02

Recomendação baseada em similaridade de genero

Transcrição

Vamos dar uma olhada em uma outra possibilidade de recomendação, agora levando em consideração os filmes que o usuário assistiu. Me utilizando como exemplo, anotei alguns filmes que eu "assisti": `1`, `21`, `19`, `10`, `11`, `7` e `2`.

Criaremos um array com esses filmes, atribuindo-o a uma variável eu_assisti, e vamos filtrá-los no nosso dataset com filmes.loc[eu_assisti]. Os filmes serão listados nessa mesma ordem: Toy Story, Get Shorty, Ace Ventura, GoldenEye, The American President, Sabrina e Jumanji.

Agora, podemos criar um sistema de recomendação ignorando as informações colaborativas que nos foram passadas, e nos baseando somente nas informações que temos sobre o usuário: você sabe que eu assisti esses filmes e sabe os gêneros desses filmes.

Por exemplo, imagine que o último filme que eu assisti foi o Jumanji original, que é dos gêneros Aventura, Infantil e Fantasia. Vamos copiar esses três gêneros e, com filmes.query("generos=='Aventure|Children|Fantasy'"), buscaremos no nosso dataframe outros filmes cujos gêneros sejam exatamente esses três.

Como retorno, teremos diversos filmes desse conjunto de gêneros. Agora que os filtramos, vamos ordená-los, por exemplo, pelas notas médias. Porém, teremos diversos filmes cujo total de votos é muito baixo, já que estamos trabalhando com todo o dataset filmes. Será que devemos trabalhar com filmes que têm um total de votos baixo? Eu acho que não.

Portanto, basearemos nossa query() nos filmes_com_mais_de_50_votos, atribuindo o seu retorno a uma variável aventura_infantil_e_fantasia. Em seguida, vamos ordená-los da maior média para a menor e exibir os 10 primeiros.

Na nossa lista de recomendações para alguém que viu Jumanji, temos Harry Potter e a Pedra Filosofal, A História sem Fim (que faz parte da minha infância), Crônicas de Nárnia, A Bússola Dourada, o próprio Jumanji e A Chave Mágica (um filme que desconheço).

Nesse momento, é interessante removermos o Jumanji da lista, afinal eu já o assisti. Portanto, antes do sort_values(), faremos um drop(), passando como parâmetro a variável eu_assisti. Porém, nosso código deixará de funcionar, pois nem todos os filmes contidos no array eu_assisti estão presentes em aventura_infantil_e_fantasia.

A função drop() do Pandas possui um parâmetro extra, chamado erros, que por padrão retorna um erro quando ela não encontra um elemento. Vamos setar esse parâmetro como errors='ignore'. Dessa vez, conseguiremos remover o Jumanji da lista com sucesso, obtendo 5 filmes que, com base no nosso conjunto de dados, seriam recomendados para quem acabou de assistir ao Jumanji - independentemente de sabermos se a pessoa gostou ou não de Jumanji.

Repare que nessa abordagem não utilizamos, pelo menos inicialmente, o total_de_votos ou a nota_media. Porém, quando filtramos os elementos para tentarmos atribuir mais qualidade à recomendação, nós voltamos a utilizar esses critérios: primeiro para remover os filmes de nicho, e depois para estabelecer alguma preferência na recomendação.

Fizemos isso nos baseando em um conceito de similaridade, que definimos como sendo filmes que possuem exatamente os mesmos gêneros.