

Projeto U.R.S.A - Unity of Resolutions for Softwares and Applications

Objetivo Principal

Apresentação final do projeto U.R.S.A e autocrítica reflexiva.

1. Objetivo do Projeto

Visão Geral: O projeto U.R.S.A (Unity of Resolutions for Softwares and Applications) é um sistema de recomendação focado em melhorar a experiência do usuário em plataformas de e-commerce. O sistema utiliza técnicas de Inteligência Artificial, como Deep Analytics e Machine Learning, para gerar recomendações personalizadas que ajudam as empresas a atrair e reter clientes.

Aplicação Prática: A primeira implementação do projeto foi direcionada a uma startup de e-commerce de livros, onde o sistema foi configurado para analisar dados dos usuários e das avaliações de livros, oferecendo sugestões que melhoram o engajamento e aumentam a satisfação do cliente.

Ferramentas e Tecnologias:

Google Colab: para execução do código e manipulação dos dados.

Pandas: biblioteca para análise e estruturação dos dados.

GitHub: documentação e armazenamento do projeto.

2. Demonstração da Versão Final

Código Explicado: O sistema U.R.S.A funciona através dos seguintes passos principais:

Coleta e Estruturação de Dados: As informações sobre livros, usuários e avaliações são estruturadas em DataFrames, a partir de dicionários iniciais.

Exportação e Carregamento dos Dados: Após a criação dos DataFrames, eles são exportados para arquivos CSV e recarregados para garantir que a análise seja feita a partir de uma base de dados externa.

Cálculo das Médias: O sistema agrupa as avaliações por livro, calcula a média de cada um e exibe os livros ordenados das melhores para as piores avaliações.

Demonstração Prática do Código: Ao rodar o código, os seguintes resultados são exibidos:

As primeiras linhas dos dados, mostrando os detalhes dos livros, usuários e avaliações.

A lista de recomendações de livros, com os títulos e suas médias de avaliação em ordem decrescente.

Demonstração de código para a recomendação baseada em média de avaliações

```
import pandas as pd
```

```
# Carregar dados
```

```
livros = pd.read_csv("livros.csv", sep=";", encoding="utf-8")
```

```
usuarios = pd.read_csv("usuarios.csv", sep=";", encoding="utf-8")
```

```
avaliacoes = pd.read_csv("avaliacoes.csv", sep=";", encoding="utf-8")
```

```
# Exibir primeiros dados para verificar
```

```
print("Dados dos Livros:\n", livros.head())
```

```
print("Dados dos Usuários:\n", usuarios.head())
```

```
print("Dados das Avaliações:\n", avaliacoes.head())
```

```
# Recomendação baseada em média de avaliações
```

```
livros_avalidados = avaliacoes.groupby('id_livros').agg({'avaliacao': 'mean'}).reset_index()
```

```
livros_avalidados = livros_avalidados.rename(columns={'avaliacao': 'media_avaliacao'})
```

```
livros_recomendados = pd.merge(livros, livros_avalidados, on='id_livros')
```

```
livros_recomendados = livros_recomendados.sort_values(by='media_avaliacao',  
ascending=False)
```

```
print("Livros recomendados baseados nas avaliações:\n", livros_recomendados[['titulo',  
'media_avaliacao']]))
```

3. Autocrítica e Reflexão

Eficiência do Algoritmo: O sistema de recomendação com base na média de avaliações funcionou bem para fornecer uma análise inicial.

Ferramentas: A escolha do Google Colab e do Pandas facilitou muito o processo de análise e manipulação de dados.

Preparação dos Dados: A manipulação dos dados para garantir a consistência das informações e o tratamento de valores faltantes foi um ponto desafiador.

Escalabilidade: O sistema funciona bem em pequena escala, mas adaptações seriam necessárias para grandes volumes de dados.

Importância da Estruturação dos Dados: Aprendemos que a qualidade das recomendações depende da forma como os dados são organizados e preparados.

Análise Crítica: Refletimos sobre as diferentes técnicas de recomendação e como elas poderiam ser aplicadas ao sistema para aprimorar os resultados.

Planejamento de Dados: Melhor planejamento das variáveis de entrada poderia permitir uma recomendação mais personalizada.

Exploração de Técnicas Adicionais: Experimentar técnicas de recomendação híbridas (baseadas em conteúdo e colaborativas) para um sistema mais robusto.

4. Planos para o Futuro do Projeto

Funcionalidades Adicionais:

Introdução de novos métodos de recomendação para fornecer uma experiência mais personalizada, como filtragem colaborativa.

Inclusão de recomendações em tempo real, ajustadas ao comportamento atual dos usuários.

Interface e Usabilidade:

Desenvolvimento de uma interface amigável onde os usuários possam ver recomendações personalizadas e aplicar filtros.

Outras Áreas de Aplicação:

Expansão do sistema para outros tipos de produtos e serviços de e-commerce, como moda, eletrônicos e até serviços digitais.

Git Link: <https://github.com/caiorfer/Sprint3-AI>