Grupo 3: Serpentes Tech

Equipe: Daniel Brambila, Felipe Martins, Lucas Sandes, Malu Fazendo, Danilo Pinho, Edvaldo Oliveira

---

Visão Geral

Este documento apresenta uma análise crítica e detalhada do projeto submetido no módulo 1 da formação da Globo. O objetivo é avaliar a implementação do sistema de análise de interações com conteúdos, considerando boas práticas de programação, clareza lógica, uso adequado dos recursos da linguagem Python e cumprimento dos requisitos propostos no enunciado.

Itens Atendidos

1. Leitura do arquivo CSV

- Realizada com a função csv_para_lista, com separação clara do cabeçalho e linhas de dados.
- Uso correto do csv.reader com encoding UTF-8.

2. Limpeza de dados

- Implementação da função texto_limpo para remoção de espaços em campos textuais.
- Conversão de watch_duration_seconds com tratamento de erros.
- Segue em boa parte a lógica do gabarito (0 para visualização, None para outros).

3. Estruturação dos dados em dicionários

- Feita com estruturar_interacoes_em_dicionarios de forma robusta, com checagem de consistência entre colunas.

4. Métricas implementadas

- Total de interações por conteúdo (total_interacoes_por_conteudo)
- Contagem por tipo de interação (contar_interacoes_por_tipo)
- Tempo total de visualização (metrica_somar_visualizacao_por_conteudo)
- Média de tempo de visualização (metrica_media_visualizacao_por_conteudo)
- Comentários por conteúdo (obter_comentarios_por_conteudo)
- Top 5 conteúdos mais vistos (calcular_top_5_conteudos_visualizacao)

5. Pipeline automatizado

- A função pipeline() executa todas as etapas de forma sequencial.

Feedbacks de Melhoria

Tratamento e validação

- Poderia haver uma função mais centralizada para tratar todos os campos de forma padronizada.
- input() para ID de comentários quebra a execução automática (idealmente deveria iterar sobre todos os IDs).

Modularização e exibição

- Seria interessante separar a lógica de cálculo da lógica de exibição para fins de reutilização e testes.

Boas práticas gerais

- Algumas funções usam muitos try/except, o que é bom, mas poderiam ser simplificadas.

- Funções de métricas com nomes mais descritivos melhoram a compreensão do código.

Considerações Finais

O projeto mostra um bom entendimento dos princípios de programação funcional e estruturada em Python, com divisão de tarefas, modularização coerente e entrega completa das métricas solicitadas. A abordagem mostra boa capacidade de análise, tratamento de exceções e preocupação com a robustez da execução.

Parabéns pelo trabalho!