

# Matéria: Tecnologia da Informação

## Assunto: Programação Python

---

### Resumo Teórico do Assunto

Para resolver as questões de Python e Aprendizado de Máquina apresentadas, é fundamental compreender os seguintes conceitos:

---

### 1. Fundamentos de Python e a Biblioteca NumPy

Python é uma linguagem de programação de alto nível, interpretada, de propósito geral, conhecida por sua sintaxe clara e legibilidade. Para operações numéricas e científicas, a biblioteca **NumPy** (Numerical Python) é indispensável.

#### # 1.1. NumPy Arrays (ndarrays)

- **Definição:** O tipo de dado central do NumPy é o `ndarray` (N-dimensional array), que é um array homogêneo (todos os elementos são do mesmo tipo) e multidimensional. Diferente das listas Python, os `ndarrays` são otimizados para operações numéricas e consomem menos memória para grandes volumes de dados.

- **Criação:** Arrays NumPy são geralmente criados a partir de listas ou tuplas Python usando a função `np.array()`.

- \* Exemplo de array 1D (vetor): `np.array([1, 2, 3, 4])`

- \* Exemplo de array 2D (matriz): `np.array([[1, 2, 3], [4, 5, 6]])`

#### # 1.2. Operações Element-wise (Elemento a Elemento)

- Uma das grandes vantagens do NumPy é a capacidade de realizar operações matemáticas em arrays de forma **elemento a elemento**, sem a necessidade de loops explícitos.

- Quando você aplica operadores aritméticos (`+`, `-`, `*`, `/`, etc.) entre dois `ndarrays` de dimensões compatíveis, a operação é realizada entre os elementos correspondentes de cada array. Se um dos operandos for um escalar, a operação é aplicada a cada elemento do array.

- **Exemplo:** Se `a = np.array([1, 2])` e `b = np.array([3, 4])`, então `a * b` resultará em `np.array([3, 8])`. Se `c = 2`, então `a * c` resultará em `np.array([2, 4])`.

- **Relevância para as questões:** A questão 43 explora diretamente este conceito, onde a multiplicação de dois arrays NumPy resulta em um novo array com os produtos elemento a elemento.

#### # 1.3. Indexação Booleana (Fancy Indexing)

- NumPy permite selecionar elementos de um array usando um array de booleanos (True/False) como índice.

- Quando um array booleano é usado para indexar outro array, apenas os elementos

correspondentes a `True` no array booleano são selecionados.

- **Característica Importante:** O resultado da indexação booleana é sempre um **array 1D (achatado)**, contendo todos os elementos que satisfizeram a condição, independentemente da dimensão original do array.

- **Exemplo:** Se `a = np.array([[1,2],[3,4]])` e você faz `a[a > 2]`, o resultado será `np.array([3, 4])`.

- **Relevância para as questões:** A questão 42 testa a compreensão da indexação booleana e o formato do array resultante.

---

## 2. Aprendizado de Máquina (Machine Learning)

Aprendizado de Máquina é um campo da inteligência artificial que permite aos sistemas aprenderem a partir de dados, identificar padrões e tomar decisões com mínima intervenção humana.

### # 2.1. Tipos de Aprendizado de Máquina

- **Aprendizado Supervisionado:**

- \* **Definição:** O algoritmo aprende a partir de um conjunto de dados "rotulados", ou seja, dados que já possuem a resposta correta (o "rótulo" ou "saída" esperada) associada a cada entrada. O objetivo é aprender um mapeamento entre as entradas e as saídas.

- \* **Subtipos:**

- \* **Classificação:** O objetivo é prever uma **categoria discreta** ou um rótulo para uma nova entrada. Exemplos incluem: classificar e-mails como spam/não spam, identificar se uma imagem contém um gato ou um cachorro, ou determinar o risco (alto, médio, baixo) de um terreno.

- \* **Regressão:** O objetivo é prever um **valor contínuo** (numérico). Exemplo: prever o preço de uma casa com base em suas características.

- **Aprendizado Não Supervisionado:**

- \* **Definição:** O algoritmo trabalha com dados "não rotulados", buscando padrões, estruturas ou agrupamentos intrínsecos nos dados sem ter uma saída pré-definida para aprender.

- \* **Subtipos:**

- \* **Clustering (Agrupamento):** Agrupar pontos de dados semelhantes em clusters. Exemplos: segmentação de clientes, agrupamento de documentos.

- \* **Redução de Dimensionalidade:** Reduzir o número de variáveis (características) em um conjunto de dados, mantendo a maior parte da informação.

### # 2.2. Algoritmos de Aprendizado de Máquina Relevantes (para as opções da questão 44)

- **K-Nearest Neighbors (K-NN):**

- \* **Tipo:** Algoritmo de **Aprendizado Supervisionado** para **Classificação** (e também

Regressão).

\* **Princípio:** Classifica um novo ponto de dados com base na classe majoritária de seus "K" vizinhos mais próximos no espaço de características. É um algoritmo "lazy" (preguiçoso) porque não constrói um modelo explícito durante a fase de treinamento, apenas armazena os dados de treinamento.

\* **Relevância para as questões:** A questão 44 descreve um problema clássico de classificação (prever um rótulo discreto - risco alto, médio, baixo ou nenhum - a partir de características), tornando o K-NN uma opção adequada.

• **Outros Algoritmos Mencionados (e seus tipos):**

\* **PCA (Principal Component Analysis):** Algoritmo de **redução de dimensionalidade** (Aprendizado Não Supervisionado).

\* **DBSCAN (Density-Based Spatial Clustering of Applications with Noise):** Algoritmo de **clustering** (Aprendizado Não Supervisionado).

\* **K-Medoids:** Algoritmo de **clustering** (Aprendizado Não Supervisionado), similar ao K-Means, mas usa pontos de dados reais (medoids) como centros de cluster.

\* **Redes de Kohonen (Self-Organizing Maps - SOM):** Algoritmo de **clustering** e visualização de dados (Aprendizado Não Supervisionado), que mapeia dados de alta dimensão para uma grade de baixa dimensão.

---

Compreender esses conceitos permitirá analisar os códigos Python com NumPy e identificar o tipo de problema de Aprendizado de Máquina, escolhendo o algoritmo apropriado.

---

## Questões de Provas Anteriores

Fonte: [escrituario\\_agente\\_de\\_tecnologia \(1\).pdf](#), Página: 14

pcimarkpci MjgwNDowMTRkOjE0YTU6OTI1ODozOGQ2OjNhMGM6NTM0MzplZmI1:U3V  
uLCAYNyBKdWwgMjAyNSAyMzo0Nzo0MCAtMDMwMA==  
[www.pciconcursos.com.br](http://www.pciconcursos.com.br)

14

BANCO DO BRASIL

AGENTE DE TECNOLOGIA - Microrregião 16 DF-TI GABARITO 1

42

Considere o código Python a seguir.

```
import numpy as np
```

```
a = np.array([[1,2,3],[4,5,6],[7,8,9]])
```

```
print (a[a>5])
```

O que será exibido após esse código ser executado?

- (A) [[False False False]  
[False False True]  
[True True True]]
- (B) [[False False False]  
[False False 6]  
[7 8 9]]
- (C) [[  
[6]  
[7 8 9]]
- (D) [False False False False False 6 7 8 9]
- (E) [6 7 8 9]
- 43

Considere o código Python a seguir.

```
import numpy as np
valorAplicado = np.array([5000, 6000, 7000, 8000])
taxaJuros = np.array([1, 2, 3, 4])
resultado = valorAplicado * taxaJuros
```

Ao executar o código acima, o valor final da variável resultado será

- (A) 70000
- (B) 260000
- (C) [ 5000 12000 21000 32000]
- (D) [[ 5000 10000 15000 20000]  
[ 6000 12000 18000 24000]  
[ 7000 14000 21000 28000]  
[ 8000 16000 24000 32000]]
- (E) [[ 5000 6000 7000 8000]  
[10000 12000 14000 16000]  
[15000 18000 21000 24000]  
[20000 24000 28000 32000]]
- 44

Um pesquisador conseguiu uma base de dados que mostrava terrenos classificados de acordo com:

- características físicas;
- tipo de negócio a ser nele implantado;
- risco esperado, que compreendia os rótulos alto, médio, baixo ou nenhum.

Decidiu, então, usar um algoritmo de aprendizado de máquina que, a partir das características físicas do terreno e do tipo de negócio a ser nele implantado, aprenderia a determinar o risco esperado, enquadrando o terreno em questão em um daqueles rótulos.

Nesse cenário, que algoritmo de aprendizado de máquina é indicado para resolver esse problema?

- (A) PCA (B) K-NN (C) DBSCAN (D) K-Medoids (E) Redes de Kohonen