UNIVERSIDADE FEDERAL DE SERGIPE

CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA

DEPARTAMENTO DE COMPUTAÇÃO

Sistemas de Recomendação Baseado em Otimização Multiobjetivo para Recomendação de Filmes

Trabalho de Conclusão de Curso

Matheus Santos Almeida

São Cristóvão – Sergipe

2020

UNIVERSIDADE FEDERAL DE SERGIPE

CENTRO DE CIÊNCIAS EXATAS E TECNOLOGIA

DEPARTAMENTO DE COMPUTAÇÃO

Matheus Santos Almeida

Sistemas de Recomendação Baseado em Otimização Multiobjetivo para Recomendação de Filmes

Trabalho de Conclusão de Curso submetido ao Departamento de Computação da Universidade Federal de Sergipe como requisito parcial para a obtenção do título de Bacharel em Ciência da Computação.

Orientador(a): André Britto de Carvalho

São Cristóvão – Sergipe

2020

Resumo

Os sistemas de recomendação tem sido muito estudados nos últimos anos devido a sua capacidade de filtrar a grande quantidade de informação da internet para melhorar o acesso dos usuários em aplicações web. Porém, a maior parte dos sistemas de recomendação utilizam apenas um critério no momento da recomendação, a acurácia da recomendação, mas essa abordagem é limitada podendo gerar recomendações enviesadas. Sendo assim, nesse trabalho, será proposto um sistema de recomendação baseado em conteúdo para recomendação de filmes que utiliza mais outros dois critérios além da acurácia: diversidade e novidade da recomendação, com isso, a recomendação foi modelada como um problema de otimização multiobjetivo. Portanto, neste sistema, a recomendação será feita a partir da execução de um algoritmo multiobjetivo para solucionar o problema de recomendação. Esse algoritmo fornecerá como resultado, um conjunto de soluções, mas, como não haverá garantia que essas soluções pertencem ao conjunto de itens reais, serão recomendados os filmes mais similares as soluções resultantes. O sistema será avaliado utilizando os *datasets* de filmes Movielens e IMDB e seus resultados serão comparados com sistemas de recomendação do estado da arte.

Palavras-chave: sistemas de recomendação. otimização multiobjetivo. aprendizado de máquina. recomendação de filmes.

Lista de ilustrações

Lista de tabelas

Sumário

1

Introdução

A internet já é realidade para uma grande parcela da população, em 2018, o número de usuários passou de 4 bilhões (KEMP, 2018). Como consequência, a quantidade de informação que circula na internet também cresce com grande velocidade e a perspectiva é que, em 2020, o volume de dados na internet seja aproximadamente 40 trilhões de Gigabytes (ALBUQUERQUE, 2017). Com isso, diversas ferramenta tem sido desenvolvidas para facilitar a busca de informações de um usuário na internet, com destaque para os sistemas de recomendações.

Sistemas de recomendações são técnicas que fornecem sugestões de itens para um usuário (RICCI; ROKACH; SHAPIRA, 2011). Essas técnicas são bastante utilizadas em aplicações que lidam com uma grande quantidade informações e que precisam filtrar uma parte dessas informações para melhorar a navegação do usuário na aplicação, algumas aplicações que utilizam essas ferramentas são o *You Tube*, *Netflix*, *Amazon* e etc.

Uma das áreas mais recorrentes para sistemas de recomendações é a área de recomendações de filmes. Essa área ganhou muito destaque devido ao Netflix Prize, que foi uma competição criada pela empresa de *streaming* de filmes, Netflix, para encontrar um sistema de recomendação que apresente melhores resultados com relação ao sistema utilizado pela empresa (NETFLIX, 2009). Essa competição não trouxe somente importância para recomendação de filmes, fazendo com que diversas aplicações relacionadas ao tema tenham surgido, mas também ajudou a transformar sistemas de recomendação em uma área muito importante para pesquisas, fazendo com que diversos sistemas de recomendação tenham sido propostos e novas técnicas tenha sido estudadas e apresentadas.

Na literatura, quatro tipos de sistemas de recomendações são recorrentes: baseado em conteúdo, com filtro colaborativo, baseado em conhecimento e demográfica. O baseado em conteúdo busca recomendar itens semelhantes aos itens que usuário gostou anteriormente. Esse método é muito indicado para lidar em situações em que muitos itens não foram avaliados apesar de ter problemas em que os usuários fizeram poucas avaliações. Os sistemas com filtro

colaborativo buscam recomendar itens com base nas avaliações de usuário com gostos parecidos, esse tipo método se destaca pelo fato de que não é necessário utilizar a representação do item, porém, esses sistemas também apresentam problemas para recomendar itens para usuários com poucas avaliações. Os sistemas de recomendação baseado em conhecimentos buscam recomendar itens utilizando conjuntos de regras, já os sistemas de recomendação demográficas recomendam os mesmos itens para usuários com características demográficas semelhantes.

O tipo de sistema que será apresentado nesse trabalho é o baseado em conteúdo. Neste sistema, três componentes são utilizados no processo de recomendação: análise do conteúdo, aprendizado do perfil e a filtragem. A análise de conteúdo busca transformar um conjunto de informações de um item em uma representação vetorial, o aprendizado de perfil tem como proposito a criação de modelos que representam os gostos do usuário e a filtragem busca encontrar os itens que satisfação critérios relacionados a qualidade da recomendação para o usuário (LOPS; GEMMIS; SEMERARO, 2011).

Os sistemas de recomendações tradicionais levam em consideração apenas um critério no momento da recomendação, que é acurácia, ou seja, a chance do usuário gostar do item. Porém essa metodologia é limitada, já que ela não considera que o usuário pode utilizar mais que um critério ao fazer uma escolha(ADOMAVICIUS; MANOUSELIS; KWON, 2011). Muitos trabalhos já buscam fazer recomendações utilizando mais de um critérios, mas, grande parte busca agregar os critérios em um único critério (YAGER, 1988), (KAYMAK; LEMKE, 1994). Esse tipo de abordagem apresenta limitações, pois a composição pode ocasionar em perda de informações.

Neste trabalho foi definido um nova abordagem para sistemas de recomendações baseado em conteúdo, chamado de MOEA-RS, onde a filtragem é tratada como um problema multiobjetivo, que é um problema onde se deseja otimizar dois ou mais critérios. Para isso, dois outros critérios, além da acurácia, são considerados: diversidade e novidade. A diversidade está relacionado ao quanto os itens da listas de recomendação diferem entre si, o propósito do uso desse critério, é fazer com que a recomendações não sejam monotemáticas(HURLEY; ZHANG, 2011). A novidade é o quanto os itens recomendados são diferentes do que o usuário conhece(HURLEY; ZHANG, 2011). O problema foi modelado levando em consideração um tipo de item, que são os filmes, que foram escolhidos devido a complexidade envolvida na recomendações desses itens, já que pela grande quantidade de filmes disponíveis, a recomendação baseada apenas na acurácia é limitada.

Portanto, no problema de recomendação de filmes, a solução é um vetor que corresponde a concatenação de representações numéricas de cada característica de um filme, e o objetivo é encontrar uma solução que otimizem os três critérios definidos, como esses critérios são conflitantes, não existirá apenas um solução(ou filme) que é ótima, mas um conjunto de soluções, esse conjunto corresponderá a lista de recomendações que será fornecida como resultado para o usuário. Para encontrar esse conjunto de filmes, será utilizado um algoritmo evolucionário

multiobjetivo (MOEA, em inglês), que se destaca pela capacidade de gerar um conjunto de soluções próximas ao ótimo em apenas uma execução.

Sendo assim, no MOEA-RS, o componente de analise de conteúdo utilizará técnicas de extração de características para transformar as informações dos filmes(sinopse, avaliações, ano de estreia e etc) em uma representação numérica, o componente de aprendizado de perfil utilizará as avaliações dos usuários sobre os filmes e técnicas de aprendizagem de máquina para construir modelos que representem os gostos dos usuários e no componente de filtragem, como dito anteriormente, será utilizado um MOEA para encontrar os filmes que otimizem os critérios relacionados ao gosto do usuário(acurácia e diversidade) e ao histórico de filmes assistidos por ele(novidade).

Diversas de técnicas extração de caraterísticas e aprendizado de máquina podem ser utilizadas nos componentes de analise de conteúdo e construção de perfil do MOEA-RS, respectivamente, portanto, para definir quais técnicas serão utilizadas, uma série de experimentos serão realizadas. Por fim, o MOEA-RS será avaliado utilizando *dataset* tradicionais para recomendação de filmes e seus resultados serão comparados com sistemas de recomendação do estado da arte.

Portanto, esse trabalho busca, a partir dos experimentos, provar a seguinte hipótese: Tratar recomendação de itens baseado em conteúdo como um problema multiobjetivo é uma metodologia eficiente para gerar uma lista de recomendação que otimize diversas medidas.

Esse trabalho foi organizado da seguinte forma: na Capitulo 2 será apresentado a fundamentação teórica utilizada no desenvolvimento do trabalho, no Capitulo 3 é apresentado o método proposto no trabalho e no Capitulo 4 é apresentado os experimentos executados e os seus resultados e no Capitulo 5 é apresentada a conclusão do trabalho.

2
Fundamentação Teórica

Neste capitulo será apresentado a Fundamentação Teórica deste trabalho. O capitulo foi organizado da seguinte forma: na Seção 2.1 será descrito o que é um sistema de recomendação e sua principais técnicas, na Seção 2.2 é apresentado uma descrição das técnicas de Otimização Multiobjetivo, e na Seção 2.3 os trabalhos relacionados são descritos.

2.1 Sistemas de Recomendação

Sistemas de recomendação são ferramentas de softwares e técnicas que fornecem sugestões de itens para serem utilizados por um usuário (RICCI; ROKACH; SHAPIRA, 2011). Alguns exemplos dessa ferramenta são os sistemas de recomendação de vídeos, músicas e livros que são muito utilizados em *streamings* e *e-commerce*.

A principal motivação para uso dessa ferramenta é a grande quantidade de informação disponível para os usuários. Essa grande quantidade faz com que os usuários normalmente tenha uma alto número de itens(como filmes) para escolher, e isso pode ser positivo pelo fato de que o usuário vai ter muita liberdade para escolher. Porém, esse excesso de alternativas aumenta a expectativa do usuário sobre a sua escolha e isso acaba gerando uma paralisia e não sensação de liberdade, esse problema é conhecido como Paradoxo da Escolha (SCHWARTZ, 2004).

Para evitar esse problema, os sistemas de recomendação são utilizados, esses sistemas são utilizados como filtros que selecionam os itens que tenham uma maior chance de ser escolhido pelo usuário. Devido a isso, diversos benefícios podem ser identificados a partir do uso dessas ferramentas:

- Aumento das vendas,
- Maior diversidade nos itens vendidos,
- Aumento da satisfação dos usuários,

- Aumento da fidelidade dos usuários,
- Maior entendimento das preferências dos usuários.

Os sistema de recomendações são divididos em duas categorias principais: sistemas de recomendações personalizados e não personalizados(JAIN et al., 2015).

Sistemas de recomendação personalizados utilizam o histórico do usuário para recomendar itens cujo o usuário tenha mais chance de gostar, os sistemas não personalizados não levam em consideração o histórico do usuário para executar a recomendação, nele, apenas os itens mais bem avaliados são recomendados(KHATWANI; CHANDAK, 2016).

Os Sistemas de recomendação, normalmente, utilizam três tipos de informação para executar as recomendações(RICCI; ROKACH; SHAPIRA, 2011):

- Itens: São os objetos que são recomendados, podem ser caracterizados pela sua complexidade e seu valor ou utilidade. O valor de um item muitas vezes é representado como a relevância daquele item para o usuário, essa relevância leva em conta diversos fatores, como o gosto do usuário e o custo para obter esse item.
- Usuários: São aqueles cujo o sistema irá recomendar os itens, deve ser levado em conta o fato os usuários podem ter diferentes tipos e características. Os Sistemas Recomendação buscam explorar todas representações possíveis do usuário, desde uma simples listas de avaliação até complexas representações vetoriais.
- Transações: São interações entre o usuário e o sistema de recomendação. Um exemplo de transação é a avaliação de um usuário sobre um item, tem diversos tipos de avaliação, mas as comuns são as numéricas (ex: notas de filmes do IMDB), as ordinais (ex: "Concordo Fortemente", "Concordo", "Neutro", "Discordo", "Discordo Totalmente"), binárias (ex:"Gostei"e "Não gostei") e unárias(ex: Indicativo de que o usuário observou o item).

Existem quatro tipos de sistemas de recomendação: Filtro colaborativo, Baseado em Conhecimento, Demográfico e Baseado em Conteúdo. Nos sistemas de recomendação com Filtro Colaborativo, os itens são recomendados com base no gosto de um grupo de pessoas que fizeram boas avaliação de itens parecidos anteriormente. A premissa básica dessa técnica é que pessoas que gostaram da mesmas coisas no passado, também vão gostar de coisas parecidas no futuro(SHAH et al., 2017).

Nos sistemas Baseado em Conhecimento, as avaliações dos usuários sobre os itens não são as principais fontes de informação para a recomendação, e sim com base em medidas de similaridades que são obtidas a partir dos requisitos dos usuários e de descrições dos itens. O cálculo dessa medida é feito a partir do uso de bases de conhecimento que contém regras e funções para cálculo de similaridade (AGGARWAL et al., 2016). No sistemas de

recomendação baseado em informações demográficas, além da avaliação dos itens, as informação demográficas dos usuários são utilizadas para identificar usuários similares para auxiliar no processo de recomendação (AGGARWAL et al., 2016). A suposição desse método é que diferentes recomendações podem ser geradas para diferentes nichos demográficos (RICCI; ROKACH; SHAPIRA, 2011). Os Sistemas baseados em conteúdo buscam utilizar informações sobre os itens para recomendar. Neste trabalho, o sistema corresponde a um sistema baseado em conteúdo, esse tipo sistema será descrito na Seção 2.1.1.

2.1.1 Baseada em Conteúdo

Essa técnica busca analisar um conjunto de documentos ou/e descrições de itens previamente avaliado por um usuário, e construir um modelo dos interesses do usuário, baseado nas características dos objetos avaliados pelo usuário (LOPS; GEMMIS; SEMERARO, 2011). Sistemas de recomendação baseados em conteúdo, normalmente, são projetados para explorar cenários em que os itens podem ser descritos como um conjunto de atributos (AGGARWAL et al., 2016).

O processo de recomendação nessa técnica possui 3 componentes: análise do conteúdo, aprendizado do perfil e a filtragem (LOPS; GEMMIS; SEMERARO, 2011). O seu diagrama de fluxo pode ser visto na Figura 1.

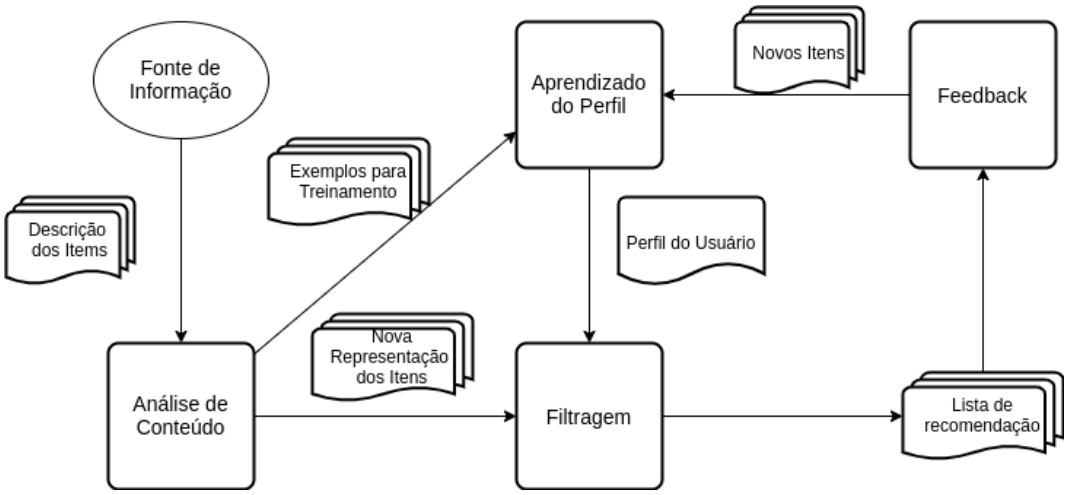

Figura 1 – Diagrama de fluxo de um sistema de recomendação baseado em conteúdo.

2.1.1.1 Análise de Conteúdo

O processo de recomendação é inicializado com o componente de análise de conteúdo, que irá receber como entrada o conjunto de dados contendo as descrições dos itens que serão utilizados na recomendação. Neste componente, os itens são analisados utilizando extração de características, gerando uma nova representação deste item. Essa nova representação será utilizada como entrada dos outros dois componentes.

Normalmente, as informações dos itens não contém apenas dados numéricos, em muitos casos os itens contém muitas informações textuais. No caso dos filmes, por exemplo, apesar de existir características numéricas como ano de estreia e avaliação geral, a maior parte das características são textuais, como sinopse, titulo, elenco e etc.

Portanto, uma importante tarefa desse componente de Análise de Conteúdo, é o uso de técnicas para extração de características de textos. Existem diversas técnicas para extrair valores numéricos de textos, as mais populares são as seguintes:

- **TF-IDF**: Essa técnica busca transformar cada texto (também chamado de documento) em um vetor, onde cada posição desse vetor é associado a um termo e o seu valor corresponde a um peso(chamado de tf-idf) que equivale a importância deste termo para o documento(SCHÜTZE; MANNING; RAGHAVAN, 2008).

  Esse peso é uma composição de duas medidas: TF(*Term frequence*, em inglês) e IDF(*Inverse document frequency*, em inglês). TF é uma medida que corresponde a quantidade de ocorrências de um termo em um documento (SCHÜTZE; MANNING; RAGHAVAN, 2008), normalmente denotada por $tf_{t,d}$, onde t é o termo e d é o documento. Já o IDF é calculada com base em outra medida, df (*Document frequency*, em inglês), que corresponde a quantidade documentos que contém um termo. A equação do IDF é dada da seguinte forma:

  $$idf_t = \log(\frac{N}{df}) \tag{2.1}$$

  Onde N é o número total de documentos. Logo, pode-se afirmar que esta medida dá um maior grau de importância para termos que aparecem em poucos documentos. Então, a o valor do tf-idf é dado da seguinte forma:

  $$\text{tf-idf}_{t,d} = tf_{t,d} \times idf_t \tag{2.2}$$

  A partir dessa equação, três princípios básicos desse medida podem observados. O primeiro é que termos que ocorrem muito em um documento e aparecem em poucos documentos, são muito importantes para aquele documento. O segundo é que termos que aparecem em muitos documentos ou que aparece poucas vezes em um documento, tem pouca importância. O terceiro é que se o termo aparece em todos os documentos ele vai ser pouco relevante para todos os documentos.

- **Word2Vec**: É uma técnica que busca aprender representações vetoriais de palavras a partir de uma grande quantidade de dados textuais (MIKOLOV et al., 2013b), portanto, esse método não pode ser usado diretamente para gerar um vetor de um texto, para isso, pode ser utilizar técnicas de agregação de vetores, como a soma de vetores. Word2Vec corresponde a uma extensão do método Skip-Gram, esse método busca criar representações vetoriais de palavras que seja útil para prever palavras próximas em uma sentença ou documento (MIKOLOV et al., 2013a).

O objetivo do Skip-Gram, dado uma sequencia de palavras $w_1, w_2, w_3, ..., w_T$, é maximizar a probabilidade logarítmica média.

$$\frac{1}{T}\sum_{t=1}^{T}\sum_{-c\leq j\leq c, j\neq 0}\log p(w_{t+j}|w_t) \tag{2.3}$$

A principal diferença entre o Skip-gram e WordVec está na definição do $p(w_{t+j}|w_t)$, onde, no Skip-gram é uma função *softmax*, que é definido da seguinte forma:

$$p(w_O|w_I) = \frac{\exp({v'_{w_O}}^T v_{w_I})}{\sum_{w=1}^{W}\exp(v'^T_w v_{w_I})} \tag{2.4}$$

Onde v_w e v'_w são as representações vetoriais de entrada e saída da palavra w, respectivamente. No Word2Vec, duas definições para o calculo de $p(w_{t+j}|w_t)$, a primeira é utilizando o *softmax* hierárquico, que é uma forma mais eficiente de calcular o *softmax*.

No *softmax* hierárquico, é utilizado uma árvore binária para representar a distribuição de probabilidade, onde cada nó contém a probabilidade dos nós filhos e os nos folhas representam as palavras do dicionário (MIKOLOV et al., 2013b). O calculo do *softmax* hierárquico é definido da seguinte forma:

$$p(w|w_I) = \prod_{j=1}^{L(w)-1}\sigma([[n(w,j+1) = ch(n(w,j))]]\cdot {v'_{n(w,j)}}^T v_{w_I}) \tag{2.5}$$

Onde $n(w, j)$ é o j-ésimo nó do caminho entre a raiz e w, $L(w)$ é o tamanho deste caminho, $ch(n)$ é um nó filho arbitrário, $[[x]]$ é um função que retorna 1 caso x seja verdadeiro e -1 caso contrário, e $\sigma(x)$ é a função sigmoid.

A outra forma que o Word2Vec utiliza para calcular o $p(w_{t+j}|w_t)$ é a partir de Amostragem Negativa (MIKOLOV et al., 2013b), que é definida da seguinte forma:

$$p(w|w_I) = \log\sigma({v'_{w_O}}^T v_{w_I}) + \sum_{i=1}^{k}\mathop{\mathbb{E}}_{w_i\sim P_n(w)}[\log\sigma(-{v'_{w_i}}^T v_{w_I})] \tag{2.6}$$

- **Doc2Vec**: Essa técnica busca criar representações vetoriais para textos, ela é baseada em técnicas que aprendem representações de palavras para palavras próximas em uma frase ou texto (LE; MIKOLOV, 2014). Nesse método, para prever a próxima palavra em um texto, são utilizados os vetores das palavras anteriores a esta e o vetor que representa os documentos, para isso os vetores são agregado utilizando concatenação ou agregação. O framework do Doc2Vec pode ser visto na Figura 2.

O analisador de conteúdo processará esses dados e, com as técnicas de extração de características, gerará um novo conjunto de dados que conterá as novas representações dos itens.

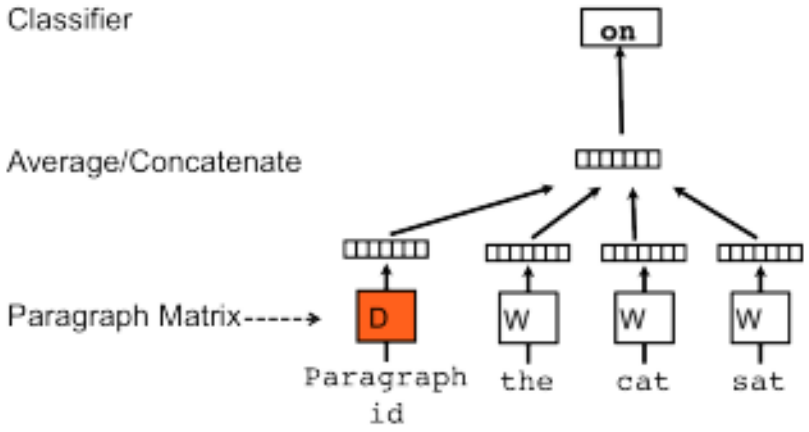

Figura 2 – Framework do Doc2Vec.

2.1.1.2 Aprendizado de Perfil

Após a criação do conjunto de dado contendo as novas representação dos itens, é iniciado o processo de coleta de dados sobre as avaliações feitas pelos usuários sobre os itens. As avaliações podem ser obtidas de algumas formas, como avaliações numéricas, feedback implícito(e.g. ações do usuário), opiniões textuais. Essas avaliações são convertidas em uma avaliação no formato numérico(AGGARWAL et al., 2016). Com isso, são construídos conjuntos de dados para cada usuário contendo as representações dos itens avaliados por esses usuários, cada item será rotulado pela avaliação do item feita pelo usuário. O componente de aprendizado de perfil receberá esses conjuntos de dados como entrada e criará um perfil para cada usuário, esse perfil representará os gostos do usuário.

O aprendizado de perfil coleta dados que representam as preferências do usuário, com esses dados, o componente tenta generalizar esses dados utilizando aprendizado de máquina para regressão, pois o valor que o modelo deve estimar é real. Existem diversas técnicas de aprendizado de máquina porém, as mais utilizadas são as seguintes:

- **Regressão Linear**: Esse é o modelo mais simples para regressão. Esse modelo, tem como proposito, modelar a relação entre as variáveis de entrada $X = x_1, x_2, ..., x_m$ e a variável alvo y com uma combinação linear, (BISHOP, 2006):

$$y = w_0 + w_1 * x_1 + w_2 * x_2 + ... + w_m * x_m = Xw \tag{2.7}$$

  Com isso, o modela tentar encontrar, ou aprender, os coeficientes $w = w_0, w_1, w_2, ..., w_m$, também chamados de pesos, para que minimize a seguinte função de custo, chamada de erro quadrático:

$$L(X, w) = ||Xw - y||_2^2 \tag{2.8}$$

  Uma forma utilizada para otimizar o vetor de pesos w é a partir do método de descida de gradiente estocástica, onde, dado um conjunto de vetores de entrada $X^{(1)}, X^{(2)}, ..., X^{(n)}$ e o

somatório da função de erro sobre o conjunto de vetores $E(w) = \sum_{i=1}^{n} L(X^{(i)}, w)$ e uma taxa de aprendizado α, é executado um processo iterativo, onde a cada iteração t o vetor de pesos é alterado da seguinte forma:

$$w^{(t+1)} = w^{(t)} + \alpha * \Delta E \tag{2.9}$$

- **Regressão Logística**: A regressão logística é uma extensão do regressão linear, onde a relação entre as variáveis não definidas por uma combinação linear, e sim, pela equação abaixo:

  $$y = \phi(Xw) \tag{2.10}$$

  Onde, ϕ é a função logística, definida da seguinte forma:

  $$\phi(a) = \frac{1}{1 + \exp(-a)} \tag{2.11}$$

- **Modelo Ridge**: Esse modelo é uma variação regressão linear que busca resolver uma dificuldade desse método, o *overfitting*, que é quando o modelo fica superespecializado nos dados de treinamento e não consegue generalizar para dados novos(GOODFELLOW; BENGIO; COURVILLE, 2016).

  Para isso o modelo utiliza uma penalidade na função de custo, para que vetores pesos com valores menores tenham preferência no processo de otimização(GOODFELLOW; BENGIO; COURVILLE, 2016), esse penalidade é chamada de regularização. No modelo Ridge é utilizado a norma l^2 como regularização (PEDREGOSA et al., 2011), assim a função de erro fica da seguinte forma:

  $$l(w) = ||y - Xw||_2^2 + \alpha * ||w||_2^2 \tag{2.12}$$

- **Modelo Lasso**: Esse modelo tem uma abordagem semelhante ao modelo Ridge. Porém, nesse modelo, a regularização é dada pela norma l^1 (PEDREGOSA et al., 2011), dessa forma a função de erro desse modelo é a seguinte:

  $$l(w) = ||y - Xw||_2^2 + \alpha * ||w||_1 s \tag{2.13}$$

- **Elastic Net**: Esse modelo corresponde a uma combinação dos modelos Ridge e Lasso, onde a regularização é dada pela soma ponderada das normas l^1 e l^2 (PEDREGOSA et al., 2011). A função de custo do Elastic Net pode ser vista na equação abaixo.

  $$l(w) = \frac{1}{2 * n} * ||y - Xw||_2^2 + \alpha * ||w||_1 + \beta * ||w||_2^2 \tag{2.14}$$

- ***Gradient Boosting Regressor*(GBR)**: Esse método busca construir um modelo a partir de um conjunto de modelos simples, usualmente os modelos utilizados são arvores de decisão, a construção do modelo é dada pela soma dos modelos mais simples (FRIEDMAN, 2002).

2.1.1.3 Filtragem

Dados os perfis gerados pelo componente de Aprendizado de Perfil, o componente de Filtragem tenta prever quais itens cada usuário poderá se interessar. Esse componente, normalmente, gera uma lista ordenada dos itens mais relevantes para o usuário(LOPS; GEMMIS; SEMERARO, 2011). Como o gosto do usuário tende a mudar com o passar do tempo, é necessário que todos os componentes sejam atualizados com base em *feedbacks* dos usuários sobre os itens recomendados.

A filtragem explora o perfil do usuário para sugerir itens relevantes, a relevância do item pode ser dada tanto pela chance do usuário gostar do item quanto por outros critérios relacionados a qualidade da recomendação. Dessa forma, a filtragem pode ser modelada como um problema de otimização multiobjetivo, em que o proposito é encontrar uma solução, que é uma representação numérica de um item(que neste trabalho é um filme), que otimizem os diversos critérios relacionadas a recomendação.

2.2 Otimização Multiobjetivo

Um problema de otimização multiobjetivo (MOP, do inglês *Multi-Objective Optimization*) é um problema onde o propósito é otimizar duas ou mais funções objetivos, podendo ser tanto a maximização dessas funções quanto a minimização(COELLO et al., 2007).

Uma solução de MOP minimiza (ou maximiza) os componentes de um vetor f(x) onde x é um vetor de variáveis de decisão n-dimensional, x = (x_1,..., x_n), a partir de um universo Ω. O universo Ω contém todo x possível que pode ser utilizado para satisfazer uma avaliação de f(x). Λ é o espaço do vetor das funções objetivo para o problema.

Neste tipo de problema as funções objetivos são conflitantes, logo não existe uma melhor solução, mas um conjunto com as melhores soluções. Para obter esse conjunto de soluções é utilizada a Teoria da Otimalidade de Pareto(COELLO et al., 2007). Em problemas multiobjetivo o ótimo é definido através dos termos a seguir (minimização).

- **Dominância de Pareto:** dadas duas soluções x e y, representadas pelos seus vetores de variáveis no espaço de objetivos, f(x) = (f_1(x),...,f_m(x)) e f(y)= (f_1(y),...,f_m(y)), respectivamente, dizemos que f(x) domina f(y) se, e somente se, f(x) é parcialmente inferior a f(y), ou seja, $\forall$ i $\in$ {1,...,m},$f(x)_i \leq f(y)_i \bigwedge \exists$ i $\in$ {1,...,m} tal $f(x)_i < f(y)_i$.Ou seja, uma solução domina outra, se ela é pelo menos igual em todas as funções objetivo e é obrigatoriamente melhor em pelo menos uma função objetivo.

- **Ótimo de Pareto:** A solução, f(x), é dito Ótimo de Pareto se o seu vetor de funções objetivo é não dominado, ou seja, não existe uma solução que pertença a Ω e domine f(y).

- **Conjunto Ótimo de Pareto:** para um MOP, o Conjunto Ótimo Pareto, P* , é o conjunto das melhores soluções em Ω, ou seja, é o conjunto de soluções do problema que são Ótimo Pareto.

- **Fronteira de Pareto:** cada solução em P* possui uma imagem de um ponto não dominado em Λ. O conjunto de todos os pontos não dominados no espaço das funções objetivo é chamado de fronteira de Pareto. Um exemplo de uma fronteira de Pareto pode ser visto na Figura 3, em que a fronteira está em destaque.

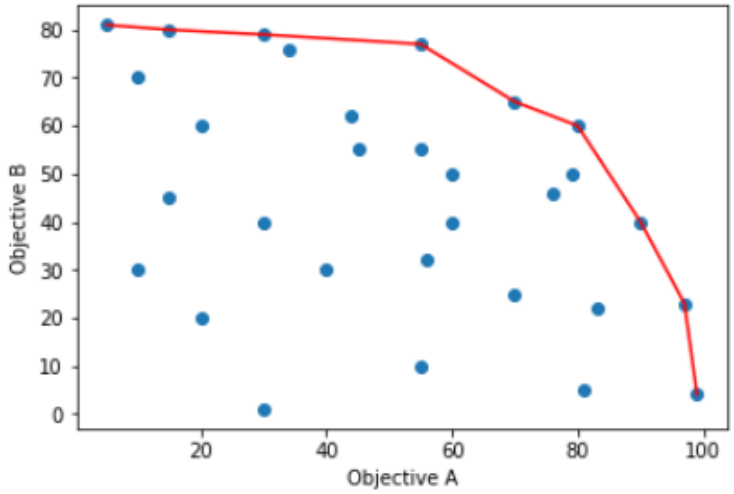

Figura 3 – Exemplo de uma Fronteira de Pareto.

Os MOPs são solucionados por diferentes áreas de pesquisa, dentre as quais destacam-se os Algoritmos Multiobjetivo Evolucionários(MOEA, do inglês *Multi-Objective Evolutionary Algorithm*). Em geral, os algoritmos evolucionários utilizam uma população de soluções na busca e assim permitem a geração de diversos elementos da fronteira de Pareto em uma só execução.

A área da Otimização Evolucionária Multiobjetivo visa a aplicação de MOEAs para a solução de MOPs (CARVALHO, 2013). Os MOEAs modificam os algoritmos evolucionários de duas maneiras: incorporam um mecanismo de seleção, geralmente baseado nos conceitos da Otimalidade de Pareto, e adotam mecanismos para a preservação da diversidade, para evitar a convergência para uma só solução. Como a maioria dos MOPs são problemas complexos, os MOEAs focam em determinar um conjunto de soluções mais próximo possível do Conjunto Ótimo de Pareto, denominado de conjunto de aproximação.

O funcionamento de um MOEA é dada da seguinte forma: inicialmente, uma população de indivíduos é gerada e seus indivíduos são avaliados, após isso, os indivíduos dominados são removidos da população, para manter a diversidade da população técnicas para estimar densidade são utilizadas, a partir da população resultantes, são aplicados os operadores evolucionários para gerar novos indivíduos, com essa nova população gerada, o operador de seleção é aplicado para

encontrar os melhores indivíduos e o processo é repetido até satisfazer as condições de parada (COELLO et al., 2007).

Um MOEA deve convergir para as melhores soluções do problema, porém deve cobrir de melhor forma as diferentes regiões para possibilitar a geração de melhor conjunto para o tomador de decisão (CARVALHO, 2013). Os MOEAs mais recorrentes na literatura são os seguintes:

- ***Multiobjective Genetic Algorithm*(MOGA)**: Nesse MOEA, os indivíduos são ordenados de acordo com a quantidade de cromossomos da população que são dominados pelo individuo (COELLO et al., 2007).
- ***Nondominated Sorting Genetic Algorithm*(NSGA)**:O NSGA separa os indivíduos da população em categorias que estão ligadas a relação de dominância dos indivíduos da categoria com o restante da população, onde, a primeira categoria contém apenas as soluções não dominadas. O valor de *fitness* de cada individuo corresponde a um valor relacionado a sua categoria, para que os indivíduos de uma mesma categoria tenha a mesmo potencial para reprodução (COELLO et al., 2007). Esse algoritmo tem duas variações: NSGA-II e NSGA-III. O NSGA-II, utiliza uma ordenação baseado em dominância mais eficiente e utiliza o elitismo como operador de seleção, esse operador seleciona os melhores indivíduos dentre a população atual e a nova população (DEB et al., 2002). O NSGA-III é uma modificação do NSGA-II que busca aumentar a diversidade da população a partir da escolha de indivíduos que estão mais próximos de um conjunto de pontos de referências (DEB; JAIN, 2013).
- ***Pareto Archived Evolution Strategy*(PAES)**: Esse algoritmo utiliza um arquivo para armazenar todas as soluções não dominadas encontradas durante as iterações. Esse arquivo é utilizado para comparar todos os indivíduos gerados por mutação (COELLO et al., 2007).

2.3 Trabalhos Relacionados

Na literatura científica, poucos estudos tratam da aplicação de algoritmos multiobjetivos em sistemas de recomendação. Nessa seção, serão descritos trabalhos que apresentaram sistemas de recomendação, para filmes, utilizando técnicas tradicionais(baseado em conteúdo e filtro colaborativo) e técnicas de otimização multiobjetivo.

2.3.1 Baseado em conteúdo

Em (CAMI; HASSANPOUR; MASHAYEKHI, 2017), é proposto um sistema de recomendação baseado em conteúdo que utiliza preferências temporais do usuário na construção do perfil. Portanto, nesse método o perfil do usuário corresponde a uma sequência de atividades do usuário, onde cada atividade indica qual item foi acessado pelo usuário e a data em que aquele item foi acessado.

A partir do perfil do usuário, é utilizado um *framework* Bayesiano não paramétrico é para modelar o gosto do usuário, esse *framework* contém três componentes: Extração de interesses, Inferência de preferências e Previsão. Neste trabalho, os componentes de extração de interesses e inferências de preferência são utilizados para modelar os interesses do usuário e o componente de previsão é utilizado para gerar a lista de recomendação.

O método foi avaliado utilizando o *dataset* MovieLens, e os seus resultados foram comparados com os resultado do sistema de recomendação Time-SVD++. A partir dos experimentos, foi observado que o método proposto obteve melhores valores de acurácia.

Em (HIMEL et al., 2017), um sistema de recomendação baseado em pesos é proposto. Esses pesos são valores associados aos filmes e são calculados de acordo com os dados relacionados às preferências do usuário. Os pesos correspondem a uma relação entre a quantidade de filmes que contêm uma característica, que faz parte da preferência do usuário, e a quantidade de filmes totais. Após o cálculo desses pesos, é criado um *dataset* de filmes com os pesos e o algoritmo de clusterização K-means é aplicado no *dataset*. Com o cálculo dos *clusters*, é escolhido o *cluster* com a maior média da soma dos pesos de cada filme, para ser recomendado para o usuário.

Para avaliar o algoritmo, os autores utilizaram o *dataset* do IMDB, para obter as características dos filmes e as avaliações dos usuários. Os resultados obtidos pelo método proposto foram comparados com os resultados obtidos por uma variação desse método sem o uso da clusterização. A partir dos experimentos, os autores identificaram que o método proposto obteve resultados melhores que a sua variação.

Em (YOON; LEE, 2018), é proposto um método baseado em conteúdo que utiliza uma técnica de *deep learning* para extrair informações de características textuais dos filmes. A técnica utilizada é a Word2Vec, que converte uma palavra em um vetor numérico, utilizando redes neurais. Nesse método, cada informação do filme(diretor, roteiristas e título) é convertida para uma forma vetorial e esses vetores são concatenados. O vetor resultante da concatenação é utilizado para calcular a similaridade dos filmes e encontrar os filmes mais próximos dos interesses do usuários.

Os autores utilizaram o *dataset* MovieLens para avaliar o método e seus resultados foram comparados com os resultados de outros dois métodos: SVD e o Item2Vec. Com base nos experimentos, os autores observaram que o métodos proposto obteve melhores resultados que os outros dois métodos.

2.3.2 Filtro Colaborativo

Em (KATARYA; VERMA, 2016), é apresentado um método que combina duas técnicas de filtro colaborativo, o método de cálculo de matriz de similaridade assimétrica utilizando fatoração de matriz e o método Tyco, que é um filtro colaborativo que utiliza tipicalidade cognitiva para reduzir problemas comuns de sistemas baseado em filtro colaborativo, como esparsidade e

baixa acurácia.

O método funciona da seguinte forma: primeiramente, o método Tyco é utilizado para dividir o *dataset* em *cluster* para tratar problemas com esparsidade, depois é calculado a matriz de similaridade assimétrica, então é aplicado o método de fatoração nessa matriz, e por fim, as avaliações são calculadas utilizando um método de regressão Linear. O *dataset* utilizando neste trabalho foi o MovieLens e o método foi comparado a um sistema tradicional baseado em Fatoração de Matriz. A partir dos experimentos, os autores concluíram que o método obteve melhores valores de acurácia que o método tradicional.

Em (UYANGODA; AHANGAMA; RANASINGHE, 2018), é proposto um sistema de recomendação com filtro colaborativo baseado no usuário. No sistema, diferentemente do métodos baseados no usuário tradicionais, o perfil do usuário corresponde a um conjunto de características, onde cada característica é um medida de gênero de filme relacionado às avaliações anteriores do usuário, e quantidade de filmes que usuário assistiu para cada gênero.

Os autores utilizaram o *dataset* Movielens para avaliar o desempenho do algoritmo e os resultados foram comparados aos resultados de um método tradicional. A partir dos experimentos, os autores observaram que o sistema proposto obteve melhores resultados que o método tradicional e que quando o número de avaliações do usuários eram menores a diferença entre os resultados aumentava.

2.3.3 Otimização Multiobjetivo

Em (IRFAN et al., 2015), é proposto um sistema de recomendação baseado em filtro colaborativo que utiliza algoritmos de otimização multiobjetivo para recomendações de locais. Esse *framework*, chamado de MobiContext, consiste de um sistema baseado em filtro colaborativo híbrido(uma combinação das abordagens baseadas em vizinhança e baseada em modelo). O *framework* busca recomendar uma lista de locais para um usuário com base em duas características: preferência do usuário pelos locais e distância do usuário aos locais. Para isso, os autores modelaram esse problema de recomendação como um problema multiobjetivo, onde os objetivos são as características e a solução é um vetor que corresponde a lista de locais.

O *framework* foi avaliado utilizando um *dataset* de *benchmark* relacionado a locais e seus resultados foram comparados a outros *framework*s da literatura. Os autores concluíram que o *framework* obteve melhores resultados que as outras abordagens associadas, mas que o MobiContext pode obter melhores resultados com a adição de técnicas de aprendizado máquina, mineração de texto e redes neurais.

Em (WANG et al., 2014), é proposto um algoritmo multiobjetivo baseado em decomposição para recomendação. Esse algoritmo busca resolver um problema onde a solução corresponde a uma lista de itens e os objetivos são: chance do usuário gostar dos itens e a popularidade desses itens. O algoritmo foi avaliado utilizando *datasets* de *benchmark* e comparado a algoritmos de

recomendação tradicionais. Os autores concluíram que os que o algoritmo fornece diversas boas alternativas de recomendações para um usuário e que o algoritmo é efetivo em recomendar itens novos e/ou impopulares.

Em (WANG et al., 2017) é proposto um sistema de recomendação híbrido, que combina as abordagens de filtro colaborativo baseado no usuário e baseado em item com uma abordagem de fatoração de matriz. O sistema recomenda uma lista de itens com base em duas características, a diversidade da lista e a acurácia.

Para isso, os autores propõem um método, onde primeiramente são gerados diversas listas de recomendação, que são geradas por três métodos de recomendação, e depois é utilizado um algoritmo multiobjetivo para encontrar um conjunto de listas de itens que otimizem esses objetivos. O algoritmo utilizado nesse trabalho foi o NSGA-II. O sistema foi avaliado utilizando o *dataset Movielens*, que é um *dataset* muito comum na área de sistemas de recomendação. A partir dos experimentos, os autores concluíram que o método proposto obteve melhores resultados que os outros métodos comparados, com relação à diversidade e acurácia da recomendação.

No trabalho (ZHENG; PU, 2018) é proposto um sistema de recomendação para problemas com mais de um *stakeholder*, um *stakeholder* é alguém que o sistema levará em consideração no momento da recomendação de um item.

No trabalho, foi utilizado um *dataset* de um site de encontros, nesse tipo de recomendação cada possível companheiro é um *stakeholders*. Os autores utilizaram quatro critérios para recomendação, que são, dados dois usuários u e v, a utilidade da recomendação do usuário v para o usuário u e a utilidade da recomendação do usuário u para o usuário u, a utilidade geral de u e v e precisão.

Utilizando esses critérios, o sistema foi modelado como um problema de otimização onde a solução do problema é uma lista de usuários. Para solucionar o problema, foi utilizado o algoritmo evolucionário multiobjetivo NSGA-II. O sistema foi comparado com outros sistemas de recomendação que foram aplicados ao mesmo problema, mas não utilizam um MOEA. Os autores concluíram que os MOEAs podem ser capazes de encontrar boas recomendações em sistemas de recomendações com múltiplos *stakeholders*.

Em (ZUO et al., 2015) é proposto um sistema de recomendação multiobjetivo com dois objetivos a se maximizar. O sistema busca recomendar uma lista de itens para um grupo de usuários com características parecidas com base em duas características, diversidade dos itens recomendados e acurácia da recomendação. Para isso, a solução do problema foi modelada com uma matriz em cada linha corresponde a um lista de itens que serão recomendados para um usuário do grupo. A otimização foi feita utilizando o MOEA NSGA-II.

O sistema foi avaliado utilizando o *dataset* de *benchmark Movielens*, e seu desempenho foi comparado com o desempenho de sistemas utilizados em trabalhos anteriores dos autores e sistemas do estado da arte. A partir dos experimentos, os autores concluíram que o método

proposto pode gerar recomendações diversificadas e com uma boa acurácia, mas a acurácia ainda pode ser melhorada.

Em (RIBEIRO et al., 2015), é apresentado uma abordagem de sistemas de recomendação multiobjetivo com filtro colaborativo que utiliza o conceito de eficiência de Pareto. que corresponde a uma ordenação onde os objetivos são conflitantes. O sistema busca encontrar uma lista de itens que otimizem três critério: acurácia, novidade e diversidade. O sistema foi avaliado utilizando o *dataset* de *benchmark Movielens*, seus resultados foram comparados com os resultados de outros métodos multiobjetivo. Os autores concluíram que o sistema pode ser efetivo em cenário onde é exigido um alto grau de acurácia, diversidade e novidade.

Em (OLIVEIRA et al., 2018), é proposto um sistema de recomendação com filtro colaborativo baseado otimização multiobjetivo, nesse trabalho é utilizado um algoritmo baseado SPEA2 para encontrar uma lista de itens que otimizem três objetivos: diversidade, acurácia e novidade. O método foi avaliado utilizando três *datasets*: *Movielens*, *Jester* e *Filmtrust*. Seus resultados foram comparados com outros sistemas do estado da arte. Os autores concluíram que o sistema foi capaz de gerar recomendações que equilibrassem os três objetivos.

Como pode ser visto, todos os trabalhos encontrados apresentaram uma característica em comum, que é a definição da solução do problema de recomendação como uma lista de itens. Esse ponto é onde a abordagem proposta neste Trabalho de Conclusão de Curso diverge dos trabalhos relacionados, pois, neste trabalho, o propósito é definir a solução como um item a ser recomendado e, com isso, explorar o poder dos MOEAs em gerar uma população de soluções diversificada.

Além disso, outra crítica que pode ser feita da metodologia que define a solução do problema como uma lista de itens, é que pode haver perda de informação devidos aos objetivos de uma lista de recomendação serem calculados como uma soma ponderada dos objetivos para cada item da lista.

As diferenças entre o método proposto nesse trabalho e os métodos relacionadas pode ser vista na Tabela 2.3.3. Além das diferenças com relação a modelagem do problema de recomendação como um problema multiobjetivo, outra diferença que é importante notar é o tipo de sistema de recomendação, já que, neste trabalho, o sistema de recomendação será baseado em conteúdo, diferentemente dos método que utilizam filtro colaborativo. O sistema baseado em conteúdo foi escolhido, pois queremos explorar as características dos itens ao fazer uma recomendação.

Tabela 1 – Diferenças entre o trabalho proposto e os trabalhos relacionados

Trabalho	Tipo de sistema	Solução do Problema	Número de objetivos
(IRFAN et al., 2015)	Filtro Colaborativo	Lista de itens	2
(WANG et al., 2014)	Filtro Colaborativo	Lista de itens	2
(WANG et al., 2017)	Filtro Colaborativo	Lista de itens	2
(ZHENG; PU, 2018)	Filtro Colaborativo	Lista de itens	2
(RIBEIRO et al., 2015)	Filtro Colaborativo	Lista de itens	3
(ZUO et al., 2015)	Filtro Colaborativo	Matriz de itens	2
(OLIVEIRA et al., 2018)	Filtro Colaborativo	Lista de itens	3
Trabalho Proposto	Baseado em conteúdo	Item	3

3
MOEA-RS

Neste capitulo será apresentado o método proposto nesse trabalho. Para isso o capitulo foi organizado da seguinte forma: Na Seção 3.1 será descrito o funcionamento geral do método, na Seção 3.2, serão apresentados os *datasets* de filmes que serão utilizados, na Seção 3.3 será descrito o componente de Análise de conteúdo, a Seção 3.4 descreverá o componente de aprendizado de perfil, e na Seção 3.5 será apresentado o componente de filtragem.

3.1 Funcionamento do MOEA-RS

O método proposto nesse trabalho consiste de um sistema de recomendação multiobjetivo com baseado em conteúdo para filmes. O sistema apresenta os mesmos componentes que um sistema de recomendação baseado em conteúdo tradicional, a principal diferença está na etapa de filtragem, onde, ao diferentemente dos sistemas tradicionais que utilizam abordagens comuns de filtragem, o MOEA-RS trata esse componente como problema de otimização multiobjetivo, sendo assim, o sistema utiliza um MOEA para buscar os itens(que nesse trabalho serão filmes) que otimizem os critérios definidos. O diagrama de fluxo pode ser visto na Figura 4.

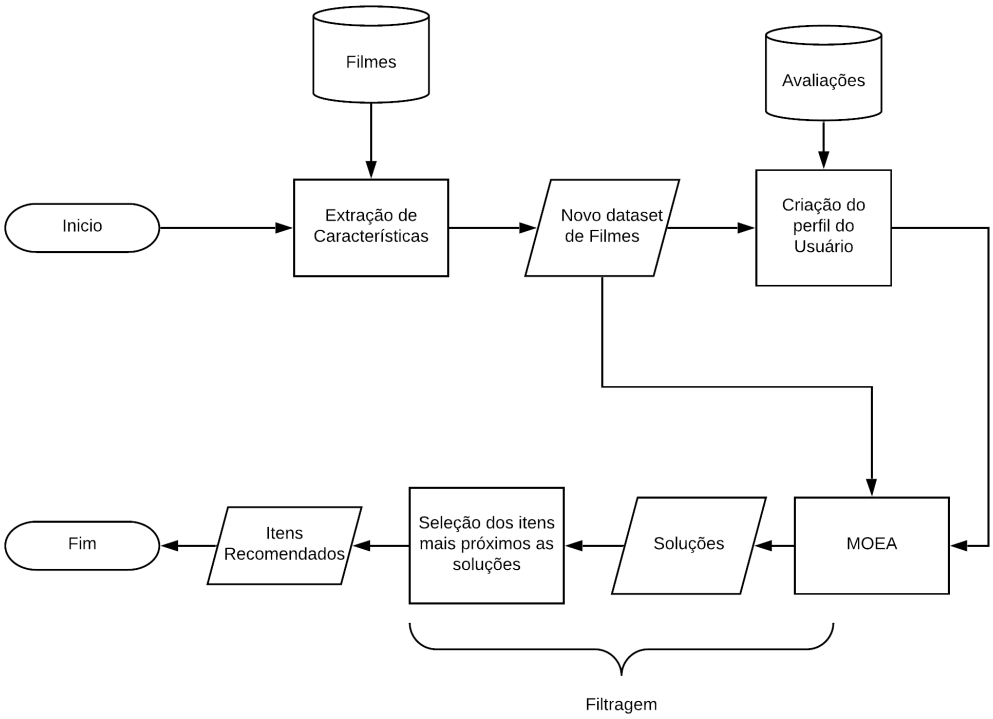

Figura 4 – Diagrama de fluxo do MOEA-RS

3.2 *Datasets* de Filmes

Esse método utilizará dois *datasets* para auxiliar o sistema de recomendação. O primeiro *dataset* é o MovieLens(HARPER; KONSTAN, 2016), que contém a avaliação de diversos usuários sobre filmes, essas avaliações foram obtidas a partir das interações desses usuários com a plataforma online do Movielens. O segundo *dataset* é o IMDB, que contém informações sobre diversos filmes, como sinopse, atores, diretores e etc (IMDB, 2019).

Como o sistema de recomendação proposto é um sistema baseado em conteúdo, o *dataset* com as avaliações será divididos em diversos *datasets*, onde cada *dataset* resultante terá as avaliações de apenas um usuário.

3.3 Análise de conteúdo

Esse componente é responsável por transformar a representação de um filme em um vetor de características. No IMDB, os filmes são representados por diversos tipos de atributos, porém, boa parte dos atributos(*e.g.*, sinopse, gêneros, titulo, diretores, etc), são textuais e

categóricos, portanto, o analisador de conteúdo terá que ser capaz de transformar informação textuais e categóricas em um vetor de valores numéricos. Nesse componente, diversas abordagens de extração de características podem ser utilizadas, contanto que a abordagem converta a característica do filme em um vetor numérico.

Portanto, o resultado do analisador de conteúdo corresponde a um novo *dataset*, que conterá as novas representações dos filmes, essa representação corresponderá a uma concatenação dos vetores numéricos que representam cada característica do filme. Esse *dataset* será utilizado componente de construção do perfil dos usuários e na filtragem.

A abordagem de análise de conteúdo escolhida nesse trabalho será definida nos experimentos, onde serão comparadas diversas abordagens. Essas abordagens correspondem a combinação de uma representação vetorial da sinopse do filme a partir de técnicas de extração de características de textos(que nesse trabalho, duas técnicas serão comparadas: Word2Vec e TF-IDF), com algumas características do filme.

3.4 Construção do perfil usuário

Esse componente recebe como entrada um conjunto de *datasets*, onde cada *dataset* corresponde às avaliações de um usuário sobre diversos filmes, além do *dataset* resultante análise de conteúdo. Como resultado, o componente irá construir um modelo de regressão $R_u(i)$ para cada usuário u, que representará a chance de um usuário gostar de um filme i. Esses modelos serão utilizados no componente de filtragem. Diversos modelos de aprendizado de máquina podem ser utilizados nesse componente, nesse trabalho, o modelo utilizado será definido a partir dos experimentos. Os modelos comparados nos experimentos serão o Ridge, Lasso, ElasticNet e GBR.

3.5 Filtragem

Para executar a filtragem, o componente responsável pela recomendação dos filmes, a recomendação será modelada como um problema multiobjetivo, sendo que os filmes recomendados serão aquele que otimizem os critérios definidos.

3.5.1 Problema de Recomendação

O sistema de recomendação proposto neste trabalho busca recomendar filmes com três objetivos: estimativa da avaliação do usuário sobre o filme (também chamada de acurácia), novidade e diversidade.

A estimativa da avaliação do usuário é obtida a partir do modelo gerado no componente

de aprendizado de perfil, assim, o objetivo é dada pela função:

$$F1(i) = R^*(u, i), \tag{3.1}$$

onde $R*$ é a estimativa de avaliação do usuário u sobre o filme i. A novidade de um item é dado pela seguinte equação (ZHANG, 2013):

$$F2(i) = \min_{j \in A_u}(1 - similaridade(i, j)), \tag{3.2}$$

sendo A_u o conjunto de itens que foram avaliados pelo usuário u, portanto, essa função objetivo pode ser interpretada como a similaridade entre o item recomendado e os itens que já foram avaliados pelo usuário e a função $similaridade(i, j)$ é um mediada de calcula a similaridade entre vetores, nesse trabalho a medida utilizada foi a similaridade do cosseno, que é dada da seguinte forma.

$$similaridade(i, j) = \frac{\langle i, j \rangle}{|i||j|} \tag{3.3}$$

A terceira função - a diversidade - corresponde a uma função que busca relacionar um item recomendado ao outros itens da lista de recomendação, essa função calcula o quão diferente é o item em comparação aos itens da lista. O propósito dessa medida é garantir que a lista de recomendação seja mais heterogênea. A diversidade de um item em comparação aos outros itens da lista pode ser calculada da seguinte forma:

$$F3(i) = \frac{1}{|P - i|} \sum_{j \in P - i} (1 - similaridade(i, j)), \tag{3.4}$$

onde P é o conjunto de itens a serem recomendados, que nesse método será a população gerada pelo MOEA.

Portanto a recomendação pode ser modelado como um problema de otimização multiobjetivo(minimização), onde, dado um usuário u, a solução corresponde a representação vetorial de um filme i e o propósito do problema é encontrar uma solução que otimize as seguintes funções:

$$\begin{cases} \min -F1(i) = R^*(u, i) \\ \min -F2(i) = \min_{j \in A_u}(1 - similaridade(i, j)) \\ \min -F3(i) = \frac{1}{|P|-1} \sum_{j \in P-i}(1 - similaridade(i, j)) \end{cases}$$

Para isso, a solução do problema foi definida como um vetor que representará o item, onde cada elemento desse vetor corresponde a uma característica do item.

3.5.2 Processo de Recomendação

A recomendação é feita a partir da execução de um MOEA para solucionar o problema descrito na seção anterior. Vale ressaltar que o MOEA não vai executar uma busca dentro do

conjunto dos filmes, a busca será feita em um conjunto S onde, sendo $s = [s_1, s_2, ..., s_m]$, se $s \in S$, então $l_i < s_i < u_i$, onde l_i e u_i é o menor e o maior valor possível da variável i da representação vetorial dos filmes do *dataset*, respectivamente, e m é o numero de variáveis. Em outras palavras, o espaço de busca do MOEA contém o espaço de filmes, mas, diferentemente do espaço de filmes, esse espaço de busca é continuo.

O motivo para que a busca não seja feita no espaço de filmes é que cada solução, gerada durante a execução do MOEA, teria que ser comparada aos filmes do *dataset*, o que aumentaria a complexidade, e dependendo do tamanho do *dataset*, a execução do algoritmo se tornaria inviável.

Como a busca é feita em um espaço maior que o espaço de filmes, não há garantia de que as soluções pertencentes a população resultante do MOEA são filmes do *dataset*. Assim, os filmes recomendados não serão os individuos da população resultante, e sim, os itens do *dataset* com maior similaridade a esses indivíduos. Logo, sendo $P = [p_1, p_2, ..., p_k]$ a população resultante do MOEA e D o conjunto de itens do *dataset*, a lista de recomendação foi definida como $Q = [q_1, q_2, ..., q_k]$, onde q_i é dado pela equação 3.5. Caso o filme mais similar a solução já tenha sido adicionada a lista de recomendação, o próximo filme mais similar é o escolhido.

$$q_i = \arg\min_{j \in D}(1 - similaridade(j, p_i)) \tag{3.5}$$

Um exemplo de recomendação pode ser vista na Figura 5, onde, na imagem mais a esquerda, estão os filmes de um *dataset*(em verde) fictício com uma representação vetorial de duas dimensões e as soluções geradas pelo MOEA(em vermelho), e na imagem da direita está o resultado da recomendação com filmes que estavam mais próximos de cada solução.

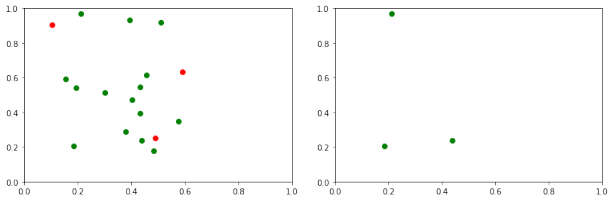

Figura 5 – Conjunto de filmes e as soluções geradas pelo MOEA (esquerda) e o resultado da recomendação (direita).

4
Experimentos

Para avaliar o método proposto neste trabalho, foram planejados uma série de experimentos, com intuito de escolher técnicas com os melhores desempenhos para serem agregadas ao método. Por fim, o método será comparado a outras técnicas tradicionais para recomendações de filmes.

Para isso, esse capitulo está organizado da seguinte forma: na Seção 1 será apresentada metodologia dos experimentos, na Seção 2 serão descritos os *datasets* utilizados nos experimentos, na Seção 3 serão apresentadas as medidas de avaliação, na Seção 4 serão apresentados os experimentos para encontrar a melhor modelagem dos conteúdos dos filmes para serem utilizados no componente de análise de conteúdo e também os melhores modelos de aprendizagem de máquina para criação do perfil do usuário, na Seção 5 serão descritos os algoritmos do estado da arte utilizados para comparação e na Seção 6 serão apresentados os resultados.

4.1 Metodologia dos experimentos

Neste trabalho, os experimentos foram projetados com o propósito de responder 2 questões de pesquisa:

- Qual a melhor representação de um filme para ser utilizada no processo de recomendação?
- Modelar um sistema de recomendação baseado em conteúdo como um problema multi-objetivo pode gerar melhores recomendações com relação a medidas de qualidade como acurácia, diversidade e novidade?

Para responder essas perguntas, foram planejados uma série de experimentos, que inicialmente identificaram as melhores técnicas para análise de conteúdo e construção de perfil, depois essas técnicas serão comparadas com relação ao desempenho do MOEA-RS ao utilizar cada técnica. Após esses experimentos, o MOEA-RS, junto com as técnicas escolhidas, será comparado com técnicas tradicionais de recomendação e com um trabalho relacionado.

4.2 *Datasets*

Para avaliar o método, dois *datasets* serão utilizados. O primeiro conterá as informações dos filmes, esse *dataset* foi obtido da base de dados do IMDB, este *dataset*, possui diversas informações sobre 3706 filmes, porém, algumas características não estão presentes em todos os filmes, na Tabela 2, pode ser visto a quantidade de filmes que não possuem certas características.

Característica	Filmes sem a característica	%
Titulo	2	0.007
Sinopses	366	1.34
Elenco	310	1.13
Gêneros	15	0.05
Diretores	280	1.02
Ano	2	0.007
Roteiristas	1328	4.88
Duração	74	0.27

Tabela 2 – Quantidade de filmes que não possuem determinada características

Portanto, nos experimentos foi considerado que, ao escolher certas características para o componente de análise de conteúdo, uma parte dos filmes será removida, porém, como pode ser visto na tabela, apenas uma pequena porcentagem dos filmes não contém as características.

O segundo *dataset* utilizado nesses experimentos foi o dataste MovieLens 1M, esse *dataset* contém 1 milhão de avaliações de usuário sobre filmes. Para executar os experimentos, apenas os usuários que avaliaram pelo menos 100 filmes foram considerados. As informações do *dataset* utilizado nos experimentos pode ser visto na Tabela 3.

Informação	Valor
Avaliações	1000209
Usuários	6040
Filmes	3706
Maior número de avaliações	2314
Menor número de avaliações	20
Média de avaliações	165

Tabela 3 – Informações sobre o *dataset* de avaliações utilizado nos experimentos

4.3 Medidas de avaliação

Para avaliar os resultados obtidos, serão utilizadas 4 medidas: Precisão, *Recall*, Novidade e Diversidade. As duas primeiras medidas são utilizados em problemas de classificação binária, portanto, para utilizar essas medidas, foi definido que se o usuário deu uma nota acima de 3(em uma escala de 1 a 5), o filme é relevante para o usuário, essas medidas foram escolhidas pois

o propósito desses experimentos é avaliar as listas de recomendações geradas pelo sistema de recomendação. As medidas de *Recall* e Novidade serão utilizadas pois, são medidas utilizadas pelo MOEA-RS.

Precisão é uma medida que busca determinar a porcentagem de itens recomendados corretamente (OLSON; DELEN, 2008). O cálculo dessa função é dado pela Equação 4.1:

$$Pr = \frac{tp}{tp + fp} \tag{4.1}$$

Onde tp é o número de recomendações relevantes, também chamado de verdadeiro positivo e fp é o número de recomendações erradas, também chamado de falso positivo.

Recall é uma medida que representa a fração de itens relevantes que foram recomendados (OLSON; DELEN, 2008). O cálculo dessa função é dado pela Equação 4.2:

$$Re = \frac{tp}{tp + fn} \tag{4.2}$$

Onde fn é o número de itens relevantes que não foram recomendados para o usuário.

4.4 Análise de Conteúdo e Construção de Perfil

Para definir os métodos que serão utilizados para análise de conteúdo e construção, foram projetados uma série de experimentos, com o proposito de avaliar diversas abordagens diferentes. Para avaliar todas essas abordagens, foi utilizado uma pequena parcela do *dataset* original, essa parcela foi escolhido, para viabilizar os experimentos, pois como serão testados diferentes variações métodos, poderia se tornar inviável a realização desses experimentos com o *dataset* completo. As informações desse *dataset* podem ser vistas na Tabela 4.

Informação	Valor
Avaliações	6761
Usuários	25
Filmes	2014
Maior número de avaliações	1220
Menor número de avaliações	120
Média de avaliações	270,4

Tabela 4 – Informações sobre o *dataset* de avaliações utilizado nos experimentos

4.4.1 Análise de Conteúdo

Para a Análise de conteúdo, os métodos de extração de características serão utilizados para criar um nova representação para a sinopse dos filmes, como alguns filmes tem mais de uma sinopse, esses textos serão concatenados. Com relação às características categóricas dos filmes,

como foi descrito na definição método, será criado uma coluna para categoria da característica. As características Gênero, Avaliação média, Duração e Ano foram utilizadas nos experimentos.

Para extração de características dois métodos serão comparados: TF-IDF e Word2vec. Para cada método serão propostos um conjunto de variações com relação aos parâmetros do modelo e as características dos filme que serão considerados na construção do *dataset*. Nas subseções abaixo, serão apresentadas as variações do métodos de extração de características.

Para cada variação de método, serão criada outras variações com uma combinação de variáveis categóricas e numéricas diferentes. Na Tabela 5 podem ser vistas todas as combinações possíveis.

Combinação	Gênero	Avaliação média	Duração	Ano
1	-	-	-	-
2	-	-	-	X
3	-	-	X	-
4	-	-	X	X
5	-	X	-	-
6	-	X	-	X
7	-	X	X	-
8	-	X	X	X
9	X	-	-	-
10	X	-	-	X
11	X	-	X	-
12	X	-	X	X
13	X	X	-	-
14	X	X	-	X
15	X	X	X	-
16	X	X	X	X

Tabela 5 – Combinação de Características

4.4.1.1 TF-IDF

Para o TF-IDF, as variações serão dadas a partir da alteração da quantidade máxima de termos, todas as variações podem ser vistas na Tabela 6:

Variação	Número de termos
TFIDF-1000	1000
TFIDF-1500	1500
TFIDF-2000	2000

Tabela 6 – Variações do TF-IDF

Todas essas variações utilizaram um mesmo de conjunto de parâmetros, esses valores podem ser vistos na Tabela 7.

Parâmetro	Valor
Norma	l2
Maior n-gram	3
Menor n-gram	1

Tabela 7 – Parâmetros fixos do TF-IDF

4.4.1.2 Word2Vec

Para o Word2Vec, as variações serão dadas a partir da alteração da dimensão do vetor que representa um termo, todas as variações podem ser vistas na Tabela 8:

Variação	Número de Variáveis
W2V-150	150
W2V-200	200
W2V-250	250
W2V-300	300

Tabela 8 – Variações do Word2Vec

Todas essas variações utilizaram um mesmo de conjunto de parâmetros, esses valores podem ser vistos na Tabela 9.

Parâmetro	Valor
Janela	3
Exemplos negativos	20
Amostragem	6e-5

Tabela 9 – Parâmetros fixos do Word2Vec

4.4.2 Construção do Perfil

Na construção do perfil, quatro modelos de aprendizado de máquina serão comparados: Ridge, Kernel Ridge, Elastic Net e Gradient Boost. Para cada método, serão avaliados diversas combinações de parâmetros, utilizando General Cross Validation, que busca encontrar a melhor combinação de parâmetros para um modelo de aprendizado de máquina, a partir da avaliação do modelo para diferente divisões do *dataset* (KOHAVI et al., 1995).

4.4.3 Resultados

Para avaliar os métodos de extração de característica e construção do perfil, os dataset de avaliações dos usuários, descrito na Tabela 4, serão divididos, onde 70% do dataset será utilizado para treinamento e 30% para teste. Assim, a partir desses *datasets*, serão construídos novos *dataset* utilizando os métodos de extração de características.

Com a nova representação dos datasets, os métodos de aprendizado de máquinas serão utilizados para construir o perfil do usuário, utilizando o conjunto de treino. Com isso, esses métodos serão avaliados, no dataset de teste, utilizando as medidas de Precisão e *Recall*. Os métodos que obtiverem os melhores resultados serão novamente avaliados com relação ao desempenho do MOEA-RS utilizando cada método.

No Apêndice A são apresentadas as tabelas com os resultados da medida de precisão e *recall* obtidas de cada variação dos Word2Vec e TFID com relação aos modelos de aprendizado de máquina utilizados para a construção de perfil, nas tabelas os melhores resultados estão destacados em negrito.

Dado esses resultados, 3 métodos foram selecionados para serem avaliados na próxima etapa de experimentos. O primeiro critério para escolha desses métodos foi o valor da precisão dos seus resultados, a medida de *recall* foi considerada como um critério de desempate. Na Tabela 10, pode ser vista as variações selecionados, com a quantidade de variáveis de cada representação e o seus valores de precisão e *recall*.

Variação	Modelo	Número de Variáveis	Precisão	*Recall*
w2v-150-10	Ridge	152	0.92343	0.95990
w2v-250-9	Ridge	252	0.92276	0.96378
w2v-250-13	GBR	2954	0.92227	0.96444

Tabela 10 – Variações selecionados para serem avaliados em conjunto com o MOEA-RS

As variações serão avaliadas em duas instâncias do MOEA-RS, que diferem apenas na quantidade máxima de iterações do NSGA-III, essas instâncias serão chamadas de MOEA-RS-200 e MOEA-RS-300, onde, o NSGA-III terá no máximo 200 iterações no MOEA-RS-200 e 300 iterações no MOEA-RS-300.

Para as duas instâncias do MOEA-RS, o NSGA-III, com exceção da quantidade máxima de iterações, foi parametrizado da mesma forma, com população de tamanho 16, o *crossover* sendo calculado utilizando o método SBX (DEB; SINDHYA; OKABE, 2007) com um probabilidade de *crossover* em 0.9, a mutação sendo calculada com a técnica de Mutação Polinomial (DEB; DEB, 2014) com taxa de 1/n, onde n é o tamanho do individuo, que é o número de variáveis da representação gerada no aprendizado de perfil.

Na Tabela 11 são apresentados os resultados de Precisão, *Recall*, Diversidade e Novidade para as recomendações de cada variação. Como pode ser visto, o variação do MOEA-RS com 200 iterações, que utiliza o w2v-250-9 para análise de conteúdo e o modelo Ridge para construção de perfil, obteve o melhor resultado de precisão, mas teve um resultado pior para a medida de diversidade em comparação com os outros métodos.

O método que obteve altos valores das medidas de diversidade e novidade foi o MOEA-RS-300 em conjunto com o w2v-250-13, mas esse método obteve um valor baixo de precisão

e *recall*. O outro método que também obteve valores altos de diversidade e novidade mas que teve um valor alto de precisão e *recall* foi o MOEA-RS-300 em conjunto com o w2v-150-10. A diferença entre o valor de precisão e de *recall* entre os métodos pode ter sido ocasionado pelo desempenho do MOEA no processo de otimização, pois o w2v-250-13 utiliza os genêros dos filmes para gerar a representação, e essa característica é convertido em um vetor em que variável só pode assumir dois valores, esse afeta o desempenho do MOEA, pois foram utilizados operadores evolucionários para problemas cujo a solução é um vetor com valores pertencentes ao conjuntos do reais.

Características	MOEA-RS	Modelo	Precisão	*Recall*	Diversidade	Novidade
w2v-150-10	MOEA-RS-200	Ridge	0.95447	0.135454	0.176556	0.050551
w2v-150-10	MOEA-RS-300	Ridge	0.94859	0.128578	0.181483	0.050680
w2v-250-9	MOEA-RS-200	Ridge	0.94409	0.12679	**0.24496**	0.03137
w2v-250-9	MOEA-RS-300	Ridge	**0.97177**	0.12763	0.18501	0.03096
w2v-250-13	MOEA-RS-200	GBR	0.92000	**0.024781**	0.221339	**0.152953**
w2v-250-13	MOEA-RS-300	GBR	0.88000	0.02200	0.14000	0.13945

Tabela 11 – Variações selecionados para serem avaliados em conjunto com o MOEA-RS

O outro método que obteve altos valores para todas as medidas foi o w2v-150-10 em conjunto com MOEA-RS-200 e o w2v-150-10 em conjunto com MOEA-RS-300, porém, como os dois métodos que se destacaram tiveram valores muito próximos, foi utilizado como critérios principais as medidas de diversidade e *recall*. Portanto, a variação selecionada para execução dos experimentos foi o w2v-150-10 em conjunto com o MOEA-RS-300 0para análise de conteúdo junto com o modelo *Ridge* para construção de perfil.

4.5 Algoritmos para comparação

Os resultados obtidos pelo método proposto serão comparados aos resultados de outros três sistemas de recomendações: baseado em conteúdo tradicional, filtro colaborativo tradicional e filtro colaborativo com otimização objetivo.

O sistema de recomendação baseado em conteúdo utilizado foi o Sistema baseado em conteúdo auxiliado pelo K-means, o método com filtro colaborativo utilizado foi o sistema de recomendação baseado em itens e a técnica baseado em filtro colaborativo com otimização multiobjetivo utilizada foi o método MOEA/D-RS.

4.6 Resultados

Nesta seção serão apresentados os resultados dos experimentos que buscam avaliar o MOEA-RS para recomendação de filmes, utilizando o *dataset* descrito na Tabela 3, em

comparação com um trabalho relacionado e com sistemas de recomendações tradicionais. Os resultados podem ser visualizados na Tabela 12, na Tabela, para cada medida de avaliação, são apresentados o valores médios de todas as recomendações feitas por cada método, o desvio padrão, o maior(e o menor) valor obtido por cada método.

Método	Medida	Precisão	*Recall*	Diversidade	Novidade
MOEA-RS	Média	**0.949282**	0.141767	**0.201282**	**6.2386e-02**
	Desvio Padrão	0.105653	0.097391	0.156934	6.3856e-02
	Mínimo	0.000000	0.000000	0.000000	2.4023e-07
	Máximo	1.000000	0.900000	1.010196	1.4323e-01
CF	Média	0.941084	**0.274845**	0.072433	6.2384e-02
	Desvio Padrão	0.081718	0.138983	0.004091	6.3928e-02
	Mínimo	0.266667	0.025467	0.071429	2.8306e-07
	Máximo	1.000000	1.000000	0.097844	1.4316e-01
CB K-Means	Média	0.896302	0.234154	0.111522	6.1519e-02
	Desvio Padrão	0.143168	0.147292	0.161155	6.1829e-02
	Mínimo	0.000000	0.000000	0.000000	2.5439e-07
	Máximo	1.000000	0.833333	1.004367	1.4326e-01
MOEA-D/RS	Média	0.828279	0.194398	0.072255	6.2214e-02
	Desvio Padrão	0.167902	0.093568	0.003956	6.3908e-02
	Mínimo	0.000000	0.000000	0.071429	2.5417e-07
	Máximo	1.000000	0.625000	0.131161	1.4315e-01

Tabela 12 – Variações selecionados para serem avaliados em conjunto com o MOEA-RS

Como pode ser visto na Tabela, o MOEA-RS superou o método com proposta relacionada (MOEA-D/RS) em todos os aspectos, com destaque para as medidas de precisão e diversidade, onde o MOEA-RS apresentou uma melhora de 14%, para a medida de precisão, e 178% para a medida de diversidade.

Com relação aos métodos tradicionais(CF baseado em itens e CB-Kmeans), o MOEA-RS obteve melhores resultados para as medidas de precisão, diversidade e novidade, com destaque para medida de precisão, em que o MOEA-RS apresentou uma melhora de 0.8% com relação ao CF baseado em itens, e para a medida de diversidade, que o MOEA-RS apresentou uma melhora de 80% com relação ao CB K-Means.

Outro fator que pode ser analisado nos resultados é a relação entre a quantidade de filmes utilizados no processo de recomendação e a precisão da recomendação, esse fator é importante para identificar se o MOEA-RS apresenta uma dificuldade que é comum em sistemas de recomendação, que é a baixa precisão na recomendações de itens para usuários com poucas avaliações. Na Tabela 13, pode ser visto a relação entre a quantidade de filmes e a precisão.

Quantidade de filmes	Número de usuários	MOEA-RS	CF-Item
Entre 70 e 100	719	0.942672	0.925359
Entre 100 e 150	725	0.946123	0.938851
Entre 150 e 200	417	0.949416	0.941966
Entre 200 e 300	504	0.959696	0.953704
Entre 300 e 400	236	0.949976	0.951130
Entre 400 e 800	256	0.950488	0.954167
Entre 800 e 1613	26	0.994505	0.969231

Tabela 13 – Relação entre a quantidade de filmes utilizados no processo de recomendação e a precisão da recomendação.

Como pode ser visto na Tabela acima, a quantidade de filmes avaliados pelos usuários influencia na precisão do MOEA-RS, mas a precisão recomendações para usuários com poucas avaliações foram maiores do que o CF baseado em itens.

5
Conclusão

Neste trabalho foi apresentado o MOEA-RS, que é um sistema de recomendação para filmes baseado em conteúdo auxiliado por um algoritmo de otimização multiobjetivo que tem como propósito encontrar filmes que otimizem três medidas, acurácia, diversidade e novidade.

Para definir as técnicas que seriam utilizadas nos componentes do método, foram realizados uma série de experimentos com o propósito de comparar diversas técnicas de aprendizado de máquina e extração de características. Após a definição das técnicas, o método foi avaliado no *dataset* MovieLens, com mais de 1 milhão de avaliações de usuários sobre filmes. Os resultados do MOEA-RS foram comparados aos resultados de outras técnicas da literatura.

Para avaliar os resultados, foram utilizadas quatro medidas: precisão, *recall*, diversidade e novidade. Como foi apresentado nos experimentos, o MOEA-RS obteve melhores resultados que os outros métodos com relação as medidas de precisão, diversidade e novidade, e só não obteve melhor resultado, para a medida de *recall*, que um dos métodos.

Portanto, pode-se concluir, que o MOEA-RS é um método com qualificado para recomendação de filmes, pois conseguiu superar métodos da literatura em medidas de qualidade tradicionais, como a precisão, e em medidas de qualidade que tem agregar muito valor em uma recomendação, como diversidade e novidade.

Porém, o método ainda pode ser melhorado, principalmente a partir do uso de diferentes técnicas de extração de características e aprendizado de máquina. Outros algoritmos de otimização de objetivo podem ser utilizados e avaliados e outros operadores evolucionários também podem aplicados.

A aplicação de algoritmos evolucionário em sistemas de recomendações de filmes baseado em conteúdo é poderosa, neste trabalho foi apresentado um sistema utilizando esses algoritmos que obteve melhores resultados que métodos tradicionais, e ainda tem muito espaço para ser melhorada, já que envolve duas áreas com uma comunidade cientifica muito ativa, que são as de

otimização bio-inspirada e sistemas de recomendações e a sua aplicação tem um grande apelo mercadológico, que é a recomendação de filmes.

Referências

ADOMAVICIUS, G.; MANOUSELIS, N.; KWON, Y. Multi-criteria recommender systems. In: *Recommender systems handbook*. [S.l.]: Springer, 2011. p. 769–803. Citado na página 8.

AGGARWAL, C. C. et al. *Recommender systems*. [S.l.]: Springer, 2016. Citado 3 vezes nas páginas 11, 12 e 15.

ALBUQUERQUE, E. M. *O que faremos com os 40 trilhões de gigabytes de dados disponíveis em 2020?* 2017. Accessed: 2019-08-03. Disponível em: <https://br.okfn.org/2017/09/29/o-que-faremos-com-os-40-trilhoes-de-gigabytes-de-dados-disponiveis-em-2020/>. Citado na página 7.

BISHOP, C. M. *Pattern recognition and machine learning*. [S.l.]: springer, 2006. Citado na página 15.

CAMI, B. R.; HASSANPOUR, H.; MASHAYEKHI, H. A content-based movie recommender system based on temporal user preferences. In: IEEE. *2017 3rd Iranian Conference on Intelligent Systems and Signal Processing (ICSPIS)*. [S.l.], 2017. p. 121–125. Citado na página 19.

CARVALHO, A. B. d. Novas estratégias para otimização por nuvem de partículas aplicadas a problemas com muitos objetivos. 2013. Citado 2 vezes nas páginas 18 e 19.

COELLO, C. A. C. et al. *Evolutionary algorithms for solving multi-objective problems*. [S.l.]: Springer, 2007. v. 5. Citado 2 vezes nas páginas 17 e 19.

DEB, K.; DEB, D. Analysing mutation schemes for real-parameter genetic algorithms. *IJAISC*, v. 4, n. 1, p. 1–28, 2014. Citado na página 35.

DEB, K.; JAIN, H. An evolutionary many-objective optimization algorithm using reference-point-based nondominated sorting approach, part i: solving problems with box constraints. *IEEE transactions on evolutionary computation*, IEEE, v. 18, n. 4, p. 577–601, 2013. Citado na página 19.

DEB, K. et al. A fast and elitist multiobjective genetic algorithm: Nsga-ii. *IEEE transactions on evolutionary computation*, IEEE, v. 6, n. 2, p. 182–197, 2002. Citado na página 19.

DEB, K.; SINDHYA, K.; OKABE, T. Self-adaptive simulated binary crossover for real-parameter optimization. In: . [S.l.: s.n.], 2007. p. 1187–1194. Citado na página 35.

FRIEDMAN, J. H. Stochastic gradient boosting. *Computational statistics & data analysis*, Elsevier, v. 38, n. 4, p. 367–378, 2002. Citado na página 16.

GOODFELLOW, I.; BENGIO, Y.; COURVILLE, A. *Deep Learning*. [S.l.]: MIT Press, 2016. <http://www.deeplearningbook.org>. Citado na página 16.

HARPER, F. M.; KONSTAN, J. A. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)*, ACM, v. 5, n. 4, p. 19, 2016. Citado na página 26.

HIMEL, M. T. et al. Weight based movie recommendation system using k-means algorithm. In: IEEE. *2017 International Conference on Information and Communication Technology Convergence (ICTC)*. [S.l.], 2017. p. 1302–1306. Citado na página 20.

HURLEY, N.; ZHANG, M. Novelty and diversity in top-n recommendation–analysis and evaluation. *ACM Transactions on Internet Technology (TOIT)*, ACM, v. 10, n. 4, p. 14, 2011. Citado na página 8.

IMDB. *What is IMDB?* 2019. Accessed: 2019-08-08. Disponível em: <https://help.imdb.com/article/imdb/general-information/what-is-imdb/G836CY29Z4SGNMK5?ref_=helpsect_cons_1_1#>. Citado na página 26.

IRFAN, R. et al. Mobicontext: A context-aware cloud-based venue recommendation framework. *IEEE transactions on cloud computing*, IEEE, v. 5, n. 4, p. 712–724, 2015. Citado 2 vezes nas páginas 21 e 24.

JAIN, S. et al. Trends, problems and solutions of recommender system. In: IEEE. *International Conference on Computing, Communication & Automation*. [S.l.], 2015. p. 955–958. Citado na página 11.

KATARYA, R.; VERMA, O. P. Effectivecollaborative movie recommender system using asymmetric user similarity and matrix factorization. In: IEEE. *2016 International Conference on Computing, Communication and Automation (ICCCA)*. [S.l.], 2016. p. 71–75. Citado na página 20.

KAYMAK, U.; LEMKE, H. van N. Selecting an aggregation operator for fuzzy decision making. In: IEEE. *Proceedings of 1994 IEEE 3rd International Fuzzy Systems Conference*. [S.l.], 1994. p. 1418–1422. Citado na página 8.

KEMP, S. *Digital in 2018: World's internet users pass the 4 billion mark*. 2018. <https://wearesocial.com/blog/2018/01/global-digital-report-2018>. Accessed: 2019-08-03. Citado na página 7.

KHATWANI, S.; CHANDAK, M. Building personalized and non personalized recommendation systems. In: IEEE. *2016 International Conference on Automatic Control and Dynamic Optimization Techniques (ICACDOT)*. [S.l.], 2016. p. 623–628. Citado na página 11.

KOHAVI, R. et al. A study of cross-validation and bootstrap for accuracy estimation and model selection. In: MONTREAL, CANADA. *Ijcai*. [S.l.], 1995. v. 14, n. 2, p. 1137–1145. Citado na página 34.

LE, Q.; MIKOLOV, T. Distributed representations of sentences and documents. In: *International conference on machine learning*. [S.l.: s.n.], 2014. p. 1188–1196. Citado na página 14.

LOPS, P.; GEMMIS, M. D.; SEMERARO, G. Content-based recommender systems: State of the art and trends. In: *Recommender systems handbook*. [S.l.]: Springer, 2011. p. 73–105. Citado 3 vezes nas páginas 8, 12 e 17.

MIKOLOV, T. et al. Efficient estimation of word representations in vector space. *arXiv preprint arXiv:1301.3781*, 2013. Citado na página 13.

MIKOLOV, T. et al. Distributed representations of words and phrases and their compositionality. In: *Advances in neural information processing systems*. [S.l.: s.n.], 2013. p. 3111–3119. Citado 2 vezes nas páginas 13 e 14.

NETFLIX. *Netflix Prize*. 2009. Disponível em: <https://web.archive.org/web/20090924184639/http://www.netflixprize.com/community/viewtopic.php?id=1537>. Citado na página 7.

OLIVEIRA, S. et al. Multi-objective evolutionary rank aggregation for recommender systems. In: IEEE. *2018 IEEE Congress on Evolutionary Computation (CEC)*. [S.l.], 2018. p. 1–8. Citado 2 vezes nas páginas 23 e 24.

OLSON, D. L.; DELEN, D. *Advanced data mining techniques*. [S.l.]: Springer Science & Business Media, 2008. Citado na página 32.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, v. 12, p. 2825–2830, 2011. Citado na página 16.

RIBEIRO, M. T. et al. Multiobjective pareto-efficient approaches for recommender systems. *ACM Transactions on Intelligent Systems and Technology (TIST)*, ACM, v. 5, n. 4, p. 53, 2015. Citado 2 vezes nas páginas 23 e 24.

RICCI, F.; ROKACH, L.; SHAPIRA, B. Introduction to recommender systems handbook. In: *Recommender systems handbook*. [S.l.]: Springer, 2011. p. 1–35. Citado 4 vezes nas páginas 7, 10, 11 e 12.

SCHÜTZE, H.; MANNING, C. D.; RAGHAVAN, P. Introduction to information retrieval. In: *Proceedings of the international communication of association for computing machinery conference*. [S.l.: s.n.], 2008. v. 4. Citado na página 13.

SCHWARTZ, B. The paradox of choice: Why more is less. In: ECCO NEW YORK. [S.l.], 2004. Citado na página 10.

SHAH, K. et al. Recommender systems: An overview of different approaches to recommendations. In: IEEE. *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*. [S.l.], 2017. p. 1–4. Citado na página 11.

UYANGODA, L.; AHANGAMA, S.; RANASINGHE, T. User profile feature-based approach to address the cold start problem in collaborative filtering for personalized movie recommendation. In: IEEE. *2018 Thirteenth International Conference on Digital Information Management (ICDIM)*. [S.l.], 2018. p. 24–28. Citado na página 21.

WANG, P. et al. A multiobjective genetic algorithm based hybrid recommendation approach. In: IEEE. *2017 IEEE Symposium Series on Computational Intelligence (SSCI)*. [S.l.], 2017. p. 1–6. Citado 2 vezes nas páginas 22 e 24.

WANG, S. et al. Decomposition based multiobjective evolutionary algorithm for collaborative filtering recommender systems. In: IEEE. *2014 IEEE Congress on Evolutionary Computation (CEC)*. [S.l.], 2014. p. 672–679. Citado 2 vezes nas páginas 21 e 24.

YAGER, R. R. On ordered weighted averaging aggregation operators in multicriteria decisionmaking. *IEEE Transactions on systems, Man, and Cybernetics*, IEEE, v. 18, n. 1, p. 183–190, 1988. Citado na página 8.

YOON, Y. C.; LEE, J. W. Movie recommendation using metadata based word2vec algorithm. In: IEEE. *2018 International Conference on Platform Technology and Service (PlatCon)*. [S.l.], 2018. p. 1–6. Citado na página 20.

ZHANG, L. The definition of novelty in recommendation system. *Journal of Engineering Science and Technology Review*, v. 6, p. 141–145, 06 2013. Citado na página 28.

ZHENG, Y.; PU, A. Utility-based multi-stakeholder recommendations by multi-objective optimization. In: IEEE. *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)*. [S.l.], 2018. p. 128–135. Citado 2 vezes nas páginas 22 e 24.

ZUO, Y. et al. Personalized recommendation based on evolutionary multi-objective optimization [research frontier]. *IEEE Computational Intelligence Magazine*, IEEE, v. 10, n. 1, p. 52–62, 2015. Citado 2 vezes nas páginas 22 e 24.

Apêndices

APÊNDICE A – Resultados dos experimentos para escolha das abordagens para análise de conteudo e construção de perfil

Variação	Ridge	SVM	Elastic	GBR	Agregação
w2v-150-1	0.90403	0.90286	0.90240	0.90493	0.90424
w2v-150-2	0.90968	0.90647	0.90290	0.90479	0.90716
w2v-150-3	0.92106	0.91559	0.91867	0.91605	0.91820
w2v-150-4	0.90371	0.90224	0.90240	0.90499	0.90444
w2v-150-5	0.90287	0.90318	0.90240	0.90574	0.90406
w2v-150-6	0.92032	0.91721	0.91185	0.91846	0.91671
w2v-150-7	0.90958	0.90647	0.90290	0.90488	0.90721
w2v-150-8	0.90952	0.90659	0.90288	0.90620	0.90764
w2v-150-9	0.92200	0.91701	0.91931	0.91680	0.92007
w2v-150-10	**0.92343**	0.91690	0.91965	0.91620	0.91942
w2v-150-11	0.90418	0.90340	0.90271	0.90586	0.90396
w2v-150-12	0.92048	0.91816	0.91185	0.91849	0.91758
w2v-150-13	0.91962	0.91699	0.91254	0.91840	0.91787
w2v-150-14	0.91110	0.90659	0.90288	0.90563	0.90774
w2v-150-15	0.92224	0.91751	0.91962	0.91705	0.91960
w2v-150-16	0.92034	0.91695	0.91254	0.91908	0.91787

Tabela 14 – Precisão dos método w2v-150 com diferentes combinações de características

Variação	Ridge	SVM	Elastic	GBR	Agregação
w2v-200-1	0.90403	0.90266	0.90240	0.90219	0.90365
w2v-200-2	0.90921	0.90647	0.90290	0.90286	0.90490
w2v-200-3	0.91959	0.91676	0.91867	0.91906	0.91740
w2v-200-4	0.90402	0.90336	0.90240	0.90483	0.90365
w2v-200-5	0.90271	0.90384	0.90240	0.90266	0.90261
w2v-200-6	0.92047	0.91723	0.91185	0.91858	0.91578
w2v-200-7	0.90958	0.90647	0.90290	0.90510	0.90455
w2v-200-8	0.90952	0.90665	0.90288	0.90234	0.90564
w2v-200-9	0.91970	0.91695	0.91931	0.92038	0.91761
w2v-200-10	0.91950	0.91573	0.91954	0.91914	0.91712
w2v-200-11	0.90269	0.90307	0.90271	0.90462	0.90306
w2v-200-12	0.92039	0.91833	0.91185	0.91877	0.91578
w2v-200-13	0.92032	0.91722	0.91254	0.92014	0.91688
w2v-200-14	0.91115	0.90659	0.90288	0.90433	0.90564
w2v-200-15	**0.92109**	0.91750	0.91868	0.92058	0.91816
w2v-200-16	0.92034	0.91717	0.91254	0.92029	0.91710

Tabela 15 – Precisão dos método w2v-200 com diferentes combinações de características

Variação	Ridge	SVM	Elastic	GBR	Agregação
w2v-250-1	0.90404	0.90337	0.90240	0.90364	0.90251
w2v-250-2	0.90921	0.90647	0.90290	0.90423	0.90536
w2v-250-3	0.92161	0.91534	0.91857	0.91981	0.91626
w2v-250-4	0.90403	0.90259	0.90240	0.90581	0.90251
w2v-250-5	0.90336	0.90357	0.90240	0.90375	0.90347
w2v-250-6	0.92050	0.91712	0.91185	0.92138	0.91655
w2v-250-7	0.90980	0.90647	0.90290	0.90378	0.90561
w2v-250-8	0.90949	0.90659	0.90288	0.90360	0.90496
w2v-250-9	**0.92276**	0.91642	0.91931	0.92034	0.91691
w2v-250-10	0.92060	0.91580	0.91956	0.91989	0.91769
w2v-250-11	0.90283	0.90384	0.90271	0.90410	0.90327
w2v-250-12	0.92024	0.91824	0.91185	0.92086	0.91612
w2v-250-13	0.92032	0.91699	0.91254	0.92227	0.91629
w2v-250-14	0.91122	0.90659	0.90288	0.90353	0.90506
w2v-250-15	0.92162	0.91908	0.91859	0.92116	0.91735
w2v-250-16	0.92034	0.91703	0.91254	0.92194	0.91624

Tabela 16 – Precisão do método w2v-250 com diferentes combinações de características

Variação	Ridge	SVM	Elastic	GBR	Agregação
w2v-300-1	0.90398	0.90208	0.90240	0.90179	0.90271
w2v-300-2	0.90909	0.90647	0.90290	0.90195	0.90438
w2v-300-3	0.92050	0.91680	0.91867	0.91453	0.91737
w2v-300-4	0.90348	0.90235	0.90240	0.90268	0.90271
w2v-300-5	0.90283	0.90291	0.90240	0.90140	0.90261
w2v-300-6	0.92056	0.91724	0.91185	0.91504	0.91518
w2v-300-7	0.90969	0.90647	0.90290	0.90286	0.90447
w2v-300-8	0.90998	0.90659	0.90288	0.90193	0.90422
w2v-300-9	**0.92117**	0.91788	0.91931	0.91492	0.91741
w2v-300-10	0.92109	0.91527	0.91960	0.91508	0.91729
w2v-300-11	0.90404	0.90390	0.90271	0.90278	0.90377
w2v-300-12	0.92051	0.91817	0.91185	0.91591	0.91526
w2v-300-13	0.91961	0.91710	0.91254	0.91496	0.91661
w2v-300-14	0.91115	0.90659	0.90288	0.90315	0.90425
w2v-300-15	0.92102	0.91689	0.91859	0.91497	0.91805
w2v-300-16	0.92034	0.91702	0.91254	0.91599	0.91661

Tabela 17 – Precisão do método w2v-300 com diferentes combinações de características

Variação	Ridge	SVM	Elastic	GBR	Agregação
tfid-500-1	0.90340	0.90282	0.90240	0.90575	0.90224
tfid-500-2	0.91003	0.90558	0.90277	0.90650	0.90499
tfid-500-3	0.91346	0.91152	0.91065	0.91566	0.91187
tfid-500-4	0.90325	0.90319	0.90240	0.90549	0.90224
tfid-500-5	0.90550	0.90429	0.90235	0.90615	0.90382
tfid-500-6	0.91793	0.91123	0.91304	0.91771	0.91441
tfid-500-7	0.90994	0.90558	0.90277	0.90607	0.90509
tfid-500-8	0.91105	0.90592	0.90288	0.90754	0.90544
tfid-500-9	0.91346	0.91141	0.91021	0.91608	0.91198
tfid-500-10	0.91400	0.91123	0.91144	0.91700	0.91261
tfid-500-11	0.90542	0.90429	0.90235	0.90723	0.90382
tfid-500-12	0.91793	0.91147	0.91304	0.91701	0.91417
tfid-500-13	0.91821	0.91121	0.91325	0.91801	0.91464
tfid-500-14	0.91115	0.90592	0.90288	0.90744	0.90554
tfid-500-15	0.91400	0.91119	0.91144	0.91597	0.91117
tfid-500-16	**0.91823**	0.91126	0.91325	0.91738	0.91464

Tabela 18 – Precisão do método tfid-500 com diferentes combinações de características

Variação	Ridge	SVM	Elastic	GBR	Agregação
tfid-1000-1	0.90290	0.90210	0.90267	0.90846	0.90230
tfid-1000-2	0.90962	0.90611	0.90277	0.90838	0.90479
tfid-1000-3	0.91523	0.91533	0.91371	0.91904	0.91653
tfid-1000-4	0.90322	0.90220	0.90257	0.90759	0.90230
tfid-1000-5	0.90387	0.90237	0.90256	0.90769	0.90285
tfid-1000-6	0.91896	0.91249	0.91347	**0.92030**	0.91572
tfid-1000-7	0.90951	0.90611	0.90277	0.90700	0.90534
tfid-1000-8	0.91126	0.90473	0.90288	0.90809	0.90545
tfid-1000-9	0.91530	0.91651	0.91371	0.91732	0.91558
tfid-1000-10	0.91454	0.91473	0.91249	0.91861	0.91570
tfid-1000-11	0.90401	0.90243	0.90267	0.90607	0.90255
tfid-1000-12	0.91933	0.91249	0.91347	0.91958	0.91572
tfid-1000-13	0.91979	0.91277	0.91401	0.92007	0.91598
tfid-1000-14	0.91133	0.90473	0.90288	0.90697	0.90545
tfid-1000-15	0.91466	0.91467	0.91260	0.91746	0.91476
tfid-1000-16	0.91984	0.91288	0.91401	0.91917	0.91598

Tabela 19 – Precisão do método tfid-1000 com diferentes combinações de características

Variação	Ridge	SVM	Elastic	GBR	Agregação
tfid-1500-1	0.90263	0.90308	0.90240	0.90569	0.90230
tfid-1500-2	0.91208	0.90607	0.90301	0.90615	0.90550
tfid-1500-3	0.91733	0.91528	0.91434	0.91620	0.91685
tfid-1500-4	0.90286	0.90340	0.90260	0.90605	0.90240
tfid-1500-5	0.90356	0.90256	0.90329	0.90626	0.90256
tfid-1500-6	0.92017	0.91433	0.91324	0.91762	0.91520
tfid-1500-7	0.91189	0.90607	0.90301	0.90548	0.90481
tfid-1500-8	0.91139	0.90546	0.90288	0.90632	0.90581
tfid-1500-9	0.91733	0.91528	0.91435	0.91369	0.91685
tfid-1500-10	0.91626	0.91636	0.91630	0.91744	0.91838
tfid-1500-11	0.90346	0.90256	0.90345	0.90620	0.90267
tfid-1500-12	0.92017	0.91434	0.91324	0.91469	0.91530
tfid-1500-13	**0.92046**	0.91415	0.91363	0.91703	0.91539
tfid-1500-14	0.91166	0.90552	0.90288	0.90618	0.90511
tfid-1500-15	0.91626	0.91636	0.91630	0.91567	0.91838
tfid-1500-16	0.91987	0.91415	0.91363	0.91494	0.91549

Tabela 20 – Precisão do método tfid-1500 com diferentes combinações de características

Variação	Ridge	SVM	Elastic	GBR	Agregação
tfid-2000-1	0.90280	0.90253	0.90271	0.90412	0.90232
tfid-2000-2	0.91224	0.90584	0.90301	0.90423	0.90434
tfid-2000-3	0.91585	0.91488	0.91422	0.91814	0.91614
tfid-2000-4	0.90344	0.90253	0.90255	0.90422	0.90242
tfid-2000-5	0.90399	0.90353	0.90326	0.90404	0.90240
tfid-2000-6	0.92038	0.91308	0.91269	0.91873	0.91665
tfid-2000-7	0.91251	0.90584	0.90301	0.90462	0.90424
tfid-2000-8	0.91295	0.90495	0.90288	0.90448	0.90503
tfid-2000-9	0.91585	0.91488	0.91416	0.91683	0.91614
tfid-2000-10	0.91694	0.91780	0.91626	0.91691	0.91819
tfid-2000-11	0.90405	0.90344	0.90396	0.90420	0.90295
tfid-2000-12	0.92058	0.91308	0.91269	0.91781	0.91660
tfid-2000-13	0.92049	0.91352	0.91309	**0.92070**	0.91697
tfid-2000-14	0.91406	0.90495	0.90288	0.90477	0.90465
tfid-2000-15	0.91671	0.91802	0.91660	0.91644	0.91828
tfid-2000-16	0.92049	0.91352	0.91309	0.91769	0.91697

Tabela 21 – Precisão do método tfid-2000 com diferentes combinações de características

Variação	Ridge	SVM	Elastic	GBR	Agregação
w2v-150-1	0.99636	0.99600	**1.00000**	0.97101	0.99879
w2v-150-2	0.93588	0.97972	0.99726	0.97021	0.99144
w2v-150-3	0.96236	0.96279	0.96785	0.95230	0.96994
w2v-150-4	0.99374	0.99854	**1.00000**	0.97078	0.99758
w2v-150-5	0.99266	0.99359	**1.00000**	0.96850	0.99595
w2v-150-6	0.93307	0.95970	0.98190	0.95732	0.97924
w2v-150-7	0.93557	0.97972	0.99726	0.97072	0.99194
w2v-150-8	0.93472	0.97538	0.99726	0.97239	0.98820
w2v-150-9	0.96161	0.96190	0.96659	0.95362	0.96943
w2v-150-10	0.95990	0.96133	0.96802	0.95311	0.96753
w2v-150-11	0.98950	0.99236	**1.00000**	0.97005	0.99541
w2v-150-12	0.93797	0.95841	0.98190	0.95818	0.97935
w2v-150-13	0.94120	0.95237	0.98002	0.95708	0.97783
w2v-150-14	0.93553	0.97538	0.99726	0.97081	0.98874
w2v-150-15	0.95970	0.96201	0.96793	0.95472	0.96838
w2v-150-16	0.94139	0.95171	0.98002	0.95942	0.97783

Tabela 22 – Recall dos método w2v-200 com diferentes combinações de características

Variação	Ridge	SVM	Elastic	GBR	Agregação
w2v-200-1	0.99636	0.99746	**1.00000**	0.96961	**1.00000**
w2v-200-2	0.93588	0.97972	0.99726	0.97121	0.99359
w2v-200-3	0.96371	0.96132	0.96785	0.96046	0.97070
w2v-200-4	0.99625	0.99541	**1.00000**	0.97302	**1.00000**
w2v-200-5	**1.00000**	0.99351	**1.00000**	0.97395	0.99784
w2v-200-6	0.93488	0.95861	0.98190	0.96315	0.98135
w2v-200-7	0.93557	0.97972	0.99726	0.97523	0.99359
w2v-200-8	0.93472	0.97584	0.99726	0.97130	0.99154
w2v-200-9	0.96797	0.96320	0.96659	0.96165	0.97202
w2v-200-10	0.96614	0.96525	0.96748	0.96743	0.97100
w2v-200-11	0.99766	0.99104	**1.00000**	0.97524	0.99676
w2v-200-12	0.93718	0.95949	0.98190	0.96556	0.98135
w2v-200-13	0.94108	0.95528	0.98002	0.96627	0.97945
w2v-200-14	0.93604	0.97538	0.99726	0.97455	0.99154
w2v-200-15	0.96242	0.96356	0.96818	0.96321	0.97097
w2v-200-16	0.94139	0.95421	0.98002	0.96821	0.98145

Tabela 23 – *Recall* dos método w2v-200 com diferentes combinações de características

Variação	Ridge	SVM	Elastic	GBR	Agregação
w2v-250-1	0.99750	0.99792	**1.00000**	0.96462	0.99871
w2v-250-2	0.93588	0.97972	0.99726	0.96483	0.98968
w2v-250-3	0.96229	0.96298	0.96671	0.95982	0.96819
w2v-250-4	0.99738	0.99820	**1.00000**	0.96401	0.99871
w2v-250-5	0.99668	0.99368	**1.00000**	0.96302	0.99892
w2v-250-6	0.93464	0.95797	0.98190	0.96544	0.98037
w2v-250-7	0.93665	0.97972	0.99726	0.96410	0.98968
w2v-250-8	0.93441	0.97538	0.99726	0.96474	0.98947
w2v-250-9	0.96378	0.96151	0.96659	0.95843	0.96943
w2v-250-10	0.96365	0.96141	0.96767	0.95950	0.96788
w2v-250-11	0.99564	0.99074	**1.00000**	0.96064	0.99784
w2v-250-12	0.93549	0.95862	0.98190	0.96466	0.98028
w2v-250-13	0.94120	0.95237	0.98002	0.96444	0.97812
w2v-250-14	0.93661	0.97538	0.99726	0.96443	0.99001
w2v-250-15	0.96421	0.96224	0.96739	0.95757	0.96789
w2v-250-16	0.94139	0.95225	0.98002	0.96571	0.97742

Tabela 24 – *Recall* do método w2v-250 com diferentes combinações de características

Variação	Ridge	SVM	Elastic	GBR	Agregação
w2v-300-1	0.99566	0.99711	**1.00000**	0.96912	**1.00000**
w2v-300-2	0.93533	0.97972	0.99726	0.96730	0.99264
w2v-300-3	0.96180	0.96301	0.96785	0.95613	0.97116
w2v-300-4	0.99113	0.99610	**1.00000**	0.96925	**1.00000**
w2v-300-5	0.99436	0.99251	**1.00000**	0.96749	0.99784
w2v-300-6	0.93555	0.95979	0.98190	0.95862	0.98038
w2v-300-7	0.93611	0.97972	0.99726	0.96979	0.99306
w2v-300-8	0.93472	0.97538	0.99726	0.96644	0.99139
w2v-300-9	0.96242	0.96354	0.96659	0.95855	0.97139
w2v-300-10	0.96205	0.96280	0.96818	0.95766	0.97072
w2v-300-11	0.99045	0.99020	**1.00000**	0.96830	0.99730
w2v-300-12	0.93718	0.95886	0.98190	0.95653	0.98092
w2v-300-13	0.94042	0.95419	0.98002	0.95808	0.97917
w2v-300-14	0.93615	0.97538	0.99726	0.96911	0.99170
w2v-300-15	0.96146	0.96386	0.96739	0.95513	0.97092
w2v-300-16	0.94139	0.95306	0.98002	0.95790	0.97850

Tabela 25 – *Recall* do método w2v-300 com diferentes combinações de características

Variação	Ridge	SVM	Elastic	GBR	Agregação
tfid-500-1	0.98208	0.99593	**1.00000**	0.97319	0.99740
tfid-500-2	0.92967	0.98322	0.99837	0.97497	0.99174
tfid-500-3	0.97721	0.98552	0.98353	0.95247	0.98537
tfid-500-4	0.98258	0.99661	**1.00000**	0.97013	0.99740
tfid-500-5	0.98069	0.99387	0.99930	0.96494	0.99670
tfid-500-6	0.94348	0.97565	0.98330	0.95022	0.98355
tfid-500-7	0.92839	0.98322	0.99837	0.97257	0.99228
tfid-500-8	0.93049	0.98213	0.99726	0.97144	0.99174
tfid-500-9	0.97721	0.98547	0.98353	0.95685	0.98612
tfid-500-10	0.97435	0.98895	0.98289	0.95678	0.98651
tfid-500-11	0.98162	0.99387	0.99930	0.96752	0.99670
tfid-500-12	0.94348	0.97703	0.98330	0.95138	0.98517
tfid-500-13	0.94589	0.97540	0.98192	0.95358	0.98468
tfid-500-14	0.93129	0.98213	0.99726	0.96904	0.99228
tfid-500-15	0.97435	0.98849	0.98289	0.95321	0.98522
tfid-500-16	0.94632	0.97586	0.98192	0.95270	0.98468

Tabela 26 – *Recall* do método tfid-500 com diferentes combinações de características

Variação	Ridge	SVM	Elastic	GBR	Agregação
tfid-1000-1	0.98829	0.99492	0.99889	0.98774	**0.99946**
tfid-1000-2	0.93102	0.98310	0.99837	0.98629	0.99279
tfid-1000-3	0.97367	0.97967	0.97456	0.97141	0.98156
tfid-1000-4	0.98798	0.99546	0.99912	0.98762	0.99946
tfid-1000-5	0.98931	0.99346	0.99889	0.98648	0.99888
tfid-1000-6	0.94592	0.97085	0.98553	0.97182	0.98618
tfid-1000-7	0.93206	0.98310	0.99837	0.98758	0.99420
tfid-1000-8	0.93693	0.98222	0.99726	0.98578	0.99334
tfid-1000-9	0.97499	0.97852	0.97444	0.97077	0.98144
tfid-1000-10	0.97366	0.98452	0.97803	0.96770	0.98528
tfid-1000-11	0.98919	0.99392	0.99900	0.98557	0.99900
tfid-1000-12	0.94550	0.97085	0.98553	0.96784	0.98618
tfid-1000-13	0.94742	0.97102	0.98302	0.96744	0.98577
tfid-1000-14	0.93579	0.98222	0.99726	0.98426	0.99322
tfid-1000-15	0.97366	0.98302	0.97803	0.96675	0.98540
tfid-1000-16	0.94777	0.97102	0.98302	0.96773	0.98565

Tabela 27 – *Recall* do método tfid-1000 com diferentes combinações de características

Variação	Ridge	SVM	Elastic	GBR	Agregação
tfid-1500-1	0.98890	0.99733	**1.00000**	0.98619	0.99946
tfid-1500-2	0.94166	0.98343	0.99837	0.98375	0.99450
tfid-1500-3	0.97988	0.97968	0.97387	0.97029	0.98344
tfid-1500-4	0.99022	0.99686	0.99988	0.98571	0.99946
tfid-1500-5	0.99045	0.99278	0.99627	0.98511	0.99884
tfid-1500-6	0.94948	0.96405	0.98396	0.96853	0.98501
tfid-1500-7	0.94049	0.98343	0.99837	0.98663	0.99461
tfid-1500-8	0.94169	0.97946	0.99726	0.98473	0.99358
tfid-1500-9	0.97988	0.97979	0.97399	0.96864	0.98344
tfid-1500-10	0.97792	0.97805	0.97504	0.96889	0.98313
tfid-1500-11	0.99056	0.99278	0.99546	0.98432	0.99884
tfid-1500-12	0.94948	0.96377	0.98396	0.96674	0.98601
tfid-1500-13	0.94935	0.96601	0.98296	0.96787	0.98523
tfid-1500-14	0.94265	0.97992	0.99726	0.98507	0.99358
tfid-1500-15	0.97792	0.97805	0.97504	0.96859	0.98313
tfid-1500-16	0.94754	0.96577	0.98296	0.96484	0.98623

Tabela 28 – *Recall* do método tfid-1500 com diferentes combinações de características

Variação	Ridge	SVM	Elastic	GBR	Agregação
tfid-2000-1	0.99307	0.99671	0.99404	0.98525	**0.99846**
tfid-2000-2	0.94691	0.97718	0.99837	0.98509	0.99599
tfid-2000-3	0.97700	0.97725	0.97337	0.97314	0.98373
tfid-2000-4	0.99119	0.99671	0.99368	0.98541	0.99846
tfid-2000-5	0.99303	0.99439	0.99575	0.98536	0.99746
tfid-2000-6	0.95470	0.96603	0.98244	0.97235	0.98672
tfid-2000-7	0.94691	0.97718	0.99837	0.98484	0.99545
tfid-2000-8	0.95018	0.97892	0.99726	0.98461	0.99548
tfid-2000-9	0.97700	0.97725	0.97240	0.97587	0.98373
tfid-2000-10	0.97684	0.97729	0.97539	0.97091	0.98320
tfid-2000-11	0.99360	0.99362	0.99357	0.98477	0.99678
tfid-2000-12	0.95543	0.96615	0.98244	0.97326	0.98626
tfid-2000-13	0.95341	0.96445	0.98144	0.97109	0.98781
tfid-2000-14	0.95201	0.97892	0.99726	0.98599	0.99514
tfid-2000-15	0.97518	0.97937	0.97454	0.97180	0.98416
tfid-2000-16	0.95307	0.96433	0.98144	0.97076	0.98781

Tabela 29 – *Recall* do método tfid-2000 com diferentes combinações de características