UNIVERSIDADE FEDERAL FLUMINENSE
CAMPUS DE RIO DAS OSTRAS
INSTITUTO DE CIÊNCIA E TECNOLOGIA
BACHARELADO EM CIÊNCIA DA COMPUTAÇÃO

LUCAS CORDEIRO DA SILVA

Sistema de recomendação de filmes usando a técnica de filtragem colaborativa baseada no usuário

RIO DAS OSTRAS, RJ
2017

LUCAS CORDEIRO DA SILVA

Sistema de recomendação de filmes usando a técnica de filtragem colaborativa baseada no usuário

Monografia apresentada ao Curso de Bacharelado em Ciência da Computação do Instituto de Ciência e Tecnologia da Universidade Federal Fluminense, como requisito parcial para obtenção do Grau de Bacharel.

Orientadora:
Prof.ª Leila Weitzel Coellho Da Silva

Rio das Ostras, RJ

2018
Lucas Cordeiro da Silva

Sistema de recomendação de filmes usando a técnica de filtragem colaborativa baseada no usuário

Monografia apresentada ao Curso de Bacharelado em Ciência da Computação do Instituto de Ciência e Tecnologia da Universidade Federal Fluminense, como requisito parcial para obtenção do Grau de Bacharel.

Aprovado em 12 de julho de 2018.

BANCA EXAMINADORA

Prof.ª Leila Weitzel Coellho Da Silva - UFF
Orientadora

Prof. Carlos Bazilio Martins - UFF

Prof.ª Marcilene De Fatima Dianin Vianna - UFF

Rio das Ostras, RJ
2018

1 AGRADECIMENTOS

Agradeço a Deus pela sabedoria dada nesses anos de batalhas.

Aos meus pais, Waldira Cordeiro e Epitacio Cordeiro, por toda confiança, amor e compreensão durante esses anos. Estiveram sempre no meu time, torcendo por mim, sempre na primeira arquibancada, gritando palavras de coragem, fé e perseverança.

À minha namorada Maria Clara, por todo apoio dado. Cada palavra de incentivo e colaboração fomentava uma centelha para criatividade e enxergar o mundo de outra forma.

Aos meus avós pela confiança em mim, sempre acreditaram que tudo fosse dar certo. O amor de vocês me deu forças para lutar.

Aos meus amigos, do grupo MULEKADINHA, pela ajuda durante toda a graduação.

Agradeço também a todos os professores do curso de Ciência da Computação da Universidade Federal Fluminense de Rio das Ostras por terem ajudado neste trabalho, seja de forma direta ou indireta.

E por fim, agradeço, a minha orientadora Prof.ª Leila Weitzel pelo apoio e confiança durante o desenvolvimento deste trabalho.

2 RESUMO

A world Wide Web trouxe inúmeras mudanças como vivemos e nos comunicamos. Algumas mudanças foram, a utilização de plataformas de vídeo e músicas sob demanda e utilização de comercio eletrônico. No entanto, essa mudança vem com o custo da sobrecarga de informação, mecanismos de busca podem não recuperar informações relevantes para o usuário. Para lidar com esse tipo de problema, um sistema de recomendação funciona como um assistente, que irá encontrar informações e fazer recomendações personalizadas e relevantes para cada usuário. Para realizar recomendações personalizadas, os SR utilizam técnicas de filtragens, onde é realizado um processo de heurística com as informações dos usuários e da informação buscada. Este trabalho propõe avaliar a performance dos algoritmos KNN e K-Means em um sistema de recomendação de filmes. Será utilizado um dataset de filmes fornecido pela GroupLens para simular um problema real na área de recomendação de filmes.

Palavras-chave:
Sistema de Recomendação, MovieLens, KNN, K-Means

3 ABSTRACT

The World Wide Web brought many changes as we live and communicate. Some changes were, the use of video platforms and music on demand and use of e-commerce. However, this change comes with the cost of information overload, search engines may not retrieve relevant information for the user. To deal with this type of problem, a recommendation system functions as a wizard, which will find information and make recommendations that are personalized and relevant to each user. To make personalized recommendations, RS use filtering techniques, where a heuristic process is performed with the information of the users and the information sought. This work proposes to evaluate the performance of the KNN and K-Means algorithms on a movie recommender system. A movie dataset provided by GroupLens will be used to simulate a real problem in the movie recommendation area.

Keywords:
Recommender System, MovieLens, KNN, K-Means

LISTA DE ABREVIATURAS

SR – Sistema de Recomendação

FBC – Filtragem Baseada em Conteúdo

FCBI – Filtragem Colaborativa Baseada em Item

FCBU – Filtragem Colaborativa Baseada em Usuário

IMDB – Internet Movie Database

URL – Uniform Resource Locator

MAE – Mean Absolute Error

NMAE – Normalized Mean Absolute Error

RMSE – Root Mean Squared Error

MSE – Mean Squared Error

KNN – K - Nearest Neighbors

LISTA DE FIGURAS

Lista de Tabelas

Sumário

4 INTRODUÇÃO

4.1 Motivação e Justificativa

A crescente diversidade de informações disponíveis nos serviços de e-bussines na última década impulsionou a criação de sistemas inteligentes que pudessem aprender sobre seus usuários com a finalidade de aumentar o engajamento desse com a plataforma na qual está presente.

Na vida cotidiana é necessário fazer escolhas sem experiência pessoal sobre todas as alternativas envolvidas. Dessa forma, contamos com a recomendação de outras pessoas para nos apoiar na tomada de decisão. Os sistemas de recomendação auxiliam nesse processo de indicação que é presente na relação social entre seres humanos (RESNICK e VARIAN, 1997).

Os SR surgem a partir da necessidade de filtrar conteúdo para o usuário dentre todas as opções disponíveis e automatização da geração de recomendação baseada na análise de dados (MELVILLE e SINDHWANI, 2010).

Em plataformas que utilizam sistemas de Recomendação, o cliente é visto como um consumidor que é parte fundamental no processo de desenvolvimento do serviço. A observação de sua interação com um serviço on-line – incluindo sites de relacionamentos sociais, plataformas de streaming de vídeos e música – ajuda a desenvolver insights de seus comportamentos que levam ao melhor entendimento sobre o usuário. A busca por metodologias e design de interação centrado no usuário inferem em ver o serviço da perspectiva do cliente. Segundo (CHANG, 2015), existe uma preocupação substancial em resolver problemas na área da recomendação, com estudos que geralmente se concentram em soluções algorítmicas que maximizam as informações obtidas sobre os usuários a partir da classificação de itens. Entretanto, otimizar ainda mais a seleção de itens para classificação continue sendo um problema em aberto, é postulado que avanços maiores podem ser possíveis repensando o modelo de interação do usuário com o sistema.

O cinema está se tornando um entretenimento cada vez mais importante na vida das pessoas. Com isso, a sobrecarga de informação se torna um problema real nesta

área, e novas técnicas de recomendação de filmes surgem. Conforme o número de usuários e filmes crescem em qualquer plataforma, a busca por tentar sugerir filmes relevantes é de grande importância. Dentre uma variedade de algoritmos de recomendação, técnicas e estratégias, os cientistas de dados precisam escolher a melhor combinação de acordo com as limitações existentes.

Existem dezenas de mecanismos de recomendação de filmes na web, alguns exigem pouca ou nenhuma informação para sugerir algum título, outros querem descobrir exatamente quais são os interesses reais de seus usuários.

4.2 Objetivo

O objetivo desta pesquisa é avaliar a performance dos algoritmos KNN e K-Means em um sistema de recomendação de filmes.

5 SISTEMAS DE RECOMENDAÇÃO

Os Sistemas de Recomendação são um conjunto de ferramentas e técnicas que tem o objetivo de gerar recomendações relevantes para os usuários. As sugestões fornecidas visam apoiar o usuário na tomada de decisão, como quais itens comprar, qual filme assistir, qual musica escutar ou qual livro ler, por exemplo. Segundo (RICCI, et al., p. VII, 2011) os SR provaram ser valiosos para resolver problemas de sobrecarga de informação e tornaram-se ferramentas poderosas e populares no comércio eletrônico.

Existem vários motivos para a utilização de sistemas de recomendação:

- **Aumentar o número de itens vendidos**: Recomendar itens com um grau de certeza que o usuário vai adquirir pode implicar numa maior probabilidade de venda desse item.

- **Vender mais itens diversificados**: Sistemas podem selecionar itens que podem ser difíceis de serem encontrados, como itens impopulares, resolvendo o problema de *LongTail*.

- **Aumentar a satisfação do cliente**: O usuário encontrará produtos interessantes e relevantes o que implicará na satisfação do cliente com a plataforma.

- **Aumentar a quantidade de assinantes**: Com o maior engajamento e fidelidade dos usuários, novos clientes aparecerão.

A elaboração de um sistema de recomendação envolve equipes multidisciplinares:

> “O desenvolvimento de sistemas de recomendação é um esforço multidisciplinar que envolve especialistas de várias áreas, como Inteligência Artificial, Interação Humano-Computador, Tecnologia da Informação, Mineração de Dados, Estatística, Interfaces de

> Usuários Adaptáveis, Sistemas de suporte a decisão, marketing ou Comportamento do consumidor.”
>
> (RICCI, et al., p.45, 2011)

Um motor de busca é feito para auxiliar o usuário na procura de informações dentre várias disponíveis. O’Brien, escritor da revista Fortune, nos ajuda a entender a diferença entre buscar uma informação e receber uma informação recomendada:

> “A web, dizem eles, está deixando a era das buscas e entrando na descoberta. Qual é a diferença? A pesquisa é o que você faz quando procura algo. Descoberta é quando algo maravilhoso que você não sabia que existia, ou não sabia como pedir, encontra você.”
>
> (Jeffrey M. O’ Brian, 2006)

Quando O’Brien escreveu essa frase ele se referia aos sistemas de recomendação. Para um mecanismo de busca ser útil, o usuário teria de ter uma ideia do que está procurando e fazer uma consulta, em um SR, o usuário não teria em mente o que de fato ele quer, mas é surpreendido por recomendações relevantes baseadas nele.

O design de cada *engine* de recomendação depende do domínio e de características particulares dos dados disponíveis. Por exemplo, usuários do Spotify fornecem hearts(gostar) para cada música da plataforma. Esse atributo explícito registra a qualidade da música pela quantidade de usuários que a escutaram. Além disso o sistema pode ter acesso a dados de perfil do usuário, como informações demográficas, idade e preferência por gênero musical. Os SR diferem na maneira como analisam essas fontes de dados para analisar a similaridade entre usuários e itens.

(RICCI, et al., 2015) define classes de domínios para os aplicativos de sistemas de recomendação mais comuns:

- **Conteúdo**: Recomendação de filmes, músicas e jogos.

- **E-Comerce**: Recomendação de produtos para compra, como livros, câmeras, computadores etc., para consumidores

- **Serviços**: Recomendação de serviços de viagem, especialistas para consulta, casas para alugar.

- **Social**: Recomendação de pessoas em redes sociais e recomendação de conteúdo de mídia social, como tweets, feeds do facebook, atualizações do LinkedIn.

Pelo fato de boas e inesperadas recomendações impulsionarem os números de inscritos em serviços online, grandes empresas ao redor do mundo embarcam em uma corrida multibilionária para o investimento na área.

5.1 Técnicas de Recomendação

Diferentes Técnicas são utilizadas na construção de sistemas de Recomendação conforme o objetivo a ser atingido e o tipo dos dados envolvidos.

O núcleo de um SR tem como objetivo identificar itens uteis para o usuário, ou seja, deve prever um item que vale a pena recomendar. A ausência de informações precisas sobre as preferências dos usuários pode implicar que o núcleo recomende itens que podem não ser relevantes, mas que sejam populares.

Para fornecer uma taxonomia para os diferentes tipos de SR, (BURKE, 2002) apresenta um clássico modelo que distingue em seis classes, que são:

- Colaborativa
- Baseada em Conteúdo
- Demográfica
- Baseada em Utilidade
- Baseada no conhecimento

Dentre essas, será abordado nesta seção apenas as duas primeiras – que são representadas pela Figura (1) – que foram utilizadas na implementação do sistema exemplo deste trabalho, além de serem as mais utilizadas.

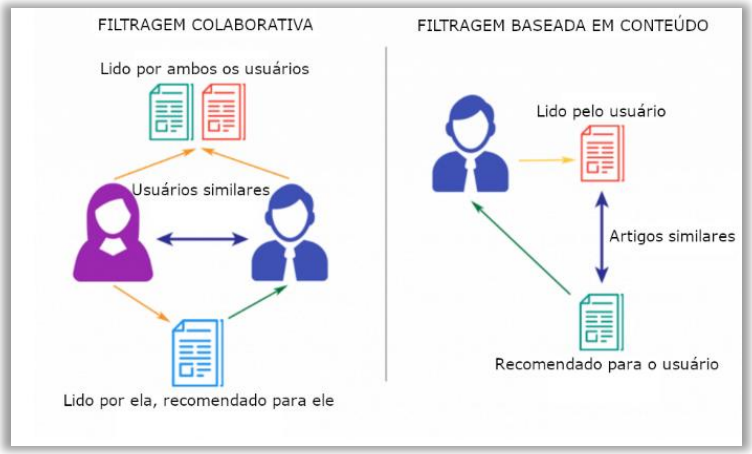

Figura 1 Exemplo de Filtragem Colaborativa e Filtragem Baseada em Conteúdo

Fonte: http://blog.avenuecode.com/how-to-build-a-recommender-system-in-less-than-1-hour, 2017

A interpretação da Figura (1) pode ser feita da seguinte forma, em FC (lado esquerdo), artigos de ficção cientifica são lidos por dois usuários (roxo e azul presentes na imagem), com isso, o SR entende que os usuários são similares. O usuário roxo leu um outro artigo sobre um assunto diferente, e o SR recomenda esse artigo ao usuário azul, pois os dois tem gostos similares. Em FBC (lado direito), o usuário azul gosta de ler artigos sobre ficção científica, o SR entende que artigos similares devem ser recomendados, pois o usuário se interessa por determinado assunto.

5.1.1 Filtragem baseada em conteúdo

As recomendações são baseadas na correlação do conteúdo dos itens e as preferências do usuário ao invés da opinião dos usuários em relação aos itens. Os itens podem conter como atributos, por exemplo, no caso de filmes, gênero, diretor, ano, faixa etária, título, descrição, etc.

Se um usuário assiste mais filmes de terror do que qualquer outro gênero, é interessante que um SR recomende lançamentos do mesmo gênero, pois terror é de maior preferência do usuário. Um outro exemplo poderia ser, o usuário que compra vários livros de um mesmo autor provavelmente gosta desse autor, que implica em gostar de outros livros escritos pelo mesmo autor.

O sistema aprende a recomendar itens baseados no perfil do usuário, que pode ser computado por itens que o usuário gostou no passado ou atributos pré-definidos em seu perfil, como preferência por gênero, faixa etária, etc. A similaridade é gerada com base no feedback implícito e explícito do usuário.

5.1.2 Filtragem Colaborativa

Uma outra abordagem é a Filtragem Colaborativa, que segundo (BURKE, 2002), é provavelmente a técnica mais familiar, e a mais amplamente implementada e mais madura das tecnologias. De acordo com (HERLOCKER, 2002) esta abordagem foi desenvolvida para atender pontos que estavam em aberto na filtragem baseada em conteúdo. A FC se diferencia da FBC em não exigir a compreensão ou reconhecimento do conteúdo dos itens.

Um exemplo de ambiente baseado em FC é o sistema de recomendação de filmes MovieLens. Nele, o usuário avalia filmes que tenha assistido e o sistema utiliza essas avaliações para encontrar pessoas com gostos parecidos. No tópico (10 Trabalhos Relacionados) será explicado o funcionamento da plataforma do MovieLens com mais detalhes.

Existem duas maneiras básicas de filtrar informações para os usuários, usando a filtragem colaborativa baseada em item (FCBI) e a filtragem colaborativa baseada no usuário (FCBU). Para (SAWAR, et al., 2001), a filtragem baseada em item analisa a base de itens para identificar relacionamentos entre diferentes itens e então usam esses relacionamentos para calcular recomendações para os usuários. E a filtragem baseada no usuário, as recomendações são geradas com base nas avaliações dos usuários mais semelhantes. A diferença está em computar a similaridade entre itens ao invés de

usuários. A continuação deste tópico abordará a FCBU para explicar a filtragem colaborativa.

A abordagem baseada no usuário procura predizer itens para um consumidor em particular, baseado em como os outros usuários com gostos similares – que são calculados através de técnicas estatísticas – ao usuário alvo previamente se interessaram pelo mesmo item. A ideia principal é que usuários semelhantes atribuam um feedback semelhante aos mesmos itens. A diferença entre a variedade de sistemas que utilizam esta abordagem está em como a similaridade é calculada. As recomendações colaborativas são baseadas na qualidade dos itens avaliados pelos pares, em vez de depender de conteúdo que pode ser um mau indicador de qualidade.

De acordo com (HERLOCK, 2002), esta técnica pode ser descrita em 3 passos:

Passo 1: Calcular os pesos: Calcular o peso de cada usuário do sistema em relação à similaridade com o usuário alvo utilizando uma métrica de similaridade.

Passo 2: Buscar os vizinhos: Selecionar um subconjunto de usuários com maiores similaridades para considerar na predição.

Passo 3: Realizar a predição: Normalizar as avaliações fornecidas pelos usuários e computar as predições ponderando as avaliações dos vizinhos com seus respectivos pesos.

	Titanic	Batman	Vingadores	Rocky
Lucas	5	2	5	4
Tiago	2	3	-	2
Bruno	1	1	2	2
Pedro	5	1	4	**(?)**

Tabela 1 usuario-filme com avaliaçoes dos usuários

Continuando o raciocínio de (HERLOCK, 2002), a Filtragem Colaborativa pode ser representada como o problema de prever valores não encontrados em uma matriz usuário-Item. A Tabela (1) é um exemplo de usuário-filme onde cada célula representa a

avaliação que um usuário deu a um filme. O motor de predição deverá prever qual nota Pedro daria ao filme “Rocky”. Lucas é o vizinho mais próximo do Pedro, porque os dois avaliaram os filmes de forma parecida. Para o resultado, a opinião de Lucas sobre o filme “Rocky”, é a que terá mais influência na predição da nota. Tiago e Bruno não são vizinhos próximos de Pedro, pois eles discordam de Pedro na maioria dos filmes. E para o resultado eles terão menos influencia que o Lucas.

Esta abordagem pode apresentar problemas como, *Cold-Start*, que é a falta de avaliações de usuários com os filmes, com isso a dificuldade em estimar a similaridade entre os usuários baseando-se em suas colaborações explicitas, implicam na utilização de técnicas que procuram inferir as preferências dos usuários através de seus atributos implícitos.

A medida de similaridade em um SR baseado em filtragem colaborativa procura encontrar itens ou usuários para a predição, dessa forma, técnicas de *machine learning* podem ser utilizadas para melhorar a qualidade da filtragem. Será abordado duas técnicas, as mais utilizadas na literatura, KNN e K-MEANS.

Em um caso pratico de sistema de recomendação baseado em filmes, e com a implementação das técnicas de ML citadas acima, a recomendação é feita a partir da descoberta dos k vizinhos mais próximos no caso do KNN ou dos vizinhos pertencentes ao mesmo cluster do usuário no caso do K-MEANS. Após executar os passos e encontrar os vizinhos, significa, encontrar usuários parecidos, que tem gostos similares em relação aos filmes. Com isso, e sabendo quais filmes o usuário não assistiu, é possível descobrir quais são os filmes que o usuário alvo gostaria de assistir.

A FC facilita a recomendação de itens com características diferentes das preferências do usuário, podendo fornecer recomendações inesperadas.

5.1.2.1 K-Nearest Neighbor

O algoritmo K-Nearest Neighbor, algoritmo de classificação de vizinhos mais próximos, é um dos métodos mais simples e populares de aprendizagem de máquina.

Este método tem sido usado em diversas aplicações no campo de data-mining, métodos estatísticos, reconhecimento de padrões e processamento de imagens.

O algoritmo requer três dados como entrada:

- O conjunto de instancias (amostras)
- Métrica de distância para calcular a distância entre os vizinhos (amostras)
- O valor de K, o número de vizinhos mais próximos a serem recuperados

Em geral, para classificar um item, será executado dois passos:

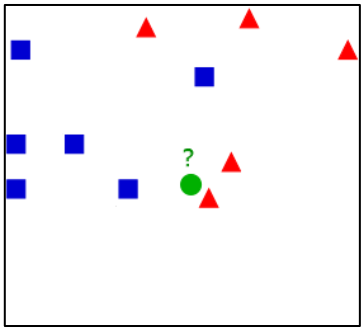

Figura 2 Conjunto de N Itens

Passo 1: Computar a distância entre um item alvo até outros itens do treinamento
Passo 2: Identificar os k vizinhos mais próximos do item alvo

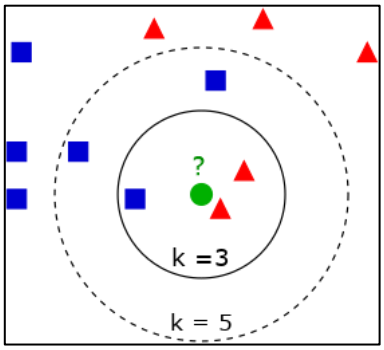

Figura 3 Exemplo de classificação KNN

Na Figura (3), a amostra de teste é o círculo verde, que deve ser classificada. Se k=3 compreendido pelo círculo de linha continua, o objeto verde é mais semelhante ao

triangulo do que ao quadrado, porque há dois triângulos e apenas um quadrado dentro do círculo interno.

5.1.2.2 K-Means

O algoritmo K-Means é um método de clusterização que particiona N itens em K subconjuntos disjuntos, onde cada subgrupo é compreendido por objetos similares fornecidos pela medida de distância. Esta técnica pode ser empregada em diversos problemas:

- **Marketing**: Identificar grupos distintos de clientes.
- **Geografia**: Identificar áreas usadas com o mesmo proposito com observações da terra.
- **Seguro**: Identificar grupos de clientes que fazem comunicação de sinistro com alta frequência.
- **Planejamento**: Identificar grupos de casa de acordo com o tipo, valor e localização geográfica.
- **Saúde**: Agrupamento de pacientes com os mesmos sintomas.
- **Filmes**: Identificar grupos de usuários que assistem a determinados gêneros ou filmes.

Cada cluster gerado é definido por seus membros e por seu centroide**.** O algoritmo requer três dados como entrada:

- O conjunto de instancias (amostras)
- Métrica de distância para calcular a distância entre os vizinhos (amostras)
- O valor de K que é o número de centroides ou clusters a serem formados

Em geral, para encontrar os clusters, (OLIVEIRA, 2002) define alguns passos que o algoritmo executa:

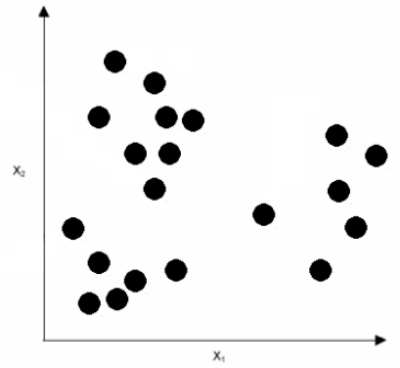

Figura 4 Conjunto de N itens, algoritmo ainda não executado

Passo 1: Selecionar arbitrariamente K itens para serem as sementes (centroides) dos clusters iniciais.

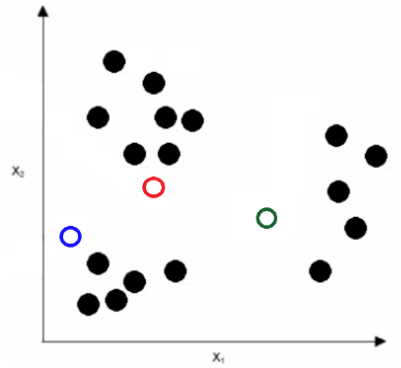

Figura 5 K itens selecionados para servirem como sementes

Passo 2: Associar cada item ao Centro do Cluster (centroide) mais próximo (maior similaridade) formando grupos.

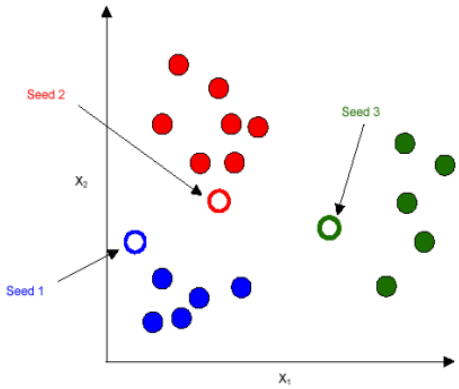

Figura 6 formação inicial de grupos

Passo 3: Calcular os novos centroides a partir de todos os elementos existentes em cada cluster.

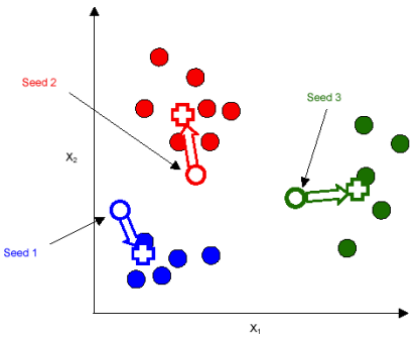

Figura 7 Recálculo dos centroides

Passo 4: Associar cada registro aos novos Centroides.

Passo 5: Repita a partir do passo 2 até que não ocorra mais mudanças nos itens que compõe o cluster.

Segundo RICCI (2010), a maior parte da convergência ocorre durante as primeiras iterações do algoritmo e, portanto, a condição de parada pode ser alterada para "*até relativamente poucos pontos mudarem de cluster*" para melhorar a eficiência do algoritmo. É desejável que as distancias entre os vetores(objetos) dentro de um grupo(cluster) sejam pequenas e que as distancias entre diferentes clusters sejam grandes.

5.1.2.3 Determinação dos Vizinhos

Uma forma para identificar os vizinhos mais próximos de um usuário alvo é utilizando uma métrica n-dimensional de distância.

Existem várias formas de calcular a distância para a encontrar os vizinhos. Algumas funções que podem ser utilizadas são distância Euclidiana e distância Manhattan. A escolha da função depende de muitos fatores, incluindo as características dos dados. O exemplo mais simples, mais comum e que é abordado no protótipo para o cálculo da distância, é distancia Euclidiana demonstrada pela seguinte equação:

$$d(u,v) = \sqrt[2]{\sum_{i=1}^{n}(r_{ui} - r_{vi})^2} \quad (1)$$

Onde $d(u,v)$ representa a distância entre dois pontos, e n o número de dimensões e $u = (r_{u1}, r_{u2}, \dots, r_{un})$ e $v = (r_{v1}, r_{v2}, \dots, r_{vn})$ são pontos no espaço n-dimensional. A equação é entendida como a raiz quadrada da soma dos quadrados das diferenças das coordenadas.

A medida de distância mais fácil de calcular é a distância Manhattan, representada a seguir.

$$d(u,v) = \sum_{i=1}^{n} |r_{ui} - r_{vi}| \quad (2)$$

A Equação (2) apresenta a mesma interpretação para as variáveis da Equação (1) para encontrar a distância entre os pontos u e v no espaço n-dimensional. A equação pode ser entendida como a soma dos valores absolutos das diferenças das coordenadas.

Em algumas implementações de sistemas de recomendação, a similaridade é definida como:

$$sim(u,v) = \frac{1}{1 + d(u,v)} \quad (3)$$

Onde $sim(u, v)$ representa a similaridade entre dois usuários e $d(u, v)$ representa a distância entre eles**.**

5.1.2.4 Determinação da Predição e Recomendação

O objetivo de um algoritmo de filtragem colaborativa é prever a utilidade de um item para o usuário ativo. Itens uteis podem ser definidos da seguinte forma: Seja A o conjunto de itens avaliados pelos usuários similares e B o conjunto de itens avaliados pelo usuário ativo, o conjunto de itens uteis é A - B. Segundo (SAWAR, et Al., 2001) A tarefa da FC pode ser dividida em duas partes:

- **Predição**: É um valor numérico que expressa a probabilidade de avaliação de um item para o usuário ativo.

- **Recomendação**: É uma lista de N itens que o usuário gosta mais. A recomendação também é conhecida como os Top-N itens.

Após a etapa que determina quais os vizinhos mais próximos e como definir a similaridade entre eles apresentado pelo tópico anterior, a determinação da predição encontrará o melhor valor para um item. O valor pode ser calculado através da média aritmética dos valores das avaliações dos vizinhos. Outro artifício matemático que pode ser utilizado é o conceito de moda para predição.

A média é uma medida que calcula o valor aritmético médio de um conjunto. É somado todos os valores e depois dividido pela quantidade de ocorrências.

A moda é um valor que surge com mais frequência nos dados discretos. Caso o conjunto retorne duas ou mais notas com o mesmo número de ocorrências, é feita uma escolha aleatória para predição. E caso não haja retorno também é feito uma escolha aleatória.

Segundo (RICCI, et Al., p103, 2011) o problema da média aritmética simples é que não é levado em conta o fato de que os vizinhos podem ter diferentes níveis de

similaridade. Uma solução comum para este problema é pesar a contribuição de cada vizinho pela sua semelhança com o usuário alvo. Dessa forma, a equação para calcular uma estimativa de classificação em um sistema baseado no usuário é fornecida da seguinte forma:

$$\hat{r}_{ui} = \frac{\sum_{v \in V} sim(u,v) x\, r_{vi}}{\sum_{v \in V} |sim(u,v)|} \quad (4)$$

Onde V representa os vizinhos, u o usuário alvo, i o item a ser predito, r_{vi} a nota que o vizinho deu para o item e $sim(u,v)$ a similaridade entre o usuário u e v. A Equação (4) prevê a nota para um item do conjunto de itens uteis para o usuário alvo.

Uma outra maneira de calcular a estimativa de classificação em um sistema baseado no usuário com uma abordagem diferente é apresentada por (MAHDAVI, MORADI, 2011). A abordagem consistia em realizar uma votação, onde era analisado as notas de cada usuário para os filmes a fim de encontrar quais eram as notas mais populares. Mahdavi e Moradi também apresentaram uma função para introduzir um refinamento nas suas predições, que consistia em atribuir um peso as notas fornecidas por usuários mais próximos. A equação utilizada em sua pesquisa é apresentada a seguir:

$$\hat{r}_{ui} = argmax_{reR} \left(\sum_{v \in Neighbors} W_v\, \partial(r, f_{v,i}) \right) (5), \partial\left(r, f_{v,i}\right) = \left\{ \begin{matrix} 1 & if(r = f_{v,i}) \\ 0 & Otherwise \end{matrix} \right\} (6)$$

$$W_v = \frac{1}{d(u,v)^2} \quad (7)$$

A Equação (5) encontrará a avaliação pertencente ao conjunto R = {1, 2, 3, 4, 5} que tem maior frequência dada pelos vizinhos ao item i. A Equação (6) identifica se a

nota $f_{n,i}$ dada pelo usuário ao item i é igual a nota r, se for verdade é retornado o valor 1, e 0 para falso, e então é feito o incremento pela função superior. W_n expressa o peso que o usuário terá em sua sugestão. As predições são obtidas após ser executado $\hat{r}_{ui}$ para cada item pertencente ao conjunto de itens uteis para o usuário ativo.

Após realizar as predições das notas para os filmes desejados, é feito a recomendação para o usuário. Para gerar recomendação é feito um ordenamento dos valores obtidos na predição – não perdendo a referência do item – e os Top N itens com as maiores notas serão recomendados.

5.1.3 Filtragem Híbrida

Na filtragem hibrida, dois ou mais métodos de filtragem são combinados para obter melhor desempenho. De acordo com (BURKE, 2002) a combinação de diferentes métodos de filtragem resulta em desempenho máximo e alivia problemas enfrentados por diferentes técnicas de filtragem quando usadas separadamente. Alguns pontos fortes que podem ser extraídos são:

- **Filtragem Colaborativa**: Descoberta de novos relacionamentos entre usuários e recomendação de itens relacionados ao histórico.

- **Filtragem Baseada em Conteúdo**: Precisão independentemente do número de usuários.

Existem várias técnicas de hibridização que podem abordar a FC e FBC para obter um melhor desempenho e que podem gerar diferentes saídas.

- Implementar FC e FBC separadamente e combinar suas previsões.
- Incorporar características baseadas em conteúdo na FC.
- Incorporar características colaborativas na FBC.

5.2 Formação do perfil do usuário

Um dos pontos fundamentais abordados em SR, está na definição do perfil dos usuários do sistema. Um SR não inicia suas atividades sem que o perfil do usuário seja definido.

Os SR trabalham com informações que permitem a verificação de feedback do usuário em relação aos itens. Por exemplo, o feedback positivo significa que houve interesse do usuário em relação ao item adquirido e o feedback negativo que não houve interesse.

Existem duas maneiras de obter feedback do usuário, na observação implícita da interação do usuário com o sistema ou no uso de informação explicita fornecida pelo usuário.

O feedback explícito requer que o usuário avalie os filmes numa escala que pode classificar um filme sendo bom ou ruim, de forma binaria gostar/não gostar ou através de comentários via texto não estruturado.

As avaliações implícitas incluem histórico de compra, histórico de navegação, padrões de busca, movimentação do cursor do mouse, etc. Em situações práticas, as avaliações implícitas são utilizadas quando o sistema não consegue coletar ou ainda não recebeu o feedback explícito – por exemplo, classificações - do usuário (HU, et. al, 2009).

5.3 Estratégias de recomendação

O principal objetivo de um SR está na fidelidade do usuário-sistema e no aumento da lucratividade das empresas. Diferentes estratégias são implementadas para personalizar ofertas para um usuário, visando a melhor interação usuário-sistema. Cada estratégia tem seu grau de complexidade na geração das informações. As estratégias de recomendação mais utilizadas segundo (REATEGUI, et.al, 2005) serão apresentadas a seguir:

- **Listas de Recomendação**: Consiste em apresentar listas de itens organizados por tipos. Não há necessidade de uma análise profunda em relação aos dados dos usuários para a criação das listas. Alguns tipos de listas são: “Itens mais vendidos”, “itens populares”, “Novidades”, “ideias para presentes”, entre outros. A principal vantagem desta estratégia está na sua facilidade de implementação. As listão não são personalizadas e são construídas de acordo com a necessidade de marketing.

- **Avaliação dos usuários**: Caracteriza-se em disponibilizar ao usuário a média da avaliação geral dada pelos usuários ao item. Pode ser considerada a estratégia mais simples, pois o sistema apenas disponibiliza a média das avaliações. Uma forma de utilizar é: disponibilizar as avaliações dos usuários sobre determinado item em momentos apropriados.

- **Recomendações personalizadas**: Esta estratégia apresenta recomendações individuais com base no feedback implícito ou explicito do usuário. As recomendações são computadas com a utilização de técnicas de recomendação.

- **Análise de sequência de ações**: Outra forma de fazer recomendações é através da análise das ações do usuário. Como cliques e compras em produtos efetuados por ele durante a navegação no site. Este tipo de análise pode determinar quais são os itens que usuários compram depois de comprar determinado item. Alguns sites de comercio eletrônico utilizam a seguinte frase, “O que os clientes compram depois de ver um item igual a este?” para descrever recomendações baseada na análise de sequência de ações.

- **Recomendações por associação**: Este tipo de estratégia faz a associação entre itens avaliados por usuários. É muito comum no universo do comercio eletrônico, ebay.com, amazon.com, por exemplo. Pode ser considerada a

forma mais complexa de recomendação pois exige uma análise profunda dos hábitos dos usuários para identificar padrões.

- **Associação por conteúdo**: É apresentado uma lista com recomendações geradas a partir da associação por conteúdo dos itens. Em um caso prático, em SR de filmes, o sistema pode se basear no gênero, autor, atores, diretor, entre outros atributos. O SR associa um filme de um diretor A a outro de um diretor B com a seguinte ideia, usuários que assistem filme do diretor A também assistem a filmes do diretor B.

5.4 Arquitetura Geral

Sistemas de Recomendação sugerem itens para o usuário considerando seus comportamentos e preferências. Um modelo geral de processo de recomendação é mostrado na Figura (8).

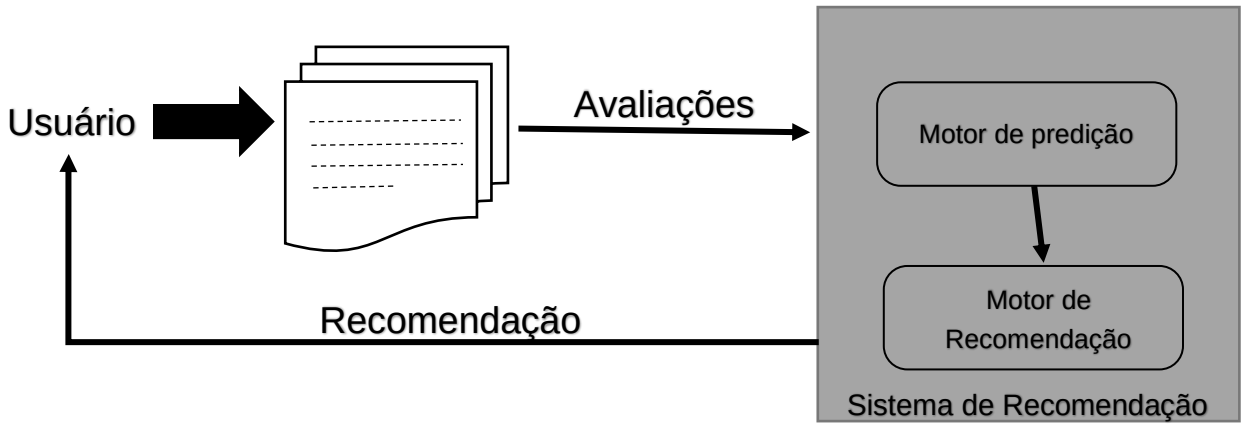

Figura 8 Modelo geral de um sistema de recomendação

De acordo com (KHUSRO, 2016), é possível definir formalmente o sistema de recomendação da seguinte maneira, *S* sendo a coleção de todos os itens que podem ser recomendados aos usuários, *U* como a coleção de todos os usuários, e *F* como uma função utilidade – é a modelagem do conceito de utilidade, que faz uma ordenação dos benefícios para uma pessoa, de acordo com a satisfação que estes lhe trarão – que mede

a utilidade de um determinado item $s\ tal\ que\ s \in S$, para um usuário específico $u\ tal\ que\ u \in U$. Matematicamente, esta função pode ser demonstrada pela Equação (8), onde R representa o conjunto total de itens recomendados.

$$F = U\ X\ S \rightarrow R \quad (8)$$

Para cada usuário, um perfil é criado e atualizado contendo informações como, itens visitados, classificados, comprados ou baixados, por exemplo. Para cada item, metadados são definidos. O sistema de recomendação recebe como input uma matriz usuário-item e produz como output recomendações para um usuário em particular.

	Itens						
	1	2	3	4	5	...	m
Usuários	2	5	3		1		2
	3		2				
	4			5			
	5	3		2	4		
	6						4
	:		3	2			
	n						

Tabela 2 Matriz avaliação dos usuários

De acordo com (MELVILLE e SINDHWANI, 2010), a configuração mais geral na qual sistemas de recomendação são estudados é apresentada na Tabela (2). As preferências dos usuários conhecidas são representadas como uma matriz de n usuários e m itens. A tarefa é prever as avaliações desconhecidas. As avaliações são previstas para todos os itens não observados por um usuário, e os Top N itens com classificação mais alta são apresentados como recomendações. O usuário para qual o sistema computa as recomendações é definido como usuário ativo ou usuário alvo. Cada usuário é representado como um vetor de itens.

5.5 Empresas que usam Sistemas de Recomendação

5.5.1 NetFlix

A NetFlix é uma empresa que fornece filmes e séries de televisão via *streaming.* Cada programa visto e avaliado pelo usuário é utilizado como base para o SR. Em (THIS IS HOW NETFLIX'S TOP-SECRET RECOMMENDATION SYSTEM WORKS, 2017), mais de 80% dos filmes assistidos no Netflix são recomendados pela *Engine* da plataforma.

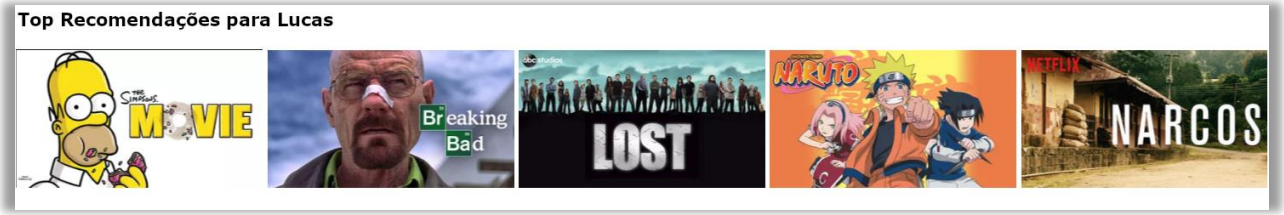

Figura 9 Exemplo de recomendação de filme da Netflix com base no usuário

Em 2009 a empresa concedeu um prêmio de 1 milhão de dólares a uma equipe de desenvolvedores por um algoritmo que aumentasse a precisão de seu SR em 10%. Esse prêmio foi parte da *Netflix Prize*, que foi uma competição pelo melhor algoritmo para recomendar filmes com base na classificação dada pelo usuário.

5.5.2 Amazon

É uma empresa de comércio eletrônico que é referência na utilização de Técnicas de sistemas de recomendação. Cada usuário visualiza no site recomendaçoes baseadas em compras passadas, carrinhos de compras e pesquisas. De acordo com (SMITH, LINDEN, 2017) a empresa lançou a filtragem colaborativa baseada em itens em 1998 que forneceu simplicidade na construção de um SR e escalabilidade mantendo recomendações uteis. Um dos maiores segredos de seu sucesso foi investigado pela revista (FORTUNE, 2012) que concluiu, que para a Amazon, seu SR forneceu um aumento nas vendas de 29%, esse crescimento se deve a forma como a Amazon integrou

recomendações em quase todas as partes do processo de compra, desde a descoberta do produto até a finalização da compra.

Figura 10 Exemplo de recomendação da Amazon.com de acordo com histórico do cliente

Fonte: https://www.mageplaza.com/blog/product-recomendation-how-amazon-succeeds-with-it.html

5.5.3 Spotify

É um serviço de streaming de música que possui um sistema de recomendação personalizado. Cada música escutada pelo usuário é utilizada como base para as próximas recomendações. O Spotify trabalha com várias formas de recomendação, baseada no agrupamento de artistas, em músicas escutadas, usuários similares, gêneros mais escutados, etc.

Uma das estratégias mais interessantes se chama “Discover Weekly” lançada em 2015, é uma lista compreendida por 30 músicas que o usuário nunca escutou antes e que provavelmente vá gostar. Esta lista é atualizada toda segunda feira para milhões de usuários, na qual se baseia nas músicas curtidas por usuários com gostos similares.

Figura 11 Discover weekly do Spotify

6 INVESTIGAÇAO DA RECOMENDAÇÃO

6.1 Confiança em Sistemas de Recomendação

Para (RICCI, et al., 2011), é fato que os Sistemas de recomendação têm uma variedade de propriedades que podem afetar a experiência do usuário, como a precisão, robustez e escalabilidade. Neste tópico será apresentado os métodos Off-line – alguns autores classificam este como método acadêmico de avaliação – e online para realizar a avaliação de SR. Será abordado métricas de avaliação de recomendação Off-line para comparação entre algoritmos.

A confiança em Sistemas de Recomendação pode ser definida como a acurácia do sistema em suas recomendações. Como exemplificado no item (5.1 Técnicas de recomendação), as técnicas nos ajudam a criar um sistema de recomendação mais preciso, visando o benefício direto do usuário.

(HERLOCKER, et al., 2004) propõe dois métodos diferentes para realizar a avaliação de sistemas de recomendação:

- **Análise Off-line**: É implementada utilizando uma base de dados (dataset) com as avaliações fornecidas por usuários. A ideia é simular o comportamento de usuários que interagem com um SR. É considerada como um experimento mais simples, pois as informações dos usuários já foram previamente coletadas e por não ser realizada em produção. Algumas métricas comumente utilizadas são: MAE, RMSE.

- **Online, testes com usuários ativos**: Uma opção mais cara, em que um pequeno grupo de usuários é solicitado para executar um conjunto de tarefas usando o sistema, e respondendo a questionários sobre a sua experiência.

A qualidade de um SR pode ser avaliada através da comparação de recomendações preditas para um usuário alvo à um conjunto de teste de classificações previamente conhecidas deste usuário.

O desenho de critérios de avaliação e seleção de métricas adequadas são um problema fundamental nos sistemas de recomendação. A maioria dos sistemas tradicionais avaliam os resultados e algoritmos considerando um conjunto de dados de teste e aplicam métricas de avaliação em estudos off-line (KHUSRO, 2016).

A métrica mais comum utilizada na literatura é Mean Absolute Error (MAE), dada por:

$$MAE = \frac{\sum_{i=1}^{n} |p_i - r_i|}{n} \quad (9)$$

A partir da definição formal da métrica MAE dada pela Equação (9), a interpretação das variáveis é, n o número de elementos, p_i a avaliação predita e r_i o valor real da avaliação.

Esta formula, calcula o desvio absoluto médio entre o valor estimado pelo SR e o valor real das avaliações do usuário. Dessa forma, encontramos o erro da predição (HERLOCKER, *et al*., 2004). O valor de MAE varia de 0 a ∞, onde 0 significa concordância total e ∞ discordância total. Quanto menor o erro, maior a acurácia da recomendação que implica em gerar recomendações mais adequadas para o usuário.

Outra técnica que também é utilizada, é a *Root Mean Squared Error* ou simplesmente RMSE:

$$RMSE = \sqrt[2]{\frac{\sum_{i=1}^{n} (p_i - r_i)^2}{n}} \quad (10)$$

O RMSE também mede a magnitude média do erro. É a raiz quadrada da média das diferenças quadradas entre a previsão e observação real. Esta técnica foi utilizada como medida de avaliação no famoso prêmio Netflix Prize. A mesma interpretação para

as variáveis da Equação (9) pode ser feita para as da Equação (10). O resultado de RMSE também varia de 0 a ∞ com a mesma interpretação do resultado de MAE para sistemas de recomendação. Entretanto, o RMSE penaliza mais os desvios maiores.

Uma outra técnica que pode ser menos comum de se encontrar na literatura relacionada a SR, mas que tem uma interpretação intuitiva e simples, e que é utilizada para medir a associação ordinal entre grandezas medidas, é o *Kendall Tau Rank Correlation*. Esta técnica avalia o grau de similaridade entre dois conjuntos (ABDI, 2007).

$$\tau = \frac{C-D}{C+D} = \frac{C-D}{\frac{1}{2}n(n-1)} \quad (11)$$

A interpretação das variáveis da Equação (11) é: n o número de classificações, C número de pares concordantes e D número de pares discordantes.

O coeficiente de relação de Kendall terá valores entre -1 e +1, com -1 correspondendo a discordância total entre os pares e +1 a concordância total entre os pares (SZMIDT, KACPRZYK, 2007).

6.2 Desafios e Limitações

Sistemas de recomendação foram propostos com base em técnicas de filtragens. E a implementação de um SR pode gerar alguns desafios que podem dificultar a geração de boas recomendações. Alguns desafios serão abordados a seguir:

- **The Cold-Start Problem:** O termo em português, partida a frio, posse ser visto como uma analogia ao funcionamento de um carro, quando o carro ainda está com o motor frio, ele não funciona tão bem, mas quando atinge a temperatura ideal o carro passa a funcionar bem. Novos itens e novos usuários representam um desafio para os SR. Para o mecanismo de recomendação, significa que as

condições ainda não são as ideais para que o funcionamento seja como esperado, fornecendo boas recomendações.

- **User Problem:** O problema do novo usuário que não tem preferencias prévias implica em não ser possível encontrar usuários semelhantes ou construir um perfil baseado em conteúdo. Tanto a técnica de filtragem colaborativa quanto a filtragem baseada em conteúdo sofrem com esse problema.
- **Item Problem:** O problema surge na filtragem colaborativa, em que um item não pode ser recomendado, a menos que algum usuário o tenha classificado. Obstáculo que não se aplica somente a novos itens, mas a itens escondidos, que é prejudicial para usuários com gostos ecléticos.

As abordagens baseadas em conteúdo não dependem de classificações de outros usuários, elas podem ser usadas para recomendar todos os itens, desde que os atributos dos itens estejam disponíveis para encontrar itens similares. Apenas a técnica de filtragem colaborativa sofre com o Item Problem pois leva em consideração a avaliação do usuário.

- **Sparsity:** Quando, a maioria dos usuários não classifica a maioria dos itens, a matriz de classificação usuário por Item fica muito esparsa. Este é um problema comumente encontrado na utilização de filtragem colaborativa, uma vez que diminui a probabilidade de encontrar um conjunto de usuários com classificações semelhantes (RICCI, et al., p.13, 2011). Um exemplo, no caso de recomendação de filmes, os filmes não populares são raramente recomendados. Uma maneira de tentar resolver esse problema é usando informações adicionais – que podem ser dados demográficos, gêneros de preferência, profissão do usuário - de domínio para o cálculo de similaridade.

- **Grey Sheep:** O problema da ovelha cinza ocorre em sistemas que só utilizam a filtragem colaborativa, onde os usuários tem gostos muito diferentes dos outros, não sendo possível a criação de grupos de usuários com alta similaridade e, portanto, não são capazes de obter benefícios do sistema de

recomendação. Uma forma de resolver esse problema é usando a filtragem baseada em conteúdo onde os itens são sugeridos ao explorar o perfil pessoal do usuário e o conteúdo dos itens (KHUSRO, 2016).

- **Privacy Issue**: Para produzir recomendações personalizadas de qualidade, os SR coletam o máximo de dados do usuário para análise. Isso pode criar uma impressão negativa sobre a privacidade do usuário com o sistema.

Esses desafios são explorados em trabalhos de pesquisa. A investigação nessa área de pesquisa visa explorar e fornecer novos métodos que possam melhorar a qualidade das recomendações.

7 METODOLOGIA

O método cientifico utilizado para o desenvolvimento deste trabalho foi baseado em observar como é o funcionamento dos sistemas de recomendação, estudar a elaboração e as técnicas apresentadas em artigos acadêmicos, escolher um domínio para implementar um caso de estudo para realização de experimentos e analisar os resultados obtidos pelo modelo em questão.

A pesquisa exploratória foi feita através do estudo de artigos, revistas digitais e livros. O material utilizado para estudo pode ser encontrado em ufpr.br, ieee.org, Google scholar, researchgate.net, wired.co.uk, fortune.com e no livro “Recomender Systems Handbook” por (RICCI, et al., 2011), entre outros. Para a construção do protótipo a pesquisa realizada por (MAHDAVI, MORADI, 2011) foi de grande colaboração e incentivo.

Este trabalho tem como finalidade apresentar um estudo na área sistemas de recomendação com a analise da performance de duas técnicas muito utilizadas em plataformas de recomendação que utilizam a filtragem colaborativa. Um sistema de recomendação utiliza um algoritmo de FC para predizer notas desconhecidas para itens não avaliados e recomenda os itens com as melhores notas para o usuário ativo. Neste trabalho será apresentado um protótipo de predição de notas para filmes que um usuário alvo já tenha previamente avaliado. Esta forma será abordada para que seja possível avaliar a qualidade de recomendações através da técnica de filtragem colaborativa com utilização dos algoritmos KNN e K-Means.

Para o desenvolvimento do protótipo foi utilizado a linguagem de programação Python que fornece uma série de ferramentas para o desenvolvimento de um sistema de recomendação. Uma das bibliotecas escolhidas foi Scikit-Learn, que é uma biblioteca para o aprendizado de máquina, na qual auxiliará na criação do protótipo. Também foi utilizada a biblioteca Numpy que auxiliará na manipulação de vetores e a Scipy que fornecerá métodos estatísticos. Outra biblioteca utilizada foi Matplotlib que é uma biblioteca de plotagem 2D em Python. Através desta biblioteca foi possível gerar gráficos que auxiliarão no entendimento do resultado do protótipo.

Para a realização dos testes, foi utilizado o dataset MovieLens 100k. Este dataset foi construído pela GroupLens através da coleta de dados na base do MovieLens.org. O movielens.org é um site de recomendação de filmes mantido pela GroupLens. O dataset está disponível em: https://grouplens.org/datasets/movielens/. A GroupLens Research é um laboratório de pesquisa do Departamento de Ciência da computação e Engenharia da Universidade de Minnesota, na qual é especializado em sistemas de recomendação, comunidades on-line, tecnologias móveis e ubíquas, bibliotecas digitais e sistemas de informações geográficas.

O conjunto de dados do dataset do MovieLens fornece a classificação de 1(ruim) a 5(bom) dada aos filmes pelos usuários. Essa base conta com 100.000 avaliações para 1682 filmes por 943 usuários, cada usuário avaliou pelo menos 20 filmes. Para a construção do sistema que será apresentado por este trabalho, será utilizado os atributos fornecidos pela Tabela (3), como “Usuário”, “Item” e “Avaliação”. O *dataset* contém outras informações que não serão utilizadas pelo motor de recomendação, como dados demográficos sobre o usuário, idade, gênero, ocupação, cep, nome do filme, data de publicação e URL para o site do IMDB (é uma base de dados online de informação sobre música, cinema, filmes).

Usuário	Item	Avaliação
1	1	1
2	2	5
3	10	1
1	3	2
5	1	3

Tabela 3 Modelo da Base de Dados que será utilizada pelo protótipo

Na prática comercial é muito mais comum ter conjunto de dados baseados em ações implícitas dos usuários. Para este trabalho será utilizado apenas dados explícitos fornecidos pelos usuários.

Existem outros datasets que também poderiam ser utilizados para um protótipo de pesquisa em sistemas de recomendação, como TMDB (https://www.themoviedb.org), IMDB (https://www.imdb.com), NetFlix Prize Data (https://www.kaggle.com/netflix-

inc/netflix-prize-data), e, que também são disponíveis para estudo, mas mantidos por outras comunidades.

A incompletude presente no dataset devido a todos os usuários não avaliarem todos os filmes, reduz a precisão na recomendação de filmes. Segundo (CARVALHO, 2017), dados faltantes são um problema para a análise de informações e reconhecimento de padrões em todas as áreas do conhecimento. E que a análise incompleta pode gerar respostas com desvios, tendências e erros que afetam a tomada de decisão. Para obter recomendações e poder computar predições com baixa taxa de erro seria necessário a completude dos dados. Com isso, em (8 Desenho Experimental) será abordado uma forma para resolver este problema.

Os SR foram investigados por décadas em pesquisas acadêmicas. Há muitos trabalhos de pesquisa introduzindo diferentes algoritmos e, para comparar diferentes técnicas são utilizadas medidas de avaliação e métodos de simulação como *Hold-out validation* e *K-fold cross validation*. Neste trabalho será realizado um estudo off-line, com a validação de divisão (Hold-out validation) das avaliações processadas pelo protótipo. Para investigarmos a eficiência das predições das notas dos filmes realizadas pelo protótipo, será criado dois subconjuntos a partir do dataset. Um subconjunto A com 70% das classificações dos usuários, que será usado para treinamento dos dados, e o restante, conjunto B com 30% para teste. As avaliações fornecidas pelos usuários do conjunto B serão chamadas de RR (Rank Real) e as computadas pelo protótipo de predições ou RP (Rank Predito).

O estudo off-line foi escolhido por causa da sua facilidade de implementação para pesquisa. Segundo (JANNACH, et al., 2012) é a avaliação predominante na comunidade de recomendação. Para (BELL, et al., 2015) também é verdade no campo de pesquisa de sistemas de recomendação, onde a maioria das abordagens de recomendação é avaliada offline, apenas 34% das abordagens são avaliadas com estudos de usuários e apenas 7% com avaliação on-line.

Para a execução dos algoritmos propostos será usado vários valores de vizinhança para o KNN, de modo igual, vários tamanhos de grupos diferentes para o K-Means. O resultado de cada experimento será a média das predições para os usuários do conjunto

de teste. Tanto a vizinhança do KNN quanto a quantidade de clusters do K-Means são definidos por um único parâmetro K que define o desempenho de cada abordagem.

A escolha do valor de K é investigada por inúmeras pesquisas. Valores muito grandes ou muito pequenos para K podem implicar em resultados com baixa precisão dependendo do problema envolvido. Para (HASSANAT, et al., 2014), escolher o valor do parâmetro K para o KNN envolve uma escolha empírica, na qual para cada problema são testados diferentes números de vizinhos mais próximos e o parâmetro com melhor desempenho é escolhido. A mesma interpretação formulada pode ser feita para o K-Means, em que, para encontrar o melhor valor para K trata-se de uma escolha empírica.

Todo o código desenvolvido neste trabalho está hospedado na plataforma de controle de versão GitHub, e pode ser acessado através do link: https://github.com/lukascivil/TCC

No próximo capitulo deste trabalho será apresentado a estrutura de dados escolhida e tratamento dos dados para o input no protótipo, o desenvolvimento do sistema e a geração das predições. No Tópico (9 Análise dos Resultados) será apresentado os resultados do protótipo. Para apresentar os resultados de forma didática, a fim de expressar visualmente dados e valores numéricos facilitando a compreensão, serão utilizados tabelas e gráficos em linhas.

8 DESENHO EXPERIMENTAL

Neste capitulo, será apresentado o desenvolvimento do protótipo e experimentos realizados no qual consistem na mudança do paramento K, que representa os vizinhos mais próximos ou a quantidade de grupos. Este capitulo será apresentado em 3 partes, arquitetura do protótipo, tratamento dos dados e motor de recomendação.

Um sistema de recomendação de filmes usa os algoritmos de filtragem colaborativa para predizer notas de filmes não avaliados e recomenda os Top N para o usuário alvo. No protótipo proposto, o motor de recomendação prediz as notas de filmes já avaliados pelo usuário alvo. Para não induzirmos o sistema ao erro, o sistema não leva em consideração as avaliações reais (RR) do usuário alvo para o conjunto de treinamento. A ideia é analisar a precisão dos algoritmos de mineração de dados, KNN e K-Means no mundo da recomendação de filmes. Dessa forma é possível medir a acurácia das técnicas envolvidas e poder inferir que o sistema faz recomendações de qualidade.

8.1 Arquitetura do protótipo

A arquitetura geral do protótipo desenvolvido tem a seguinte projeção:

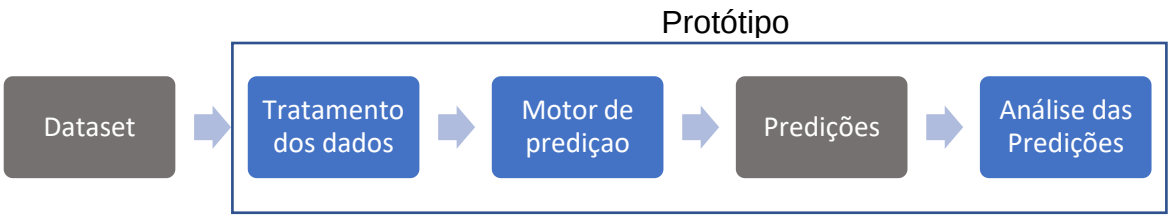

Figura 12 Modelo criado para simulação de um sistema de predição

O modelo adotado utiliza a técnica de filtragem colaborativa baseada no usuário para computar as predições. Como pode ser observado pelo modelo da Figura (12), a arquitetura possui algumas etapas que podem ser resumidas da seguinte maneira:

- Os dados serão estruturados a fim de se obter duas matrizes, que serão inputs para o motor de predição. Será gerado, uma matriz de treinamento usuário-

filme onde cada vetor/linha é correspondido por um usuário e suas avaliações e uma outra matriz de teste também usuário-filme.

- O motor de predição irá calcular quais são as melhores notas para os filmes assistidos por um usuário em particular da matriz de teste.
- A análise das predições consiste em encontrar o erro e a similaridade entre o RP e RR.

8.2 Tratamento dos dados

O GroupLens disponibiliza seus *datasets* em formatos diferentes, para este projeto foi utilizado o 100k que é compreendido pelos arquivos u.data, u.genre, u.info, u.item, u.occupation. O arquivo utilizado é o u.data que é composto por 100.000 avaliações de usuários.

Na implementação do modelo proposto, o tratamento de dados consiste de um conjunto de operações automatizadas para organização e estruturação dos dados a serem processados. Dessa forma o arquivo u.data é utilizado para o processamento a fim de construir a matriz geral.

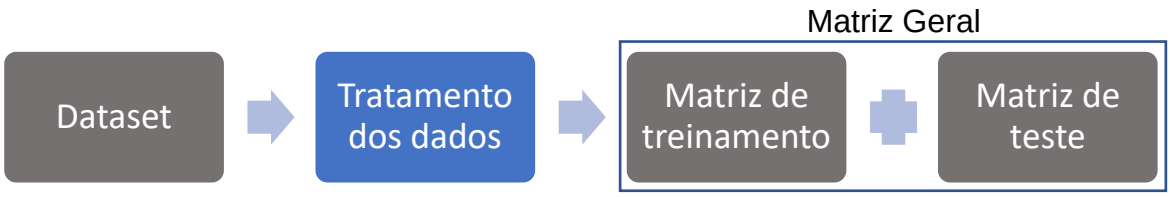

Figura 13 Tratamento dos dados

Para representarmos as informações do *dataset* em estrutura de dados para input no protótipo, será definido uma matriz geral Tabela (4) usuário-filme onde cada célula nxm corresponde a nota que usuário n deu ao filme m:

		Filmes					
	1	2	3	4	5	...	m
	2	5	3		1		2
	3		2				
	4			5			
Usuários	5	3		2	4		
	6						4
	:		3	2			
	n						

Tabela 4 Matriz Geral

O nível de dispersão de uma base de dados é determinado pela proporção de entradas ausentes para o número total de entradas. O nível de dispersão dessa base de dados pode ser calculado da seguinte maneira:

$$S = 1 - \left(\frac{R}{UxM}\right) \quad (12)$$

$$S = 1 - \left(\frac{100000}{943 * 1682}\right) \approx 0{,}937$$

Na Equação (12), R é a quantidade de avaliações, U é a quantidade de usuários e M é a quantidade de filmes. A interpretação é que para cada 1000 entradas na matriz UxM, 937 estão vazias, ou seja, sem avaliação.

Para resolvermos o problema da incompletude, vamos preencher os espaços vazios da matriz com a nota “0” para os filmes que o usuário não avaliou. Esta matriz é apresentada pela tabela abaixo:

		Filmes					
		1	2	3	4	5	...	m
Usuários	2	5	3	0	1	0	2
	3	0	2	0	0	0	0
	4	0	0	5	0	0	0
	5	3	0	2	4	0	0
	6	0	0	0	0	0	4
	:	0	3	2	0	0	0
	n	0	0	0	0	0	0

Tabela 5 Matriz Geral usuário-filme tratada

As avaliações dadas pelos usuários estão entre 1 e 5. O valor 1 para o menor interesse do usuário em relação ao filme, 5 para o maior interesse e 0 para “sem avaliação”, que indica que o usuário não assistiu ao filme.

A matriz geral tratada será particionada em duas submatrizes, de teste com 30% dos dados da matriz geral e de treinamento com 70%.

8.3 Motor de Predição

Nesta seção serão apresentados os passos utilizados para a geração da lista de notas de filmes, que é interpretada como a predição das notas dos filmes do usuário de teste ou RP. Para gerar esta lista é necessário um motor de predição, e para cria-lo serão utilizados os algoritmos KNN e K-Means fornecidos pela biblioteca Scikit-Learn. O motor foi dividido em quatro etapas para que as predições possam ser geradas. As etapas são as seguintes:

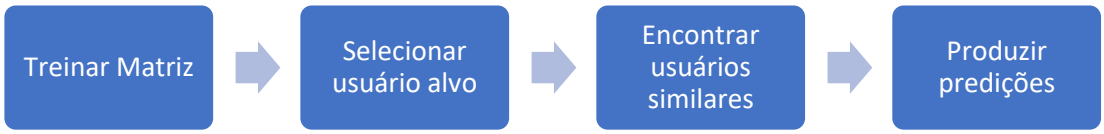

Figura 14 Funcionamento do motor de predição

Processar o conjunto de treinamento, escolher um usuário alvo da matriz de teste, encontrar os usuários similares e predizer notas para uma lista de filmes com base nos usuários similares, compreendem a execução de um experimento. Cada experimento é compreendido pelos dois algoritmos descritos a seguir:

- **Algoritmo KNN:**

  Passo 1: Definir parâmetros de entrada: K número de vizinhos similares e métrica que será utilizada para calcular a distância entre os vizinhos.

  Passo 2: Processar matriz de treinamento, esta matriz é composta por todos os usuários. de treinamento por todos os filmes.

  Passo 3: Definir o vetor usuário alvo, este vetor é composto por todos os filmes.

  Passo 4: Encontrar na matriz de treinamento, os K usuários similares ao usuário alvo.

- **Algoritmo K-Means:**

  Passo 1: Definir parâmetros de entrada: K número de clusters que serão formados pelos usuários da matriz de treinamento

  Passo 2: Processar matriz de treinamento, esta matriz é composta por todos os usuários de treinamento por todos os filmes. Será criado k grupos de usuários.

  Passo 3: Definir o vetor usuário alvo, este vetor é composto por todos os filmes.

  Passo 4: Encontrar dentre os k grupos de usuários, o grupo no qual o usuário alvo pertence. Os n usuários pertencentes a este grupo, são os usuários similares ao usuário alvo.

Cada algoritmo executado encontrou um conjunto de usuários similares. Após a construção da vizinhança, as predições serão computadas para cada conjunto através de alguns passos:

Passo 1: Selecionar os filmes dos usuários similares que foram avaliados pelo usuário alvo.

Passo 2: Dentre as notas que os usuários similares deram para os filmes, encontrar a nota mais popular para cada filme selecionado no passo anterior. E para cada nota dada por cada usuário similar, um peso é atribuído com o intuito de valorizar os usuários mais similares. Este passo é definido pela Equação (5).

A predição de notas foi computada baseada no conhecimento de uma rede de confiança do usuário. Essa rede é definida pelos vizinhos mais próximos com gostos parecidos, e que vão ajudar na predição das melhores notas para uma lista de filmes. As melhores notas foram computadas através das notas mais comuns para a lista de filmes e são entendidas como “as melhores” por expressarem baixa diferença quando comparadas as notas reais.

A precisão preditiva será estudada através de valores fornecidos pelo protótipo no próximo tópico, onde serão utilizadas métricas de avaliação comuns em sistemas de recomendação.

9 ANÁLISE DOS RESULTADOS

Nesta seção, serão apresentados os resultados obtidos pelo protótipo. Os algoritmos foram experimentados com vários valores de K. Diferentes métricas de avaliação foram utilizadas nos experimentos off-line do protótipo. Ao final deste tópico será discutido sobre os valores observados.

9.1 Avaliação off-line

Os resultados que serão exibidos nesta seção foram obtidos através da avaliação off-line com a utilização de métricas como MAE, RMSE e Kendall-Tau Rank Correlation. A avaliação foi computada para cada algoritmo com diferentes valores de K, e por fim é calculado a média dos resultados das métricas. Os valores de K que serão utilizados pertencem ao conjunto {1, 2, 3, 4, 5, 6, 7, 8, 9, 10}. O processo é explicado pelo pseudocódigo a seguir:

```
Para KNN e K-Means execute
    Para cada valor de K execute
        Para cada usuário u da matriz de teste execute
            RP = ComputarPredicoes()
            Temp_MAE[u] = MAE(RP, RR)
            Temp_RMSE[u] = RMSE(RP, RR)
            Temp_KENDALLTAU[u] = KENDALLTAU(RP, RR)
        FimPara
        MAE =  (Temp_MAE / usuários similares)
        RMSE =  (Temp_RMSE / usuários similares)
        KENDAU = (Temp_KENDALLTAU / usuários similares)
    FimPara
FimPara
```

O pseudocódigo apresentado reproduz a estratégia utilizada para a captura dos dados que demonstram a performance da simulação dos algoritmos utilizados.

A biblioteca Scikit-Learn forneceu os métodos MAE e MSE. Para calcular o RMSE foi calculado a raiz quadrada de MSE. Para encontrar o valor de Kendall-Tau foi utilizada a Scipy. A função de Kendall Rank Correlation fornecida pela biblioteca Scipy é uma variação da função apresentada por (SZMIDT, KACPRZYK, 2007).

A seguir será apresentado duas tabelas com os resultados obtidos, a primeira tabela fornece os valores de MAE e RMSE e a segunda os valores de Kendall-tau. Os resultados serão fixados em 4 casas decimais sem arredondamento.

	MAE		RMSE	
K	KNN	K-Means	KNN	K-Means
1	1,9815	0,8461	2,3166	1,1474
2	1,7662	0,8757	2,1294	1,1845
3	1,6465	0,8791	2,0210	1,1885
4	1,5560	0,8866	1,9342	1,1992
5	1,4978	0,9021	1,8804	1,2174
6	1,4483	0,9246	1,8285	1,2421
7	1,4213	0,9201	1,8022	1,2382
8	1,3944	0,9140	1,7742	1,2342
9	1,3749	0,9309	1,7556	1,2505
10	1,3529	0,9399	1,7338	1,2652

Tabela 6 Resultados obtidos através das métricas MAE e RMSE

	Kendall-Tau	
K	KNN	K-Means
1	0,1600	0,2843
2	0,1837	0,2610
3	0,1939	0,2612
4	0,1986	0,2565
5	0,1965	0,2456
6	0,2069	0,2440
7	0,2062	0,2437
8	0,2088	0,2396
9	0,2095	0,2365
10	0,2104	0,2339

Tabela 7 Resultados obtidos através de Kendall-Tau

A seguir será apresentado três gráficos – gerados através da utilização da biblioteca Matplotlib - que demonstram o resultado da estratégia implementada para analisar a performance dos algoritmos. O primeiro fornece os resultados com a utilização de MAE, o segundo com RMSE e o terceiro através da utilização do Kendall-Tau. Nos gráficos que serão apresentados, os valores no eixo vertical representam valores referentes a métrica abordada e os valores no eixo horizontal representam valores de K. Para cada algoritmo o valor de K tem significado diferente, para o KNN significa a quantidade de usuários vizinhos (similares), e para o K-Means representa a quantidade de clusters.

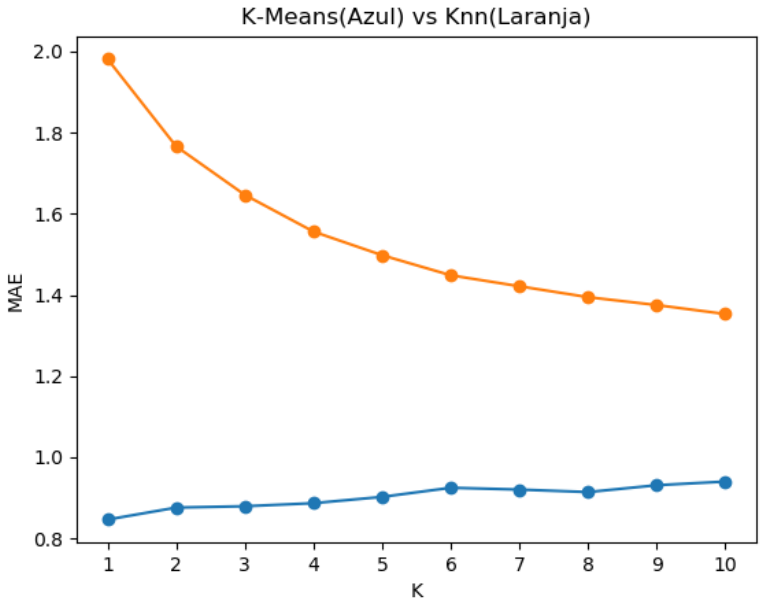

Figura 15 Execução dos algoritmos com a variação de k para análise de MAE

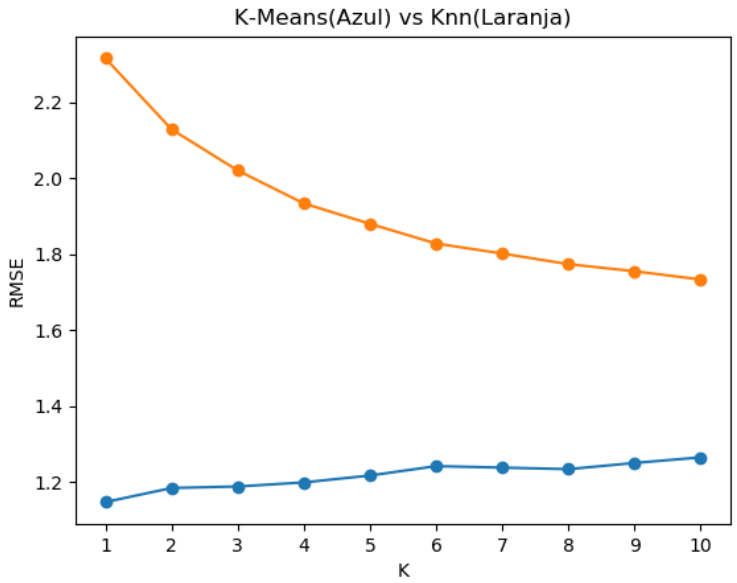

Figura 16 Execução dos algoritmos com a variação de k para análise de RMSE

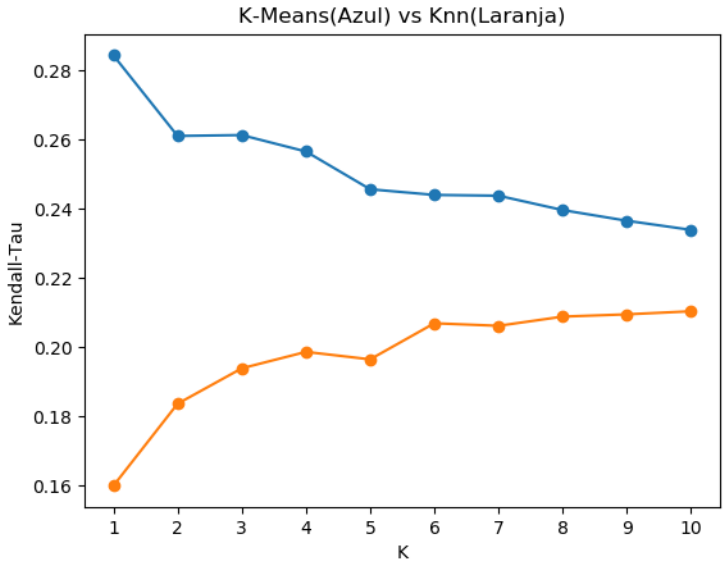

Figura 17 Execução dos algoritmos com a variação de k para análise de Kendall-Tau

9.2 Conclusão da avaliação

As variações do parâmetro k, apresentadas na Tabela (6 e 7) promoveram mudanças sensíveis nos resultados tanto para o KNN quanto para o K-Means. Em particular, as métricas que expressam o erro de previsão e o cálculo da correlação funcionaram conforme o esperado.

A partir da análise das tabelas e gráficos apresentados no tópico anterior, é possível notar que quanto mais usuários como vizinhos mais próximos, menor é o erro apresentado pelas predições. E quanto maior o número de clusters, maior é o erro apresentado. Os valores de RMSE são maiores que os de MAE, por que RMSE enfatiza erros maiores no cálculo. Também é possível notar que os valores da correlação de Kendall variaram conforme a variação dos erros.

Outros testes foram realizados como parte do estudo. Primeiro teste, a matriz de treinamento para cada usuário foi gerada com base nos filmes que o usuário assistiu, dessa forma foi possível observar que os erros retornados pelas métricas foram menores para os dois algoritmos e que a execução do algoritmo exigiu um maior tempo de processamento. Segundo teste, foi feito a alteração da métrica do KNN que define os vizinhos mais próximos para a *correlation (*correlação entre as distancias fornecido pelo scikitLearn). Essa métrica apresentou erros menores se comparados aos erros obtidos neste trabalho que se basearam na utilização da distância euclidiana. E o terceiro teste consistiu em não utilizar o peso, que dá maior importância para os usuários mais similares. Dessa forma os resultados para o KNN e o K-Means demonstraram erros maiores.

10 TRABALHOS RELACIONADOS

Após a apresentação sobre sistemas de recomendação e um modelo que exemplificasse o funcionamento com a avaliação das predições a fim para saber o quão fiel os algoritmos seriam para recomendação, será apresentado alguns trabalhos relacionados a área de recomendação de informação. Esses serviram como base para o desenvolvimento deste corrente trabalho.

10.1 MovieLens

O MovieLens.org é um site de pesquisa na área de Sistemas de recomendação administrado pela GroupLens Research da Universidade de Minnesota. De acordo com o time do MovieLens:

> O MovieLens usa a tecnologia “Filtragem Colaborativa” para fazer recomendações de filmes que você possa gostar, e para ajudá-lo a evitar os que você não deseja. Com base nas suas classificações de filmes, o MovieLens gera previsões personalizadas para filmes que você ainda não assistiu. O MovieLens é um veículo de pesquisa exclusivo para dezenas de alunos de graduação e pós-graduação que pesquisam vários aspectos das tecnologias de personalização e filtragem. (Fonte: movielens.org/info/about)

Como visto em capítulos anteriores, foi utilizado o *dataset* do MovieLens para a construção do sistema exemplo. Para manterem os *datasets* atualizados eles disponibilizam um sistema de recomendação de filmes para interação com o usuário, que segundo (CHANG, 2015), baseia suas recomendações em informações fornecidas pelos usuários do site, como a classificação de filmes. O Site utiliza vários algoritmos de recomendação, incluindo algoritmos de filtragem colaborativa baseado em item e baseado em usuário.

Inicialmente, logo após o usuário entrar no sistema pela primeira vez, ele deverá avaliar grupos de filmes – gerados por algoritmos de agrupamento – nos quais tenha mais interesse. As preferencias registradas por essa pesquisa permitem que o MovieLens faça recomendações iniciais de filmes que provavelmente o usuário o classificará.

Figura 18 Processo inicial de escolha de grupo de filmes

Após este procedimento, o usuário deverá avaliar pelo menos 15 filmes com a nota de 1 a 5 como parte do processo de inscrição.

Esse passo inicial, conhecido como *Preference Elicitation Method*, ajuda a resolver o problema de *ColdStart* de usuário. De acordo com uma análise dos dados feita pelos próprios pesquisadores do MovieLens, os usuários levam em média 6,8 minutos para concluir esse processo, e 12,6% desses usuários não conseguem concluir o processo inicial e nem sequer chegam a primeira página, onde recebem as recomendações. Por outro lado, os sistemas que não aprendem as preferências do usuário fornecerão recomendações mais pobres e podem não ganhar a confiança dos usuários. Conforme o estudo, também foi possível perceber que os usuários que recebem recomendações ruins são menos ativos que outros usuários (CHANG, 2015).

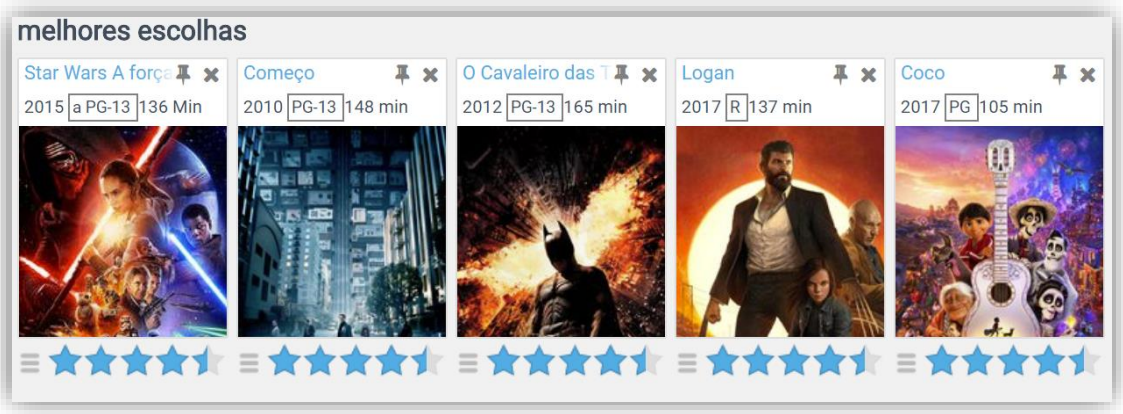

Figura 19 Recomendação inicial de filmes do MovieLens

A Figura (19) exibe recomendações personalizadas com base no perfil do usuário. As recomendações foram geradas após o usuário executar as etapas *Preference Elicitation* e avaliar 15 filmes.

O MovieLens proporcionou o estudo de sistemas de recomendação através de um exemplo online e o *Dataset* para a implementação do modelo proposto por este trabalho.

10.2 Mahdavi e Moradi (2011)

Mahdavi e Moradi, (2011), propuseram a construção de um filtro colaborativo usando algoritmo de clusterização que se baseia na frequência de avaliações dos usuários para computar predições.

O objetivo deste trabalho é criar um algoritmo de votação para gerar recomendação para um usuário ativo e comparar ao tradicional algoritmo de filtragem colaborativa. A comparação consiste em analisar o erro das predições computadas e o tempo de execução dos algoritmos.

Foi utilizado o *dataset* do MovieLens para a realização dos experimentos. A estrutura dos dados consiste em uma matriz M por N, onde M é o número de usuários, que representa a quantidade de pontos no espaço n dimensional e N é o número de itens.

Cada célula da matriz pode conter um valor de 1 a 5 que representa a avaliação do usuário para o item ou o valor 0 para que significa que o usuário não avaliou o item.

Os autores explicam que o tradicional filtro colaborativo tem dois passos principais, o primeiro é computar a similaridade entre o usuário ativo e todos os outros usuários e depois gerar predições com base no primeiro passo. Como métrica de similaridade foi utilizado correlação de Pearson. As predições são computadas levando em consideração a similaridade do usuário ativo aos outros usuários, ou seja, usuários com maior similaridade tem maior peso na sugestão. A construção se baseou em dois tipos de sistemas colaborativos, baseado no usuário e em item.

A proposta era desenvolver um algoritmo baseado no K-Means para encontrar um grupo de usuários similares. Para encontrar os usuários similares foi usado *Minkowski distance* ao invés de formulas de similaridade que foram usadas no filtro colaborativo tradicional. Após encontrar os n vizinhos mais próximos, que pertencem ao cluster, foi possível executar a análise de votação para predição. A análise de votação baseou-se em procurar as notas para os itens com base na popularidade das notas dadas pelos usuários aos itens. Os autores explicam que para refinarem o resultado, atribuíram um peso para cada vizinho de acordo com a distância do usuário ativo, o que adiciona um peso maior para usuários mais similares.

A metodologia de avaliação dos autores consistiu em utilizar métodos para analisar a acurácia dos algoritmos de recomendação e analisar o tempo consumido por cada algoritmo. O *dataset* foi dividido em 80% para treinamento e 20% para teste. O método *5-fold cross validation* foi utilizado para comparação. Foi utilizado NMAE - que é a métrica MAE normalizada – para mostrar os erros em forma de porcentagem. Quanto mais baixo for o valor de NMAE, menor o erro e maior a acurácia. O algoritmo tradicional foi executado usando vários tamanhos de vizinhança e foi calculado a média dos resultados. De modo igual, o algoritmo K-Means foi experimentado com vários tamanhos de grupos diferentes, e o resultado é a média dos experimentos.

Os autores concluíram que a proposta é mais precisa do que o filtro colaborativo tradicional em aproximadamente 60%. E que a proposta apresentou uma qualidade satisfatória nas recomendações mesmo com vários tamanhos diferentes de grupos de

usuários. Entretanto exige mais tempo para a execução. O NMAE para o método tradicional apresentou 85% e para o método proposto 22%.

Essa pesquisa forneceu artifícios necessários para a elaboração do corrente trabalho. O objetivo e a implementação do algoritmo que fornece predições a partir da clusterização serviram como base para a implementação do protótipo.

11 CONCLUSÃO E TRABALHOS FUTUROS

Um Sistema de Recomendação é uma aplicação de inteligência compreendida por um subconjunto de sistemas de filtragem de informações. É muito útil para serviços que apresentam problemas devido ao excesso de informação, os quais dificultam os usuários em escolher algum produto dentre várias alternativas apresentadas. Além disso, a necessidade de sugestões personalizadas está cada vez mais crescente, visando a busca por informações relevantes.

Todo o estudo apresentado, abordou exemplos na área de Sistemas de recomendação, que ilustraram claramente como é o funcionamento e a elaboração de um estudo off-line. Um questionamento que pode ser feito é, a falta de poder de previsão das avaliações está intimamente ligada a ignorância dos fatores humanos? A avaliação off-line implica na falta desses fatores – de satisfação do usuário com as recomendações - que podem influenciar fortemente na qualidade das recomendações.

Este trabalho também exemplificou algoritmos que buscam otimizar a qualidade das recomendações colaborativas e a análise dos resultados obtidos. O protótipo de um sistema de recomendação de filme implementado em um ambiente de teste funcionou como esperado, e os algoritmos geradores de predição efetuaram os cálculos corretamente. Embora os resultados tenham sido satisfatórios, é necessário uma pesquisa futura que utilize a técnica de filtragem hibrida com a finalidade de obter melhores resultados.

12 REFERENCIAS BIBLIOGRÁFICAS

[1] **Amazon's recommendation secret**, Disponível em: http://fortune.com/2012/07/30/ amazons-recommendation-secret, acesso em: 14 Abril 2018.

[2] SMITH, Brent, et al., **Two Decades of Recommender Systems at Amazon.com**, Disponível em: https://www.computer.org/csdl/mags/ic/2017/03/mic2017030012.html, Acesso em: 29 Abril 2018.

[3] MELVILLE, Prem, et al., **Recommender Systems,** Disponível em: http://www.prem-melville.com/publications, Acesso em: 29 de Abril 2018.

[4] **This is how Netflix's top-secret recommendation system works.** Disponível em: http://www.wired.co.uk/article/how-do-netflixs-algorithms-work-machine-learning-helps-to-predict-what-viewers-will-like. Acesso em: 15 Abril 2018

[5] RICCI, et al., **Recommender Systems Handbook**, edição 2011, NewYork: Springer Science+Business Media, 2010.

[6] O'BRIEN, Jeffrey, **The race to create a 'smart' Google**, Disponível em: http://archive.fortune.com/magazines/fortune/fortune_archive/2006/11/27/8394347/index.htm Acesso em: 29 Abril 2018.

[7] BURKE, Robin. **Hybrid recommender systems: survey and experiments**, 2002

[8] HU, Yifan, et. al, **Collaborative Filtering for Implicit Feedback Datasets** Disponível em: https://ieeexplore.ieee.org/document/4781121/, acessado em: 01/05/2018

[9] ABDI, Hervé, **The Kendall Rank Correlation Coefficient**, The University of Texas at Dallas, 2007

[10] SZMIDT, Eulalia, KACPRZYK, Janusz, **The Spearman and Kendall rank correlation coefficients between intuitionistic fuzzy sets**, WIT – Warsaw School of Information Technology ul. Newelska ,2011.

[11] HERLOCKER, Jonathan L. et Al., **Evaluating collaborative filtering recommender systems,** ACM Transactions on Information Systems, 2004

[12] HERLOCKER, Jonathan L., **Understanding and Improving Automated Collaborative Filtering Systems**, University of Minisota, Minisota.

[13] KHUSRO, Shah et al., **Recommender Systems: Issues, Challenges, and Research Opportunities,** Pakistan: Springer Science+Business Media Singapore, 2016

[14] CHANG, Shuo, **Using Groups of Items for Preference Elicitation in Recommender Systems**, Vancouver: CSCW, 2015

[15] OLIVEIRA. **Clusterização ou Agrupamento de Dados**. IME: IME Unicamp, 2012. 17 slides. Disponível em: https://www.ime.unicamp.br/~wanderson/Aulas/Aula6/MT803-Aula06-Clusterizacao.pdf, acessado em: 15/05/2018

[16] CARVALHO, Melissa, **Dados faltantes em análises: uma revisão sobre métodos estatísticos flexíveis a incompletude**, UFPR, 2017, Disponível em: https://eventos.ufpr.br/smne/SMNE2017/paper/viewFile/576/223, acessado em: 28/05/2018.

[17] REATEGUI, Cazella et al., "**Personalização de Páginas Web através dos Sistemas de Recomendação"**, 2005, Disponível em: http://osorio.wait4.org/publications/2006/Reategui-et-al-IHC2006.pdf, acessado em: 28/05/2018.

[18] MAHDAVI, Mehregan; Moradi, Gilda. "**A New Collaborative Filtering Algoritm Using K-Means Clustering and Neighbors Voting**" IEEE, 2011.

[19] JANNACH, Dietmar et. Al., **Recommender Systems in Computer Science and Information Systems – A Landscape of Research**" Proceedings of the 13th International Conference, ECWeb, 2012.

[20] BELL, Joeran et. Al., **Research Paper Recommender Systems: A Literature Survey**, International Journal on Digital Libraries, 2015.

[21] HASSANAT, Ahmad, et Al., **Solving the Problem of the K Parameter in the KNN Classifier Using an Ensemble Learning Approach**, International Journal of Computer Science and Information Security, 8 de Agosto de 2014. Disponível em: https://arxiv.org/abs/1409.0919, acessado em : 26/06/2018.

[22] SAWAR, Badrul, et Al., **Item-based Collaborative Filtering Recommendation Algorithms**, GroupLens Research Group/Army HPC Research Centerv, University of Minnesota, 1-5 de Maio de 2001.

13 APÊNDICE – PARTE DO CODIGO FONTE

- **Encontrar as melhores notas para uma lista de filmes. Função utilizada tanto pelo KNN quanto pelo K-Means**

```
# Gerar predicao de notas para lista de filmes
    def predictRank(self, similarusers_matrix,
activeuser_whatchedmovies, weights):

        # Contador de avaliacoes
        ratings = {1: 0, 2: 0, 3: 0, 4: 0, 5: 0}
        rp = {}

        # Predizer notas aos filmes
        for movie in activeuser_whatchedmovies:
            ratings = {1: 0, 2: 0, 3: 0, 4: 0, 5: 0}
            i = 0
            for user in similarusers_matrix:
                rating = user[movie]
                if rating > 0:
                    ratings[rating] += (1 * weights[i])
                i += 1
            rp.setdefault(movie, None)
            # Pega a nota com maior ocorrencia , ex: {1: 3, 2: 7, 3:
19, 4: 26, 5: 6} => 4
            rp[movie] = max(
                ratings.items(), key=operator.itemgetter(1))[0]

        return rp
```

- **KNN com a métrica de distância euclidiana como parâmetro para encontrar os vizinhos mais próximos**

```
    # Train KNN
    def trainknn(self, X, k):
        # Instanciar o objeto KNN e treinar
        self.nbrs = NearestNeighbors(n_neighbors=k, metric='euclidean')
        self.nbrs.fit(X)
```