

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
Instituto de Informática  
Departamento de Informática Aplicada

## Aula 6: Aprendizado supervisionado para textos

Prof. Dennis Giovani Balreira



INF01221 - Tópicos Especiais em Computação XXXVI:  
Processamento de Linguagem Natural



# Conteúdo

- Aprendizado supervisionado para textos:
  - Classificação de texto com algoritmos clássicos
    - Visão Geral
    - Naive Bayes
  - Avaliação de desempenho para texto:
    - Divisão dos dados
    - Generalização
    - Matriz de confusão
    - Acurácia
    - Precisão (precision)
    - Revocação (recall)
    - F-score (micro/macro)
    - *Ensemble learning*
    - Hiperparâmetros

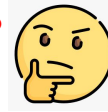
# Onde estamos em PLN?

- Algoritmos tradicionais
  - Predominantes entre o final dos anos 1990 até ~2016
  - BoW features + Aprendizado de Máquina
- Embeddings fixas + Deep Learning
  - Predominates de ~2014 até ~2019
  - Word2vec, Glove, FastText + LSTM
- Embeddings contextuais + Large Language Models
  - Estado da arte em diversas tarefas
  - BERT, GPT, etc.

Aprendizado supervisionado para textos

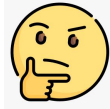
# Aprendizado supervisionado

- O que é “Aprendizado de Máquina Supervisionado”?



# Aprendizado supervisionado

- O que é “Aprendizado de Máquina Supervisionado”?
  - É uma área da Inteligência Artificial (IA)
  - Mas antes disso... o que é IA?



# Inteligência Artificial

- Mas antes disso... o que é IA?
  - “Inteligência Artificial é uma disciplina científica e de engenharia cujo objetivo é criar máquinas **inteligentes**”

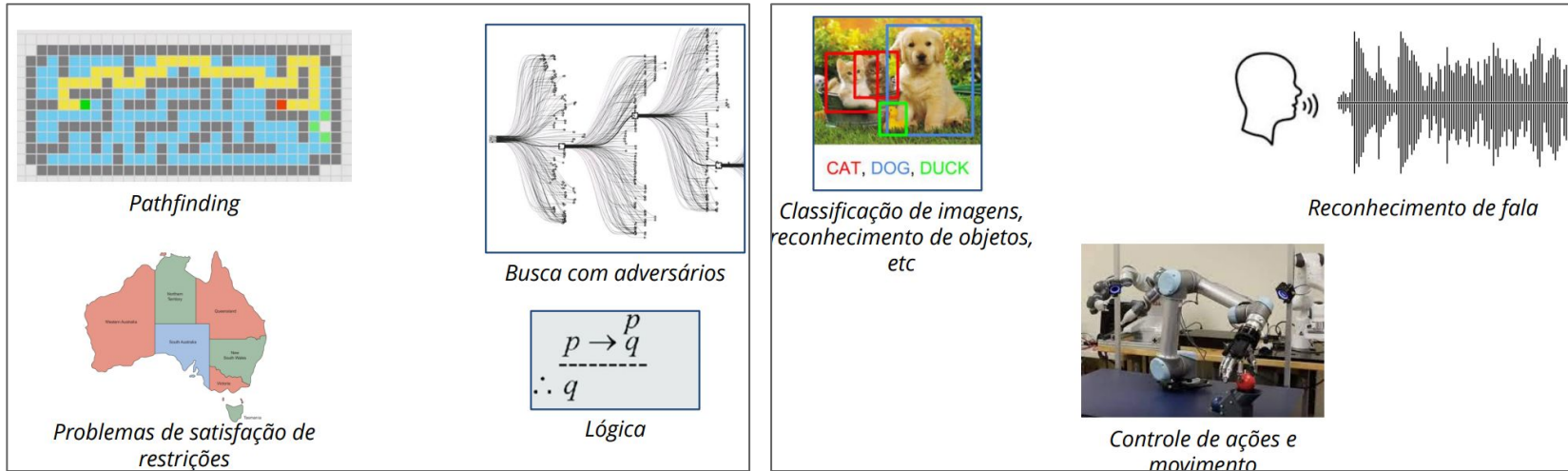
OU

- “O esforço para automatizar tarefas (intelectuais) normalmente realizadas por **humanos**”



# Inteligência Artificial

- IA é dividida em IA clássica (simbólica) e IA moderna

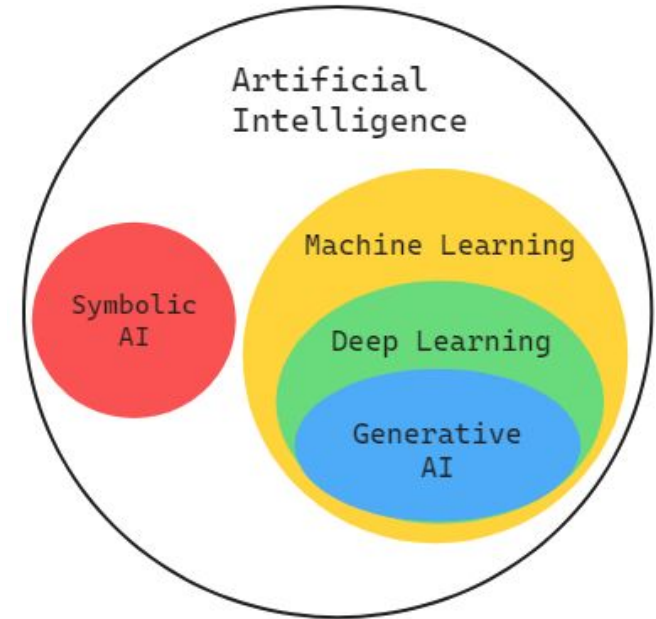




# Aprendizado de máquina

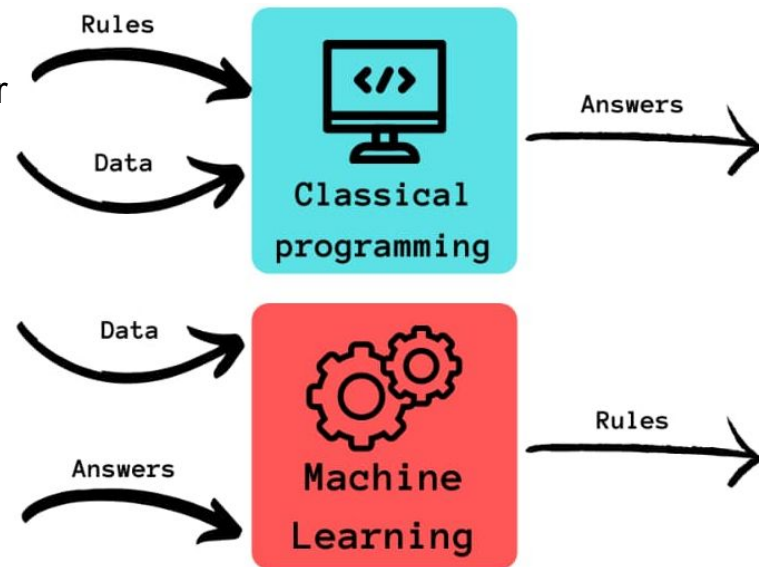
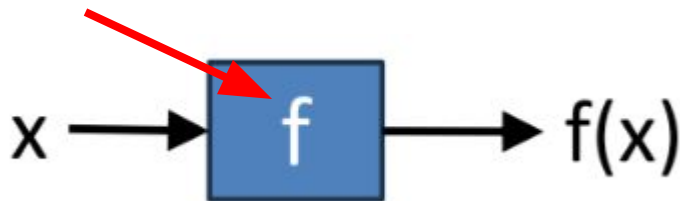
- Aprendizado de máquina

Sistemas que são treinados para realizar uma tarefa, ao invés de serem explicitamente programados para tal



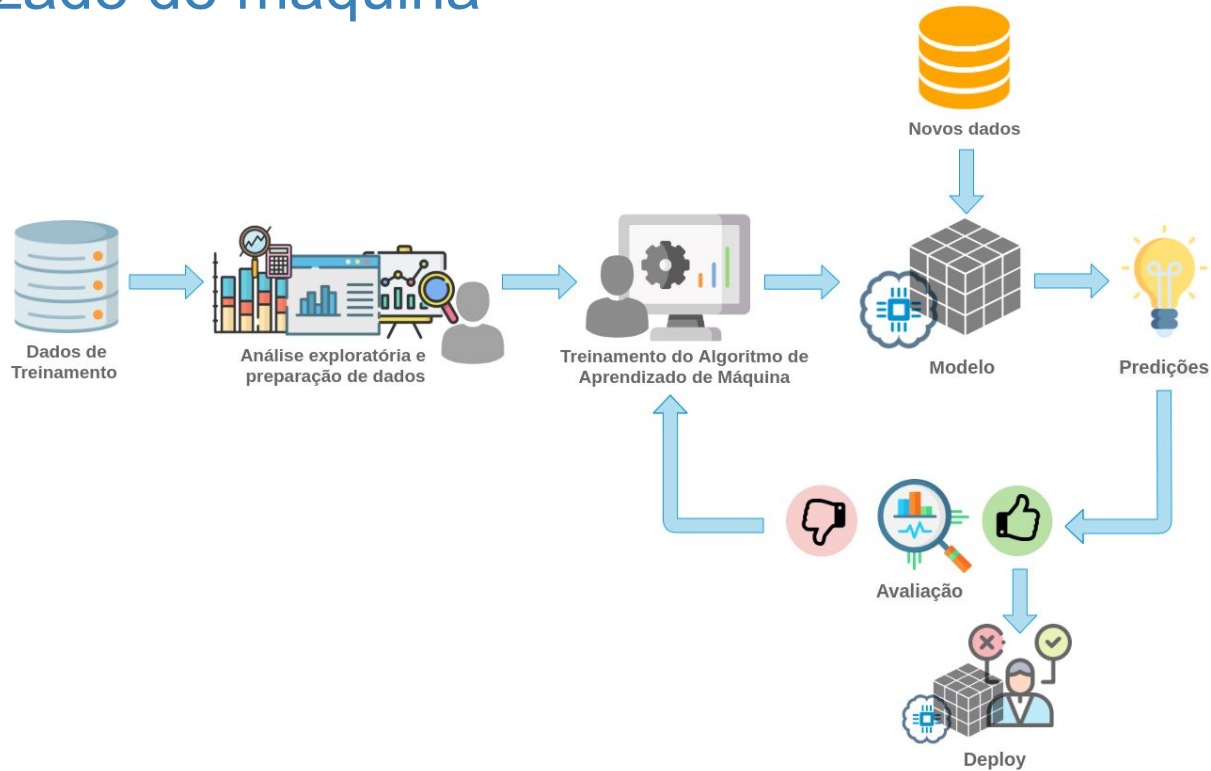
# Aprendizado de máquina

- Queremos encontrar **automaticamente** a “melhor função” **f** a partir dos dados
- Espera-se que **f** consiga prever um novo dado
- Com programação, o programador define como fazer
- Em aprendizado de máquina a solução vem “automaticamente” de **f**



# Aprendizado de máquina

- Pipeline



# Aprendizado de máquina

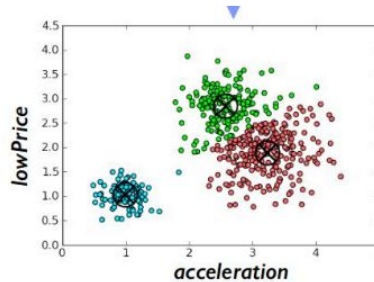
- Tipos

Aprendizado supervisionado

{email\_1, SPAM}  
{email\_2, NOT\_SPAM}  
{email\_3, NOT\_SPAM}

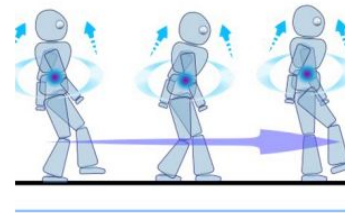
Requer dados

Aprendizado não supervisionado



Requer dados

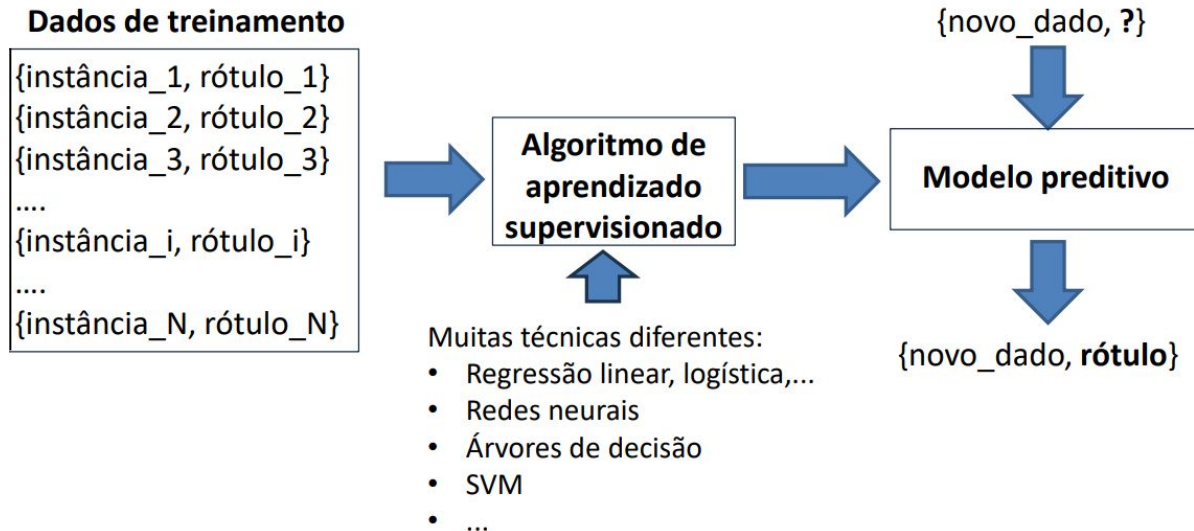
Aprendizado por reforço



Pode gerar os próprios dados

# Aprendizado de máquina supervisionado

- Visa aprender uma **função** que **mapeia dados de entrada a respostas/decisões de saída** a partir de um conjunto de **dados rotulados**



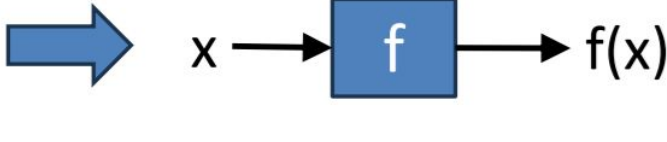
# Aprendizado de máquina supervisionado

- Precisamos de **dados estruturados** (ou semi-estruturados)
  - Busca aprender uma **função  $f$**  capaz de prever o valor do **atributo alvo** de uma instância  $x$  a partir dos atributos preditivos de  $x$

Atributos preditores			Atributo alvo (a ser predito)
Radius	Texture	Perimeter	Diagnosis
14	23	94	M
15	28	97	M
15	20	95	M
11	19	72	B
9	17	59	B
13	16	81	B
...	...	...	...

Atributos

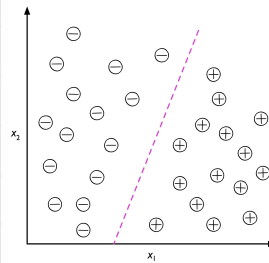
Instâncias / amostras



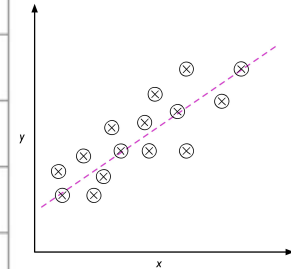
# Aprendizado de máquina supervisionado

- Tarefa varia conforme o tipo de dado a ser encontrado (atributo alvo):
  - **Classificação:** atributo alvo é um valor discreto (categórico)
  - **Regressão:** atributo alvo é um valor numérico real

Radius	Texture	Perimeter	Diagnosis
14	23	94	M
15	28	97	M
15	20	95	M
11	19	72	B
9	17	59	B
13	16	81	B
...	...	...	...

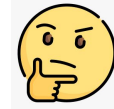


x	y
1	5
2	8
8	26
10	32



# Aprendizado de máquina supervisionado

- Exemplo:
  - Qual a função **f** mapeia valores de **x** no valor de **y** adequado?



x	y
1	5
2	8
8	26
10	32



# Aprendizado de máquina supervisionado

- Exemplo:
  - Qual a função  $f$  mapeia valores de  $x$  no valor de  $y$  adequado?

$x$	$y$
1	5
2	8
8	26
10	32

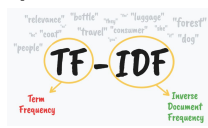
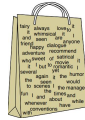


$$f(x) = 3x + 2$$

Precisamos aprender automaticamente (a partir do conjunto de dados) a função adequada

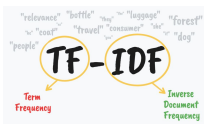
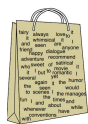
# Aprendizado supervisionado para textos

- E para textos?
  - Textos podem ser descritos por **vetores numéricos**!
    - Bag of Words
    - TF-IDF



# Aprendizado supervisionado para textos

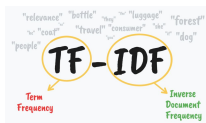
- E para textos?
  - Textos podem ser descritos por **vetores numéricos**!
    - Bag of Words
    - TF-IDF
- Vamos supor que vamos trabalhar com **análise de sentimentos**
  - Analisar textos para compreender as opiniões das pessoas
    - Reviews de produtos
    - Tweets
    - Posts



Muito bom!

# Aprendizado supervisionado para textos

- E para textos?
  - Textos podem ser descritos por **vetores numéricos!**
    - Bag of Words
    - TF-IDF
- Vamos supor que vamos trabalhar com **análise de sentimentos**
  - Analisar textos para compreender as opiniões das pessoas
    - Reviews de produtos
    - Tweets
    - Posts
- Tarefas comuns:
  - Polaridade: positivo, negativo etc.
  - Emoção: alegria, tristeza, raiva, medo, nojo etc.



Muito bom!



# Aprendizado supervisionado para textos

- Quais abordagens para resolver problemas de Análise de Sentimentos?
  - Usar um léxico pré-construído composto por palavras e suas polaridades
  - Aprender um modelo de classificação a partir de exemplos rotulados

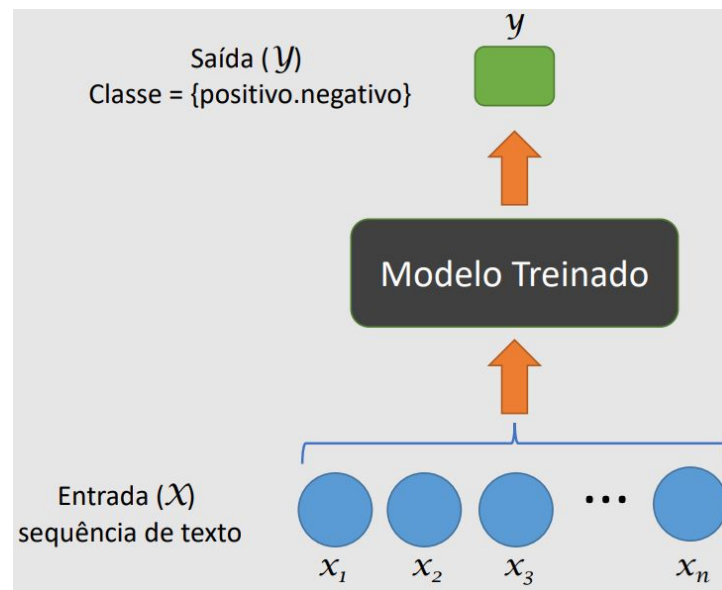
Categorical		Numerical	
word	sentiment	word	sentiment
nice	pos	nice	2
beautiful	pos	beautiful	3
amazing	pos	amazing	4
ugly	neg	ugly	-3
stupid	neg	stupid	-2

<https://mboyanov.medium.com/embeddings-transformations-for-sentiment-lexicon-enrichment-768c5eb06e55>

# Aprendizado supervisionado para textos

- Quais abordagens para resolver problemas de Análise de Sentimentos?
  1. Usar um léxico pré-construído composto por palavras e suas polaridades
  2. Aprender um modelo de classificação a partir de exemplos rotulados

Resultados mais interessantes!  
Foco desta e das próximas aulas!



# Aprendizado supervisionado para textos

- Exemplo de dataset de análise de sentimentos (polaridade - positivo e negativo)

1. "O notebook é muito rápido." positivo
2. "Excelente duração da bateria." positivo
3. "Carregamento do Windows muito rápido." positivo
4. "Muito ruim, vou devolver." negativo
5. "O notebook é bem fininho." positivo
6. "O teclado numérico é muito pequeno." negativo

# Aprendizado supervisionado para textos

- Exemplo de dataset de análise de sentimentos (polaridade - positivo e negativo)

1. "O notebook é muito rápido." positivo
2. "Excelente duração da bateria." positivo
3. "Carregamento do Windows muito rápido." positivo
4. "Muito ruim, vou devolver." negativo
5. "O notebook é bem fininho." positivo
6. "O teclado numérico é muito pequeno." negativo

Tokenização +  
pontuação



1. "O notebook é muito rápido" positivo
2. "Excelente duração da bateria" positivo
3. "Carregamento do Windows muito rápido" positivo
4. "Muito ruim vou devolver" negativo
5. "O notebook é bem fininho" positivo
6. "O teclado numérico é muito pequeno" negativo



# Aprendizado supervisionado para textos

- Exemplo de dataset de análise de sentimentos (polaridade - positivo e negativo)

1. "O notebook é muito rápido" **positivo**
2. "Excelente duração da bateria" **positivo**
3. "Carregamento do Windows muito rápido" **positivo**
4. "Muito ruim vou devolver" **negativo**
5. "O notebook é bem fininho" **positivo**
6. "O teclado numérico é muito pequeno" **negativo**

Case folding

1. "o notebook é muito rápido" **positivo**
2. "excelente duração da bateria" **positivo**
3. "carregamento do windows muito rápido" **positivo**
4. "muito ruim vou devolver" **negativo**
5. "o notebook é bem fininho" **positivo**
6. "o teclado numérico é muito pequeno" **negativo**

# Aprendizado supervisionado para textos

- Exemplo de dataset de análise de sentimentos (polaridade - positivo e negativo)

1. "o notebook é muito rápido" **positivo**
2. "excelente duração da bateria" **positivo**
3. "carregamento do windows muito rápido" **positivo**
4. "muito ruim vou devolver" **negativo**
5. "o notebook é bem fininho" **positivo**
6. "o teclado numérico é muito pequeno" **negativo**

## Remoção de Stopwords

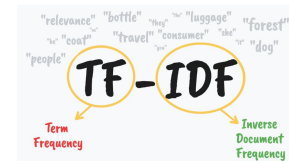
1. "o notebook é muito rápido" **positivo**
2. "excelente duração da bateria" **positivo**
3. "carregamento de windows muito rápido" **positivo**
4. "muito ruim vou devolver" **negativo**
5. "o notebook é bem fininho" **positivo**
6. "o teclado numérico é muito pequeno" **negativo**

# Aprendizado supervisionado para textos

- Exemplo de dataset de análise de sentimentos (polaridade - positivo e negativo)

1. “o notebook é muito rápido” **positivo**
2. “excelente duração da bateria” **positivo**
3. “carregamento de windows muito rápido” **positivo**
4. “muito ruim ~~vou~~ devolver” **negativo**
5. “o notebook é bem fininho” **positivo**
6. “o teclado numérico é muito pequeno” **negativo**

Representação  
com técnicas  
tradicionais



Vocabulário: [bateria, bem, carregamento, devolver, duração, excelente, fininho, muito, notebook, numérico, pequeno, ruim, rápido, teclado, windows]

Atributos

Geração dos vetores (um para cada documento) depende da técnica usada!

Instâncias

- Bag of Words
- Term Frequency - Inverse Document Frequency

# Aprendizado supervisionado para textos

- Exemplo de dataset de análise de sentimentos (polaridade - positivo e negativo)

1. “o notebook é muito rápido” **positivo**
2. “excelente duração da bateria” **positivo**
3. “carregamento de windows muito rápido” **positivo**
4. “muito ruim vou devolver” **negativo**
5. “o notebook é bem fininho” **positivo**
6. “o teclado numérico é muito pequeno” **negativo**

Representação  
com BoW



Vocabulário: [bateria, bem, carregamento, devolver, duração, excelente, fininho, muito, notebook, numérico, pequeno, ruim, rápido, teclado, windows]

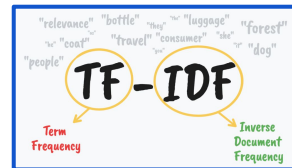
1. “O notebook é muito rápido” **positivo** = [0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, positivo]
2. “Excelente duração da bateria” **positivo** = [1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, positivo]
3. “Carregamento do Windows muito rápido” **positivo** = [0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, positivo]
4. “Muito ruim vou devolver” **negativo** = [0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, negativo]
5. “O notebook é bem fininho” **positivo** = [0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, positivo]
6. “O teclado numérico é muito pequeno” **negativo** = [0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, negativo]

# Aprendizado supervisionado para textos

- Exemplo de dataset de análise de sentimentos (polaridade - positivo e negativo)

1. “o notebook é muito rápido” **positivo**
2. “excelente duração da bateria” **positivo**
3. “carregamento de windows muito rápido” **positivo**
4. “muito ruim vou devolver” **negativo**
5. “o notebook é bem fininho” **positivo**
6. “o teclado numérico é muito pequeno” **negativo**

## Representação com TF-IDF



Vocabulário: [bateria, bem, carregamento, devolver, duração, excelente, fininho, muito, notebook, numérico, pequeno, ruim, rápido, teclado, windows]

1. “O notebook é muito rápido” **positivo** = [0, 0, 0, 0, 0, 0, 0.18, 0.48, 0, 0, 0, 0.48, 0, 0 **positivo**]
2. “Excelente duração da bateria” **positivo** = [0, 0, 0, 0.78, 0.78, 0, 0, 0, 0, 0, 0, 0, 0, 0, **positivo**]
3. “Carregamento do Windows muito rápido” **positivo** = [0, 0.78, 0, 0, 0, 0, 0.18, 0, 0, 0, 0, 0.48, 0, 0.78, **positivo**]
4. “Muito ruim vou devolver” **negativo** = [0, 0, 0.78, 0, 0, 0, 0.18, 0, 0, 0, 0.78, 0, 0, 0, **negativo**]
5. “O notebook é bem fininho” **positivo** = [0.78, 0, 0, 0, 0, 0, 0.78, 0, 0.48, 0, 0, 0, 0, 0, **positivo**]
6. “O teclado numérico é muito pequeno” **negativo** = [0, 0, 0, 0, 0, 0, 0.18, 0, 0.78, 0.78, 0, 0, 0.78, 0, **negativo**]

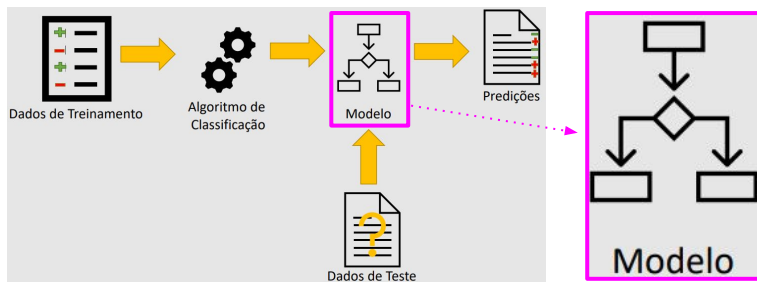
# Aprendizado supervisionado para textos

- Exemplo de dataset de análise de sentimentos (polaridade - positivo e negativo)

Vocabulário: [bateria, bem, carregamento, devolver, duração, excelente, fininho, muito, notebook, numérico, pequeno, ruim, rápido, teclado, windows]

1. "O notebook é muito rápido" **positivo** = [0, 0, 0, 0, 0, 0, 0.18, 0.48, 0, 0, 0, 0.48, 0, 0 positivo]
2. "Excelente duração da bateria" **positivo** = [0, 0, 0, 0.78, 0.78, 0, 0, 0, 0, 0, 0, 0, 0, 0, positivo]
3. "Carregamento do Windows muito rápido" **positivo** = [0, 0.78, 0, 0, 0, 0, 0.18, 0, 0, 0, 0, 0.48, 0, 0.78, positivo]
4. "Muito ruim vou devolver" **negativo** = [0, 0, 0.78, 0, 0, 0, 0.18, 0, 0, 0, 0.78, 0, 0, 0, negativo]
5. "O notebook é bem fininho" **positivo** = [0.78, 0, 0, 0, 0, 0.78, 0, 0.48, 0, 0, 0, 0, 0, 0, positivo]
6. "O teclado numérico é muito pequeno" **negativo** = [0, 0, 0, 0, 0, 0, 0.18, 0, 0.78, 0.78, 0, 0, 0.78, 0, negativo]

Dados para o processo de  
aprendizado de máquina

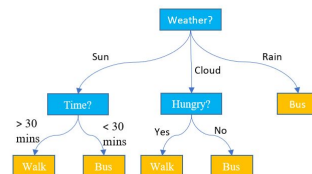
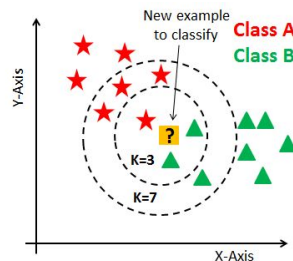




# Aprendizado supervisionado para textos

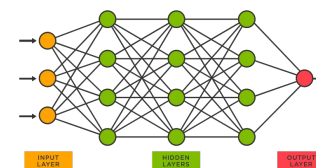
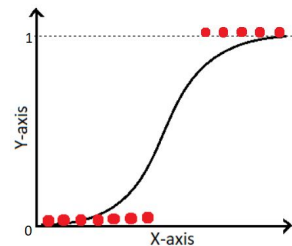
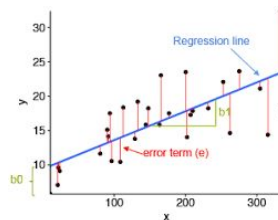
- Exemplos de algoritmos de classificação tradicionais:

- K-Nearest Neighbors
- Árvores de Decisão
- Naive Bayes
- Regressão Linear
- Regressão Logística
- Redes Neurais



$$P(A|B) = \frac{P(B|A) P(A)}{P(B)}$$

Annotations:  
-  $P(B|A)$ : THE PROBABILITY OF "B" BEING TRUE GIVEN THAT "A" IS TRUE  
-  $P(A)$ : THE PROBABILITY OF "A" BEING TRUE  
-  $P(B)$ : THE PROBABILITY OF "B" BEING TRUE  
-  $P(A|B)$ : THE PROBABILITY OF "A" BEING TRUE GIVEN THAT "B" IS TRUE



# Aprendizado supervisionado para textos

- Exemplos de algoritmos de classificação tradicionais:
  - K-Nearest Neighbors
  - Árvores de Decisão
  - Naive Bayes
  - Regressão Linear
  - Regressão Logística
  - Redes Neurais

A handwritten diagram illustrating Bayes' Theorem. The central equation is 
$$P(A|B) = \frac{P(B|A) P(A)}{P(B)}$$
. Annotations with arrows explain each term: 

- An arrow from  $P(A|B)$  points to the text "THE PROBABILITY OF 'A' BEING TRUE GIVEN THAT 'B' IS TRUE".
- An arrow from  $P(B|A)$  points to the text "THE PROBABILITY OF 'B' BEING TRUE GIVEN THAT 'A' IS TRUE".
- An arrow from  $P(A)$  points to the text "THE PROBABILITY OF 'A' BEING TRUE".
- An arrow from  $P(B)$  points to the text "THE PROBABILITY OF 'B' BEING TRUE".



# Naive Bayes (para texto)

- Algoritmo probabilístico baseado no Teorema de Bayes
  - Calcula a probabilidade posterior de um evento com base em evidências observadas
- O Naive Bayes usa o teorema para calcular a probabilidade de uma amostra pertencer a uma classe dado o vetor de frequências das palavras



Thomas Bayes (1701~1761)

Qual é a probabilidade da amostra pertencer à classe C (positivo ou negativo), dado o vetor X

Qual é a chance de ver as palavras nas frequências observadas, assumindo que a classe é C

$$P(C|X) = \frac{P(X|C) \cdot P(C)}{P(X)}$$

Qual é a probabilidade de a classe ser C antes de observar as características da amostra?

Qual é a chance de observar essas frequências das palavras, independentemente da classe?

# Naive Bayes (para texto)

- Algoritmo probabilístico baseado no Teorema de Bayes
  - Calcula a probabilidade posterior de um evento com base em evidências observadas
- O Naive Bayes usa o teorema para calcular a probabilidade de uma amostra pertencer a uma classe dado o vetor de frequências das palavras



Thomas Bayes (1701~1761)

Qual é a probabilidade da amostra pertencer à classe C (positivo ou negativo), dado o vetor X

Qual é a chance de ver as palavras nas frequências observadas, assumindo que a classe é C

$$P(C|X) = \frac{P(X|C) \cdot P(C)}{P(X)}$$

Qual é a probabilidade de a classe ser C antes de observar as características da amostra?

A ideia é apenas “comparar”,  
não calcular o valor exato!

Qual é a chance de observar essas frequências das palavras, independentemente da classe?

# Naive Bayes (para texto)

- Algoritmo probabilístico baseado no Teorema de Bayes
  - Calcula a probabilidade posterior de um evento com base em evidências observadas
- O Naive Bayes usa o teorema para calcular a probabilidade de uma amostra pertencer a uma classe dado o vetor de frequências das palavras



Thomas Bayes (1701~1761)

Qual é a probabilidade da amostra pertencer à classe C (positivo ou negativo), dado o vetor X

Qual é a chance de ver as palavras nas frequências observadas, assumindo que a classe é C

$$P(C|X) \propto P(X|C) \cdot P(C)$$

Qual é a probabilidade de a classe ser C antes de observar as características da amostra?

“Naive” (ingênuo) porque é assumido que as features são independentes entre si (cada palavra tem sua probabilidade calculada independentemente das demais)

$$P(X|C) = P(x_1|C) \cdot P(x_2|C) \cdot \dots \cdot P(x_n|C)$$

$$P(C|X) \propto P(X|C) \cdot P(C)$$

## Naive Bayes (para texto)

$$P(X|C) = P(x_1|C) \cdot P(x_2|C) \cdot \dots \cdot P(x_n|C)$$

- Para o nosso exemplo (assumindo BoW):

1. "O notebook é muito rápido" **positivo** = [0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, positivo]
2. "Excelente duração da bateria" **positivo** = [1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, positivo]
3. "Carregamento do Windows muito rápido" **positivo** = [0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, positivo]
4. "Muito ruim vou devolver" **negativo** = [0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, negativo]
5. "O notebook é bem fininho" **positivo** = [0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, positivo]
6. "O teclado numérico é muito pequeno" **negativo** = [0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, negativo]

Para ser usado como conjunto de teste (falaremos em breve)

$$P(C|X) \propto P(X|C) \cdot P(C)$$

## Naive Bayes (para texto)

$$P(X|C) = P(x_1|C) \cdot P(x_2|C) \cdot \dots \cdot P(x_n|C)$$

- Para o nosso exemplo (assumindo BoW):

$$P(X_i|C) = \frac{\text{soma das frequências da palavra} + 1}{\text{número total de amostras na classe} + \text{tamanho do vocabulário}}$$

1. "O notebook é muito rápido" **positivo** = [0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, positivo]
2. "Excelente duração da bateria" **positivo** = [1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, positivo]
3. "Carregamento do Windows muito rápido" **positivo** = [0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, positivo]
4. "Muito ruim vou devolver" **negativo** = [0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, negativo]
5. "O notebook é bem fininho" **positivo** = [0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, positivo]
6. "O teclado numérico é muito pequeno" **negativo** = [0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, negativo]

Suavização de Laplace  
(evita probabilidade 0)

Para ser usado como conjunto  
de teste (falaremos em breve)

$$P(C|X) \propto P(X|C) \cdot P(C)$$

## Naive Bayes (para texto)

$$P(X|C) = P(x_1|C) \cdot P(x_2|C) \cdot \dots \cdot P(x_n|C)$$

- Para o nosso exemplo (assumindo BoW):

$$P(X_i|C) = \frac{\text{soma das frequências da palavra} + 1}{\text{número total de amostras na classe} + \text{tamanho do vocabulário}}$$

1. "O notebook é muito rápido" **positivo** = [0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, positivo]
2. "Excelente duração da bateria" **positivo** = [1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, positivo]
3. "Carregamento do Windows muito rápido" **positivo** = [0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, positivo]
4. "Muito ruim vou devolver" **negativo** = [0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, negativo]
5. "O notebook é bem fininho" **positivo** = [0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, positivo]
6. "O teclado numérico é muito pequeno" **negativo** = [0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, negativo]

Suavização de Laplace  
(evita probabilidade 0)

Para ser usado como conjunto  
de teste (falaremos em breve)

$$P(C)$$

$$P(\text{positivo}) = \frac{4}{5} = 0.8$$

$$P(\text{negativo}) = \frac{1}{5} = 0.2$$

$$P(C|X) \propto P(X|C) \cdot P(C)$$

## Naive Bayes (para texto)

$$P(X|C) = P(x_1|C) \cdot P(x_2|C) \cdot \dots \cdot P(x_n|C)$$

- Para o nosso exemplo (assumindo BoW):

$$P(X_i|C) = \frac{\text{soma das frequências da palavra} + 1}{\text{número total de amostras na classe} + \text{tamanho do vocabulário}}$$

- “O notebook é muito rápido” **positivo** = [0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, **positivo**]
- “Excelente duração da bateria” **positivo** = [1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, **positivo**]
- “Carregamento do Windows muito rápido” **positivo** = [0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, **positivo**]
- “Muito ruim vou devolver” **negativo** = [0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, **negativo**]
- “O notebook é bem fininho” **positivo** = [0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, **positivo**]
- “O teclado numérico é muito pequeno” **negativo** = [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, **negativo**]

Suavização de Laplace  
(evita probabilidade 0)

Para ser usado como conjunto  
de teste (falaremos em breve)

$$P(C)$$

$$P(\text{positivo}) = \frac{4}{5} = 0.8$$

$$P(\text{negativo}) = \frac{1}{5} = 0.2$$

$$P(X_i|C)$$

$$P(\text{muito}|\text{positivo}) = \frac{2+1}{4+15} = \frac{3}{19} \approx 0.1579$$

$$P(\text{numérico}|\text{positivo}) = \frac{0+1}{19} = 0.0526$$

$$P(\text{pequeno}|\text{positivo}) = \frac{0+1}{19} = 0.0526$$

$$P(\text{teclado}|\text{positivo}) = \frac{0+1}{19} = 0.0526$$

$$P(\text{muito}|\text{negativo}) = \frac{1+1}{1+15} = \frac{2}{16} = 0.125$$

$$P(\text{numérico}|\text{negativo}) = \frac{0+1}{16} = 0.0625$$

$$P(\text{pequeno}|\text{negativo}) = \frac{0+1}{16} = 0.0625$$

$$P(\text{teclado}|\text{negativo}) = \frac{0+1}{16} = 0.0625$$



$$P(C|X) \propto P(X|C) \cdot P(C)$$

## Naive Bayes (para texto)

$$P(X|C) = P(x_1|C) \cdot P(x_2|C) \cdot \dots \cdot P(x_n|C)$$

- Para o nosso exemplo (assumindo BoW):

$$P(X_i|C) = \frac{\text{soma das frequências da palavra} + 1}{\text{número total de amostras na classe} + \text{tamanho do vocabulário}}$$

- “O notebook é muito rápido” **positivo** = [0, 0, 0, 0, 0, 0, 0, 1, 1, 0, 0, 0, 1, 0, 0, **positivo**]
- “Excelente duração da bateria” **positivo** = [1, 0, 0, 0, 1, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, **positivo**]
- “Carregamento do Windows muito rápido” **positivo** = [0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 1, **positivo**]
- “Muito ruim vou devolver” **negativo** = [0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, 1, 0, 0, 0, **negativo**]
- “O notebook é bem fininho” **positivo** = [0, 1, 0, 0, 0, 0, 1, 0, 1, 0, 0, 0, 0, 0, 0, **positivo**]
- “O teclado numérico é muito pequeno” **negativo** = [0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 1, 1, 0, 0, 1, 0, **negativo**]

Suavização de Laplace  
(evita probabilidade 0)

Para ser usado como conjunto  
de teste (falaremos em breve)

$$P(C)$$

$$P(\text{positivo}) = \frac{4}{5} = 0.8$$

$$P(\text{negativo}) = \frac{1}{5} = 0.2$$

$$P(X_i|C)$$

$$P(\text{muito}|\text{positivo}) = \frac{2+1}{4+15} = \frac{3}{19} \approx 0.1579$$

$$P(\text{numérico}|\text{positivo}) = \frac{0+1}{19} = 0.0526$$

$$P(\text{pequeno}|\text{positivo}) = \frac{0+1}{19} = 0.0526$$

$$P(\text{teclado}|\text{positivo}) = \frac{0+1}{19} = 0.0526$$

$$P(\text{muito}|\text{negativo}) = \frac{1+1}{1+15} = \frac{2}{16} = 0.125$$

$$P(\text{numérico}|\text{negativo}) = \frac{0+1}{16} = 0.0625$$

$$P(\text{pequeno}|\text{negativo}) = \frac{0+1}{16} = 0.0625$$

$$P(\text{teclado}|\text{negativo}) = \frac{0+1}{16} = 0.0625$$

(6) seria  
classificado  
como “negativo”

$$P(\text{positivo}|X) \propto 0.8 \times 0.1579 \times 0.0526 \times 0.0526 \times 0.0526$$

$$P(\text{positivo}|X) \approx 0.8 \times 0.0000228 \approx 0.0000182$$

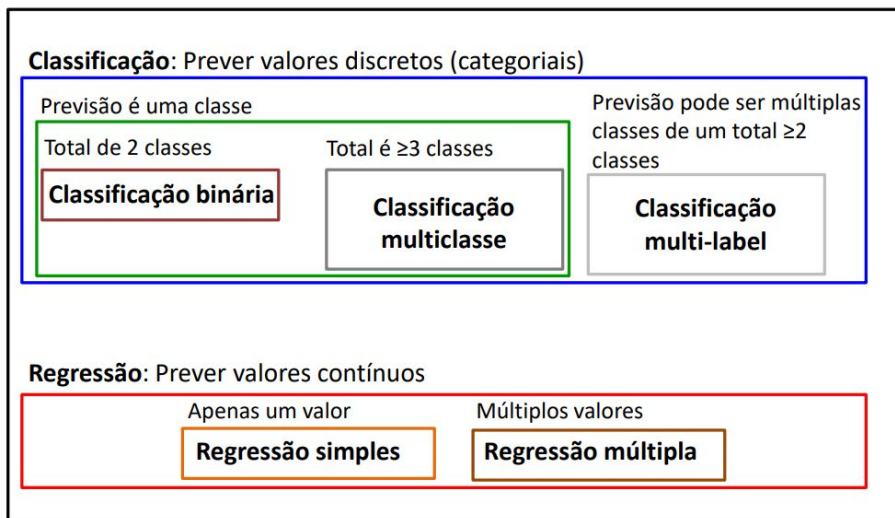
$$P(\text{negativo}|X) \propto 0.2 \times 0.125 \times 0.0625 \times 0.0625 \times 0.0625$$

$$P(\text{negativo}|X) \approx 0.2 \times 0.000122 \approx 0.0000244$$



# Tarefas de aprendizado

- Algoritmos de aprendizado de máquina podem ser aplicados a diferentes problemas



## Classificação:

### Classificação binária

#### Diagnóstico Médico:

Determinar se o paciente tem ou não uma doença a partir de resultados de exames médicos

### Classificação multiclasse

#### Reconhecimento de Dígitos

**Manuscritos:** Dada uma imagem de um dígito, determinar qual é o dígito representado na imagem

### Classificação multi-label

#### Etiquetagem de Imagens:

Dada uma imagem, identificar os objetos presentes na imagem

## Regressão:

### Regressão simples

#### Previsão de consumo de energia:

Dado o consumo de energia dos últimos N dias, prever o consumo do dia seguinte

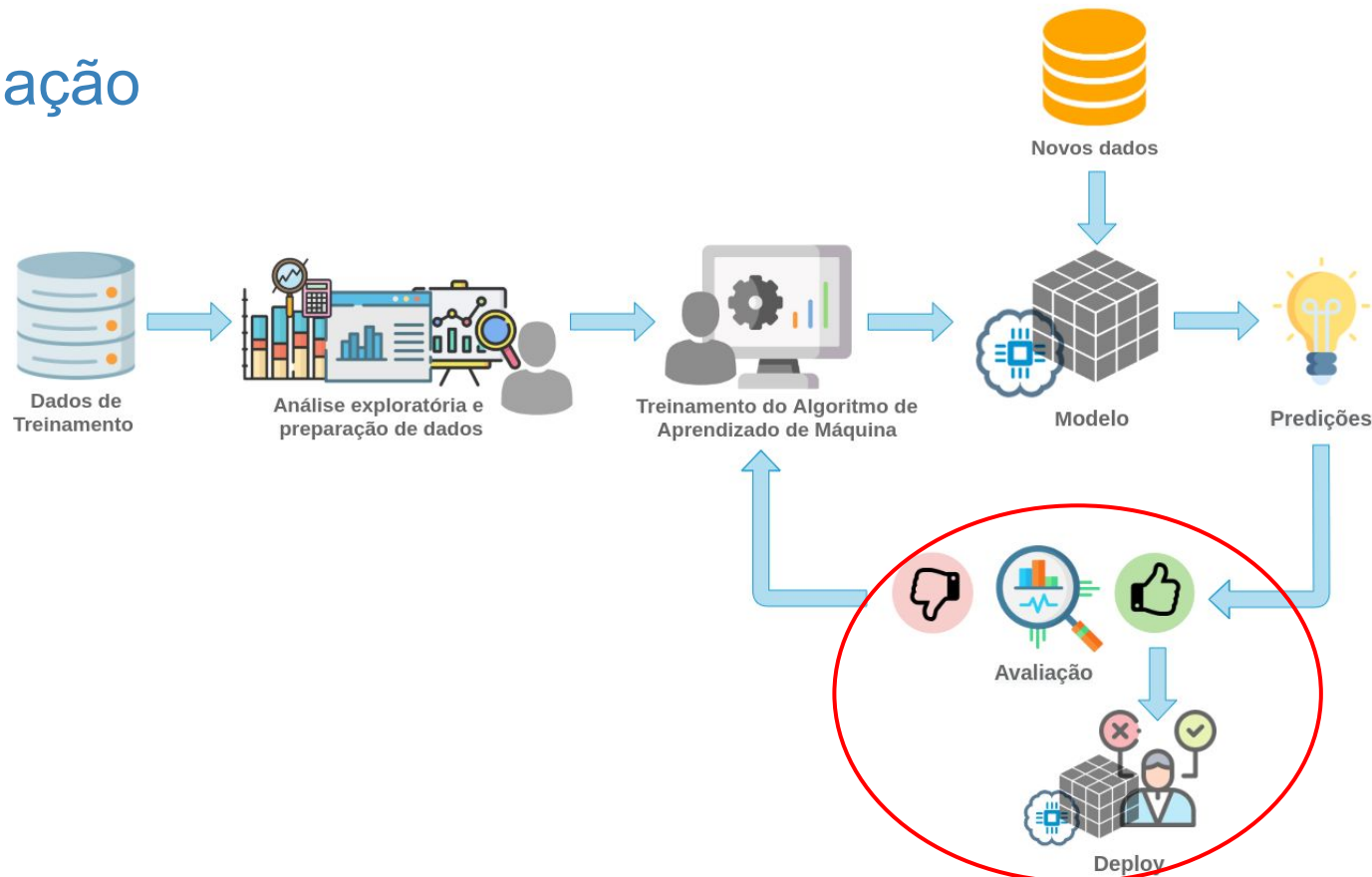
### Regressão múltipla

#### Previsão do clima:

Dados os valores históricos de diferentes dados do clima (pressão, temperatura, umidade) dos últimos dias, prever esses dados para o dia seguinte

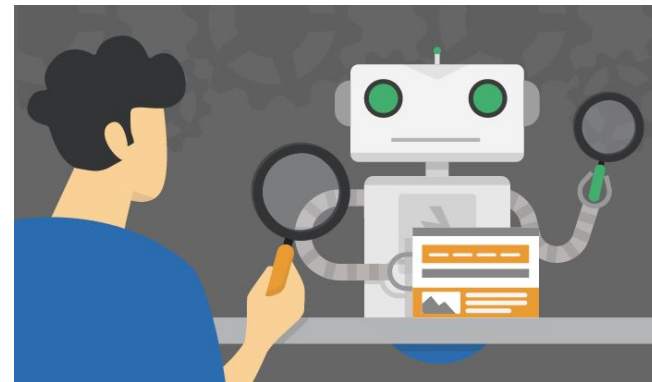
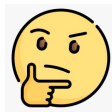
# Avaliação de modelos supervisionados para texto

# Avaliação



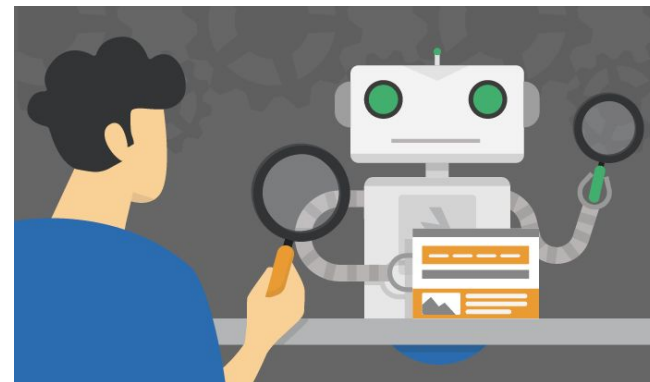
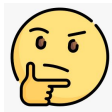
# Avaliação

- Como garantir que o modelo é bom o suficiente para auxiliar na tomada de decisão?



# Avaliação

- Como garantir que o modelo é bom o suficiente para auxiliar na tomada de decisão?
  - Modelo precisar **generalizar** o aprendizado!
- É suficiente avaliar a performance do modelo apenas no conjunto de dados de treinamento?



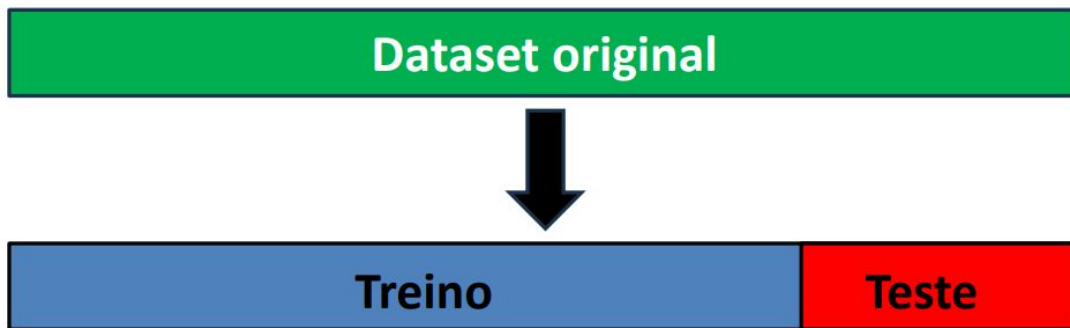
# Avaliação

- Como garantir que o modelo é bom o suficiente para auxiliar na tomada de decisão?
  - Modelo precisar **generalizar** o aprendizado!
- É suficiente avaliar a performance do modelo apenas no conjunto de dados de treinamento?
  - **Não!**
  - Uma performance boa no conjunto de treinamento **não é evidência** da performance para dados “novos” (que não foram usados durante o treinamento)
  - É fundamental separar “**dados de teste**” (não são usados durante o treinamento) para **avaliar o modelo**



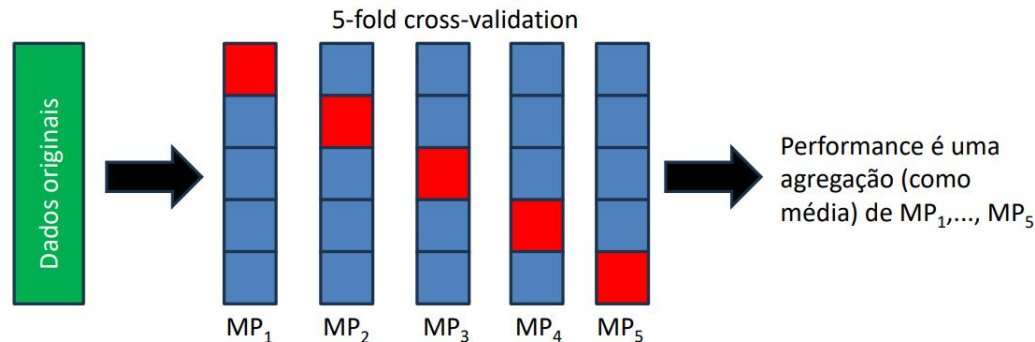
# Avaliação: Divisão dos dados

- Com *holdout*:
  - O conjunto original deve ser separado **aleatoriamente** em dois subconjuntos:
    - **Treino**: Dados utilizados para ajustar os parâmetros (70~80% dos dados)
    - **Teste**: Dados utilizados para testar o modelo (30~20% dos dados)
  - Utilize sempre que possível divisão estratificada (representação proporcional dos estratos de todo o dataset em cada conjunto)



# Avaliação: Divisão dos dados

- Com *K-fold cross validation*:
  - O conjunto original de dados é separado em k subconjuntos (folds):
  - São realizadas k iterações, onde em cada iteração i:
    - O i-ésimo fold é usado como teste
    - A *união dos demais folds* é utilizada como dados de treinamento
    - Treina-se um modelo por iteração, avaliando-o no conjunto de teste
    - A performance final do modelo é uma *agregação* (média, por exemplo) da performance considerando a performance em todos os testes



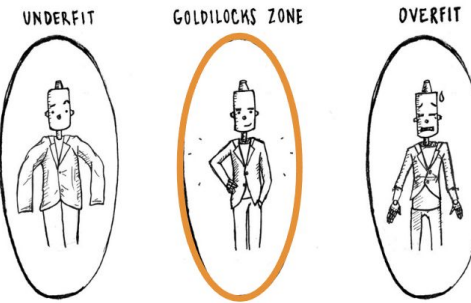


# Avaliação: Divisão dos dados

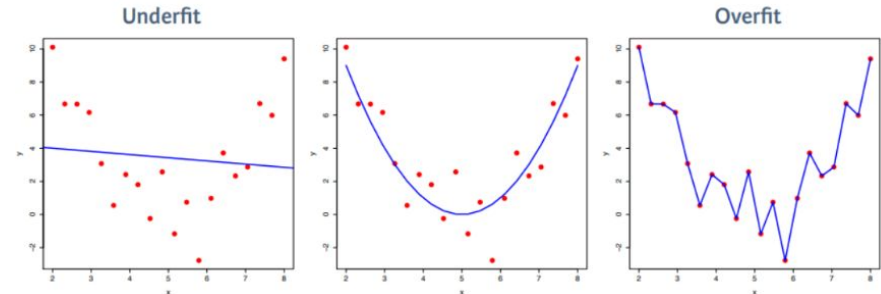
- Utilizando conjunto de validação (dev):
  - **Treino:** Ajusta os parâmetros do modelo
  - **Teste:** Mede a performance final
  - **Validação (*development test*):**
    - Permite parar prematuramente durante o treinamento caso a performance piore ou trave (*early stopping*)
    - Permite selecionar hiperparâmetros (arquitetura, taxa de aprendizado etc.) que geram a melhor performance no conjunto de validação

# Avaliação: Problemas na generalização

## MACHINE LEARNING GENERALIZATION FINDING THE PERFECT FIT



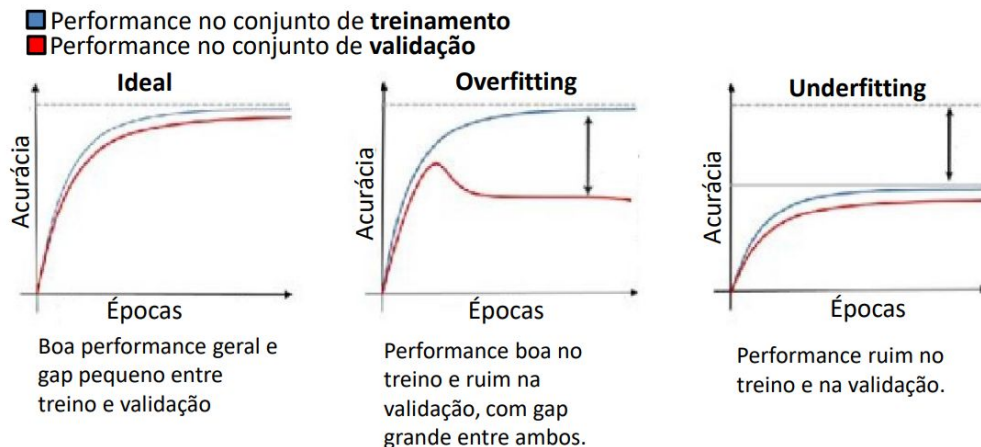
- **Underfitting** (subajuste):
  - Ocorre quando o modelo não é capaz de se ajustar corretamente aos dados de treinamento
  - Modelo muito simples para o problema
- **Overfitting** (sobreajuste):
  - Ocorre quando o modelo se ajusta demais aos dados de treinamento, impactando negativamente a sua generalização
  - Modelo muito complexo para o problema



# Avaliação: Problemas na generalização

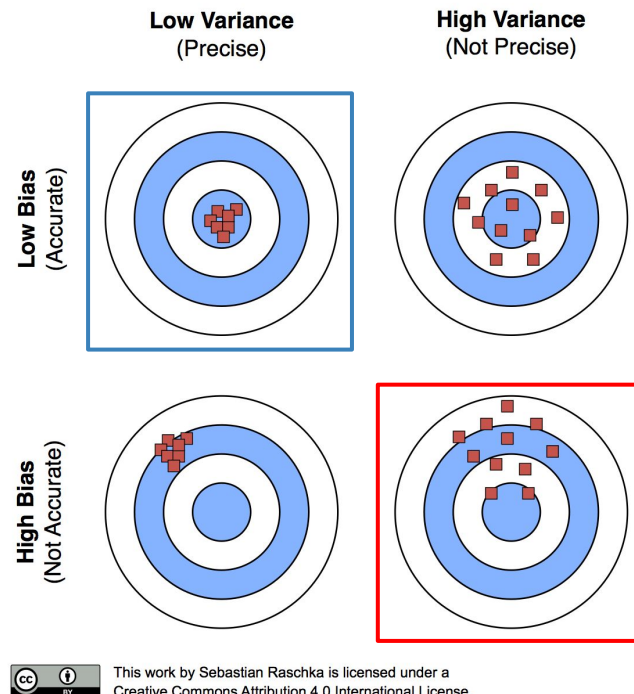
- **Época:**
  - Ciclo completo de passagem pelos dados de treino durante o processo de aprendizado de um modelo
  - Cada época ocorre quando o modelo vê todos os exemplos do treinamento uma vez

- **Muitas épocas?**
  - Muitas vezes o modelo precisa repetir o processo para ajustar seus pesos para minimizar o erro
  - Curvas de aprendizado auxiliam na visualização



# Viés vs. Variância (Taxa de erro)

- Erros cometidos por algoritmos supervisionados podem ser decompostos em viés e variância
- **Viés (bias):** erro que deriva de suposições (errôneas) assumidas pelo algoritmo na construção do modelo
  - **Alto viés:** adota suposições fortes, dificuldade em capturar relações importantes entre atributos e saídas
- **Variância:** reflete a sensibilidade do modelo a variações nos dados de treinamento, isto é, o quanto seu desempenho pode flutuar com estas variações
  - **Alta variância:** pequenas mudanças no dado podem gerar perturbações significativas no modelo



# Métricas de avaliação

- Desempenho é avaliado comparando-se o **valor predito** com o **valor real** do atributo alvo
  - Existem diversas métricas!
- Diferem para problemas de classificação e de regressão
  - Regressão:
    - **Erro absoluto médio (MAE)**: média da diferença absoluta entre o valor previsto e o valor real
    - **Erro quadrático médio (MSE)**: média da diferença entre o valor previsto e o valor real ao quadrado

$$MAE(f) = \frac{1}{n} \sum_{i=1}^n |y_i - f(\mathbf{x}_i)|$$

$$MSE(f) = \frac{1}{n} \sum_{i=1}^n (y_i - f(\mathbf{x}_i))^2$$

# Métricas de avaliação

- Desempenho é avaliado comparando-se o **valor predito** com o **valor real** do atributo alvo
  - Existem diversas métricas!
- Diferem para problemas de classificação e de regressão (binários)
  - Classificação:
    - **Matriz de confusão**: ferramenta visual que mostra a distribuição de previsões corretas e incorretas para cada classe

		Classe predita	
		+	-
Classe verdadeira	+	VP	FN
	-	FP	VN

**Verdadeiros Positivos (VP)**: instâncias positivas corretamente classificadas como positivas

**Verdadeiros Negativos (VN)**: instâncias negativas corretamente classificadas como negativas







**Falsos Positivos (FP)**: instâncias negativas incorretamente classificadas como positivas

**Falsos Negativos (FN)**: instâncias positivas incorretamente classificadas como negativas

# Métricas de avaliação

- Desempenho é avaliado comparando-se o **valor predito** com o **valor real** do atributo alvo
  - Existem diversas métricas!
- Diferem para problemas de classificação e de regressão (**multiclasse**)
  - Classificação:
    - **Matriz de confusão (multiclasse)**

		Classe predita	
		+	-
Classe verdadeira	+	VP	FN
	-	FP	VN

Real \ Predito	 Maçã	 Banana	 Laranja
 Maçã	10	2	1
 Banana	1	15	0
 Laranja	0	3	8

# Métricas de avaliação

- Desempenho é avaliado comparando-se o **valor predito** com o **valor real** do atributo alvo
  - Existem diversas métricas!
- Diferem para problemas de classificação e de regressão (binários)
  - Classificação:
    - Matriz de confusão
    - **Acurácia (taxa de acerto)**: proporção de exemplos classificados corretamente
      - Problemas com dados desbalanceados (medida que “engana”)

$$\text{Acurácia} = \frac{\text{Número de Previsões Corretas}}{\text{Número Total de Previsões}}$$



# Métricas de avaliação

- Desempenho é avaliado comparando-se o **valor predito** com o **valor real** do atributo alvo
  - Existem diversas métricas!
- Diferem para problemas de classificação e de regressão (binários)
  - Classificação:
    - Matriz de confusão
    - Acurácia (taxa de acerto)
    - **Precisão (*precision*)**: proporção de previsões positivas corretas

		Classe predita	
		+	-
Classe verdadeira	+	VP	FN
	-	FP	VN

$$prec(f) = \frac{VP}{VP + FP}$$

# Métricas de avaliação

- Desempenho é avaliado comparando-se o **valor predito** com o **valor real** do atributo alvo
  - Existem diversas métricas!
- Diferem para problemas de classificação e de regressão (binários)
  - Classificação:
    - Matriz de confusão
    - Acurácia (taxa de acerto)
    - Precisão (*precision*)
    - **Recall (revocação)**: proporção de verdadeiros positivos identificados corretamente

		Classe predita	
		+	-
Classe verdadeira	+	VP	FN
	-	FP	VN

$$rev(f) = \frac{VP}{VP + FN}$$

# Métricas de avaliação

- Desempenho é avaliado comparando-se o **valor predito** com o **valor real** do atributo alvo
  - Existem diversas métricas!
- Diferem para problemas de classificação e de regressão (binários)
  - Classificação:
    - Matriz de confusão
    - Acurácia (taxa de acerto)
    - **Precisão (*precision*)**
    - **Recall (revocação)**

**Tradeoff** entre **Precisão e Recall!**

Precisão: exatidão do modelo

Recall: completude do modelo

Dependendo do domínio, há interesse em dar ênfase à minimização de um tipo de erro específico:

Modelo orientado à precisão : visa minimizar FP

Ex: Sistemas de busca; classificação de documentos

Modelo orientado ao recall : visa minimizar FN

Ex: Domínios médicos (ex: detecção de tumores)

# Métricas de avaliação

- Desempenho é avaliado comparando-se o **valor predito** com o **valor real** do atributo alvo
  - Existem diversas métricas!
- Diferem para problemas de classificação e de regressão (binários)
  - Classificação:
    - Matriz de confusão
    - Acurácia (taxa de acerto)
    - **Precisão (*precision*)**
    - **Recall (revocação)**

**Tradeoff** entre **Precisão e Recall!**

Precisão: exatidão do modelo

Recall: completude do modelo

Dependendo do domínio, há interesse em dar ênfase à minimização de um tipo de erro específico:

Modelo orientado à precisão : visa minimizar FP

Ex: Sistemas de busca; classificação de documentos

Modelo orientado ao recall : visa minimizar FN

Ex: Domínios médicos (ex: detecção de tumores)

# Métricas de avaliação

- Desempenho é avaliado comparando-se o **valor predito** com o **valor real** do atributo alvo
  - Existem diversas métricas!
- Diferem para problemas de classificação e de regressão
  - Classificação:
    - **F1-Score**: média harmônica entre precisão e revocação ( **muito usado!**)
      - **Binário**: quando há apenas duas classes

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall}$$

Mesmo peso para precisão e recall

# Métricas de avaliação

- Desempenho é avaliado comparando-se o **valor predito** com o **valor real** do atributo alvo
  - Existem diversas métricas!
- Diferem para problemas de classificação e de regressão
  - Classificação:
    - **F1-Score**: média harmônica entre precisão e revocação ( **muito usado!**)
      - Binário: quando há apenas duas classes
      - Multiclasse:
        - **F1-Score Micro**: calcula as métricas globalmente, somando todos os TPs, FPs e FNs das classes. Indicado para classes desbalanceadas, mas deseja-se dar o mesmo peso para cada instância (independente da sua classe)

$$F1_{\text{micro}} = 2 \times \frac{Precision_{\text{micro}} \times Recall_{\text{micro}}}{Precision_{\text{micro}} + Recall_{\text{micro}}}$$

# Métricas de avaliação

- Desempenho é avaliado comparando-se o **valor predito** com o **valor real** do atributo alvo
  - Existem diversas métricas!
- Diferem para problemas de classificação e de regressão
  - Classificação:

- **F1-Score**: média harmônica entre precisão e revocação (**muito usado!**)

- Binário: quando há apenas duas classes
- Multiclasse:

- **F1-Score Macro**: calcula o F1 para cada classe separadamente e tira a média aritmética. Indicado para classes balanceadas, onde todas são igualmente importantes e minoritárias não são ignoradas

$$F1_{\text{macro}} = \frac{F1_{\text{Classe 1}} + F1_{\text{Classe 2}} + \dots + F1_{\text{Classe N}}}{N}$$

# Métricas de avaliação

- Desempenho é avaliado comparando-se o **valor predito** com o **valor real** do atributo alvo
  - Existem diversas métricas!
- Diferem para problemas de classificação e de regressão
  - Classificação:

- **F1-Score**: média harmônica entre precisão e revocação (**muito usado!**)

- Binário: quando há apenas duas classes
- Multiclasse:

- **F1-Score Weighted**: calcula o F1 para cada classe separadamente e tira a média ponderada pelo número de instâncias da classe. Indicado para classes desbalanceadas, onde classes maiores têm mais importância

$$F1_{\text{weighted}} = \sum_{i=1}^N (F1_{\text{Classe } i} \times w_i)$$



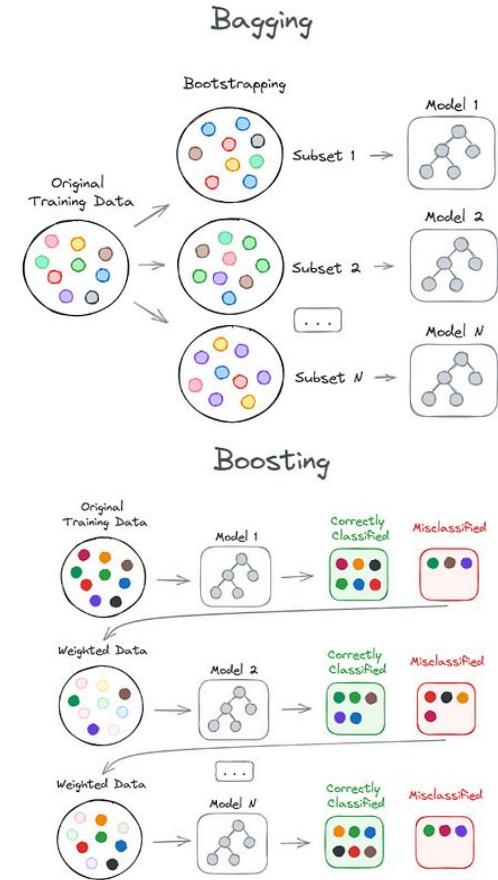
# Métricas de avaliação

- Desempenho é avaliado comparando-se o **valor predito** com o **valor real** do atributo alvo
  - Existem diversas métricas!
- Diferem para problemas de classificação e de regressão
  - Classificação:
    - **F1-Score**

Métrica	Cálculo
<b>Macro</b>	Média dos F1s por classe
<b>Weighted</b>	F1s ponderados pelo tamanho
<b>Micro</b>	Soma global de TP/FP/FN

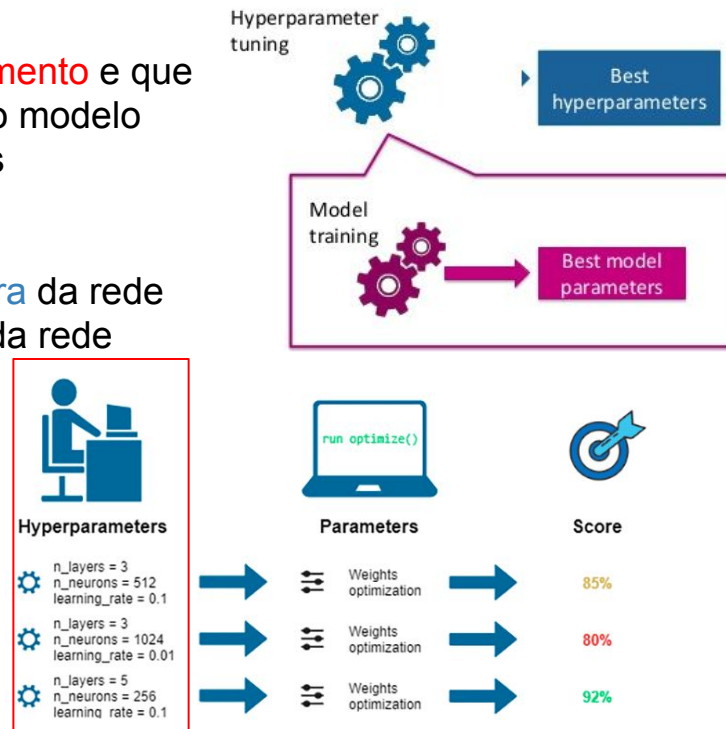
# Ensemble Learning

- Combina **múltiplos modelos** (*base learners*) para melhorar o desempenho e a robustez das previsões
  - Ideia: a diversidade entre os modelos pode reduzir erros e aumentar a precisão
- Exemplos:
  - **Bagging** (Bootstrap Aggregating): treina vários modelos em diferentes subconjuntos do mesmo conjunto de dados  
Exemplo: Random Forest (múltiplas árvores de decisão)
  - **Boosting**: treina sequencialmente, dando mais peso aos exemplos que foram mal classificados em etapas anteriores  
Exemplos: AdaBoost, Gradient Boosting, XGBoost
  - **Voting**: combina as previsões de vários modelos usando votação (maioria) ou média das previsões



# Hiperparâmetros

- Parâmetros **configurados manualmente antes do treinamento** e que influenciam diretamente o desempenho e a eficiência do modelo
  - São diferentes dos parâmetros, que são ajustados automaticamente durante o treino
- Controlam o **comportamento** do treinamento e a **estrutura** da rede
  - Possuem grande impacto direto no desempenho da rede
- Exemplos:
  - Número de camadas e neurônios por camada
  - Taxa de aprendizado inicial
  - Decaimento da taxa de aprendizado
  - Tipo de otimizador
  - Função de ativação, função de perda, *batch size*



# Próximas aulas

- Aula prática (Laboratório 3)
- Aula teórica:
  - Representação de textos com word embeddings fixas

# Material complementar

- Para mais detalhes:
  - Slides do Prof. Anderson Tavares (Moodle) sobre Avaliação de Modelos

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL  
Instituto de Informática  
Departamento de Informática Aplicada

**Obrigado pela atenção!**  
**Dúvidas?**

Prof. Dennis Giovani Balreira

(Material adaptado dos Profs. Joel Carbonera, Anderson Tavares, Viviane Moreira e Dan Jurafsky)



INF01221 - Tópicos Especiais em Computação XXXVI:  
Processamento de Linguagem Natural

