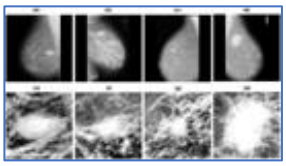


Machine Learning (Processamento, Visualização, Classificação)
(Dataset - indicadores de pacientes com cancro da mama)

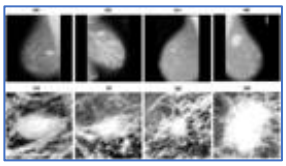
1. Objetivos do Projeto

Pretende-se desenvolver um programa em Python para extrair e visualizar caraterísticas com informação útil de um dataset com "*indicadores de pacientes com cancro da mama*". Terá ainda de explorar e desenvolver diversos modelos de classificadores (i.e., SVM, RF, e ANN/DNN) que permitam predizer (diagnosticar) se um paciente é suspeito ou não de ter cancro da mama. Nesse sentido, deverá aplicar / utilizar métricas apropriadas (e.g., ACCURACY, PRECISION, RECALL, etc.) para selecionar e informar qual é o classificador desenvolvido com maior performance. O dataset consta com 44 variáveis (campos): 16 do tipo inteiro (int), 27 do tipo real (float), e 1 do tipo object (string) como se descreve a seguir:

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 352 entries, 0 to 351
Data columns (total 44 columns):
 #   Column                                Non-Null Count  Dtype
---  ------                                --------------  -----
 0   patient_id                            352 non-null    int64
 1   study_id                              352 non-null    int64
 2   series                                352 non-null    int64
 3   lesion_id                             352 non-null    int64
 4   segmentation_id                       352 non-null    int64
 5   image_view                            352 non-null    int64
 6   mammography_type                      352 non-null    int64
 7   mammography_nodule                    352 non-null    int64
 8   mammography_calcification             352 non-null    int64
 9   mammography_microcalcification        352 non-null    int64
 10  mammography_axillary_adenopathy       352 non-null    int64
 11  mammography_architectural_distortion  352 non-null    int64
 12  mammography_stroma_distortion         352 non-null    int64
 13  age                                   352 non-null    int64
 14  density                               352 non-null    int64
 15  i_mean                                352 non-null    float64
 16  i_std_dev                             352 non-null    float64
 17  i_maximum                             352 non-null    float64
 18  i_minimum                             352 non-null    float64
 19  i_kurtosis                            352 non-null    float64
 20  i_skewness                            352 non-null    float64
 21  s_area                                352 non-null    int64
 22  s_perimeter                           352 non-null    float64
 23  s_x_center_mass                       352 non-null    float64
 24  s_y_center_mass                       352 non-null    float64
 25  s_circularity                         352 non-null    float64
 26  s_elongation                          352 non-null    float64
 27  s_form                                352 non-null    float64
 28  s_solidity                            352 non-null    float64
 29  s_extent                              352 non-null    float64
 30  t_energ                               352 non-null    float64
 31  t_contr                               352 non-null    float64
 32  t_corr                                352 non-null    float64
 33  t_sosvh                               352 non-null    float64
 34  t_homo                                352 non-null    float64
 35  t_savgh                               352 non-null    float64
 36  t_svarh                               352 non-null    float64
 37  t_senth                               352 non-null    float64
 38  t_entro                               352 non-null    float64
 39  t_dvarh                               352 non-null    float64
 40  t_denth                               352 non-null    float64
 41  t_inf1h                               352 non-null    float64
 42  t_inf2h                               352 non-null    float64
 43  classification                        352 non-null    object
dtypes: float64(27), int64(16), object(1)
memory usage: 121.1+ KB
```

Junta-se ainda com este enunciado, alem do dataset, um ficheiro "README.pdf" no qual poderá conhecer detalhes da informação que representa em cada um dos campos (colunas).

O programa consistirá em **desenvolver um interpretador de comandos** que permita ao utilizador extrair diversos tipos de informação e garantir o seu processamento e visualização.



Se for preciso, na implementação dos comandos descritos neste enunciado se podem utilizar / definir outros tipos de dados auxiliares, que considere necessários para a implementação dos comandos.

1.1 Dados de entrada

É disponibilizado um ficheiro de entrada para teste:

"`bcdr_f01_features.csv`"

1.3 Comandos

O programa deverá implementar um total de 12 comandos, que são apresentados a seguir; 2 comandos para carregar os dados, 1 comando para limpeza de dados, 1 comando para sair do programa (aplicação), e 8 comandos para extração, visualização e processamento da informação.

1. LOAD

- Solicita ao utilizador o nome de um ficheiro (dataset) com informação relativa a "*indicadores recolhidos de pacientes com cancro da mama*" e carrega-o em memória na forma de um DataFrame, mostrando informação resumida, nomeadamente; rango de filas (índices), dados das colunas (nome, tipo de dado, e elementos não nulos), assim como a quantidade de memoria ocupada. Deverá ainda imprimir os primeiros 5 e os últimos 5 registos do dataset respetivamente.
- Se o ficheiro não puder ser aberto, escreve "**Ficheiro não encontrado…"** e o DataFrame fica vazio.

2. LOADF

- Abre o ficheiro "bcdr_f01_features.csv" e carrega-o em memória na forma de um DataFrame, mostrando informação resumida, nomeadamente; rango de filas (índices), dados das colunas (nome, tipo de dado, e elementos não nulos), assim como a quantidade de memoria ocupada. Deverá ainda imprimir os primeiros 5 e os últimos 5 registos do dataset respetivamente.
- Se o ficheiro não puder ser aberto, escreve "**Ficheiro não encontrado…"** e o DataFrame fica vazio.

3. CLEAR

- Limpa a informação contida em memória do DataFrame, mantendo a estrutura vazia, e deverá ainda indicar o número de registos que foram descartados.

4. QUIT

- Apaga o DataFrame e termina a execução do programa.

5. DESCRIBE

- Apresenta um resumo (descrição) estatística das colunas de dados numéricos do DataFrame.

- Conta e visualiza o número de total de pacientes do DataFrame com lesões benignas (‘Benign’) e malignas (‘Malign’)

6. SORT

- Este comando tem como objetivo realizar algumas operações necessárias, que irão facilitar posteriormente a criação dos modelos de classificação. Nesse sentido, o comando SORT deverá realizar as seguintes tarefas:
  a) Ordenar os dados do DataFrame pelo campo “`patient_id`” em ordem ascendente.
  b) Verificar se o DataFrame tem campos com valores nulos, e se for o caso deverá apagar todas as filas com valores nulos.
  c) O campo “`classification`” que será usado como label (classe), contem valores de tipo string: ‘`Bening`” e “`Malign`”, para facilitar o desenvolvimento dos modelos de classificação, deverá substituir os valores de este campo ‘`Benign`’ por `0` e ‘`Malign`’ por `1`, como resultado este campo ficará só com valores do tipo inteiro (`0` e `1`).
  d) Visualize o DataFrame ordenado resultante, mas só os campos ‘`patient_id`’ e ‘`classification`’.

7. CORRELATION

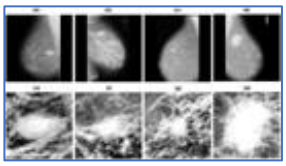
- Este comando permitirá eliminar (apagar) os campos do DataFrame que representam dados dos pacientes que não tem influência no desenvolvimento dos modelos de classificação, nomeadamente os campos: patient_id, study_id, series, lesion_id, segmentation_id, image_view, mammography_type.
- Com o DataFrame resultante deverá calcular a correlação entre as diferentes colunas, que visa conhecer quais são aquelas caraterísticas (features) mas correlacionadas com o campo “`classification`”, que como foi anteriormente dito será usado como label.
- Visualize graficamente os resultados da correlação com recurso a um gráfico de tipo heatmap.

8. SPLITSCALE

- O comando SPLITSCALE tem como objetivo criar os datasets de treino e teste, e ainda escalar as diferentes características (features) do DataFrame. Nesse sentido, se pretende que sejam implementadas as seguintes funcionalidades:
  a) Separar / criar os conjuntos ***X*** e ***y*** para separar o conjunto das features do label (campo “classification”), e logo escalar as features (conjunto ***X***), de forma que os valores de todas as características fiquem no intervalo de valores contínuos [0.0, 1.0]
  b) Com base no ponto anterior, construir os datasets: ***x_train, y_train, x_test, e y_test***, sendo que o dataset de treino deverá conter dados do 80% dos pacientes e do teste o resto (20%), tenha em conta que é normal que existam vários registos de um mesmo paciente no dataset. Portanto, deverá evitar a toda costa esta situação.

9. SVM

- O comando SVM, permitirá desenvolver um modelo de classificador utilizando os conhecimentos adquiridos sob máquinas de suporte vetorial, que permita predizer/diagnosticar se um paciente é suspeito de padecer cancro da mama".



a) Deve utilizar o modelo SVC (vetor de suporte para classificação) utilizando algum dos kernels estudados (e.g., 'rbf' ou 'linear').
b) Teste o modelo de classificação desenvolvido no dataset de teste (***x_test***) que foi anteriormente criado (comando SPLITSCALE).
c) Como resultado deverá imprimir os 5 primeiros e os 5 últimos valores previstos pelo classificador.

10. RANDOMFOREST

- O comando RANDOMFOREST, permitirá desenvolver um modelo de classificador utilizando os conhecimentos adquiridos sob florestas aleatórias, que permita predizer/diagnosticar se um paciente é suspeito ou não de padecer cancro da mama". Deverá utilizar o modelo SVC (vetor de suporte para classificação) utilizando algum dos kernels estudados (e.g., 'rbf' e 'linear').
  a) Teste o modelo de classificação desenvolvido no dataset de teste (***x_test***) que foi anteriormente criado (comando SPLITSCALE).
  b) Como resultado deverá imprimir os 5 primeiros e os 5 últimos valores previstos pelo classificador.

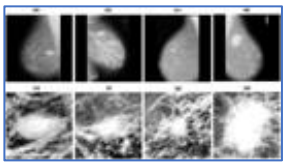
11. ANN

- Similar ao comando anterior, no comando ANN irá criar um modelo de classificação com recurso a redes de neurónios artificiais, poderá usar um modelo com base em MLP – "multi-layer perceptron", ou pode ser também um modelo de deep learning. Pode auxiliar-se das bibliotecas / módulos (scikit-learn) e/ou (tensorflow).
  a) Teste o modelo de classificação desenvolvido no dataset de teste (***x_test***) que foi anteriormente criado (comando SPLITSCALE).
  b) Como resultado devera imprimir os 5 primeiros e os 5 últimos valores previstos pelo classificador.

12. METRICS

- Este comando calcula e visualiza algumas métricas com vistas a avaliar a performance dos modelos de classificação desenvolvidos (SVM, RANDOM FOREST, ANN), nomeadamente:
  a) Matriz de confusão
  b) Accuracy
  c) Precision
  d) Recall

  Com base nas métricas calculadas deverá identificar qual é o modelo de classificação com melhor desempenho.



2. Relatorio e Código

2.1 Código

- ✓ Todo o código será desenvolvido em um dos IDEs "Visual Studio Code" ou "Spyder", a selecionar pela equipa de alunos do projeto. Ainda que não recomendado, será possível também apresentar o código com recurso ao Jupyter Notebook.

- ✓ O código de cada função / comando desenvolvido deverá ser documentado utilizando docstrings.

2.2 Relatório

No relatório deverão constar as seguintes secções (para além de capa com identificação dos alunos e índice):

a) Introdução / Motivação / Objetivos.
b) **Comandos/Funções** - descrição breve dos comandos (algoritmos) implementados;
c) **Limitações** - quais os comandos que apresentam dificuldades ou não foram implementados;
d) **Conclusões** – análise crítica do trabalho desenvolvido.

3. Tabela de Cotações e Penalizações

A avaliação do trabalho será feita de acordo com os seguintes princípios:

- **Estruturação:** o programa deve estar estruturado de uma forma modular e procedimental;
- **Correção:** o programa deve executar as funcionalidades, tal como pedido.
- **Legibilidade e documentação:** o código deve ser escrito, formatado e comentado de acordo com os conhecimentos adquiridos na UC.
- **Desempenho:** Os comandos / algoritmos implementados devem ter em conta a complexidade do mesmo, valorizando-se a implementação de algoritmos com menor complexidade.

A nota final obtida, cuja tabela de cotações se apresenta a seguir, será ponderada de acordo com os princípios acima descritos.

Descrição	Cotação de valores
Menu de opções, leitura de comandos, tratamento de situação de ficheiro inexistente / vazio, limpeza de memória e saída do programa (QUIT)	2
Importação de dados comandos (LOAD， LOADF)	2
Comando CLEAR	1
Comando DESCRIBE	1
Comando SORT	2
Comando CORRELATION	2
Comando SPLITSCALE	1

Comando SVM	2
Comando RANDOMFOREST	2
Comando ANN	2
Comando METRICS	1
Entrega do Relatório e Código	2
Total	20

A seguinte tabela contém penalizações a aplicar:

Descrição	Penalização (valores)
Não separação de funcionalidades	2
Não comentar o programa	1

3. Tabela de Cotações e Penalizações

O não cumprimento das regras a seguir descritas implica uma penalização na nota do projeto. Se ocorrer alguma situação não prevista nas regras a seguir expostas, essa ocorrência deverá ser comunicada ao docente responsável pela UC.

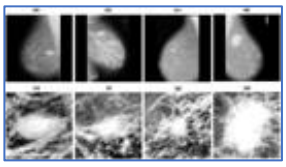
Regras:

a) O Projeto deverá ser elaborado por um grupo de **3 a 4 alunos como máximo**, cada grupo terá que eleger um líder, que será o encarregado de submeter o projeto e distribuir as tarefas.
b) A nota do Projeto será atribuída individualmente a cada um dos elementos do grupo após a discussão. As discussões serão orais, feitas com todos os elementos do grupo presentes em simultâneo.
c) **A apresentação de relatórios ou implementações plagiadas leva à imediata atribuição de nota zero a todos os trabalhos com semelhanças, quer tenham sido o original ou a cópia**.
d) No rosto do relatório e nos ficheiros de implementação deverá constar o número, nome e turma dos autores e o nome do docente.
e) O trabalho deverá ser submetido no Moodle, no link indicado pelo docente criado para o efeito, até às **12:00 horas do dia 22 de Janeiro de 2023**.

Para tal terão de criar uma pasta com o nome: **nomeAluno1_númeroAluno1-nomeAluno2_númeroAluno2-…**, onde colocarão o ficheiro do relatório em formato **pdf** e uma pasta com todo o código desenvolvido no projeto. Os alunos terão de submeter essa **pasta compactada em formato ZIP ou RAR**. Apenas será permitido submeter um ficheiro.

f) Não serão aceites trabalhos entregues que não cumpram na íntegra o ponto anterior.

A data prevista para a discussão do projeto é o dia 24 de janeiro de 2023.



Machine Learning (Processamento, Visualização, Classificação)
(Dataset - indicadores de pacientes com cancro da mama)

(fim de enunciado)

Elaborado por: Miguel A. Guevara Lopez

Data: 11/12/2022