

Prova P1 | 2021

PUC-SP | Ciência de Dados e Inteligência Artificial | Algoritmos e Programação

Em toda a prova, utilize suas próprias palavras. Prova individual, com consulta.

RA:

Nome:

01) Explique qual a relação entre variáveis e tipos de dados primitivos. Quais são os tipos de dados primitivos em Python? Explique cada um deles.

Sua resposta

02) A série de Fettuccine pode ser descrita como similar à de Fibonacci. A relação de recorrência para seus elementos é dada a seguir.

$$Fet_n = Fet_{n-3} * Fet_{n-2} * Fet_{n-1}, \text{ para } n \geq 3$$

$$Fet_0 = 1$$

$$Fet_1 = 1$$

$$Fet_2 = 2$$

Escreva um programa para calcular Fet_n .

```
In [ ]: # Seu código aqui
```

03) Escreva um programa que some todos os divisores de um número n (excluindo ele mesmo), que deve ser informado pelo usuário.

```
In [1]: # Seu código aqui
```

04) Em muitas situações é útil criar um dataset com dados aleatórios para fins de validação de prova de conceito. Durante o curso, criamos dois datasets assim. Na lista preparatória para esta prova, o código implementado na função `obter_dados_pesq_campo(n_livros)` cria uma lista `livros`, que por sua vez contém listas com alguns atributos relacionados à produção de livros. Naquela questão, cada livro é descrito pela lista: `[n_identificação_livro, n_vendas_prováveis, custo_fixo, n_páginas]`. Diz-se que cada uma destas listas é uma instância deste dataset. Python implementa o conceito de matrizes com listas contendo outras listas. Um outro caso refere-se à lista de pesos de jogadores de futebol (semana 06). Neste caso, uma lista unidimensional é suficiente, mas cada peso de jogador foi gerado aleatoriamente.

Nesta questão, você deverá gerar um dataset com dados aleatórios. Faz parte da sua tarefa criar um contexto de uso, assim como feito com o dataset do custo de produção

de livros ou dos pesos dos jogadores. Qualquer contexto é válido e exemplos incluem o hospitalar, financeiro, políticas públicas, educacional etc. Embora os dados seja aleatórios, os intervalos devem ser realistas, tanto quanto possível. Por exemplo, idades devem ser compatíveis com idades dos seres humanos bem como pesos e alturas. Salários como R\$ 0,56 por mês não fazem sentido, e seu dataset deve evitar ao máximo estas ocorrências.

Você pode utilizar as bibliotecas como `random` e a `numpy.random.normal(média, desvio_padrao, 1)`, mas você só pode gerar um único valor por vez, muito embora estas bibliotecas ofereçam alternativas para se criar distribuições com um único comando. Assim como acontece no dataset do custo de produção de livros, cada instância do seu dataset deve conter ao menos 4 atributos (valores).

```
[id da instância, valor 1, valor 2, valor 3, ..., valor n]
```

Ao final, espera-se o dataset que você criou. Leia a próxima questão antes de começar a trabalhar nesta.

Contexto

Explique o contexto aqui.

Descrição do dataset

- `id` : identificação da
- `valor 1` : atributo 1 da
- `valor 2` : atributo 2 da
- ...
- `valor n` : atributo n da

```
In [ ]: # Seu código aqui
```

05) Utilizando o laço `while` e o método `pop()`, que está disponível para as listas, remova todas as identificações das instâncias do dataset que você criou. Por exemplo, um dataset possível de custos de produção de livros de tamanho 5 poderia ser:

```
[[1, 98296, 458.41, 266],  
 [2, 89354, 9264.77, 297],  
 [3, 44216, 6418.27, 752],  
 [4, 63999, 6195.78, 727],  
 [5, 78403, 3152.79, 347]]
```

em que os valores de 1 a 5 da primeira coluna representam as identificações das instâncias (livros, neste caso). Ao final da remoção, o dataset deveria ficar como a seguir:

```
[[98296, 458.41, 266],  
 [89354, 9264.77, 297],  
 [44216, 6418.27, 752],
```

```
[63999, 6195.78, 727],  
[78403, 3152.79, 347]]
```

In [2]: *# Seu código aqui*