Sistema para disponibilização e recomendação de eventos baseado em histórico e relacionamentos do usuário

Henrique Olivares Dell Agnolo[1], Guilherme Wentz de Moura[2], Daniela Vieira Cunha (Orientadora)[1]

[1]Faculdade de Computação e Informática – Universidade Presbiteriana Mackenzie

R. da Consolação, 930 – 01301-000 – Consolação – São Paulo, SP – Brasil

{henriquedell1@gmail.com, guiwentz97@gmail.com, daniela.cunha@mackenzie.br}

Abstract. *This paper has the purpose to do a study about the feasibility of the Artificial Intelligence algorithm named Hybrid Collaborative Filtering on the events recommendation, based on events characteristics as well as users history. The application was built using the Python programming language; the library called Turi Create for the algorithm usage and, for visualization, it was developed a Web interface. The feasibility of the recommendation algorithm to users was proven through the application results.*

Keywords: *Angular, AI, Collaborative Filtering, Turi Create, Recommendation*

Resumo. *Este artigo tem como propósito realizar um estudo sobre a viabilidade do algoritmo de Inteligência Artificial denominado Filtragem Colaborativa Híbrida na recomendação de eventos, usando como base as características dos eventos assim como o histórico dos usuários. A aplicação foi construída usando a linguagem de programação Python, a biblioteca Turi Create para a criação do algoritmo e, para visualização, foi construída uma interface Web. A viabilidade do algoritmo de recomendação para os usuários foi comprovada através dos resultados da aplicação.*

Palavras-chave: *Angular, IA, Filtragem Colaborativa Híbrida, Turi Create, Recomendação*

1. Introdução

Num mundo cada vez mais globalizado e conectado, a melhor forma de *marketing* é através de um meio digital. O *marketing* de eventos normalmente é feito através de publicações em redes sociais, panfletos, banners entre outros. Porém muitas pessoas só acabam descobrindo sobre algum evento que gostariam de ir após o seu término, tendo em vista a quantidade de pesquisas que precisam fazer para conseguir filtrar e encontrar as atividades que mais lhes agradam além do tempo que aqueles meios de propaganda necessitam para informar o maior número de pessoas. Ou seja, o alcance da divulgação do evento não é eficaz como deveria ser.

Esse problema frustra não só o usuário, por não conseguir encontrar os eventos que lhe agrada de maneira simplificada, mas também o dono do evento que poderia atrair mais pessoas através de um método de divulgação mais eficiente. Um sistema de recomendação capaz de indicar os eventos aos usuários certos minimizaria esse problema.

Por conta de toda praticidade e conforto que os sistemas digitais trouxeram, os usuários não gostam de gastar muito tempo pesquisando pelo que querem, preferem um sistema inteligente que reconheça seus gostos e faça recomendações específicas para eles.

Isso já é realidade em redes sociais, que usam sistemas de recomendação para recomendar amigos além de decidirem quais conteúdos serão mostrados na *timeline (*termo que designa o mural, local no site que junta várias postagens). Em sites de e-commerce, algoritmos de recomendação são usados para recomendação de produtos baseando-se nos produtos já comprados ou acessados anteriormente; em serviços de streaming, a recomendação apresentada, e que se espera ser a mais próxima da vontade do usuário, é realizada de acordo com os programas já assistidos e as avaliações dos usuários, entre outros.

Sistemas de recomendação são basicamente softwares, ferramentas ou técnicas para recuperar, filtrar dados e prover significantes recomendações para usuários. As recomendações são basicamente de dois tipos: personalizadas e não personalizadas. As personalizadas são geradas para um indivíduo único, com base em dados do mesmo, já as não personalizadas são as geradas para um grupo de usuários. Quanto a forma de efetuar uma recomendação, os algoritmos podem ser divididos em 3 principais categorias [Douglas e Gustavo, 2018]:

- recomendações por filtros colaborativos baseados em usuário;
- recomendações por filtros colaborativos baseados no item ou conteúdo;
- recomendações por filtros colaborativos híbridos, criados com base em, pelo menos, uma de algumas formas combinatórias.

Assim, a proposta deste projeto é a verificação da viabilidade do desenvolvimento de um sistema o qual, além de centralizar as informações importantes de um evento, como local, data, tipo de evento, lista de convidados, lista de confirmados, entre outros, é capaz de avaliar o histórico de eventos dos usuários para

poder recomendar para cada um o evento que mais se alinha com suas preferências, usando a filtragem colaborativa híbrida estruturada pela combinação Mista através da análise das recomendações criadas para os usuários.

2. Sistemas de Recomendação

A crescente variedade de informações disponíveis na web e o rápido surgimento de novos serviços de e-business proporcionaram uma sobrecarga de opções aos usuários. Ter opções é algo bom, mas "infinitas" opções nem sempre é o melhor [Hanne, 2017]. Surgem então, os Sistemas de Recomendação (SR), a partir da necessidade de filtrar a quantidade de opções disponíveis para o usuário, automatizando a geração de recomendações baseadas na análise dos dados [Melville e Sindhwani, 2010].

Sistemas de Recomendação é uma área de pesquisa bastante rica que se utiliza de várias técnicas e ferramentas para prover sugestões de itens que sejam úteis para um usuário. No contexto destes sistemas, um item pode ser qualquer coisa que possa ser recomendado a um usuário, tal como um livro, um filme ou um pacote de viagem. [Barbosa, 2014]

Esses sistemas utilizam técnicas de filtragem de Aprendizagem de Máquina (AM) para criar as recomendações. Filtragem de AM é um processo que visa analisar um aglomerado de dados, identificar suas características e extrair, filtrar, os dados pertinentes ao tipo de filtragem que está sendo usado.

2.1. Filtragem Colaborativa Baseada em Usuário

O algoritmo de filtragem baseada em usuário faz recomendações usando dados pré existentes de outros usuários. Dados que podem ser desde o uso de produtos ou serviços, até compras e/ou avaliações. Ou seja, o algoritmo procura usuários com perfis de consumo similares para fazer recomendações. Por exemplo, um usuário X está comprando frutas, morango e melancia por exemplo, em um sistema de mercado online. O sistema então identifica que outros usuários que tiveram um pedido similar também compraram uvas e laranjas, então o sistema irá recomendar ao usuário esses produtos, como demonstrado na Figura 1.

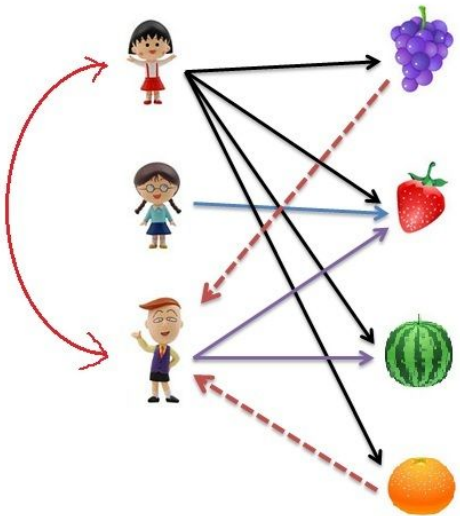


Figura 1. Exemplo de algoritmo de recomendação baseada em usuário. [Fonte: Devmedia, 2015]

Existem algumas vantagens de se usar esse tipo de filtragem. A vantagem mais importante é a capacidade de recomendar itens variados, visto que não leva em consideração as características dos dados analisados.

Existem duas desvantagens que podem ser destacadas:

- quando um usuário acaba de realizar seu cadastro, não há como recomendar algo à ele. Por conta de não haver um histórico de uso não é possível determinar sua similaridade com outro usuário; e
- quando um item é cadastrado, também é um problema conseguir recomendá-lo para os usuários por não estar presente no histórico de ninguém e o sistema não analisar suas características.

Por apresentar mais desvantagens do que vantagens, esse tipo de recomendação não é muito usada de maneira isolada, geralmente é combinada com outros algoritmos que atenuam suas desvantagens.

2.2. Filtragem Colaborativa Baseada em Conteúdo ou Item

No caso do algoritmo de filtragem baseada em conteúdo ou item, ele utiliza as características dos itens em que o usuário demonstrou interesse anteriormente para construir as recomendações e/ou as configurações de preferência do usuário. Ao invés de usar o histórico dos usuários, ele usa a similaridade entre itens. Redes sociais utilizam muito esse tipo de filtragem em seus anúncios, baseando-se em postagens anteriores, configurações do perfil, seus amigos, etc. Um outro exemplo é em sistemas de filmes, um usuário X classifica dois filmes com uma boa nota. Por conta da similaridade desses filmes com um terceiro, o sistema irá recomendar esse para o usuário, como exemplificado pela Figura 2.

Figura 2. Exemplo de algoritmo de recomendação baseada em conteúdo. [Fonte: Fortes, 2014]

Dentre as vantagens deste tipo de filtragem, pode-se citar:

- Todos os itens podem ser indicados, mesmo que sejam novos, já que o sistema utiliza configurações do usuário e não necessita exclusivamente de avaliações anteriores para gerar as recomendações;
- As recomendações tendem a ser muito precisas devido ao fato de o sistema saber quais itens tiveram uma maior interação do usuário.

Há duas desvantagens, principalmente:

- Usuários novos não terão recomendações muito precisas devido ao pouco uso, ou seja, o sistema não tem informações sobre o histórico do usuário.
- Muitas vezes o usuário pode ficar limitado a poucas opções. Na Figura 2 pode-se ver que o filme recomendado é da mesma categoria dos filmes já vistos. Se o usuário gostar de outros tipos, o algoritmo não irá recomendá-los a não ser que o usuário interaja com eles.

2.3. Filtragem Colaborativa Híbrida

A filtragem colaborativa híbrida é a junção das duas citadas nas seções 2.1 e 2.2, filtragem colaborativa baseada em usuário e filtragem colaborativa baseada em conteúdo, respectivamente.

Existem quatro formas principais sobre como essas recomendações serão combinadas [Barbosa, 2014]:

- **Ponderada**: nesta abordagem a filtragem baseada em conteúdo e a filtragem baseada em usuário são implementadas separadamente e uma combinação linear é feita com os seus resultados. Pode ser necessária uma normalização nos resultados individuais, antes de aplicar a combinação linear, caso as técnicas gerem valores em escalas diferentes.
- **Mista**: nesta abordagem as recomendações geradas pelas duas técnicas são combinadas no processo final de recomendação, de tal forma que as duas recomendações sejam apresentadas ao usuário na mesma lista.

- **Combinação sequencial**: nesta abordagem a filtragem baseada em conteúdo cria os perfis dos usuários e, posteriormente, estes perfis são usados no cálculo da similaridade da filtragem colaborativa.
- **Comutação**: nesta abordagem o sistema utiliza algum critério, como por exemplo a confiança no resultado, para comutar ou chavear entre a filtragem baseada em conteúdo e a filtragem baseada em usuário. Pode-se também realizar a comutação de uma técnica nos pontos de desvantagem da outra técnica.

A técnica híbrida possibilita fazer tanto recomendações mais precisas quanto variadas. As recomendações mais precisas serão aquelas baseadas em conteúdo, ou seja, são aquelas em que o sistema identifica os melhores eventos para o usuários baseado em seus interesses. As recomendações mais variadas são aquelas baseadas em usuário, ou seja, o sistema analisa as atividades do usuário e a atividade de seus amigos para fazer recomendações.

As maiores vantagens desta filtragem são:

- recomendações precisas para itens parecidos;
- além de abranger tipos diferentes, sendo capaz de variar as recomendações.

Como essa filtragem abrange duas opções de recomendação, o sistema possuirá recomendações mais diversificadas possibilitando que o usuário sempre tenha recomendações diferentes.

3. Projeto

3.1 Visão geral

O projeto foi pensado e criado a partir de três ideias principais:

- Auxiliar os organizadores de eventos na divulgação de seus projetos;
- Centralizar eventos em uma plataforma única;
- Utilizar IA para recomendação de eventos.

O software proporciona ao usuário uma experiência diferente em relação a busca livre de eventos utilizada através de motores de busca. Por exemplo: hoje o usuário precisa buscar no google, facebook ou outros meios por eventos que ele gostaria de ir. O sistema concentra e especializa essa busca, proporcionando ao usuário mais agilidade nessa pesquisa.

Para os organizadores de eventos, o sistema possibilita um alcance maior de número de pessoas já que muitos usuários, que teriam interesse em seu evento, provavelmente não encontrarão especificamente seu evento.

3.1.1. Cadastro e Login

Para ter acesso ao sistema, é necessário efetuar um cadastro (Figura 4). Caso o cadastro já tenha sido feito anteriormente, é necessário apenas efetuar o Login (Figura 3).

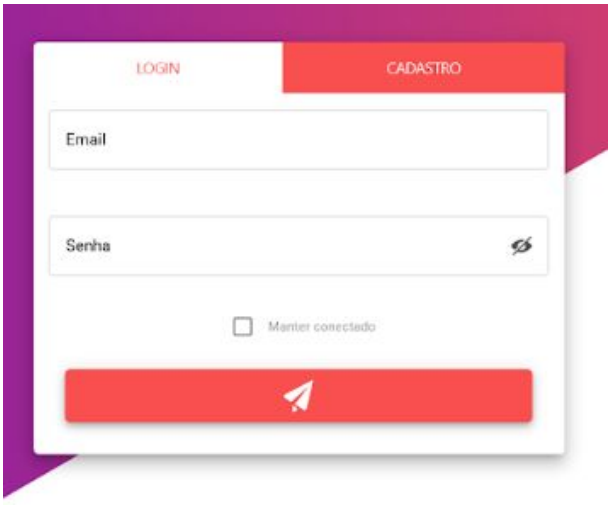


Figura 3. Tela de Login. [Fonte: Autores]

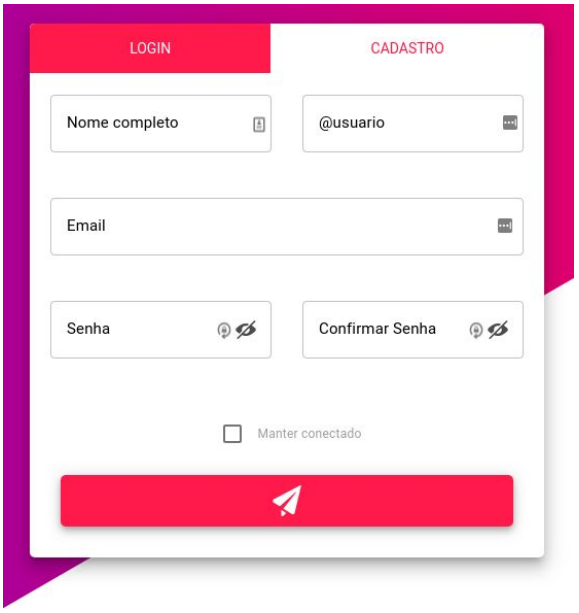


Figura 4. Tela de cadastro. [Fonte: Autores]

3.1.2. Tela principal

Nela, são apresentadas as recomendações, a maneira como são geradas será explicada na seção 3.3 deste artigo. Além disso é possível abrir um menu lateral com outras opções do sistema, apenas clicando nas 3 linhas horizontais (Figura 5 - seta 1). No futuro, poderá ser realizada uma pesquisa através do mecanismo de busca (Figura 5 - seta 2). Para realizar *logoff* do sistema é necessário clicar no ícone que representa uma porta (Figura 5 - seta 3). E por último, clicando no ícone representado por um caderno e uma caneta (Figura 5 - seta 4), é possível criar novos eventos.

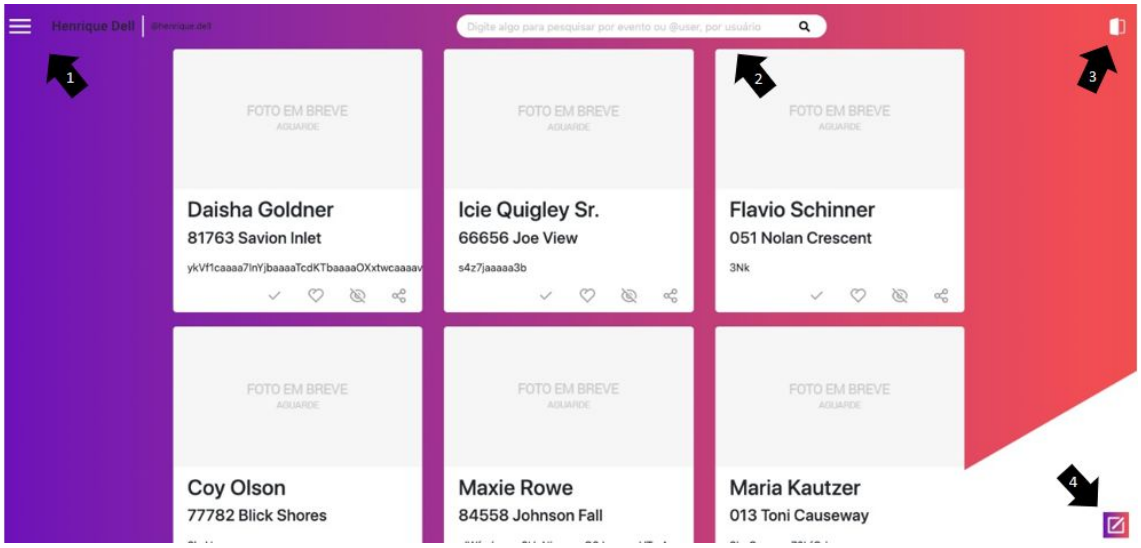


Figura 5. Tela principal da aplicação. [Fonte: Autores]

3.1.3. Opções do Evento na Tela Principal

O usuário possui algumas funcionalidades em relação ao evento através de botões. São 4 opções:

- Confirmação de comparecimento ao evento (Figura 6 - seta 1);
- Adicionar na lista de interesses - Uma marcação de lembrete do evento, porém não é uma confirmação (Figura 6 - seta 2);
- Não ver mais o evento - Caso o usuário não tenha interesse no evento, ele pode marcá-lo com essa opção (Figura 6 - seta 3);
- Compartilhamento de evento (Figura 6 - seta 4).

Figura 6. Card do evento. [Fonte: Autores]

3.1.4. Cadastro de Evento

Uma das principais funcionalidades no sistema é a criação de eventos, qualquer usuário pode criar. Conforme citado na seção 3.1.2, para criar um evento é necessário clicar no ícone representado por um caderno e uma caneta (Figura 5 - seta 4) na tela principal (Figura 5). As informações requisitadas são poucas, porém importantes para o algoritmo de recomendação. As informações são as seguintes:

- Nome do evento (Figura 7 - seta 1);
- Descrição (Figura 7 - seta 2);
- Local (Figura 7 - seta 3);
- Data (Figura 7 - seta 4);
- Idade mínima (Figura 7 - seta 5);
- Idade máxima (Figura 7 - seta 6);
- Categoria (Figura 7 - seta 7).

Para inserir a categoria, é necessário escrever ou selecionar uma opção da lista que aparecerá acima do campo de texto. Após selecionar, é necessário clicar no botão representado pelo símbolo ‘+’ (Figura 7 - seta 8).
Após completar essas informações, o usuário deve clicar no botão salvar (Figura 7 - seta 9).

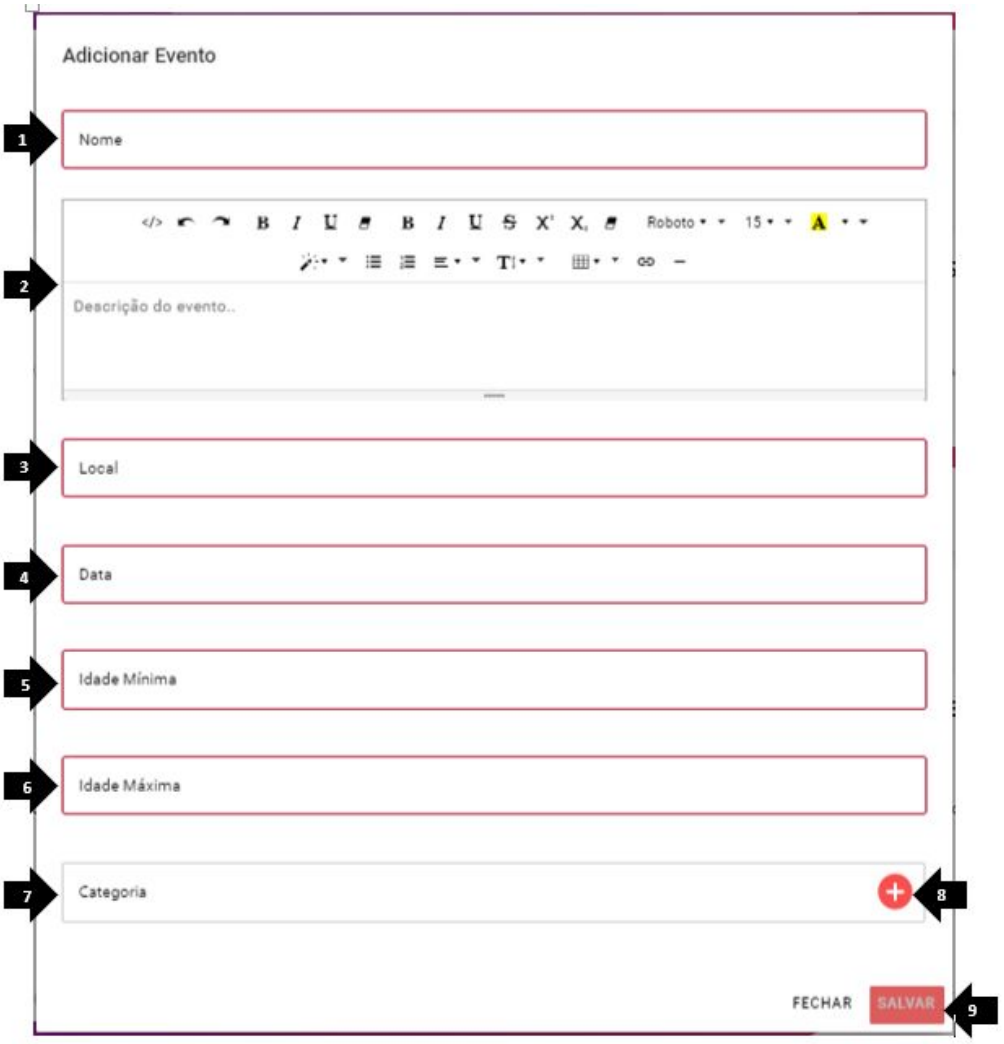


Figura 7. Modal de cadastro de evento. [Fonte: Autores]

3.2 Estrutura

Este projeto foi construído em duas partes:

- aplicação *front-end* para a visualização dos dados e interação dos usuários com a plataforma, desenvolvida usando-se o *framework* de desenvolvimento Web chamado Angular baseado na linguagem JavaScript, o que torna o desenvolvimento muito mais rápido, eficiente e o funcionamento do site mais orgânico, por conta da possibilidade de atualizar as páginas, e partes de páginas, de forma assíncrona diminuindo a carga do servidor, já que o mesmo não precisará mais renderizar as páginas; e
- aplicação *back-end* para controle de acesso, gerenciamento do banco de dados e uso da Inteligência Artificial, desenvolvida usando-se o *framework* de desenvolvimento de aplicações para servidor Web chamado Django baseado em Python. Python foi a linguagem escolhida pois é a linguagem mais preparada para uso de IA, contando com suporte a dezenas de algoritmos diferentes e Django por permitir criar APIs com extrema facilidade, reduzindo muito o tempo de desenvolvimento.

O banco de dados escolhido foi o MongoDB devido ao fato dele ser um tipo de repositório da dados NoSQL, ou seja não estruturado. Ele foi o selecionado para esta aplicação por conta da sua alta capacidade de mutabilidade, o que permitiu que fossem feitas alterações tanto nos modelos quanto nos dados persistidos ao longo do projeto sem necessidade de migrações do banco de dados.

Para uso dos algoritmos de recomendação de IA, foi usado uma biblioteca para Python chamada Turi Create [Apple Github, 2019]. Ela é capaz de criar vários tipos de algoritmos diferentes, desde classificadores de imagem e som até detectores de objetos e identificadores de similaridade de imagem, entre outros. Porém, para esta aplicação foram escolhidos os de recomendação baseada em usuário e baseada em conteúdo.

A aplicação desenvolvida pelos autores foi montada usando-se representações dos objetos reais necessários para seu funcionamento. A Figura 8 ilustra o modelo de domínio da aplicação, a qual reúne as representações:

- Usuário - controle de acesso, ela é quem guarda as credenciais de acesso dos usuários.
- Participante - responsável por conter as informações de identificação do usuário, como seu nome, seu identificador em formato de nome - usado pelos usuários para buscarem por outros usuários -, e seu histórico, os eventos que foram adicionados à lista de interesse ou tiveram presença confirmada. É a representação do perfil do usuário na aplicação;
- Evento - reúne todas as informações relativas aos eventos, como seu nome, sua descrição, entre outros - todas as informações mencionadas na seção 3.1.4 - exceto pelo endereço e categoria, os quais têm suas respectivas classes. Evento é a classe que descreve as características dos eventos;
- Local - guarda as informações de localização geográfica dos eventos, salvando rua, número, cep, entre outros. Ela é usada para impedir repetição desnecessária de informações de endereço no banco de dados;
- Categoria_Evento - uma lista, onde dado um evento, consegue recuperar quais categorias foram cadastradas para ele; e
- Categoria - mantém as diferentes categorias que podem ser usadas no cadastro de eventos. Não é uma listagem final, sendo possível adicionar novas entradas caso necessário pelo formulário de adição de evento, Figura 7.

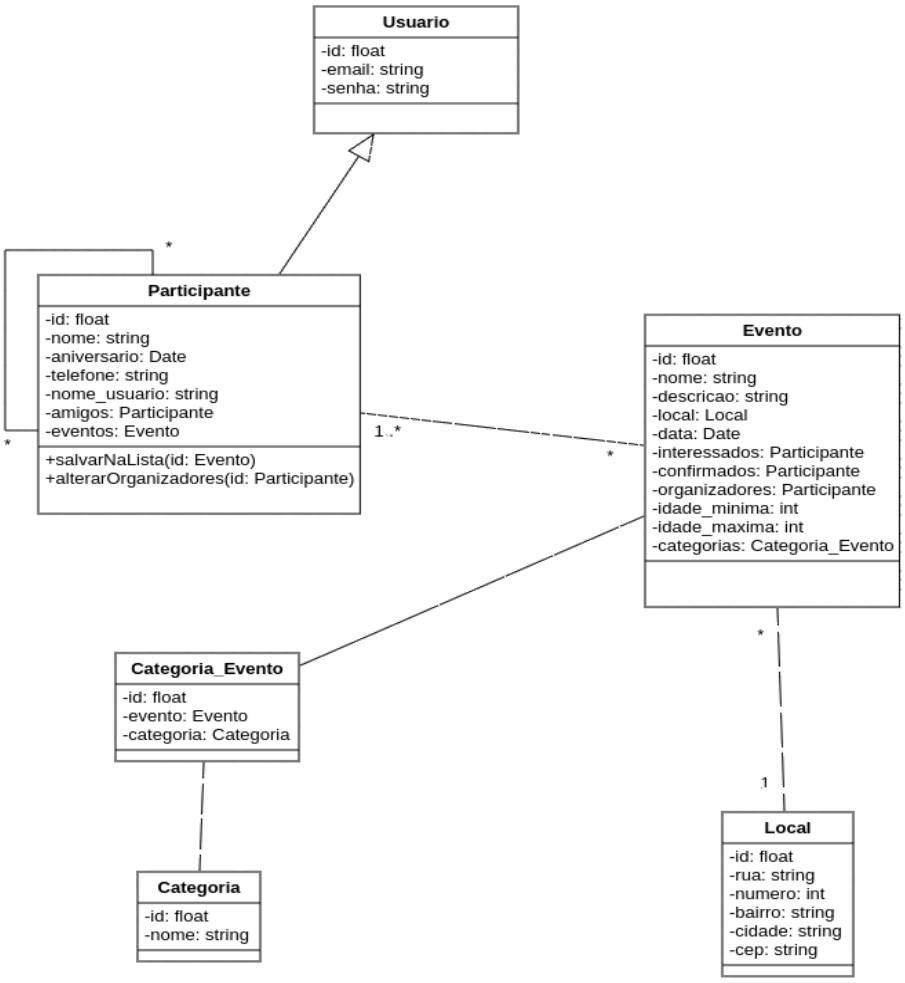


Figura 8. Modelo de domínio do projeto. [Fonte: Autores]

3.3 Recomendação Utilizada

Na situação de um cadastro novo de usuário, por conta das desvantagens apontadas nas seções 2.1 e 2.2 deste artigo, uma recomendação baseada em popularidade é gerada.

Pelo sistema aqui descrito não ter avaliações para os eventos, ou um sistema equivalente, não é passado um campo-alvo para o algoritmo usar de referência, então a recomendação é criada através do número de interações usuário-evento para cada evento. A Figura 9 ilustra a implementação desse algoritmo.

```
# Carregamento do dataset na memória
historyFrame = turicreate.SFrame.read_csv('ai/dataset.csv', delimiter=',', header=True, column_type_hints=[str, str, int])

# Treinamento do algoritmo usando o dataset
popularity_model = turicreate.popularity_recommender.create(historyFrame, user_id='user_id', item_id='event_id')

# A recomendação, por popularidade, de 5 itens é gerada para os 3 primeiros usuários da lista
popularity_recomm = popularity_model.recommend(users=[1,2,3],k=5)
popularity_recomm.print_rows(num_rows=15)
```

Figura 9. Implementação do algoritmo de recomendação por popularidade. [Fonte: Autores]

Por sua vez, quando um evento novo é cadastrado, o algoritmo de recomendação baseada em conteúdo analisa suas características, as informações apontadas na seção 3.1.4 - nome do evento, descrição, local, data, idade mínima e máxima e categoria -, de cada evento em si para calcular sua similaridade. Essa similaridade é calculada primeiramente encontrando-se a similaridade entre as características de um dado evento e depois fazendo-se a média ponderada da similaridade de cada característica - o algoritmo atribui um peso *p* para cada similaridade *s* encontrada e depois faz o somatório de todos os pesos e suas similaridades, ao final dividindo pelo somatório dos pesos - gerando o *score* de similaridade final do evento. Por fim, é feito um cálculo de média do *score* de um evento candidato a ser recomendado com o *score* dos eventos que estão presentes no histórico de um dado usuário. A Figura 10 demonstra a implementação desse algoritmo.

```
# Carregamento do dataset na memória
historyFrame = turicreate.SFrame.read_csv('ai/dataset.csv', delimiter=',', header=True, column_type_hints=[str, str, int])

# Treinamento do algoritmo usando o dataset
item_sim_model = turicreate.item_content_recommender.create(
    eventsFrame,
    item_id='event_id',
    observation_data=historyFrame,
    user_id='user_id',
    similarity_metrics='cosine'
)

# A recomendação de 5 itens é gerada para o primeiro usuário da lista
item_sim_model_recomm = item_sim_model.recommend(users=[1],k=5)
item_sim_model_recomm.print_rows(num_rows=5)
```

Figura 10. Implementação do algoritmo de recomendação baseada em conteúdo. [Fonte: Autores]

O algoritmo de recomendação baseada em usuário analisa os dados dos históricos dos usuários, quais eventos eles adicionaram às suas listas de interesse e em quais confirmaram presença, para identificar perfis de interação com eventos similares, maior número de eventos em comum, entre eles e com isso criar uma lista de itens que não estejam na intersecção desses perfis. A Figura 11 demonstra esse algoritmo.

```
# Carregamento do dataset na memória
historyFrame = turicreate.SFrame.read_csv('ai/dataset.csv', delimiter=',', header=True, column_type_hints=[str, str, int])

# Treinamento do algoritmo usando o dataset
item_user_sim_model = turicreate.item_similarity_recommender.create(
    history_train,
    user_id='user_id',
    item_id='event_id',
    similarity_type='cosine'
)

# A recomendação de 10 itens é gerada para os 5 primeiros usuários da lista
item_user_sim_model_recomm = item_user_sim_model.recommend(users=[1,2,3,4,5],k=10)
item_user_sim_model_recomm.print_rows(num_rows=50)
```

Figura 11. Implementação do algoritmo de recomendação baseada em conteúdo. [Fonte: Autores]

Por fim, para montar as recomendações que serão mostradas aos usuários, usa-se

o algoritmo de filtragem híbrida, o qual é a combinação dos dois algoritmos citados anteriormente, na abordagem mista, onde as recomendações geradas por cada algoritmo separadamente são combinados numa listagem final.

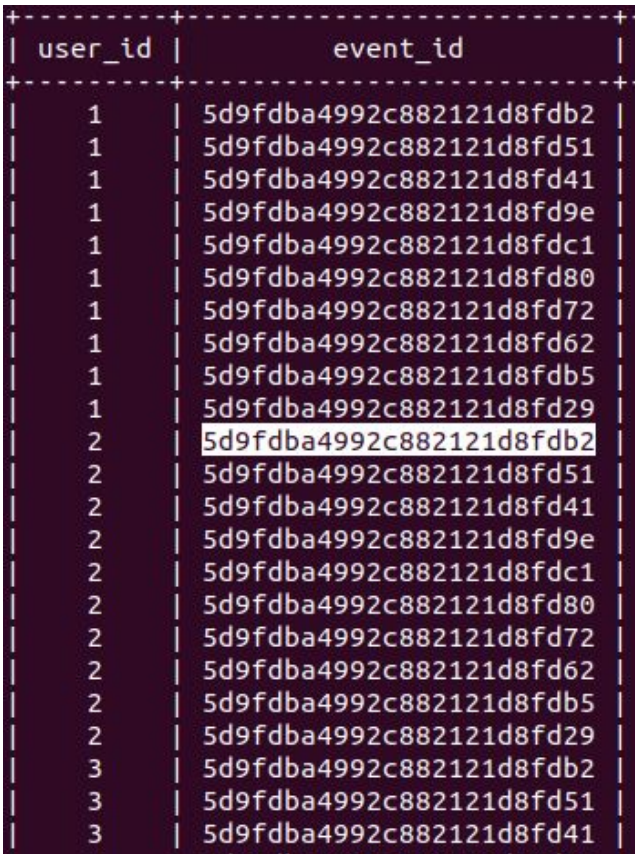
3.4 Validação

Para validar a viabilidade do algoritmo era necessário um dataset grande o suficiente que permitisse que ele fosse capaz de gerar recomendações variadas entre os usuários. Os autores não conseguiram encontrar um dataset real que atendesse as características dessa aplicação, então montaram um usando-se de geradores de dados *fake*, falsos. Por conta de não ter sido possível encontrar um dataset real, o gerado apresentou alguns problemas para a viabilidade do algoritmo. As Figuras 12, 13 e 14 ilustram os resultados obtidos pelos algoritmos quando aplicados ao dataset gerado.

user_id	event_id
1	5d9fdba4992c882121d8fa02
1	5d9fdba4992c882121d8fa01
1	5d9fdba4992c882121d8fa00
1	5d9fdba4992c882121d8f9ff
1	5d9fdba4992c882121d8f9fe
2	5d9fdba4992c882121d8fa02
2	5d9fdba4992c882121d8fa01
2	5d9fdba4992c882121d8fa00
2	5d9fdba4992c882121d8f9ff
2	5d9fdba4992c882121d8f9fe
3	5d9fdba4992c882121d8fa02
3	5d9fdba4992c882121d8fa01
3	5d9fdba4992c882121d8fa00
3	5d9fdba4992c882121d8f9ff
3	5d9fdba4992c882121d8f9fe

Figura 12. Recomendação por popularidade. [Fonte: Autores]

Como é possível notar na Figura 12, para os usuários 1, 2 e 3 foram recomendados os mesmos itens, o que era esperado.



user_id	event_id
1	5d9fdba4992c882121d8fdb2
1	5d9fdba4992c882121d8fd51
1	5d9fdba4992c882121d8fd41
1	5d9fdba4992c882121d8fd9e
1	5d9fdba4992c882121d8fdc1
1	5d9fdba4992c882121d8fd80
1	5d9fdba4992c882121d8fd72
1	5d9fdba4992c882121d8fd62
1	5d9fdba4992c882121d8fdb5
1	5d9fdba4992c882121d8fd29
2	5d9fdba4992c882121d8fdb2
2	5d9fdba4992c882121d8fd51
2	5d9fdba4992c882121d8fd41
2	5d9fdba4992c882121d8fd9e
2	5d9fdba4992c882121d8fdc1
2	5d9fdba4992c882121d8fd80
2	5d9fdba4992c882121d8fd72
2	5d9fdba4992c882121d8fd62
2	5d9fdba4992c882121d8fdb5
2	5d9fdba4992c882121d8fd29
3	5d9fdba4992c882121d8fdb2
3	5d9fdba4992c882121d8fd51
3	5d9fdba4992c882121d8fd41

Figura 13. Recomendação pela filtragem baseada em usuário. [Fonte: Autores]

Nesse caso, Figura 13, é percebido que os itens recomendados são diferentes daqueles da recomendação por popularidade, embora sejam os mesmos entre os usuários. O motivo disso será descrito na seção 4 deste artigo.

user_id	event_id
1	5d9fdba4992c882121d8fa02
1	5d9fdba4992c882121d8fa01
1	5d9fdba4992c882121d8fa00
1	5d9fdba4992c882121d8f9ff
1	5d9fdba4992c882121d8f9fe

Figura 14. Recomendação pela filtragem baseada em conteúdo. [Fonte: Autores]

Aqui, ocorreu-se o contrário da recomendação mencionada anteriormente, na Figura 13. Os itens dessa recomendação são diferentes as daquela, porém acabaram sendo os mesmos da recomendação por popularidade. A causa disso é a mesma do resultado da recomendação baseada em usuário, a qual será explicada na seção 4.

4. Conclusões

A aplicação apresentada nesse artigo é um sistema de recomendação de eventos que

analisa o histórico dos usuários bem como as características dos eventos, através do uso do algoritmo de filtragem híbrida, o qual é a junção dos resultados obtidos pelos algoritmos de filtragem baseada em usuário e filtragem baseada em conteúdo.

Os algoritmos de recomendação mostraram-se capazes de criar recomendações para os usuários, embora não tenham sido capazes de variar tanto as recomendações para cada usuário. Isso acontece devido ao dataset usado, como não foi encontrado um conjunto de dados real que estivesse de acordo com a realidade da aplicação deste artigo, foi-se necessário criar um do zero, o que ocasionou ao banco de dados conter um conjunto de informações muito parecidas já que os autores não conseguiram encontrar uma biblioteca de geração de dados capaz de criar conjuntos muito distintos.

Como passos futuros para essa aplicação, podem ser listadas:

- análise do funcionamento do algoritmo com um dataset mais próximo da realidade a qual o sistema será submetido quando estiver em funcionamento;
- implementação de um mecanismo de busca para encontrar eventos específicos mais facilmente dentro da plataforma e para encontrar outros usuários;
- criação de uma página de detalhe do evento, que contenha todas as informações pertinentes;
- criação de uma interface mobile, para um acesso mais fácil através de celulares; e
- envio de avisos através de email ou notificações *push*, tanto para informar de algum evento novo que pode ser do interesse do usuário quanto para informar que certo evento está para começar.

5. Bibliografia

Apache Spark: Como criar um mecanismo de sugestão de produtos. Disponível em: <https://www.devmedia.com.br/apache-spark-como-criar-um-mecanismo-de-sugestao-de-produtos/33459>. Acesso em: 2 nov. 2019.

ARTHUR FORTES. **Sistemas de Recomendação Aplicados a Web**. Disponível em: <https://pt.slideshare.net/fortesarthur/sistemas-de-recomendao-aplicados-a-webcomo-converter-visitantes-em-compradores-fatec-lins-2014>. Acesso em: 2 nov. 2019

BARBOSA, C. E. M. **Estudo de Técnicas de Filtragem Híbrida em Sistemas de Recomendação de Produtos**. p. 99, 2014.

Comprehensive Guide to build Recommendation Engine from scratch. Disponível em: <https://www.analyticsvidhya.com/blog/2018/06/comprehensive-guide-recommendation-engine-python/>. Acesso em: 15 out. 2019.

Engineering and Service Science (ICSESS). **Anais**... In: 2018 IEEE 9TH INTERNATIONAL CONFERENCE ON SOFTWARE ENGINEERING AND SERVICE SCIENCE (ICSESS). nov. 2018

Entenda como funcionam os Sistemas de Recomendação. **IGTI Blog**, 3 nov. 2017. Disponível em: <http://igti.com.br/blog/como-funcionam-os-sistemas-de-recomendacao/>. Acesso em: 15 out. 2019

FREITAS, D. W. **RECOMENDAÇÃO DE ANIMES UTILIZANDO MACHINE LEARNING, UMA ABORDAGEM BASEADA EM AVALIAÇÕES DOS USUÁRIOS**. p. 20, [s.d.].

LÁZARO, Alexsandra da Silva. **Análise e seleção de Algoritmos de filtragem de informação para solução do problema *Cold-start Item***. Universidade Federal de Lavras. 2010, MG.

PARIKH, V. et al. **A Tourist Place Recommendation and Recognition System**. 2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT). **Anais**... In: 2018 SECOND INTERNATIONAL CONFERENCE ON INVENTIVE COMMUNICATION AND COMPUTATIONAL TECHNOLOGIES (ICICCT). abr. 2018

SINEVA, I. S.; DENISOV, V. Y.; GALINOVA, V. D. **Building Recommender System for Media with High Content Update Rate**. 2018 IEEE International Conference "Quality Management, Transport and Information Security, Information Technologies" (IT QM IS). **Anais**... In: 2018 IEEE INTERNATIONAL CONFERENCE "QUALITY MANAGEMENT, TRANSPORT AND INFORMATION SECURITY, INFORMATION TECHNOLOGIES" (IT QM IS). set. 2018

Sistemas de recomendação com filtros colaborativos. **MOVILE**, 2 maio 2019. Disponível em: <https://movile.blog/sistemas-de-recomendacao-com-filtros-colaborativos/>. Acesso em: 2 nov. 2019

WU, C. M.; GARG, D.; BHANDARY, U. **Movie Recommendation System Using Collaborative Filtering**. 2018 IEEE 9th International Conference on Software

Turi Create API Documentation — Turi Create API 5.8 documentation. Disponível em: <https://apple.github.io/turicreate/docs/api/index.html>. Acesso em: 25 nov. 2019.