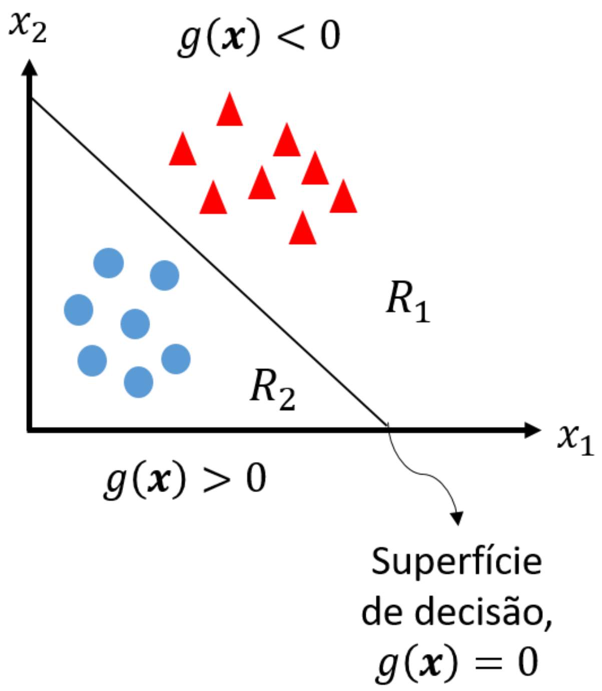
- As ***superfícies de separação*** podem ser ***lineares*** (e.g., retas e planos) ou ***não-lineares*** (e.g., círculos e elipses).
- ***Superfícies de separação*** são ***funções*** (lineares ou não) que separam as classes.
- Essas funções são chamadas de ***funções discriminantes***, pois separam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





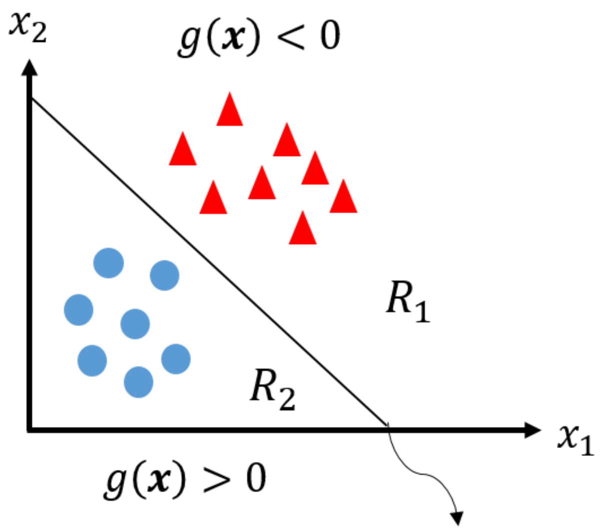


Funções discriminantes



- Uma ***função discriminante*** com formato de ***hiperplano*** é dada por
$$g(\boldsymbol{x}) = a_0 + a_1 x_1 + a_2 x_2 + \cdots + a_K x_K = \boldsymbol{a}^T \boldsymbol{x}.$$
- 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.
  - O restante dos pesos determina a orientação do ***hiperplano***.
- Lembrem-se que ***hiperplanos*** fazem ***mapeamentos lineares*** dos atributos nas saídas.

Funções discriminantes



- Nosso ***objetivo é encontrar os pesos da função discriminante*** de tal forma que que a classe atribuída a um exemplo de entrada, $\boldsymbol{x}$, seja:

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

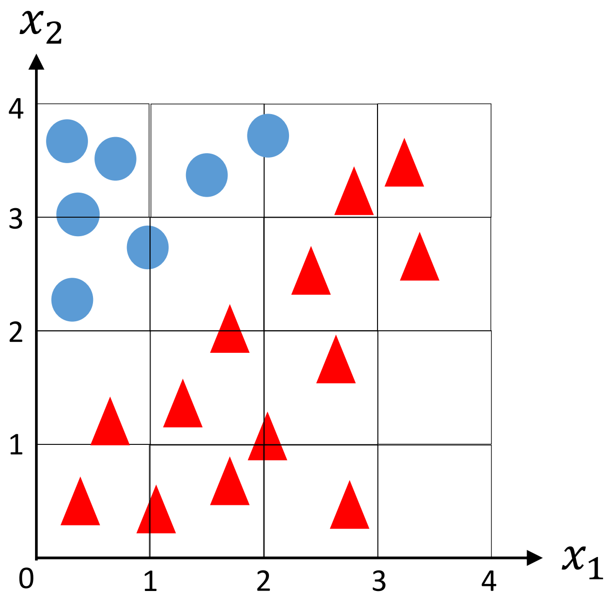
Acima da função.

Abaixo da função.

empate entre as classes, pois a amostra está sobre a função.

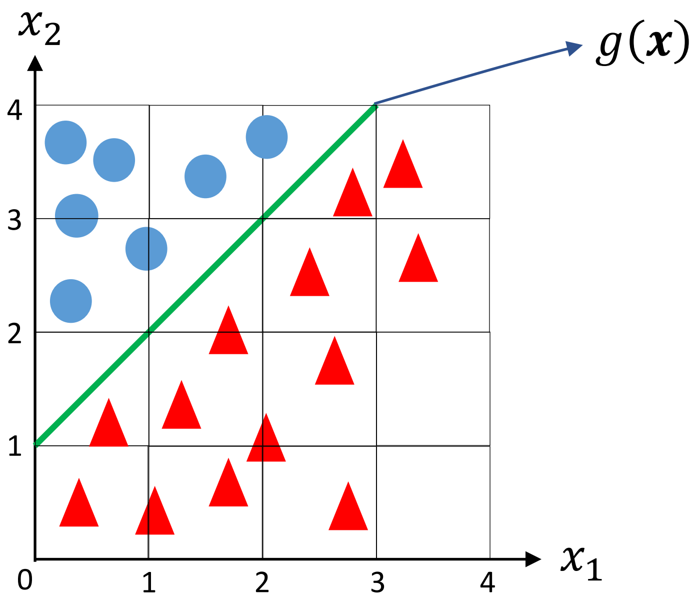
- OBS.: Podemos usar também ***funções discriminantes com mapeamentos não-lineares.***
  - Exemplo: $g(\boldsymbol{x}) = a_0 + x_1^2 + x_2^2$ (eq. de um círculo centrado na origem, onde $a_0 = -r^2$).

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



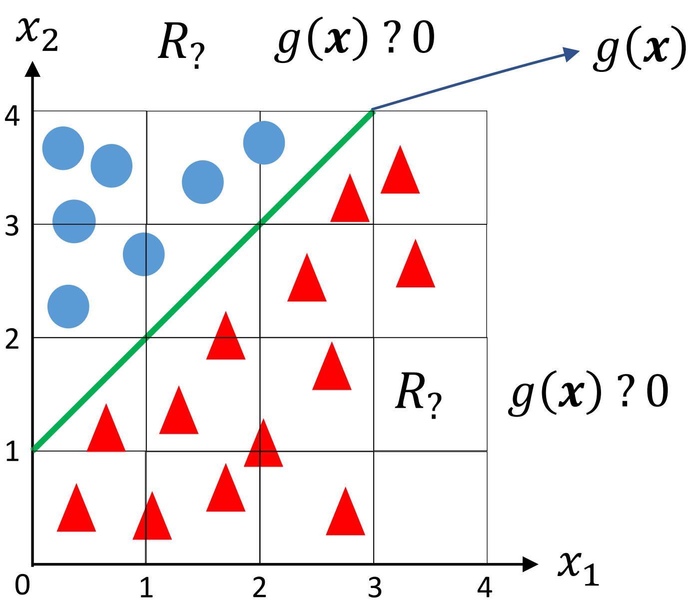
- Analisem a figura ao lado.
- Temos 2 classes, 2 atributos, x_1 e x_2, e queremos encontrar uma ***função discriminante***, $g(\boldsymbol{x})$, que as separe.
- Qual formato deve ter esta ***função discriminante*** para que ela tenha boa capacidade de generalização?
  - Lembrem-se do princípio da navalha 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, $g(\boldsymbol{x})$



- 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*** traçada no plano formado por x_1 e x_2.

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

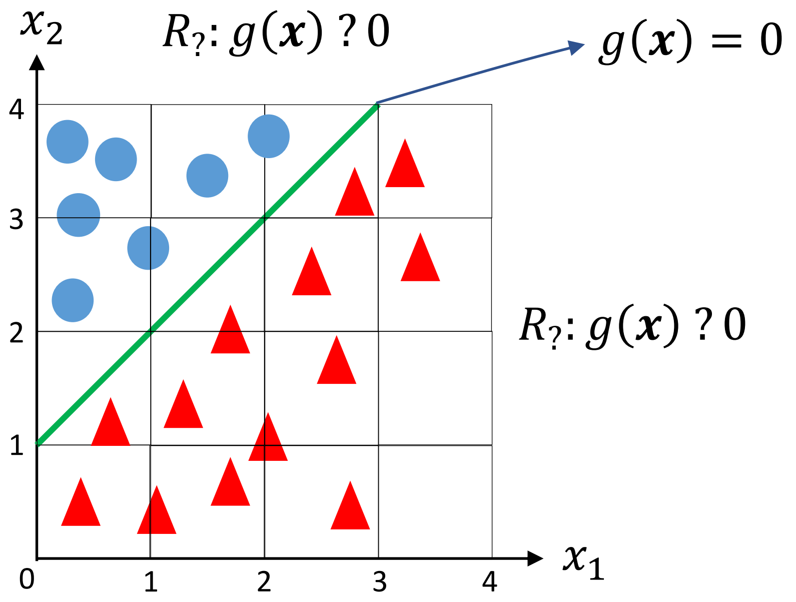


- *Visualmente*, nós traçamos a reta em uma posição que separe as classes da melhor forma possível.
- A ***função discriminante*** que representa esta reta é definida como

$$g(\boldsymbol{x}) = a_0 + a_1x_1 + a_2x_2 \quad \text{(plano)}$$

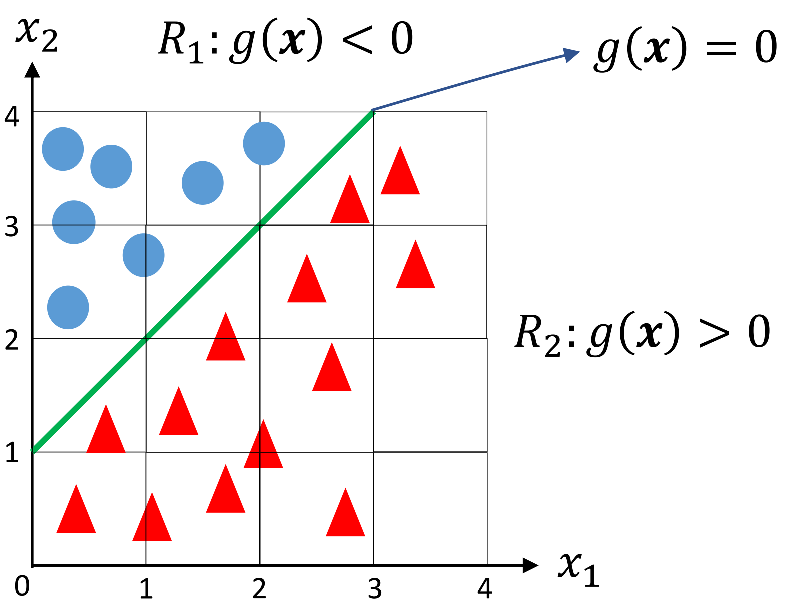
- Agora que definimos o formato da função e sua posição no gráfico, precisamos ***encontrar os pesos*** e, com isso, ***definir as regiões de decisão***.
- Como podemos encontrar os pesos?

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



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

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



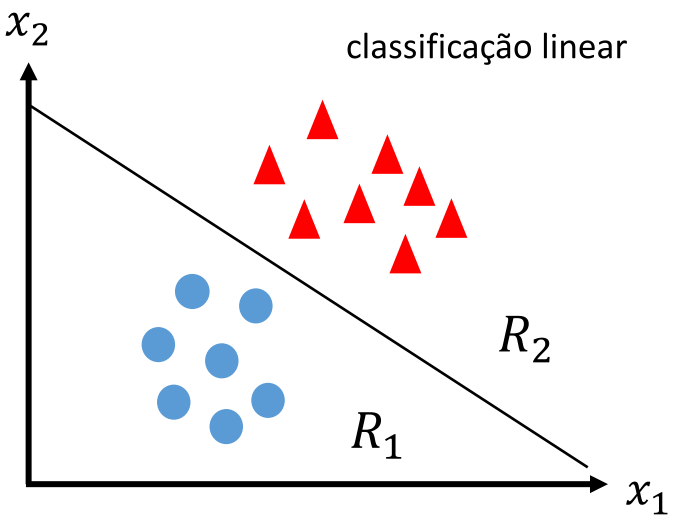
- Vamos definir as ***regiões de decisão*** substituindo alguns valores em $g(\boldsymbol{x}) = 1 + x_1 - x_2$.
  - $x_1 = 1$ e $x_2 = 1$ resulta em $g(\boldsymbol{x}) > 0$.
    - ✓ Região da classe ***positiva***, C_2.
  - $x_1 = 1$ e $x_2 = 3$ resulta em $g(\boldsymbol{x}) < 0$.
    - ✓ Região da classe ***negativa***, C_1.
  - $x_1 = 1$ e $x_2 = 2$ resulta em $g(\boldsymbol{x}) = 0$.
    - ✓ ***Indeterminação***: não podemos afirmar a qual classe o exemplo pertence.
    - ✓ Podemos atribuir arbitrariamente a uma das duas classes, escolher a classe que possui maior número de exemplos ou a mais frequente.
- O classificador pode ser implementado como uma ***estrutura de controle de fluxo***.

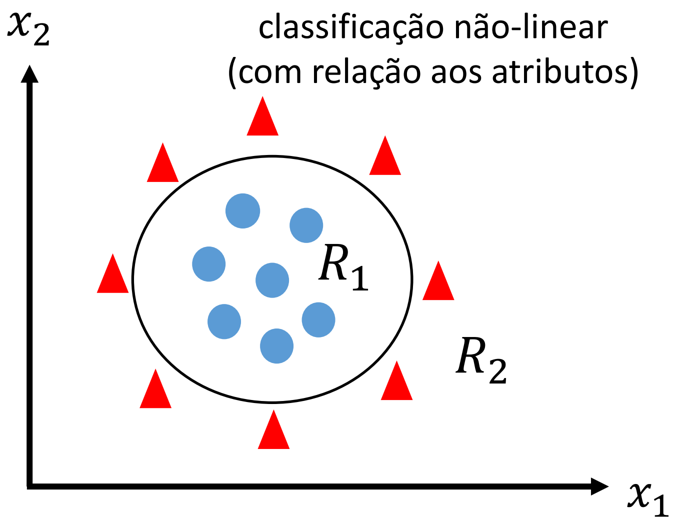
Exemplo: encontrando_pesos_da_função_discriminante.ipynb

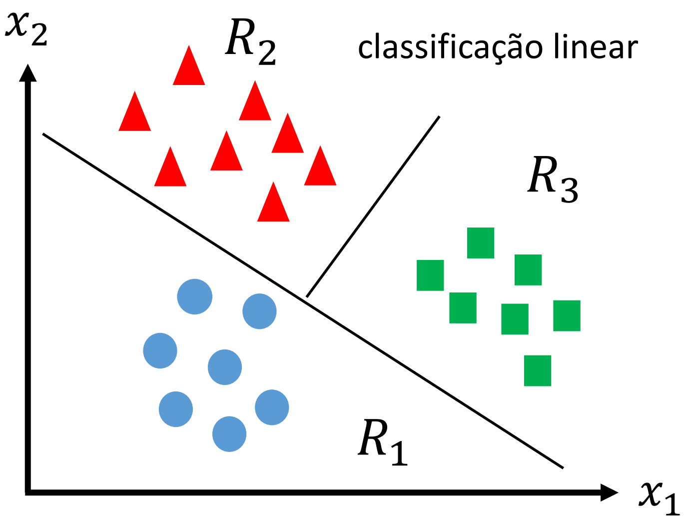
Tarefas

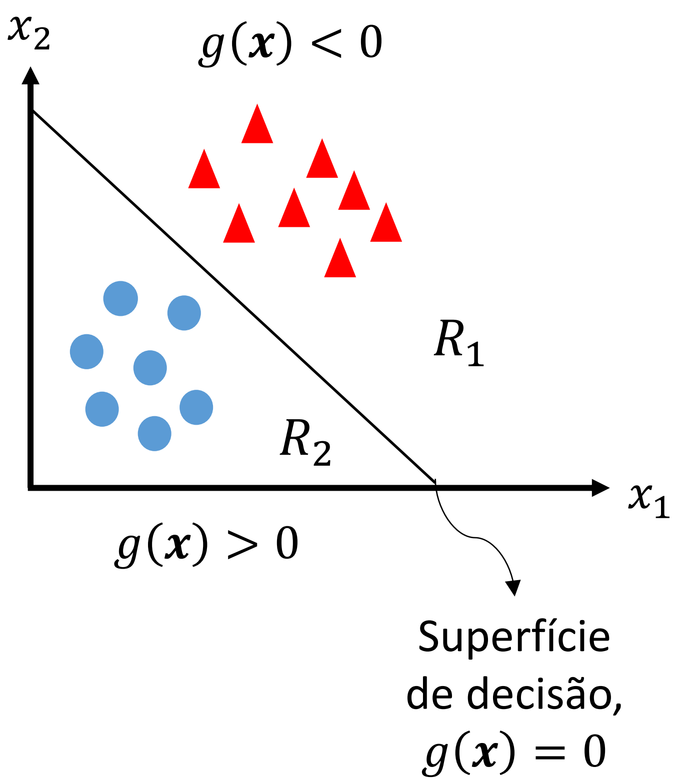
- **Quiz**: *"T320 - Quiz - Classificação (Parte I)"* que se encontra no MS Teams.
- **Exercício Prático**: **Laboratório #1**.
  - Pode ser acessado através do link acima (Google Colab) ou no GitHub.
  - Se atentem aos prazos de entrega.
  - Instruções para resolução e entrega dos laboratórios.

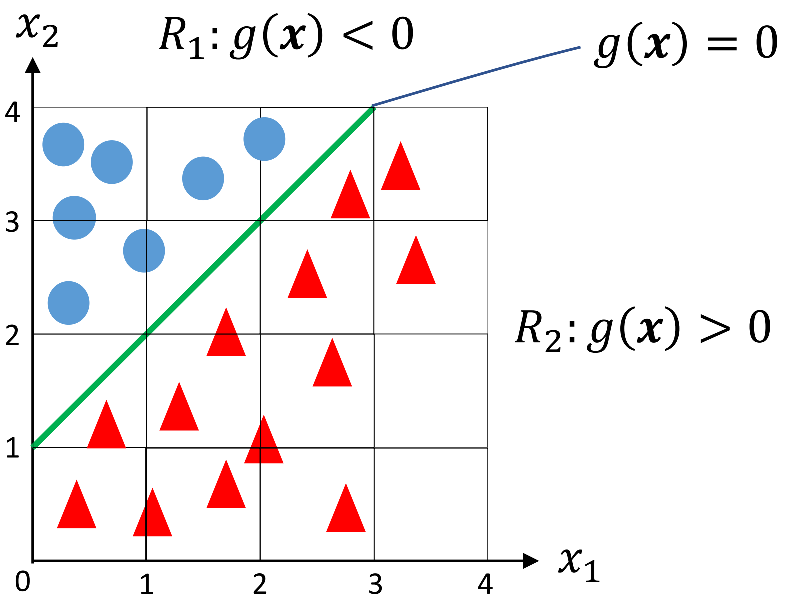
Obrigado!


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