**Do demográfico ao híbrido: performances e resultados de algoritmos de recomendação**

Gabriel Felix dos Santos¹\*;Prof. Dr. Renato Máximo Sátiro2

1 Universidade Paulista UNIP. Bacharel em Ciência da Computação. Rua Maria Milhan Biazotto, 317 – Jardim Sampaio – CEP 168000-000 – Mirandópolis, São Paulo, Brasil.

2 Escola Superior de Agricultura "Luiz de Queiroz" - USP/ESALQ – Professor do curso de MBA em Data Science e Analytics – Rua Alexandre Herculano 120, sala T4 – CEP 13418-445 – Piracicaba (SP), Brasil.

Universidade Federal de Goiás – UFG. Doutor em Administração pelo Programa de Pós-Graduação em Administração da FACE/UFG. Avenida Esperança s/n, Câmpus Samambaia – Prédio da Reitoria. CEP 74690-900.

\*autor correspondente: [csfelix08@gmail.com](mailto:csfelix08@gmail.com)

**Do demográfico ao híbrido: performances e resultados de algoritmos de recomendação**

**Resumo**

Os sistemas de recomendação estão em todos lugares, desde lojas virtuais até plataformas de vídeos, melhorando as experiências dos usuários e, consequentemente, alavancando os lucros das empresas que os utilizam. Com o passar do tempo, novos algoritmos foram desenvolvidos para que as recomendações fossem as mais próximas possíveis dos gostos reais dos usuários, contudo, demandaram uma troca equivalente: alguns requereram uma vasta base de dados, enquanto que outros necessitaram de maior poder computacional. Por esta razão, a partir de bases de dados de animes, usuários e avaliações, investigou-se qual das principais abordagens de recomendação consome mais tempo de execução, processamento de CPU e memória RAM, bem como se as recomendações geradas são diversificadas e de acordo com os gostos individuais dos usuários. Os resultados indicaram que o uso de Filtragem Demográfica consome uma parcela insignificante de processamento, memória e tempo, mas resultam em recomendações menos diversificadas; por outro lado, os modelos de Filtragem Baseada em Conteúdo com Processamento de Linguagem Natural, Filtragem Colaborativa e Filtragem Híbrida, nesta ordem, requereram maior poder computacional e maior variedade do banco de dados, conseguindo, consequentemente, fornecer recomendações mais próximas aos gostos reais dos usuários. Entretanto, é importante realizar os mesmos testes em diferentes dispositivos e base de dados, pois as mensurações de desempenho dos algoritmos são limitadas às configurações da máquina e às bases de dados utilizadas neste estudo. Para trabalhos futuros recomenda-se a aplicação de Aprendizado de Máquina Profundo e base de dados de diferentes plataformas para uma melhor comparação entre os algoritmos.

**Palavras-chave**: desempenho de algoritmos; sistemas de recomendação; processamento de linguagem natural; tempo de execução; animes.

**Introdução**

Em nenhum momento da história, a humanidade esteve tão conectada como na era atual, graças aos avanços computacionais e à acessibilidade à rede de internet. Relações pessoais agora estão a apenas um clique de distância e milhões de informações são trocadas a todo momento, desde textos até vídeos e áudios.

Esta abundância de dados tem beneficiado a população mundial de diversas maneiras. Basta realizar uma pequena pesquisa sobre um determinado assunto que o usuário deparar-se-á com diversos conteúdos e até mesmo pesquisas científicas atuais do ramo. No entanto, esse conglomerado de informações também apresenta desafios significativos, pois o ser humano não consegue lidar com muitas informações simultâneas e, consequentemente, acaba tendo dificuldades de filtrar os conteúdos que o interessa Silva (2014), levando-o ao desgaste e a decisões equivocadas (Ricci et al., 2011).

Devido a estes problemas, foram desenvolvidos os Sistemas de Recomendação [SR], com a pesquisa ministrada por Whittaker e Sidner (1996) sobre a interação dos usuários com a sobrecarga de e-mails um dos pontapés iniciais de estudos no ramo.

Com o passar do tempo, novas formas de SRs foram desenvolvidas e aperfeiçoadas e, segundo Burke (2002), destacam-se alguns grupos de algoritmos, como os de Filtragem Demográfica, que agrupam os usuários em comunidades e realizam recomendações específicas para cada nicho; os de Filtragem Baseada em Conteúdo, que indicam itens similares aos que os usuários avaliaram positivamente; os de Filtragem Colaborativa, que agrupam os usuários com gostos semelhantes e realizam recomendações de itens de acordo com as similaridades entre estes grupos; e os de Filtragem Híbrida, que combinam duas ou mais abordagens de Filtragem para fornecerem melhores recomendações.

O motivo pelo qual há diferentes metodologias de filtragem está relacionado aos problemas encontrados nas aplicações dos SRs. Alguns algoritmos realizam recomendações muito próximas e flexíveis aos gostos dos usuários, mas requerem uma quantidade vasta de dados e um alto custo de processamento e tempo computacional. Outros algoritmos são menos custosos, mas fornecem recomendações mais genéricas, como é o caso da Filtragem Baseada em Conteúdo, onde os gostos dos usuários nem sempre são bem representados apenas pela similaridade dos itens (Das et al., 2007).

Diante disso, muitas empresas que dependem da venda de produtos se questionam sobre qual forma de recomendação aplicar para maximizar os lucros com menores custos: devem-se optar por uma abordagem que gere resultados mais precisos e personalizados a um custo computacional maior, ou por uma menos custosa, mas com resultados mais generalizados?

Portanto, o presente trabalho foi desenvolvido com o objetivo de analisar o desempenho, em termos de custo de hardware e tempo de execução, das principais abordagens de recomendação descritas anteriormente, a fim de pontuar os prós e contras de cada uma. Como objetivo secundário, buscou-se comparar as recomendações geradas pelos modelos e seus impactos nos seus usuários.

**Material e Métodos**

Este trabalho utilizou o método de experimentação e, a fim de facilitar a compreensão dos materiais e métodos aplicados, esta seção foi dividida em cinco partes: Ambiente de Desenvolvimento, Coleta de Dados, Dicionário das Variáveis, Limpeza e Transformação dos Dados, e Modelos de Recomendação.

**Ambiente de Desenvolvimento**

Os códigos foram desenvolvidos na versão 3.10.9 da linguagem de programação Python nos ambientes de desenvolvimento Jupyter Lab e Jupyter Notebook, ambos na versão 3.5.3. Além da aplicação de pacotes do Python para otimizar o processo de desenvolvimento, com suas versões e finalidades listadas na Tabela 1.

Tabela 1. Pacotes da linguagem Python aplicados

|  |  |  |  |
| --- | --- | --- | --- |
| Aplicação | Pacote | Versão | Uso |
| Avaliação de performance | Psutil | 5.9.0 | Medição do uso de “Random-Access Memory [RAM]” e de processador |
| Avaliação de performance | Time | 3.10.9 | Medição do tempo de execução dos códigos |
| Modelagem e validação | SKLearn | 1.2.1 | Treinamento e validação dos modelos de Filtragem Baseada em Conteúdo |
| Modelagem e validação | Spacy | 3.7.2 | Preparação dos dados para os modelos de Filtragem Baseada em Conteúdo |
| Processamento de dados | Chardet | 4.0.0 | Checagem e transformação do formato de codificação das bases de dados para “Unicode Transformation Format 8 [UTF-8]” |
| Processamento de dados | Numpy | 1.23.5 | Manipulação das bases de dados |
| Processamento de dados | Pandas | 1.5.3 | Leitura e manipulação das bases de dados |
| Processamento de dados | Re | 2.2.1 | Aplicação de expressões regulares na limpeza e transformação dos dados |
| Visualização de dados | Matplotlib | 3.6.3 | Criação de gráficos |
| Visualização de dados | Mplcyberpunk | 0.7.0 | Estilização de gráficos |
| Visualização de dados | Plotly | 5.22.0 | Criação de gráficos |
| Visualização de dados | Seaborn | 0.12.2 | Criação de gráficos |

Fonte: Dados originais da pesquisa

Por fim, vale salientar que os códigos foram executados em um notebook com processador Intel® Core™ i5-8300H com frequência de 2,30GHz, placa de vídeo NVIDIA GeForce GTX 1050, 12 GB de RAM e sistema operacional Windows 10 baseado em 64 bits.

**Coleta de Dados**

As bases de dados encontram-se publicamente disponíveis na plataforma Kaggle e consistem em informações de animes, usuários e avaliações dos usuários sobre os animes assistidos em um dos principais sites do ramo.

A coleta foi realizada por um usuário do Kaggle utilizando a “Application Programming Interface [API]” Jikan durante 06 jul. 2023 a 06 out. 2023 abrangendo dados até o final do processo.

Foram exportadas três bases de dados em formato “Comma-Separated Values [CSV]”, no qual a primeira contém informações sobre os animes disponíveis, com 24 variáveis e 24.905 observações. A segunda contém informações sobre os usuários disponíveis, com 16 variáveis e 731.290 observações. E a terceira é constituída pelas avaliações dos usuários sobre os animes assistidos, com cinco variáveis e 24.325.191 observações.

**Dicionário das Variáveis**

O próximo passo foi realizar o estudo das bases e entender o que cada variável representa, resultando nos dicionários de variáveis abaixo representados pela Tabela 2, Tabela 3 e Tabela 4.

Ademais, como os modelos de Filtragem Baseada em Conteúdo utilizam Processamento de Linguagem Natural [PLN], é importante destacar que os metadados dos animes, como os gêneros atribuídos, o formato da animação e o material original da história, foram mantidos como texto, em vez de serem transformados em categorias.

Tabela 2. Dicionário das variáveis da base de dados de animes

| Variável | Descrição | Tipo de dado | Amostra |
| --- | --- | --- | --- |
| Anime ID | Número de identificação | Número inteiro | 1 |
| Name | Nome original | Texto | Cowboy Bebop |
| English Name | Versão do nome em Inglês | Texto | Cowboy Bebop |
| Other Name | Versão do nome em Japonês | Texto | カウボーイビバップ |
| Score | Pontuação média seguindo a equação Bayesiana | Texto | 8,75 |
| Genres | Gêneros atribuídos | Texto | Action, Award Winning, Sci-Fi |
| Synopsis | Breve descrição da história | Texto | Crime is timeless. By the year 2071... |
| Type | Formato da animação | Texto | TV |
| Episodes | Número de episódios. Filmes são considerados como um único episódio | Texto | 26,0 |
| Aired | Período do início e final do lançamento | Texto | Apr 3, 1998 to Apr 24, 1999 |
| Premiered | Temporada de lançamento | Texto | Spring 1998 |
| Status | Status de lançamento | Texto | Finished Airing |
| Producers | Produtoras envolvidas na produção | Texto | Bandai Visual |
| Licensors | Licenciadoras e plataformas de vídeo envolvidas na distribuição | Texto | Funimation, Bandai Entertainment |
| Studios | Estúdios envolvidos na produção | Texto | Sunrise |
| Source | Material original da história | Texto | Original |
| Duration | Duração de cada episódio | Texto | 24 min per ep |
| Rating | Classificação etária | Texto | R - 17+ (violence & profanity) |
| Rank | Classificação na plataforma pelo “Score” | Texto | 41,0 |
| Popularity | Classificação na plataforma pela popularidade do momento | Número inteiro | 43 |
| Favorites | Quantidade de usuários que marcaram o item como favorito | Número inteiro | 78.525 |
| Scored By | Quantidade de usuários que avaliaram o item | Texto | 914.193,0 |
| Members | Quantidade de usuários que incluíram o item na lista para assistir | Número inteiro | 1.771.505 |
| Image Url | Imagem promocional | Texto | https://cdn.myanimelist.net... |

Fonte: Dados originais da pesquisa

Tabela 3. Dicionário das variáveis da base de dados de usuários

| Variável | Descrição | Tipo de dado | Amostra |
| --- | --- | --- | --- |
| MAL ID | Número de identificação | Número inteiro | 1 |
| Username | Nome na plataforma | Texto | Xinil |
| Gender | Gênero sexual | Texto | Male |
| Birthday | Data de nascimento | Texto | 1985-03-04T00:00:00+00:00 |
| Location | Localização, podendo ser expressa em país, estado e cidade ou na combinação dos três | Texto | California |
| Joined | Data de quando a conta foi criada | Texto | 2004-11-05T00:00:00+00:00 |
| Days Watched | Quantidade de dias assistidos na plataforma | Número decimal | 142,3 |
| Mean Score | Pontuação média das avaliações seguindo a Média Bayesiana | Número decimal | 7,37 |
| Watching | Quantidade de itens sendo assistidos | Número decimal | 1,0 |
| Completed | Quantidade de itens já completamente assistidos | Número decimal | 233,0 |
| On Hold | Quantidade de itens que o usuário parou de assistir mas deixou na lista | Número decimal | 8,0 |
| Dropped | Quantidade de itens que o usuário parou de assistir e retirou da lista | Número decimal | 93,0 |
| Plan to Watch | Quantidade de itens que o usuário adicionou na lista e ainda não começou a assistir | Número decimal | 64,0 |
| Total Entries | Quantidade total de itens na lista | Número decimal | 399,0 |
| Rewatched | Quantidade de itens reassistidos | Número decimal | 60,0 |
| Episodes Watched | Quantidade de episódios assistidos de todos os itens | Número decimal | 8.458,0 |

Fonte: Dados originais da pesquisa

Tabela 4. Dicionário das variáveis da base de dados de avaliações

|  |  |  |  |
| --- | --- | --- | --- |
| Variável | Descrição | Tipo de dado | Amostra |
| User ID | Número de identificação do usuário na plataforma | Número inteiro | 1 |
| Username | Nome do usuário na plataforma | Texto | Xinil |
| Anime ID | Número de identificação do anime na plataforma | Número inteiro | 21 |
| Anime Title | Nome original do anime na plataforma | Texto | One Piece |
| Rating | Pontuação que o usuário atribuiu ao anime | Número inteiro | 9 |

Fonte: Dados originais da pesquisa

**Limpeza e Transformação dos Dados**

Esta etapa foi dividida em quatro partes e trata-se apenas das transformações aplicadas às três primeiras bases de dados. A primeira é composta por limpezas e transformações aplicadas nas bases de forma geral, enquanto as demais consistem em processos aplicados individualmente em cada uma.

**Limpezas e Transformações Gerais**

As três primeiras bases de dados passaram pela checagem e transformação da codificação do arquivo para o formato UTF-8, além de padronizar as variáveis em minúsculas, substituir espaços por sublinhados e adotar palavras no idioma inglês americano para o nome das variáveis. Os nomes finais das variáveis são os abaixo:

• Base de Dados de Animes: “id”, “title”, “english\_title”, “japanese\_title”, “score”, “genres”, “synopsis”, “type”, “episodes”, “aired”, “premiered”, “status”, “producers”, “licensors”, “studios”, “source”, “duration”, “rating”, “rank”, “popularity”, “favorites”, “scored\_by”, “members” e “image\_url”;

• Base de Dados de Usuários: “id”, “name”, “gender”, “birthday”, “location”, “joined”, “days\_watched”, “mean\_score”, “watching”, “completed”, “on\_hold”, “dropped”, “plan\_to\_watch”, “total\_entries”, “rewatched” e “episodes\_watched”;

• Base de Dados de Avaliações: “user\_id”, “username”, “anime\_id”, “anime\_title” e “rating”.

**Limpeza e Transformação na Base de Dados de Animes**

Foram removidos animes não anunciados ou lançados ainda, identificados pelos valores “Not yet aired” na variável “status” e “Not available” na variável “aired” respectivamente. Em seguida, foram descartadas as variáveis “english\_title”, “japanese\_title”, “aired” e “premiered”, pois somente a variável “title” foi utilizada para identificar os itens pelo nome, e os itens não anunciados e/ou lançados foram removidos.

Depois, foi necessário tratar os valores ausentes, substituindo todos os “UNKNOWN” por -1 nas variáveis “score”, “episodes”, “rank” e “scored\_by” e converter o tipo destas variáveis de texto para número inteiro; enquanto que os “UNKNOWN” em “genres”, “producers”, “licensors”, “studios” e “’rating” foram substituídos por um hífen “-”. Já na variável “synopsis”, os valores ausentes estavam representados como “No description available for this anime” e foram também substituídos por um hífen “-”.

Após filtrar as observações e substituir os valores ausentes, foi realizada a padronização das variáveis “title” e “synopsis”, convertendo os valores para minúsculo, sem acentos, e consistidos em apenas letras do alfabeto arábico, números, vírgulas, espaços e hifens, sendo "◯" o único caractere japonês mantido e substituído pela sua forma em "Rōmaji", "Maru". Ele foi mantido pelo fato de que há uma obra no qual o título original é apenas o caractere e, consequentemente, ficaria com o nome em branco após o descarte de todos os caracteres japoneses.

Por fim, foi necessário verificar a existência de observações duplicadas. Alguns itens possuíam o mesmo título e, após checá-los na plataforma onde os dados foram extraídos, foi constatado de que consistem em sequências das obras, portanto, foi apenas adicionado uma numeração ao final do título. Por exemplo, o título “5-toubun no hanayom” aparece duas vezes porque o anime possui duas temporadas, logo, uma observação ficou com o título original, enquanto que a outra ficou com o título “5-toubun no hanayom 2”.

Já sobre os gêneros, produtoras, licenciadoras e estúdios, os valores duplicados foram excluídos. Tomando os gêneros como exemplo, o valor “mistery, sci-fi, mistery, action” tornou-se “mistery, sci-fi, action”.

**Limpeza e Transformação na Base de Dados de Usuários**

Foram removidos usuários que desativaram as contas, identificados pelos valores nulos nas variáveis “days\_watched”, “mean\_score”, “watching”, “completed”, “on\_hold”, “dropped”, “plan\_to\_watch”, “total\_entries”, “rewatched” e “episodes\_watched”. Em seguida, foram descartados “birthday” e “episodes\_watched”, pois não seriam utilizados no estudo.

Depois, os valores ausentes foram tratados, sendo substituídos por hifens “-” em “gender”, além de forçar o nome de usuário “None” para ser do tipo de dado texto, pois este valor é considerado como nulo dependendo da versão do Python.

Após isto, todas as variáveis numéricas foram convertidas de ponto flutuante para inteiro, pois todos os valores contidos nelas são e podem ser apenas inteiros.

Posteriormente, foi verificado se todos os usuários possuem pontuação média de avaliação entre zero e dez, sendo constatado apenas um com valor acima do esperado e, portanto, sendo truncado para dez.

Por fim, os valores das variáveis “name”, “joined” e “gender” foram padronizados para minúsculo.

**Limpeza e Transformação na Base de Dados de Avaliações**

Foram removidas observações que contêm os animes e/ou os usuários que foram removidos nas duas bases de dados anteriores, além de forçar o nome de usuário “None” para ser do tipo de dado texto.

Em sequência, foi necessário realizar a padronização da variável “anime\_title” da mesma forma que foi feito nas “title” e “synopsis” da base de animes, além de tratar os títulos duplicados conforme realizado na mesma base de dados. Por fim, os valores da variável “username” foram padronizados para minúsculo.

A Tabela 5 indica a quantidade de variáveis e observações antes e depois da limpeza e transformação dos dados em cada base.

Tabela 5. Quantidade de variáveis e observações antes e depois da Limpeza e Transformação dos dados

|  | Base de dados de animes | Base de dados de usuários | Base de dados de avaliações |
| --- | --- | --- | --- |
| Variáveis iniciais | 24 | 16 | 5 |
| Variáveis finais | 21 | 14 | 5 |
| Variáveis removidas (%) | 12,5000% | 12,5000% | 0,0000% |
| Observações iniciais | 24.905 | 731.290 | 24.325.191 |
| Observações finais | 23.748 | 731.282 | 23.796.586 |
| Observações removidas (%) | 4,6456% | 0,0011% | 2,1731% |

Fonte: Dados originais da pesquisa

**Modelos de Recomendação**

Sete modelos de recomendação foram criados neste trabalho, sendo dois de Filtragem Demográfica, dois de Filtragem Baseada em Conteúdo, dois de Filtragem Colaborativa e um de Filtragem Híbrida. As recomendações de todos retornam o código de identificação do item, título, pontuação média, gêneros, imagem promocional e pontuação estimada dos animes.

De todas as bases disponíveis, apenas as de usuários e avaliações foram utilizadas no processo de treinamento, validação e recomendação, sendo a primeira base aplicada em todos os modelos, e a segunda, somente nos de Filtragem Colaborativa e Filtragem Híbrida.

O processo de preparação dos dados, treinamento e validação de cada modelo foram descritos nas subseções Filtragem Demográfica, Filtragem Baseada em Conteúdo, Filtragem Colaborativa e Filtragem Híbrida.

Além disso, foram realizadas seções de desempenho após o treinamento de cada modelo. Esta etapa consistiu no treinamento, validação e geração de recomendações por dez iterações para cada modelo. Os consumos de processamento, memória e tempo de execução foram medidos por meio de processamento paralelo e, ao final, armazenados na base de dados de performances, resultando em nove variáveis e 70 observações como representado pela Tabela 6.

Tabela 6. Dicionário das variáveis da base de dados de performances dos algoritmos de recomendação

|  |  |  |  |
| --- | --- | --- | --- |
| Variável | Descrição | Tipo de dado | Amostra |
| Iteration | Número da iteração do algoritmo | Número inteiro | 0 |
| Algorithm | Nome do algoritmo | Texto | Demographic Filtering – Bayesian Mean |
| Execution Time | Tempo de execução em segundos | Número decimal | 0,1201249998994171 |
| Avg CPU Usage | Consumo médio de CPU em porcentagem | Número decimal | 0,0 |
| Min CPU Usage | Consumo mínimo de CPU em porcentagem | Número decimal | 0,0 |
| Max CPU Usage | Consumo máximo de CPU em porcentagem | Número decimal | 0,0 |
| Avg RAM Usage | Consumo médio de RAM em porcentagem | Número decimal | 0,7660954924023228 |
| Min RAM Usage | Consumo mínimo de RAM em porcentagem | Número decimal | 0,7660954924023228 |
| Max RAM Usage | Consumo máximo de RAM em porcentagem | Número decimal | 0,7660954924023228 |

Fonte: Dados originais da pesquisa

**Filtragem Demográfica**

Dois modelos foram criados utilizando a metodologia de Filtragem Demográfica, enquanto o primeiro (modelo A) realiza recomendações dos itens com maiores pontuações de Média Bayesiana, o segundo (modelo B) realiza recomendações levando em consideração a popularidade dos itens entre os usuários.

Como a plataforma onde os dados foram coletados utiliza a Média Bayesiana para as pontuações dos animes[[1]](#footnote-2), o modelo A precisou apenas ordenar os itens pela variável “score” de modo descendente e retornar os dez com maiores valores.

Já o modelo B trabalhou com a variável “popularity”, ordenando os itens de modo ascendente e retornando os dez com menores valores, ou seja, os dez mais acima na classificação de popularidade.

A Média Bayesiana se mostra mais vantajosa em comparação com a Média Aritmética, pois combina a média geral da base de dados com a média individual de cada item. À medida que um item recebe mais avaliações, o resultado final tende a refletir mais a média individual daquele item. Por outro lado, quando um item tem poucas avaliações, o resultado será mais influenciado pela média geral da base de dados. Essa característica impede que um item, como um anime com poucas avaliações, seja injustamente classificado como ruim com base na Média Aritmética. Por exemplo, um anime com média igual a dois em um escala de zero a dez, mesmo com avaliações de menos de 10% dos usuários, não será considerado ruim pela Média Bayesiana.

Caso fosse necessário calcular a Média Bayesiana no primeiro modelo, aplicar-se-ia a fórmula (1) (Khatri, 2017).

|  | (1) |
| --- | --- |

onde c: é a quantidade mínima de avaliações que o item necessita obter para ser incluído no cálculo; m: é a Média Aritmética de toda a base de dados; n: é a quantidade de avaliações que o item obteve; e r: é a Média Aritmética do item.

Devido ao fato de mais de 75% dos usuários não terem informado suas localizações e uma parte significativa dos restantes forneceram localizações inexistentes, todos os usuários foram considerados como uma única comunidade na Filtragem Demográfica.

**Filtragem Baseada em Conteúdo**

Dois modelos foram criados utilizando a metodologia de Filtragem Baseada em Conteúdo, enquanto o primeiro (modelo C) realiza recomendações de itens com sinopses similares, o segundo (modelo D) realiza recomendações de itens com os metadados de gêneros, tipos e fontes originais semelhantes. Com exceção da remoção de Palavras de Parada e da Lematização, todos os métodos descritos abaixo foram aplicados em ambos os modelos, sendo as variáveis utilizadas a única diferença no processo.

Inicialmente, todas as palavras foram padronizadas em minúsculas, sem quebras de linha, sem caracteres especiais e sem pontuações. Em seguida e apenas no modelo C, todas as Palavras de Parada foram removidas, pois seus significados não tinham tanta relevância e iriam apenas enviesar o treinamento do modelo. Também, todos os pronomes próprios foram removidos porque foi constatado de que este tipo de palavra possuía um grande peso no cálculo de similaridade, resultando em recomendações de animes com sinopses distantes em similaridade, mas com o pronome próprio em comum.

Considerando uma palavra equivalente a um “token”, uma Bolsa de Palavras foi criada no qual cada elemento foi transformado em sua forma de dicionário por meio da Lematização e teve seu peso de frequência obtido aplicando a técnica Frequência do Termo – Frequência Inversa dos Documentos [FT–FID].

Depois, a similaridade entre cada elemento foi calculada utilizando a Similaridade do Cosseno, dada pela equação (2) (Rahutomo et al., 2012).

|  | (2) |
| --- | --- |

onde Wq e Wd: são as observações utilizadas no treinamento do modelo; k: é cada palavra presente na Bolsa de Palavras; t: é a quantidade de palavras na Bolsa de Palavras; e Wqk e Wdk: são a frequência da palavra “k” nas observações “Wq” e “Wd”.

Por fim, foi criada a função de recomendação de itens, que recebe o título do item como parâmetro e retorna os dez animes mais similares em questão de sinopse (modelo C) e de metadados (modelo D).

**Filtragem Colaborativa**

Dois modelos foram criados utilizando a metodologia de Filtragem Colaborativa. Enquanto o primeiro (modelo E) realiza recomendações de itens que usuários semelhantes avaliaram positivamente, o segundo (modelo F) realiza recomendações de itens semelhantes aos que o usuário selecionado avaliou positivamente nas predições, sendo todos os itens recomendados consistentes em animes que usuários semelhantes ao selecionado avaliaram positivamente.

Iniciando pelo modelo E, as variáveis “username” e “anime\_title” foram descartadas, e as observações foram filtradas para considerar somente as avaliações pertencentes aos animes que estão no quantil 75 ou acima na quantidade de usuários que avaliaram o item, reduzindo a base de dados significativamente de 23 milhões de amostras para 82.653.

Em seguida, a base de dados foi dividida em treino e validação. A matriz de avaliações, que organiza as interações entre usuários e animes, foi montada por meio de uma tabela pivô da base de dados, tomando o número de identificação do usuário como índice. Depois, foi padronizada subtraindo as avaliações de cada anime pela respectiva média aritmética. Por fim, foi calculada a matriz de similaridade com base na Correlação de Pearson aplicada às avaliações. A Correlação de Pearson consiste na equação (3) (Fávero e Belfiore, 2017).

|  | (3) |
| --- | --- |

onde Xi e Yi: são duas observações que estão sendo utilizadas para o cálculo de correlação; X e Y: são as médias aritméticas de ambas as observações; e n: é a quantidade total de pares de observações.

Posteriormente, predições de avaliações foram calculadas para cada par de usuário e anime que apresentavam valores zerados nas avaliações. No cálculo de cada predição, foi realizada uma iteração entre todos os usuários similares ao usuário de contexto, incrementando os valores de duas variáveis: pontuação total e peso total. A pontuação total foi incrementada pelo valor da avaliação padronizada multiplicada pelo peso de similaridade do usuário de iteração em relação ao usuário de contexto. O peso total foi incrementado pelo valor absoluto da similaridade do usuário de iteração em relação ao de contexto.

Após este processo, a pontuação predita foi obtida como a média aritmética de todas as avaliações fornecidas pelo usuário de contexto somada à pontuação total e, no final, dividida pelo peso total. Depois de todos os cálculos de predições, a validação do modelo deu-se pela métrica Erro Quadrático Médio da Raiz [EQMR], dada pela equação (4) (Chai e Draxler, 2014).

|  | (4) |
| --- | --- |

onde n: é a quantidade total de predições realizadas; i: é a predição atual que está sendo avaliada; ei2: é o erro elevado ao quadrado, sendo que o erro é consistido pela diferença entre o valor predito e o valor real esperado.

O modelo F passou pelos mesmos processos com algumas diferenças. As observações foram filtradas para considerar somente as avaliações pertencentes aos animes que possuem quantidade de avaliações igual ou maior do que 75 mil, aumentando a base de dados para 651.318 amostras. A matriz de avaliações foi montada usando o número de identificação do anime com índice ao invés do de usuário. Nos cálculos das pontuações preditas, foram utilizados os animes ao invés dos usuários nas iterações.

Vale salientar de que o EQMR para o modelo E foi de 1,2704, e para o modelo F, 5,7741.

**Filtragem Híbrida**

Apenas um modelo (modelo G) foi criado utilizando a metodologia de Filtragem Híbrida, mesclando o modelo A (Filtragem Demográfica utilizando a Média Bayesiana), o modelo D (Filtragem Baseada em Conteúdo utilizando os metadados) e o modelo E (Filtragem Colaborativa utilizando usuários semelhantes).

Os processos de treino e validação foram os mesmos explicados nas subseções anteriores de cada modelo, diferindo apenas na etapa de geração das sugestões.

Durante as recomendações, o algoritmo segue os seguintes passos: 1) verifica se o usuário selecionado existe na matriz de avaliações. Caso exista, o modelo E é ativado; 2) se o usuário não existir e o anime escolhido como base estiver contido na matriz de similaridade dos itens, o modelo D é ativado; e 3) caso contrário, a recomendação geral do modelo A é retornada.

**Resultados e Discussão**

A fim de facilitar o entendimento de cada segmento dos resultados, esta seção foi subdividida em três partes: Análise de Performances, Análise de Recomendações e Limitações.

**Análise de Performances**

Mensurações de performances foram realizadas após o treinamento de cada modelo levando em consideração sete variáveis: tempo de execução, uso mínimo de processamento, uso máximo de processamento, uso médio de processamento, uso mínimo de memória, uso máximo de memória e uso médio de memória; sendo a primeira medida em segundos, e as demais, em porcentagem.

O processo de coleta foi realizado individualmente para cada modelo durante dez iterações, englobando as fases de treinamento, validação e recomendação de itens. Após isto, as informações foram exportadas para a base de dados de Performances de Algoritmos de Recomendação em formato CSV, conforme descrito na seção “Material e Métodos”.

De todas as variáveis, destacam-se a de tempo de execução, uso médio de processamento e uso médio de memória. Como presente na Tabela 7, Figura 1, Figura 2 e Figura 3, percebe-se que os modelos de Filtragem Demográfica são os menos custosos nos três quesitos, já que não utilizam nenhum cálculo complexo ou algoritmo de inteligência artificial, mas sim, apenas filtram a base de dados diretamente para fornecer as recomendações.

Por outro lado, os de Filtragem Baseada em Conteúdo apresentaram-se mais custosos que os anteriores. O modelo C, que utilizou as sinopses das obras para medir a similaridade dos itens, levou em média 288 segundos para ser treinado, validado e testado; enquanto o D, que utilizou os metadados, levou em média apenas 7 segundos. Essa diferença significativa entre os tempos dá-se pelo motivo das sinopses apresentarem muitas palavras diferentes e, consequentemente, resultando numa bolsa de palavras densa de termos e num maior processamento utilizado no cálculo das similaridades dos itens.

Outro fator importante sobre esses modelos é a quantidade de memória utilizada. O primeiro consumiu, em média, apenas 3,80% de RAM, enquanto que o segundo consumiu 19,09%. Portanto, o uso de memória apresenta correlação negativa com o tempo de execução no contexto da Filtragem Baseada em Conteúdo pelos metadados.

Nos modelos de Filtragem Colaborativa, ambos apresentaram a mesma proporção média de processamento, no entanto, o tempo gasto e a média de uso de memória no modelo F, que utilizou a similaridade dos itens dos usuários similares ao usuário de contexto, foram bem maiores. Isto acontece devido ao fato da matriz de similaridade ser esparsa, ou seja, contém muitos itens que não foram avaliados por muitos usuários, resultando em mais cálculos sequenciais de predições de avaliações e, consequentemente, em um maior espaço na memória para armazenar os resultados de cada predição. O consumo de processamento não foi afetado pelos cálculos de predições devido ao fato de terem sido realizados linearmente, sem o uso de computação paralela; e como ambos os modelos aplicam a mesma equação para encontrar as similaridades entre as entidades de estudo, os consumos de processamento deles foram similares.

Por fim, o modelo de Filtragem Híbrida (G) apresentou médias de execução, de consumo de processamento e de consumo de memória esperadas devido ao fato de combinar três modelos diferentes pertencentes às abordagens de filtragem anteriores.

Tabela 7. Média e desvio padrão do tempo de execução em segundos, uso médio de memória em porcentagem e uso médio de processamento em porcentagem para cada algoritmo de recomendação

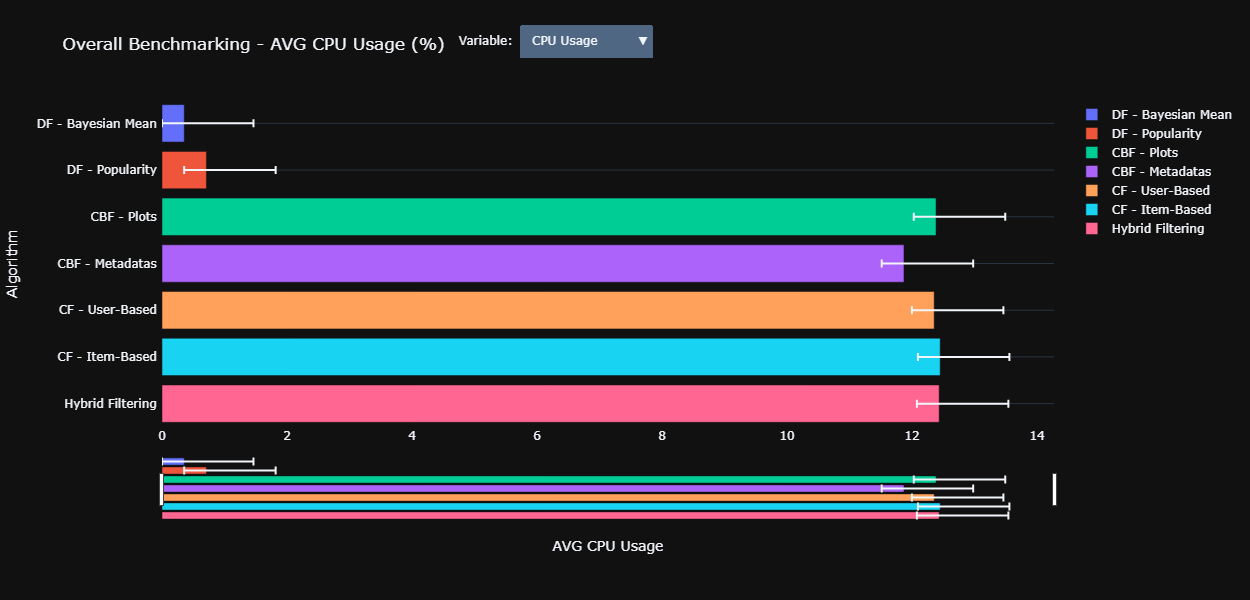
| Algoritmo | Tempo de execução (segundos) | | Uso médio de RAM (%) | | Uso Médio de CPU (%) | |
| --- | --- | --- | --- | --- | --- | --- |
| Média | Desvio padrão | Média | Desvio padrão | Média | Desvio padrão |
| FD – Média Bayesiana | 0,1097 | 0,0038 | 0,7663 | 0,0001 | 0,3562 | 0,7510 |
| FD – Popularidade | 0,1095 | 0,0030 | 0,7695 | 0,0001 | 0,7112 | 0,9182 |
| FBC – Sinopses | 288,5661 | 23,1040 | 3,8119 | 0,2412 | 12,3847 | 0,0515 |
| FBC – Metadados | 7,5576 | 1,8656 | 19,0978 | 2,2515 | 11,8716 | 0,3874 |
| FC – Usuários | 101,3247 | 2,3180 | 1,4885 | 0,0229 | 12,3551 | 0,0650 |
| FC – Itens | 587,7954 | 17,9260 | 2,2197 | 0,3075 | 12,4511 | 0,0295 |
| Filtragem Híbrida | 161,7637 | 3,1330 | 34,2298 | 3,3361 | 12,4347 | 0,0866 |

Fonte: Resultados originais da pesquisa

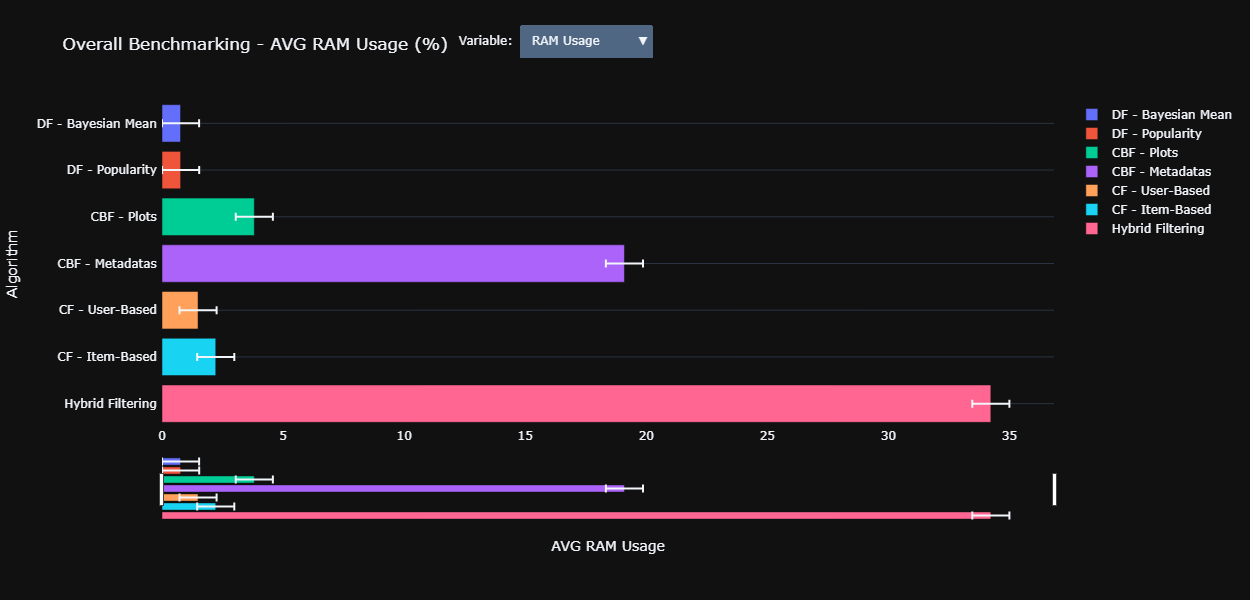
Nota: Onde está escrito “FD” lê-se “Filtragem Demográfica”, “FBC” lê-se “Filtragem Baseada em Conteúdo” e “FC” lê-se “Filtragem Colaborativa”

  
Figura 1: Tempo médio de execução dos modelos

Fonte: Resultados originais da pesquisa

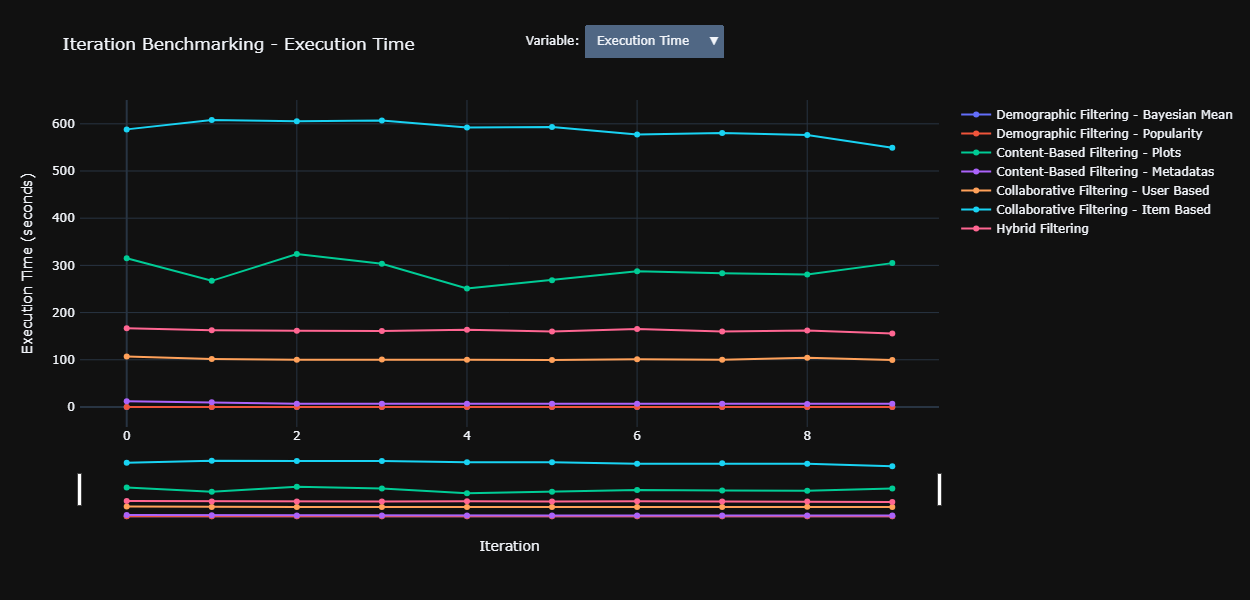
  
Figura 2: Consumo médio de processamento de CPU dos modelos

Fonte: Resultados originais da pesquisa

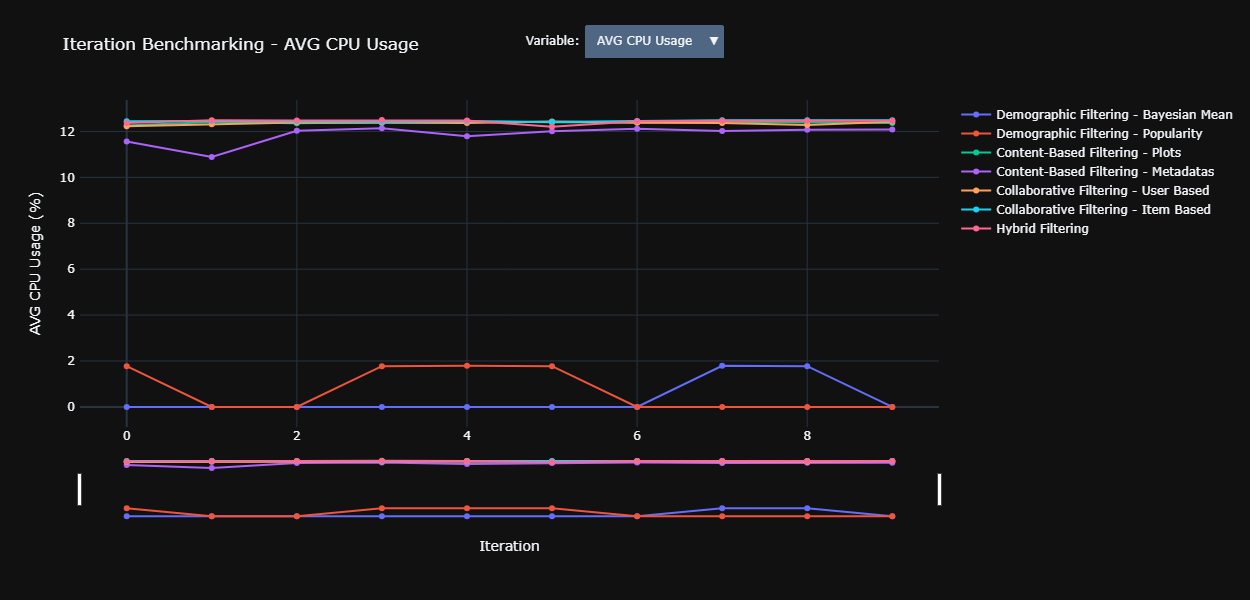
  
Figura 3: Consumo médio de memória RAM dos modelos

Fonte: Resultados originais da pesquisa

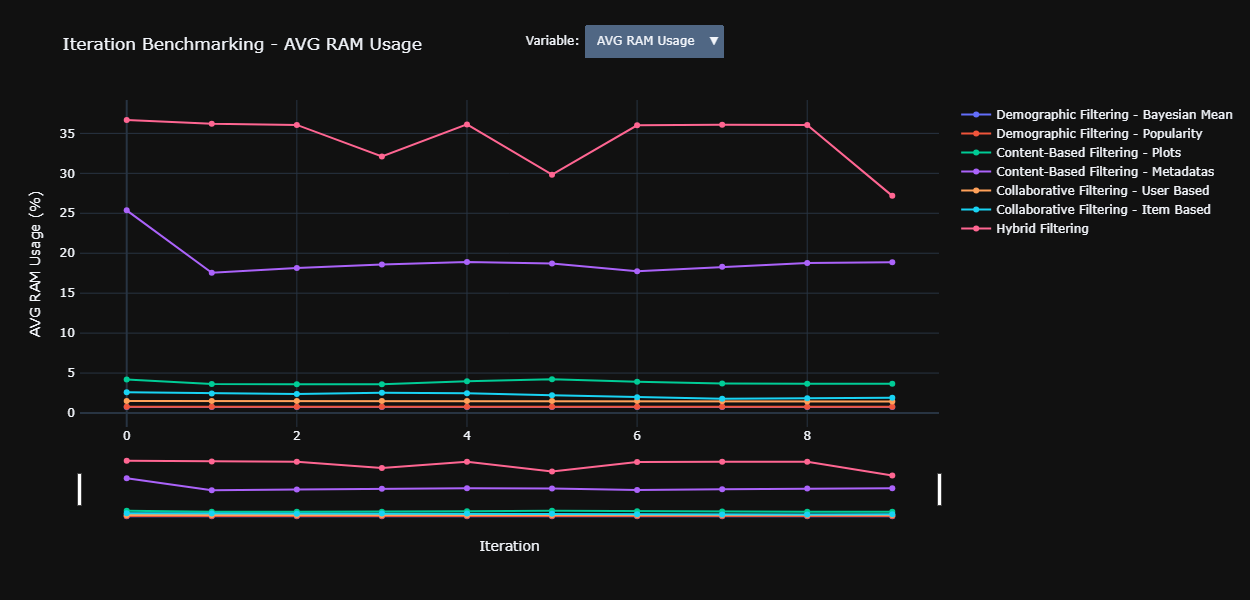
Por fim, vale salientar de que as três variáveis não apresentaram alta variabilidade durante cada iteração, corroborando a hipótese de que os resultados de performances possuem boa acurácia na máquina utilizada pelo autor. A Figura 4, Figura 5 e Figura 6 apresentam os valores de cada variável ao decorrer das dez iterações.

  
Figura 4: Tempo de execução ao decorrer das iterações

Fonte: Resultados originais da pesquisa

  
Figura 5: Consumo de processamento de CPU ao decorrer das iterações

Fonte: Resultados originais da pesquisa

  
Figura 6: Consumo de memória RAM ao decorrer das iterações

Fonte: Resultados originais da pesquisa

**Análise de Recomendações**

Iniciando pelos modelos A e B, ambos pertencentes à Filtragem Demográfica, as recomendações sempre serão as mesmas das apresentadas na Tabela 8 e Tabela 9, independente de qual seja o usuário e o item de contexto. Isto ocorreu devido ao fato dos modelos apenas realizarem filtragens diretamente na base de dados, sem envolver nenhum cálculo matemático.

Tabela 8. Recomendações realizadas pelo modelo A – Filtragem Demográfica pela Média Bayesiana

| Id | Título | Média Bayesiana | Popularidade |
| --- | --- | --- | --- |
| 5114 | Fullmetal Alchemist Brotherhood | 9,10 | 3 |
| 41467 | Bleach Sennen Kessen-Hen | 9,07 | 464 |
| 9253 | Steins Gate | 9,07 | 13 |
| 28977 | Gintama 2 | 9,06 | 331 |
| 38524 | Shingeki no Kyojin Season 3 Part 2 | 9,05 | 24 |

Fonte: Resultados originais da pesquisa

Tabela 9. Recomendações realizadas pelo modelo B – Filtragem Demográfica pela Popularidade

| Id | Título | Média Bayesiana | Popularidade |
| --- | --- | --- | --- |
| 16498 | Shingeki no Kyojin Season 1 | 8,54 | 1 |
| 1535 | Death Note | 8,62 | 2 |
| 5114 | Fullmetal Alchemist Brotherhood | 9,10 | 3 |
| 30276 | One Punch Man | 8,50 | 4 |
| 11757 | Sword Art Online | 7,20 | 5 |

Fonte: Resultados originais da pesquisa

Em relação aos modelos com abordagem de Filtragem Baseada em Conteúdo, o que utilizou as sinopses das obras para o cálculo de similaridades tendeu a recomendar sequências da mesma obra, como filmes, episódios especias e temporadas de continuidade. Esse padrão ocorreu por causa das sequências dos animes apresentarem muitas palavras similares em suas sinopses.

Já o modelo que utilizou os metadados tendeu a recomendar itens com informações similares, ou seja, itens de um mesmo universo de gêneros, corroborando o resultado obtido por Pinheiro (2018) no qual, aplicando Filtragem Baseada em Conteúdo para recomendação de vídeos no YouTube, somente conteúdos relacionados a Programação Orientada a Objetos e Linguagem Java foram recomendados ao buscar recomendações pela palavra-chave “Herança” com filtro para conteúdos educativos.

A Tabela 10 e a Tabela 11 listam as recomendações de ambos os modelos simultaneamente.

Tabela 10. Recomendações realizadas pelo modelo C – Filtragem Baseada em Conteúdo pela sinopse considerando “Fullmetal Alchemist Brotherhood” como item de contexto

| Id | Título | Similaridade |
| --- | --- | --- |
| 121 | Fullmetal Alchemist | 0,5001 |
| 10842 | Fullmetal Alchemist – The Sacred Star of Milos – Specials | 0,2818 |
| 430 | Fullmetal Alchemist – The Conqueror of Shamballa | 0,2635 |
| 9135 | Fullmetal Alchemist – The Sacred Star of Milos | 0,2624 |
| 6421 | Fullmetal Alchemist Brotherhood – Specials | 0,2092 |

Fonte: Resultados originais da pesquisa

Tabela 11. Recomendações realizadas pelo modelo D – Filtragem Baseada em Conteúdo pelos metadados considerando “Fullmetal Alchemist Brotherhood” como item de contexto

| Id | Título | Gêneros | Similaridade |
| --- | --- | --- | --- |
| 28249 | Arslan Senki | Adventure, action, fantasy, drama | 1,0000 |
| 31821 | Arslan Senki Fuujin Ranbu | Adventure, action, fantasy, drama | 1,0000 |
| 589 | Ginga Nagareboshi Gin | Adventure, action, drama | 0,9217 |
| 2243 | Karasu Tengu Kabuto | Adventure, action, drama | 0,9217 |
| 37521 | Vinland Saga | Adventure, action, drama | 0,9217 |

Fonte: Resultados originais da pesquisa

Por outro lado, o modelo E pertencente à Filtragem Colaborativa realizou recomendações diversificadas, listando obras com sinopses e metadados variados e, consequentemente, dando a entender de que os usuários normalmente possuem gostos variados entre si. O mesmo comportamento foi observado por Pinheiro (2018) utilizando o mesmo exemplo de busca por vídeos no YouTube pela palavra-chave “Herança”. Ao adicionar informações sobre o usuário autenticado na plataforma, os conteúdos recomendados não ficaram somente no ramo da programação, mas sim, em ramos variados, tais como músicas, aulas de genética e vídeos humorísticos.

Já o modelo F obteve uma maior tendência de recomendar obras com metadados parecidos, pois foi modelado para realizar indicações de itens similares aos itens bem avaliados por usuários similares ao de contexto.

A Tabela 12 e a Tabela 13 apresentam, simultaneamente, as recomendações listadas pelos modelos.

Tabela 12. Recomendações realizadas pelo modelo E – Filtragem Colaborativa baseada no usuário considerando o usuário com id 609.917 como contexto

| Id | Título | Gêneros | Avaliação Predita |
| --- | --- | --- | --- |
| 4181 | Clannad After Story | Supernatural, romance, drama | 9,7688 |
| 1575 | Code Geass Season 1 | Action, sci-fi, award winning, drama | 9,7496 |
| 2001 | Tengen Toppa Gurren Lagann | Adventure, action, sci-fi, award winning | 9,6618 |
| 9989 | Anohana | Supernatural, drama | 9,6324 |
| 11741 | Fate Zero Season 2 | Action, fantasy, supernatural | 9,6239 |

Fonte: Resultados originais da pesquisa

Tabela 13. Recomendações realizadas pelo modelo F – Filtragem Colaborativa baseada no item considerando o usuário com id 1.129.199 como contexto

| Id | Título | Gêneros | Avaliação Predita |
| --- | --- | --- | --- |
| 20 | Naruto | Adventure, action, fantasy | 9,6196 |
| 269 | Bleach | Adventure, action, fantasy | 9,5821 |
| 1535 | Death Note | Suspense, supernatural | 9,5687 |
| 1575 | Code Geass Season 1 | Action, sci-fi, award winning, drama | 9,4935 |
| 2904 | Code Geas Season 2 | Action, sci-fi, award winning, drama | 9,4912 |

Fonte: Resultados originais da pesquisa

Por fim, como o modelo de Filtragem Híbrida (G) consiste na combinação dos modelos A, D e E, as recomendações seguem os mesmos padrões descritos para cada um destes.

**Limitações**

Vale salientar que os resultados deste trabalho apresentam algumas limitações importantes a serem consideradas, como, por exemplo, os dados das performances coletadas. Estas informações estão sujeitas às configurações da máquina utilizada pelo autor e podem divergir quando executadas em dispositivos com hardwares diferentes.

Além disso, os dados utilizados no treinamento dos modelos são pertencentes a apenas uma plataforma de animes, portanto, todas as recomendações resultantes são válidas apenas para esta plataforma em específico.

Outro ponto relevante é que os dados utilizados abrangem até o período de 06 out. 2023, assim, todos os animes lançados, todas as contas de usuários criadas e todas as avaliações realizadas posteriormente não têm influência alguma nos modelos.

Por fim, como os dados estão relacionados a animes, os modelos estão sujeitos a realizarem recomendações apenas sobre este tipo de conteúdo.

**Conclusões**

Foi investigado o desempenho dos algoritmos de recomendação seguindo as abordagens de Filtragem Demográfica, Filtragem Baseada em Conteúdo, Filtragem Colaborativa e Filtragem Híbrida, mensurando o consumo de processamento, de memória RAM e de tempo de execução, além dos impactos das recomendações geradas pelos modelos.

Em linhas gerais, verificou-se que a complexidade dos algoritmos e a acurácia das recomendações são correlacionados com o maior consumo de hardware e tempo de execução. Também foi observado que quanto maior a acurácia das recomendações desejadas, maior a variedade de dados dos usuários e dos itens requerida e, consequentemente, as recomendações tornam-se personalizadas e mais próximas aos gostos individuais dos usuários.

Em trabalhos futuros, base de dados de diferentes plataformas poderiam ser adicionadas para expandir o contexto dos modelos, além de acrescentar modelos de recomendação utilizando Aprendizado de Máquina Profundo, como Incorporação de Palavras (Word Embedding) e Word2Vec, para a Filtragem Baseada em Conteúdo e Filtragem Híbrida.

**Agradecimento**

Gostaria de expressar minha profunda gratidão a todos que me acompanharam nesta caminhada, fornecendo conhecimentos que levarei por toda a minha vida, principalmente a meus pais e ao meu irmão pelo suporte necessário, não apenas no desenvolvimento deste trabalho, mas também na formação da pessoa que sou hoje.

**Referências**

Burke, R. 2002. Hybrid recommender systems: Survey and experiments. Journal User Modeling and User-Adapted Interaction 12: 331-370. Disponível em: <<https://fcd.ugr.es/sites/centros/fcd/public/ficheros/PROYECTO2023/Burke2002_Article_HybridRecommenderSystemsSurvey.pdf>>. Acesso em: 20 mar. 2024.

Chai, T; Drawler, R.R. 2014. Root mean square error (RMSE) or mean absolute error (MAE)? - Arguments against avoiding RMSE in the literature. Geosci Model Dev 7: 1247-1250. Disponível em: <<https://gmd.copernicus.org/articles/7/1247/2014/gmd-7-1247-2014.pdf>>. Acesso em: 30 maio 2024.

Das, A.; Datar, M.; Garg, A.; Rajaram, S. 2007. Google News Personalization: Scalable Online Collaborative Filtering. In: WWW 2007 / Track: Industrial Practice and Experience, 2007, Banff, Alberta, Canadá. Anais... p. 271-280. Disponível em: <<https://www.ri.cmu.edu/pub_files/2009/0/p271-das.pdf>>. Acesso em: 20 mar. 2024.

Fávero, L.P.; Belfiore, P. 2017. Manual de análise de dados: estatística e modelagem multivariada com Excel®, SPSS® e Stata®. 1ed. Elsevier Editora Ltda, Rio de Janeiro, RJ, Brasil.

Khatri, A. 2017. Popularity based product rating system using Bayesian model. Dissertação de Mestrado em Engenharia da Computação. University of Victoria, Victoria, Colúmbia Britânica, Canadá. Disponível em: <https://dspace.library.uvic.ca/server/api/core/bitstreams/c615f973-11b4-417c-b196-7b4d94bd55b8/content>. Acesso em: 15 mar. 2024.

Pinheiro, R. R. A. 2018. Sistema de recomendação de vídeos educacionais: um estudo de caso no YouTube. Dissertação de Mestrado em Modelagem Computacional de Conhecimento. Universidade Federal de Alagoas, Maceió, Alagoas, Brasil. Disponível em: <[https://www.repositorio.ufal.br/bitstream/riufal/4820/1/Sistema%20de%20recomendação%20de%20vídeos%20educacionais%20um%20estudo%20de%20caso%20no%20You%20tube.pdf](https://www.repositorio.ufal.br/bitstream/riufal/4820/1/Sistema de recomendação de vídeos educacionais um estudo de caso no You tube.pdf)>. Acesso em: 05 ago. 2024.

Rahutomo, F.; Kitasuka, T.; Aritsugi, M. 2012. Semantic Cosine Similarity. In: The 7th International Student Conference on Advanced Science and Technology ICAST, 2012, Seul, Coreia do Sul. Anais... p. 1-2. Disponível em: <<https://www.researchgate.net/profile/Faisal-Rahutomo/publication/262525676_Semantic_Cosine_Similarity/links/0a85e537ee3b675c1e000000/Semantic-Cosine-Similarity.pdf>>. Acesso em: 15 mar. 2024.

Ricci, F.; Rokach, L.; Shapira, B.; Kantor, P.B. 2011. Recommender Systems Handbook. 1ed. Springer, New York, NY, USA. Disponível em: <<https://d1wqtxts1xzle7.cloudfront.net/32978074/Recommender_systems_handbook.pdf>>. Acesso em: 20 mar. 2024.

Silva, R.G.N. e. 2014. Sistema de Recomendação baseado em conteúdo textual: avaliação e comparação. Dissertação de Mestrado Multi-institucional em Ciência da Computação MMCC. Universidade Federal da Bahia e Universidade Estadual de Feira de Santana, Salvador, Bahia, Brasil. Disponível em: <<https://repositorio.ufba.br/bitstream/ri/19281/1/dissertacao_mestrado_ciencia_computacao_rafael_glauber.pdf>>. Acesso em: 20 mar. 2024.

Whittaker, S.; Sidner, C. 1996. Email overload: exploring personal information management of email. In: CHI – Conference on Human Factors in Computing Systems, 1996, Vancouver, British Columbia, Canadá. Anais... p. 276-283. Disponível em: <<https://dl.acm.org/doi/pdf/10.1145/238386.238530>>. Acesso em: 20 mar. 2024.

1. <https://myanimelist.net/info.php?go=topanime> [↑](#footnote-ref-2)