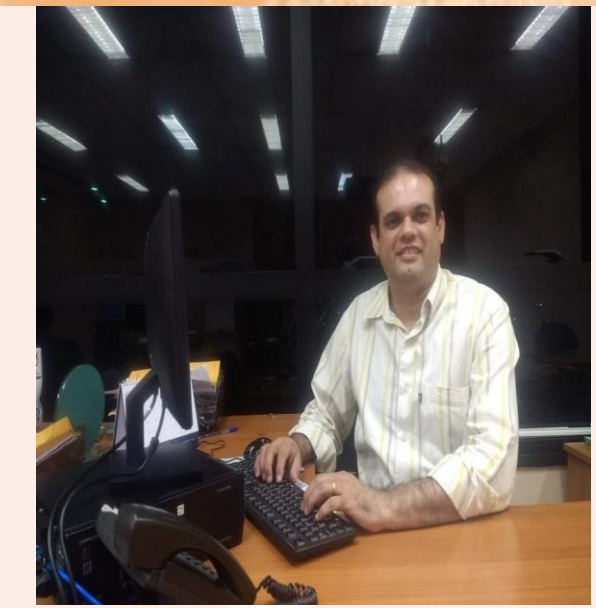

VERITAS
UNINORTE
ser
educacional

Machine Learning

Prof. Wanderlan Carvalho de Albuquerque

Machine Learning

- Objetivo: Avaliar a Matriz de Desemepenho



Avaliação da Matriz de Confusão

Etapas Fundamentais para construção do Algoritmo de IA

1) Análise do problema.

2) Exploração, Tratamento e Análise dos dados.

3) Pré processamento dos dados.

4) Escolha do grupo de algoritmos que podem ser utilizados.

5) Criação dos algoritmos de Machine Learning.

6) Comparação e escolha do melhor algoritmo.

Repositório de dados

INEP: https://www.gov.br/inep/pt-br/acesso-a-informacao/dados-abertos/microdados

Google dataset Search: https://datasetsearch.research.google.com/

Portal brasileiro de dados abertos: www.dados.gov.br

Kaggle (competições Machine Learning): www.kaggle.com

UCI Machine Learning Repository: https://archive.ics.uci.edu/ml/index.php

OMS: https://www.who.int/

Paho (organização panamericana de saúde): https://www.paho.org/en

DrivenData (competições Ciência de Dados): https://www.drivendata.org/

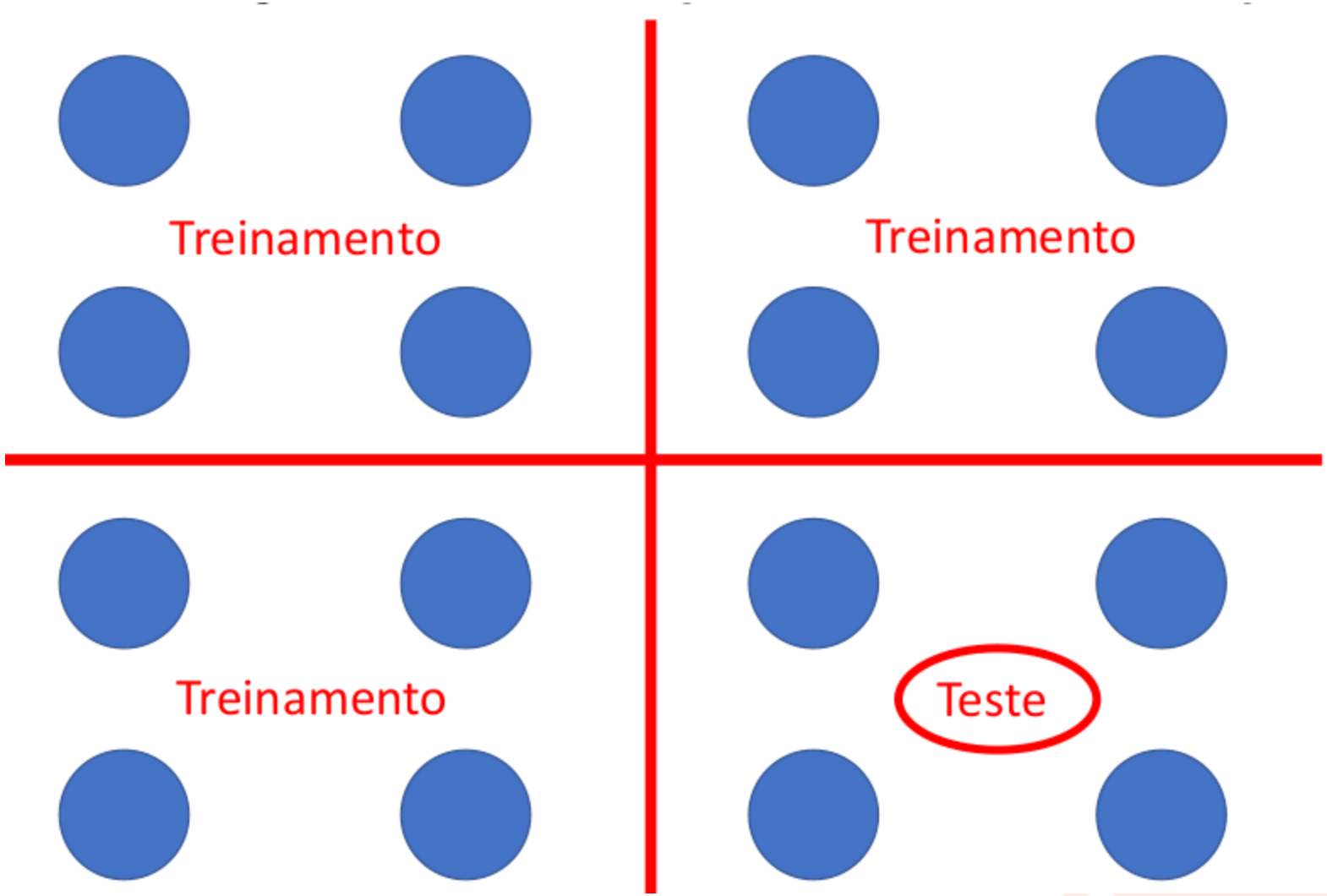
Separação de Dados de Treino e Teste

Dados de treino: Certa quantidade dos dados (aproximadamente 70%) destinada para treinar o algoritmo.

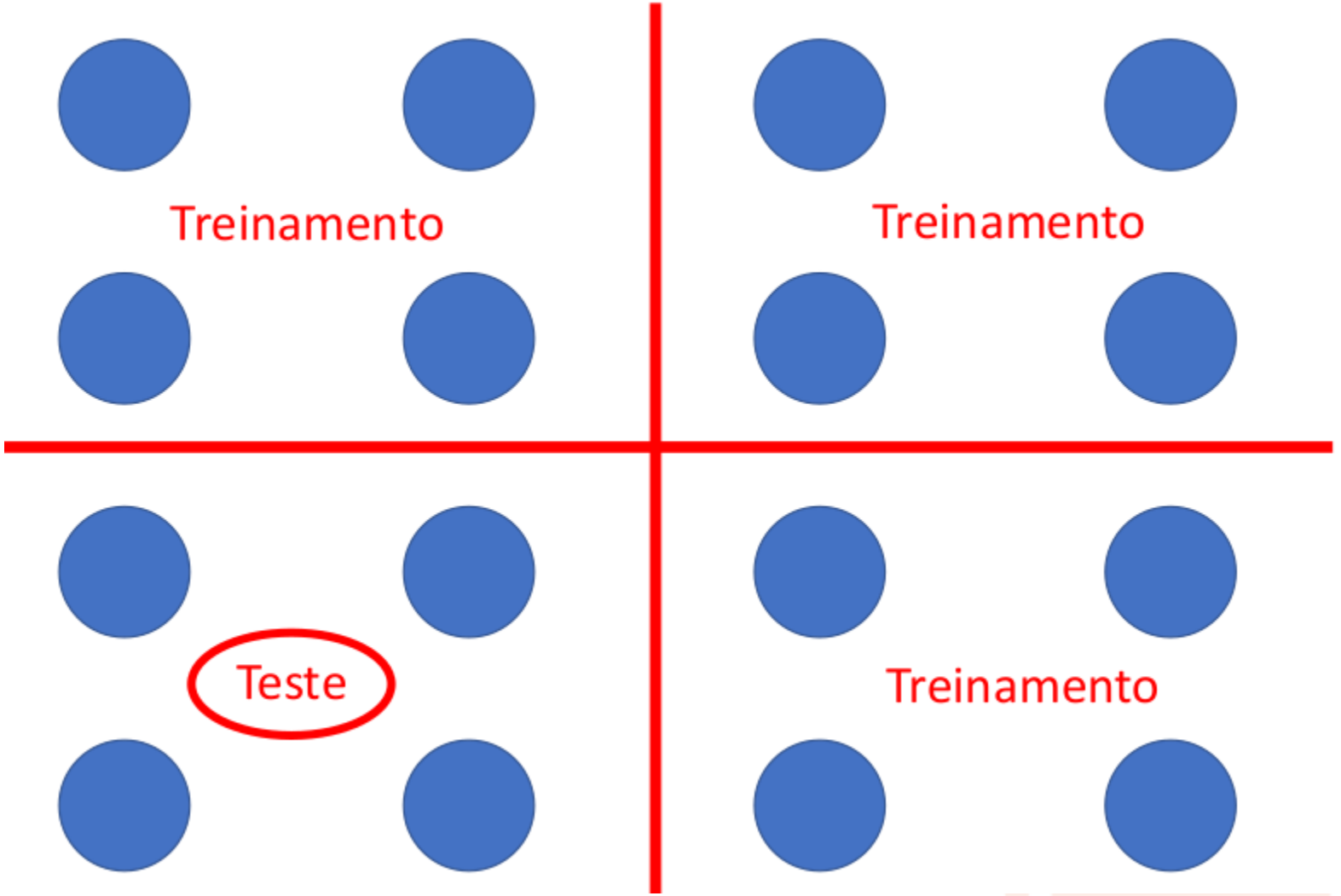
Dados de teste: Quantidade restante dos dados (aproximadamente 30%) para analisar o desempenho do algoritmo.

Essa separação deve ocorrer de maneira aleatória para evitar problemas nos modelos criados (Exemplo: ter uma quantidade de dados que aparecem em pequena quantidade ou nem aparecem nos dados de teste).

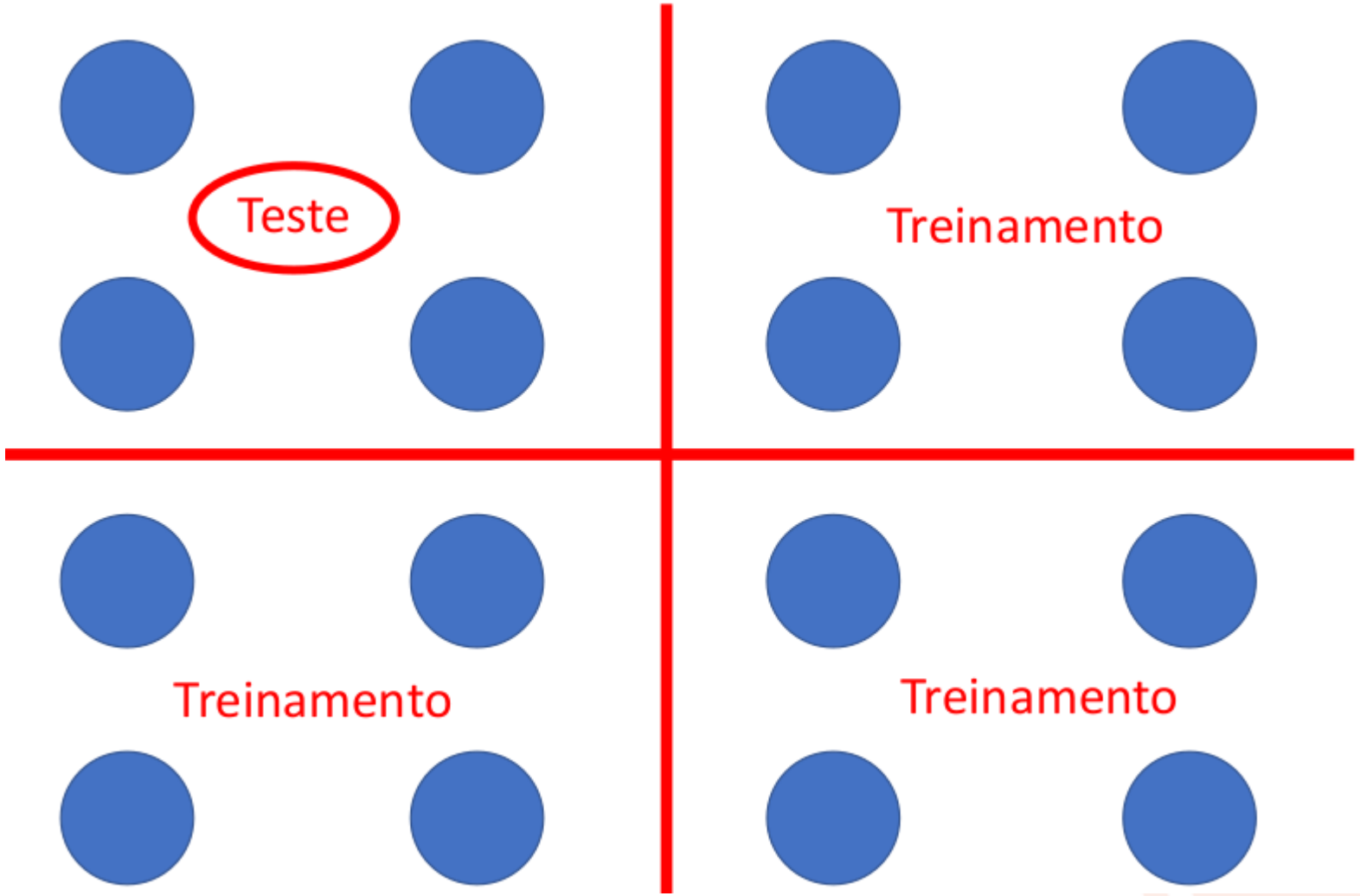
Validação de dados



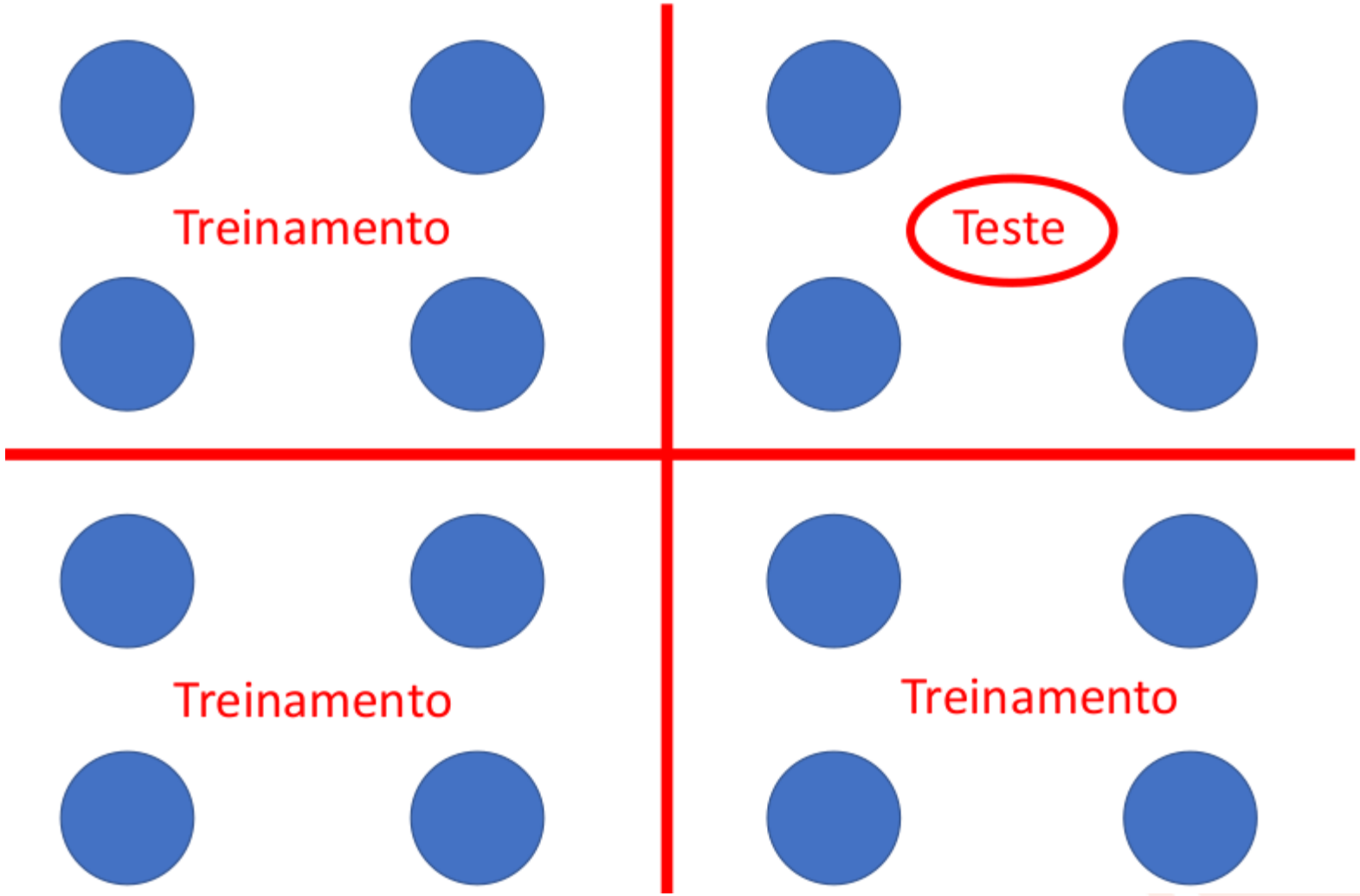
Validação de dados



Validação de dados

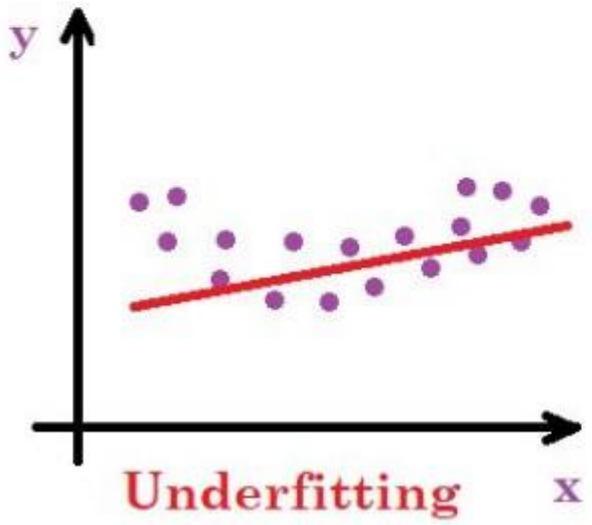


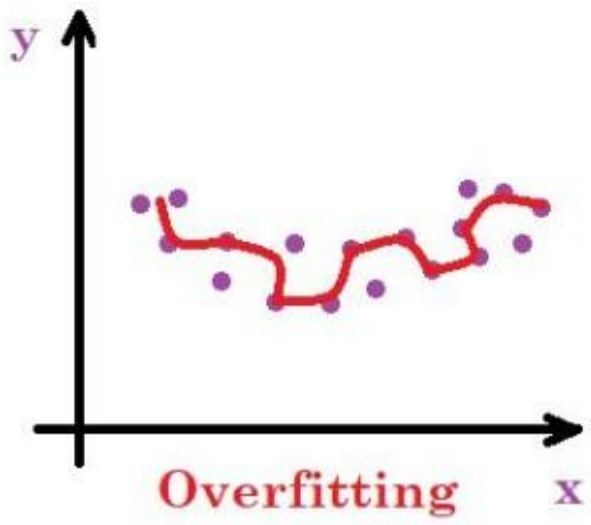
Validação de dados

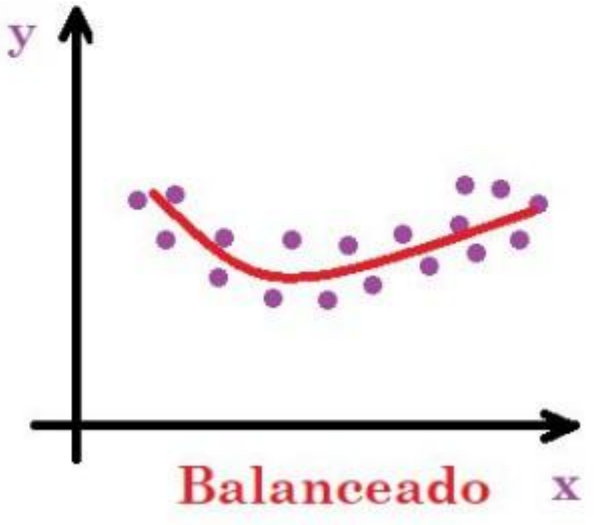


Considerações para escolha dos dados

Atenção a dois problemas no treinamento		
1	Underfitting (alto viés)	Algoritmo que não se encaixa com os dados de entrada.
2	Overfitting (alta variância)	Algoritmo ótimo para os dados de entrada e ruim para dados de teste.







Classificador

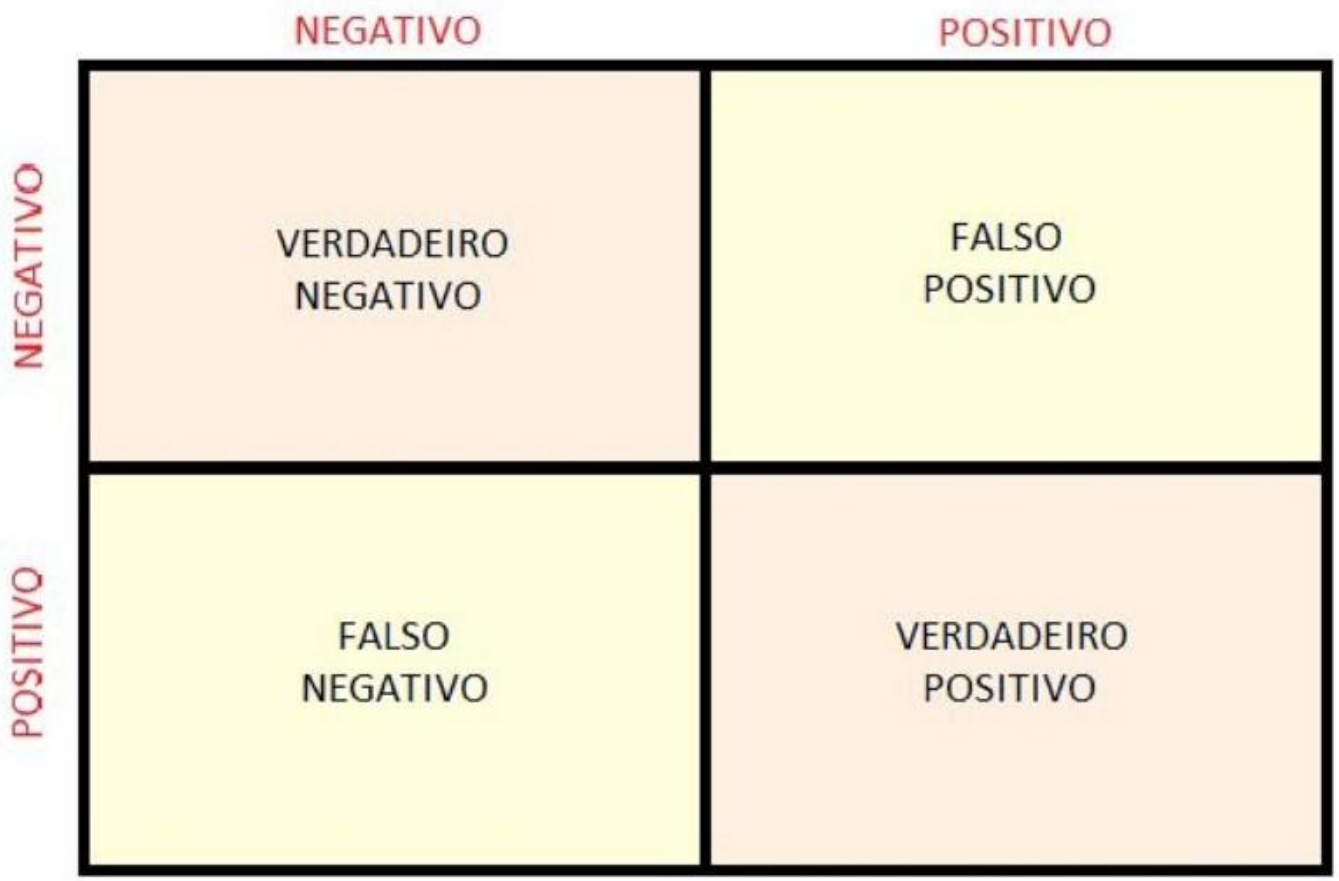
Com base nos dados de entrada estima-se um “classificador” que gera como saída uma classificação qualitativa de um dado não observado (Ex.: análise de crédito, chances de desenvolver doenças).

Paciente	Pressão Alta	Colesterol Alto	Triglicérides Alto	Pratica Esporte	TEVE AVC?
1	SIM	SIM	SIM	NÃO	SIM
2	NÃO	SIM	NÃO	SIM	NÃO
3	NÃO	NÃO	SIM	NÃO	SIM
4	SIM	NÃO	NÃO	SIM	NÃO
5	SIM	SIM	NÃO	NÃO	NÃO

Paciente	Pressão Alta	Colesterol Alto	Triglicérides Alto	Pratica Esporte	PODERÁ TER AVC?
A	NÃO	SIM	SIM	SIM	?????

Matriz de Confusão/Erros

Análise do desempenho: Matriz de confusão



Matriz de Confusão/Erros

```
In [2]: # 1 para grávida, 0 para não grávida
        valores_reais    = [1, 0, 1, 0, 0, 0, 1, 0, 1, 0]
        valores_preditos = [1, 0, 0, 1, 0, 0, 1, 1, 1, 0]
```

- Como saber se meu modelo previu bem? Como saber se ele prevê bem a classe que queremos (Grávida)?
- Essas e outras questões podemos entender com as matrizes de confusão.

Matriz de Confusão/Erros

- **Verdadeiro positivo (true positive — TP):** ocorre quando no conjunto real, a classe que estamos buscando foi prevista corretamente.
  - Por exemplo, quando a mulher está grávida e o modelo previu corretamente que ela está grávida.
- **Falso positivo (false positive — FP):** ocorre quando no conjunto real, a classe que estamos buscando prever foi prevista incorretamente.
  - Exemplo: a mulher não está grávida, mas o modelo disse que ela está..

Matriz de Confusão/Erros

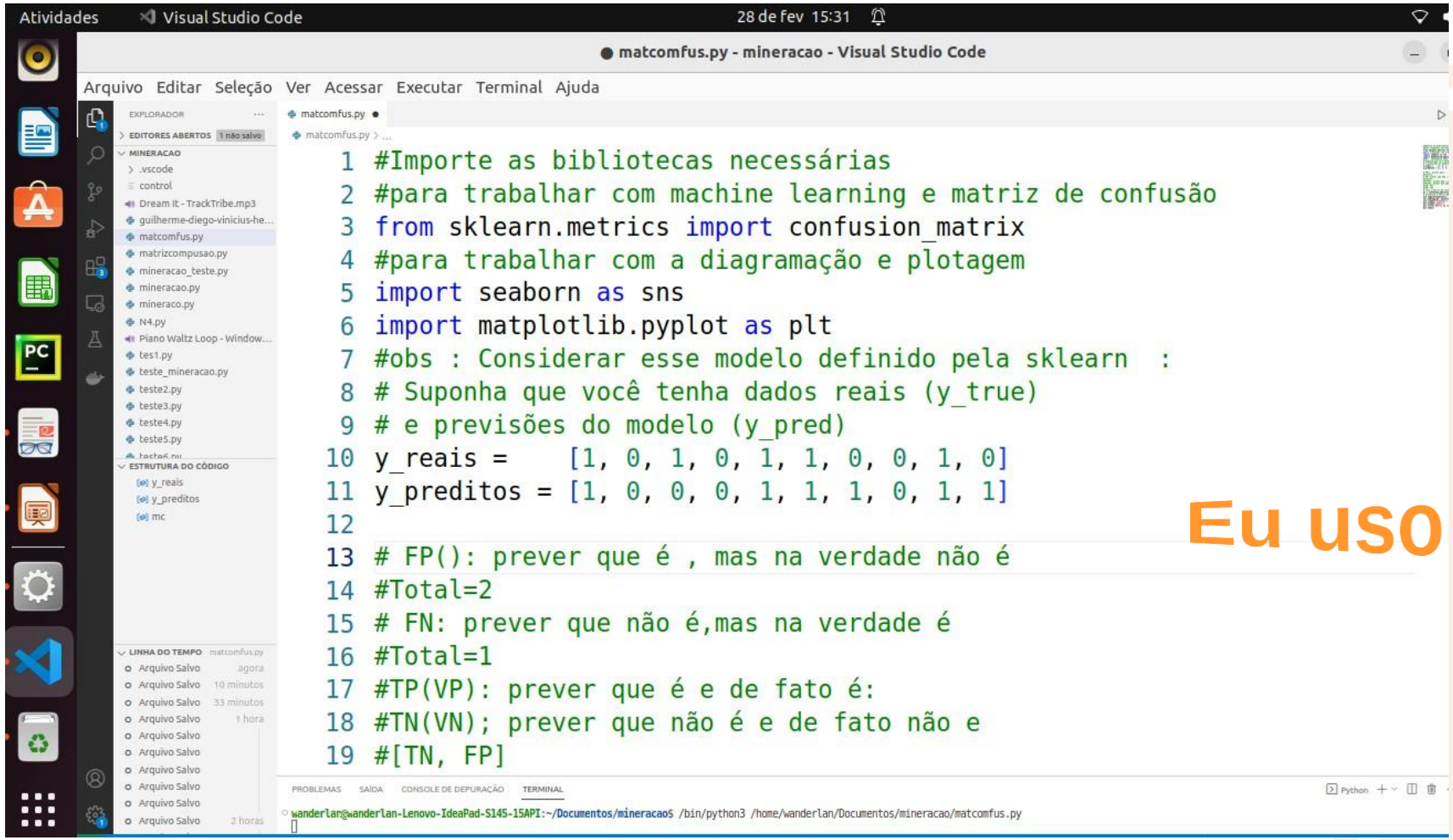
- **Falso verdadeiro (true negative — TN)**: ocorre quando no conjunto real, a classe que não estamos buscando prever foi prevista corretamente.

  Exemplo: a mulher não estava grávida, e o modelo previu corretamente que ela não está.

Falso negativo (false negative — FN): ocorre quando no conjunto real, a classe que não estamos buscando prever foi prevista incorretamente.

Exemplo, quando a mulher está grávida e o modelo previu incorretamente que ela não está grávida.

Matriz de Confusão/Erros



Eu uso o Visual Studio

Matriz de Confusão/Erros

Usem o Google Colab !

Matriz de Confusão/Erros

- Acesse o site do Google Colab: https://colab.research.google
- Faça login com sua conta do Google, se necessário.

Passo 2: Crie um novo notebook ou abra um existente

- No Google Colab, você pode criar um novo notebook clicando em "Arquivo" > "Novo notebook" ou abrindo um notebook existente.

Passo 3: Instale uma biblioteca scikit-learn

- Na primeira célula de código do notebook, digite o seguinte comando e execute a célula:

pyt (e pyt — Código de cópia

```
!pip install scikit-learn
```

Matriz de Confusão/Erros

Passo 4: Importe a biblioteca scikit-learn

- Na próxima célula de código, importe a biblioteca scikit-learn usando o seguinte código:

```
python (em (piona                                    Código de cópia
import sklearn
```

Passo 5: Utilize o scikit-learn

- Agora você pode usar as funcionalidades do scikit-learn em seu notebook. Por exemplo, você pode importar um modelo específico e treiná-lo com dados:

```
py (pia)                                             Código de cópia
from sklearn.linear_model import LinearRegression
from sklearn.datasets import make_regression
```

Matriz de Confusão/Erros

py (pia) Código de cópia

```
from sklearn.linear_model import LinearRegression
from sklearn.datasets import make_regression

# Criação de dados de exemplo
X, y = make_regression(n_samples=100, n_features=1, noise=0.1)

# Criação do modelo de regressão linear
model = LinearRegression()

# Treinamento do modelo
model.fit(X, y)

# Previsões
predictions = model.predict(X)
```

Matriz de Confusão/Erros

matcomfus.py > ...

```python
#Importe as bibliotecas necessárias
#para trabalhar com machine learning e matriz de confusão
from sklearn.metrics import confusion_matrix
#para trabalhar com a diagramação e plotagem
import seaborn as sns
import matplotlib.pyplot as plt
```

Primeira parte no ambiente de desenvovimento

Matriz de Confusão/Erros

```
#obs : Considerar esse modelo definido pela sklearn  :
# Suponha que você tenha dados reais (y_true)
# e previsões do modelo (y_pred)
y_reais =    [1, 0, 1, 0, 1, 1, 0, 0, 1, 0]
y_preditos = [1, 0, 0, 0, 1, 1, 1, 0, 1, 1]
# FP(): prever que é , mas na verdade não é
#Total=2
# FN: prever que não é,mas na verdade é
#Total=1
#TP(VP): prever que é e de fato é:
#TN(VN); prever que não é e de fato não e
#[TN, FP]
#[FN, TP]
```

Segunda Parte

Matriz de Confusão/Erros

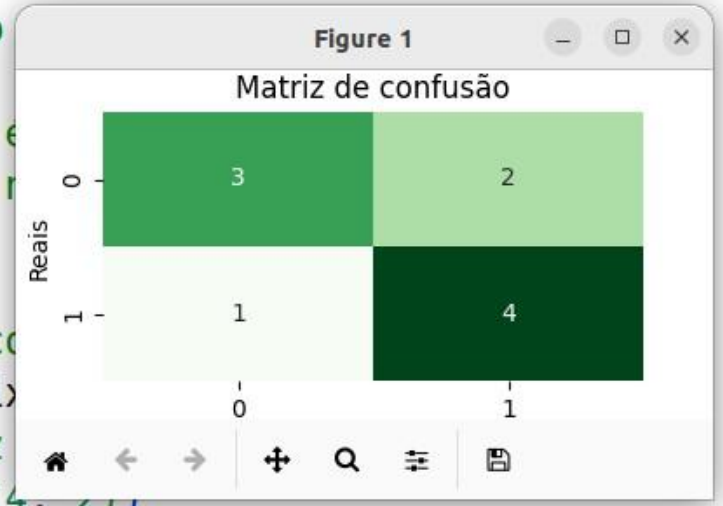
```
20 # Crie a matriz de confusão
21 mc = confusion_matrix(y_reais, y_preditos)
22 # Visualize a matriz de confusão
23 plt.figure(figsize=(4, 2))
24 sns.heatmap(mc, annot=True, cmap='Greens', fmt='d', cbar=False)
25 plt.xlabel('Preditos')
26 plt.ylabel('Reais')
27 plt.title('Matriz de confusão')
28 plt.show()
```

Terceira Parte

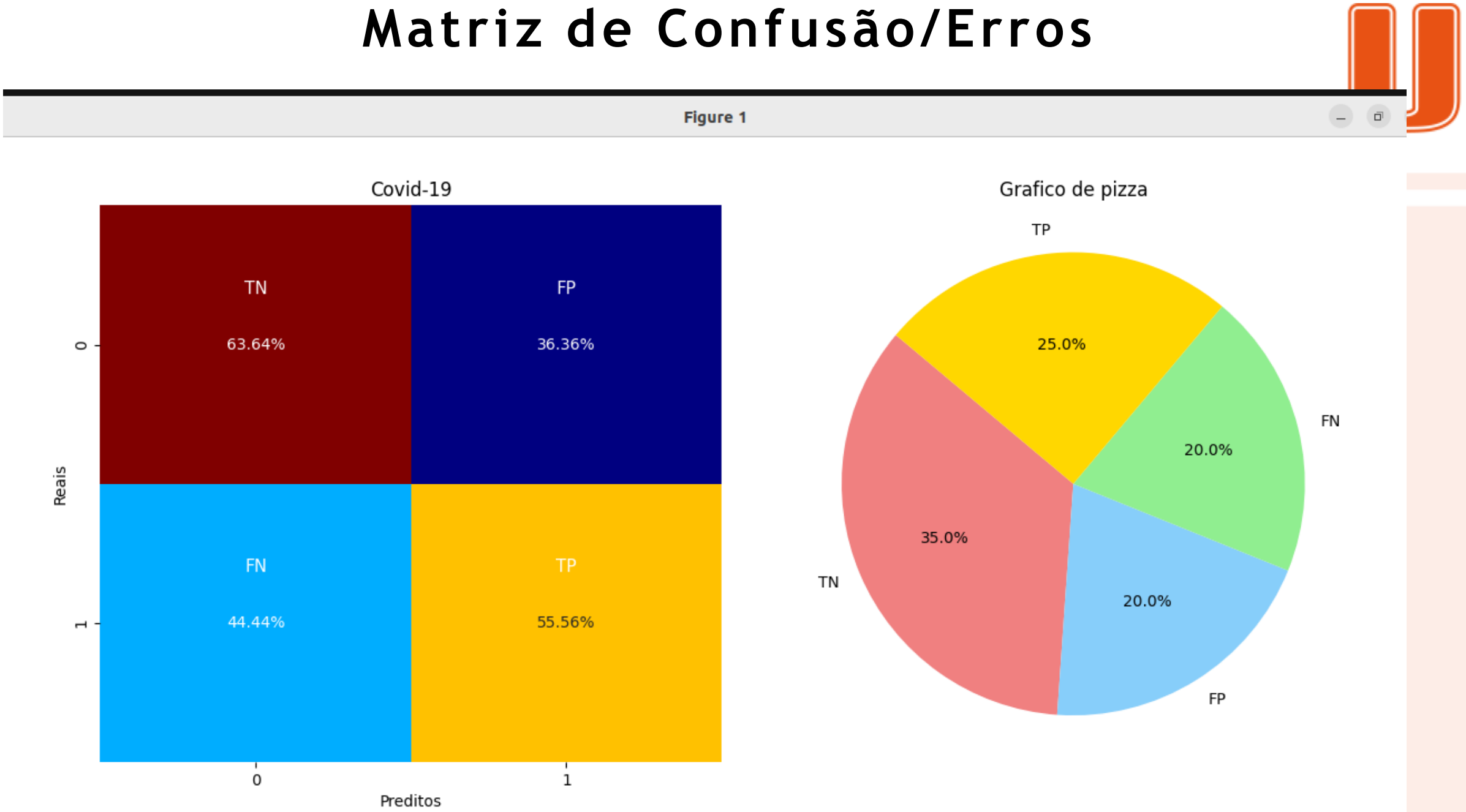
Matriz de Confusão/Erros

matcomfus.py - mineracao - Visual Studio Code

```
11 y_preditos = [1, 0, 0, 0, 1, 1, 1, 0, 1, 1]
12 # FP(): prever que é , mas na verdade não é
13 #Total=2
14 # FN: prever que não
15 #Total=1
16 #TP(VP): prever que e
17 #TN(VN); prever que n
18 #[TN, FP]
19 #[FN, TP]
20 # Crie a matriz de co
21 mc = confusion_matrix
22 # Visualize a matriz
23 plt.figure(figsize=(4, 2))
24 sns.heatmap(mc, annot=True, cmap='Greens', fmt='d', cbar=False)
25 plt.xlabel('Preditos')
26 plt.ylabel('Reais')
27 plt.title('Matriz de confusão')
28 plt.show()
```



Matriz de Confusão/Erros



Matriz de Confusão/Erros

Acurácia : é uma métrica fundamental para avaliar a performance geral de um modelo de classificação. Ela mede a proporção de exemplos classificados corretamente em relação ao total de exemplos.

Matriz de Confusão/Erros

A fórmula para calcular a acurácia é:

$$\text{Acurácia} = \frac{\text{Verdadeiros Positivos (TP)} + \text{Verdadeiros Negativos (TN)}}{\text{Total de Amostras}}$$

Em outras palavras, a acurácia responde à pergunta: "Qual a proporção de exemplos que o modelo classificou corretamente, independentemente da classe?"

- Verdadeiros Positivos (TP) são os casos em que o modelo previu corretamente a classe positiva.
- Verdadeiros Negativos (TN) são os casos em que o modelo previu corretamente a classe negativa.
- Total de Amostras é a soma de Verdadeiros Positivos, Falsos Positivos, Verdadeiros Negativos e Falsos Negativos.

Matriz de Confusão/Erros

Precisão : é uma métrica que mede a proporção de exemplos positivos previstos corretamente em relação ao total de exemplos positivos previstos pelo modelo.

Matriz de Confusão/Erros

A fórmula para calcular a precisão é:

$$\text{Precisão} = \frac{\text{Verdadeiros Positivos (TP)}}{\text{Verdadeiros Positivos (TP)} + \text{Falsos Positivos (FP)}}$$

Em outras palavras, a precisão responde à pergunta: "De todas as instâncias que o modelo classificou como positivas, quantas realmente são positivas?"

Matriz de Confusão/Erros

Sensibilidade: também conhecida como recall ou taxa de verdadeiros positivos (TPR - True Positive Rate), é uma métrica que mede a proporção de exemplos positivos que foram corretamente identificados pelo modelo em relação ao total de exemplos positivos reais.

Matriz de Confusão/Erros

A fórmula para calcular a sensibilidade é:

$$\text{Sensibilidade (Recall)} = \frac{\text{Verdadeiros Positivos (TP)}}{\text{Verdadeiros Positivos (TP)} + \text{Falsos Negativos (FN)}}$$

Em outras palavras, a sensibilidade responde à pergunta: "Quão bom o modelo é em detectar os verdadeiros positivos em relação a todos os exemplos positivos reais?"

Matriz de Confusão/Erros

F1 Score : também conhecido como pontuação F1, é uma métrica que combina precisão e sensibilidade (recall) em um único valor, proporcionando uma medida geral do desempenho de um modelo de classificação.

Matriz de Confusão/Erros

O F1 Score é calculado pela média harmônica da precisão e da sensibilidade (recall). A média harmônica dá mais peso aos valores menores. A fórmula para calcular o F1 Score é:

$$\text{F1 Score} = 2 \times \frac{\text{Precisão} \times \text{Sensibilidade (Recall)}}{\text{Precisão} + \text{Sensibilidade (Recall)}}$$

Em outras palavras, o F1 Score é uma medida do equilíbrio entre precisão e sensibilidade. Ele é útil quando há um desequilíbrio entre as classes de interesse.

Um valor de F1 Score próximo de 1 indica um modelo com boa precisão e sensibilidade. Um valor de 0 indica um desempenho muito ruim.

O F1 Score é particularmente útil em problemas de classificação binária, onde há duas classes de interesse, mas também pode ser calculado para problemas de classificação multiclasse usando a média ponderada dos F1 Scores de cada classe.

Visão Geral

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

$$Precision = \frac{VP}{VP + FP}$$

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

$$F1\ Score = 2.\frac{Precision\,.\,Recall}{Precision + Recall}$$

Visão Geral

```python
from sklearn.metrics import confusion_matrix
y_true = [0, 1, 0, 1, 1, 1]
y_pred = [0, 1, 1, 1, 1, 0]
# Calculando a matriz de confusão
cm = confusion_matrix(y_true, y_pred)
# Imprimindo a matriz de confusão
print("Matriz de Confusão:")
print(cm)
```

Visão Geral

```
# Calculando métricas de avaliação a partir
TP = cm[1, 1]  # True Positives
TN = cm[0, 0]  # True Negatives
FP = cm[0, 1]  # False Positives
FN = cm[1, 0]  # False Negatives
```

Visão Geral

```
# Calculando a precisão
precision = TP / (TP + FP)
# Calculando a sensibilidade (recall)
recall = TP / (TP + FN)
```

Visão Geral

```
# Calculando a pontuação F1
f1_score = 2 * (precision * recall) /
(precision + recall)
```

Visão Geral

```
# Imprimindo as métricas de avaliação
print("\nMétricas de Avaliação:")
print("Precisão:", precision)
print("Sensibilidade (Recall):", recall)
print("Pontuação F1:", f1_score)
```

Visão Geral

```
wanderlan@wanderlan-Lenovo-IdeaPad-S145-15API:~/Documentos/mineracao$ /bin/python3 "/home/wanderlan/Documentos/mi
neracao/# Valores_de_precisai.py"
Matriz de Confusão:
[[1 1]
 [1 3]]

Métricas de Avaliação:
Precisão: 0.75
Sensibilidade (Recall): 0.75
Pontuação F1: 0.75
wanderlan@wanderlan-Lenovo-IdeaPad-S145-15API:~/Documentos/mineracao$
```

ATIVIDADE

- Plote uma Matriz de confusão para testar se as pessoas estão com Covid. Considere uma amostra de tamanho de 20 pessoas para valores reais e valores preditivos. Vale 1,0 ponto
- Implemente em Python as porcentagens dos resultados de acurácia, precisão , Sensibilidade e F1 Score e mostre o resultado em um gráfico.

OBRIGADO