| | |
|---|---|
|  | **UNIVERSIDADE ESTÁCIO DE SÁ**<br>POLO CENTRO – SANTA ROSA – RS<br><br>**DESENVOLVIMENTO FULL STACK**<br>Relatório da Trabalho Prático – Mundo 5 |
| **Disciplina:** | DGT2823 Tecnologias para desenv. de soluções de big data |
| **Aluno/Matrícula:** | Anderson Rech - 202304442215 |
| **Turma:** | 2023.2 |
| **Repositório:** | https://github.com/4nderech/Mundo5-Trabalho-Pratico |

## DGT2823 – Tecnologias para desenvolvimento de soluções de Big Data

## Objetivos da prática

- Descrever como ler um arquivo CSV usando a biblioteca Pandas (Python);
- Descrever como criar um subconjunto de dados a partir de um conjunto existente
- usando a biblioteca Pandas (Python);
- Descrever como configurar o número máximo de linhas a serem exibidas na
- visualização de um conjunto de dados usando a biblioteca Pandas (Python);
- Descrever como exibir as primeiras e últimas “N” linhas de um conjunto de dados
- usando a biblioteca Pandas (Python); Descrever como exibir informações gerais
- sobre as colunas, linhas e dados de um conjunto de dados usando a biblioteca
- Pandas (Python);

Através dessa atividade o aluno realizará a limpeza de um conjunto de dados, tornando-o apto a ser usado em tarefas de mineração/análise de dados.

## Contextualização

Como Analista de Dados, você recebeu, em um novo projeto, um conjunto de dados. Sua principal tarefa é tratar os dados desse conjunto a fim de que possam ser utilizados para a descoberta de conhecimento através de sua posterior análise e interpretação. Para tal tarefa, você deverá utilizar a linguagem Python e a biblioteca Pandas. O passo-a-passo de todo o processo de tratamento dos dados é apresentado a seguir, no roteiro de prática.

## MICROATIVIDADE 01

```python
# Microatividade 01
import pandas as pd

# Lendo o arquivo CSV
df = pd.read_csv("dados.csv", delimiter=";", engine="python", encoding="utf-8")

# Exibindo os dados do arquivo
print(df)
```

*SNIPPET DE CÓDIGO DA MICROATIVIDADE 01.*

## MICROATIVIDADE 02

```python
# Microatividade 02
import pandas as pd

# Lendo o arquivo CSV
df = pd.read_csv("dados.csv", delimiter=";", engine="python", encoding="utf-8")

# Exibindo as 3 primeiras colunas
subset = df[["ID", "Duration", "Date"]]
print(subset)
```

*SNIPPET DE CÓDIGO DA MICROATIVIDADE 02.*

## MICROATIVIDADE 03

```python
# Microatividade 03
import pandas as pd

# Lendo o arquivo CSV
df = pd.read_csv("dados.csv", delimiter=";", engine="python", encoding="utf-8")

# Exibindo os dados
pd.set_option("display.max_rows", 9999)
print(df.to_string())
```

*SNIPPET DE CÓDIGO DA MICROATIVIDADE 03.*

## MICROATIVIDADE 04

```python
# Microatividade 04
import pandas as pd

# Lendo o arquivo CSV
df = pd.read_csv("dados.csv", delimiter=";", engine="python", encoding="utf-8")

# Exibindo as 10 primeiras linhas do arquivo
print("Primeiras 10 linhas:")
print(df.head(10))

# Exibindo as 10 ultimas linhas do arquivo
print("\nÚltimas 10 linhas:")
print(df.tail(10))
```

*SNIPPET DE CÓDIGO DA MICROATIVIDADE 04.*

## MICROATIVIDADE 05

```python
# Microatividade 05
import pandas as pd

# Lendo o arquivo CSV
df = pd.read_csv("dados.csv", delimiter=";", engine="python", encoding="utf-8")

# Exibindo os dados gerais
print("Informações gerais:")
print(df.info())
```

*SNIPPET DE CÓDIGO DA MICROATIVIDADE 05.*

# TRABALHO PRÁTICO

**Resumo da Tarefa**

Nesta atividade, foi realizado o tratamento e a manipulação de um conjunto de dados fornecido em formato CSV, contendo as colunas ID, Duration, Date, Pulse, Maxpulse e Calories.

As etapas principais foram:

- Criação de um novo script para leitura do arquivo CSV, com atenção a parâmetros como separador, encoding e engine.
- Importação dos dados para uma variável e verificação da integridade do conjunto.
- Criação de uma cópia do conjunto original para aplicar alterações sem modificar os dados originais.
- Substituição de valores nulos: na coluna 'Calories' por 0 e na coluna 'Date' inicialmente por "1900/01/01", posteriormente ajustado para 'NaN' devido a incompatibilidade de formato.
- Conversão da coluna 'Date' para o tipo 'datetime', resolvendo erros de formato específicos, incluindo a transformação do valor "20201226" para o formato correto.
- Remoção dos registros que permaneciam com valores nulos após as transformações.
- Impressão e verificação do dataframe para garantir que todas as alterações foram aplicadas corretamente.

O resultado foi um conjunto de dados limpo e pronto para análises futuras, com todas as colunas corretamente formatadas e sem valores ausentes.

Link do repositório no GitHub:

https://github.com/4nderech/Mundo5-Trabalho-Pratico-DGT2823.git

```python
# Trabalho Prático | DGT2823 Tecnologias para desenv. de soluções de big data
# Dev Full Stack - Estacio
print("\nTrabalho Prático | DGT2823 Tecnologias para desenv. de soluções de big data")
print("Desenvolvimento Full Stack - Faculdade Estacio de Sa")

# Importar bibliotecas necessarias
import pandas as pd

# Ler o CSV (ajuste o caminho, separador e encoding se necessário)
df = pd.read_csv('dados.csv', sep=';', engine='python', encoding="utf-8")
print("\nDataframe Original e Informações Gerais pré-tratamento:")
print(df)
print(df.info())

# Criar variavel e atribuir a cópia do arquivo original
copy_df = df.copy()

# Corrigir valores na coluna 'Calories'
copy_df['Calories'] = copy_df['Calories'].astype(str).str.extract(r'(\d+)')
copy_df['Calories'] = pd.to_numeric(copy_df['Calories'], errors='coerce')
# Substituir valores ausentes em 'Calories' por 0
copy_df['Calories'] = copy_df['Calories'].fillna(0)

# Corrigir os valores da coluna 'Date'
copy_df['Date'] = copy_df['Date'].astype(str).str.replace("'", "")
# Substituir valores ausentes em 'Date'
copy_df['Date'] = copy_df['Date'].fillna('1900/01/01')
# Corrigir formato incorreto de datas ('20201226')
copy_df['Date'] = copy_df['Date'].replace('20201226', '2020/12/26')
# Converter coluna 'Date' para datetime
copy_df['Date'] = pd.to_datetime(copy_df['Date'], format='%Y/%m/%d', errors='coerce')

# Remover valores nulos restantes
copy_df = copy_df.dropna()

# Imprima o dataframe com os resultados já tratados
print("\nDataFrame final após tratamento dos dados:")
print(copy_df)

print("\nInformações Gerais do DataFrame Tratado:")
print(copy_df.info())
```

*SNIPPET DE CÓDIGO DO TRABALHO PRÁTICO.*