Machine Learning Package

Portfolio de algoritmos de Machine Learning

Sumário

- Ambiente de trabalho: Python3
- Requerimentos: numpy, pandas, scipy, matplotlib
- Repositório modelo: https://github.com/cruz-f/si
- Preparação: https://github.com/cruz-f/si#setup
  - Criar fork do repositório no GitHub para a conta pessoal
  - Clonar o repositório da conta pessoal
  - Instalar requerimentos
  - Alterar a autoria no __*init*__.*py* (src->si->__init__.py)
  - Gravar (git commit) e publicar (git push) alterações
- Créditos: Vítor Pereira pela implementação original

Objeto Dataset

- Na pasta data, adiciona o modulo *dataset.py* que deve conter o objeto *Dataset.*
- *Class Dataset:*
  - Atributos:
    - X – a matriz/tabela de features (variáveis independentes)
    - y – o vetor da variável dependente
    - features – o vetor do nome das features
    - label – o nome do vetor da variável dependente
  - Métodos:
    - shape – dimensões do dataset
    - has_label – verifica se o dataset tem y
    - get_classes – devolve as classes do dataset (valores possíveis de y)
    - get_mean, get_variance, get_median, get_min, get_max – devolve média, variância, mediana, valor mínimo e máximo para cada feature/variável dependente
    - summary – devolve um pandas *DataFrame* com todas as métricas descritivas

io sub-package

- Adiciona agora outro sub-package chamado *io* com dois módulos chamados *csv.py* e *data_file.py.* Vamos adicionar métodos para ler e escrever datasets

- *def read_csv*
  - assinatura/argumentos:
    - filename – nome/caminho do ficheiro
    - sep – separador entre valores
    - features – booleano. O ficheiro tem o nome das features?
    - label – booleano. O ficheiro tem y?
  - ouput esperado:
    - objeto *Dataset*
    - Lê o ficheiro especificado e retorna um Dataset
    - Hint: podem usar packages como pandas.read_csv

io sub-package

- *def write_csv*
  - assinatura/argumentos:
    - filename – nome/caminho do ficheiro
    - dataset – objecto dataset para gravar em ficheiro
    - sep – separador entre valores
    - features – booleano. O ficheiro tem o nome das features?
    - label – booleano. O ficheiro tem y?
  - ouput esperado:
    - escreve o ficheiro especificado com os argumentos indicados
    - Hint: podem usar packages como pandas.write_csv

io sub-package

- *def read_data_file*
  - assinatura/argumentos:
    - filename – nome/caminho do ficheiro
    - sep – separador entre valores
    - label – booleano. O ficheiro tem y?
  - ouput esperado:
    - objeto *Dataset*
    - Lê o ficheiro especificado e retorna um Dataset
    - Hint: podem usar packages como numpy.genfromtxt

io sub-package

- *def write_data_file*
  - assinatura/argumentos:
    - filename – nome/caminho do ficheiro
    - dataset – objecto dataset para gravar em ficheiro
    - sep – separador entre valores
    - label – booleano. O ficheiro tem y?
  - ouput esperado:
    - escreve o ficheiro especificado com os argumentos indicados
    - Hint: podem usar packages como numpy.savetxt

Avaliação

- Exercício 1: NumPy array Indexing/Slicing
  - 1.1) Neste exercício, vamos usar o iris dataset. Carrega o iris.csv usando o método *read* apropriado para o tipo de ficheiro.
  - 1.2) Seleciona a primeira variável independente e verifica a dimensão do array resultante.
  - 1.3) Seleciona as últimas 5 amostras do iris dataset. Qual a média das últimas 5 amostras para cada variável independente/feature?
  - 1.4) Seleciona todas as amostras do dataset com valor superior ou igual a 1. Nota que o array resultante deve ter apenas amostras com valores iguais ou superiores a 1 para todas as features.
  - 1.5) Seleciona todas as amostras com a classe/label igual a 'Iris-setosa'. Quantas amostras obténs?

Avaliação

- Exercício 2: NumPy array Indexing/Slicing
  - 2.1) Adiciona um método ao objeto Dataset que remove todas as amostras que contêm pelo menos um valor nulo (NaN). Nota que o objeto resultante não deve conter valores nulos em para nenhuma feature/variável independente. Nota também que deves atualizar o vetor y removendo as entradas associadas às amostras a remover. Deves usar apenas funções no NumPy.
    Nome do método: *dropna*
  - 2.2) Adiciona um método ao objeto Dataset que substitui todas os valores nulos por outro valor (argumento da função/método). Nota que o objeto resultante não deve conter valores nulos em para nenhuma feature/variável independente. Deves usar apenas funções no NumPy.
    Nome do método: fillna
  - Opcional: Podes adicionar exemplos de como utilizar estes métodos à script do exercício 1.