04

Heurística de recomendação e entendendo o que é colaboração

Transcrição

Vamos começar o nosso projeto? O conjunto de dados que utilizaremos é o do Movielens, um site pertencente ao Grouplens que tem um sistema de avaliação de vários filmes - ou seja, as pessoas podem dar notas de alguma maneira e encontrar outros filmes que lhes interessam, tudo muito ligado à questão de recomendação.

No Movielens, é possível baixar um conjunto de dados enorme contendo 20 milhões de avaliações dos usuários, ou outras duas variações desse conjunto, contendo 27 milhões ou 100 mil avaliações. Usaremos essa última, que é uma versão menor e mais simples. Claro, no momento que você tiver um algoritmo que julga ser bom, poderá baixar o arquivo com o conjunto completo de dados e rodar o algoritmo para ele.

Após baixarmos e descompactarmos o .ZIP disponibilizado no site, teremos dois arquivos que nos interessam: movies.csv, que são as informações gerais do filme; e ratings.csv, que são diversas avaliações.

Como plataforma, utilizaremos o Google Colaboratory, mas você pode utilizar um notebook Python local, no Kaggle ou onde quiser. Criaremos um Python 3 notebook, que renomearemos como "Introdução a Recomendação". Clicando em "Files" no lado esquerdo, o notebook irá se conectar com uma máquina virtual do Python e subiremos os arquivos, começando com movies.csv. Enquanto ele sobe, o Colab mostrará um aviso de que quando fecharmos a conexão, perderemos o espaço virtual e também esse arquivo, mas não tem problema.

Quando o arquivo terminar de subir, ele será listado no menu à esquerda. Poderemos então utilizar o pandas para ler esse arquivo, chamando pd.read_csv("movies.csv"). Atribuiremos essa leitura a uma variável filmes, e faremos filmes.head() para listarmos os 5 primeiros elementos.

Teremos os cinco primeiros filmes desse conjunto de dados, com os movieIds, os titles e os genres desses filmes. Queremos trabalhar com as colunas em português, portanto iremos sobrescrever as três colunas (ignorando o índice) com filmeId, título e gêneros.

Voltando ao menu "files", subiremos também o arquivo ratings.csv, que tem 800mb e contém as avaliações. Carregaremos as notas com pd.read_csv("ratings.csv"), chamaremos esse conjunto de notas e imprimiremos os cinco primeiros elementos para visualizá-los.

Renomearemos as colunas para usuarioId (quem deu a avaliação), filmeId (o filme que foi avaliado), nota (a avaliação) e momento (quando a avaliação foi feita). Podemos utilizar outras funções para descrever e entender melhor esse conjunto de dados. Por exemplo, com o Pandas, podemos fazer notas.describe(), verificando que as notas vão de um mínimo 0,5 até o máximo 5; que a mediana é 3,5; temos 492 mil avaliações; entre outras informações.

A grande questão agora é: queremos recomendar algo, mas como fazer isso? Temos vários filmes e várias notas que as pessoas deram para eles. Nesse cenário, que filme recomendar?

Na nossa primeira tentativa de recomendação, vamos analisar as notas, que representam votos que pessoas que assistiram aos filmes deram na plataforma. Vamos acreditar que essas notas são uma representação razoavelmente justa do mundo real, que é o nosso objeto de análise, apesar disso ser discutível.

Com base nesses dados, verificaremos que existem filmes que foram assistidos por muitas pessoas, e outros que foram assistidos por poucas pessoas. Vamos pensar, então, em um cenário bem genérico, no qual não sabemos nada sobre o usuário do sistema. Nessa situação, como iremos recomendar um filme ao usuário?

Bom, sabemos muito sobre os filmes! Sabemos que temos uma amostra das notas desses filmes, e podemos analisar, por exemplo, quantas pessoas deram nota para o filme Jumanji, para Toy Story ou para O Pai da Noiva Parte 2. Se assumirmos que há uma representatividade desses dados para a vida real, "saberemos" o quão populares esses filmes são.

Podemos, então, pegar todas essas notas (que foram fornecidas por outros usuários do sistema) e agrupá-las pelo campo filme. Veremos, então, que a primeira nota foi para o filme 307, a segunda para o 481, depois 1091, e assim por diante. Agora, com value_counts(), contaremos quantas vezes cada filme apareceu - ou seja, a frequência.

Com isso, teremos que, nesse conjunto de dados, o filme 318 foi avaliado 1739 vezes; o filme 352, 1698 vezes; o filme 296, 1628 vezes; e assim por diante. Se assumirmos essas avaliações como medida de popularidade, podemos recomendar o filme 318 para um usuário sobre o qual não temos nenhuma informação, apenas com base na colaboração de diversos outros usuários.

Agora vamos descobrir que filme é o 318. Antes disso, para facilitarmos nossa busca, setaremos o index como o próprio filmeId. Para isso, faremos filmes = filmes.set_index("filmeId"). Com filmes.loc[318], iremos localizar esse elemento pelo seu índice. Como retorno, teremos o filme The Shawshank Redemption.

Se procurarmos por esse nome no IMDB, veremos que o filme é muito bem avaliado e muitíssimo popular, estando no top 100 filmes da história do site. Portanto, parece razoável recomendarmos o filme mais popular (no nosso caso, o mais votado). Na prática, queremos ordenar pelos filmes mais votados e recomendar os primeiros - afinal, podemos concluir que muitas pessoas têm interesse neles. Claro, estamos tirando uma heurística para definir um padrão de recomendação, sem provas do que estamos concluindo.

Nesse cenário, no qual não temos informações sobre nosso usuário, utilizar o critério do mais popular (que é uma informação colaborativa) parece funcionar. Se soubéssemos que o usuário é brasileiro, poderíamos filtrar apenas filmes brasileiros. É isso, por exemplo, que o Youtube tenta fazer: se um usuário usa um navegador em português no Brasil para acessar o Youtube, ele infere que esse usuário é brasileiro e lhe recomenda conteúdos que são mais populares ou que funcionam melhor no Brasil (dois critérios que não são necessariamente a mesma coisa).

Vamos atribuir o retorno do nosso value_counts() a uma variável total_de_votos, e chamaremos total_de_votos.head() para exibirmos os cinco primeiros elementos da lista. Em seguida, aproveitaremos que o conjunto filmes e a série total_de_votos estão ambos indexados pelo Id para adicionarmos esse último conjunto como coluna do primeiro, faazendo filmes['total_de_votos'] = total_de_votos.

Com isso, teremos um dataframe que mostra também o total de votos de cada filme. Agora, vamos ordenar os flmes pelo total de votos com filmes.sort_values("total_de_votos"). Porém, a ordenação padrão do Pandas é crescente (do menor para o maior). Como queremos uma ordenação decrescente (do maior para o menor), utilizaremos o parâmetro nomeado ascending = False.

Agora temos uma lista com The Shashank Redemption, Forrest Gump, Pulp Fiction, The Silence of the Lambs, The Matrix, Star Wars: Episode IV, e por aí vai. Com .head(), mostraremos somente os cinco primeiros filmes dessa lista.

A primeira heurística de recomendação é: com base nas informações que outros usuários e usuárias passaram (o número de votos), podemos indicar os cinco filmes mais populares (que definimos como sendo os mais avaliados) para uma pessoa sobre quem não sabemos nada.