

Semana 3 - Complementar

Objetivo: Introdução a bibliotecas Python.

Conteúdo: Bibliotecas:

1. Pandas
2. NumPy
3. Seaborn
4. NLTK

Desafio: Responder todos os exercícios propostos neste material



Bibliotecas Complementares

Conceito:

Bibliotecas por definição são pacotes que fornecem funções para que os usuários possam um conjunto de funcionalidades prontas para uso sem ter que se preocupar com outros pacotes. É usual encontrar bibliotecas para as mais diversas tarefas do dia a dia.

Assim como a maioria das linguagens de programação, o Python possui diversas bibliotecas que possibilitam a utilização de scripts comumente utilizados sem ter a necessidade de reescrevê-los.

A especialização e/ou utilização dessas bibliotecas vai variar de acordo com a necessidade do projeto e está fortemente atrelada a função do profissional, exemplificando este fato as bibliotecas que um cientista de dados utiliza normalmente serão diferentes das que um engenheiro ou muitas vezes um analista pode vir a utilizar.

Pandas:



- Pandas é uma biblioteca para uso em Python, open-source e de uso gratuito (sob uma licença BSD), que fornece ferramentas para análise e manipulação de dados.
- Pandas possui duas estruturas básicas: Series e DataFrame
- Series é utilizado para representar séries unidimensionais de informações, como um atributo (i.e. coluna) ou uma instância (i.e. linha)
- DataFrame é utilizado para representar informações bidimensionais, como datasets
- Documentação: <https://pandas.pydata.org/pandas-docs/stable/dsintro.html>

Exercícios – Pandas

Exercício 1

Leia o arquivo actors.csv e faça os seguintes cálculos sobre o conjunto de dados utilizando Pandas

- O ator/atriz com maior número de filmes e o respectivo número de filmes.
- A média do número de filmes.
- O ator/atriz com a maior média por filme.
- O nome do(s) filme(s) mais frequente(s) e sua respectiva frequência.



Exercício 2

Leia o arquivo csv googleplaystore.csv e realize as seguintes atividades sobre o dataset utilizando Pandas:

- Faça um gráfico de barras contendo os top 5 apps por número de instalação.
- Faça um gráfico de pizza (pie chart) mostrando as categorias de apps existentes no dataset de acordo com a frequência em que elas aparecem.
- Mostre qual o app mais caro existente no dataset.
- Mostre quantos apps são classificados como "Mature 17+".
- Mostre o top 10 apps por número de reviews bem como o respectivo número de reviews. Ordene a lista de forma decrescente por número de reviews.

Exercício 3

Leia o arquivo csv wines_SPA.csv e realize as seguintes atividades sobre o dataset utilizando Pandas:

- Mostre qual a vinícola (winery) que possui o maior número de reviews (num_reviews) nos últimos 20 anos.
- Mostre quais são as 5 regiões com a média de preço mais alta e mais baixa.
- A acidez (acidity) varia entre valores até 5, mostre a média de acidez por ano em todo o intervalo existente na coluna ano (year).
- Mostre qual vinho (wine) possui o maior número de safras em todo o intervalo existente na coluna ano (year).



NumPy:

- NumPy é uma biblioteca para cálculo vetorial e matricial disponibilizada em Python.
- Várias outras bibliotecas utilizam NumPy como base para seus cálculos.
- Utilizar NumPy ao invés das estruturas básicas de Python (e.g. listas) apresenta melhorias de desempenho.
- Documentação: <https://numpy.org/doc/stable/>

Exercícios – NumPy

Exercício 1

Crie um array 5x5 com a sequência 1...25 (incluso) e faça a soma dos elementos da diagonal.

Exercício 2

Crie um array 5x5 com a sequência de números pares entre 0...50 (incluso) e faça as seguintes operações:

- Soma das linhas
- Soma das Colunas
- Média dos elementos da última linha
- Média dos elementos da última coluna

Exercício 3

Dado o array 2D abaixo, substitua o seu maior valor por 0.

- `np.random.normal(0, 3, size=(5, 5))`

Exercício 4

A partir das arrays abaixo, caso seja possível, realize o produto das mesmas:

- `np.random.uniform(size=(5, 3))`
- `np.random.uniform(size=(3, 4))`

Exercício 5

Dado o array abaixo, converta-o para um array do tipo `int32`.

- `np.arange(9, dtype=np.float32).reshape(3, 3)`

Exercício 6

Dado o array abaixo, some 10 em todos os elementos da última coluna.

- `np.random.uniform(size=(8, 8))`



Exercício 7

Dados os dois arrays 2x4 abaixo, empilhe ambos formando um novo array 4x4.

- `np.array([[0,3,4,5], [7, 10, 9, 2]])`
- `np.array([[1,4,9,12], [15, 22, 19, 17]])`

Exercício 8

Dado o array definido por `np.arange(25).reshape(5, 5)`, realize as seguintes operações sem utilizar nenhum tipo de loop (for / while):

- A média das linhas 0, 2 e 3.
- A média das colunas 0, 1 e 4.
- A soma dos elementos das duas diagonais.

Exercício 9

Dado o array abaixo, selecione os valores Pares e todos os valores maiores que 8, seu código não deve conter nenhum tipo de loop (for / while):

- `np.arange(25).reshape(5, 5)`

Exercício 10

Dado o array 2D abaixo, substitua os 5 menores valores por -1, seu código não deve conter nenhum tipo de loop (for / while):

- `a = np.random.randint(0, 50, (5, 5))`

