



CURSO: <BÁSICO EM MACHINE LEARNING>

- **Atividade 02 (ATIV-02)**

- Tipo: Somativa;
- Tema: Algoritmos básicos de programação em linguagem python e análise de dados.
- Conteúdo: Módulo 1 e 2.
- Participantes: Individual.
- Avaliação do aluno.
 - Objetivo: Avaliar desempenho do aluno sobre conhecimentos básicos de programação em linguagem python e visualização e análise de dados.
 - Nota: 0 a 3 supercrítico, 4 a 6 crítico, 5 a 7 razoável e 8 a 10 bom;
 - Critérios avaliados: Respostas com coerência, coesão e com exemplos.
- Informações adicionais: A atividade é composta por 5 questões dissertativas sobre python e 5 questões sobre visualização e análise de dados.
- **AO CONCLUIR A ATIVIDADE: ENVIAR APENAS O LINK DO REPOSITÓRIO GITHUB (ESPECIFICAR A BRANCH) PÚBLICO.**

1. Escreva uma função que receba uma lista de números e retorne outra lista com os números ímpares.

```
def filtrar_impares(lista):  
    return [num for num in lista if num % 2 != 0]
```

2. Escreva uma função que receba uma lista de números e retorne outra lista com os números primos presentes.

```
def eh_primo(n):  
    if n < 2: return False  
    for i in range(2, int(n**0.5)+1):  
        if n % i == 0:  
            return False  
    return True
```

```
def filtrar_primos(lista):  
    return [num for num in lista if eh_primo(num)]
```

3. Escreva uma função que receba duas listas e retorne outra lista com os elementos que estão presentes em apenas uma das listas.

```
def elementos_unicos(lista1, lista2):  
    return list(set(lista1) ^ set(lista2))
```

4. Dada uma lista de números inteiros, escreva uma função para encontrar o segundo maior valor na lista.

```
def segundo_maior(lista):  
    if len(set(lista)) < 2:  
        return None  
    return sorted(set(lista))[-2]
```

5. Crie uma função que receba uma lista de tuplas, cada uma contendo o nome e a idade de uma pessoa, e retorne a lista ordenada pelo nome das pessoas em ordem alfabética.

```
def ordenar_por_nome(pessoas):  
    return sorted(pessoas, key=lambda x: x[0])
```

6. Como identificar e tratar outliers em uma coluna numérica usando desvio padrão ou quartis?

Outliers são aqueles valores que fogem muito do padrão dos dados e podem atrapalhar as análises. A gente consegue encontrar esses “valores fora da curva” usando dois jeitos bem comuns: desvio padrão ou quartis. Com o desvio padrão, primeiro a gente calcula a média da coluna e vê o quanto os valores se afastam dessa média. Depois, considera outlier tudo que estiver muito distante — mais de 3 vezes esse desvio, pra mais ou pra menos. Já com os quartis, a ideia é dividir os dados em partes: pegamos o valor que representa os 25% menores (Q1) e os 75% maiores (Q3), e calculamos a diferença entre eles (isso é o IQR). Se algum valor estiver muito abaixo ou muito acima dessa faixa (fora



de $Q1 - 1,5 \times IQR$ ou $Q3 + 1,5 \times IQR$), ele é um outlier. Depois de encontrar esses valores, dá pra escolher se você quer remover, substituir por algo mais comum (como a média ou a mediana) ou manter, caso eles façam sentido no contexto.

7. Como concatenar vários DataFrames (empilhando linhas ou colunas), mesmo que tenham colunas diferentes?

Dica: Utiliza-se `pd.concat()` especificando `axis=0` (linhas) ou `axis=1` (colunas). Quando há colunas diferentes, os valores ausentes são preenchidos com `NaN`.

```
import pandas as pd
```

```
df_concat = pd.concat([df1, df2], axis=0)
df_concat_colunas = pd.concat([df1, df2], axis=1)
```

8. Utilizando pandas, como realizar a leitura de um arquivo CSV em um DataFrame e exibir as primeiras linhas?

```
import pandas as pd
```

```
df = pd.read_csv('arquivo.csv')
print(df.head())
```

9. Utilizando pandas, como selecionar uma coluna específica e filtrar linhas em um “DataFrame” com base em uma condição?

```
filtro = df[df['idade'] > 30]
coluna_filtrada = filtro['idade']
```

10. Utilizando pandas, como lidar com valores ausentes (NaN) em um DataFrame?

```
df.isnull()
df_sem_nan = df.dropna()
df_preenchido = df.fillna(0)
df['coluna'] = df['coluna'].fillna(df['coluna'].mean())
```