Construindo um sistema de recomendação de filmes

GIOVANNA BATALHA
MATHEUS CEZAR NUNES
RAISSA CAVALCANTE CORREIA
RAYSA MASSON BENATTI

I. INTRODUÇÃO

O presente trabalho apresenta a descrição das conclusões e estratégias adotadas para a resolução do projeto final da disciplina Introdução à Inteligência Artificial (MC906), ministrada no primeiro semestre de 2019 na Unicamp pela professora Esther Luna Colombini.

A proposta do trabalho é a implementação de um sistema de aprendizado de máquina que resolva um problema determinado pelo grupo. Os autores optaram pela construção de um sistema de recomendação de filmes, similar a ferramentas usadas, por exemplo, em serviços de *streaming* como Netflix.

Na seção II do relatório, trazemos a definição do problema. A seção III dedica-se à descrição das técnicas adotadas para buscar soluções e especificidades da implementação. A seção IV descreve os resultados observados. Finalmente, na seção V, discutimos tais resultados e apresentamos conclusões. As referências, bem como a lista de responsabilidades de cada participante e descrição da organização da equipe, encontram-se ao final do trabalho.

II. O PROBLEMA

Sistemas de recomendação são aplicações usadas em diversos setores para auxiliar a execução da tarefa de prever o comportamento de um usuário com base em determinadas métricas — mais especificamente, prever de qual(is) produto(s), ou subclasse(s) de produtos, o consumidor gostará mais, e/ou qual(is) terá maior probabilidade de adquirir. Há sistemas de recomendação implementados, por exemplo, em *sites* de compras (Amazon), serviços de música (Last.fm, Spotify, Pandora Radio) e serviços de vídeo (YouTube, Netflix), dentre outros [1].

Em geral, tais sistemas são construídos com base em dados do comportamento prévio dos usuários, como classificações já fornecidas por eles a determinados produtos. A aplicação de recomendação basicamente filtra esses dados e, com a aplicação de algoritmos sobre eles, obtém como resultado produto(s) de que o usuário provavelmente gostará, ou notas que ele presumivelmente lhe(s) daria [2].

Diversos algoritmos podem ser usados para realizar a filtragem de dados e cálculo das recomendações. No presente trabalho, foram quatro as abordagens seguidas na solução final: **filtragem baseada em conteúdo**, que se fundamenta na similaridade entre os itens e nas escolhas passadas do usuário; **filtragem colaborativa**, que utiliza, além de escolhas passadas, comparações entre usuários distintos; **fatorização matricial**, ou filtragem colaborativa baseada em modelo, que reduz dimensionalidade em relação à abordagem anterior; e **regressão linear**. Na próxima seção, exploraremos em detalhes cada uma delas.

III. SOLUÇÕES COM APRENDIZADO DE MÁQUINA

Após a realização de testes exploratórios de diferentes configurações de *datasets* e algoritmos — processo descrito em maiores detalhes no anexo deste trabalho —, as soluções elegidas para a construção final do sistema são aquelas descritas abaixo. A implementação baseou-se nos algoritmos propostos por Pulkit Sharma em [2] e James Le em [3] (cujo código completo encontra-se em [4]), com algumas modificações. A linguagem adotada em todos os casos é Python.

Na implementação de todas as soluções descritas abaixo, foi utilizado um dos *datasets* **MovieLens** [5]. A versão escolhida, `ml-latest-small`, possui pouco mais de 100 mil classificações anônimas dadas por cerca de 600 usuários a aproximadamente 9700 filmes. O *dataset* é composto por dois arquivos relevantes para os algoritmos implementados, cada um dos quais com os seguintes atributos:

1) Movies:

- `movieId`: número que identifica um filme;
- `title`: título de um filme, inserido manualmente ou importado de The Movie Database, com o ano de lançamento entre parênteses;
- `genres`: gênero(s) associado(s) a um filme, separados por uma barra vertical, podendo incluir "*action*", "*fantasy*", "*sci-fi*", dentre outros (lista completa em [5]).

2) Ratings:

- `userId`: identificador de um usuário;
- `movieId`: identificador de um filme;
- `rating`: classificação, de 1 a 5 estrelas, dada pelo usuário `userId` ao filme `movieId`;
- `timestamp`: valor que representa o momento em que a classificação foi feita, em segundos.

A. Filtragem baseada em conteúdo

Técnica que se baseia nas similaridades entre os itens. Geralmente funciona melhor e mais facilmente quando há um

critério objetivo e claro de similaridade entre 2 itens. Há 2 coisas que precisamos citar:

TF é simplesmente a frequência de um termo num arquivo. IDF é o inverso da frequência do arquivo em relação ao conjunto de arqiuvos. TF-IDF é usado principalmente pelo seguinte fato:

Numa busca textual tem certas palavras, principalmente preposições e conjunções que se repetem muito, e têm uma frequência alta, porém são irrelevantes no assunto tratado. Já a palavra principal de uma busca tem um TF-IDF alto, pois não tem uma frequência muito alta, porém tem um IDF relevante.

Após este cálculo podemos determinar o quão próximos os itens são entre si. Isso é possível através do cálculo do cosseno entre os 2 vetores que representam os parâmetros de cada item num espaço n-dimensional.

Num caso canônico, o perfil de usuário também tem seu próprio vetor de preferências com base nos conteúdos que consome, portanto a similaridade produto-usuário ou usuário-usuário pode ser determinada de forma similar complementando uma a outra. No algoritmo implementado, é calculada a similaridade item-item, entre filmes.

A preferência pelo cosseno é o fato de ser uma função que sempre cresce conforme o ângulo aumenta entre 0 e 180 graus, e depois volta a diminuir pois se aproximam "pelo outro lado".

Prós dessa abordagem:

- Não há necessidade de dados de outros usuários, nem problemas de inicialização para treinamento, problema de dados esparsos e afins;
- Pode recomendar bem para usuários de preferências muito distintas;
- Pode recomendar itens novos e impopulares para os usuários certos;
- Pode facilmente explicar por que um item foi recomendado com *trace-back*.

Os contras:

- Difícil de encontrar as características apropriadas;
- Não recomenda nada que fuja ao perfil do usuário;
- Não explora o julgamento dos usuários em torno de um conteúdo.

A implementação dessa técnica no presente trabalho é feita com auxílio dos métodos `scatter_matrix` (Pandas), `TfidVectorizer` e `linear_kernel` (scikit-learn), além de demais funções da biblioteca Pandas. O algoritmo de referência retorna 20 filmes sugeridos a partir da entrada do título de um filme que consta no *dataset*, fazendo o cálculo de similaridade com base na lista de gêneros associados a cada filme. Na implementação adaptada pelo grupo, as recomendações foram limitadas a cinco por filme. Foi acrescentada uma funcionalidade de recomendação a um usuário específico a partir da lista de filmes classificados por ele com cinco estrelas, além de testes com filmes de gêneros variados e *sequels*, para investigar a validade do recomendador.

B. Filtragem colaborativa

Inteiramente baseado no histórico de consumo do usuário, auxiliado também pela similaridade entre perfis de histórico entre diferentes usuários. Esse algoritmo é o mais comum e tem como qualidade ser capaz de aprender as melhores características a se analisar entre os usuários.

A comparação usuário-usuário toma muito tempo computacionalmente, apesar de ser muito paralelizável porém ele é complementado pela comparação item-item, cuja ideia é criar uma matriz de similaridade entre usuários e tornar o processo mais rápido e com um aprendizado mais eficaz com o tempo.

Existem 3 formas de similaridade: a de cosseno (já discutida), a de Jaccard e a de Pearson, no caso do nosso projeto implementamos a de Pearson e avaliamos a qualidade da recomendação pelo RMSE e MAE.

C. Fatorização matricial

Esse método escala de forma bem ruim e num computador normal só pode ser usado em *datasets* menores na casa dos milhares ao invés de milhões de itens. Se o *dataset* não tiver uma boa qualidade que é grande variadade de usuários, podemos sofrer com *overfitting*.

Algo recomendado para se fazer com o *dataset* é a redução de dimensionalidade, necessária para: descobrir correlações e características no dado bruto; remover redundância e ruído; interpretar e visualizar mais facilmente.

O filtro colaborativo com base em modelo é baseado na Fatoração Matricial, principalmente por ser uma forma não supervisionada de obter correlações em *datasets*.

O objetivo da fatoração é descobrir as preferências do usuário, bem como atributos dos itens e assim prever as avaliações ainda não realizadas através do produtivo escalar entre as características dos usuários com as características dos itens. Essa técnica permite lidar com matriz esparsas de alta dimensionalidade através da reestruturação.

Um método muito usado de fatoração matricial que usaremos aqui é a SVD. Ele decompõe a matriz na sua melhor classificação mais baixa aproximada da matriz original. O resultado é criar 3 matrizes, 2 unitárias e uma diagonal. Uma relacionada aos usuários, uma aos itens, e a diagonal representando os pesos de cada conceito relacionando usuários e itens.

D. Regressão Linear

O método de Regressão Linear é um método matemático que relaciona N variáveis independentes de eixo X, com uma variável dependente do eixo Y. Dado isso, o processo de treinamento desse modelo utiliza exemplos das variáveis do eixo X com seus respectivos valores em Y para gerar uma função linear.

Após o processo de treinamento, o modelo está pronto para receber valores de X e realizar previsões para o valor no eixo Y baseado na função linear gerada.

Aplicando o conceito de separação do *dataset* entre dados de treino e dados de teste, é possível avaliar a acurácia do modelo com os valores esperados e obtidos.

Neste método, utilizamos os *datasets* de *movies* e *ratings* para aplicar um Regressão Linear com múltiplas variáveis, utilizando a biblioteca SciKit Learn.

IV. Resultados

A. Filtragem baseada em conteúdo

Três blocos de testes foram executados com o recomendador de filtragem baseada em conteúdo. O primeiro deles consistiu em avaliar as recomendações de *sequels*, isto é, sequências de filmes, a fim de verificar a consistência dos resultados — a partir de intuição segundo a qual, via de regra, filmes pertencentes a uma mesma sequência resultariam em recomendações similares. No segundo bloco, seis filmes aleatórios diferentes, com diferentes gêneros associados, foram fornecidos como entrada para a função de cômputo de recomendações, para analisar o comportamento do sistema. Por fim, escolhe-se um usuário aleatório do *dataset* para que o programa devolva as recomendações que sugeriria a ele com base nos filmes que classificou com cinco estrelas.

1) Sequels: As sequências testadas foram as dos filmes *Free Willy* e *Karate Kid*. Os resultados encontram-se nas tabelas I e II.

2) Filmes aleatórios: Nessa etapa, foram obtidas recomendações para seis filmes distintos, conforme descrito na Tabela III.

3) Usuário aleatório: Nessa modalidade de teste, um ID de usuário é fornecido para que o programa calcule recomendações a ele a partir dos filmes classificados por esse usuário com 5 estrelas. Escolhemos aleatoriamente o **ID 2** para receber recomendações. A Tabela IV mostra os resultados obtidos.

Tabela I
Free Willy

Filme	Recomendações
Free Willy (1993) (aventura, infantil, drama)	Black Beauty (1994); Free Willy (1993); Andre (1994); Homeward Bound: The Incredible Journey (1993); Buddy (1997)
Free Willy 2: The Adventure Home (1995) (aventura, infantil, drama)	Black Beauty (1994); Free Willy (1993); Andre (1994); Homeward Bound: The Incredible Journey (1993); Buddy (1997)

Tabela II
Karate Kid

Filme	Recomendações
Karate Kid, The (1984) (drama)	Othello (1995); Dangerous Minds (1995); Cry, the Beloved Country (1995); Restoration (1995);Georgia (1995)
Karate Kid, Part II, The (1986) (ação, aventura, drama)	Ben-Hur (1959); Man in the Iron Mask, The (1998); Poseidon Adventure, The (1972); Seven Samurai (Shichinin no samurai) (1954); Emerald Forest, The (1985)
Karate Kid, Part III, The (1989) (ação, aventura, infantil, drama)	Free Willy 2: The Adventure Home (1995); Black Beauty (1994); Free Willy (1993); Andre (1994); Homeward Bound: The Incredible Journey (1993)
Next Karate Kid, The (1994) (ação, infantil, romance)	Mighty Morphin Power Rangers: The Movie (1995); 3 Ninjas Knuckle Up (1995); Jungle Book, The (1994); 3 Ninjas: High Noon On Mega Mountain (1998); 3 Ninjas (1992)
Karate Kid, The (2010) (ação, infantil, drama)	Mighty Morphin Power Rangers: The Movie (1995); 3 Ninjas Knuckle Up (1995); Jungle Book, The (1994); 3 Ninjas: High Noon On Mega Mountain (1998); 3 Ninjas (1992)

Tabela III
Filmes variados

Filme	Recomendações
Mulan (1998) (aventura, animação, infantil, comédia, drama, musical, romance)	High School Musical (2006); Shrek 2 (2004); Ernest & Célestine (Ernest et Célestine); Pocahontas (1995); Hunchback of Notre Dame, The (1996)
Life Is Beautiful (La Vita è bella) (1997) (drama, romance, guerra, comédia)	Life Is Beautiful (La Vita è bella) (1997); Train of Life (Train de vie) (1998); Tiger and the Snow, The (La tigre e la neve) (2005); I Served the King of England (Obsluhoval jsem anglického krále) (2006); Colonel Chabert, Le (1994)
Sound of Music, The (1965) (musical, romance)	Sound of Music, The (1965); From Justin to Kelly (2003); Easter Parade (1948); Umbrellas of Cherbourg, The (Parapluies de Cherbourg, Les) (1964); Dirty Dancing (1987)
Shining, The (1980) (terror)	Castle Freak (1995); Cemetery Man (Dellamorte Dellamore) (1994); Eyes Without a Face (Yeux sans visage, Les) (1959); Children of the Corn IV: The Gathering (1996); Fog, The (1980)
Paranormal Activity 3 (2011) (terror)	Castle Freak (1995); Cemetery Man (Dellamorte Dellamore) (1994); Eyes Without a Face (Yeux sans visage, Les) (1959); Children of the Corn IV: The Gathering (1996); Fog, The (1980)
10 Things I Hate About You (1999) (comédia, romance)	Sabrina (1995); Clueless (1995); Two if by Sea (1996); French Twist (Gazon maudit) (1995); If Lucy Fell (1996)

Tabela IV
RECOMENDAÇÕES PARA O USUÁRIO DE ID 2

Porque o usuário gostou de:	Recomendamos também:
Step Brothers (2008)	Four Rooms (1995); Ace Ventura: When Nature Calls (1995); Bio-Dome (1996); Friday (1995); Black Sheep (1996)
Inside Job (2010)	Heidi Fleiss: Hollywood Madam (1995); Catwalk (1996); Anne Frank Remembered (1995); Jupiter's Wife (1994); Man of the Year (1995)
Warrior (2011))	Othello (1995); Dangerous Minds (1995); Cry, the Beloved Country (1995); Restoration (1995); Georgia (1995)
Wolf of Wall Street, The (2013)	Jimmy Hollywood (1994); Trainspotting (1996); Fried Green Tomatoes (1991); I Went Down (1997); Player, The (1992)
Mad Max: Fury Road (2015)	Independence Day (a.k.a. ID4) (1996); Escape from L.A. (1996); Abyss, The (1989); Escape from New York (1981); Star Trek: First Contact (1996)
The Jinx: The Life and Deaths of Robert Durst (2015)	Heidi Fleiss: Hollywood Madam (1995); Catwalk (1996); Anne Frank Remembered (1995); Jupiter's Wife (1994); Man of the Year (1995)

B. Filtragem colaborativa

A ideia é classificar os filmes através da fórmula de correlação presente no scipy spatial distance [6], e realizar separadamente para um grupo de teste e para um grupo de treinamento. Por fim é calculado o erro RMSE entre as recomendações dos dois grupos.

$$1 - \frac{(u-\bar{u})\cdot(v-\bar{v})}{||(u-\bar{u})||_2||(v-\bar{v})||_2}$$

Os resultados obtidos foram 2 prints: O primeiro se trata das predições de usuário, e o segundo da RMSE entre as predições proporcionadas pelo grupo de teste comparado com o grupo de treinamento.

$$\begin{bmatrix}[1.000e+002.000e+003.500e+00] \\ [1.000e+002.900e+013.500e+00] \\ [1.000e+003.200e+013.500e+00] \\ \dots \\ [6.600e+017.450e+025.000e+00] \\ [6.600e+011.148e+035.000e+00] \\ [6.600e+011.175e+035.000e+00]]\end{bmatrix}$$

prediction:

$$\begin{bmatrix}[1348.03511016 - 2721.993444551380.4583344] \\ [-2019.742410514102.49967923 - 2049.25726872] \\ [-2018.183107434102.39163936 - 2047.70853193] \\ \dots \\ [-1747.610217364340.92495043 - 1777.31473306] \\ [-1615.355594394479.38482345 - 1645.02922906] \\ [-1606.441538364488.55540886 - 1636.1138705]]\end{bmatrix}$$

User-based CF RMSE - Test Data: 11676.570087134485

C. Fatorização matricial

A execução do algoritmo de fatorização matricial foi feita de maneira a verificar as recomendações dadas ao usuário de ID 2, como no recomendador baseado em conteúdo, em prol da consistência de nossa avaliação.

Após gerar as matrizes de *ratings* e predições, o algoritmo executa a função `recommend_movies`, que fornece a lista dos cinco primeiros filmes que o usuário determinado avaliou — resumida na Tabela V — e outra com os cinco filmes de que ele provavelmente mais gostará — resumida na Tabela VI.

Tabela V
CLASSIFICAÇÕES DO USUÁRIO DE ID 2

Classificação	Título	Gêneros
5.0	The Jinx: The Life and Deaths of Robert Durst (2015)	documentário
5.0	Mad Max: Fury Road (2015)	ação, aventura, sci-fi, thriller
5.0	Wolf of Wall Street, The (2013)	comédia, crime, drama
5.0	Warrior (2011)	drama
5.0	Step Brothers (2008)	comédia

Tabela VI
FILMES RECOMENDADOS PARA O USUÁRIO DE ID 2

ID	Título	Gêneros
2959	Fight Club (1999)	ação, crime, drama, thriller
2571	Matrix, The (1999)	ação, sci-fi, thriller
7153	Lord of the Rings: The Return of the King, The (2003)	ação, aventura, drama, fantasia
296	Pulp Fiction (1994)	comédia, crime, drama, thriller
4993	Lord of the Rings: The Fellowship of the Ring, The (2001)	aventura, fantasia

Para avaliar o recomendador, a fatorização matricial foi novamente executada, dessa vez com auxílio do método `svd` e treinamento do dataset com a técnica *k-folding*, sendo k = 5. Para cada *folding*, foram calculados a raiz quadrada do erro médio (*RMSE*), o tempo de *fit* e o tempo de teste, com os resultados arredondados para duas casas decimais descritos na Tabela VII.

Tabela VII
MÉTRICAS DE AVALIAÇÃO DE K-FOLDING

K	RMSE	Fit time	Test time
1	0.87	1.96	0.06
2	0.86	1.90	0.08
3	0.87	2.06	0.07
4	0.86	1.89	0.06
5	0.88	1.81	0.12

Em seguida, foram geradas previsões de classificações que o usuário determinado (no caso, de ID 2) daria para filmes, e esses valores foram confrontados com as classificações reais. Uma parte dos resultados está descrita na Tabela VIII.

Tabela VIII
CLASSIFICAÇÕES PREVISTAS PARA USER DE ID 2 COM K-FOLDING

ID do filme	Classificação prevista	Classificação real
318	3.59	3.0
333	3.59	4.0
1704	3.59	4.5

D. Regressão Linear

Durante a etapa de preparação dos dados, o *dataset* de *movies* passou por um processo de *dummyfication* na coluna de gêneros, onde quebramos os genêros que estavam concatenados por | em multiplas colunas, onde cada uma possuia o valor 0 ou 1.

Após esse processo foi feita a limpeza de dados que não importam na definição da nossa varíavel Y, ou seja, os que não são dependentes desse dado. Foram eles os valores de *userId*, *movieId* e *title*.

Ao aplicar a Regressão Linear com o intuito de descobrir a avaliação que o usuário faria para um filme não assistido, não foi possível utilizar todos os dados disponíveis de avaliações por conta do pressuposto que usuários avaliam filmes de formas diferentes e com gostos diferentes. Logo, os dados disponíveis são aqueles que o próprio usuário avaliou outros filmes.

Essa limitação causou uma restrição muito grande nos dados, mantendo em média, cerca de 30 a 40 registros por usuário. Portanto, foi notado um grande *Underfitting* dos dados devido a grande quantidade de features (cerca de 40 gêneros dummyficados) e, principalmente, pela falta de registros para que a função linear fosse bem treinada em todos os genêros.

Após realizar testes variando entre 10%, 20% e 30% de separação entre a base de teste e treino, o melhor resultado obtido para o modelo foi a separação de 20% que resultou em 25% de acuracia, concluindo que não seria viável recomendar um filme com este *dataset* através dessa abordagem.

V. DISCUSSÃO E CONCLUSÕES

De modo geral, todas as técnicas utilizadas — à exceção da regressão linear — trouxeram resultados coerentes para o que se espera de um recomendador de filmes, ainda que com ressalvas. Nesta seção, sublinhamos alguns aspectos que merecem destaque a partir dos resultados observados.

A filtragem baseada em conteúdo não possui métrica de avaliação quantitativa, mas pode ser analisada qualitativamente. Sua execução revela apresentar resultados consistentes quando testamos para filmes que fazem parte de um mesmo *sequel*. *'Free Willy'* e *'Free Willy 2'*, por exemplo, recebem recomendações idênticas, resultado esperado por terem ambos exatamente a mesma lista de gêneros associados. Curiosamente, todos os filmes recomendados têm protagonistas animais — o que não é avaliado diretamente pelo recomendador, e sim indiretamente, pelo fato de se tratar de característica aparentemente comum a filmes associados aos gêneros indicados (aventura, infantil, drama). Nota-se também que se recomenda o próprio *'Free Willy'* para uma entrada *'Free Willy'*, o que ocorreu também com outros filmes, algo que não se costuma observar em aplicações reais.

Para a *sequel* de *'Karate Kid'*, os resultados distintos porém similares sugerem bom funcionamento da ferramenta; afinal, diferente do que ocorre com *'Free Willy'*, os gêneros associados aos filmes de *'Karate Kid'* não são sempre idênticos, mas parecidos.

É importante destacar que, embora o funcionamento ocorra conforme a especificação sugere, os resultados não são necessariamente ideais. Os filmes *'The Shining'* e *'Paranormal Activity 3'* recebem exatamente a mesma lista de recomendações, uma vez que ambos têm exatamente o mesmo gênero associado a si (terror). Embora sejam do mesmo gênero, as semelhanças entre eles praticamente param por aí; não têm muitos aspectos em comum fora esse, o que chama a atenção para a importância de usar *datasets* com dados bem coletados — em especial quando se trata das métricas que embasam o cálculo —, e/ou reavaliar as métricas usadas dependendo do caso.

Assim, no caso de um recomendador baseado em conteúdo cujo cálculo tenha por referência o gênero dos filmes, por exemplo, é importante que o *dataset* faça a associação filme-gênero(s) da maneira mais acurada possível, o que influenciará diretamente a qualidade do resultado da recomendação. Além disso, recomendadores que apliquem a mesma técnica poderiam variar as métricas de cálculo, usando, por exemplo, ano de lançamento do filme, diretores ou atores principais, entre outras.

Os métodos de filtragem colaborativa e fatorização matricial, que deriva daquela, em geral proporcionam resultados mais acurados por utilizarem mais de uma métrica no cômputo das recomendações. De fato, os filmes recomendados para o usuário de ID 2 pelo recomendador baseado em fatorização matricial foram completamente diferentes daqueles sugeridos pelo sistema baseado em conteúdo, não havendo um ponto em comum sequer nos filmes mais recomendados. Nota-se que os títulos recomendados via fatorização matricial são consideravelmente mais populares, o que faz sentido dado que o método leva em consideração *ratings* dos usuários como um todo.

A execução da fatorização matricial com a função `recommend_movies` implementada manualmente parece ter sido mais bem-sucedida que quando a executamos com auxílio do método nativo `svd`, dado que, com o último, as classificações previstas têm sempre valor idêntico. Com a emergência desse erro, prejudicou-se a avaliação quantitativa do método, restando mais confiável a avaliação qualitativa, conforme foi feito com o recomendador baseado em conteúdo.

Referências

[1] Wikipedia, the free encyclopedia, "Recommender system," 2019. [Online]. Available: https://en.wikipedia.org/wiki/Recommender_system 1

[2] Pulkit Sharma, "Comprehensive Guide to build a Recommendation Engine from scratch (in Python)," *Analytics Vidhya*, 2018. [Online]. Available: https://bit.ly/2PvhIxV 1

[3] James Le, "The 4 Recommendation Engines That Can Predict Your Movie Tastes," 2018. [Online]. Available: https://bit.ly/2TrYco3 1

[4] ——, "MovieLens Recommendation Systems," 2018. [Online]. Available: https://bit.ly/2XCUQzM 1

[5] GroupLens Research Project, "MovieLens ml-latest-small Dataset," 2018. [Online]. Available: https://grouplens.org/datasets/movielens/ 1

[6] Scipy, "Scipy.spatial.distance.correlation," 2014. [Online]. Available: https://docs.scipy.org/doc/scipy-0.14.0/reference/generated/scipy.spatial.distance.correlation.html 4

Anexo: lista de responsabilidades e organização da equipe

Todos os membros do grupo participaram das tarefas relacionadas à escolha do tema e pesquisa de material pertinente. A comunicação da equipe aconteceu em alguns encontros presenciais espaçados ao longo do tempo de execução do trabalho, bem como *online*, diariamente pelo mesmo período, pelo aplicativo Telegram. Todos os integrantes participaram ativamente da comunicação.

A produção de código foi organizada em um repositório GitHub, em que cada membro criou sua *branch* para trabalhar explorando configurações diferentes de algoritmos e técnicas, com posterior junção dos trabalhos individuais na *branch* ***master***. Listas com a descrição das responsabilidades individuais de cada membro encontram-se abaixo — cada uma tendo sido elaborada pelo próprio membro e validada pelos demais.

Giovanna

Pesquisa Inicial: Explorado datasets e temas para elaboração do projeto. Exploração inicial: implementação inicial dos métodos que foram analisados nesta pesquisa e entendimento da base teórica que compões estas soluções. Organização final: Organização de alguns métodos para exibir os resultados mais interessantes; elaboração do conteúdo e gravação do vídeo.

Matheus

Pesquisa inicial utilizando o kaggle para explorar datasets e temas interessantes para o projeto. Implementação inicial do método baseado em conteúdo e Regressão Linear. Exploração e variação do método de filtragem colaborativa. Manipulação do dataset e finalização do método da Regressão Linear. Escrita de partes do relatório e ajustes na criação do vídeo.

Raissa

Pesquisa inicial para obtenção dos links utilizados na bibliografia, busca e escolha do dataset ml-latest-small, implementação dos tutoriais 1, 2 e 3 presentes na pasta tutorials, escrita das seções 3 e 4B do relatório, intervenções pontuais na apresentação e nas implementações dos métodos, introdução ao problema e ao dataset na apresentação.

Raysa

Etapa exploratória: implementação de tutorial com método de *deep learning* (descartado); implementação: execução e definição de testes do recomendador baseado em conteúdo, intervenções pontuais no recomendador por fatorização matricial; relatório: organização, formatação, produção (seções I, II, IV-A, IV-C, V, apresentação da seção III) e intervenções pontuais ao longo de todo o relatório; apresentação: edição de *slides* e apresentação relativa ao recomendador baseado em conteúdo.