

Análise de dados e recomendação de músicas

Prova DataScience

Sistemas de recomendação estão se tornando indispensáveis para diversas plataformas que procuram oferecer ao usuário uma experiência personalizada e simplificada. A indicação correta de um filme, por exemplo, pode fazer a diferença entre conquistar o usuário ou perdê-lo. Antes de criar um sistema de recomendação é essencial uma análise de dados para descobrir informações com os dados disponíveis, o que ajuda a ter um melhor entendimento do usuário. Muitas sites e aplicativos conhecidos utilizam sistemas de recomendação para indicar diferentes itens para seus usuários, auxiliando a encontrar itens relevantes no meio de um conjunto vasto de opções.

Objetivo

O seu objetivo é desenvolver um notebook (.ipynb) para estudar os dados de usuários e músicas, avaliando possíveis padrões de comportamento. Em seguida, você deve criar uma função com o seguinte cabeçalho:

```
def get_recomendations ( n ):
```

Que tem o objetivo de recomendar n músicas para cada usuário no arquivo *target.csv*

O retorno da função deve ser um [dict](#) com o seguinte formato :

- A chave do dict é um inteiro que representa o `user_id`.
- O conteúdo do dict é uma lista de inteiros que representam as `music_ids` recomendadas por ordem de melhor recomendação.
- Exemplo
 - `dict[user_id] = [music_id_1, music_id_2 , ... , music_id_n]`

Exemplo de notebooks com análise de dados:

- [Análise de dados - Músicas](#)
 - Um exemplo bem básico de análise nos dados passados. Esperamos uma análise mais aprofundada.
- [Data Exploration Using Python](#)
 - Uma análise de dados em um dataset de preço de residências

Na função `get_recomendations()` use o algoritmo e estratégia que preferir para recomendar músicas. Não é necessário usar todos os dados para gerar as recomendações, você pode usar apenas os dados que lhe forem convenientes de acordo com o algoritmo escolhido.

Descrição dos arquivos

Todos os dados estão salvos em 5 arquivos no formato csv com um cabeçalho identificando o dado de cada coluna.

Os arquivos estão organizados da seguinte forma:

hits.csv:

Lista de música ouvidas recebida em um período de 20 dias

- user_id : id único usuário
- music_id : id único música
- genre_id : id do gênero da música
- state_id : id do estado onde aconteceu o hit (localidade/região)
- value : tempo que o usuário escutou daquela música durante os 20 dias

music_data.csv:

Dados adicionais sobre as músicas

- music_id : mesmo identificador único da música
- id_artist : identificador único do artista da música
- duration : duração total da musica
- plays : número de plays da música desde o seu upload
- downloads : número de downloads da música desde o seu upload

genre.csv:

Encoder para o genre_id

- genre_id : mesmo id do gênero de hits.csv
- genre : string do gênero

state.csv:

Encoder para o state_id

- state_id : mesmo id do state_id de hits.csv
- state : sigla do estado correspondente

target.csv:

Lista de 100 usuários alvo para recomendações

- user_id : id do usuário alvo

Observações

- Documente as suas decisões tomadas com células de texto (markdown) para entendermos o máximo possível do seu código.
- É permitido o uso de qualquer aplicação como [colab](#), [jupyter](#), [ipython](#), [kaggle](#). Envie apenas o arquivo `.ipynb`, que pode ser gerado localmente ou baixado de qualquer serviço citado.

- O seu notebook será executado por completo, sendo assim qualquer output que esteja salvo no notebook mas não é recriado pela célula de código correspondente será apagado.

Critérios de Avaliação

- Execução sem crashes
- Funcionamento correto do programa
- Performance do algoritmo
- Informações colhidas no estudo dos dados
- Qualidade das recomendações
- Descrição do funcionamento do programa (deve condizer com o que foi implementado)
- Organização e clareza do código

Boa prova =)