MACHINE LEARNING APLICADO À ANÁLISE DE DADOS

AULA 2 – 16/03/2021

GOOGLE COLAB E FUNDAMENTOS EM PYTHON

APRESENTANDO A FERRAMENTA GOOGLE COLAB:

É uma plataforma online do Google que nos permite criar e executar códigos na linguagem Python.

POR QUE USAR O GOOGLE COLAB?

- Não requer nenhuma instalação ou configuração na sua máquina local
- Executado em nuvem
- Oferece gratuitamente recursos de CPU, GPU e RAM
- Bibliotecas pré-instaladas (TensorFlow, Scikit-learn, Matplotlib)

FUNDAMENTOS EM PYTHON

VARIÁVEIS E OPERADORES

- Para lembrar:
  - Indentação (Tab ou 4 espaços);
  - Em Python, a indexação começa em 0;
  - Comentários (# ou “””...”””);
  - Int: números inteiros, positivos ou negativos;
  - Float: números fracionários, positivos ou negativos;
  - Funções Built-in: type(), print(), int(), etc.

OPERADORES

- Operações com números

+, -, * e /	• Soma, subtração, multiplicação e divisão
% e **	• Módulo e potência
Int() e float()	• Conversão para inteiro e float

OPERADORES

- Operações relacionais

==	• Igualdade
!=	• Diferença
>	• Maior que
<	• Menor que
>=	• Maior que ou igual a
<=	• Menor que ou igual a

VARIÁVEIS

- Para lembrar:

  - Os nomes das variáveis não podem começar com números;

  - Não pode haver espaço no nome;

  - Não é possível usar : ’ ” , <> / | \ ( ) @ # $ % ^ & * ~ - + ! em nome de variáveis;

  -Não se pode usar palavras reservadas como nome de variáveis.

- Para definir o valor de uma variável, utiliza-se o “=“
- Para imprimir o valor de uma variável, utiliza-se a função “print()”

VARIÁVEIS

- Em python, tanto os atributos quanto os métodos são acessados usando ponto (.)

  objeto.atributo

  objeto.método()

  objeto.método(parâmetros)

DEMONSTRAÇÃO NO GOOGLE COLAB

ESTRUTURAS DE DADOS: LISTAS

- As listas são construídas com o uso de colchetes [ ] e vírgulas.

  lista = [item1, item2, item3]

- Algumas funções Built-in em listas:

  - len(lista) – retorna o comprimento da lista
  - max(lista) – retorna o maior valor da lista
  - min(lista) – retorna o menor valor da lista
  - lista.append() – adiciona um novo item ao fim da lista

ESTRUTURAS DE DADOS: DICIONÁRIOS

- Os dicionários são construídos com o uso de chaves { } e vírgulas. Um dicionário em Python consiste em uma chave e, em seguida, um valor associado.

```
dicionario = {chave1: valor1, chave2: valor2}
```

ESTRUTURAS DE DADOS: TUPLAS

- As tuplas são estruturas bastante semelhantes às listas, porém são imutáveis.
- As tuplas são construídas com o uso de parênteses ( ) e vírgulas.

  tupla = (item1, item2, item3)

- Tuplas não suportam:

  - tupla.append()

  - del tupla[ ]

  - atribuição de um novo item

- Em caso de erro, para alterar um item da dupla, basta convertê-la em uma lista, fazer a alteração desejada e em seguida convertê-la novamente para uma tupla.

DEMONSTRAÇÃO NO GOOGLE COLAB

CONDICIONAIS: IF/ELSE

- O condicional if nos permite fazer com que programa execute determinada tarefa com base em uma ou mais condições.

```
if (condição1):
        print("comando executado caso a condição1 seja Verdadeira")
else:
        print("comando executado caso a condição1 seja Falsa")
```

CONDICIONAIS: ELIF

- O condicional “elif” pode ser utilizado para evitar muitos “ifs” aninhados.

```
if (condição1):
        print(“comando executado caso a condição1 seja Verdadeira”)
elif (condição2):
        print(“comando executado caso a condição2 seja Verdadeira”)
...
else:
        print(“comando executado caso nenhuma condição seja Verdadeira”)
```

ESTRUTURAS DE REPETIÇÃO: LOOP FOR

- A estrutura For valida cada item em uma série de valores. É utilizada quando desejamos executar um comando ou conjunto de comandos um número determinado de vezes.

```
for item in série-de-itens:
    executar comandos
```

- Podemos utilizar o loop for em objetos sequenciais, tipo: strings, listas, tuplas, elementos de dicionários e arquivos.

ESTRUTURAS DE REPETIÇÃO: WHILE

- A instrução definida em uma estrutura while será executada repetidamente enquanto uma condição for verdadeira.

```
while(condição1):
    print("comando executado caso a condição1 seja verdadeira")
```

Obs.: a condição1 precisa deixar de ser verdadeira em algum momento para que o loop não fique executando infinitamente.

- Alternativas para interromper a execução do loop while: pass, break e continue

DEMONSTRAÇÃO NO GOOGLE COLAB

LINKS ÚTEIS:

- Funções embutidas — documentação Python 3.9.1

  (https://docs.python.org/pt-br/3/library/functions.html)

REFERÊNCIAS BIBLIOGRÁFICAS

- Anaconda Python (datascienceacademy.com.br) (https://www.datascienceacademy.com.br/path-player?courseid=python-fundamentos&unit=5722d08747d7dd30ac8b457bUnit)
- Python for Data Science and Machine Learning Bootcamp | Udemy (https://www.udemy.com/course/python-for-data-science-and-machine-learning-bootcamp/learn/lecture/5733280?start=225#overview)