

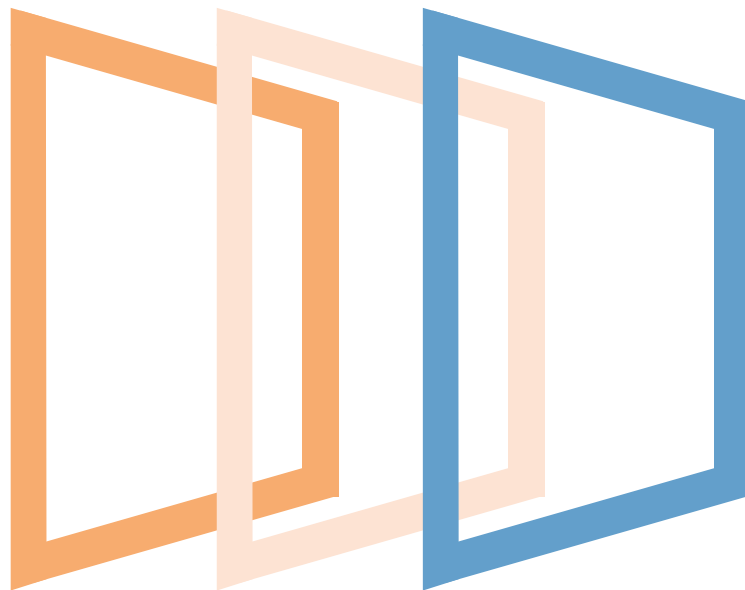
Desvendando IA e Machine Learning

Thaís Ratis

Diego Alexandre

Práticas Tecnológicas, 10.10.2023

minsoit



An Indra company

Thaís Ratis

Instrutora

Formação Acadêmica

B.Sc. Ciência da Computação

Esp. Bioinformática

M.Sc. Bioinformática

Ph.D. Bioinformática

Área Profissional

Cientista de Dados

Contato no Teams

talmeidar@minsait.com



Diego Alexandre

Instrutor

Formação Acadêmica

B.Sc. Engenharia de Computação

Mestrando Informática

Área Profissional

Cientista de Dados

Contato no Teams

dpatricio@minsait.com



Conteúdo programático

Aula 1: Revisão de pré-processamento e ML

Aula 2: Redes Neurais e Modelo Ensemble

Aula 3: Redes Neurais Convolucionais

Aula 4: Mlops

Conteúdo programático

Revisão

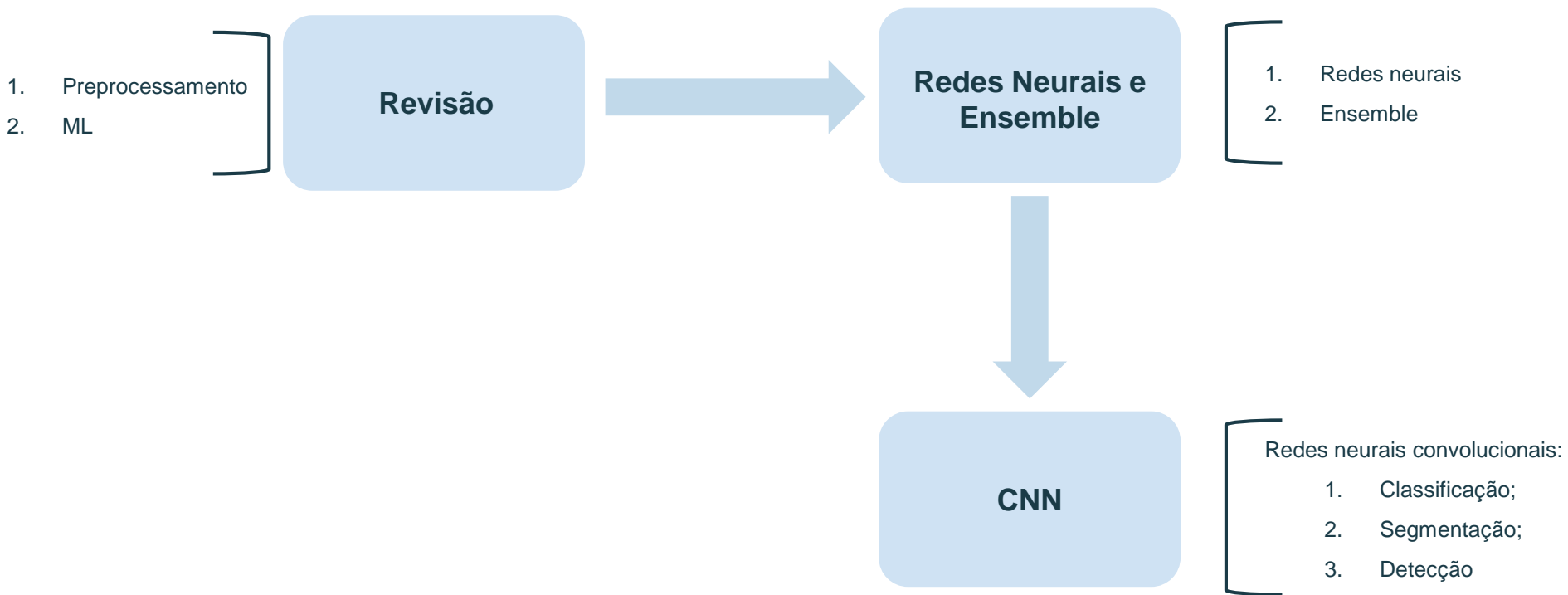
Revisão:

1. Préprocessamento
2. ML

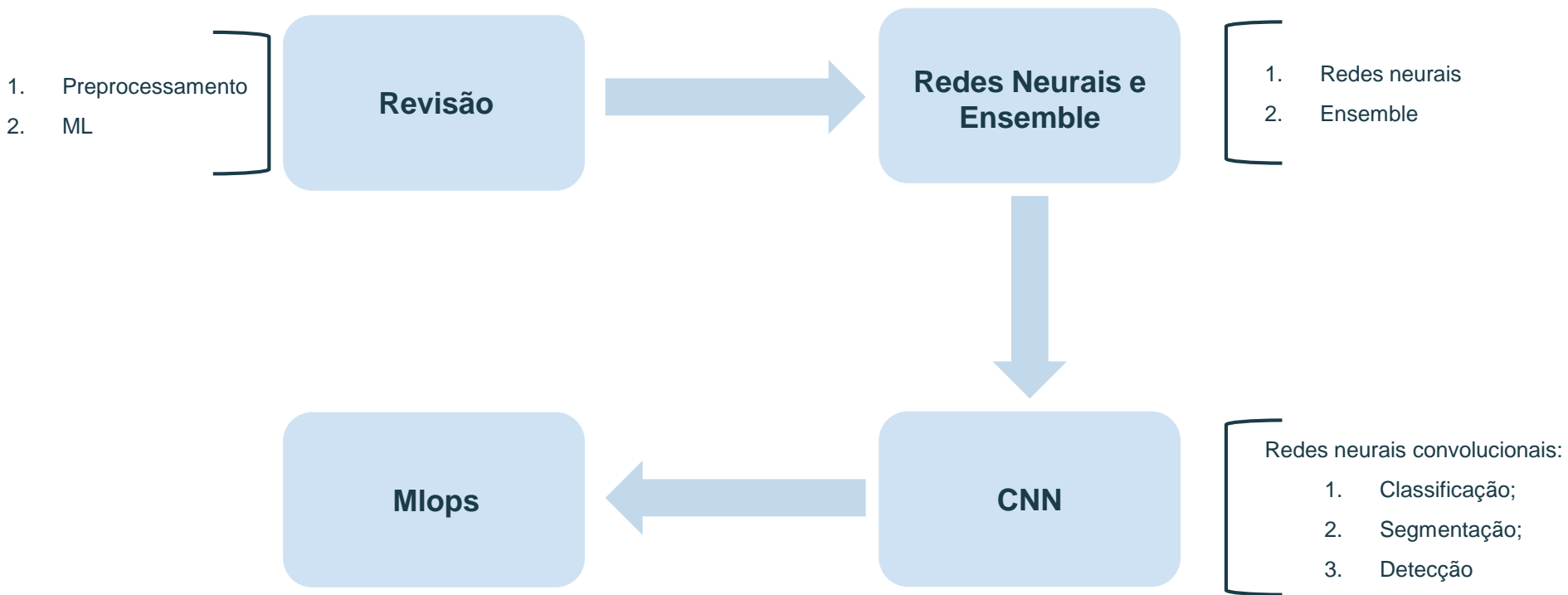
Conteúdo programático



Conteúdo programático



Conteúdo programático



Revisão pré-processamento e ML

Aula 01

Objetivo

Capacitar os alunos em pré-processamento relevantes para análise de dados, incluindo limpeza, transformações, imputação de valores ausentes e inferências, garantindo resultados precisos e confiáveis. Além disso, oferecer uma base sólida em conceitos e técnicas fundamentais de Machine Learning (ML), abordando diferentes tipos de algoritmos e princípios de aprendizado supervisionado e não supervisionado, como regressão, classificação e agrupamento. Os alunos serão capacitados a identificar problemas adequados para ML, selecionar e avaliar algoritmos, interpretar resultados e aplicar técnicas em análise de dados, preparação e avaliação de modelos.

Redes Neurais e Ensemble

Aula 02

Objetivo

Explorar um modelo inspirado no funcionamento do cérebro humano, bem como dos algoritmos de ensemble, que combinam múltiplos modelos para melhorar o desempenho. A aula visa capacitar os alunos a compreender a arquitetura de redes neurais, suas aplicações e treinamento, bem como a importância de escolher estratégias de ensemble adequadas para melhorar a precisão e robustez dos modelos de aprendizado de máquina. Isso prepara os alunos para abordar problemas complexos e desafiadores em ciência de dados e inteligência artificial.

Redes Neurais Convolucionais

Aula 03

Objetivo

Capacitar os alunos a compreender e aplicar redes neurais convolucionais, uma classe de modelos de aprendizado profundo, para tarefas avançadas e de processamento de imagens. Isso inclui a capacidade de classificar objetos em imagens, segmentar regiões de interesse e detectar objetos específicos em cenas complexas. A aula visa fornecer uma compreensão sólida das arquiteturas de CNN, técnicas de treinamento e estratégias de ajuste fino para essas aplicações específicas, preparando os alunos para enfrentar desafios de visão computacional e desenvolver soluções eficazes em uma ampla gama de campos, como reconhecimento de imagem, medicina, veículos autônomos e muito mais.

Mlops

Aula

04

Objetivo

Fornecer aos alunos uma compreensão abrangente e prática de como gerenciar e implantar modelos de machine learning em ambientes de produção de forma eficiente e escalável. Isso inclui a automação de pipelines de treinamento, avaliação e implantação de modelos, bem como o monitoramento contínuo do desempenho dos modelos em produção. A aula visa capacitar os alunos a aplicar práticas de engenharia de software e DevOps aos desafios exclusivos apresentados pelo ciclo de vida de machine learning, garantindo a entrega eficaz e confiável de modelos em aplicações do mundo real.

Revisão pré-processamento e ML

Aula 01

Conteúdo programático

1. O que é pré-processamento?
2. Principais tipos de pré-processamento
3. O que é ML?
4. Aprendizado Supervisionado
5. Aprendizado Não supervisionado

Pré-processamento

01

O que é Pré-processamento
e qual seu objetivo?

1.1

Objetivo do pré-processamento

De forma resumida, o objetivo do pré-processamento de dados é garantir que os dados estejam prontos para serem usados em modelos de análise e para que os resultados da análise sejam confiáveis e precisos.

Quais as principais etapas
de um pré-processamento?

1.2

Quais as principais etapas de um pré-processamento?

Limpeza de dados

Transformação de dados

Redução de dimensionalidade

Balanceamento de dados

Separação de dados

Principais etapas no processo de limpeza dos dados

**Eliminação
manual do
dados**

**Identificação de
erros e
inconsistências**

**Remoção de
duplicatas**

**Tratamento de
valores
ausentes**

**Verificação de
integridade**

**Validação de
dados**

Limpeza dos dados - Eliminação manual do dados

Quando um atributo não contribui para a estimativa do valor do atributo alvo, ou seja, ele é considerado irrelevante.

```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

df = pd.read_csv("hospital.csv", sep = ';')

df = df.drop(columns=['identificador', 'nome'])

df
```

	identificador	nome	idade	sexo	peso	manchas	temperatura	internacoes	estado	diagnostico
0	4201	Joao	28	M	72.0	Concentradas	38.0	2	SP	Doente
1	3217	Maria	18	F	67.0	Inexistentes	39.5	4	MG	Saudavel
2	4039	Luiz	49	M	92.0	Espalhadas	38.0	2	RS	Doente
3	1920	Jose	18	M	43.0	Inexistentes	38.5	8	MG	Saudavel
5	2301	Ana	22	F	72.0	Inexistentes	58.0	3	RJ	Doente
6	2301	Ana	22	F	72.0	Inexistentes	58.0	3	RJ	Doente
7	2301	Ana	22	F	72.0	Inexistentes	58.0	3	RJ	Doente
8	1322	Marta	19	F	87.0	Espalhadas	39.0	6	AM	Saudavel
9	3027	Paulo	34	M	67.0	Uniformes	38.4	2	GO	Saudavel

Limpeza dos dados - Identificação de erros e inconsistências

Nesta etapa, os dados são analisados para identificar valores ausentes, erros de digitação, duplicatas, inconsistências e outros problemas.

	identificador	nome	idade	sexo	peso	manchas	temperatura	internacoes	estado	diagnostico
0	4201	Joao	28	M	NaN	Concentradas	38.0	2	SP	Doente
1	3217	Maria	18	F	67.0	Inexistentes	39.5	4	MG	Saudavel
2	4039	Luiz	49	M	92.0	Espalhadas	38.0	2	RS	Doente
3	1920	Jose	18	M	43.0	Inexistentes	38.5	8	MG	Saudavel
4	4340	Claudia	21	F	52.0	Uniformes	NaN	1	PE	Doente
5	2301	Ana	22	F	72.0	Inexistentes	58.0	3	RJ	Doente
6	2301	Ana	22	F	72.0	Inexistentes	58.0	3	RJ	Doente
7	2301	Ana	22	F	72.0	Inexistentes	58.0	3	RJ	Doente
8	1322	Marta	19	F	87.0	Espalhadas	39.0	6	AM	Saudavel
9	3027	Paulo	34	M	67.0	Uniformes	38.4	2	GO	Saudavel

Limpeza dos dados - Remoção de duplicatas

Processo de pré-processamento de dados que envolve a identificação e exclusão de observações ou registros que são duplicados em um conjunto de dados. Em outras palavras, a remoção de duplicatas é uma técnica utilizada para garantir que cada registro ou observação em um conjunto de dados seja único. Isso é importante porque as duplicatas podem distorcer a análise de dados e prejudicar a qualidade dos resultados.

ANTES

	identificador	nome	idade	sexo	peso	manchas	temperatura	internacoes	estado	diagnostico
0	4201	Joao	28	M	NaN	Concentradas	38.0	2	SP	Doente
1	3217	Maria	18	F	67.0	Inexistentes	39.5	4	MG	Saudavel
2	4039	Luiz	49	M	92.0	Espalhadas	38.0	2	RS	Doente
3	1920	Jose	18	M	43.0	Inexistentes	38.5	8	MG	Saudavel
4	4340	Claudia	21	F	52.0	Uniformes	NaN	1	PE	Doente
5	2301	Ana	22	F	72.0	Inexistentes	58.0	3	RJ	Doente
6	2301	Ana	22	F	72.0	Inexistentes	58.0	3	RJ	Doente
7	2301	Ana	22	F	72.0	Inexistentes	58.0	3	RJ	Doente
8	1322	Marta	19	F	87.0	Espalhadas	39.0	6	AM	Saudavel
9	3027	Paulo	34	M	67.0	Uniformes	38.4	2	GO	Saudavel

DEPOIS

```
# Elimina todas as linhas com dados duplicados  
df.drop_duplicates()
```

	identificador	nome	idade	sexo	peso	manchas	temperatura	internacoes	estado	diagnostico
0	4201	Joao	28	M	NaN	Concentradas	38.0	2	SP	Doente
1	3217	Maria	18	F	67.0	Inexistentes	39.5	4	MG	Saudavel
2	4039	Luiz	49	M	92.0	Espalhadas	38.0	2	RS	Doente
3	1920	Jose	18	M	43.0	Inexistentes	38.5	8	MG	Saudavel
4	4340	Claudia	21	F	52.0	Uniformes	NaN	1	PE	Doente
5	2301	Ana	22	F	72.0	Inexistentes	58.0	3	RJ	Doente
8	1322	Marta	19	F	87.0	Espalhadas	39.0	6	AM	Saudavel
9	3027	Paulo	34	M	67.0	Uniformes	38.4	2	GO	Saudavel

Limpeza dos dados - Tratamento de valores ausentes

Valores ausentes devem ser tratados de forma adequada, seja através de imputação (preenchimento com valores substitutos, por ex: média, moda ou mediana) ou remoção de linhas ou colunas inteiras.

ANTES

	identificador	nome	idade	sexo	peso	manchas	temperatura	internacoes	estado	diagnostico
0	4201	Joao	28	M	NaN	Concentradas	38.0	2	SP	Doente
1	3217	Maria	18	F	67.0	Inexistentes	39.5	4	MG	Saudavel
2	4039	Luiz	49	M	92.0	Espalhadas	38.0	2	RS	Doente
3	1920	Jose	18	M	43.0	Inexistentes	38.5	8	MG	Saudavel
4	4340	Claudia	21	F	52.0	Uniformes	NaN	1	PE	Doente
5	2301	Ana	22	F	72.0	Inexistentes	58.0	3	RJ	Doente
6	2301	Ana	22	F	72.0	Inexistentes	58.0	3	RJ	Doente
7	2301	Ana	22	F	72.0	Inexistentes	58.0	3	RJ	Doente
8	1322	Marta	19	F	87.0	Espalhadas	39.0	6	AM	Saudavel
9	3027	Paulo	34	M	67.0	Uniformes	38.4	2	GO	Saudavel

DEPOIS

```
#Substituir o valor faltante pela mediana  
median = df['peso'].median()  
df['peso'].fillna(median, inplace=True)
```

```
df
```

```
Valores faltantes: identificador    0  
nome                        0  
idade                      0  
sexo                       0  
peso                       1  
manchas                    0  
temperatura                1  
internacoes                0  
estado                     0  
diagnostico                0  
dtype: int64
```

	identificador	nome	idade	sexo	peso	manchas	temperatura	internacoes	estado	diagnostico
0	4201	Joao	28	M	72.0	Concentradas	38.0	2	SP	Doente
1	3217	Maria	18	F	67.0	Inexistentes	39.5	4	MG	Saudavel

Limpeza dos dados - Verificação de integridade

A verificação de integridade é um processo de pré-processamento de dados que tem como objetivo garantir que os dados estejam completos e não contenham erros ou valores inconsistentes que possam comprometer a qualidade da análise ou do modelo de aprendizado de máquina construído a partir desses dados. Assim, os dados são verificados para garantir que as relações estão corretas e que não há valores discrepantes.

	identificador	nome	idade	sexo	peso	manchas	temperatura	internacoes	estado	diagnostico
0	4201	Joao	28	M	67	Concentradas	38.0	2	SP	Doente
1	3217	Maria	18	F	67	Inexistentes	39.5	4	MG	Saudavel
2	4039	Luiz	49	M	92	Espalhadas	38.0	2	RS	Doente
3	1920	Jose	18	M	43	Inexistentes	38.5	8	MG	Saudavel
4	4340	Claudia	21	F	52	Uniformes	38.5	1	PE	Doente
5	4340	Claudia	21	F	52	Uniformes	38.5	1	PE	Saudavel
6	2301	Ana	22	F	72	Inexistentes	58.0	3	RJ	Doente
7	2301	Ana	22	F	72	Inexistentes	58.0	3	RJ	Doente
8	2301	Ana	22	F	72	Inexistentes	58.0	3	RJ	Doente
9	1322	Marta	19	F	87	Espalhadas	39.0	6	AM	Saudavel
10	3027	Paulo	34	M	67	Uniformes	38.4	2	GO	Saudavel

Limpeza dos dados - Validação de dado

A validação de dados é uma etapa importante do processo de pré-processamento de dados que tem como objetivo verificar se os dados são adequados e relevantes para a análise ou modelo de aprendizado de máquina em questão. Assim, os dados limpos são validados para garantir que atendam aos requisitos de qualidade necessários para a análise, visto que dados inadequados podem levar a resultados incorretos e conclusões inválidas.

	identificador	nome	idade	sexo	peso	manchas	temperatura	internacoes	estado	diagnostico
0	4201	Joao	28	M	72.0	Concentradas	38.0	2	SP	Doente
1	3217	Maria	18	F	67.0	Inexistentes	39.5	4	MG	Saudavel
2	4039	Luiz	49	M	92.0	Espalhadas	38.0	2	RS	Doente
3	1920	Jose	18	M	43.0	Inexistentes	38.5	8	MG	Saudavel
5	2301	Ana	22	F	72.0	Inexistentes	58.0	3	RJ	Doente
6	2301	Ana	22	F	72.0	Inexistentes	58.0	3	RJ	Doente
7	2301	Ana	22	F	72.0	Inexistentes	58.0	3	RJ	Doente
8	1322	Marta	19	F	87.0	Espalhadas	39.0	6	AM	Saudavel
9	3027	Paulo	34	M	67.0	Uniformes	38.4	2	GO	Saudavel



Transformação de dados

Inclui a normalização (transformar as variáveis em uma escala comum), padronização (conversão de valores em uma forma consistente e uniforme, por exemplo, a padronização de datas em um único formato), conversão de valores e outros tipos de transformações para garantir que os dados sejam comparáveis e que o modelo de análise possa funcionar corretamente.

	name	gender	age	city		name	gender	age	city	
a	Abby	F	33	Berlin	→	a	Abby	0	33	0
b	Ben	M	16	Tokyo		b	Ben	1	16	2
c	Charlie	M	22	Sydney		c	Charlie	1	22	1
d	Dave	M	65	York		d	Dave	1	65	3
e	Ella	F	18	Sydney		e	Ella	0	18	1

Transformação de dados

	identificador	nome	idade	sexo	peso	manchas	temperatura	internacoes	estado	diagnostico
0	4201	Joao	28	M	72.0	Concentradas	38.0	2	SP	Doente
1	3217	Maria	18	F	67.0	Inexistentes	39.5	4	MG	Saudavel
2	4039	Luiz	49	M	92.0	Espalhadas	38.0	2	RS	Doente
3	1920	Jose	18	M	43.0	Inexistentes	38.5	8	MG	Saudavel
5	2301	Ana	22	F	72.0	Inexistentes	58.0	3	RJ	Doente
6	2301	Ana	22	F	72.0	Inexistentes	58.0	3	RJ	Doente
7	2301	Ana	22	F	72.0	Inexistentes	58.0	3	RJ	Doente
8	1322	Marta	19	F	87.0	Espalhadas	39.0	6	AM	Saudavel
9	3027	Paulo	34	M	67.0	Uniformes	38.4	2	GO	Saudavel

```
from sklearn import preprocessing

le = preprocessing.LabelEncoder()
for column in df.columns:
    if df[column].dtypes == 'object':
        df[column] = le.fit_transform(df[column])

print(df)
```

	identificador	nome	idade	sexo	peso	manchas	temperatura	internacoes
0	4201	2	28	1	67	0	38.0	2
1	3217	5	18	0	67	2	39.5	4
2	4039	4	49	1	500	1	38.0	2
3	1920	3	18	1	43	2	38.5	8
4	4340	1	21	0	52	3	38.5	1
5	4340	1	21	0	52	3	38.5	1
6	2301	0	22	0	72	2	58.0	3
7	2301	0	22	0	72	2	58.0	3
8	2301	0	22	0	72	2	58.0	3
9	1322	6	19	0	87	1	39.0	6
10	3027	7	34	1	67	3	38.4	2

Redução de dimensionalidade

Reduzir a dimensão dos dados, geralmente por meio de técnicas de redução de dados como a Análise de Componentes Principais (PCA) ou seleção de características, para simplificar a análise e evitar problemas de alta dimensionalidade.

	sepal length	sepal width	petal length	petal width
0	-0.900681	1.032057	-1.341272	-1.312977
1	-1.143017	-0.124958	-1.341272	-1.312977
2	-1.385353	0.337848	-1.398138	-1.312977
3	-1.506521	0.106445	-1.284407	-1.312977
4	-1.021849	1.263460	-1.341272	-1.312977

PCA
(2 components)



	principal component 1	principal component 2
0	-2.264542	0.505704
1	-2.086426	-0.655405
2	-2.367950	-0.318477
3	-2.304197	-0.575368
4	-2.388777	0.674767

Redução de dimensionalidade

```
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler

features = ['sepal length', 'sepal width', 'petal length', 'petal width']

# Separação características
x = df.loc[:, features].values

# Separação alvo
y = df.loc[:, ['target']].values

# Normaliza as características
x = StandardScaler().fit_transform(x)

pca = PCA(n_components=2)

principalComponents = pca.fit_transform(x)

principalDf = pd.DataFrame(data = principalComponents ,
                           columns = ['principal component 1', 'principal component 2'])
```

	principal component 1	princial component 2
0	-2.264542	0.505704
1	-2.086426	-0.655405
2	-2.367950	-0.318477
3	-2.304197	-0.575368
4	-2.388777	0.674767

Redução de dimensionalidade

```
principalDf = pd.DataFrame(data = principalComponents ,  
                           columns = ['principal component 1', 'principal component 2'])  
  
finalDf = pd.concat([principalDf, df[['target']], axis = 1)
```

principal component 1 principal component 2					target
0	-2.264542	0.505704	0	Iris-setosa	
1	-2.086426	-0.655405	1	Iris-setosa	
2	-2.367950	-0.318477	2	Iris-setosa	
3	-2.304197	-0.575368	3	Iris-setosa	
4	-2.388777	0.674767	4	Iris-setosa	

principalDf

df[['target']]

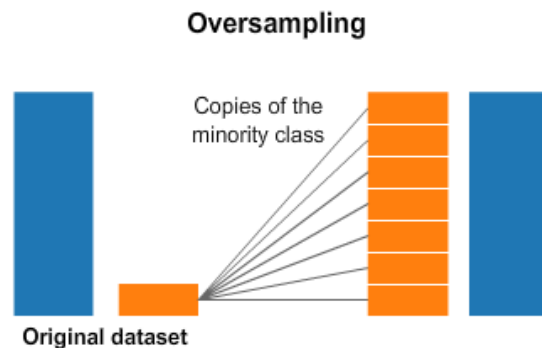
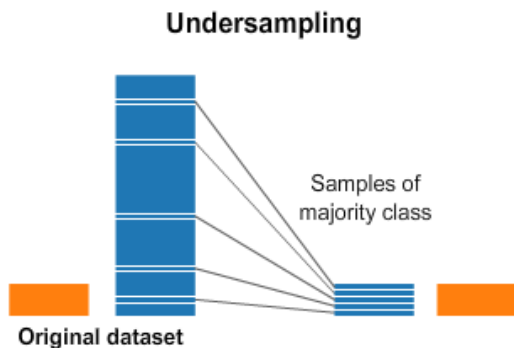
pd.concat(axis = 1)

principal component 1 principal component 2			target
0	-2.264542	0.505704	Iris-setosa
1	-2.086426	-0.655405	Iris-setosa
2	-2.367950	-0.318477	Iris-setosa
3	-2.304197	-0.575368	Iris-setosa
4	-2.388777	0.674767	Iris-setosa

finalDf

Balanceamento dos dados

Refere-se ao processo de equilibrar a distribuição de classes nas amostras de dados. Em muitos casos, os dados podem estar desbalanceados, ou seja, uma classe pode estar sub-representada em comparação com outras. Isso pode levar a modelos de aprendizado de máquina tendenciosos, que favorecem a classe majoritária. Para evitar esse problema, é necessário equilibrar a distribuição de classes, o que pode ser feito por meio de técnicas como *oversampling*, *undersampling* e geração sintética de dados.



Balanceamento dos dados - Exemplo

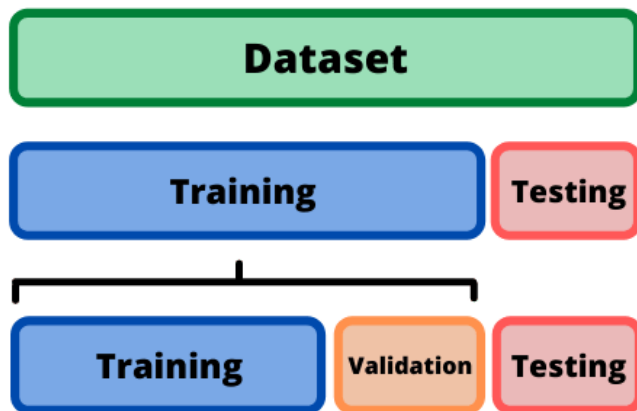
	identificador	nome	idade	sexo	peso	manchas	temperatura	internacoes	estado	diagnostico
4	4340	Claudia	21	F	52	Uniformes	37.6	1	PE	Doente
5	2301	Ana	22	F	72	Inexistentes	58.0	3	RJ	Doente
2	4039	Luiz	49	M	92	Espalhadas	38.0	2	RS	Doente
0	4201	Joao	28	M	79	Concentradas	38.0	2	SP	Doente
8	1322	Marta	19	F	87.0	Espalhadas	39.0	6	AM	Saudavel



	identificador	nome	idade	sexo	peso	manchas	temperatura	internacoes	estado	diagnostico
0	4201	Joao	28	M	72.0	Concentradas	38.0	2	SP	Doente
1	3217	Maria	18	F	67.0	Inexistentes	39.5	4	MG	Saudavel
2	4039	Luiz	49	M	92.0	Espalhadas	38.0	2	RS	Doente
3	1920	Jose	18	M	43.0	Inexistentes	38.5	8	MG	Saudavel
5	2301	Ana	22	F	72.0	Inexistentes	58.0	3	RJ	Doente
6	2301	Ana	22	F	72.0	Inexistentes	58.0	3	RJ	Doente
7	2301	Ana	22	F	72.0	Inexistentes	58.0	3	RJ	Doente
8	1322	Marta	19	F	87.0	Espalhadas	39.0	6	AM	Saudavel
9	3027	Paulo	34	M	67.0	Uniformes	38.4	2	GO	Saudavel

Separação de dados

Refere-se ao processo de dividir os dados em conjuntos de treinamento, validação e teste. O conjunto de treinamento é usado para treinar o modelo, o conjunto de validação é usado para ajustar os hiperparâmetros do modelo e o conjunto de teste é usado para avaliar o desempenho do modelo em dados nunca antes vistos. É importante garantir que os dados sejam divididos de forma aleatória e estratificada, a fim de evitar a introdução de viés nos modelos.



Como o pré-processamento melhora as tomadas de decisões informadas?

**Melhora a
qualidade dos
dados**

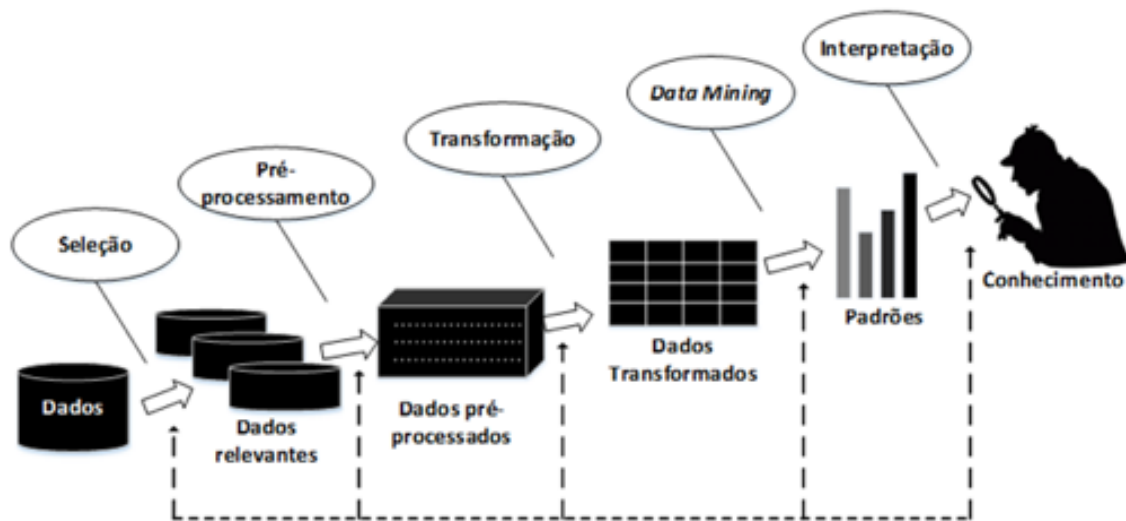
**Simplifica a
análise de
dados**

**Ajuda a
identificar
tendências e
padrões**

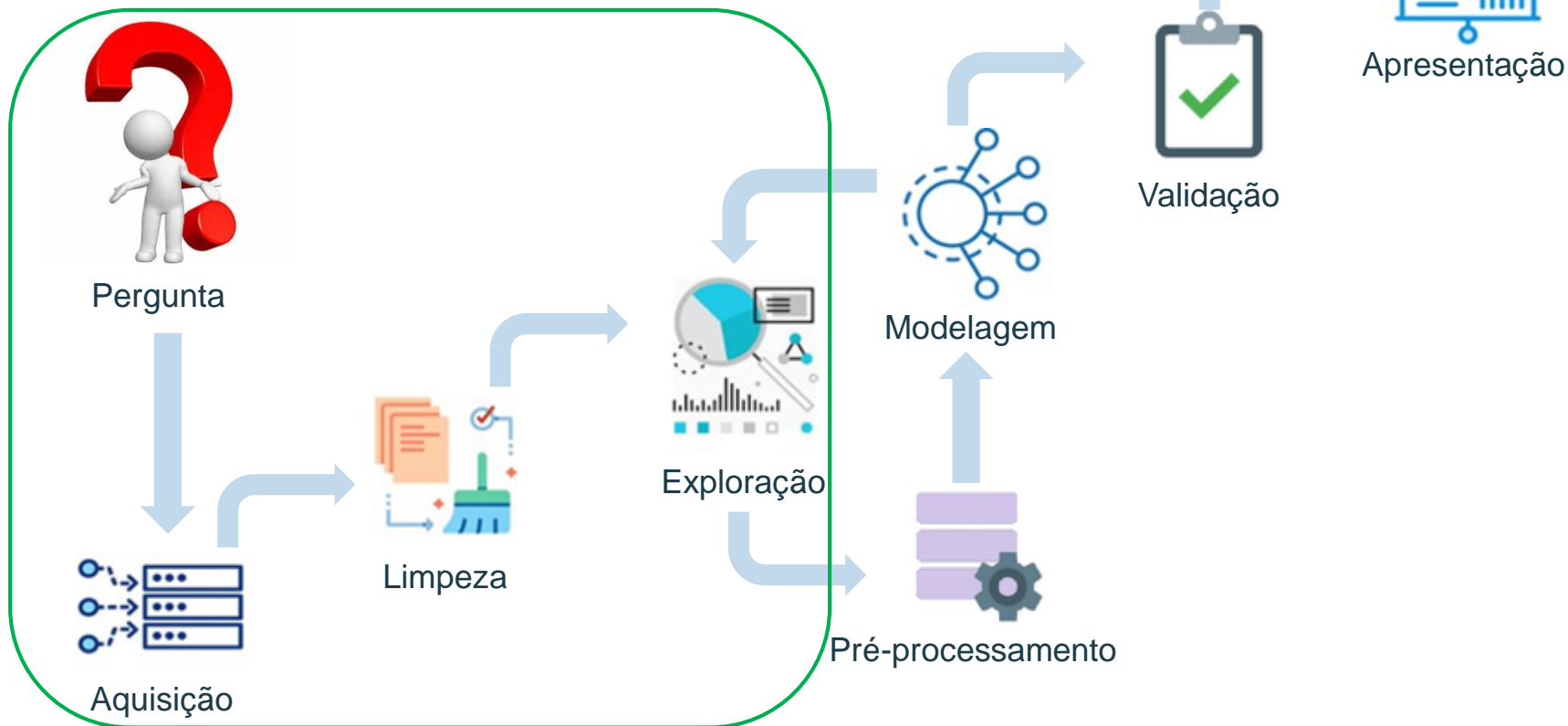
**Reduz o tempo
de análise**

Etapas em processos de aprendizado de máquina

As etapas de citadas de pré-processamento e as mostradas abaixo são, geralmente, realizadas em uma ordem específica, ou seja, cada etapa é derivada da anterior.



Pipeline cientista de dados



Pipeline cientista de dados



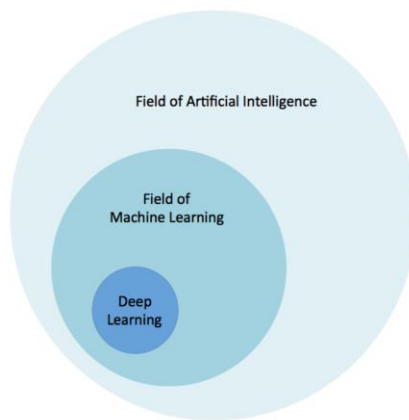
O que é ML?



02

O que é Machine Learning

Machine Learning (Aprendizado de Máquina) é um subcampo da inteligência artificial que envolve o desenvolvimento de algoritmos e técnicas que permitem que os sistemas computacionais aprendam e melhorem automaticamente a partir dos dados, sem serem explicitamente programados para realizar tarefas específicas. (ChatGPT)



Vantagens de utilizar Machine Learning

Capacidade de lidar com grandes volumes de dados

Personalização e recomendações precisas

Automação de tarefas complexas

Detecção de padrões e insights valiosos

Melhoria contínua do desempenho

Previsão e análise de dados

Funcionamento do Machine Learning

Em vez de seguir instruções detalhadas, os algoritmos de Machine Learning são projetados para analisar e interpretar padrões nos dados de entrada, a fim de fazer previsões, tomar decisões ou realizar tarefas específicas. Eles aprendem com exemplos e experiências anteriores, ajustando seus modelos e parâmetros de acordo com os dados disponíveis.

Tipos Machine Learning

- ⚙️ Aprendizagem supervisionada;
- ⚙️ Aprendizagem não supervisionada;
- ⚙️ Aprendizagem semi-supervisionada;
- ⚙️ Aprendizagem por reforço.

Princípio do Machine Learning

Melhorar a realização de uma tarefa a partir da experiência.

- ⚙ Melhorar a realização da tarefa T.
- ⚙ Em relação a uma medida de desempenho P.
- ⚙ Baseada na experiência E.

Aprendizagem Supervisionada



Peixe



Boi



Peixe



Boi



Peixe



Boi



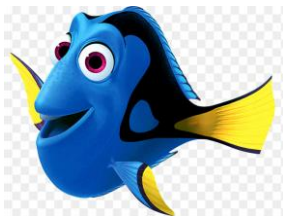
Aprendizagem Supervisionada



Peixe



Boi



Peixe



Boi



Peixe



Boi



Peixe

Aprendizagem Supervisionada



Peixe



Boi



Peixe



Boi



Peixe



Boi



?

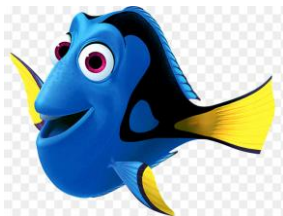
Aprendizagem Supervisionada



Peixe



Boi



Peixe



Boi



Peixe



Boi



Boi

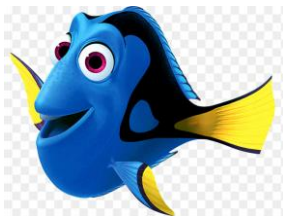
Aprendizagem Supervisionada



Peixe



Boi



Peixe



Boi



Peixe



Boi

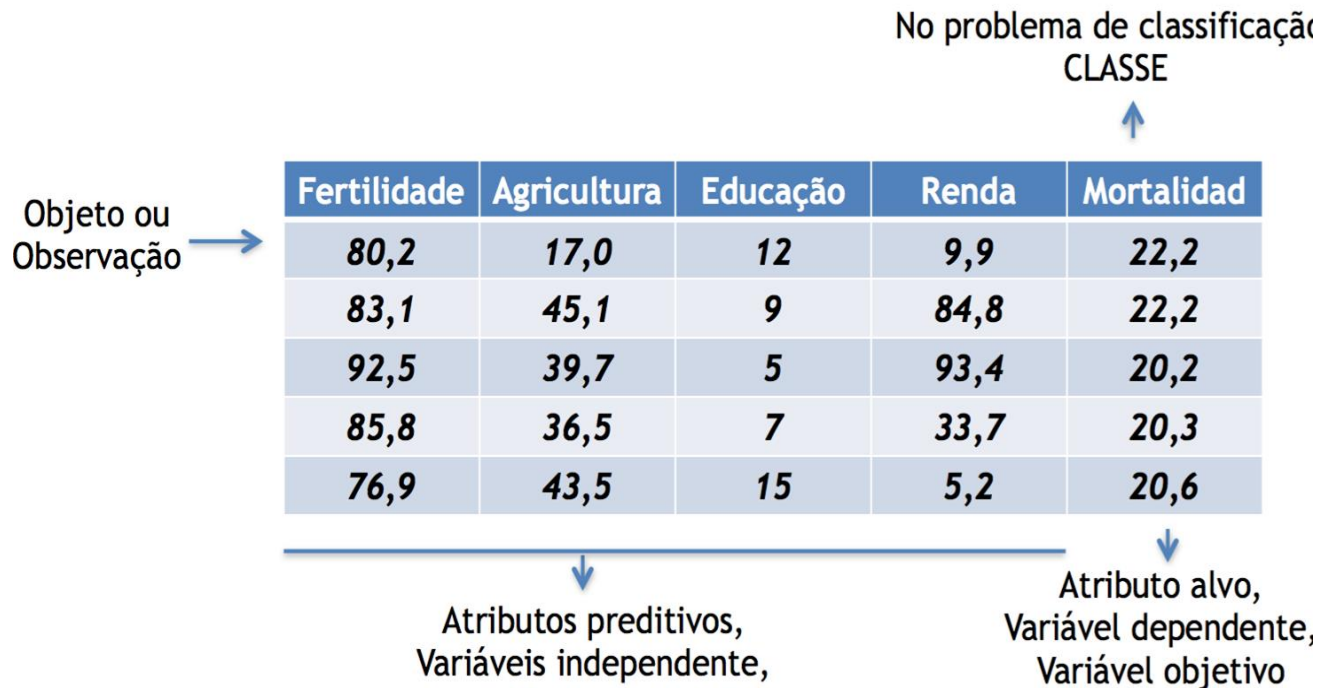


Aprendizagem Supervisionada - Classificação



Tamanho (P)	Largura (P)	Tamanho (S)	Largura (S)	Espécie
5,1	3,5	1,4	0,2	<i>Setosa</i>
4,9	3,0	1,4	0,2	<i>Setosa</i>
7,0	3,2	4,7	1,4	<i>Versicolor</i>
6,4	3,2	4,5	1,5	<i>Versicolor</i>
6,3	3,3	6,0	2,5	<i>Virginica</i>
5,8	2,7	5,1	1,9	<i>Virginica</i>

Aprendizagem Supervisionada - Regressão



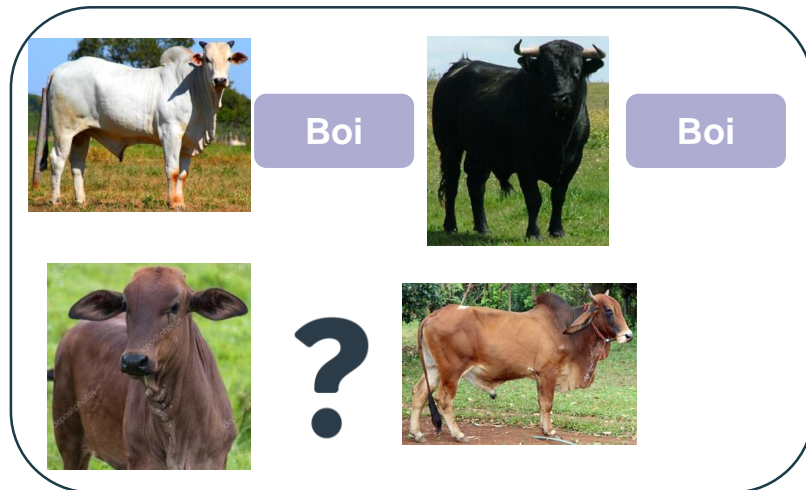
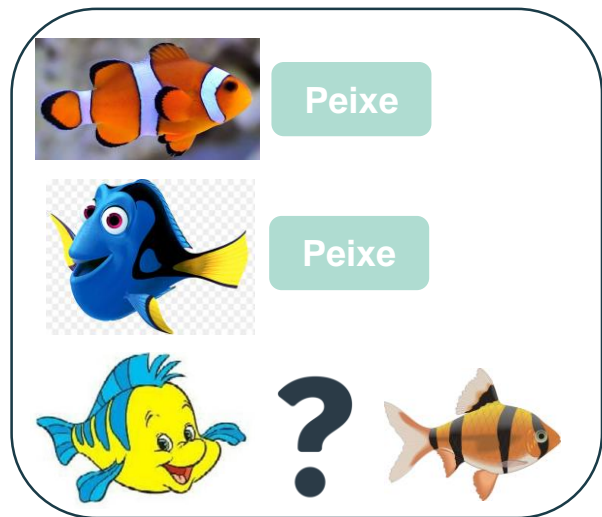
Aprendizagem Não Supervisionada



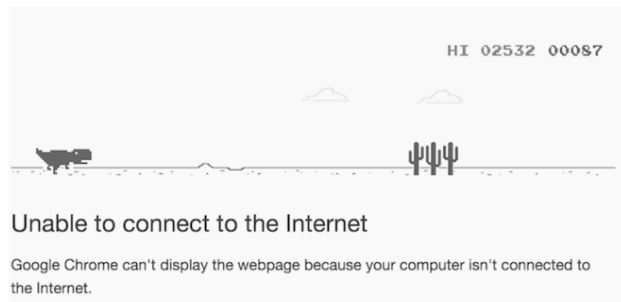
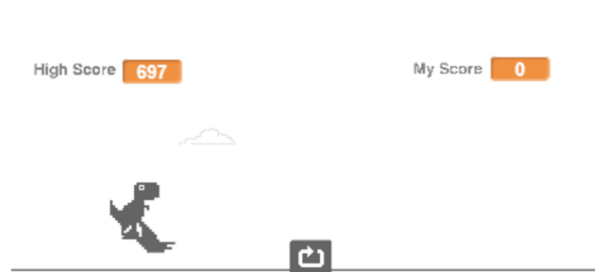
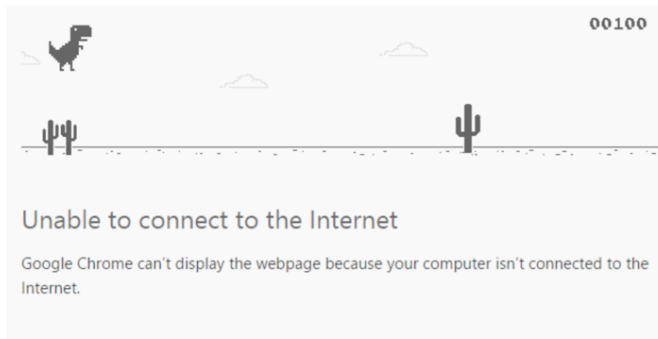
Aprendizagem Não Supervisionada



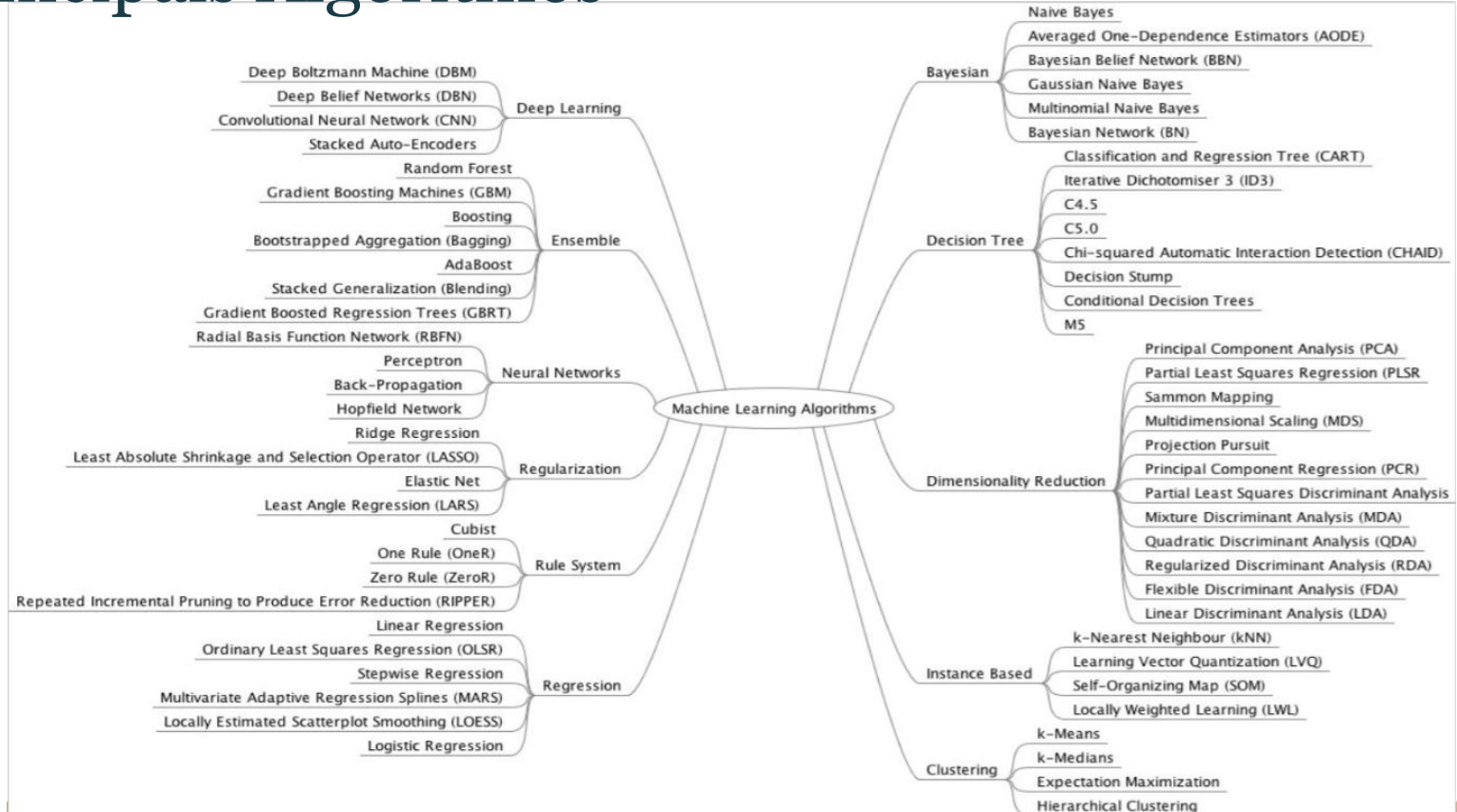
Aprendizagem Semi Supervisionada



Aprendizagem Por Reforço



Principais Algoritmos



Supervisionado

03

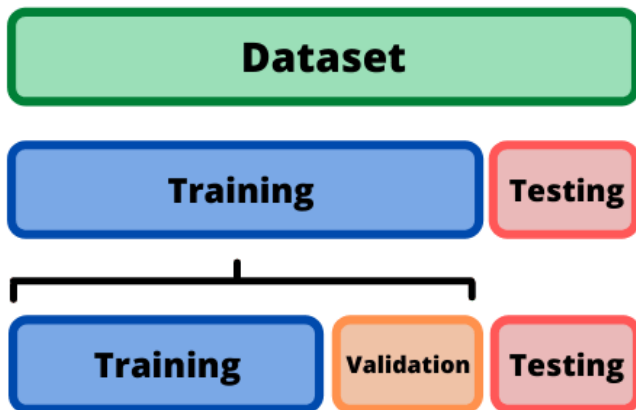
Divisão dos dados

3.1

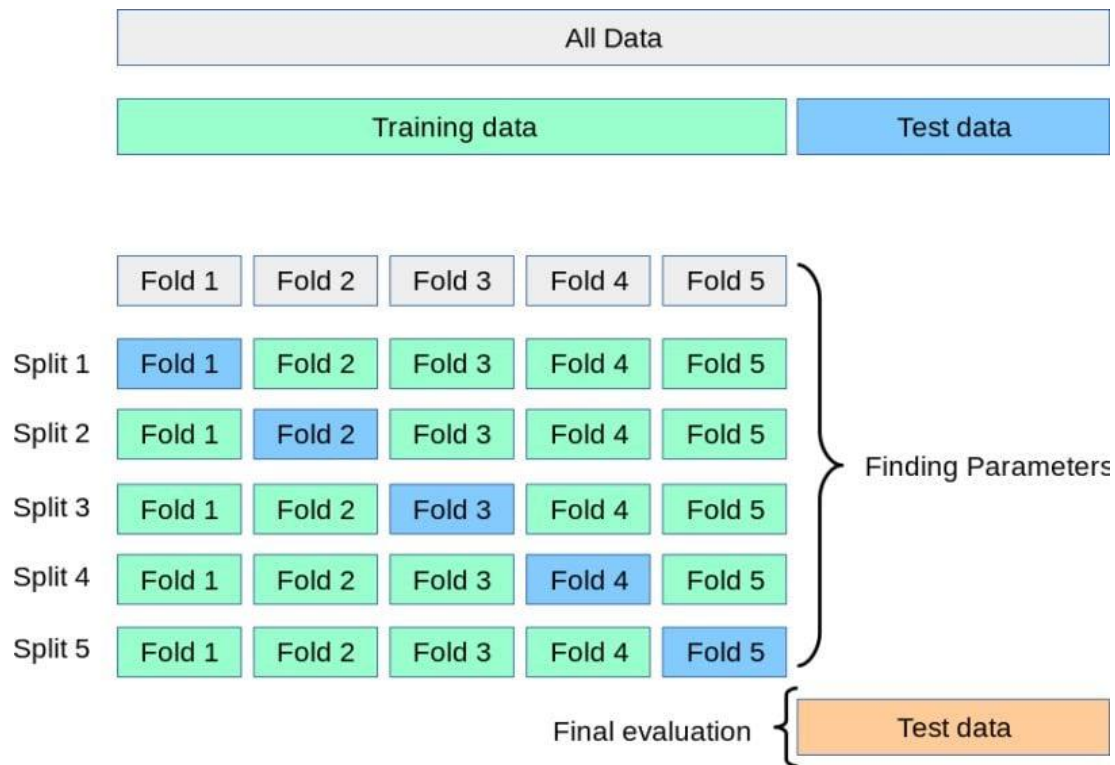
Divisão dos dados

Refere-se ao processo de dividir os dados em conjuntos de treinamento, validação e teste.

- Treinamento é usado para treinar o modelo;
- Validação é usado para ajustar os hiperparâmetros do modelo;
- Teste é usado para avaliar o desempenho do modelo em dados nunca antes vistos.

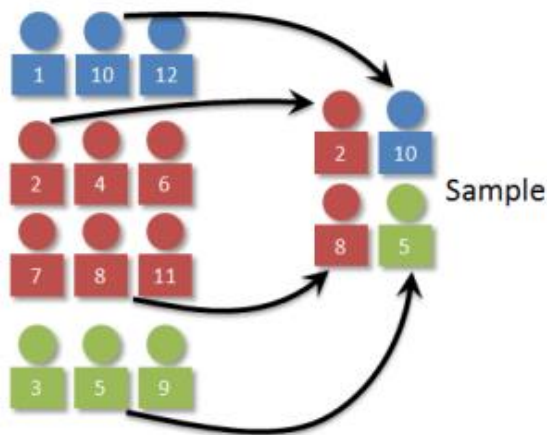


Divisão dos dados



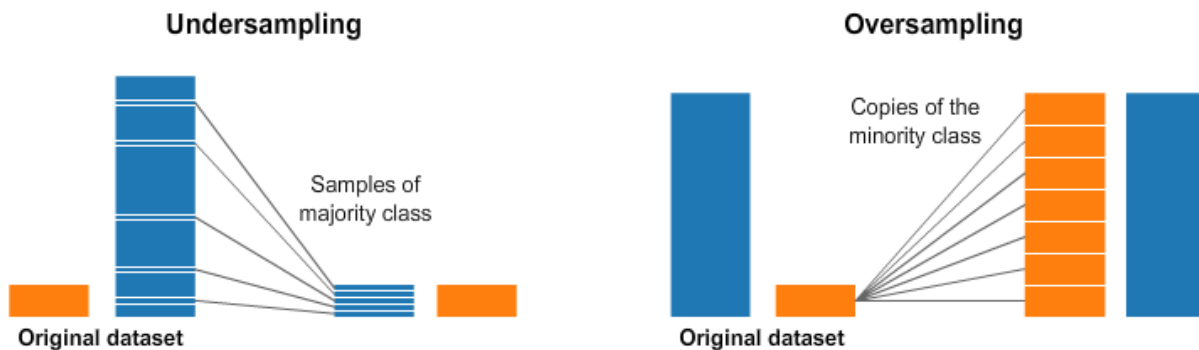
Divisão dos dados - Amostragem

É importante garantir que os dados sejam divididos de forma aleatória e estratificada (mantem o mesmo número de objetos para cada classe, proporcional ao conjunto original), a fim de evitar a introdução de viés nos modelos.



Divisão dos dados - Balanceamento dos dados

Refere-se ao processo de equilibrar a distribuição de classes nas amostras de dados. Em muitos casos, os dados podem estar desbalanceados, ou seja, uma classe pode estar sub-representada em comparação com outras. Isso pode levar a modelos de aprendizado de máquina tendenciosos, que favorecem a classe majoritária. Para evitar esse problema, é necessário equilibrar a distribuição de classes, o que pode ser feito por meio de técnicas como *oversampling*, *undersampling* e geração sintética de dados.



Métricas de avaliação de modelos

3.2

Métricas de avaliação de modelos

Métricas que refletem a qualidade de um modelo, portanto se forem mal escolhidas, será impossível avaliar se o modelo de fato está atendendo os requisitos necessários. Ademais, também são utilizadas para realizar comparação de modelos.

Métricas de avaliação de modelos de classificação

3.2.1

Métricas de avaliação de modelos de classificação

Um modelo de classificação binária tem como objetivo decidir em qual classe uma nova observação pertence dentre duas classes possíveis. Em geral as duas classes, denominadas de positiva (P) e negativa (N), indicam a ocorrência ou não de um determinado evento. Um exemplo seria classificar se um determinado paciente possui uma determinada doença (positivo) ou não (negativo).

A avaliação de um modelo de classificação é feita a partir da comparação entre as classes preditas pelo modelo e as classes verdadeiras de cada exemplo. Todas as métricas de classificação têm como objetivo comum medir quão distante o modelo está da classificação perfeita, porém fazem isto de formas diferentes.

Matriz de confusão

Uma maneira simples de se representar os resultados de um método de classificação de dados é através da chamada matriz de confusão.

<i>Matriz de confusão</i>		Classe predita	
		Positiva	Negativa
Classe original	Positiva	VP	FN
	Negativa	FP	VN

diegomariano.com

Matriz de confusão - Exemplo

Um programa de previsão de chuva foi usado durante 100 dias. Dos 100 dias, o programa disse que iria chover em 55 e que não iria chover nos outros 45 dias. Entretanto, após os 100 dias, percebemos que choveu em 50 e não choveu nos outros 50 dias. Vamos observar a matriz de confusão dos resultados do nosso programa

- VP = 40**: o programa disse que em 40 dos 100 dias **iria chover** e realmente **choveu**.
- FP = 15**: o programa disse que em 15 dos 100 dias **iria chover**, mas **não choveu**.
- FN = 10**: o programa disse que em 10 dos 100 dias **não iria chover**, mas **choveu**.
- VN = 35**: o programa disse que em 35 dos 100 dias **não iria chover** e realmente **não choveu**.

Matriz de confusão		O que o programa disse:	
		Vai chover	Não vai chover
O que aconteceu de verdade:	Choveu	40	10
	Não choveu	15	35

diegomariano.com

Matriz de confusão - Exemplo

Total de predições realizadas em cada classe: $\text{pred}_p = \text{VP} + \text{FP}$ $\text{pred}_n = \text{VN} + \text{FN}$

Total de valores reais: $\text{real}_p = \text{VP} + \text{FN}$ $\text{real}_n = \text{VN} + \text{FP}$

Erros = $\text{FP} + \text{FN}$

Acertos = $\text{VP} + \text{VN}$

Matriz de confusão		Classe predita	
		Positiva	Negativa
Classe original	Positiva	VP	FN
	Negativa	FP	VN

Matriz de confusão		O que o programa disse:	
		Vai chover	Não vai chover
O que aconteceu de verdade:	Choveu	40	10
	Não choveu	15	35

Acurácia

A **acurácia** é considerada uma das métricas mais simples e importantes. Ela avalia simplesmente o percentual de acertos, ou seja, ela pode ser obtida pela razão entre a quantidade de acertos e o total de entradas:

$$acurácia = \frac{VP + VN}{VP + FN + VN + FP}$$

Matriz de confusão		Classe predita	
		Positiva	Negativa
Classe original	Positiva	VP	FN
	Negativa	FP	VN

Matriz de confusão		O que o programa disse:	
		Vai chover	Não vai chover
O que aconteceu de verdade:	Choveu	40	10
	Não choveu	15	35

diegomariano.com

Sensibilidade

Sensibilidade (também conhecida como *recall* ou revocação). Essa métrica avalia a capacidade do método de detectar com sucesso resultados classificados como positivos.

$$sensibilidade = \frac{VP}{VP + FN}$$

Matriz de confusão		Classe predita	
		Positiva	Negativa
Classe original	Positiva	VP	FN
	Negativa	FP	VN

Matriz de confusão		O que o programa disse:	
		Vai chover	Não vai chover
O que aconteceu de verdade:	Choveu	40	10
	Não choveu	15	35

diegomariano.com

Especificidade

Especificidade avalia a capacidade do método de detectar resultados negativos.

$$especificidade = \frac{VN}{VN + FP}$$

Matriz de confusão		Classe predita	
		Positiva	Negativa
Classe original	Positiva	VP	FN
	Negativa	FP	VN

Matriz de confusão		O que o programa disse:	
		Vai chover	Não vai chover
O que aconteceu de verdade:	Choveu	40	10
	Não choveu	15	35

diegomariano.com

Precisão

A **precisão** é uma métrica que avalia a quantidade de verdadeiros positivos sobre a soma de todos os valores positivos.

$$precisão = \frac{VP}{VP + FP}$$

Matriz de confusão		Classe predita	
		Positiva	Negativa
Classe original	Positiva	VP	FN
	Negativa	FP	VN

Matriz de confusão		O que o programa disse:	
		Vai chover	Não vai chover
O que aconteceu de verdade:	Choveu	40	10
	Não choveu	15	35

diegomariano.com

F1-score

Média harmônica calculada com base na precisão e na sensibilidade.

$$f1 = 2 * \frac{\text{precisão} * \text{sensibilidade}}{\text{precisão} + \text{sensibilidade}}$$

Matriz de confusão		Classe predita	
		Positiva	Negativa
Classe original	Positiva	VP	FN
	Negativa	FP	VN

Matriz de confusão		O que o programa disse:	
		Vai chover	Não vai chover
O que aconteceu de verdade:	Choveu	40	10
	Não choveu	15	35

diegomariano.com

Métricas de avaliação de modelos de regressão

3.2.2

R² score

Medida estatística que representa a proporção da variância para uma variável dependente que é explicada por uma variável independente em um modelo de regressão. Enquanto a correlação explica a força da relação entre uma variável independente e uma variável dependente, o R-quadrado explica até que ponto a variância de uma variável explica a variância da segunda variável. Portanto, se o R² de um modelo for 0,50, aproximadamente metade da variação observada pode ser explicada pelas entradas do modelo, logo R² Score, normalmente, está entre 0 e 1, quanto mais próximo de 1, melhor o ajuste da regressão.

$$R^2 = 1 - \frac{\text{Unexplained Variation}}{\text{Total Variation}}$$

Erro médio (ME)

Média da diferença do realizado e do previsto.

Previsto	Realizado	Dif.
3,34	3,00	-0,34
4,18	4,00	-0,18
3,00	3,00	0
2,99	3,00	0,01
4,51	4,50	-0,01
5,18	4,00	-1,18
8,18	4,50	-3,68

$$ME = \sum_{i=1}^N \frac{p_i - t_i}{n}$$

$$ME = \frac{-5,38}{7} = -0,76$$

Erro médio absoluto (MAE)

Média da diferença absoluta do realizado e do previsto.

Previsto	Realizado	Dif. Absoluta
3,34	3,00	0,34
4,18	4,00	0,18
3,00	3,00	0
2,99	3,00	0,01
4,51	4,50	0,01
5,18	4,00	1,18
8,18	4,50	3,68
		5,4

$$MAE = \sum_{l=1}^N \frac{|p_i - t_i|}{n}$$

$$MAE = \frac{5,4}{7} = 0,77$$

Erro médio quadrático (RMSE)

O desvio da amostra da diferença entre o previsto e o realizado.

Previsto	Realizado	Dif. ao Quad.
3,34	3,00	0,1156
4,18	4,00	0,0324
3,00	3,00	0
2,99	3,00	1E-04
4,51	4,50	1E-04
5,18	4,00	1,3924
8,18	4,50	13,5424

$$RMSE = \sqrt{\frac{\sum_{i=1}^N (p_i - t_i)^2}{N}}$$

$$RMSE = \sqrt{\frac{15,083}{7}}$$

$$RMSE = 1,46$$

Erro de porcentagem absoluta média (MAPE)

Diferença absoluta percentual de erro.

Previsto	Realizado	Erro % abs.
3,34	3,00	0,1133333
4,18	4,00	0,045
3,00	3,00	0
2,99	3,00	0,0033333
4,51	4,50	0,0022222
5,18	4,00	0,295
8,18	4,50	0,8177778

$$MAPE = \frac{\sum_{i=1}^N \frac{|p_i - t_i|}{|t_i|}}{N} \times 100$$

$$MAPE = \frac{1,2766667}{7} \times 100$$

$$MAPE = 18\%$$

Regressão

3.3

Regressão

Para o funcionamento do modelo de regressão serão fornecidas informações na forma de variáveis atributos e o modelo estimará o valor da variável resposta usando dados de referência durante o treinamento. Lembrando que, para o caso da Regressão, o tipo de resposta esperado na saída será um valor contínuo representativo de acordo com a variável resposta. Especificamente quando se trata de **Regressões Lineares**, a inferência feita sobre a relação entre as variáveis é que pode ser descrita por uma **equação de reta**.

Regressão linear

Na regressão linear simples, tem-se um conjunto de dados formado por um único atributo X e a variável resposta Y . O modelo vai procurar estabelecer a melhor equação de reta que descreva o conjunto de dados, ou seja define a equação como:

$$Y \approx \beta_0 + \beta_1 X$$

Regressão linear

Uma forma para estimar os valores de β_0 e β_1 é a partir dos **valores médios** para X e Y, onde β_0 é o **coeficiente independente** (onde a equação de reta vai cortar o eixo Y no gráfico) e o β_1 é o **coeficiente angular** desta reta (indicando a inclinação da reta a ser ajustada). Assim os valores dos coeficientes também serão um valor médio da forma β_0 e β_1 , dados pelas seguintes equações:

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2} = \frac{\sigma_{xy}}{\sigma_{xx}} = \frac{Covar(x, y)}{Var(x)}$$
$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

- Covariância indica o grau de interdependência entre duas variáveis;
- Variância é uma medida de o quão disperso estão os dados, ou seja o quão distante está cada valor desse conjunto do valor médio.

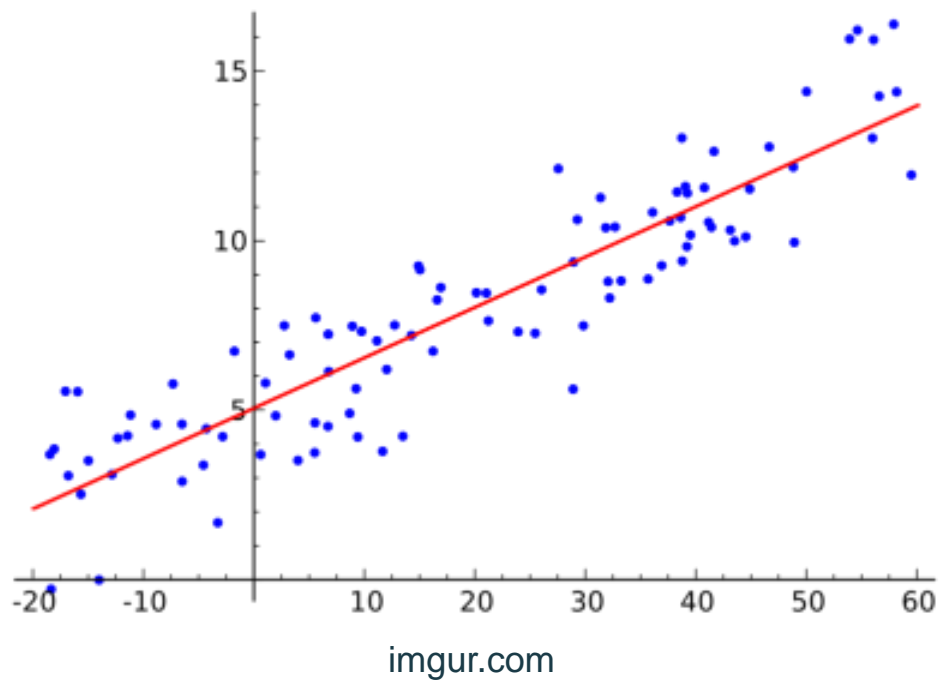
Implementação

```
def linear_regression(x, y):  
    # Definir as médias  
    mean_x, mean_y = np.mean(x), np.mean(y)  
  
    # Calcular a covariância e variância  
    S_xy = 0  
    S_xx = 0  
  
    # Laço para o somatório  
    for i in range(0, len(x)):  
        # termo covariância  
        S_xy += (x[i] - mean_x)*(y[i] - mean_y)  
        # termo variância  
        S_xx += (x[i] - mean_x)**2  
  
    # Calcular os coeficientes de regressão  
    beta_1 = S_xy / S_xx  
    beta_0 = mean_y - beta_1*mean_x  
  
    # Retorna os coeficientes  
    return beta_0, beta_1
```

$$\hat{\beta}_1 = \frac{\sum_{i=1}^n (x_i - \bar{x})(y_i - \bar{y})}{\sum_{i=1}^n (x_i - \bar{x})^2} = \frac{\sigma_{xy}}{\sigma_{xx}} = \frac{Covar(x, y)}{Var(x)}$$
$$\hat{\beta}_0 = \bar{y} - \hat{\beta}_1 \bar{x}$$

```
from sklearn.linear_model import LinearRegression  
  
# Instanciar o modelo  
model = LinearRegression()  
  
# Treinamento do Modelo  
model.fit(X_train, y_train)  
  
# Gerar novas previsões  
y_pred = model.predict(X_test)
```

Regressão linear



Algoritmo KNN

3.4

Vizinhos mais próximos

- Classifica um novo objeto com base nos exemplos do conjunto de treinamento que são próximos a ele;
- Pode ser utilizado tanto para classificação quanto para regressão;
- Tem variações definidas (principalmente) pelo número de vizinhos considerados.

1- Vizinho mais próximo

- 1-NN, 1-Nearest neighbour;
- Calcula distância entre cada 2 pontos;
- A métrica mais usual é a distância Euclidiana.

$$d_{L2}(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$

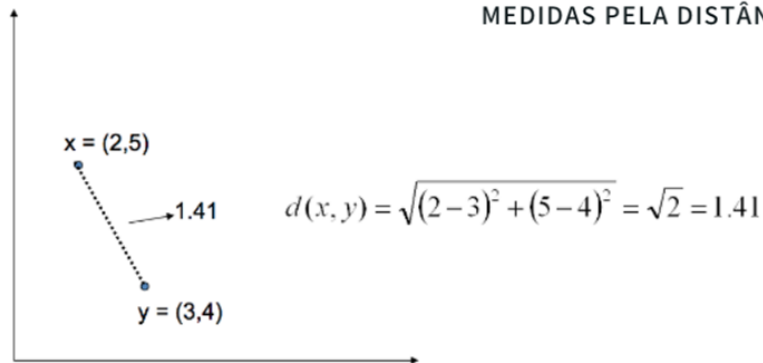
Exemplo - Distância Euclidiana

$$d(x, y) = \sqrt{\sum_{i=1}^n (x_i - y_i)^2}$$



PODEM SER USADAS QUAISQUER MÉTRICAS QUE REPRESENTEM SIMILARIDADE ENTRE DOIS OBJETOS, COMO CORRELAÇÃO, POR EXEMPLO!!

DISTÂNCIA ENTRE STRINGS PODEM SER MEDIDAS PELA DISTÂNCIA DE HAMMING!!



- CADA DIMENSÃO REPRESENTA UM ATRIBUTO
- X E Y REPRESENTAM INSTÂNCIAS (OBJETOS)

Distância entre strings

Hamming

Strings de mesmo tamanho

- 10**111**01 and 10**010**01 is 2.
- 2**1738**96 and 2**2337**96 is 3.

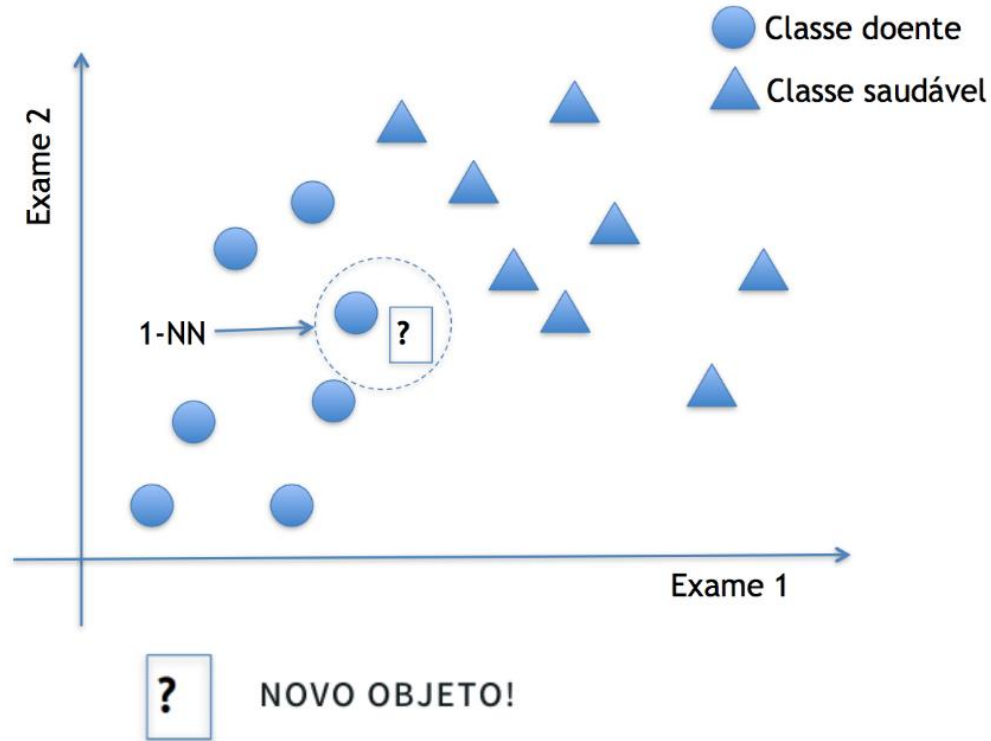
Levenshtein

Strings de tamanhos distintos

H		O	N	D	A	
H	Y	U	N	D	A	I

H	O		N	D	A	
H	Y	U	N	D	A	I

1-NN



Algoritmo

Entrada: Conjunto de treinamento (D); objeto teste (z); conjunto de classes (L)

Saída: A classe de z

PARA CADA objeto y que pertence a D faça:

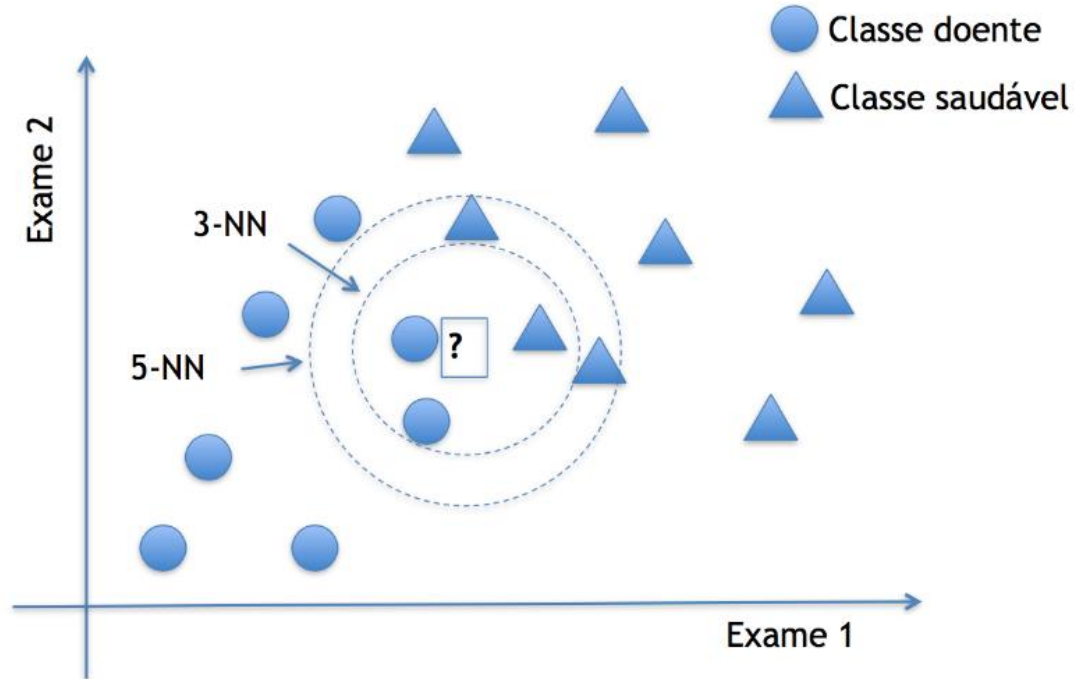
 Compute a distância entre z e y

FIM

Fixe k , o número de vizinhos mais próximos na população (z) a ser considerado

A classe de saída do objeto z será a moda, média ou mediana dos rótulos dos k vizinhos do conjunto selecionado como mais próximo

K-NN



Resumo

Em vez de 1 vizinho mais próximo, os k objetos do conjunto de treinamento mais próximo do ponto de teste X ;

Classificação:

- O objeto é classificado na classe mais votada: moda;

Regressão:

- Minimizar o erro quadrático: média;
- Minimizar o desvio absoluto: Mediana

Vantagens e Desvantagens

PRÓS:

- O algoritmo é simples e fácil de implementar;
- Não há necessidade de ajustar vários parâmetros;
- Treinamento rápido;
- Ele pode ser usado para classificação, regressão.

CONS:

- A classificação da base de teste é mais lenta e mais custosa em termos de tempo e memória;
- KNN também não é adequado para grandes dados dimensionais

Algoritmo Árvore de decisão

3.5

Árvore de Decisão

O principal objetivo do algoritmo árvore de decisão é encontrar o atributo ou variável independente que melhor realiza a divisão dos dados.

Uma árvore de decisão, utiliza a técnica de dividir para conquistar:

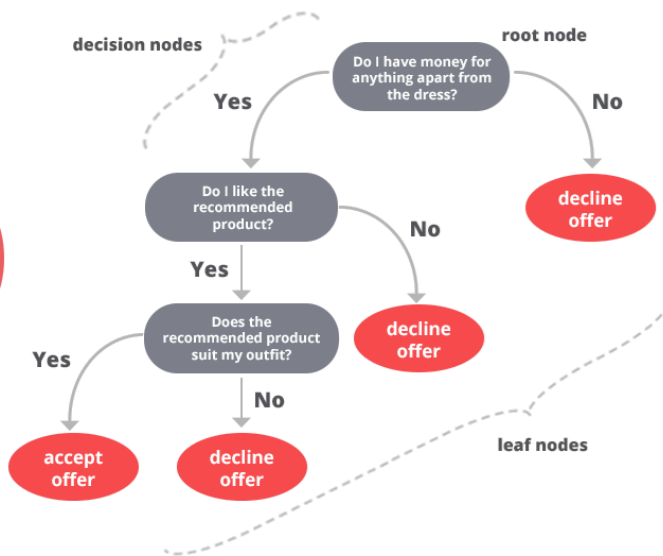
- Um problema complexo é decomposto em subproblemas mais simples;
- Recursivamente a mesma estratégia é aplicada em cada subproblema.



Árvore de Decisão

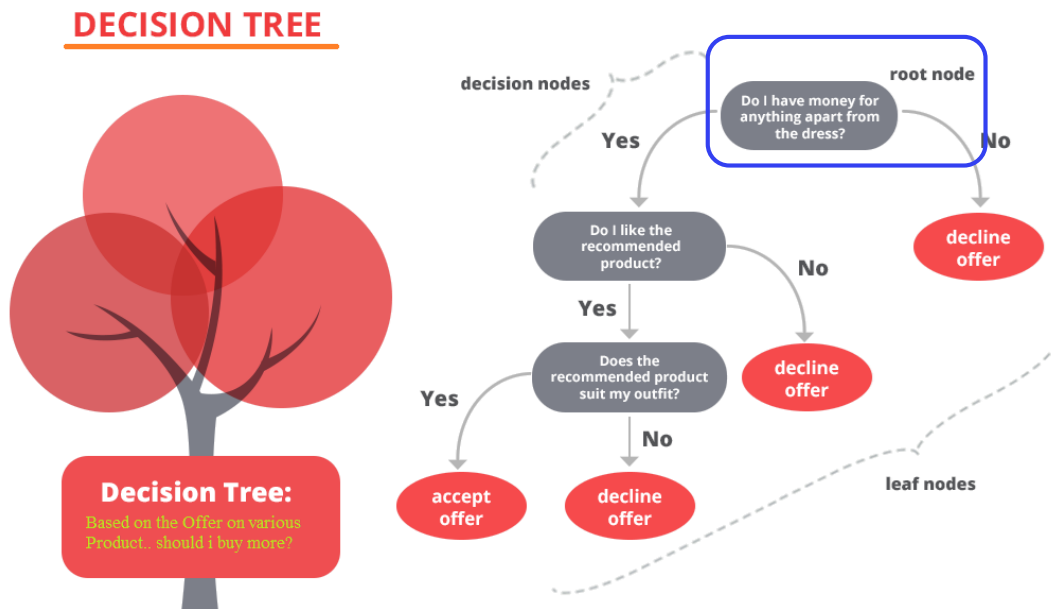
Podemos assimilar o funcionamento deste algoritmo como um fluxograma. As árvores de decisão consistem em nós de decisão interconectados em uma hierarquia, incluindo o nó raiz e os nós folha.

DECISION TREE



Árvore de Decisão

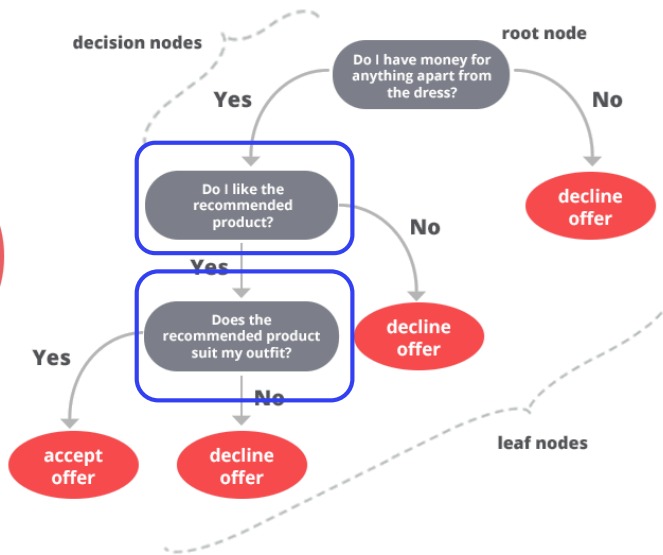
O nó raiz é o mais importante, pois separa o conjunto de dados em partições mais puras.



Árvore de Decisão

Os nós de decisão são atributos da base de dados que levam às ramificações da árvore.

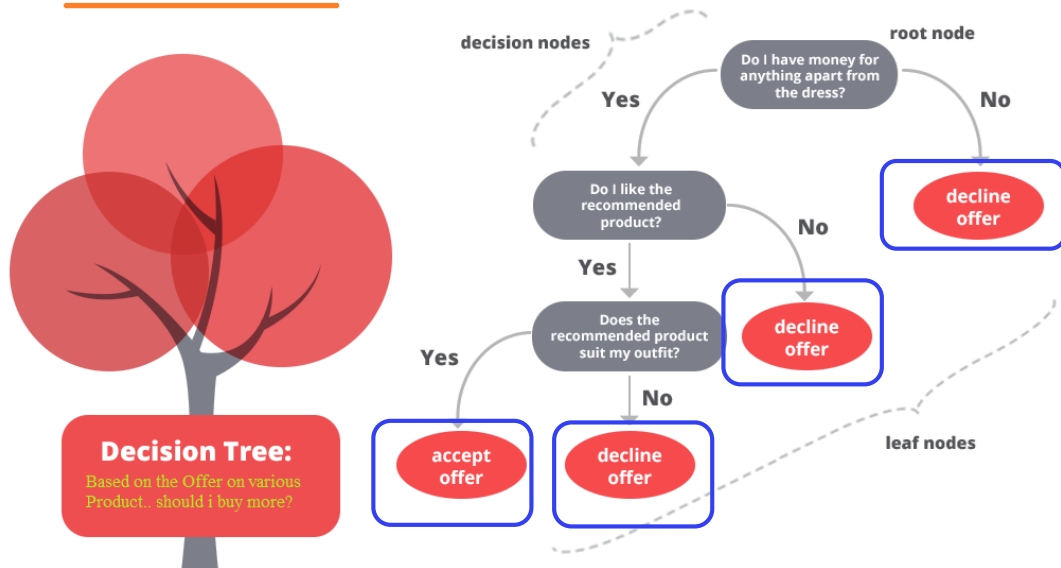
DECISION TREE



Árvore de Decisão

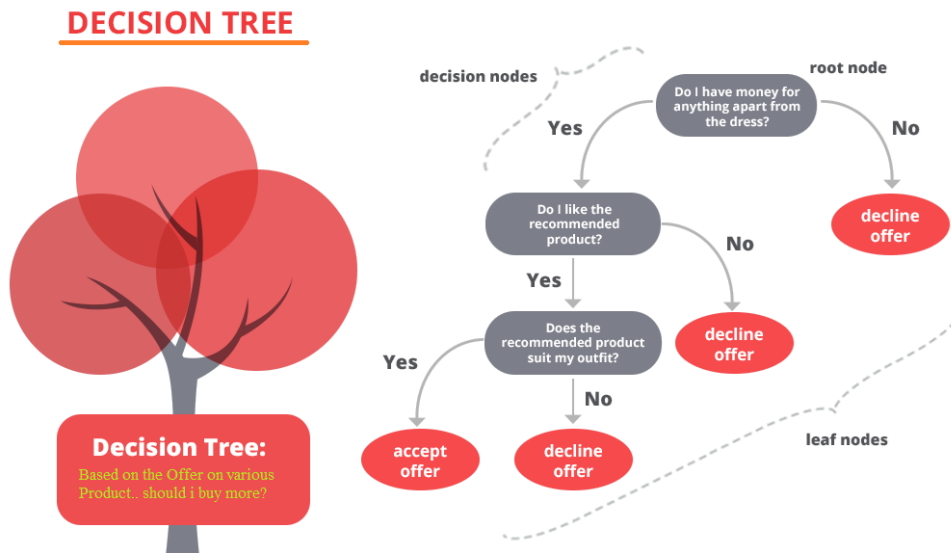
Os nós folha representam as respostas finais do algoritmo, como classes ou valores dependendo do tipo de problema (classificação ou regressão).

DECISION TREE



Árvore de Decisão

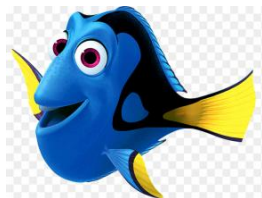
Em cada ligação entre os nós, o algoritmo pergunta acerca de uma condição (“if-else”) gerada fundamentada no conhecimento absorvido dos dados pelo algoritmo e realiza a divisão dos dados com base nas regras ou condições. Por exemplo, se estamos lidando com uma base de pessoas, uma condição que poderia ser gerada seria considerar pessoas com idade maior de 18 anos.



Não Supervisionado

04

Aprendizagem Não Supervisionada

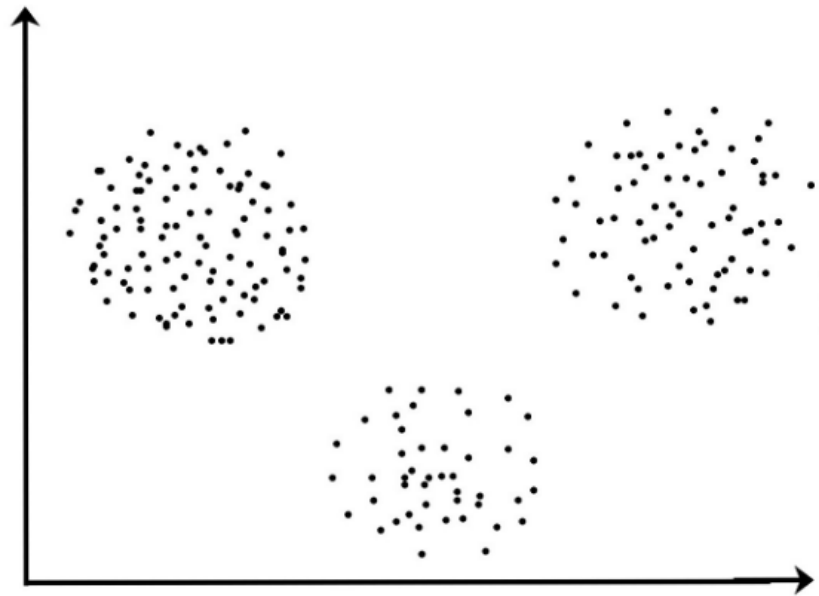


Aprendizagem Não Supervisionada

Os objetos pertencentes a cada cluster compartilham alguma característica.

Cluster: uma coleção de objetos próximos ou que satisfazem alguma relação espacial.

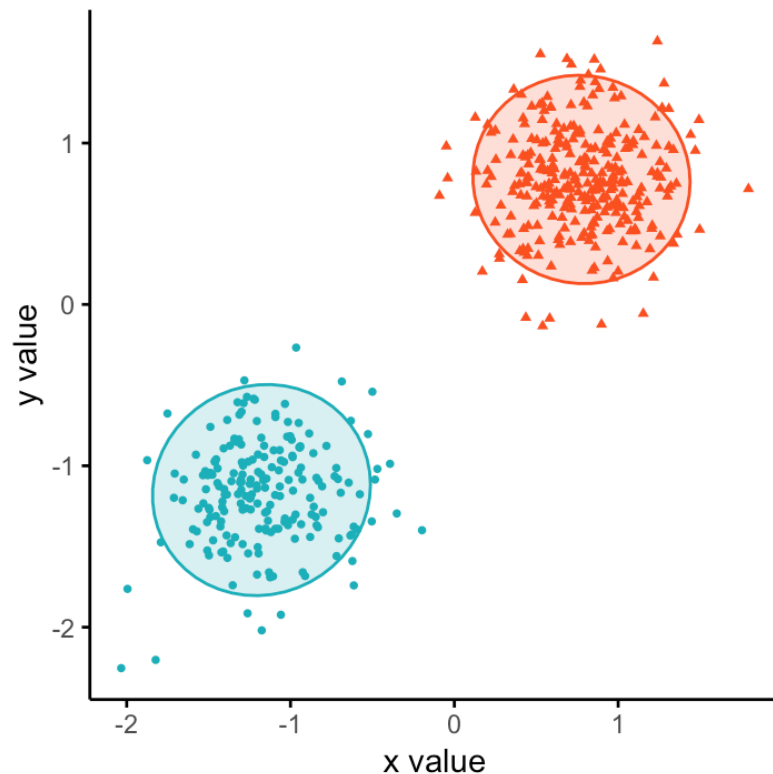
Cluster bem separado: conjunto de pontos tal que qualquer ponto em um determinado cluster está mais próximo (ou é mais similar) dos pontos de seu cluster do que de qualquer ponto de outros clusters.



Aprendizagem Não Supervisionada

Cluster baseado em centro: conjunto de pontos tal que qualquer ponto em um dado cluster está mais próximo (ou é mais similar) ao centro do cluster do que ao centro de qualquer outro cluster.

O centro de um cluster pode ser um centróide, como a média aritmética dos pontos do cluster.



Principais Características

Não Supervisionado

4.1

Matriz de Similaridade e Dissimilaridade

Representa a similaridade ou a dissimilaridade entre cada par de objetos.

Cada elemento da matriz $S_{n \times n}$, s_{ij} , é dado pela distância, $d(x_i, x_j)$, ou pela similaridade, $s(x_i, x_j)$, entre os objetos x_i e x_j .

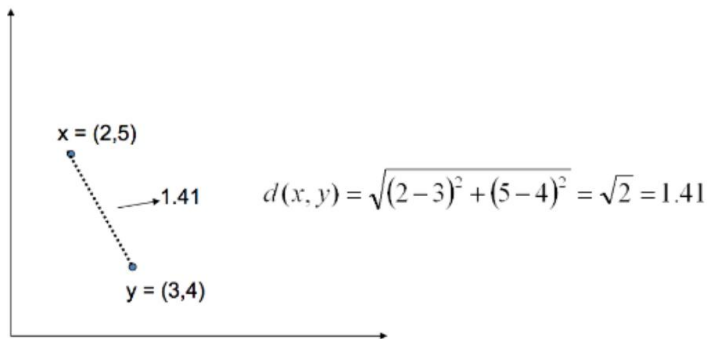
Objeto	Coordenada X1	Coordenada X2
A	1	1
B	1.5	1.5
C	5	5
D	3	4
E	4	4
F	3	3.5

Dist	A	B	C	D	E	F
A	0.00	0.71	5.66	3.61	4.24	3.20
B	0.71	0.00	4.95	2.92	3.54	2.50
C	5.66	4.95	0.00	2.24	1.41	2.50
D	3.61	2.92	2.24	0.00	1.00	0.50
E	4.24	3.54	1.41	1.00	0.00	1.12
F	3.20	2.50	2.50	0.50	1.12	0.00

Proximidade

Uma das medidas de dissimilaridade mais comum para objetos cujos atributos são todos contínuos é a distância euclidiana;

Uma das medidas de similaridade mais usadas é a correlação.



```
corr = df.corr()  
corr.style.background_gradient(cmap='coolwarm')
```

	identificador	idade	peso	temperatura	internacoes
identificador	1	0.48618	0.0620799	-0.319808	-0.817134
idade	0.48618	1	0.560835	-0.191917	-0.512349
peso	0.0620799	0.560835	1	0.0593191	-0.28261
temperatura	-0.319808	-0.191917	0.0593191	1	-0.035277
internacoes	-0.817134	-0.512349	-0.28261	-0.035277	1

Validação

Determina se os clusters são significativos, ou seja, se a solução é representativa para o conjunto de dados analisado.

Determina o número apropriado de clusters para um conjunto de dados, que em geral não é conhecido previamente.

Alguns fatores têm grande influência no desempenho das técnicas de agrupamento:

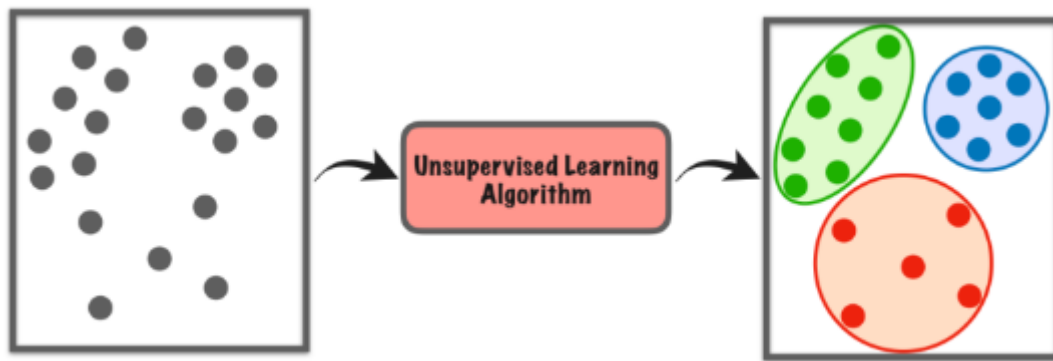
- Estrutura dos clusters (forma, tamanho, número de clusters);
- Presença de *outliers*;
- Grau de sobreposição dos clusters;
- Escolha da medida de similaridade.

K-médias (K-Means)

4.2

K-Means

O K-Means é um dos principais algoritmos de clusterização utilizados em *Machine Learning* pois é um modelo de simples **interpretabilidade** e apresenta uma boa **eficiência computacional**, ou seja, consegue modelar com facilidade utilizando o *K-Means*.



towardsdatascience

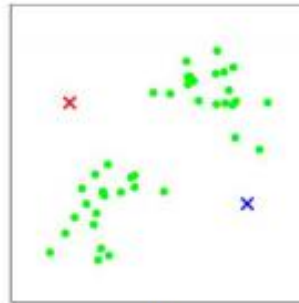
K-Means - Algoritmo

- 1) **Definição dos Centroides:** Supondo que deve-se separar os dados em K grupos (também chamados de *clusters*), para inicializar o modelo define-se K centroides iniciais aleatoriamente, que serão utilizados para os cálculos do modelo;
- 2) **Agrupamentos dos K grupos:** Dado os K grupos utilizados no modelo, cada uma das observações será associada ao **centroide mais próximo** utilizando de cálculos de distância (no caso, distância euclidiana);
- 3) **Reposicionamento dos Centroides:** Separados as observações em cada um dos K grupos, a partir das observações recalcula-se a **posição dos centroides** como a média da posição das observações dentro de determinado grupo (isto fazendo referência ao nome de *K-Means*);
- 4) **Processo Iterativo:** Os passos 2 e 3 serão repetidos até que o modelo considere que não houve mais alterações na **posição dos centroides**, levando em consideração uma margem de erro para as variações do posicionamento do centroide, ou mesmo quando o número máximo de iterações é atingido.

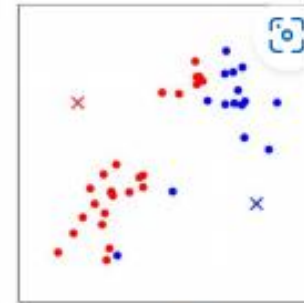
K-Means - Funcionamento



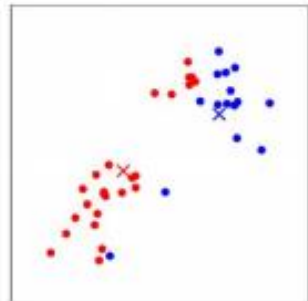
(a)



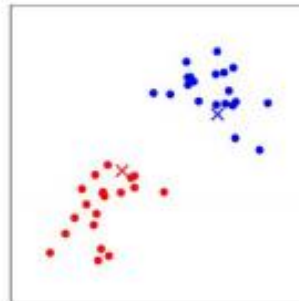
(b)



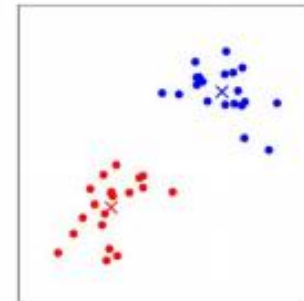
(c)



(d)



(e)



(f)

Método do cotovelo

4.2.1

Método do cotovelo

A ideia é executar o KMeans para várias quantidades diferentes de clusters e dizer qual dessas quantidades é o **número ótimo de clusters**.

O que geralmente acontece ao aumentar a quantidade de clusters no KMeans é que as diferenças entre clusters se tornam muito pequenas, e as diferenças das observações intra-clusters vão aumentando. Então é preciso achar um equilíbrio em que as observações que formam cada agrupamento sejam o mais homogêneas possível e que os agrupamentos formados sejam o mais diferentes um dos outros.

Método do cotovelo

Como o KMeans calcula a distância das observações até o centro do agrupamento que ela pertence, o ideal é que essa distância seja a menor viável. Matematicamente falando, nós estamos buscando uma quantidade de agrupamentos em que a soma dos quadrados intra-clusters (ou do inglês *within-clusters sum-of-squares* (wcss)) seja a menor possível, sendo zero o resultado ótimo.

Método do cotovelo

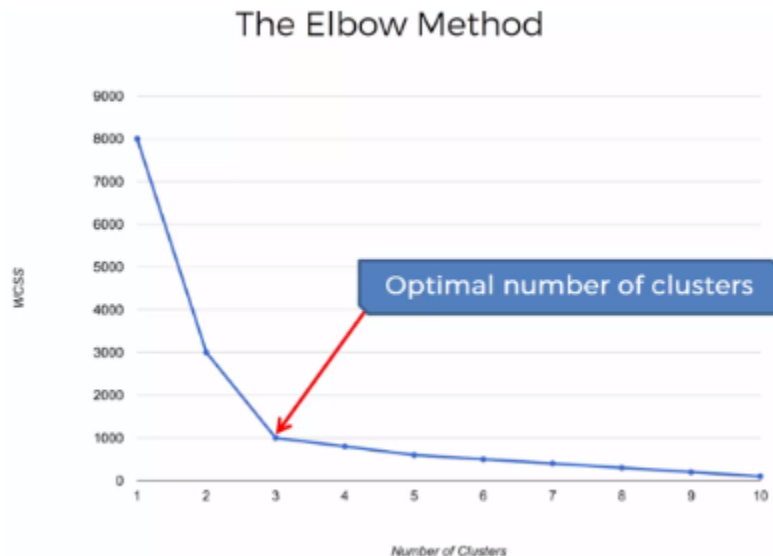
Por exemplo, quando $K = 3$, a distância de soma (p, c) é a soma da distância dos pontos em um cluster do centróide.

$$WCSS = \sum_{P_i \text{ in Cluster 1}} \text{distance}(P_i, C_1)^2 + \sum_{P_i \text{ in Cluster 2}} \text{distance}(P_i, C_2)^2 + \sum_{P_i \text{ in Cluster 3}} \text{distance}(P_i, C_3)^2$$

↳

Método do cotovelo

Na representação abaixo podemos ver que após 3 não há diminuição significativa no WCSS, então 3 é o melhor aqui. Portanto, há um formato de cotovelo que se forma e geralmente é uma boa ideia escolher o número onde esse cotovelo é formado. Muitas vezes o gráfico não seria tão intuitivo, mas com a prática fica mais fácil.



Método de silhueta

4.2.2

Método de silhueta

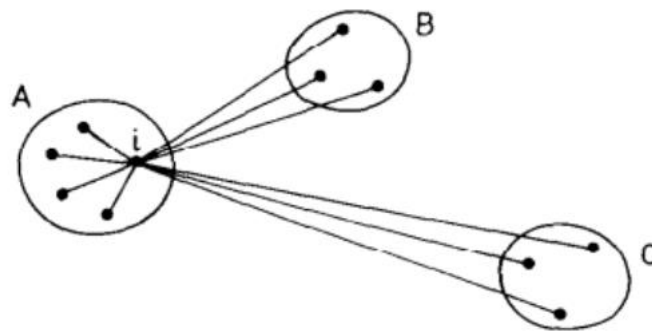
Silhueta refere-se a um método estatístico utilizado para interpretação e validação de consistência dentro dos clusters formados. O valor da silhueta (também chamado de coeficiente de silhueta) é uma medida de quão semelhante um objeto é ao seu próprio cluster (coesão), em comparação com outros clusters (separação). Este valor pode ser calculado com qualquer métrica de distância, como a euclidiana ou a de Manhattan.

Os coeficientes de silhueta variam de -1 a +1. Os valores próximos a +1 indicam que a amostra está longe dos clusters vizinhos, ou seja, indica que o objeto está no grupo que deveria se encontrar e que não deveria ser agrupado aos grupos vizinhos. Um valor 0 indica que a amostra está dentro ou muito perto do limite de decisão entre dois clusters vizinhos. Valores negativos indicam que essas amostras podem ter sido atribuídas ao cluster errado.

Método de silhueta

1. Para cada ponto i , calcula-se a distância intra-cluster, denominada de $a(i)$. Esta distância é calculada através da distância média de i para todos os outros pontos que foram agrupados dentro deste mesmo *cluster*. Pode-se interpretar $a(i)$ como quão bem i foi atribuído ao seu grupo (quanto menor o valor, melhor a atribuição). Desta forma, a média de dissimilaridade do ponto i para um grupo c é definida como a média das distâncias de i para todos os pontos em c .

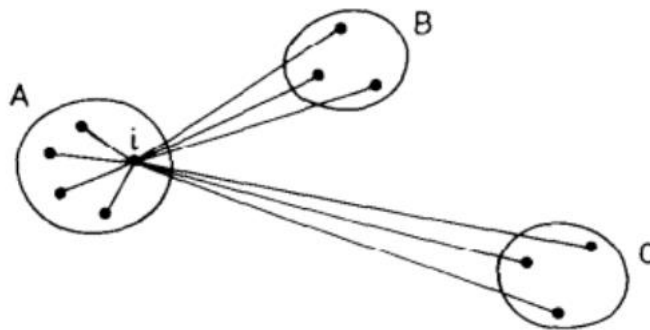
$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$



Método de silhueta

2. Para cada ponto i , calcula-se a distância inter-cluster, denominada por $b(i)$. Esta distância é calculada através da distância média de i para todos os outros pontos que foram agrupados em *clusters* distintos do ponto i . Assim, o valor de $b(i)$ será denominado pela menor distância média de i para todos os pontos pertencentes a outros grupos, do qual i não é um membro. O grupo com essa menor dissimilaridade média é o "grupo vizinho" de i .

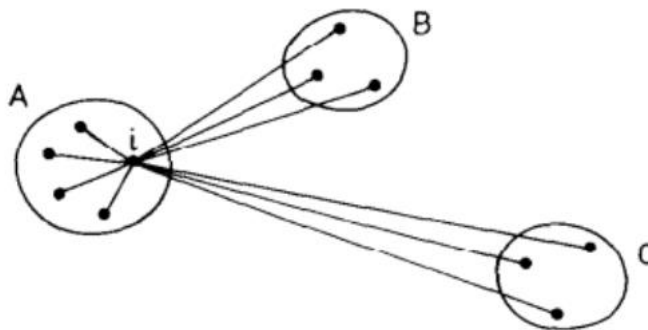
$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$



Método de silhueta

3. Para cada ponto i , o coeficiente de silhueta pode ser descrito pela equação abaixo. Ademais, uma ilustração correspondente aos elementos envolvidos no cálculo de $s(i)$ pode ser visualizada na figura abaixo.

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$

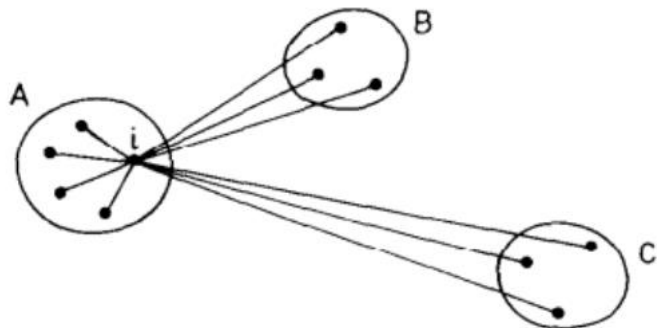


Método de silhueta

3. Para cada ponto i , o coeficiente de silhueta pode ser descrito pela equação abaixo. Ademais, uma ilustração correspondente aos elementos envolvidos no cálculo de $s(i)$ pode ser visualizada na figura abaixo.

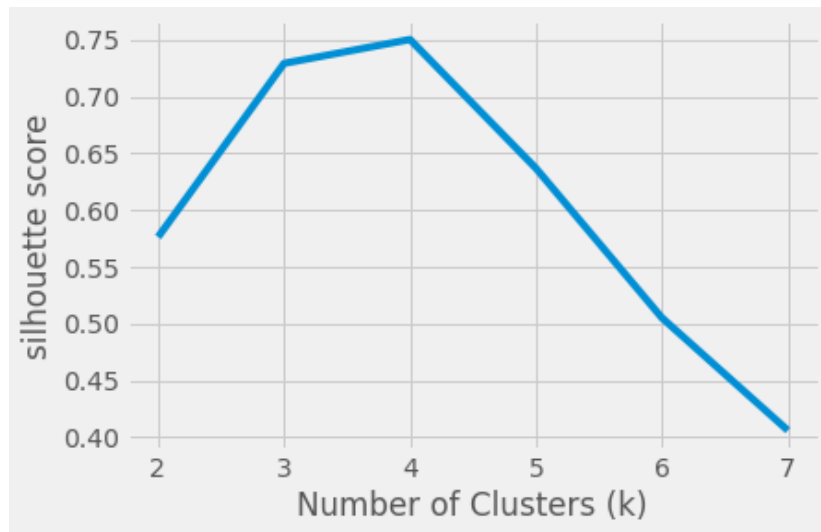
Portanto, o score de um cluster é calculado através da média dos coeficientes de silhueta de todos os pontos pertencentes a este grupo. Assim, para saber o quão bom foi o agrupamento, basta calcular o valor de silhueta do agrupamento total, ou seja, a média dos coeficientes de silhueta de todos os pontos do conjunto de dados.

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}}$$



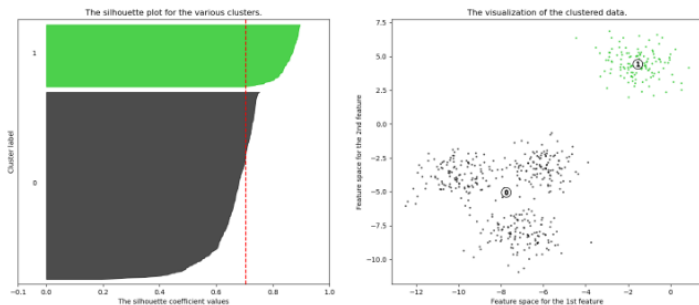
Método de silhueta

Semelhante ao método anterior, escolhemos um intervalo de valores candidatos de k (número de clusters) e, em seguida, treinamos o agrupamento K-Means para cada um dos valores de k . Para cada modelo de agrupamento k-Means representamos os coeficientes de silhueta em um gráfico e observamos as flutuações de cada cluster.

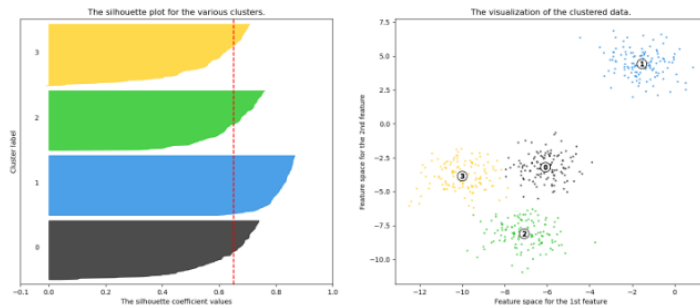


Método de silhueta

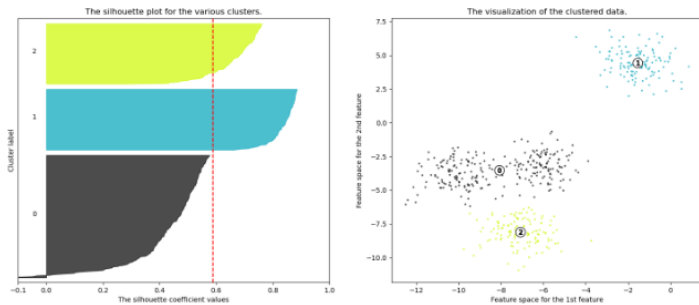
Silhouette analysis for KMeans clustering on sample data with $n_clusters = 2$



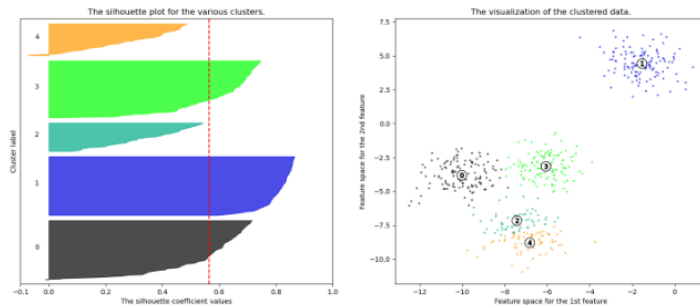
Silhouette analysis for KMeans clustering on sample data with $n_clusters = 4$



Silhouette analysis for KMeans clustering on sample data with $n_clusters = 3$



Silhouette analysis for KMeans clustering on sample data with $n_clusters = 5$



Referências

- [Inteligência Artificial - Aulas de Inteligência Artificial \(google.com\)](#)
- [Data Analytics: Correlation vs. Causality - \(datadeck.com\)](#)
- [PCA using Python \(scikit-learn, pandas\) | Codementor](#)
- [Data Analysis with Python \(udemy.com\)](#)
- [Inteligência Artificial - Aulas de Inteligência Artificial \(google.com\)](#)
- [What Really is R2-Score in Linear Regression? | by Benjamin Obi Tayo Ph.D. | Medium](#)
- [Métricas de avaliação em machine learning – Diego Mariano](#)
- [Supervised vs Unsupervised Learning in 3 Minutes | by Alan Jeffares | Towards Data Science](#)

Referências

- [CS221 \(stanford.edu\)](https://stanford.edu/cs221/)
- [What is R Squared? R2 Value Meaning and Definition \(freecodecamp.org\)](https://www.freecodecamp.org/what-is-r-squared/)
- [R-Squared: Definition, Calculation Formula, Uses, and Limitations \(investopedia.com\)](https://www.investopedia.com/terms/r/rsquared.asp)

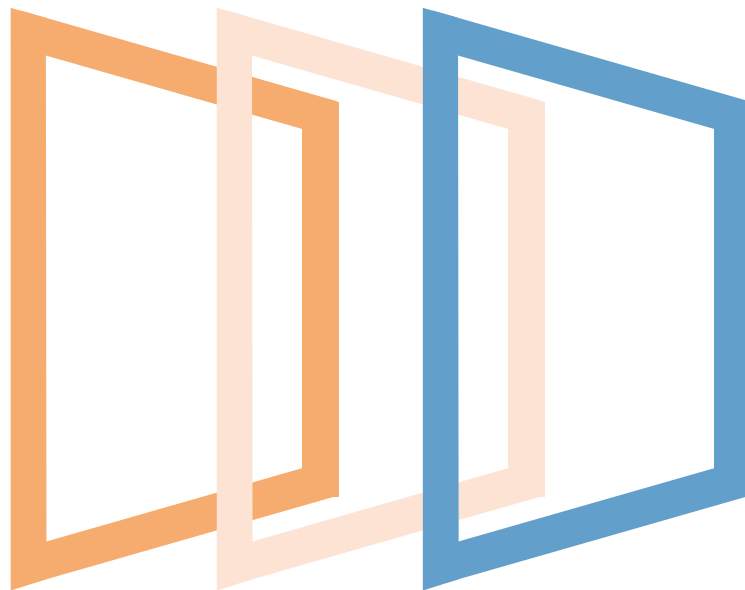
Desvendando IA e Machine Learning

Thaís Ratis

Diego Alexandre

Práticas Tecnológicas, 10.10.2023

minsoit



An Indra company