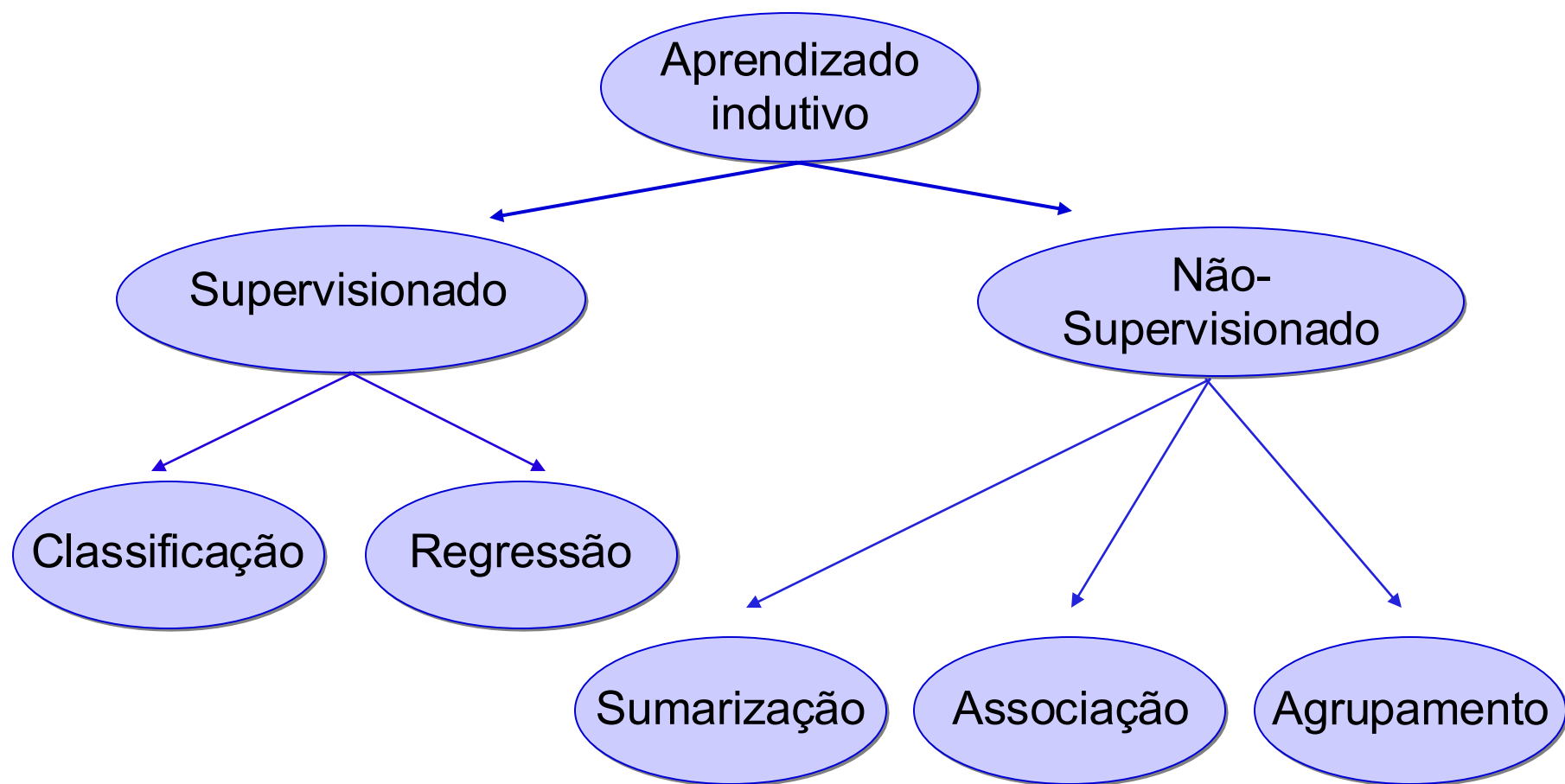


# Inteligência Artificial

## Aula 03 – Aprendizado Supervisionado

# Hierarquia de aprendizado

---



# Aprendizado Supervisionado

---

- Conjunto de dados rotulados por um “supervisor”
- Objetivo é prever a resposta para observações futuras (predição)

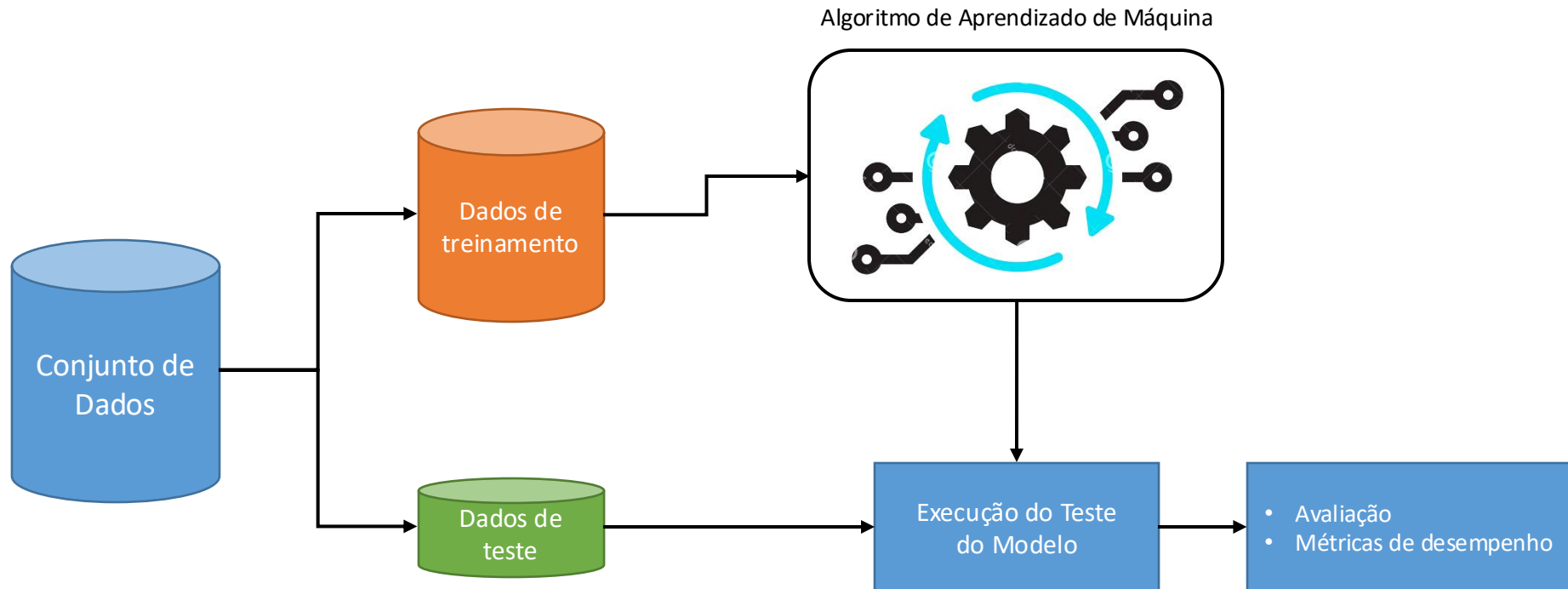
# Aprendizado Supervisionado

---

1. Iniciamos com um conjunto de dados rotulados
2. Dividimos em 2 subconjuntos:  
**treinamento** e **teste**
3. Treinamos um algoritmo com os dados de treinamento (gerando um modelo)
4. Avaliamos o modelo com os dados de teste

# Aprendizado Supervisionado

---



# Conjunto de dados

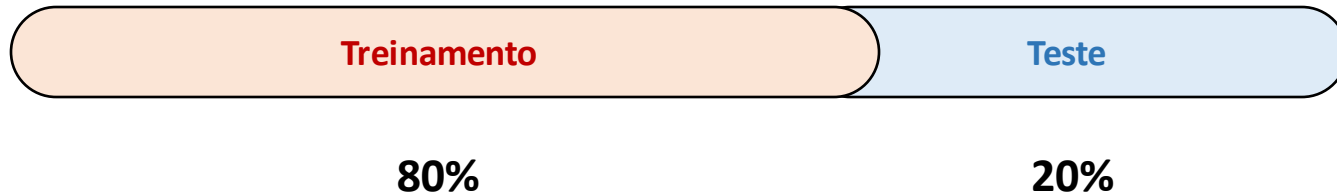
---

- Obtidos a partir de alguma fonte ou banco de dados
  - Pode haver várias operações e manipulações para se obter um conjunto de dados
  - Bancos de dados → Arquivos CSV

# Conjunto de dados

---

- Divide-se o conjunto em treinamento e teste



# Treinamento

---

- Com os dados de treinamento, constrói-se um **modelo de aprendizado de máquina**





# Teste

---

- Execução do modelo treinado com dados **novos e desconhecidos**



# Bibliotecas Python

---

- [Pandas](#)

- Manipulação de CSV
- Tratamento dos conjuntos de dados

- [Scikit-Learn](#)

- Divisão de subconjuntos (treinamento e teste)
- **Algoritmos de Machine Learning**
- Avaliação e métricas de desempenho

# Conjuntos de dados

---

- Machine Learning Data Repository UC Irvine
  - <http://archive.ics.uci.edu/ml/index.php>
- Kaggle
  - Competições práticas promovidas por empresas
  - <https://www.kaggle.com/>
- OpenML
  - <https://www.openml.org/>

# Python 3

---

- `mkdir MLtads`
- `cd MLtads`
- `python3 -m venv .venv`
- Para ativar o ambiente virtual
  - `source .venv/bin/activate` (Linux)
  - `.venv\Scripts\activate` (Windows)
- Após ativação do ambiente virtual:

```
pip3 install ipython ipykernel pandas scikit-learn
```

# VS Code

---



## Python v2024.10.0

Microsoft [microsoft.com](https://microsoft.com) | 130,809,246 | ★★★★★ (595)

Python language support with extension access points for IntelliSense (Pylance), Debugging (Pyt...

Disable

Uninstall

Switch to Pre-Release Version



## Jupyter v2024.6.0

Microsoft [microsoft.com](https://microsoft.com) | 80,643,681 | ★★★★★ (317)

Jupyter notebook support, interactive programming and computing that supports Intellisense, de...

Disable

Uninstall

Switch to Pre-Release Version



Exemplo

# Pima Indians Diabetes Database

Predict the onset of diabetes based on diagnostic measures



UCI Machine Learning • updated 5 years ago (Version 1)

Data

Tasks (10)

Code (1,770)

Discussion (34)

Activity

Metadata


Download (24 kB)

New Notebook



 Usability 8.8

 License CC0: Public Domain

 Tags earth and nature, health, diabetes, healthcare, india

## Description

### Context

This dataset is originally from the National Institute of Diabetes and Digestive and Kidney Diseases. The objective of the dataset is to diagnostically predict whether or not a patient has diabetes, based on certain diagnostic measurements included in the dataset. Several constraints were placed on the selection of these instances from a larger database. In particular, all patients here are females at least 21 years old of Pima Indian heritage.

# Pima Indians Diabetes Database

---

id	pregnant	glucose	bp	skin	insulin	bmi	pedigree	age	label
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

<https://www.kaggle.com/uciml/pima-indians-diabetes-database>



# Pima Indians Diabetes Database

---

diabetes.csv

```
Pregnancies,Glucose,BloodPressure,SkinThickness,Insulin,BMI,DiabetesPedigreeFunction,Age,Outcome
6,148,72,35,0,33.6,0.627,50,1
1,85,66,29,0,26.6,0.351,31,0
8,183,64,0,0,23.3,0.672,32,1
1,89,66,23,94,28.1,0.167,21,0
0,137,40,35,168,43.1,2.288,33,1
5,116,74,0,0,25.6,0.201,30,0
3,78,50,32,88,31,0.248,26,1
10,115,0,0,0,35.3,0.134,29,0
2,197,70,45,543,30.5,0.158,53,1
8,125,96,0,0,0,0.232,54,1
4,110,92,0,0,37.6,0.191,30,0
10,168,74,0,0,38,0.537,34,1
10,139,80,0,0,27.1,1.441,57,0
1,189,60,23,846,30.1,0.398,59,1
5,166,72,19,175,25.8,0.587,51,1
7,100,0,0,0,30,0.484,32,1
0,118,84,47,230,45.8,0.551,31,1
7,107,74,0,0,29.6,0.254,31,1
1,103,30,38,83,43.3,0.183,33,0
1,115,70,30,96,34.6,0.529,32,1
3,126,88,41,235,39.3,0.704,27,0
8,99,84,0,0,35.4,0.388,50,0
7,196,90,0,0,39.8,0.451,41,1
9,119,80,35,0,29,0.263,29,1
11,143,94,33,146,36.6,0.254,51,1
10,125,70,26,115,31.1,0.205,41,1
7,147,76,0,0,39.4,0.257,43,1
...
```

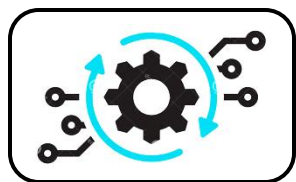
<https://www.kaggle.com/uciml/pima-indians-diabetes-database>

Year	Age	Gender	Height (cm)	Weight (kg)	Body Mass Index (kg/m <sup>2</sup> )	Waist Circumference (cm)	Hip Circumference (cm)	Waist-Hip Ratio	Visceral Fat Area (cm <sup>2</sup> )	Subcutaneous Fat Area (cm <sup>2</sup> )	Total Fat Area (cm <sup>2</sup> )
1990	20	Male	170	65	22.0	85	95	0.89	150	100	250
1995	25	Male	175	75	24.2	90	100	0.90	160	110	270
2000	30	Male	180	85	26.3	95	105	0.90	170	120	290
2005	35	Male	185	95	28.4	100	110	0.91	180	130	310
2010	40	Male	190	105	30.0	105	115	0.91	190	140	330



# Pipeline

---



Modelo

A blurred image of a table with multiple columns and rows, representing a dataset.

Conjunto de Teste



Predições

# Avaliação

- Exemplo de entrada:

id	pregnant	glucose	bp	skin	insulin	bmi	pedigree	age	label
0	6	148	72	35	0	33.6	0.627	50	1
1	1	85	66	29	0	26.6	0.351	31	0
2	8	183	64	0	0	23.3	0.672	32	1
3	1	89	66	23	94	28.1	0.167	21	0
4	0	137	40	35	168	43.1	2.288	33	1

↑  
variável alvo  
(target)

- Supondo que o algoritmo fez a seguinte predição:

id	prediction
0	1
1	1
2	0
3	0
4	1



Acurácia: 60%

# Exercício

# Leitura e manipulação dos dados

---

## Importação dos dados

```
import pandas  
  
df = pandas.read_csv("diabetes.csv")
```

## Preparação dos dados

```
# todas as colunas do conjunto  
df.columns  
  
# seleciona uma coluna específica (Outcome)  
df['Outcome']  
  
# seleciona todas as colunas exceto Outcome  
df.loc[:, df.columns != 'Outcome']
```

# Leitura e manipulação dos dados

---

Separação dos conjuntos de treinamento e teste

```
from sklearn.model_selection import train_test_split  
  
X_train, X_test, y_train, y_test = train_test_split(X, y)
```

[https://scikit-learn.org/stable/modules/generated/sklearn.model\\_selection.train\\_test\\_split.html](https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html)

Treinamento e obtenção do modelo

```
from sklearn.linear_model import LogisticRegression  
from sklearn.metrics import classification_report  
  
clf = LogisticRegression() # instância do algoritmo  
  
clf.fit(X_train, y_train) # treinamento do modelo
```

# Leitura e manipulação dos dados

---

Teste e avaliação do modelo

```
y_pred = clf.predict(X_test) # obtém coluna de predições
```

```
print(clf.score(X_test, y_test))
```

```
print(classification_report(y_test, y_pred))
```



# Exercício

---

- Crie um modelo de aprendizado supervisionado para classificar espécies de flores do conjunto de dados **Iris Dataset**, (pesquie-o no repositório UCI Machine Learning Repository).
- Utilize a implementação do Scikit-Learn para três algoritmos distintos:
  - Regressão Logística
  - Árvore de Decisão
  - Random Forest
- Execute um pipeline completo e avalie o desempenho desses modelos.