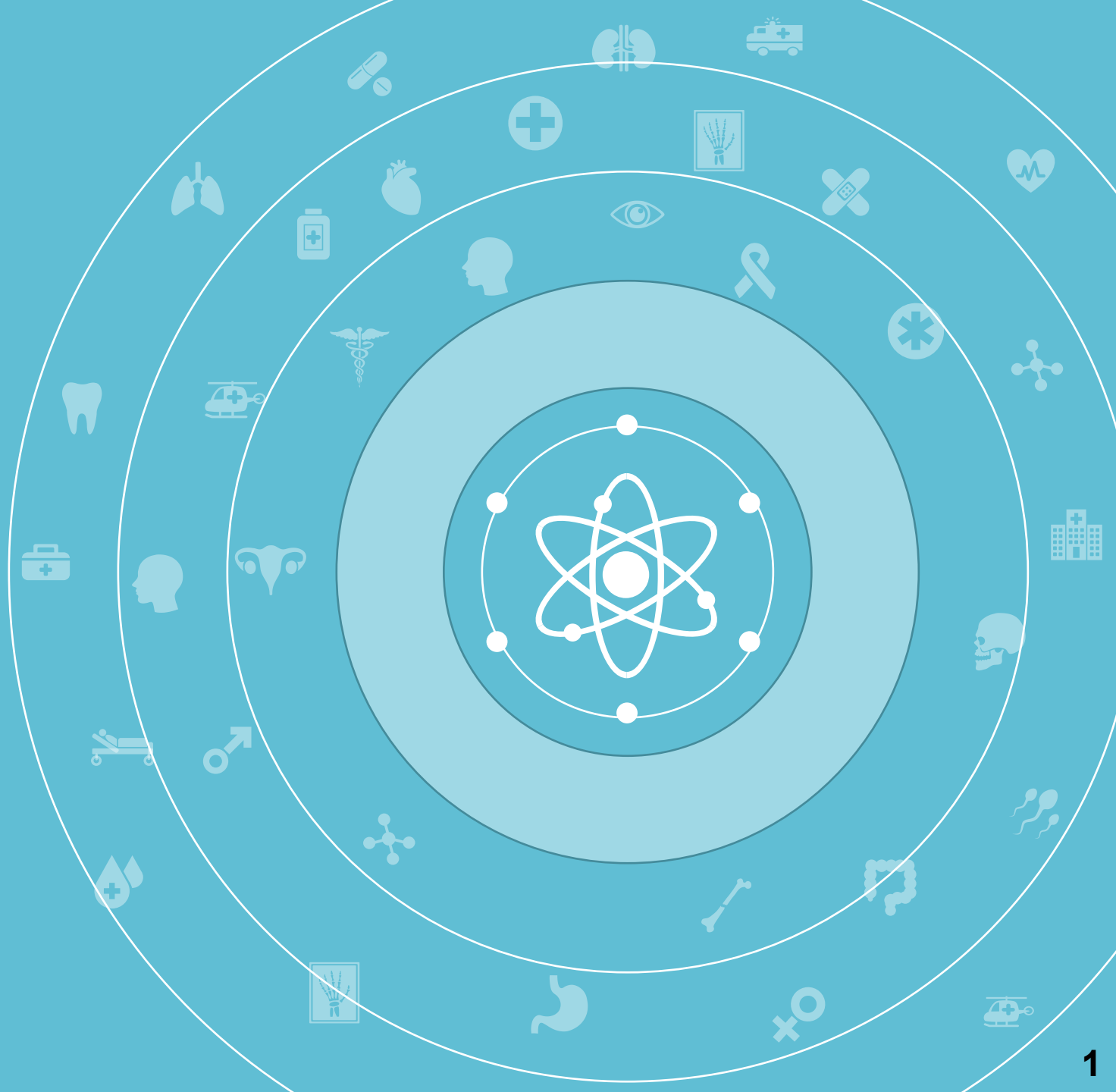
CLASSIFICAÇÃO DE MODELOS CARDÍACOS TRIDIMENSIONAIS RELACIONADA À PRESENÇA OU À AUSÊNCIA DE CARDIOMIOPATIA

SIN5007 – Reconhecimento de Padrões
1º sem. 2020
Profa. Dra. Ariane Machado Lima

Diego Pereira Dedize
Luiz Gustavo de Almeida Silva
Vagner Mendonça Gonçalves



AGENDA

01 **DEFINIÇÃO DO PROBLEMA**

Definição do objeto de estudo e problemática envolvida.

02 **CONJUNTO DE DADOS**

Descrição do dataset e das características.

03 **PRÉ-PROCESSAMENTO**

Estratégias de pré-processamento e PCA.

04 **SELEÇÃO DE CARACTERÍSTICAS**

Mais estratégias de redução da dimensionalidade.

05 **VALIDAÇÃO CRUZADA**

Função de divisão do database em k conjuntos.

VALIDAÇÃO CRUZADA

Função de Divisão da Base em k Conjuntos

- A função implementada recebe três parâmetros: **data**: um conjunto de dados (no caso, a database composta pelos vetores de características); **target**: um conjunto com as classes de cada amostra de data; e, **k**: a quantidade de conjuntos (*folds*) em que se deseja dividir a database.
- Optamos por incluir o parâmetro **target**, pois nossa database possui, originalmente, três classes (SAN, CMH e CMD). Dessa forma, poderemos generalizar a função futuramente para trabalhar com todas as classes.
- A função separa a database em k conjuntos de treinamento/teste.
- Em cada conjunto se respeita o máximo possível a proporção das classes em referência a database original completa.
- Em cada conjunto, a quantidade de amostras entre as classes poderão ter, no máximo, 1 elemento a mais ou a menos em relação a outro conjunto.
- Trabalhamos com a linguagem de programação Python e o framework Anaconda.

VALIDAÇÃO CRUZADA

Função de Divisão da Base em k Conjuntos – Código-fonte (1 de 3)

```
# Bibliotecas
import numpy as np
import pandas as pd
import random
import math
import matplotlib.pyplot as plt

from sklearn.model_selection import KFold
from sklearn.model_selection import GroupKFold
from sklearn.model_selection import StratifiedKFold
```

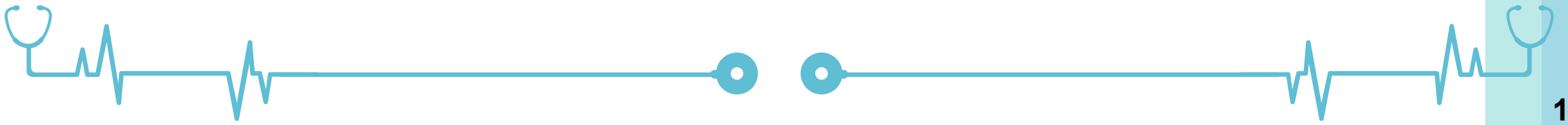
```
# Leitura do arquivo de entrada

df = pd.read_csv('CMCT_20200503.csv', )
```

```
# Seleção somente dos TARGET (Classes) 0 - Sem Anomalia e 2 - Cardiomiopatia Dilatada
dfx = df.copy()

masc = dfx['TARGET'] != 1
dfy = dfx.loc[masc, :]

X = dfy.drop(['TARGET'], axis=1)
Y = dfy['TARGET']
```



VALIDAÇÃO CRUZADA

Função de Divisão da Base em k Conjuntos

Código-fonte (2 de 3) - Função

```
##########################################
#### Separa uma database em k conjuntos ####
##########################################
def splitFolds(data, target, k=10):

    # Contadores para apresentação
    ldata = len(data)              # Quantidade de linhas da base
    numel = int(ldata / k)         # Quantidade de amostras por fold

    uclass = target.value_counts() # Classes e suas quantidades
    uclass.sort_index(inplace=True)
    nclass = uclass.index          # Classes
    qclass = uclass.values         # Quantidade de cada classe

    # Junção das classes e suas quantidades para um dicionário
    zclass = zip(nclass, qclass)
    dclass = dict(zclass)

    # Separação dos conjuntos
    partesK = []    # Conterá todos os conjuntos k de índices, cada um proporcional
                    #a cada classe
    for i in range(k):
        pk = []
        if (i < 7):
            nelem = numel + 1
        else:
            nelem = numel

        ntot = nelem

        for nc, qc in dclass.items():
            # Captura de todos os índices da coluna target
            masc = target == nc
            idclass = list(target[masc].index)

            # Montagem dos k conjuntos com a proporção o mais próxima possível da
            #database completa
            propclass = int(round(nelem * qc / ldata))
            if (ntot > propclass):
                ntot -= propclass
            else:
                propclass = ntot

            rs = random.sample(idclass, propclass)
            pk = pk + rs

        partesK.append(pk)

    return (partesK)
```

VALIDAÇÃO CRUZADA

Função de Divisão da Base em k Conjuntos

Código-fonte (3 de 3) - Execução

```
##########################################
#### Execução e impressão do relatório ####
##########################################

k = 10
partesK = splitFolds(X, Y, k)

qtdDS = len(X)
vc = Y.value_counts()
qtdT0 = vc.values[1]
qtdT2 = vc.values[0]
perT0 = round(qtdT0/qtdDS * 100, 2)
perT2 = round(qtdT2/qtdDS * 100, 2)

print('**** Dataset: CMCT_20200503.csv ****')
print('k =', k, ', Dataset:', qtdT0, 'SAN e', qtdT2, 'CMD (', perT0, '%  x ',
      perT2, '% )')
print()

qtd0 = 0
qtd2 = 0
per0 = 0.0
per2 = 0.0

for f in range(len(partesK)):
    dx = Y[partesK[f]]
    vc = dx.value_counts()
    qtd0 = vc.values[1]
    qtd2 = vc.values[0]
    per0 = round(qtd0/(qtd0 + qtd2) * 100, 2)
    per2 = round(qtd2/(qtd0 + qtd2) * 100, 2)

    print(f'Fold {f+1}: SAN: {qtd0}, CMD: {qtd2}, Total: {qtd0+qtd2},',
          f'Proporção: {per0}%; {per2}%')
```

VALIDAÇÃO CRUZADA

Função de Divisão da Base em k Conjuntos – Saída

```
**** Dataset: CMCT_20200503.csv ****
k = 10 , Dataset: 101 SAN e 116 CMD ( 46.54 %  x  53.46 % )

Fold 1: SAN: 10, CMD: 12, Total: 22, Proporção: 45.45%; 54.55%
Fold 2: SAN: 10, CMD: 12, Total: 22, Proporção: 45.45%; 54.55%
Fold 3: SAN: 10, CMD: 12, Total: 22, Proporção: 45.45%; 54.55%
Fold 4: SAN: 10, CMD: 12, Total: 22, Proporção: 45.45%; 54.55%
Fold 5: SAN: 10, CMD: 12, Total: 22, Proporção: 45.45%; 54.55%
Fold 6: SAN: 10, CMD: 12, Total: 22, Proporção: 45.45%; 54.55%
Fold 7: SAN: 10, CMD: 12, Total: 22, Proporção: 45.45%; 54.55%
Fold 8: SAN: 10, CMD: 11, Total: 21, Proporção: 47.62%; 52.38%
Fold 9: SAN: 10, CMD: 11, Total: 21, Proporção: 47.62%; 52.38%
Fold 10: SAN: 10, CMD: 11, Total: 21, Proporção: 47.62%; 52.38%
```

REFERÊNCIAS

- BERGAMASCO, Leila Cristina Carneiro. **Recuperação de imagens cardíacas tridimensionais por conteúdo**. 2013. 134 f. Dissertação (Mestrado em Ciências) - Programa de Pós-graduação em Sistemas de Informação, Escola de Artes, Ciências e Humanidades, Universidade de São Paulo, São Paulo, 2013.

- BERGAMASCO, Leila Cristina Carneiro. **Recuperação de objetos médicos 3D utilizando harmônicos esféricos e redes de fluxo**. 2018. 181 f. Tese (Doutorado em Ciências) - Escola Politécnica, Departamento de Engenharia da Computação e Sistemas Digitais, Universidade de São Paulo, São Paulo, 2018.

- KUMAR, V.; ABBAS, A. K.; FAUSTO, N.; ASTER, J. C.. **Robbins & Cotran – Patologia: Bases Patológicas das Doenças**. 8 ed. Rio de Janeiro: Elsevier, 2010.