

Nome: Emilly Mickeli Depine da Silva | **RA:** 23025480

Nome: Renan Teixeira Pinheiro | **RA:** 23025274

Nome: Fernando José dos Santos | **RA:** 23025299


Nome: Gustavo Henrique Santos Araujo | **RA:** 23025397

Inteligência Artificial e Machine Learning

```
1 !pip install pandas openpyxl
```

 **Mostrar saída oculta**

```
1 import pandas as pd
2 import os
3
4 file_path = '/content/transporte_dados.xlsx'
5
6 if not os.path.exists(file_path):
7     print(f"Erro: O arquivo {file_path} não foi encontrado.")
8     print("Certifique-se de que o arquivo foi carregado corretamente no Google Colab.")
9 else:
10     try:
11         df = pd.read_excel(file_path)
12         print("Arquivo carregado com sucesso!")
13         print(df.head())
14     except Exception as e:
15         print(f"Erro ao ler o arquivo: {e}")
```

 Arquivo carregado com sucesso!


	Origem	Destino	Valor	Tipo de Transporte	Preço	Tempo
0	Zona Leste	Zona Sul	33.71	99	33.71	12
1	Zona Norte	Hospital	18.69	Taxi	18.69	51
2	Estádio	Zona Sul	37.81	Uber	37.81	36
3	Zona Sul	Estádio	27.70	Taxi	27.70	38
4	Zona Norte	Zona Oeste	47.20	Transporte Público	47.20	47

```
1 def valor_mais_em_conta(df):
2     df_sorted = df.sort_values(by='Preço')
3     return df_sorted.iloc[0]

1 def menor_tempo(df):
2     df_sorted = df.sort_values(by='Tempo')
3     return df_sorted.iloc[0]

1 def transporte_recomendado(df):
2     df['Pontuação'] = (df['Preço'] * 0.6) + (df['Tempo'] * 0.4)
3     df_sorted = df.sort_values(by='Pontuação')
4     return df_sorted.iloc[0]
```

```
1 opcao_mais_barata = valor_mais_em_conta(df)
2 print("Opção mais em conta:")
3 print(opcao_mais_barata)
4 print("\n")
5
6 opcao_menor_tempo = menor_tempo(df)
7 print("Opção com menor tempo:")
8 print(opcao_menor_tempo)
9 print("\n")
10
11
12 opcao_recomendada = transporte_recomendado(df)
13 print("Transporte mais recomendado:")
14 print(opcao_recomendada)
```

 Opção mais em conta:

	Zona Sul
Origem	Zona Norte
Destino	
Valor	10.44
Tipo de Transporte	Transporte Público
Preço	10.44
Tempo	24

Name: 32, dtype: object

Opção com menor tempo:

	Zona Sul
Origem	Zona Leste
Destino	
Valor	15.5
Tipo de Transporte	99
Preço	15.5
Tempo	11

Name: 46, dtype: object

Transporte mais recomendado:

	Estádio
Origem	Centro
Destino	
Valor	15.25

```
Tipo de Transporte      99
Preço                  15.25
Tempo                   11
Pontuação              13.55
Name: 18, dtype: object
```

Introdução:

Este documento descreve um código em Python desenvolvido para **automatizar a análise de dados de transporte** a partir de um arquivo Excel (transporte_dados.xlsx). O código utiliza a biblioteca pandas para ler e processar os dados, e aplica algoritmos simples para identificar:

1. **A opção mais barata** (menor preço).
2. **A opção mais rápida** (menor tempo de viagem).
3. **A opção mais recomendada** (combinação de preço e tempo).

A seguir, detalhamos cada parte do código e sua utilidade no dia a dia.

1. Carregar os Dados

- **O que faz?**
 - Carrega o arquivo Excel (transporte_dados.xlsx) em um DataFrame do pandas, que é uma estrutura de dados tabular (como uma tabela).
 - O DataFrame permite manipular e analisar os dados de forma eficiente.
 - **Por que é útil?**
 - Automatiza a leitura de dados de um arquivo Excel, eliminando a necessidade de abrir o arquivo manualmente e copiar/colar dados.
-

2. Função para Encontrar a Opção Mais Barata

- **O que faz?**
 - Ordena as opções de transporte pelo preço (do menor para o maior).
 - Retorna a primeira linha (ou seja, a opção mais barata).
 - **Por que é útil?**
 - Identifica rapidamente a opção mais econômica, o que é ideal para quem deseja economizar dinheiro.
-

3. Função para Encontrar a Opção Mais Rápida

- **O que faz?**
 - Ordena as opções de transporte pelo tempo de viagem (do menor para o maior).
 - Retorna a primeira linha (ou seja, a opção mais rápida).
- **Por que é útil?**

- Identifica a opção mais rápida, ideal para quem está com pressa ou precisa otimizar o tempo.

4. Função para Encontrar a Opção Mais Recomendada

- **O que faz?**
 - Cria uma pontuação para cada opção de transporte, combinando preço e tempo (com pesos de 60% para preço e 40% para tempo).
 - Ordena as opções pela pontuação (do menor para o maior).
 - Retorna a primeira linha (ou seja, a opção mais recomendada).
- **Por que é útil?**
 - Oferece uma recomendação equilibrada, considerando tanto o custo quanto o tempo de viagem. Isso é ideal para quem busca um equilíbrio entre economia e conveniência.

5. Aplicar as Funções e Exibir os Resultados

- **O que faz?**
 - Aplica as três funções (`valor_mais_em_conta`, `menor_tempo`, `transporte_recomendado`) ao `DataFrame`.
 - Exibe os resultados no console.
- **Por que é útil?**
 - Fornece uma análise rápida e clara das melhores opções de transporte, sem a necessidade de analisar manualmente os dados.

6. Salvar os Resultados (Opcional)

- **O que faz?**
 - Cria um novo `DataFrame` com os resultados das análises.
 - Salva esse `DataFrame` em um novo arquivo Excel (`resultados_transporte.xlsx`).
 - **Por que é útil?**
 - Permite compartilhar os resultados com outras pessoas ou usá-los em relatórios.
-

Aplicações no Dia a Dia

1. Tomada de Decisão Rápida:

- O código ajuda a escolher a melhor opção de transporte com base em critérios como preço, tempo e uma combinação dos dois. Isso é útil para planejar viagens ou deslocamentos diários.

2. Automatização de Tarefas Repetitivas:

- Se você precisa analisar dados de transporte com frequência, o código elimina a necessidade de fazer cálculos manuais ou abrir planilhas repetidamente.

3. Economia de Tempo e Dinheiro:

- Ao identificar a opção mais barata ou mais rápida, você pode economizar tempo e dinheiro em suas viagens.

4. Personalização:

- O código pode ser adaptado para incluir outros critérios, como conforto, disponibilidade de horários ou preferências pessoais.

5. Uso em Empresas:

- Empresas de logística ou transporte podem usar esse tipo de análise para otimizar rotas, reduzir custos e melhorar a eficiência operacional.

Exemplo de Aplicação:

Imagine que você está planejando uma viagem de São Paulo para o Rio de Janeiro e tem várias opções de transporte (ônibus, avião, carro, etc.). Com esse código, você pode:

1. **Identificar a opção mais barata** para economizar dinheiro.
2. **Escolher a opção mais rápida** se estiver com pressa.
3. **Obter uma recomendação equilibrada** que considera tanto o preço quanto o tempo.

Isso torna o processo de decisão muito mais rápido e eficiente!
