Instituto Politécnico de Viseu
Escola Superior de Tecnologia e Gestão de Viseu

Margarida Isabel de Oliveira Jerónimo

Sistemas de Recomendação para Conteúdos de Aplicações web

dezembro de 2020

limite texto

Instituto Politécnico de Viseu

Escola Superior de Tecnologia e Gestão de Viseu

Margarida Isabel de Oliveira Jerónimo

Sistemas de Recomendação para Conteúdos de Aplicações web

Tese de Mestrado
Sistemas e Tecnologias de Informação para as Organizações

Professor Doutor Filipe Cabral Pinto

dezembro de 2020

A Ti, à Raquel, ao Tiago e às minhas mães.

RESUMO

Grandes volumes de dados, referidos como *Big Data*, são gerados diariamente a uma taxa sem precedentes, a partir de fontes heterogéneas (*e.g.*, meio ambiente, saúde, governo, redes sociais, *marketing*, transações financeiras). As novas tendências tecnológicas atuais, incluindo a Internet das Coisas (*IoT – Internet of Things*), a proliferação da *Cloud Computing* e a massificação dos dispositivos inteligentes (*e.g., smartphones, smartwatches*, dispositivos pervasivos e ubíquos), têm contribuído para esta explosão de dados. Como infraestrutura de suporte têm-se sistemas e aplicações distribuídos(as), públicos(as) e privados(as), interligados por redes de comunicação eletrónica de banda larga e elevado desempenho, normalmente com interface web.

Os Sistemas de Recomendação são sistemas que procuram facilitar a penosa atividade de busca por conteúdo de interesse no *Big Data*. As principais funções dos Sistemas de Recomendação são a análise das diversas ações do utilizador do sistema. Com essa análise é possível extrair informações úteis para futuras predições, fornecendo recomendações de diferentes itens (*e.g.*, sugestões de músicas, filmes, conteúdos de comércio eletrónico). Existem diferentes variantes nos sistemas de recomendação, nomeadamente, sistemas de filtragem colaborativa de classificações (*ratings*), filtragem baseada em conteúdo dos itens (*e.g.*, descrição, características) ou de filtragem híbrida (que combinam as duas aproximações anteriores), tendo todos por objetivo a seleção de conteúdos de interesse tendo em conta os padrões de consumo dos utilizadores.

O trabalho desenvolvido consistiu na implementação e validação de um protótipo de um Sistema de Recomendação de Conteúdos de Aplicações web, aplicado à recomendação de filmes. Para tal, utilizaram-se os *MovieLens Datasets offline* de *ratings* disponibilizados, pelo *GroupLens*, bem como informação disponibilizada *online* pelo site do *TMDb*.

O sistema desenvolvido aprende o padrão de consumo de conteúdos do utilizador, prevendo o que irá consumir no futuro com base nos itens similares aos que demonstrou interesse (classificou) no passado, bem como na similaridade com outros utilizadores (que constituem a sua vizinhança) e assim fornecer os respetivos conteúdos de interesse, permitindo criar um modelo de utilizador.

Utilizaram-se as técnicas de Filtragem Baseada em Conteúdo, Filtragem Colaborativa e Filtragem Híbrida baseadas em memória. A **Filtragem Baseada em Conteúdo** permite através da análise das caraterísticas dos itens, essencialmente baseadas na metodologia *TF-IDF* para o processamento de linguagem natural (NLP), extrair as características ou atributos fundamentais dos itens e selecionar itens semelhantes ou propor classificações previstas para os itens de interesse ainda não classificados pelo utilizador ativo. A **Filtragem Colaborativa** permite aplicar a metodologia *kNN*, para identificar a semelhança entre o utilizador ativo, situados na vizinhança e propor classificações previstas para itens de interesse ainda não classificados.

Ambas as aproximações têm inconvenientes e vantagens. A filtragem baseada em conteúdo, tende a especializar muito as recomendações em torno das características dos itens e, eventualmente, do utilizador ativo, uma vez que não tem em atenção os gostos dos restantes utilizadores do sistema. A filtragem colaborativa, essencialmente, tem a desvantagem do *cold start*, isto é, os problemas associados à admissão de novos utilizadores ou novos itens no sistema. Naturalmente que os inconvenientes de uma aproximação são as vantagens da outra e *vice-versa*. A **Filtragem Híbrida** combina as duas metodologias de forma a ultrapassar os seus inconvenientes podendo também seguir várias abordagens. No caso deste trabalho foi seguida uma abordagem *weighted*, permitindo uma combinação linear das filtragens colaborativas e baseada em conteúdo.

Na avaliação experimental, os resultados obtidos foram relevantes em termos empíricos, coincidentes com os resultados apresentados em estudos semelhantes e validados com as métricas estatísticas *MAE* e *RMSE*.

O protótipo de Sistema de Recomendação desenvolvido poderá evoluir para um sistema de recomendação de produção, sendo adaptável para outros conteúdos de aplicações *web*.

ABSTRACT

Large volumes of data, referred to as Big Data, are generated daily at an unprecedented rate, from heterogeneous sources (e.g., environment, health, government, social networks, marketing, financial transactions). Current new technological trends, including the Internet of Things (IoT), the proliferation of Cloud Computing and the smart devices generalization (e.g., smartphones, smartwatches, pervasive and ubiquitous devices), have contributed to this data explosion. As support infrastructure there are distributed, public and private systems and applications, interconnected by high-performance and high-performance electronic communication networks, usually with a web interface.

Recommender Systems are systems that seek to facilitate the painful activity of searching for content of interest in Big Data. The main functions of the Recommendation Systems are the analysis of the various actions of the system user. With this analysis it is possible to extract useful information for future predictions, providing recommendations for different items (e.g., music suggestions, films, e-commerce content). There are different variants in the recommender systems, namely, ratings collaborative filtering systems, content-based filtering systems (e.g., items' description or characteristics) or hybrid filtering (which combine the two previous approaches), all aiming at the selection of content of interest considering the consumption patterns of users.

This work consisted of the implementation and validation of a prototype of a web Content Application Recommender System, applied to movies recommendation. To this end, we used the offline MovieLens Datasets of ratings made available by GroupLens, as well as information made available online by the TMDb website.

The developed system learns the pattern of consumption of user content, predicting what it will consume in the future based on items like those it has shown interest in (rated) in the past, as well as similarity with other users (who constitute its neighborhood) and thus providing the respective contents of interest, allowing to create a user model.

Memory-based Content-based filtering, collaborative filtering and hybrid filtering techniques were used. **Content-Based Filtering** allows, through the analysis of the characteristics of the items, essentially based on the TF-IDF methodology for natural language processing (NLP), to extract the fundamental characteristics or attributes of the items and select similar items or propose predicted ratings for the items of interest not yet classified by the active user. **Collaborative Filtering** allows applying the kNN methodology, to identify the neighborhood similarity to the active user, and to propose predicted ratings for items of interest not yet classified.

Both approaches have problems and advantages. Content-based filtering tends to specialize the recommendations around the characteristics of the items and the active user, since it does not consider the tastes of the other users of the system. Collaborative filtering, essentially, has the disadvantage of cold start, that is, the problems associated with the admission of new

users or new items in the system. Naturally, the drawbacks of one approach are the advantages of the other and vice versa. **Hybrid Filtering** combines the two methodologies to overcome its drawbacks and can also follow several approaches. In the case of this work, a weighted approach was followed, allowing a linear combination of collaborative and content-based filtering.

In the experimental evaluation, the results obtained were relevant in empirical terms, coinciding with the results presented in similar studies and validated with the statistical metrics MAE and RMSE.

The developed Recommendation System prototype can evolve into a production recommender system, being adaptable to other web application content.

PALAVRAS CHAVE

Sistema de Recomendação
Filtragem Colaborativa
Filtragem Baseada em Conteúdo
Filtragem Híbrida
Machine Learning
Data Mining
Big Data
Aplicação web
Data Science
TF-IDF
NLP

KEY WORDS

Recommender System
Collaborative Filtering
Content-Based Filtering
Hybrid Filtering
Machine Learning
Data Mining
Big Data
Web Application
Data Science
TF-IDF
NLP

AGRADECIMENTOS

Ao meu orientador, Professor Doutor Filipe Cabral Pinto, por todas reuniões semanais à distância, através de plataforma eletrónica, onde para além das orientações técnico-científicas, foi uma fonte de motivação e amizade.
Ao Professor Carlos Costa, da Escola Superior de Tecnologia e Gestão de Lamego, pela orientação no desenvolvimento do protótipo e implementação efetuada.
Ao Professor Doutor Rui Pedro Duarte, da Escola Superior de Tecnologia e Gestão de Viseu, pelas sugestões na interface e de melhoria deste documento.

ÍNDICE GERAL

ÍNDICE DE FIGURAS

ÍNDICE DE QUADROS

NOTAÇÃO

a) Maiúsculas latinas

U	Conjunto de todos os utilizadores
I	Conjunto de todos os itens que podem ser recomendados
R	Matriz utilizador-item de classificações (*ratings*)

b) Minúsculas latinas

r	Função utilidade (utilidade/relevância de um item *I* para um utilizador)
u	Um utilizador específico
i	Um item específico

c) Índices superiores e inferiores gerais

u_a	Utilizador ativo
u_i	Utilizador índice *i*
I_u	Lista de itens em que o utilizador u expressou as suas preferências
U_i	Lista de utilizadores que avaliaram o item *i*
i_j	Item índice *j*
i'	Item que obedece a uma condição
$P_{a,j}$	Previsão de classificação do utilizador ativo para o item índice j
I_r	Recomendação *Top-N* de itens
$r_{i,j}$	Avaliação (rating) do utilizador i para o item j
$\bar{r}_u$	Média das classificações do utilizador u
$P_{u,i}^{(CF)}$	Previsão de classificação do utilizador *u* para o item *i* pela técnica CF

ABREVIATURAS E SIGLAS

CBF	Sistemas de Filtragem Baseada em Conteúdo ou *Content-Based Systems*
CF	Sistemas de Filtragem Colaborativa ou *Collaborative Filtering Systems*
DM	*Data Mining*
DS	*Data Science*
HF	Sistemas de Filtragem Híbrida ou *Hybrid Filtering Systems*
IoT	*Internet of Things*
KDD	*Knowledge Discovery in Databases*
kNN	*k-Nearest Neighbors*
ML	*Machine Learning* ou Aprendizagem Máquina
NLP	*Natural Language Processing*
SR	Sistema de Recomendação ou *Recommender System*
SVD	*Singular Value Decomposition*
TF-IDF	*Term Frequency-Inverse Document Frequency*

1. Introdução

A massificação do acesso às tecnologias e sistemas de informação, que se tem verificado nas últimas décadas, provocou um aumento vertiginoso na informação disponível.

A web é hoje a principal infraestrutura de suporte aos sistemas de informação à escala global e está em evolução constante em termos serviços e aplicações.

Antes da evolução do *Big Data*, as organizações não tinham possibilidade de salvaguardar os dados por longos períodos a custos comportáveis, nem estavam tipicamente habilitadas a geri-los de forma eficiente. Basicamente as organizações não dispunham de escalabilidade, flexibilidade, desempenho e economia de escala, fundamentais em contextos *Big Data* (Leskovec et al., 2014).

A manipulação de *Big Data* requer recursos computacionais significativos, novos métodos de redução de dados e tecnologias distribuídas de elevado desempenho. Concretamente, é necessário limpar, processar, analisar e assegurar um acesso, com a granularidade certa, a conjuntos de dados massivos e evolutivos (Resnick & Varian, 1997). Atualmente as empresas e indústrias estão cientes de que a análise de dados se tornou um fator vital de competitividade, descoberta de conhecimento e personalização de serviços (M. Chen et al., 2014).

Assim, foram surgindo vários modelos, plataformas e tecnologias para fornecer capacidade de armazenamento, processamento distribuído e paralelo, bem como de análise *online* de várias fontes heterogéneas. Adicionalmente, são fornecidas soluções que salvaguardam os aspetos de segurança e privacidade. Toda esta informação gerada implica gigantescas quantidades de dados, levando assim, à necessidade da utilização de arquiteturas de *Big Data*, para tratar todos esses dados e gerar informações úteis, que podem ser aplicadas para criar Sistemas de Recomendação de processamento massivo de dados (Carvalho, 2018).

A gestão do crescimento exponencial da informação criada e disponibilizada a cada instante na Internet em geral e na web particular, proveniente das mais diversas fontes, não é uma

tarefa, fácil de lidar para os utilizadores de diferentes serviços. Por outro lado, muita da informação gerada não interessa de igual forma a todas os utilizadores. Assim, é importante a criação de sistemas, que possam auxiliar os utilizadores a selecionar a informação mais útil de acordo com os seus interesses, os Sistemas de Recomendação (SR) ou *Recommender* Systems (Reis, 2012).

Estes utilizadores podem ser profissionais que necessitam de informação para a tomada de decisões ou cidadãos comuns para recomendação de produtos, atividades lúdicas e culturais.

Sistemas de Recomendação, são sistemas de filtro de informação, que utilizam técnicas de *Machine Learning (ML)* e que têm como objetivo processar dados históricos de transações, de utilização/consumo ou de preferências dos utilizadores. Um SR aprende com os utilizadores e sugere ou recomenda itens relevantes, entre os itens disponíveis, evitando expor o utilizador a um moroso processo de seleção de informação supérflua (Resnick & Varian, 1997). Os SR dependem, em grande parte da quantidade de informação que se obtém dos utilizadores, para assim, fornecer recomendações de interesse (Isinkaye et al., 2015).

Por um lado, é necessário fazer recomendações através da recolha de informação das mais diversas formas e fontes, filtrando conteúdos, seletivamente e de acordo com as preferências do utilizador. Por outro lado, para as empresas de vendas *online*, que têm como um dos grandes objetivos as vendas, a existência de recomendação, é eficaz na sugestão de novos produtos. A introdução dos SR nos serviços web registou um aumento do número de vendas e, consequentemente, o interesse das empresas em implementar novos sistemas. A qualidade das recomendações é um fator determinante, uma vez que a confiança dos utilizadores nas sugestões apresentadas, só se verifica se estas se aproximarem, o mais possível dos seus interesses (Schafer et al., 1999). Assim, torna-se emergente a necessidade de criação de mecanismos, que possam auxiliar os utilizadores de sistemas a selecionar a informação mais útil de acordo com os seus interesses.

Pelo exposto e, resultado de trabalhos desenvolvidos nestas áreas, durante o 1º ano do Mestrado em Sistemas e Tecnologias de Informação para as Organizações (MSTIO), do Departamento de Informática da Escola Superior de Tecnologia e Gestão de Viseu do Instituto Politécnico de Viseu, estas são as áreas de interesse e que se pretendem aprofundar. Trata-se efetivamente de uma área nos âmbitos da *Data Science* e *Data Mining*, em evolução crescente, do interesse pessoal e com procura crescente de profissionais.

Pretende-se, assim, documentar o trabalho desenvolvido na dissertação de mestrado, que consistiu na **implementação de um SR Híbrido, que utiliza recomendações baseadas em conteúdo e colaborativas, que poderá ser integrado em sistemas de software de recomendação para conteúdos de aplicações web**, que permite aceder a diretórios públicos relacionados com publicitações de filmes, por exemplo, uma vez que são estes os dados que constituem os *datasets* a utilizar. Neste documento são ainda descritas as várias experiências, estudos e resultados até chegar ao SR Híbrido pretendido.

A partir do conhecimento produzido, pelo estudo da interação dos utilizadores e itens, será construído o SR. Este sistema, permitirá estudar padrões do comportamento dos utilizadores das aplicações de software, permitindo ao sistema “aprender” qual o padrão de consumo de

conteúdos do utilizador e, assim, prever o que irá consumir no futuro, fornecendo-lhe os respetivos conteúdos de interesse.
Assim, este SR deverá:

- recolher informação (dados), a partir de diversas fontes;
- preparar os dados para análise (limpar, normalizar e combinar os dados);
- utilizar algoritmo(s) de extração de dados para treino e recomendação;
- classificar a informação recolhida através do reconhecimento de padrões;
- permitir ao utilizador selecionar os tópicos de interesse de sua preferência e assim ultrapassar a ausência de informação, sobre o utilizador, no arranque da aplicação;
- recomendar informação personalizada e relevante a cada utilizador, com base nas suas preferências, tirando partido do conhecimento produzido, bem como do seu *feedback*.

Este documento constitui a dissertação de mestrado e encontra-se estruturado da seguinte forma. No **capítulo 2,** será efetuada a definição do problema, que inclui a definição dos objetivos a atingir e contribuições esperadas. Segue-se, no **capítulo 3,** a descrição geral do estado da arte, com a apresentação dos conceitos fundamentais, classificação e tipos de sistemas de recomendação, bem como sistemas implementados e trabalhos relacionados. No **capítulo 4**, far-se-á a apresentação e justificação das metodologias utilizadas. No **capítulo 5**, explicar-se-ão os passos efetuados para o desenvolvimento do Sistema de Recomendação Híbrido. A seguir, haverá o **capítulo 6,** onde, é efetuada a analise e discussão dos resultados obtidos. No **capítulo 7**, são apresentadas as conclusões e trabalhos futuros. Por fim, seguem-se as referências investigadas.

2. Definição do problema

No contexto do *Big Data* é uma necessidade a criação de mecanismos que possam auxiliar os utilizadores a selecionar a informação mais útil, de acordo com os seus interesses. Assim, torna-se essencial a criação de SR capazes de recolher informação, das mais diversas formas e fontes, filtrando conteúdos, seletivamente e de acordo com as preferências do utilizador. Um SR, ajuda na tomada de decisão, baseado nas experiências, eventualmente, dos outros, assim como em conteúdos consumidos no passado. No entanto, para que possa desempenhar a sua função, é necessário que o SR "entenda" o utilizador, o que este pretende, quais as suas necessidades e preferências (Resnick & Varian, 1997).

Na web é importante a utilização de mecanismos capazes de filtrar, priorizar e fornecer informação relevante, para aliviar o problema da sobrecarga de informação. Os SR permitem pesquisar e tratar grandes volumes de informação, gerados pela interação entre utilizadores e itens. Permitem, ainda, a aprendizagem de preferências dos utilizadores de plataformas de software e, com essas informações, geram-se as recomendações de novos itens (Isinkaye et al., 2015).

Em (Adomavicius & Tuzhilin, 2005) é definido que o problema de recomendação se resume ao problema de estimar classificações para os itens que não foram vistos por um utilizador. Essa estimativa é geralmente baseada nas avaliações dadas pelo utilizador a outros itens. Assim que seja possível estimar as classificações para os itens ainda sem classificação, poder-se-á recomendar ao utilizador os itens com as classificações mais altas estimadas.

No entanto, recomendar conteúdos é um problema, uma vez que se tem de estimar avaliações para os itens que não foram consumidos por um utilizador. Esta estimativa é geralmente baseada na avaliação dada por esses utilizadores a outros itens, baseadas nas caraterísticas extraídas que caracterizam os itens ou analisando as classificações para os itens não consumidos da vizinhança de utilizadores similares. Para que se possa recomendar é

importante criar um modelo de utilizador, essencial para que o SR lhe possa fornecer recomendações que vão de encontro às suas expectativas.

Para que os SR sejam eficazes, necessitam de aprender com a interação dos utilizadores com o sistema. Para novos utilizadores, não existe qualquer registo de atividade e itens, o que significa, que não existem classificações no sistema para esses utilizadores. Torna-se um desafio para os sistemas fazerem sugestões. É o chamado *cold start*, ou "arranque a frio", em português. Um utilizador novo numa plataforma, ou um novo item, é um problema de *cold start*. O problema *cold start* pode ser visto como uma instância especial do *sparsity problem* (muitos itens sem classificações). É aconselhável, que para um novo utilizador, o sistema em primeiro lugar, extraia as suas preferências, para posteriormente poder selecionar o grupo de utilizadores similares (vizinhos) que classificaram itens e assim efetuar a recomendação (Z. Huang et al., 2004; Sarwar et al., 2001).

Para além do problema do *cold start*, um problema para os SR, é o "egoísmo" dos utilizadores, que ao possuírem o seu perfil de interesses, usam as recomendações de terceiros, sem contribuir para o sistema com novas recomendações, empobrecendo essas mesmas (Kim et al., 2010). Um outro problema poderá ser os fornecedores de conteúdos, que podem cometer fraude, uma vez que ao utilizarem o sistema, podem fazer recomendações positivas, nem sempre fiáveis, dos seus produtos e negativas aos produtos da concorrência. Os autores (Puglisi et al., 2015), ainda apresentam um terceiro problema crítico neste tipo de sistema, a questão da privacidade, uma vez que os utilizadores revelam as suas preferências, interesses e atividades.

Pode ainda ocorrer um problema de escalabilidade, associado ao crescimento exponencial do número de classificações ou opiniões de utilizadores sobre itens, o que exige um maior esforço computacional. Poder-se-á resolver este problema da escalabilidade dos dados, através da implementação de algoritmos capazes de dividir o conjunto de utilizadores em grupos de utilizadores com gostos/preferências semelhantes. Assim, quando o sistema pretender calcular uma recomendação, em vez de comparar o utilizador com todos os utilizadores do sistema compara apenas com o grupo de utilizadores que possuam maiores semelhanças com as do utilizador que se pretende recomendar (Zuva et al., 2012).

A opinião de um utilizador que não esteja nem de acordo nem em desacordo com o restante grupo de utilizadores do sistema (*gray sheep*), pode provocar limitações nas recomendações. Uma outra limitação nas recomendações é, este grupo de utilizadores dificilmente recebem recomendações úteis. Noutros casos, o utilizador (*white sheep*) classifica os itens de forma semelhante à maioria dos utilizadores, ou ainda, o utilizador (*black sheep*) que classifica os itens como "muito mau" ou "muito bom", acabando por não se poder relacionar com o restante grupo de utilizadores. Estes problemas podem ser diminuídos caso apareçam utilizadores com gostos ou formas de classificar semelhantes, ou seja, criando uma vizinhança específica para estes casos. Um último problema poderá ser o caso da sinonímia, que ocorre quando itens com nomes diferentes se referem a itens similares. A maioria dos SR não consegue descobrir essa associação e, portanto, tratam esses produtos de maneira diferente. A

prevalência de sinónimos diminui o desempenho das recomendações (Su & Khoshgoftaar, 2009; Zuva et al., 2012).

Em resumo, é fundamental ultrapassar numa primeira fase vários problemas. Desde logo, os problemas, da inicialização (*cold start*) quer de um novo utilizador, quer de um novo item.

Um outro problema prende-se com o fenómeno designado por *long tail,* onde vários itens não são avaliados, ou contém poucas classificações, tornado as recomendações pouco precisas. Segundo (Anderson, 2004, 2006), produtos com pouca procura ou com um reduzido volume de vendas podem, coletivamente, atingir uma quota de mercado que competem ou ultrapassam a dos produtos mais vendidos em menores quantidades. O autor considera que existem duas partes distintas no gráfico: a cabeça e a cauda. Em que a cabeça da distribuição corresponde ao mercado de produtos mais populares enquanto a cauda corresponde ao pequeno nicho de mercado. Este conceito de *long tail,* associado ao estudo dos mercados económicos, também pode ser utilizado na elaboração de SR, uma vez que através da análise do gráfico se pode extrair informação da distribuição dos *ratings* e efetuar recomendações personalizadas conforme a distribuição das classificações.

Um outro problema é o da vizinhança, em que utilizadores com o mesmo tipo de gostos, não avaliam os mesmos itens.

Este trabalho tem como objetivo a criação de um SR Híbrido, efetuar os estudos e retirar as conclusões de implementação das técnicas de filtragem baseada em Conteúdo e Colaborativa, individualmente e, posteriormente combinadas, de forma a permitirem realizar o SR Híbrido.

Assim, o SR permite a seguinte funcionalidade:

- **Recolher informação**

  Os dados são recolhidos a partir de diversas fontes estruturadas (*online*) e não estruturadas (*datasets offline*), relacionados com filmes, suas caraterísticas e classificações por vários utilizadores.

- **Preparar os dados para análise**

  Os dados são limpos, corrigidos e são removidas inconsistências, verificando se há dados ausentes ou incompletos. Uma caraterística destes conjuntos de dados é sua esparsidade, uma vez que há muitos itens sem classificações, assim como a distribuição de *ratings* (classificações) não é uniforme, ou seja, enquanto uma pequena parte dos itens tendem a obter a maioria das classificações dos utilizadores, uma outra parte, tende a receber poucas classificações por item.

- **Utilizar algoritmos de extração de dados**

  Utilizaram-se técnicas de filtragem baseadas em conteúdo, para fazer as previsões, baseadas nas informações textuais dos filmes, já classificados, assim como técnicas de filtragem colaborativa, para recomendação de itens, identificando utilizadores com gostos semelhantes (vizinhança) e fazendo recomendações com base na aprendizagem efetuada com base nos dados em memória. Pode ser efetuada uma combinação das técnicas de filtragem, permitindo uma aprendizagem híbrida.

- **Permitir ao utilizador selecionar os tópicos de interesse de sua preferência**

Para um novo utilizador num sistema, é importante que este dê algumas informações sobre as suas preferências, uma vez que os dados iniciais que existem sobre o utilizador, são insuficientes para efetuar recomendações. Assim, para além de se analisar a interação do utilizador com o sistema de forma implícita, com base no seu histórico de navegação, o sistema permitirá solicitar ao utilizador que forneça classificações para itens e características, permitindo construir e melhorar o modelo de utilizador e as recomendações.

- **Aprender dinamicamente os interesses do utilizador e recomendar informação relevante**

À medida que o utilizador vai interagindo com a aplicação, o sistema, deverá de forma implícita aprender com as ações do utilizador na aplicação e assim, fornecer futuras recomendações personalizadas e relevantes a cada utilizador, com base nas suas preferências, tirando partido do conhecimento produzido, bem como do seu *feedback*.

- **Identificar as classificações dos itens**

Por fim, o utilizador poderá perceber as classificações e posicionamento dos itens, atribuídos por ele e relativamente às classificações atribuídas por outros.

3. Estado da Arte

Os Sistemas de Recomendação, têm vindo a ganhar destaque em diversas áreas de investigação (M. Chen et al., 2014; Hofmann, 2003; Lee & Lee, 2019; Mazeh & Shmueli, 2019; Oussous et al., 2018; Ying et al., 2018; Yu, 2012), existindo uma melhoria e criação de novos algoritmos, mais eficientes e mais eficazes nas suas recomendações.
Pretende-se aqui, efetuar uma revisão dos sistemas de recomendação, incluindo sistemas implementados e trabalhos relacionados.
Nas secções seguintes, seguir-se-á uma caracterização dos tipos de SR, a forma de enquadramento dos Sistemas de Recomendação, nos âmbitos comerciais e académicos, sendo descritos alguns trabalhos relacionados.

3.1 Tipos de Sistemas de Recomendação

Segundo (Resnick & Varian, 1997), os SR auxiliam no aumento da capacidade e eficácia no processo de decisão. Os SR são, normalmente, classificados de acordo com a forma como interpretam a informação de cada utilizador. Um ponto comum na pesquisa de SR é a necessidade de combinar técnicas de recomendação para atingir o desempenho máximo. Existem diferentes técnicas de recomendação, assim como diferentes formas de recolha de informação. O uso de técnicas de recomendação eficientes e precisas é muito importante para um sistema, que fornecerá recomendações úteis para seus utilizadores individuais (Burke, 2002).
Os SR são classificados com base nos seus domínios de aplicação, no conhecimento utilizado, na forma como formulam recomendações e nos algoritmos que implementam (Ricci et al., 2011). Conforme a Figura 3-1, os SR, podem ser classificados, em SR baseados em

Conteúdos (Lang, 1995), baseados em Filtragem Colaborativa (Resnick et al., 1994) e Híbridos (Balabanovic, 1997).

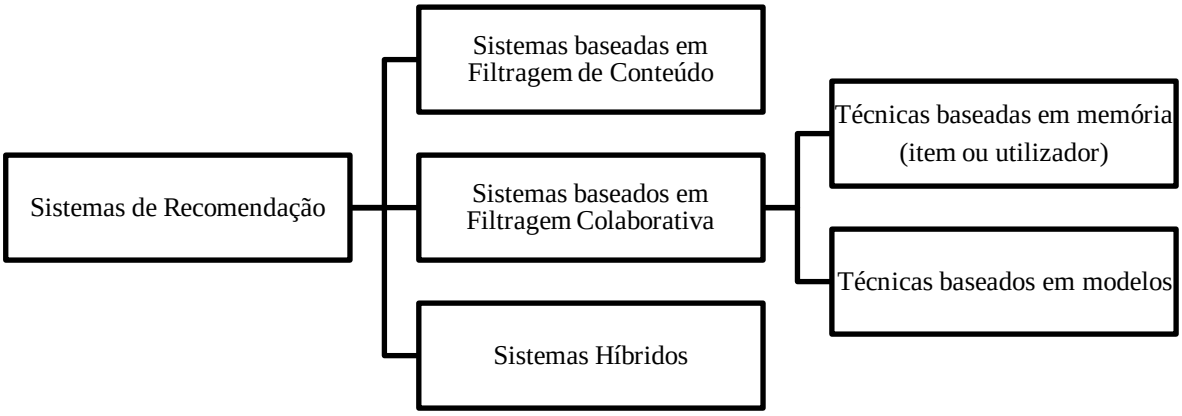

Figura 3-1: Técnicas de filtragem da informação, baseado em (Isinkaye et al., 2015)

Há autores que, ainda complementam a classificação dos SR em baseados em Dados Demográficos (Balabanovic, 1997), Conhecimento (Gennari et al., 2003) e Comunitários (Pham et al., 2011). Estas classificações serão explicadas nas secções seguintes.

3.1.1 Sistemas baseados em Filtragem de Conteúdo

A tecnologia de filtragem baseada em conteúdo, *Content-Based Filtering* (CBF), utiliza técnicas de previsão e treino, baseadas nas informações que são recolhidas dos utilizadores (ver Figura 3-2). Estes sistemas recomendam itens com base na informação textual das características de um item (metadados), partindo do pressuposto de que os utilizadores gostarão de itens semelhantes aos já consumidos, definindo então, a similaridade entre os itens (Lee Herlocker, 2000).

Há autores que defendem outras abordagens. Concretamente em (Iaquinta et al., 2008) recomendam itens que são pouco similares ao perfil do utilizador, assumindo que, quanto menor a probabilidade de um utilizador conhecer um item, mais alta é a probabilidade daquele item ser uma recomendação surpreendente.

A descrição textual dos itens é usada para construir perfis de itens. Os perfis do utilizador podem ser definidos, construindo um modelo das preferências do utilizador, usando as descrições e tipos de itens, nos quais o utilizador está interessado, ou no histórico de navegação no sistema (Ghazanfar & Prügel-Bennett, 2010). Este sistema, é capaz de investigar o padrão de navegação de um utilizador, para prever os itens em que ele pode estar interessado (Balabanovic, 1997). Os perfis são obtidos pela análise do conteúdo dos itens previamente vistos e avaliados pelo utilizador e, geralmente, são construídos usando técnicas de análise de palavras-chave a partir da recuperação de informações (Lang, 1995) (Adomavicius & Tuzhilin, 2005).

O CBF tem algumas limitações. Por exemplo, as palavras-chave usadas para representar os metadados dos itens podem não ser muito representativas. Além disso, as abordagens baseadas em conteúdo sofrem a limitação de fazer recomendações precisas para utilizadores com poucas avaliações. (De Campos et al., 2010).

A informação acerca do tipo de itens pode ser obtida implícita ou explicitamente. Para tal, usam-se métodos que podem ser, questionários, que solicitam opiniões e avaliações dos utilizadores ou algoritmos de aprendizagem, que inferem os gostos e interesses dos utilizadores, de acordo com as suas ações no sistema. Os itens relacionados e, principalmente, os que possuem classificação positiva, são recomendados aos utilizadores com perfil semelhante. O sistema usa diferentes tipos de modelos, para encontrar semelhança entre os itens, de modo a gerar recomendações significativas, como modelos probabilísticos (*e.g. Naive Bayes Classifier*, árvores de decisão ou redes neurais) ou modelos de espaço vetorial, como o TF-IDF (Balabanovic, 1997).

As técnicas de filtragem baseadas em conteúdo, têm a capacidade de recomendar novos itens, mesmo que não haja classificações fornecidas pelos utilizadores, o que não significa que haja grande precisão das recomendações para utilizadores novos. Asseguram uma maior privacidade, uma vez que não necessitam de partilhar o seu perfil. Segundo (Isinkaye et al., 2015). Um dos problemas destes sistemas poderá ser a limitação dos conteúdos, devido à escassez de dados, provocando uma menor precisão na recomendação de novos itens, uma vez que que são poucos os dados para análise. As técnicas CBF dependem dos metadados, dados sobre os atributos dos itens, ou seja, a descrição detalhada dos itens e um perfil de utilizador, muito bem organizado, antes que se possam fazer recomendações precisas. Outro problema são as recomendações semelhantes aos itens já definidos nos perfis de cada utilizador. Na Figura 3-2 poder-se-á constatar como funciona um sistema CBF.

Uma das metodologias utilizadas no CBF, é o TF-IDF (*Term Frequency-Inverse Document Frequency*), que é uma subárea do processamento de linguagem natural. O NLP (*Natural Language Processing*), é um campo da Inteligência Artificial que permite que as máquinas consigam ter a habilidade de ler, entender e derivar significado de linguagens humanas, sendo utilizado na obtenção de informação para a extração de características (*feature*). Fundamentalmente, são contadas a ocorrência de cada palavra num documento, ponderada a importância de cada palavra e calculada a classificação/pontuação (*score*) para o documento. Quanto mais alto o *score* TF-IDF (peso), mais raro o termo e *vice-versa*. (Salton & Buckley, 1988). TF (*Term Frequency*) é a frequência de uma palavra no documento corrente relativamente ao número total de palavras no documento, isto é, a ocorrência das palavras num documento e a atribuição do maior peso, à maior frequência. Esta é dividida pelo tamanho do documento (em número de palavras). IDF (*Inverse Document Frequency*) é rácio entre o número total de documentos e a frequência de ocorrência de documentos que contém a palavra em questão. Este define a raridade de palavras, uma vez que à medida que a ocorrência da palavra no documento é menor, o IDF aumenta. Auxilia a atribuição de maior pontuação a termos raros nos documentos (Christian et al., 2016).

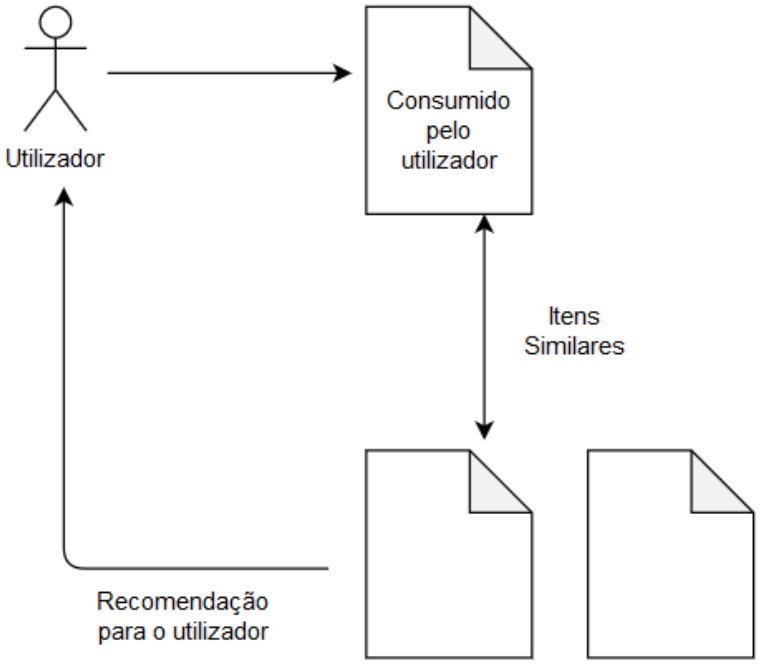

Figura 3-2: Sistemas baseados em Filtragem de Conteúdo

Em conclusão, o TF-IDF é a medida utilizada para avaliar o quão importante uma palavra é num documento em relação a um conjunto de documentos (*document corpus*). A importância das palavras aumenta proporcionalmente ao número de vezes que aparece num documento, mas é normalizado/balizada (*offset*) pela frequência de palavras no conjunto de documentos.
Uma vez que o método se baseia fortemente na descrição para distinguir cada item, a descrição deve caracterizar o produto (*e.g.*, título, sumário, *tags*, géneros), para fornecer mais informação à cerca do item.

3.1.2 Sistemas baseados em Filtragem Colaborativa

Os autores (Resnick & Varian, 1997), usam a expressão, "filtragem colaborativa", *Collaborative Filtering Systems* (CF), para sistemas em que a filtragem da informação é realizada com a intervenção de pessoas (ver Figura 3-3). Este sistema foca-se na interação entre utilizador e item, ignorando as caraterísticas dos itens e preocupando-se em quais itens já foram consumidos pelo utilizador. São usadas métricas para compreender quais as preferências (*e.g.* os utilizadores avaliam o item, informação sobre a quantidade de vezes que acedeu ao item, quanto tempo interagiu com o item) dando a possibilidade de apresentar aos utilizadores recomendações inesperadas.
As técnicas de filtragem colaborativa criam uma matriz de utilizadores com os itens de preferências destes, armazenados numa Base de Dados. Posteriormente combinam os utilizadores com interesses e preferências relevantes, calculando semelhanças entre os seus perfis para fazer recomendações, criando um grupo chamado vizinhança (*neighborhood*). Os

utilizadores recebem recomendações de itens classificados por outros na sua vizinhança (Isinkaye et al., 2015).

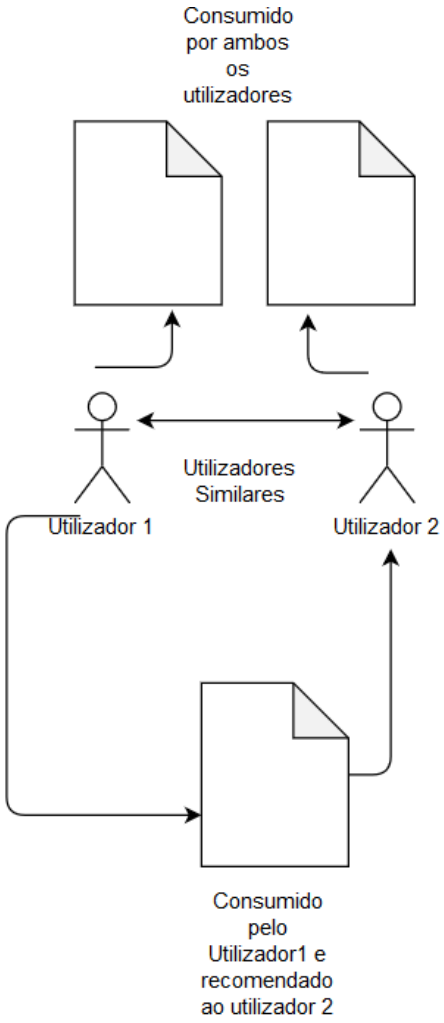

Figura 3-3: Sistemas baseados em Filtragem Colaborativa

Este tipo de filtragem tem vantagens relativamente à baseada em conteúdos, porque está apta a filtrar qualquer tipo de item, como texto, música, fotos e vídeos. No entanto, a aplicação deste método é computacionalmente dispendiosa, uma vez que, apresenta uma enorme complexidade no tratamento dos dados. A eficiência pode ser melhorada, se se diminuir a quantidade de dados dos utilizadores e se se reduzir o espaço de dados dos itens, considerando uma amostra aleatória para estudo (Lee Herlocker, 2000). Um dos problemas associados à diminuição da quantidade de dados é a diminuição da qualidade da recomendação, uma vez que, limita a recomendação a itens específicos e descartam os mais ou menos populares, diminuindo a similaridade entre os utilizadores (Ramos, 2010).

A filtragem colaborativa possui sérias limitações na qualidade das avaliações recomendadas, como o *sparsity problem*, o *cold start problem*, a escalabilidade, a fraude, a *black sheep*, a *gray sheep* e a sinonímia ou *synonymy* (Kim et al., 2010). Em (Z. Huang et al., 2004), os autores referem que o *sparsity problem* ocorre quando os dados transacionais ou de *feedback*

são escassos e insuficientes para identificar vizinhos, o que limita em geral a qualidade das recomendações e a aplicabilidade da filtragem colaborativa. Este problema acontece porque nem todos os utilizadores classificaram a grande maioria dos itens que viram ou adquiriram, o que pode influenciar a qualidade da recomendação. Pode-se colmatar este problema através da utilização da informação de perfil do utilizador ou item no cálculo da similaridade (*e.g.*, idade, profissão). Contudo estas aproximações são consideradas híbridas.

Os sistemas de filtragem colaborativa podem ser baseados em memória e baseados em modelo. As técnicas de filtragem colaborativa baseadas em memória, podem ser apoiadas em utilizadores ou itens. As que são apoiadas no utilizador, calculam a semelhança entre os utilizadores, comparando as suas classificações nos mesmos itens e calculam a classificação prevista para um item pelo utilizador ativo. Esse cálculo e classificação, são obtidos pela média ponderada das classificações dos itens, por utilizadores semelhantes ao utilizador ativo. Os pesos são as semelhanças entre utilizadores sobre os mesmos itens. As recomendações de filtragem colaborativa, baseadas em itens calculam previsões, usando a semelhança entre itens. A previsão é feita considerando uma média ponderada da classificação do utilizador ativo nos itens semelhantes. Vários tipos de medidas de similaridade são usados para calcular a similaridade entre item/utilizador, sendo as medidas de similaridade mais populares, baseadas em correlação[1] de acordo com a distância euclidiana ou similaridade do cosseno[2] e, também, baseada no coeficiente de correlação de *Pearson* (Isinkaye et al., 2015). Uma das técnicas mais utilizadas nos sistemas de recomendação baseados em memória é o kNN (*k-Nearest Neighbor*), isto é, os k vizinhos mais próximos. A definição de vizinhos mais próximos pode ser aplicada tanto à relação entre utilizadores como itens (Isinkaye et al., 2015).

As **técnicas baseadas em modelos**, utilizam a informação correspondente às classificações (*ratings*) do conjunto de dados de treino para criar um modelo estimado, usando técnicas de *Data Mining (DM)* ou *Machine Learning (ML).* Estes algoritmos fazem recomendações baseadas na estimação de parâmetros de modelos estatísticos/probabilísticos para os *ratings* dos utilizadores. Essas técnicas podem recomendar rapidamente um conjunto de itens, uma vez que utilizam um modelo pré-calculado, conseguindo resultados semelhantes às técnicas de vizinhança. Estas, agrupam os itens dos utilizadores numa matriz, na qual são identificadas as relações entre os itens, por forma a comparar a lista das principais recomendações (Isinkaye et al., 2015). Através das classificações dos utilizadores a itens é construído um modelo capaz de fazer predições. Esses modelos mapeiam utilizadores e itens através de redução da dimensão dos dados, onde ambos estão representados num mesmo conjunto de características ocultas (*e.g.*, características como género e faixa etária), aprendidos das

[1] Calcula a distância entre dois vetores que representam perfis de utilizador. O valor calculado é um número real no intervalo [0, ∞[e representa a similaridade entre os utilizadores. Quanto mais próximo de 0 mais similares são os perfis (A. Huang, 2008).

[2] Calcula o cosseno do ângulo formado pelos dois vetores que representam os perfis dos utilizadores. O valor do cosseno varia entre 0 (zero) a 1 (um), indicando a similaridade entre os utilizadores. Quanto mais próximo de 1, mais similares são os perfis (A. Huang, 2008).

classificações originais, logo diretamente comparáveis (Koren, 2010). Técnicas de redução de dimensionalidade, removem utilizadores ou itens não representativos ou insignificantes para reduzir diretamente as dimensionalidades da matriz de utilizador *vs.* itens (Su & Khoshgoftaar, 2009).

As abordagens baseadas em modelos agrupam diferentes utilizadores num pequeno número de classes com base nos padrões de classificações efetuados por estes. A classificação de um utilizador num determinado item é baseada na classificação da classe de utilizadores em que o utilizador ativo se encaixa melhor (Jin & Si, 2004).

A recomendação baseada em modelo tem sido pesquisada para tentar resolver alguns problemas decorrentes da filtragem colaborativa, como o problema de esparsidade e da escassez de dados (Cremonesi et al., 2010; Sharifi et al., 2013).

Uma das abordagens baseadas em modelo que lidam com o problema de esparsidade é a factorização matricial. Exemplo da aplicação dessa técnica é o modelo de regressão SVD (*Singular Value Decomposition*)[3], que é uma técnica algébrica de factorização de matrizes que tem como consequência a redução da dimensionalidade de um *dataset.* Esta redução é efetuada através da redução do número de recursos de um conjunto de dados, reduzindo a dimensão do espaço de dimensão N para dimensão K, sendo K<N (Cremonesi et al., 2010). Este método de decomposição de matrizes reduz uma matriz nas suas partes constituintes. Ao aplicar SVD a uma matriz de avaliações de utilizador *vs.* itens poder-se-á descobrir algumas características (*features*) dos utilizadores em relação à avaliação de itens (Alam & Fekpe, 2000; Barragáns-Martínez et al., 2010).

Os algoritmos de recomendação baseados em modelos usam outras aproximações que são descritas a seguir:

- Regras de associação: extraem regras que preveem a ocorrência de um item com base na presença de outros itens numa transação (Isinkaye et al., 2015).
- *Clustering*: tentam particionar um conjunto de dados em grupos (*sub-clusters)*, segundo o seu grau de semelhança. Após a formação dos *clusters*, as opiniões de outros utilizadores num *cluster* podem ser calculadas e usadas para fazer recomendações para utilizadores individuais. As técnicas de *clustering* foram aplicadas em diferentes domínios, como reconhecimento de padrões, análise estatística de dados e descoberta de conhecimento (Isinkaye et al., 2015).
- Árvores de decisão: a tomada de decisão é baseada na metodologia de árvores. A estrutura de uma árvore de decisão é composta por nós, ramos e folhas. Cada nó da árvore representa uma interrogação a uma variável do conjunto de dados. Os ramos correspondem a cada uma das respostas possíveis à interrogação criada, o que permite separar o conjunto de dados de acordo com alguns critérios de resposta. As folhas correspondem aos nós finais, nos quais já não existem mais interrogações possíveis (Fayyad et al., 1996). Estas árvores são treinadas de acordo com um conjunto de treino (exemplos previamente classificados) e posteriormente, outros exemplos são classificados de acordo com essas mesmas árvores (Isinkaye et al., 2015).

[3] Essa técnica foi usada na competição do Prémio da Netflix em 2009 (*Netflix Prize*, 2009).

- Redes neuronais artificiais: são constituídas por vários nós (neurónios) interligados e são inspirados no cérebro humano. Um neurónio artificial é uma função matemática que pretende modelar o comportamento do neurónio biológico. A ligação de cada nó tem pesos associados, dependendo da quantidade de influência que um nó tem sobre o outro. Possuem várias unidades de processamento, ligadas aos pesos associados, o que permite, um processamento altamente paralelo e distribuído (Isinkaye et al., 2015).
- Análise de *links*: utilizam os algoritmos *PageRank*, que mede a relevância de cada página web na Internet e *HITS (Hypertext Induced Topic Selection)*. Estes atribuem dois valores às páginas, um para os *links* que saem da página e outro para *links* que apontam para a página e é processado em tempo de consulta e não, em tempo de indexação (Kleinberg, 1999).
- Regressão linear: é um algoritmo de *Machine Learning*, que faz a modelação, da relação entre a entrada e saída. Utiliza essas duas variáveis (entrada e saída), para encontrar a melhor linha de ajuste para modelar essa relação. Para determinar se existe correlação entre duas variáveis é utilizado o coeficiente de correlação de *Pearson* (Isinkaye et al., 2015).
- Classificadores *Bayesianos:* o mais usado, é o classificador *Naive Bayes*. Estes classificadores, servem para classificar texto, *spam*, filtragem de *email*, por exemplo. A presença ou a ausência de um atributo específico não está relacionado com a presença ou ausência de outro (Isinkaye et al., 2015).

3.1.3 Sistemas baseadas em Filtragem Híbrida

Hybrid Filtering Systems (HF)**,** combinam técnicas de recomendação diferentes, com o objetivo de colmatar as falhas apresentadas por cada método, quando implementado individualmente. O sistema de filtragem híbrida pode ser implementado de várias maneiras, como por exemplo: fazendo previsões baseadas em filtragem de conteúdo e baseadas em filtragem colaborativa, separadamente, combinando as suas previsões; adicionando recursos baseados em conteúdo a uma abordagem colaborativa (e *vice-versa*) ou unificando as abordagens num modelo (Burke, 2002).

Previsões baseadas em filtragem de conteúdo e baseadas em filtragem colaborativa separadamente, podem ser implementadas, combinando os resultados (classificações) obtidos de cada sistema, numa recomendação final, usando uma combinação linear de classificações. Poder-se-á, também, utilizar recomendações individuais de cada sistema e usar o mais conveniente, com base na recomendação (Adomavicius & Tuzhilin, 2005).

Quando são adicionados recursos baseados em conteúdo a uma abordagem colaborativa, usam-se técnicas colaborativas tradicionais, em combinação com técnicas baseadas em conteúdos, para os utilizadores ativos. Os perfis baseados em conteúdo são para calcular a semelhança entre dois utilizadores, permitindo resolver o problema de escassez da abordagem colaborativa, uma vez que por norma, os pares de utilizadores não possuem um número significativo de itens com classificação comum. Nesta abordagem, os utilizadores podem ser

recomendados como um item, não apenas quando esse item é classificado positivamente (Adomavicius & Tuzhilin, 2005).

Os métodos híbridos que resultam da combinação de diferentes técnicas segundo (Burke, 2002), são os que se apresentam a seguir.

- O método *Weighted*, em que cada componente do sistema híbrido pontua um dado item, sendo que essas pontuações, são calculadas a partir dos resultados de todas as técnicas de recomendação disponíveis no sistema, ou seja, combina os resultados de diferentes recomendações para gerar uma lista ou previsão de recomendações, integrando as pontuações de cada uma das técnicas em uso por uma fórmula linear. Como exemplo do uso deste método há o trabalho sobre recomendação de recursos em sistemas de anotação social (Gemmell et al., 2012).
- O método *Switching*, em que o sistema muda os algoritmos de recomendação, dependendo da interpretação dos dados do sistema. Em (Ghazanfar & Prügel-Bennett, 2010) propuseram uma abordagem única de recomendação de híbridos de comutação combinando uma abordagem de classificação *Naive Bayes* com a filtragem colaborativa.
- O método *Mixed*, usado quando é necessário fazer um grande número de recomendações, simultaneamente, podendo ser usadas, várias técnicas em conjunto, evitando o problema de inicialização do *cold start* de item. Sobre este método há trabalhos como os de (Christakou et al., 2005) ou (Bostandjiev et al., 2012).
- O método *Feature Combination*, em que caraterísticas de diferentes técnicas de recomendação, são unificadas num único algoritmo (*e.g.* a entrada do *rating* produzido pelo sistema baseado em filtragem colaborativa num sistema baseado em conteúdo, é analisado como um item). Os autores (B.Thorat et al., 2015), fazem uma abordagem sobre SR Híbridos e a necessidade de combinar várias técnicas, como dados demográficos, conteúdo, filtros colaborativos e incorporação de informações sociais para que estes sistemas produzam recomendações relevantes.
- O método *Cascade*, onde a primeira técnica de recomendação gera uma lista aproximada de recomendações, que por sua vez é refinada pela próxima técnica de recomendação, ou seja, um algoritmo filtra as recomendações geradas por outro algoritmo (*e.g.* itens que tenham sido classificados com *ratings* idênticos pela primeira técnica, poderão vir a ser reclassificados pela segunda técnica do sistema). Os autores (Ghazanfar & Prugel-Bennett, 2010) propuseram uma abordagem única de recomendação híbrida em cascata combinando a classificação, o recurso e as informações demográficas sobre os itens.
- O método *Feature Augmentation*, onde uma primeira técnica de recomendação, é usada para produzir uma classificação ou classificação de um item, que em seguida é integrada como entrada no processo de recomendação da segunda técnica. Por exemplo, o sistema Libra (Mooney & Roy, 2000) faz recomendações baseadas em conteúdo de livros sobre dados encontrados na Amazon.com, empregando um classificador *Bayesiano* de texto.
- O método *Meta-level*, no qual o modelo aprendido é utilizado como um parâmetro de entrada de outro sistema. O modelo gerado é sempre mais rico em informações quando comparado a uma única classificação. São exemplo de estudo deste método, o trabalho teórico de (Schwab et al., 2004), que argumenta o uso da aprendizagem

instantânea para criar um perfil de utilizador baseado em conteúdo, que é comparado de maneira colaborativa. Um outro exemplo de trabalho, é o de (Zanker, 2008), onde o sistema aprende preferências baseadas em regras a partir de interações bem-sucedidas, em dados históricos de transações. Estes trabalhos usam filtragem colaborativa para derivar preferências dos vizinhos mais próximos de um utilizador, que são processados por um SR, baseado em conhecimento para derivar recomendações.

Este último método pode ser vantajoso se forem utilizadas técnicas de filtragem colaborativa e baseadas em conteúdo, uma vez que, a aprendizagem do sistema de recomendação baseado em conteúdo é feita a partir de uma representação compacta das preferências dos utilizadores. Por outro lado, o baseado em filtragem colaborativa, está apto para trabalhar com uma estrutura de dados densa.

3.1.4 Sistemas baseados em Dados Demográficos

A abordagem demográfica, baseia-se no pressuposto de que diferentes nichos demográficos têm gostos diferentes em itens. O sistema, recomenda itens aos utilizadores, com base nos seus perfis demográficos, tendo em conta, por exemplo, o idioma, país, idade, género e localização (Ricci et al., 2011). As sugestões podem ser personalizadas de acordo com esses dados. É provável que os interesses dos utilizadores variem no tempo e no espaço (Balabanovic, 1997).
Estes sistemas, são normalmente combinados com outros SR, originando sistemas híbridos. As recomendações são criadas utilizando um mecanismo de seleção que tem em conta o perfil demográfico do utilizador. A utilização de dados demográficos mostra-se eficaz como meio para aumentar a performance dos sistemas de recomendação (Burke, 2002; Gupta & Gadge, 2015).

3.1.5 Sistemas baseados em Conhecimento

Os sistemas baseados em conhecimento, replicam e aplicam, autonomamente, a experiência humana. Para esses sistemas, a engenharia do conhecimento fornece a tecnologia para converter o conhecimento humano em energia industrial. Dessa maneira, a engenharia do conhecimento acelerará o desenvolvimento, o esclarecimento e expansão do conhecimento humano. Tenta entregar sugestões ao utilizador com base no conhecimento acerca da necessidade de um utilizador para com um determinado item (Gennari et al., 2003). O conhecimento do SR pode assumir a forma de dados ou pode conter uma ontologia de domínio, que representa um conjunto de conceitos e relacionamentos (modelo de dados). A natureza da base de conhecimento e a estratégia de recomendação estão intimamente relacionadas e influenciam-se mutuamente (Bouraga et al., 2014).

3.1.6 Sistemas de Recomendação Comunitários

Atualmente, alguns SR usam as redes socias dos utilizadores como informação adicional para sugerir recomendações de itens (Pham et al., 2011). Este tipo de sistema, aglomera dados referentes às classificações e recomendações de itens, por parte de utilizadores, que fazem parte de uma mesma rede ou comunidade. Identificam semelhanças entre utilizadores, através da comparação das suas classificações, fazendo novas recomendações, que são geradas por comparação de perfis de utilizadores dentro da mesma rede ou comunidade. Os resultados experimentais mostram que a incorporação de informações sociais contextuais, podem ajudar a melhorar a qualidade da previsão, especialmente quando os dados de treino disponíveis são escassos (Ma et al., 2011).
Existem diferenças entre a recomendação comunitária e a filtragem colaborativa. A diferença reside na abrangência da "vizinhança" de utilizadores, utilizada para calcular as recomendações. No caso da recomendação colaborativa são tidos em conta apenas as classificações de itens de utilizadores próximos ("amigos") do utilizador em questão, enquanto na recomendação comunitária, em vez de usar dados de classificação, é utilizado o relacionamento social entre os utilizadores para identificar a vizinhança. Em (Pham et al., 2011) procuram melhorar as técnicas tradicionais de filtragem colaborativa, usando a técnica de *clustering* de rede, aplicada na rede social, para encontrar grupos de utilizadores semelhantes. Posteriormente, aplicam-se os algoritmos tradicionais de filtragem colaborativa, para gerar, com eficiência, as recomendações. Nas suas experiências, com dois conjuntos de dados reais, mostraram que o método de *cluster* combinado com a filtragem colaborativa, supera algoritmos de filtragem colaborativa tradicionais.

3.2 Trabalhos relacionados

O ***Tapestry***, foi um dos primeiros sistemas de recomendação comercial, desenvolvido no centro *Xerox*, que utilizava algoritmos colaborativos e de conteúdo na filtragem de *emails*, recebidos por um conjunto de utilizadores que pertenciam a uma *mailing list.* O sistema *Tapestry* para funcionar, continha um conjunto de anotações feitas pelos utilizadores às mensagens recebidas (classificando-as com "bom" ou "mau" ou através de um texto de opinião). As mensagens eram armazenadas numa base de dados e podiam ser consultadas a partir do seu conteúdo ou a partir da opinião de outros utilizadores. Neste sistema não existia um mecanismo capaz de agrupar os utilizadores por interesses similares, daí necessitar da interação do utilizador (Goldberg et al., 1992).
O ***GroupLens,*** faz parte do departamento de Engenharia e Ciência da Computação da Universidade de Minnesota, dedicando-se ao estudo de várias áreas, como sistemas de recomendação e comunidades *online.* Desenvolveram vários projetos na área dos algoritmos colaborativos e sistemas de recomendação. O primeiro sistema de recomendação surge em 1994, sendo baseado em filtragem colaborativa. A finalidade deste sistema era a procura e

recomendação de novos artigos dos fóruns, grupos de notícias, presentes no serviço *Usenet* (*RevistaUsenet.Com*, 2019). O algoritmo utilizado pelo *GroupLens*, estudava a relação entre um conjunto de utilizadores que possuíssem gostos semelhantes entre si. Atualmente o *GroupLens*, tem o sistema *MovieLens* em funcionamento. O *MovieLens* é um sistema de recomendação de filmes que também utiliza filtragem colaborativa, como técnica de recomendação. As recomendações de filmes são baseadas nos *ratings* atribuídos pelos utilizadores aos filmes. O sistema realiza várias sugestões à medida que vai "aprendendo" os gostos dos utilizadores. Se o utilizador for novo no sistema, o serviço faz uma sugestão de filmes baseada na lista de filmes com melhores classificações (GroupLens, 2020a; Resnick et al., 1994).

O ***Ringo***, foi desenvolvido no *MIT* (*Massachusetts Institute of Technology*) em 1995, por um grupo de investigadores. Era um serviço de recomendação de música. Neste sistema as pessoas indicavam as suas preferências musicais através da avaliação de canções ou álbuns, criando assim um perfil de utilizador, mas mantendo a identidade secreta de quem avaliou. Com esta informação o sistema *Ringo* agrupava os utilizadores com preferências musicais próximas e gerava recomendações com base nos *ratings* atribuídos. Para o caso dos novos utilizadores ou daqueles que ainda não tinham classificado nenhuma música, o sistema recomendava o *top* de músicas/álbuns mais votados (Cazella et al., 2010).

Atualmente existem serviços semelhantes, como é o caso do ***Last.fm***, que é um serviço que permite aos utilizadores ouvirem as suas músicas preferidas ao mesmo tempo que constroem uma base de dados, personalizada, de preferências musicais dos utilizadores. A personalização desta é feita à medida que os utilizadores vão indicando se gostam ou não duma música que lhes é apresentada (Last.fm, n.d.).

O ***PHOAKS*** (*People Helping One Another Know Stuff*)**,** é um sistema de filtragem colaborativa, que reconhece, calcula e redistribui automaticamente as recomendações de recursos da web, extraídos das mensagens de notícias da *Usenet* (*RevistaUsenet.Com*, 2019) e comparando-as com as preferências dos diferentes grupos de notícias. Esta comparação é feita a partir da contagem de páginas web mais importantes para os leitores daquele grupo de notícias (Terveen et al., 1997).

O ***Fab,*** é um sistema desenvolvido na Universidade de *Satanford*, em 1994, que combina a filtragem colaborativa com os sistemas baseados em conteúdos, sendo considerado um primeiros sistema híbrido (Burke, 2002). Foi criado para ajudar os utilizadores a lidar com o aumento de informação disponível na Internet, gerando recomendações de documentos (páginas web), através da análise do seu conteúdo com o de documentos previamente classificados com cotação elevada. No sistema híbrido *Fab*, os perfis dos utilizadores são criados a partir da análise de conteúdos. Os perfis são comparados entre si, identificando outros utilizadores com caraterísticas de perfil similar (Balabanovic, 1997).

O sistema baseado em conhecimento, ***Oncocin***, desenvolvido na Universidade de *Stanford*, é um sistema de aconselhamento para terapia do cancro, baseada em protocolo, onde as informações sobre a história de um doente específico, podem ser inseridas por um utilizador do domínio (um médico ou enfermeiro). O sistema fornece conselhos sobre tratamentos e

testes. Posteriormente, foi desenvolvida a ***Opal***, que usa conceitos e ideias específicas do domínio, para apresentar aos especialistas em cancro, formulários cuidadosamente projetados, para entrada de dados estruturados. Em vez de digitar regras de produção individuais, permitia aos médicos, descrever protocolos completos do cancro, preenchendo formulários gráficos para fins especiais (Gennari et al., 2003). Estes dois sistemas evoluíram para o ***Protégé***, que atualmente, é uma plataforma gratuita e de código aberto, com ferramentas para construir modelos de domínio e aplicações baseadas em conhecimento com ontologias, que são um modelo de dados que representa um conjunto de conceitos dentro de um domínio e os relacionamentos entre estes (*Protégé - A Free, Open-Source Ontology Editor and Framework for Building Intelligent Systems*, 2020).

O ***TheFork*** (*TheFork - Reserve Nos Melhores Restaurantes Da Europa*, n.d.)**,** é um sistema Português de pesquisa e reserva de mesas em restaurantes, que surgiu em 2011, com um *site* web e em fevereiro de 2012, como aplicação móvel em exclusivo, para os dispositivos móveis da *Apple*. Atualmente, também está disponível para *Android*. É um sistema que, durante algum tempo, só tinha restaurantes portugueses, mas atualmente já abrange dezasseis países. Este sistema faz a gestão, de forma integrada e automática, dos lugares disponíveis nos restaurantes aderentes. Os restaurantes são apresentados com informação detalhada, incluindo o menu completo e preços. Os utilizadores do sistema, fazem a avaliação dos restaurantes, através do sistema *TripAdvisor* (TripAdvisor, 2020).

A ***Amazon.com*** utiliza algoritmos colaborativos para personalizar a loja *online* com as preferências de cada cliente, o que significa, que o aspeto da loja *online*, é alterado de acordo com os interesses de cada cliente. No *site* da Amazon na seção "*Your Recommendation*", poder-se-á filtrar as recomendações por áreas de produtos, avaliar os produtos recomendados, as compras efetuadas e ver a relação entre os itens, (*e.g*. a *Amazon.com*, sugere ao utilizador uma série de livros ainda não lidos com base nos itens anteriormente vistos ou adquiridos) (Amazon.com, n.d.)**.**

O *site* de leilões *online* ***eBay.com,*** possui estratégias de recomendação, apresentadas no artigo (Schafer et al., 1999), sobre sistemas de recomendação em *e-Commerce*, permitindo aos compradores e vendedores avaliarem o seu parceiro de negócio, de acordo com o grau de satisfação da compra. Permite ainda, aos clientes indicarem os itens que têm interesse em comprar (EBay, n.d.).

As **Redes Sociais** como *Facebook*, *Instagram* e *Twitter*, utilizam os sistemas de recomendação que filtram o tipo de conteúdo que os utilizadores gostariam de ver, como novos amigos, publicações, aplicações e tendem a evidenciar os conteúdos mais relevantes dos utilizadores (Yu, 2012). Inicialmente, as redes sociais não usavam algoritmos de recomendação. O que era publicado em primeiro lugar, era o que era consumido primeiro, ou no fim de acordo com o fluxo da linha cronológica. Assim, parte da informação de interesse dos utilizadores era perdida. Atualmente, os algoritmos de recomendação utilizados em grande parte das redes sociais, são baseados em filtragem colaborativa e no conceito de vizinhança, utilizando o algoritmo *kNN* (*k-Nearest Neighbor*) que permite reconhecer

padrões, ou baseados em modelos para filtragem colaborativa (Jiang et al., 2007; Ma et al., 2011).

Segundo informação do *site* web da ***Netflix*** (*Sobre a Netflix*, n.d.), *Reed Hastings* e *Marc Randolph*, criaram a *Netflix*, em 1997, como uma empresa de aluguer de filmes *online.* No ano a seguir é lançada um *site* web, de venda e aluguer de DVDs. Posteriormente cria um serviço de subscrição, com alugueres ilimitados a um preço mensal acessível. No ano 2000, introduzem um sistema personalizado de recomendação de filmes, que se baseia nas classificações dos utilizadores para prever, com precisão, os títulos que os membros da *Netflix* vão adorar. A importância de um bom sistema de recomendação foi reconhecida pela *Netflix*, o que levou ao anúncio do concurso *Netflix Prize*, para motivar os investigadores a melhorar a precisão do sistema de recomendação da *Netflix* (Takács et al., 2009). No seguimento deste concurso disponibilizou uma base de dados, que é considerada a maior base de dados disponível para avaliação de algoritmos colaborativos.

Um dos maiores valores do ***Pinterest,*** é a capacidade de fazer recomendações visuais com base no gosto de centenas de milhões de utilizadores e, em seguida, ajudar as pessoas a descobrir ideias e itens que correspondem aos seus interesses. Com o crescimento dos utilizadores e dados, a tecnologia dos SR da aplicação, tem de estar em constante evolução, tornando as recomendações mais inteligentes. Em (Ying et al., 2018), são descritas as melhorias no sistema de recomendação do *Pinterest*, com o recurso a Redes Neuronais. Para a implementação em produção do *Pinterest,* foi desenvolvido um algoritmo *PinSage*[4] que aproveita várias informações importantes para melhorar a escalabilidade das *Graph Convolutional Networks*[5] (*GCN*), com treino de biliões de itens. O algoritmo *PinSage* gera recomendações de alta qualidade, com aprendizagem profunda comparável e alternativas baseadas em gráficos. Segundo os autores a aplicação gráfica, abre caminho para uma nova geração de escala na web, concretamente, SR baseados em arquiteturas *GCN.*

A ***Google***, utiliza vários algoritmos para fazer o posicionamento de páginas web. O primeiro algoritmo utilizado foi o *PageRank* que é um algoritmo de classificação, que determina a importância de uma página da web e sua posição hierárquica nos resultados do mecanismo de pesquisa. *Matteo Pasquinelli,* no seu artigo sobre o estudo do algoritmo *PageRank*, refere que este, introduziu uma rutura revolucionária nas tecnologias de recuperação de informação e nos motores de busca, no final dos anos 90, permitindo que a enorme quantidade de dados fosse modelada pelo *Google* em hierarquias dinâmicas, de acordo com o visibilidade e importância de cada *site*. O *ranking* de uma página da web, é assim determinado, não apenas pelo número de *links* recebidos, mas também pelo tipo de *links* que recebe. Um *link* proveniente de um nó com uma classificação alta tem mais valor que um *link* proveniente de um nó com uma classificação baixa. A fonte de inspiração para o *PageRank,* foi o sistema de citações

[4] Sistema de Recomendação que desenvolve incorporações de nós de alta qualidade usando a estrutura do gráfico e as informações de recursos do nó.

[5] Rede Neuronal utilizada para processamento e análise de imagens digitais.

académicas, em que o "valor" de uma publicação académica, é calculado de acordo com o número de citações que um artigo recebe de outros artigos (Pasquinelli, 2009).

Atualmente a *Google* usa o *Knowledge Graph* (Grafos de conhecimento) em simultâneo com os algoritmos de classificação. Segundo a *Google* (*Official Google Blog: Introducing the Knowledge Graph: Things, Not Strings*, n.d.), "os Grafos de conhecimento, permitem pesquisar coisas, pessoas ou lugares que o *Google* conhece (*e.g*. marcos, celebridades, cidades, equipas desportivas, edifícios, características geográficas, filmes, objetos celestes, obras de arte e muito mais) e obter instantaneamente informações relevantes".

Por fim e no domínio académico, para a obtenção de graus académicos equivalentes, serão descritas, a seguir, duas dissertações.

A dissertação de Mestrado (Reis, 2012), teve como objetivo criar um sistema baseado em conhecimento. O sistema é capaz de recolher, categorizar e filtrar notícias, autonomamente, de acordo com os interesses noticiosos específicos dos utilizadores. Recorre a aprendizagem computacional, o que permitiu identificar o perfil dos utilizadores, combinando o *feedback* destes, com conhecimento produzido, a partir do conteúdo das notícias, sendo possível gerar recomendações. Paralelamente, ao trabalho de dissertação, foi desenvolvida uma *interface* utilizando a plataforma *Google Android*, que admitiu recolher informação, a partir de diversas fontes noticiosas, classificar a informação recolhida através de palavras-chave; permitir ao utilizador selecionar os tópicos de interesse da sua preferência; aprender os interesses do utilizador, à medida que este vai lendo e classificando notícias; entregar informação personalizada e relevante a cada utilizador, usando métodos de extração de informação e construção de estruturas de conhecimento a partir de fontes não estruturadas (categorização de notícias e a identificação de tópicos de interesse).

Em (Marcelino, 2014), é usando SR baseados em Filtragem Colaborativa. Nesta é relatado o estudo comparativo entre algoritmos baseados em memória e em modelo, cuja implementação foi desenvolvida em linguagem *Python*, utilizando os conjuntos de dados do *MovieLens* e da *Netflix*.

4. Metodologias

Este projeto foca-se em técnicas de *Data Science (DS)*, através da análise de dados e extração de informação, representação de conhecimento e aprendizagem computacional, com o objetivo de construir um SR capaz de associar conteúdos a utilizadores.
Neste capítulo falar-se-á das fases do processo de recomendação, na formulação do modelo e nas medidas de similaridade. Haverá, ainda uma abordagem as metodologias da literatura para SR utilizadas, são as de: Filtragem Colaborativa Simples, Filtragem de Conteúdo, aplicando o TF-IDF, assim como a metodologia de Filtragem Colaborativa Baseada em Memória**,** usando a técnica *kNN*. Através da combinação destas metodologias criou-se um SR Híbrido Ponderado, utilizado o método *Weighted* que atribui um determinado peso à Filtragem de Conteúdo e outro à de Filtragem Colaborativa. Por fim será apresentada uma descrição das métricas de avaliação empregadas para a avaliação dos algoritmos implementados.

4.1 Fases do processo de recomendação

Os SR dependem, em grande parte da quantidade de informação que se obtém dos utilizadores, para assim, fornecerem recomendações de interesse. Essas informações sobre os utilizadores, podem ser adquiridas de forma explícita, através da perceção do seu interesse no item, ou por *feedback* implícito, inferindo preferências do utilizador indiretamente. As fases do processo de recomendação podem ser constatadas na Figura 4-4 (Isinkaye et al., 2015).

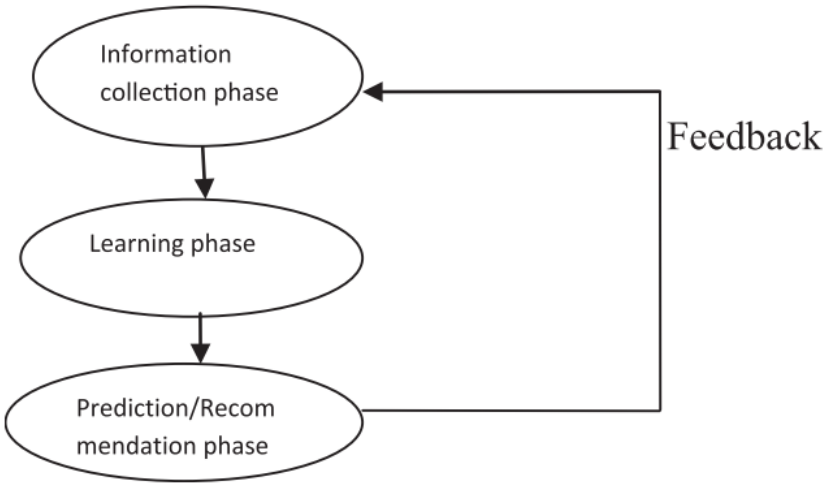

Figura 4-4: Fases de Recomendação

Em seguida falar-se-á da importância da recolha de informação e da forma como esta é efetuada, assim como da aprendizagem supervisionada e não supervisionada, por fim haverá uma abordagem das fases de previsão e recomendação,

4.1.1 Recolha de informação

Um SR recolhe informação relevante de utilizadores para gerar um perfil ou modelo de utilizador para a tarefa de predição, incluindo atributos do utilizador, comportamentos ou conteúdo dos recursos (itens) acedidos pelo utilizador (Isinkaye et al., 2015).
O processo de recolha de informação será designado por *feedback* do utilizador perante o sistema. Este poderá ser explícito ou implícito.
O ***feedback* explícito** é visto como um processo mais confiável para a extração dos dados, uma vez que não envolve extrair preferências de ações, fornecendo transparência às recomendações. Isto é, é recomendado o que realmente é de interesse do utilizador (Papagelis & Plexousakis, 2005).
Assim, o sistema solicita ao utilizador, através da sua *interface*, classificações para itens de forma a construir e melhorar seu modelo. A precisão da recomendação depende da quantidade de classificações fornecidas pelo utilizador. Uma classificação explícita identifica a preferência de um utilizador num item específico, a partir de uma pequena quantidade de dados (*e.g*. dados binários do tipo gosto, não gosto; questionários; dados numéricos numa escala; análise de um texto de opinião).
Os *ratings* explícitos são os que melhor servem os interesses dos sistemas de recomendação, mas ao requerem a participação dos utilizadores no processo de classificação dos itens, por vezes falham. No entanto, nem todos os utilizadores estão dispostos a classificar itens, tornando-se uma desvantagem deste método, ao exigir um esforço dos utilizadores. Uma outra desvantagem, é que os utilizadores, nem sempre estão dispostos a fornecer informações (Takács et al., 2009).

Nestas situações, os SR tornam-se menos precisos, uma vez que, o sistema não consegue fazer uma recomendação fiável, porque não possui informação necessária sobre as preferências do utilizador. Nesse caso, alguns sistemas recolhem a informação sobre os interesses dos utilizadores na forma de *feedback* implícito.

No ***feedback* implícito** o sistema deduz os gostos dos utilizadores com base no histórico, no tempo de permanência, por exemplo, numa página *web* ou, ainda, por padrões de pesquisa. O sistema "aprende" com a interação do utilizador através de: padrão de cliques, tempo gasto a ver um item, gestos do rato, escrita no teclado e padrões de navegação. (Papagelis & Plexousakis, 2005). A análise implícita da informação pode ser imprecisa (*e.g.*, o fato de um utilizador clicar numa notícia não significa que a leia), se o sistema realizar sugestões com base nos cliques das notícias poderá recomendar de forma errada, outras notícias que não fazem parte dos interesses desse utilizador (Isinkaye et al., 2015).

A forma como é recolhida a informação dos utilizadores, bem como a forma como a informação é disponibilizada, influencia o processo de construção do modelo de utilizador. Esse modelo pode permanecer inalterado, ou ser-lhe adicionado dados demográficos. Contudo a maioria dos SR utilizam técnicas dinâmicas de aprendizagem do modelo de utilizador, atualizando e construindo de forma incremental o perfil do mesmo.

Como resultado, é fundamental um pré-processamento dos dados recolhidos do *dataset*, com a preparação destes para ser possível converter dados brutos em **dados limpos** e úteis para análise. Os dados brutos, muitas vezes são imprecisos, incorretos e inconsistentes, daí a necessidade deste **pré-processamento dos dados**.

Para testar o desempenho futuro do SR e estimar erros de previsão, há o particionamento do conjunto de dados limpos original, em **subconjuntos de treino e teste**, numa proporção adequada. O conjunto de teste deve ser diferente e independente do conjunto treino, para se obterem estimativas confiáveis de erros.

4.1.2 Aprendizagem

As técnicas mais utilizadas para a criação de um **modelo do perfil de utilizador** podem ser estatísticas, onde é analisado, estatisticamente, um conjunto de informação que se conhece relativamente a um utilizador ou baseadas em aprendizagem computacional (Papagelis & Plexousakis, 2005).

Através de métodos estatísticos, obtidos pela análise do comportamento de outros utilizadores com comportamento similar ou de *Data Mining*, consegue-se identificar o perfil de um utilizador, podendo ser necessário construir um modelo preditivo antecipadamente.

O sucesso da **inferência de informação contextual** (prospeção e extração de conhecimento a partir dos dados) depende significativamente da qualidade deste modelo preditivo. Existem duas abordagens principais para o processo de inferência ou aprendizagem: aprendizagem não supervisionada e supervisionada (Palmisano et al., 2008). Cada uma destas técnicas de aprendizagem contém um vasto conjunto de algoritmos que permitem a construção de modelos.

Na **aprendizagem supervisionada** são usados vários algoritmos (*e.g.* árvores de decisão, redes neuronais) e na aprendizagem não-supervisionada (*clustering* e regras de associação). A aprendizagem supervisionada é usada quando existe uma predição a realizar (classificação ou regressão).
Na **aprendizagem não-supervisionada** não há o atributo de *output.* Desconhece-se a classificação a cada observação do conjunto de dados e pretende-se descobrir padrões de semelhança desconhecidos entre os dados. O objetivo passa por agrupar os dados através de uma determinada medida de semelhança (Insight, 2014).
Os algoritmos utilizados permitem agrupar dados, com atributos semelhantes, tendências, padrões ou relacionamentos que ocorrem naturalmente nos dados. Incluem técnicas de agrupamento de *clustering*. Algoritmos diferentes usam estratégias diferentes para dividir dados em grupos (Learning et al., 2001).

4.1.3 Fase da previsão/recomendação

Esta fase fornece uma recomendação ou previsão dos itens, ainda não classificados, que o utilizador poderá preferir. Esta poderá ser feita com base no *dataset* resultante da fase recolha de informação, designado baseado em memória; da fase de aprendizagem, designado baseado em modelo (ambas técnicas *off-line*) ou, ainda, *online*, através do sistema a observar a atividade do utilizador.
A tarefa de recomendação é gerar uma lista de preferências com os itens com melhores pontuações (*scores*), enquanto a tarefa individual de estimar a avaliação (*rating*) para um único item é designada como previsão. Todo o processo pode ser constatado na Figura 4-4.

4.2 Definição/Formulação do modelo

Em termos formais, num sistema de filtragem existe uma lista de *n* utilizadores $U = \{u_1, u_2, .., u_n\}$ e uma lista de *m* itens $I = \{i_1, i_2, .., i_m\}$. Cada utilizador u_i possui uma lista de itens I_{u_i}, em que o utilizador expressou as suas preferências ou gostos. $I_{u_i} \subseteq I$ e o conjunto I_{u_i} pode ser vazio. Existe um utilizador $u_a \in U$, chamado utilizador ativo, para o qual a tarefa de recomendação ou previsão é realizada. Seja ainda r a função utilidade que mede a utilidade/relevância de um item i_j para um determinado utilizador u_i, representada por: $r: U \times I \rightarrow \mathbb{R}$, em que $\mathbb{R}$ é o conjunto real. A função de otimização é dada por: $\forall\, u_a \in U,\ i'u_a = arg_{i \in I}\, max\, r(u_a, i)$. O sistema de recomendação pretende escolher para um utilizador $u_a \in U$ um item $i' \in I$ que maximiza a utilidade do mesmo para o utilizador (Adomavicius & Tuzhilin, 2005; Sarwar et al., 2001).
A Previsão é um valor numérico, $P_{a,j}$, que expressa a pontuação ou avaliação do item $i_j \notin I_{u_a}$ para o utilizador ativo u_a (potencial interesse). A Recomendação, é uma lista de *N* itens $I_r \subset I$ pelos quais o utilizador poderá apreciar mais. Esta forma de recomendação é conhecida como recomendação *Top-N* e a recomendação deve ser de itens ainda não

consumidos pelo utilizador u_a (Cremonesi et al., 2010; Mukund & George, 2004; Sarwar et al., 2001).
O *dataset* utilizador *vs.* item é utilizado nos algoritmos de recomendação, como uma matriz de classificações *R* (*ratings*), com a dimensão $n\ x\ m$ (matriz utilizador-item). Cada entrada $r_{i,j}$, representa a avaliação (*rating*) do utilizador *i* para o item *j* (Sarwar et al., 2001).
Cada lista I_u corresponde às avaliações do utilizador *u* sobre os vários itens. Cada lista U_i corresponde às avaliações dos vários utilizadores sobre o item *i*. Com base nesta representação matricial, é possível calcular a similaridade entre utilizadores, calculando as respetivas medidas sobre as linhas da matriz, ou a similaridade entre itens, calculando as respetivas medidas sobre as colunas da matriz.

4.3 Medidas de similaridade

A medida de similaridade permite quantificar a semelhança entre utilizadores ou itens. A escolha da medida de similaridade tem um impacto direto na qualidade do sistema de recomendação (Sarwar et al., 2001). Quanto mais dados são usados para comparar a opinião dos utilizadores, mais confiável será o cálculo da similaridade entre eles. (Goldberg et al., 1992). As duas medidas para calcular a similaridade entre utilizadores/itens mais populares são as baseadas no coeficiente de correlação de *Pearson* e na similaridade do cosseno. O primeiro método normalmente é utilizado quando os ratings dos utilizadores estão disponíveis numa escala numérica entre 1 a 5 (Su & Khoshgoftaar, 2009) por exemplo, em sites como *Netflix*, *MovieLens* e *Amazon.com*. Nestes o utilizador é instigado a avaliar determinado filme ou produto utilizando uma notação de estrelas (5 estrelas) que representam valores entre péssimo e excelente. Já o método de similaridade de cosseno, é mais utilizado para cálculos de similaridade quando é uma avaliação 0 ou 1 (binária) que corresponde a gostei ou não gostei.
Uma outra medida é a distância de Euclides que mede a distância entre dois pontos no espaço bidimensional ou multidimensional.

4.3.1 Coeficiente de Correlação de *Pearson*

O coeficiente de correlação de *Pearson* é usado para medir até que ponto duas variáveis se relacionam linearmente entre si. O valor da correlação encontra-se entre -1 e 1. Esse valor, dentro de um SR, indica o quanto o utilizador ativo u_a combina com cada um dos outros utilizadores nas avaliações que fizeram em comum. É necessário isolar todos os itens que foram avaliados pelo utilizador ativo e restastes utilizadores (Falk, 2019).
A função é definida pela equação (4-1) e calcula a similaridade entre o utilizador *u* e a sua vizinhança *v*.

$$sim(u,v) = \frac{\sum_{i \in I_u \cap I_u}(r_{u,i} - \bar{r}_u) \times (r_{v,i} - \bar{r}_v)}{\sqrt{\sum_{i \in I_u \cap I_u}(r_{u,i} - \bar{r}_u)^2} \times \sqrt{\sum_{i \in I_u \cap I_u}(r_{v,i} - \bar{r}_u)^2}} \quad (4\text{-}1)$$

A variável $\bar{r}_u$ é a média das classificações do utilizador *u*, $\bar{r}_v$ é a média das classificações do utilizador *v*. $r_{u,i}$ corresponde à nota dada pelo utilizador *u* ao item *i*.

Utilizador/Item	Itens 1	Itens 2	Itens 3
Utilizador X	2	4	3
Utilizador Y		1	2
Utilizador Z		1	3

Quadro 4-1: Matriz utilizador-item

Considere o Quadro 4-1. A média do utilizador X, $\bar{r_X}$ será $(2 + 4 + 3)/3 = 3$ e a média do utilizador Y, $\bar{r_Y}$ será $(1 + 2)/2 = 1{,}5$. Considerando o utilizador X como o utilizador ativo, então, para calcular a similaridade entre o utilizador X e o utilizador Y, procede-se da seguinte forma:

$$sim(X,Y) = \frac{(4-3) * (1-1{,}5) + (3-3) * (2-1{,}5)}{\sqrt{((4-3)^2 + (3-3)^2)} * \sqrt{((1-1{,}5)^2 + (2-1{,}5)^2)}} = \frac{-0{,}5}{0{,}7} = -0{,}71$$

O intervalo do coeficiente indica se caso o valor seja -1 que o relacionamento entre os dois objeto será linearmente negativa, se for 0, indica que não há relacionamento, e +1, indica um relacionamento positivo entre os utilizadores (Sarwar et al., 2001).

4.3.2 Similaridade do Cosseno

Os utilizadores ou itens são representados por vetores para se calcular a similaridade. Cada vetor do utilizador possui os ratings correspondentes aos itens que classificou. No caso dos itens, cada vetor representa o conjunto de utilizadores que classificaram esse item (Sarwar et al., 2001). Os elementos do vetor são maiores que zero, para expressar uma avaliação já efetuada e zero, para avaliações ainda não efetuadas.

A medida de similaridade do cosseno entre os ratings do utilizador ativo e os restantes utilizadores é representada pela medida do ângulo entre eles.

A similaridade do cosseno entre um utilizador *u* e outro utilizador *v*, é dada pela equação (4-2):

$$cos(\overrightarrow{r_u}, \overrightarrow{r_v}) = \frac{\overrightarrow{r_u} . \overrightarrow{r_v}}{\|\overrightarrow{r_u}\|_2 \times \|\overrightarrow{r_v}\|_2} \quad (4\text{-}2)$$

A similaridade de cosseno é sempre um número não negativo uma vez que os ratings são sempre números não negativos. A medida de similaridade varia entre 0 (fraca correlação) e 1 (correlação forte).
Para o cálculo da similaridade de cosseno entre os utilizadores *u* e *v* utiliza-se a equação (4-3):

$$sim(u, v) = \frac{\sum_i r_{u,i} \times r_{v,i}}{\sqrt{\sum_i (r_{u,i})^2} \times \sqrt{\sum_i (r_{v,i})^2}} \quad (4\text{-}3)$$

4.3.3 Distância Euclidiana

A distância euclidiana mede a distância entre dois pontos no espaço bidimensional ou multidimensional. Quanto mais próximos estiverem os pontos no espaço de referência, mais semelhantes serão. A distância é calculada com base na raiz quadrada da soma das distâncias quadradas de cada dimensão (A. Huang, 2008).
Na distância entre dois vetores que representam os perfis do utilizador, quanto menor for a distância de um vetor a outro, mais semelhantes são os utilizadores. O valor da distância varia de 0, maior similaridade, a 1, menor similaridade. A equação (4-4) define a distância euclidiana.

$$W_{u,v} = \sqrt{\sum_i (u_i - v_i)^2} \quad (4\text{-}4)$$

$W_{u,v}$ representa a distância entre o utilizador ativo *u* e outro utilizador *v*, u_i é a avaliação que o utilizador *u* deu para o item *i*, v_i, é a avaliação do outro utilizador para o mesmo item (A. Huang, 2008).
Neste caso a *sim(u, v)* é dada na ordem inversa da distância. Quanto menor a distância maior a similaridade.

4.4 Filtragem Colaborativa simples

Permite resolver o problema *cold start* de novos utilizadores do sistema, que ainda não fizeram qualquer classificação. A ideia fundamental é que normalmente os itens mais populares "pela crítica" terão maior probabilidade de serem apreciados pela maioria do público (Sharma, 2020).

Para o efeito é prevista uma classificação ponderada para cada item, (previsão da classificação a atribuir com base na popularidade). Por exemplo, no caso de classificações de filmes, a equação utilizada pelo *IMBb*[6] para a definição do "*Top Rated 250 titles*". Esta fórmula fornece uma verdadeira estimativa "*Bayesian*", que leva em consideração o número de votos que cada título recebeu, os votos mínimos necessários para estar na lista e a média de votos para todos os títulos (*IMDb Help Center - Ratings FAQ*, 2021). A previsão da classificação deste método de média ponderada (*Weighted Rating*) é a constante na equação (4-5).

$$WR = \left(\frac{v}{v+m} \times R\right) + \left(\frac{m}{v+m} \times C\right) \tag{4-5}$$

Na equação acima,

- *v* é o número de votos no filme (votos);
- *m* é o número de votos mínimos para ser considerados na lista de "*Top Rated"*;
- *R* é a avaliação média do filme (rating);
- *C* é a média de todas as avaliações médias.

A escolha do valor de *m* implica a remoção dos itens que têm um número de ratings menor que um determinado limite *m*. Por exemplo, se utilizarmos o percentil 90 do número de ratings por filme, ou seja, para que um filme seja considerado, ele deve ter mais votos que, pelo menos 90% dos filmes na lista. Por outro lado, se escolhermos o percentil 75, seriam considerados os 25% dos filmes mais populares, em termos de número de ratings. À medida que o percentil diminui, o número de filmes considerados aumenta. Atualmente, no caso do *IMDb* "*Top Rated 250 titles*", o valor de *m* é 25 000 (só os títulos com mais de 25 000 votos, são considerados).
Com o cálculo da popularidade de todos os itens, obtém-se uma recomendação *Top-N*. Este método não é sensível aos interesses e gostos de utilizadores específicos.

4.5 Filtragem Baseada em Conteúdo

Nos métodos de recomendação CBF, um item é recomendado ao utilizador com base nas características ou atributos de outros itens. As características em questão são do item, não sendo consideradas classificações atribuídas por outros utilizadores. Estes permitem ultrapassar o problema *cold start* de novos utilizadores e novos produtos no sistema. Para

[6] Base de dados online de informações relacionadas com filmes, programas de televisão, vídeos caseiros, jogos de vídeo e *streaming* de conteúdo online, incluindo elenco, equipa de produção e biografias pessoais, resumos de enredos, curiosidades, classificações e análises de fãs e críticas (*https://www.imdb.com/*).

itens textuais, como artigos, notícias, livros ou descritivos de filmes e músicas, podem ser utilizados atributos como géneros/categorias ou mesmo texto descritivo, como fonte de extração de características ou atributos.

Como já foi referido, um dos métodos mais utilizados é baseado em *TF-IDF*. É usado na recuperação de informações para fins de extração de características ou atributos de um item. Em termos simples, permite contar a ocorrência de cada palavra (característica) num documento (item) e ponderar a importância relativa, calculando uma pontuação (*score*) para esse documento.

TF (*Term Frequency*), é a frequência de uma palavra no documento atual em relação ao número total de palavras desse mesmo documento, conforme a equação (4-6).

$$TF_{t,d} = \frac{Frequência\ de\ ocorrência\ do\ termo\ t\ no\ documento\ d}{Número\ total\ de\ termos\ no\ documento} \qquad (4\text{-}6)$$

Caso o mesmo termo *t* apareça várias vezes no mesmo documento *d*, é necessário amortecer o efeito dos termos de alta frequência. Para isso é calculado para cada termo o *Weighted Term Frequency* ($W_{t,d}$) pela equação (4-7).

$$W_{t,d} = \begin{cases} 1 + log_{10} TF_{t,d}, se\ TF_{t,d} > 0 \\ 0, de\ outra\ forma \end{cases} \qquad (4\text{-}7)$$

IDF (*Inverse Document Frequency*), número total de documentos, em relação à frequência de ocorrência de documentos que contêm a palavra (termo). Este, permite atribuir classificações maiores a termos raros nos documentos (ver equação (4-8)).

$$IDF_t = log_{10}\left(\frac{Número\ Total\ de\ documentos}{Número\ de\ documentos\ que\ contêm\ o\ termo\ t}\right) \qquad (4\text{-}8)$$

Assim o *TF-IDF* é a medida utilizada para avaliar a importância de um termo (característica) num documento (item) no *corpus* de documentos (todos os itens). A importância de um termo aumenta proporcionalmente ao número de vezes que uma palavra aparece num documento, mas é compensada pela frequência da palavra no *corpus* (equação (4-9)).

$$TF\text{-}IDF_{t,d} = W_{t,d} \times IDF_t \qquad (4\text{-}9)$$

Por exemplo, no caso da avaliação de filmes, os géneros ou o título podem ser considerados como caraterísticas descritivas dos filmes. Outras podem ser, elencos, sinopses, realizadores, anos de lançamento, etc.

Podem ser seguidas duas aproximações no desenvolvimento de um CBF (Luk, 2019):

- Aproximação 1: Análise da descrição do conteúdo, exclusivamente;
- Aproximação 2: Construção de perfis de utilizador e de item, a partir do conteúdo classificado pelo utilizador ativo.

Estas têm prós e contras a caracterizar.

4.5.1 Análise da descrição do conteúdo

Esta aproximação, segue os princípios da filtragem colaborativa em memória com base nos itens, proposta em (Sarwar et al., 2001). O sistema recomenda itens semelhantes a um item já apreciado anteriormente. Contudo, nesta apreciação depende das características do item e não de classificações de outros utilizadores.
Assim, na fase de construção do modelo, inicialmente, o sistema procura a similaridade entre todos os itens, par-a-par e, posteriormente, utiliza os itens com maior similaridade, com item já apreciado pelo utilizador, para gerar a lista de recomendações, na fase de recomendação. Por exemplo, se alguém apreciar um filme em particular, o sistema poderá recomendar-lhe outros semelhantes.
Para encontrar a similaridade entre itens, esta é derivada da descrição textual de um item baseado no conceito de *TF-IDF* já apresentado.
O método normal passa por definir "sacos de termos" ("*bag of words"*) caracterizadores dos itens (vetores que contêm as ocorrências de termos). No caso dos filmes, podem ser géneros, títulos, sinopses, etc.
Cada item é representado por um vetor *TF-IDF*, cuja pontuação é obtida pela aplicação da equação (4-9). Isto é, após a definição da pontuação *TF-IDF* de cada termo, são criados vetores de termos para cada item.
Depois de calcular as pontuações do *TF-IDF*, para determinar quais os itens mais próximos, pode ser usado o modelo de espaço vetorial**.**
Para calcular a similaridade entre itens, poderá ser aplicado qualquer um dos métodos já definidos, sobre os vetores *TF-IDF*: similaridade do cosseno; correlação de *Pearson* ou distância euclidiana.

4.5.2 Construção de perfis de utilizador e de item

Este método compreende a construção de perfil de utilizador e perfil de item, a partir dos conteúdos classificados pelo utilizador. Esta "alavanca" a descrição dos atributos a partir de itens com que o utilizador já interagiu, para recomendar itens similares. Depende exclusivamente nas classificações/escolhas prévias do utilizador, tornando-o um método robusto para ultrapassar o problema de *cold start.*
Suponhamos que um utilizador classificou filmes de géneros particulares. Poderão ser recomendados filmes correspondentes a esses géneros. O título, ano de lançamento, poderão também servir para identificar conteúdos de filmes similares.
Assim, através do género dos filmes classificados é possível criar um perfil de utilizador e recomendar itens de acordo com esse perfil. Se um utilizador avaliar um filme do género animação, o sistema irá recomendar outros filmes do mesmo género (Adomavicius & Tuzhilin, 2005; Ning et al., 2015). O mesmo raciocínio se pode aplicar ao título ou sinopse de um filme.

Nesta aproximação o conteúdo do item já foi classificado, baseado nas preferências do utilizador (criação do perfil do utilizador), enquanto o género de um item é uma característica intrínseca, que será utilizada para construir o perfil do item. Uma pontuação (*score*) será prevista combinando ambos os perfis e, assim, uma recomendação pode ser feita. Aqui também a técnica *TF-IDF* pode ser utilizada.
Normalmente, só a tabela de classificações (classificações dos utilizadores) e os atributos dos itens (perfil do item) são conhecidos (ver Figura 4-5).
Por exemplo, considerando filtragem de livros, a tabela de classificações que contem a relação entre utilizadores e livros (*User* preference) e o perfil dos itens (*Item Profile*), a tabela que relaciona livros com os atributos.

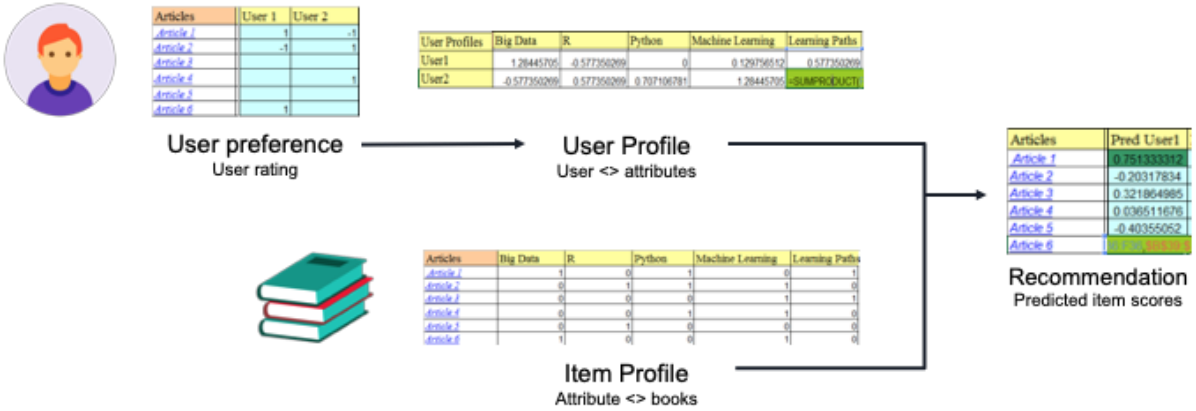

Figura 4-5: Processo CBF baseado em perfis de item e utilizador

A forma de construir um sistema CBF, pode ser traduzido nos seguintes passos (Luk, 2019).
Consideremos os elementos conhecidos: tabela de itens e atributos e tabela de classificações dos utilizadores representadas na Figura 4-6. Foi utilizada uma representação binária dos assuntos presentes nos artigos. As classificações dos utilizadores (*User 1* e *User 2*) são 1, para gosta, -1 para não gosta e não preenchido para não classificado (será preenchido a zero posteriormente). *DF* (*Data Frequency*) corresponde à frequência dos assuntos em termos da totalidade de artigos.

Articles	Big Data	R	Python	Machine Learning	Learning Paths	Total attributes	User 1	User 2
Article 1	1	0	1	0	1	3	1	-1
Article 2	0	1	1	1	0	3	-1	1
Article 3	0	0	0	1	1	2		
Article 4	0	0	1	1	0	2		1
Article 5	0	1	0	0	0	1		
Article 6	1	0	0	1	0	2	1	
DF	2	2	3	4	2			

Figura 4-6: Perfil de itens e tabela de classificações dos utilizadores

Deverão ser normalizados os vetores que constituem o perfil de itens, dividindo cada atributo pelos respetivos módulos dos vetores (*e.g., Article 1:* $atributo\ normalizado = \frac{1}{\sqrt{3}} = 0.577350269$).

Para a criação do perfil de utilizador, para cada assunto, calcula-se o produto interno entre o vetor *TF* por assunto e o vetor de classificações do utilizador. Na Figura 4-7, o exemplo apresenta a utilização da função de Excel *SUMPRODUCT* para o efeito. Neste caso, pode verificar-se que o User1 prefere mais artigos sobre *Big Data* (*TF* = 1.2844…), não gostando de artigos sobre *R* (*TF* = -0.577…).
De seguida deverá ser calculado o *IDF* (ver Figura 4-8) de cada termo, considerando o número de documentos como sendo 10 (*document corpus*).

Articles	Big Data	R	Python	Machine Learning	Learning Paths		Total attributes		User 1	User 2
Article 1	0.577350269	0	0.577350269	0	0.577350269		3		1	-1
Article 2	0	0.577350269	0.577350269	0.577350269	0		3		-1	1
Article 3	0	0	0	0.707106781	0.707106781		2			
Article 4	0	0	0.707106781	0.707106781	0		2			1
Article 5	0	1	0	0	0		1			
Article 6	0.707106781	0	0	0.707106781	0		2		1	
User Profiles										
User1	1.28445705	-0.577350269	0	0.129756512	0.577350269					
User2	-0.577350269	0.577350269	0.707106781	1.28445705	=SUMPRODUCT(F16:F21,$K$16:$K$21)					

SUMPRODUCT(array1, [array2], [array3], [array4], ...)

Figura 4-7: Perfil de itens normalizado perfil de utilizador

Por fim são calculadas as pontuações dos vetores *TF-IDF*, dos perfis de itens e do perfil de utilizador.

Articles	big data	R	python	machine learning	learning paths
Article 1	0.577350269	0	0.577350269	0	0.577350269
Article 2	0	0.577350269	0.577350269	0.577350269	0
Article 3	0	0	0	0.707106781	0.707106781
Article 4	0	0	0.707106781	0.707106781	0
Article 5	0	1	0	0	0
Article 6	0.707106781	0	0	0.707106781	0
User Profiles					
User1	1.28445705	-0.577350269	0	0.129756512	0.577350269
User2	-0.577350269	0.577350269	0.707106781	1.28445705	-0.577350269
DF	2	2	3	4	2
IDF	0.698970004	0.698970004	0.52287875	0.397940009	0.698970004

Figura 4-8: *TF* de itens (azul), *TF* de utilizadores (vermelho) e *IDF* (roxo)

Depois de calcular as pontuações *TF-IDF* para itens e utilizadores, para determinar quais os utilizadores e itens mais próximos, pode ser usado o modelo de espaço vetorial entre os perfis calculados (Figura 4-9). Aqui M1 e M2 são documentos (itens) e U1 e U2 são utilizadores (Luk, 2019).

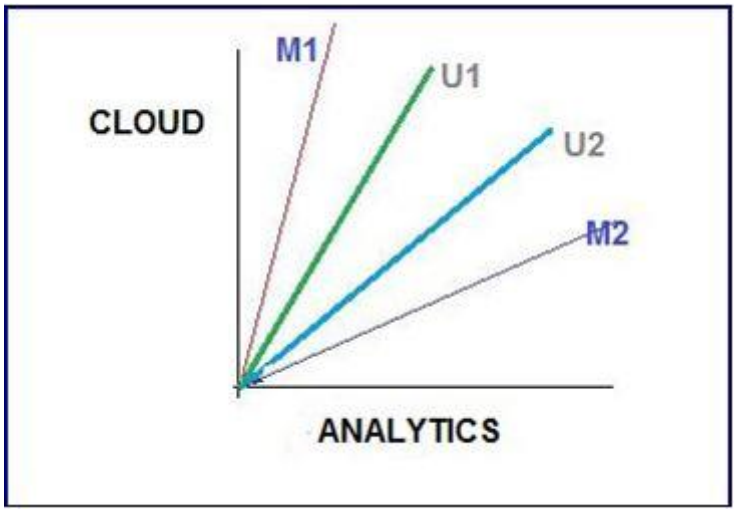

Figura 4-9: Espaço vetorial entre os termos *Cloud* e *Analytics*

No cálculo da similaridade entre utilizadores e itens, poderá ser aplicado qualquer um dos métodos, para o efeito, sobre os vetores *TF-IDF* de utilizador e itens: similaridade do cosseno; correlação de *Pearson* ou distância euclidiana.
Utilizando, por exemplo, a similaridade do cosseno, a equação adaptada é a que se apresenta em (4-10).

$$sim(U,I) = \frac{\sum_{j=1}^{t} TF-IDF_{j,U} \times TF-IDF_{j,I}}{\sqrt{\sum_{j=1}^{t}\left(TF-IDF_{j,U}\right)^2} \times \sqrt{\sum_{j=1}^{t}\left(TF-IDF_{j,I}\right)^2}} \qquad (4\text{-}10)$$

As similaridades resultantes deverão variar entre 0 e 1. Se *sim(U, I)* = 0, os dois perfis são independentes, senão os perfis têm similaridade.
Para obter uma predição ou previsão de classificação para um item ainda não classificado (I_a), utiliza-se a equação (4-11).

$$Pred(U_i, I_a) = \frac{\sum sim(U_i, I_b) \times r_{U_i,I_b}}{\sum sim(U_i, I_b)} \qquad (4\text{-}11)$$

Aqui, $sim(U_i, I_b)$ são as similaridades do utilizador com os itens já classificados e r_{U_i,I_b} as classificações atribuídas a esses itens.

4.5.3 Prós e Contras das aproximações CBF

A aproximação 1 (Análise da descrição do conteúdo, exclusivamente), contrariamente à CF, se a descrição textual for suficiente, permite ultrapassar o problema de "novo item no sistema". Por outro lado, a representação de conteúdo, pode ser variada, permitindo utilizar

outras opções de análise: outras técnicas de processamento de texto, utilização de informação semântica, inferência, etc.

A aproximação 2 (Construção de perfis de utilizador e de item, a partir do conteúdo classificado pelo utilizador ativo) apresenta algumas vantagens relativas à filtragem colaborativa e à aproximação 1. Em relação à filtragem colaborativa, esta permite uma independência entre utilizadores, uma vez que não são necessárias as classificações dos outros utilizadores, para estabelecer a similaridade entre eles e propor recomendações. Em relação à aproximação 1, permite personalizar a recomendação e sugerir uma previsão de classificação, através da análise dos perfis de itens e utilizadores (classificações já atribuídas pelo utilizador ativo). Esta também se revela transparente, relativamente à filtragem colaborativa, uma vez que as recomendações e predições, são dadas exclusivamente baseado nas características dos itens, não considerando a similaridade de gosto com outros utilizadores, que pode ser relativa.

Estas aproximações colmatam o problema de *cold start*, uma vez que novos itens são sugeridos, antes de terem sido classificados por um número substancial de utilizadores.

Em termos de contras em relação à filtragem colaborativa, estas aproximações, em geral e a aproximação 2, em particular, apresentam alguns. Por um lado, se a descrição textual do conteúdo não for suficiente para caracterizar o item com precisão, a qualidade da recomendação pode não ser boa. Por outo lado, estas aproximações tendem a ser muito parciais, uma vez que se centram exclusivamente, na similaridade entre itens, baseado no seu conteúdo (tendência de criação de "bolhas de filtragem"), apresentando recomendações similares, exclusivamente, a itens já consumidos. Por fim, no caso de novos utilizadores, pode não existir informação suficiente, para construir um perfil de utilizador sólido, comprometendo assim a qualidade da recomendação.

4.6 Filtragem Colaborativa

Filtragem Colaborativa é uma técnica comum de previsão/predição, independente do domínio, usual para a construção de sistemas de recomendação, cujo conteúdo (itens) não pode ser facilmente e adequadamente descrito por metadados. Entre outros, é o caso de classificação de filmes, músicas, produtos adquiridos e conteúdos visualizados. A CF permite fazer sugestões a um utilizador baseado nos gostos de utilizadores semelhantes.

Esta técnica funciona com base na criação de uma base de dados de preferências de itens por parte dos utilizadores (matriz de classificações ou *dataset* utilizador-item). Esta, permite combinar utilizadores com interesses e preferências relevantes, pelo cálculo da semelhança entre os respetivos perfis (linhas do *dataset*) e efetuar recomendações.

Estes utilizadores semelhantes constituem um grupo designado por vizinhança ou *neighborhood*. Um utilizador obtém uma recomendação para os itens que ainda não classificou, mas que já foram classificados pelos utilizadores da sua vizinhança. Isto é, para cada dois utilizadores, ou dois itens é calcula a similaridade entre eles. Esta é a medida que mostra a correlação, similaridade ou distância entre utilizadores ou itens. Após os *k*

utilizadores com maior similaridade terem sido descobertos, através dos itens que classificaram, é constituída a vizinhança entre utilizadores ou itens.
A CF pode ser dividida em duas categorias: baseada em memória e baseada em modelo. A Filtragem colaborativa baseada em memória pode ser com base nos utilizadores (*user-based*) ou com base nos itens (*item-based*).
Uma das técnicas mais utilizadas nos sistemas de recomendação baseados em memória é o *kNN* (*k-Nearest Neighbor*), isto é, *k* – vizinhos mais próximos. Este algoritmo permite realizar previsões de valores baseando em valores já classificados pelos *k* elementos mais similares dando um peso a sua similaridade. O sucesso deste método depende, entre outros fatores, da escolha dos pesos que cada vizinho contribuirá para a predição das avaliações desconhecidas. Esse peso é dado a partir de um cálculo de quão similar é o vizinho de um determinado elemento (F. Chen et al., 2015; Ning et al., 2015). Os modelos de vizinhança baseiam a sua previsão nos relacionamentos de similaridade entre utilizadores ou itens. Os algoritmos centrados na semelhança *user-based* preveem a classificação de um utilizador com base nas classificações expressas por utilizadores semelhantes a ele sobre esse item. Por outro lado, algoritmos centralizados na similaridade *item-based* calculam a preferência do utilizador por um item, com base nas classificações em itens semelhantes após o isolamento dos utilizadores que os classificaram (Cremonesi et al., 2010).
Para efetuar recomendações poder-se-á usar algoritmos baseados em memória, que armazenam os dados em memória e a partir destes calculam similaridades entre utilizadores ou itens para um utilizador ativo. Cada vez que se pretende fazer uma recomendação, usa-se a totalidade ou uma amostra (contruída a partir da vizinhança) do *dataset* de utilizadores e itens para gerar uma previsão (Zuva et al., 2012). O processo de Filtragem colaborativa é esquematizado na Figura 4-10.

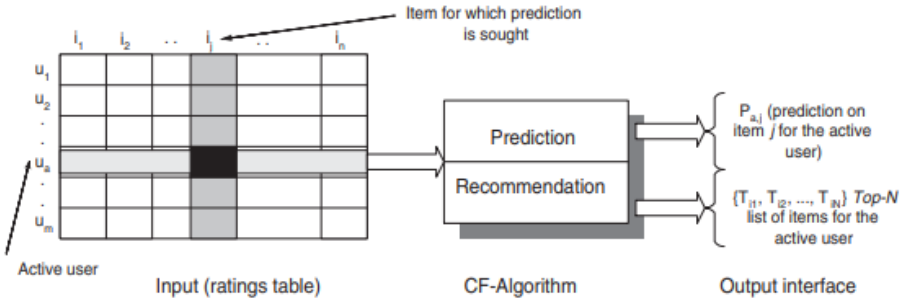

Figura 4-10: Processo de Filtragem Colaborativa

Da mesma forma, poder-se-ão usar algoritmos baseados em modelos, que utilizam a informação correspondente às classificações (*ratings*) do conjunto de dados de treino para criar um modelo estimado. Usam o conjunto de avaliações para aprender um modelo, que é usado então para fazer as predições e recomendações (Sarwar et al., 2001). Modelos são entidades que sintetizam o comportamento dos dados. Sistemas baseado em modelos foram

criados para resolver os problemas dos algoritmos baseado em memória (Karypis et al., 2015; Lee & Lee, 2019; Linden et al., 2003). Estas metodologias serão abordadas a seguir.
As metodologias de filtragem colaborativa, têm a desvantagem de sofrerem de problemas de *cold start* de itens que ocorrem quando as recomendações são feitas com base em poucas avaliações registadas. Esses problemas surgem porque a análise de similaridade não é precisa o suficiente. Nessas situações, o uso de uma abordagem baseada em conteúdo aparece como uma alternativa em complemento as anteriores descritas, criando um SR Híbrido. (De Campos et al., 2010). Esta abordagem será explorada na secção 4.7.

4.6.1 Filtragem Colaborativa em memória com base nos utilizadores

Os algoritmos baseados em memória preveem a classificação de um u_a para um item, calculando uma média de classificações dos utilizadores similares, que compartilham os mesmos interesses do u_a (Jin & Si, 2004).
O processo divide-se em duas fases: cálculo da semelhança entre utilizadores e previsão das classificações para itens ainda não classificados.
Em primeiro lugar começa-se por calcular a semelhança entre os utilizadores por forma a definir as vizinhanças entre estes. Os itens já classificados permitem selecionar um conjunto de *k* "vizinhos" do utilizador ativo u_a, com valores de semelhança mais elevados em relação a este. As classificações dos vizinhos, sobre item ainda não classificado pelo utilizador ativo, combinadas de forma específica, constituem a previsão de classificação desse item para o utilizador ativo. Assim, os itens, que são preferidos pelos vizinhos, são recomendados a u_a (Aggarwal, 2016). Caso seja pretendida a recomendação *Top N*, poder-se-á calcular a previsão para os itens ainda não classificados, apresentando os itens com as *N* previsões de classificação mais altas.
Para gerar predições ou recomendações para um utilizador *u*, primeiro é necessário utilizar a similaridade entre utilizadores para calcular a vizinhança $V \subseteq U$ ou vizinhos de *u*.
Após a determinação da similaridade entre utilizadores (ver secção 4.3) e o subconjunto de utilizadores vizinhos, é necessário agregar as informações de classificação para gerar o valor de previsão. A previsão é feita usando-se uma média ponderada das classificações dos utilizadores vizinhos no item *i* (Eksnd et al., 2010; Isinkaye et al., 2015).
A equação (4-12) permite calcular a classificação prevista, em que i corresponde ao item para o qual se pretende calcular a previsão para o utilizador u (que ainda não classificou este item), sendo $r_{v,i}$ a classificação atribuída pelo utilizador v ao item i e V representa a vizinhança do utilizador u.

$$P_{u,i} = \overline{r_u} + \frac{\sum_{v \in V} sim(r_u, r_v) \times (r_{v,i} - \overline{r_v})}{\sum_{v \in V} |sim(r_u, r_v)|} \tag{4-12}$$

Esta média é ponderada pela similaridade entre utilizadores. A subtração à classificação da média do utilizador que a efetuou ($\overline{r_v}$) permite compensar as diferenças entre utilizadores na utilização da escala de classificação (alguns utilizadores tendem a efetuar classificações mais altas que outros).
Resta saber o tamanho da vizinhança a considerar. Em alguns sistemas, tal como o original *GroupLens* (Resnick et al., 1994), todos os utilizadores são considerados, isto é, *V =U\{u}*. Com base em análise *offline* de dados de classificações de filmes, foi considerado valores de *k* entre 20 e 50 que são considerados bons em muito domínios. Noutros sistemas o tamanho da vizinhança poderá ser selecionado para cada item de forma variável (Eksnd et al., 2010).

4.6.2 Filtragem Colaborativa em memória com base nos itens

A Filtragem Colaborativa baseada em itens, considera o conjunto de itens que o utilizador ativo u_a avaliou previamente e determina quão similares eles são com relação a um item-alvo *i*. Segundo (Sarwar et al., 2001) o objetivo principal destes algoritmos é analisar a matriz dos utilizadores *vs.* itens usando essas relações para prever a avaliação de um utilizador sobre um dado item ainda não consumido. Assim, um utilizador estaria interessado em ver itens similares aos que viu antes e em evitar aqueles que não consumiu no passado.
Nesta técnica há a necessidade da identificação de utilizadores semelhantes cada vez que é solicitada uma recomendação, como resultado tendem a produzir recomendações mais rápidas (Linden et al., 2003; Sarwar et al., 2001).
O algoritmo, primeiro avalia os *K* itens mais similares para cada item de acordo com as suas similaridades, identificando o conjunto *I*= $\{i_1, i_2, .., i_n\}$ de itens candidatos a recomendação. Posteriormente, faz a união entre os *K* itens mais similares, removendo cada item no conjunto itens I_{u_i}, que o utilizador já avaliou; então calcula as similaridades entre cada item do conjunto *I* e do conjunto I_{u_i}. O conjunto resultante de itens em *I*, é ordenado em ordem decrescente de similaridade, sendo recomendado como uma lista *Top-N* (Cremonesi et al., 2010).
Para no cálculo de similaridade entre dois itens i_1, i_2 é, necessário, identificar utilizadores que tenham classificado esses itens e calcular a similaridade entre eles. Após encontrar o *cluster* de itens similares, o próximo passo é analisar as avaliações do utilizador e escolher uma técnica para gerar previsões de itens de seu interesse (Zuva et al., 2012). Na Figura 4-11 as linhas da matriz representam utilizadores e as colunas itens (Sarwar et al., 2001).

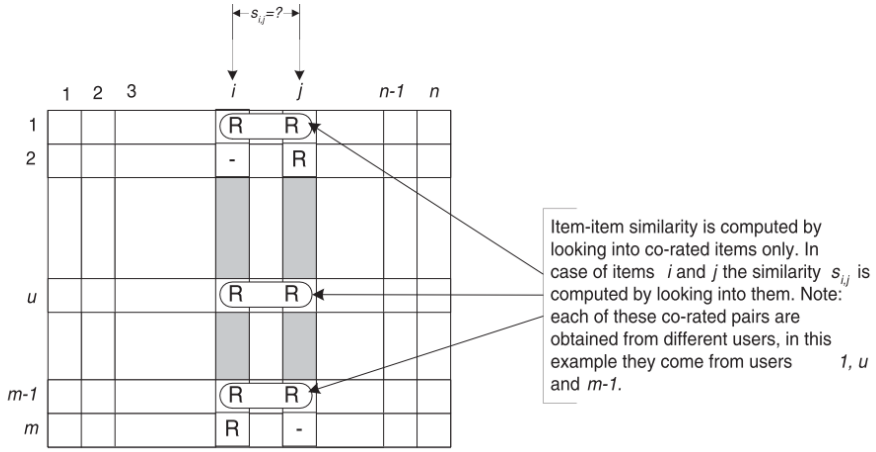

Figura 4-11: Similaridade para FC baseados em itens

A técnica utilizada foi a soma ponderada, que faz a previsão da classificação a ser atribuída a um item *i* não classificado pelo utilizador *u*, pela soma das classificações dadas pelo utilizador a um item similar a *i* (ver Figura 4-12).

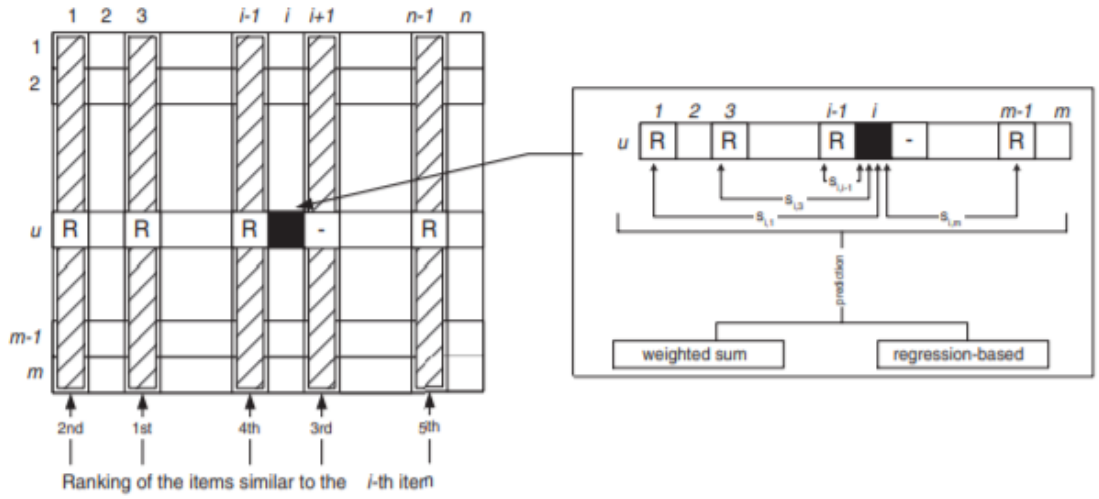

Figura 4-12: Geração da previsão para FC baseados em itens

Os métodos utilizados para calcular a similaridade entre itens e, assim possibilitar descobrir a vizinhança entre utilizadores, são: o Coeficiente de Correlação de *Pearson* (Lee Rodgers & Alan Nice Wander, 1988), a Similaridade Cosseno (Berry et al., 1999) ou a Distância Euclidiana.

No âmbito da Filtragem Colaborativa em memória com base nos utilizadores, estes métodos já foram apresentados na secção 4.3. Contudo, estes têm que ser adaptados para o cálculo da similaridade entre itens. A diferença fundamental entre o cálculo de similaridade da CF em memória (*user-based*) e da CF em memória (*item-based*) é que na primeira, a similaridade é calculada ao longo das linhas da matriz e na segunda a similaridade é calculado ao longo das

colunas. Ou seja, cada par no conjunto co-classificado corresponde a um utilizador diferente (ver Figura 4-11).

O cálculo da similaridade usando a medida de cosseno básica (ver equação (4-3)), no caso *item-based*, tem a desvantagem de que as diferenças na escala de avaliação entre os diferentes utilizadores não são levadas em consideração. Usando a similaridade de cosseno ajustada, essa desvantagem é superada, subtraindo a média do utilizador correspondente de cada par co-classificado.

Formalmente, a semelhança entre os itens i e j é dada pela equação (4-13) em que U é o conjunto formado por todos os utilizadores que classificaram ambos os itens i e j, $r_{u,i}$ e $r_{u,j}$ a classificação atribuída pelo utilizador u aos itens i e j, e $\overline{r_i}$ e $\overline{r_j}$ os ratings médios dos itens i e j (Sarwar et al., 2001).

$$sim(i,j) = \frac{\sum_{u \in U}(r_{u,i} - \overline{r_i})(r_{u,j} - \overline{r_j})}{\sqrt{\sum_{u \in U}\left(r_{u,i} - \overline{r_i}\right)^2}\sqrt{\sum_{u \in U}\left(r_{u,j} - \overline{r_j}\right)^2}} \tag{4-13}$$

Para calcular a predição entre itens é usada a fórmula (4-14), i corresponde ao item para o qual se pretende calcular a predição para o utilizador u, $r_{u,j}$ é a classificação atribuída pelo utilizador u ao item j e K representa a vizinhança de itens similares a i classificados pelo utilizador u.

$$P_{u,i} = \overline{r_u} + \frac{\sum_{j \in K} sim(i,j) \times (r_{u,j} - \overline{r_u})}{\sum_{j \in K} sim(i,j)} \tag{4-14}$$

4.7 Filtragem Híbrida

Os métodos de recomendação descritos exploram diferentes tipos de informação, tendo vantagens e desvantagens. A filtragem colaborativa foca-se nas classificações de itens, enquanto a filtragem baseada em conteúdo destaca as caraterísticas dos itens e os perfis dos utilizadores. Estas metodologias por si só, podem não fornecem resultados de recomendação que reflitam os interesses dos utilizadores.

Para o desenvolvimento dum SR que produza melhores resultados poder-se-á combinar os **métodos colaborativos** e **métodos baseados em conteúdo**. Estudos realizados mostram que a combinação linear de vários métodos pode levar a uma melhor solução na recomendação (Burke, 2002).

Recomendar itens populares é trivial segundo (Cremonesi et al., 2010) e não traz muitos benefícios para utilizadores e fornecedores de conteúdo. Por outro lado, recomendar itens menos conhecidos agrega novidade e serendipidade aos utilizadores, mas geralmente é uma tarefa mais difícil. Neste estudo, é pretendido avaliar a precisão dos algoritmos de recomendação na sugestão de itens não triviais, tendo havido um estudo prévio na recomendação de itens triviais, que serão documentados.

O método a utilizar é o *Dynamic Weighted* (Do et al., 2020), que combina de forma linear os resultados de várias técnicas. Esta recomendação híbrida ponderada baseia-se na combinação linear dos valores da recomendação baseada em conteúdo com a recomendação colaborativa. A recomendação ponderada, usa as técnicas de recomendação disponíveis no sistema e combina-as para gerar uma lista de recomendação ou uma previsão para o utilizador (Eksnd et al., 2010).

Ao utilizar esta metodologia, resolve-se o problema do *cold start* da filtragem colaborativa, combinando os dados colaborativos e de conteúdo dos itens de forma que utilizadores que pouco avaliaram, ou são novos no sistema possam ter recomendações de interesse e ainda que as recomendações sejam mais precisas e inesperadas.

Em sistemas híbridos ponderados, as saídas de vários sistemas de recomendação são combinadas usando um conjunto de pesos, conforme a equação (4-15). O ideal é ponderar os vários sistemas de forma diferenciada, de modo a dar maior importância aos sistemas mais precisos (Aggarwal, 2016; Do et al., 2020).

$$P_{u,i} = \alpha_1 \times P_{u,i}^{(1)} + \cdots + \alpha_n \times P_{u,i}^{(n)} \tag{4-15}$$

Neste método o parâmetro α da combinação linear é ajustado para variar a importância de cada técnica no resultado da recomendação híbrida. Cada uma dessas parcelas é ponderada de acordo com um parâmetro denominado confiança, que varia entre 0 e 1.

Dado que a maioria dos SR segue a filtragem colaborativa, a filtragem de conteúdo ou ambas (em termos de filtragem híbrida), serão consideradas exclusivamente estas técnicas. Assim, a confiança (α_u) da previsão da filtragem colaborativa ($P_{u,i}^{(CF)}$) deve aumentar quando o número de itens avaliados pelo utilizador *u* aumenta. A confiança do sistema baseado em conteúdo ($P_{u,i}^{(CBF)}$), pode ser assim definida como $1- \alpha_u$. Isto não significa que a recomendação do sistema CBF diminui quando um utilizador classifica mais itens, mas sim que a confiança da recomendação do sistema CF, passa a ser dominante (Do et al., 2020). Uma vez que estamos interessados em combinar duas técnicas de recomendação, o cálculo da pontuação de predição pode ser definido pela equação (4-16):

$$P_{u,i} = \alpha_u \times P_{u,i}^{(CF)} + (1 - \alpha_{u)} \times P_{u,i}^{(CBF)} \tag{4-16}$$

Em (Do et al., 2020), o α_u é dado pela fórmula (4-17), em que $k \in \mathbb{N}$, é o número de vizinhos usados na classificação colaborativa e t_u é o número de itens de $I \backslash I_u$. Caso o número de itens não classificados pelo utilizador *u* for maior que a vizinhança, então $t_u = k$. A razão entre t_u e k é o fator de confiança para cada utilizador *u*.

$$\alpha_u = \frac{t_u}{k} \times 0{,}9 \tag{4-17}$$

Para a construção do SR, as metodologias descritas são implementadas individualmente e então combinadas, permitindo construir um SR Híbrido ponderado.

4.8 Métricas de avaliação

Avaliar a qualidade de algoritmos de recomendação consiste essencialmente em avaliar o grau de aceitação das recomendações, quantificando o número de vezes que os utilizadores aceitam ou rejeitam itens recomendados. Essa avaliação pode ser efetuada usando diferentes tipos de medição, que podem ser de precisão ou de cobertura (âmbito). A precisão é a fração das recomendações corretas do total de recomendações possíveis, enquanto a cobertura mede a fração de objetos no espaço de pesquisa para o qual o sistema é capaz de fornecer recomendações (Isinkaye et al., 2015). As métricas para medir a precisão dos SR são divididas em **métricas estatísticas** e de **precisão de suporte à decisão** (Sarwar et al., 2001).

A qualidade de um SR pode ser avaliada comparando as recomendações a um conjunto de teste de classificações do utilizador conhecidas. Esta avaliação é efetuada recorrendo a um conjunto de métricas, *predictive accuracy metrics*. Estas métricas permitem comparar as classificações previstas com as classificações reais dos utilizadores (Herlocker et al., 2004).

As **métricas de precisão estatística** que avaliam a precisão de uma técnica de filtragem comparando as classificações previstas diretamente com a classificação real do utilizador. O erro médio absoluto ou *Mean Absolute Error* (MAE) e o erro quadrático médio da raiz ou *Root Mean Square Error* (RMSE) são usados como métricas de precisão estatística e medem a precisão da previsão das classificações (Shani & Gunawardana, 2011).

A abordagem da similaridade entre itens (*item-based*) é geralmente a abordagem que geralmente apresenta melhor desempenho em termos de RMSE, sendo mais escalável. Esses melhores resultados devem-se ao fato de que o número de itens geralmente ser menor que o número de utilizadores. Outra vantagem dos algoritmos baseados em itens é que o raciocínio por trás de uma recomendação para um utilizador específico pode ser explicado em termos dos itens anteriormente classificados pelo utilizador. (Cremonesi et al., 2010).

O sistema de recomendação gera uma predição de classificações dado por $\hat{r}_{ui}$ para um par utilizador-item (u, i), que pertence ao conjunto de teste τ, em que $|\tau|$ é o número de classificações. Como se conhece a classificação, r_{ui} , atribuído pelo utilizador u ao item i, tem-se que o cálculo do RMSE é dado pela equação (4-18).

$$RMSE = \sqrt{\frac{1}{|\tau|}\sum_{(u,i)\in\tau}(\hat{r}_{ui} - r_{ui})^2} \quad (4\text{-}18)$$

Quanto menor for o valor do RMSE maior será a qualidade das classificações previstas pelo algoritmo.

Em alternativa ao RMSE poder-se-á usar o MAE, que mede a diferença média entre classificações previstas e as classificações reais, com a equação (4-19):

$$MAE = \frac{1}{|\tau|}\sum_{(u,i)\in\tau}|\hat{r}_{ui} - r_{ui}| \quad (4\text{-}19)$$

À semelhança do RMSE, quanto menor a MAE melhor é a qualidade do algoritmo na predição de classificações (Shani & Gunawardana, 2011).

As métricas preditivas consideram todos os itens da mesma forma, ao atribuir-lhes a mesma relevância. No entanto existe benefício em avaliar os itens que são do interesse do utilizador, discriminando as boas das más recomendações, tendo em conta que o processo de recomendação consiste em entregar boas recomendações. Só assim é que se consegue fidelizar o utilizador. Os mecanismos baseados em precisão não consideram fatores como a proliferação de diversos interesses do utilizador e o desejo de mudanças (Ge et al., 2010).

Alguns SR não preveem as preferências de itens pelo utilizador, mas tentam recomendar ao utilizador itens que eles podem apreciar. Em (Shani & Gunawardana, 2011), dão o exemplo da *Netflix*, em que para além de sugerir um conjunto de filmes relacionados com as classificações a outros filmes ao utilizador, sugere outros filmes que podem ser do interesse do utilizador. Nesta situação não há o interesse em saber se o sistema prevê corretamente as classificações desses filmes, mas sim se o sistema prevê corretamente que o utilizador adicionará esses filmes à sua lista de interesses, ou seja, se o filme/item é consumido pelo utilizador.

Considerando um universo de itens a recomendar, podemos considerar as possibilidades apresentadas no Quadro 4-2 (Shani & Gunawardana, 2011).

	Recomendados	**Não Recomendados**
Consumido	Verdadeiro-Positivo (VP)	Falso-Negativo (FN)
Não Consumido	Falso-Positivo (FP)	Verdadeiro-Negativo (VN)

Quadro 4-2: Resultados possíveis de uma recomendação

Poder-se-á contar o número de ocorrências que se enquadram em cada uma das categorias e calcular as métricas consideradas no Quadro 4-3 (Reis, 2012).

Métrica	Expressão	Interpretação
Precision	$\frac{Nº\ de\ VP}{Nºde\ VP + Nºde\ FP}$	Mede a probabilidade de um item recomendado ser relevante.
Recall	$\frac{Nº\ de\ VP}{Nºde\ VP + Nºde\ FN}$	Mede a probabilidade de um item relevante ser recomendado.
False Positive Rate	$\frac{Nº\ de\ FN}{Nºde\ FP + Nºde\ VN}$	Mede a probabilidade de um item ser mal recomendado
F-measure (F1)	$2 * \frac{Precision + Recall}{recision * Recal}$	Combina as características do *Precision* e do *Recall* numa métrica conjunta.

Quadro 4-3: Métricas de performance de um sistema de recomendação

5. Desenvolvimento do Sistema de Recomendação

Foi desenvolvido um protótipo de um Sistema de Recomendação Híbrido, com interface web, para recomendações de filmes. Este possibilita experimentar, testar e avaliar algoritmos de recomendação baseados em CBF, CF baseados em memória e um HF, permitindo a um utilizador cinéfilo identificado, obter uma boa recomendação de filmes ainda não vistos (não avaliados). A interface web, desenvolvida, é responsiva, permitindo ser acedida a partir de qualquer dispositivo de navegação, adaptando-se à dimensão do ecrã.

O objetivo é o desenvolvimento de um sistema que tem em atenção a atividade do utilizador ativo, que utiliza fontes de dados *offline* ou *online*, para proceder à previsão/recomendação. Assim, o sistema é implementado numa plataforma de desenvolvimento e implantação web baseada em *python*, o *Django*. Integra o acesso a fontes de dados *online* de filmes, para obtenção, criação de recomendações e apresentação ao utilizador e eventual fonte, para algoritmos de filtragem baseada em conteúdo, disponibilizados pelo *site TMDb*[7], permitindo ainda, integrar um *MovieLens Datasets offline* de *ratings* disponibilizados pelo *GroupLens*[8], como fonte para os algoritmos de filtragem colaborativa e baseada em conteúdo. Este sistema implementa uma base de dados que inclui as avaliações dos utilizadores identificados do sistema, que será combinada com as fontes *online* e *offline*, já mencionadas, para melhorar as recomendações a efetuar.

[7] *The Movie Database* (*TMDb*), é uma base de dados de filmes e TV criada pela comunidade (*https://www.themoviedb.org/*).

[8] *GroupLens Research Lab* da Universidade do Minnesota que utiliza tecnologia de filtragem colaborativa para recomendações de filmes e disponibiliza no site do grupo de investigação vários *datasets* de dimensão variável (*https://grouplens.org/datasets/movielens/*).

5.1 Análise e Conceção do sistema de recomendação

O protótipo do sistema de recomendação apresenta uma interface que permite a qualquer utilizador (designado não identificado ou não autenticado) obter estatísticas, gráficos, informações de filmes e uma recomendação não personalizada do tipo *Top-N*, dos filmes mais populares a partir do *MovieLens Datasets offline* (designado no âmbito deste sistema como "ficheiros de *datasets offline* de filmes"), pela implementação do algoritmo de filtragem colaborativa simples que aplica a equação utilizada pelo *IMDb*, para definição do "*Top Rated 250 titles*" (ver secção 4.4). Estes utilizadores poderão passar a ser utilizadores autenticados, após o processo de registo e autenticação no sistema.

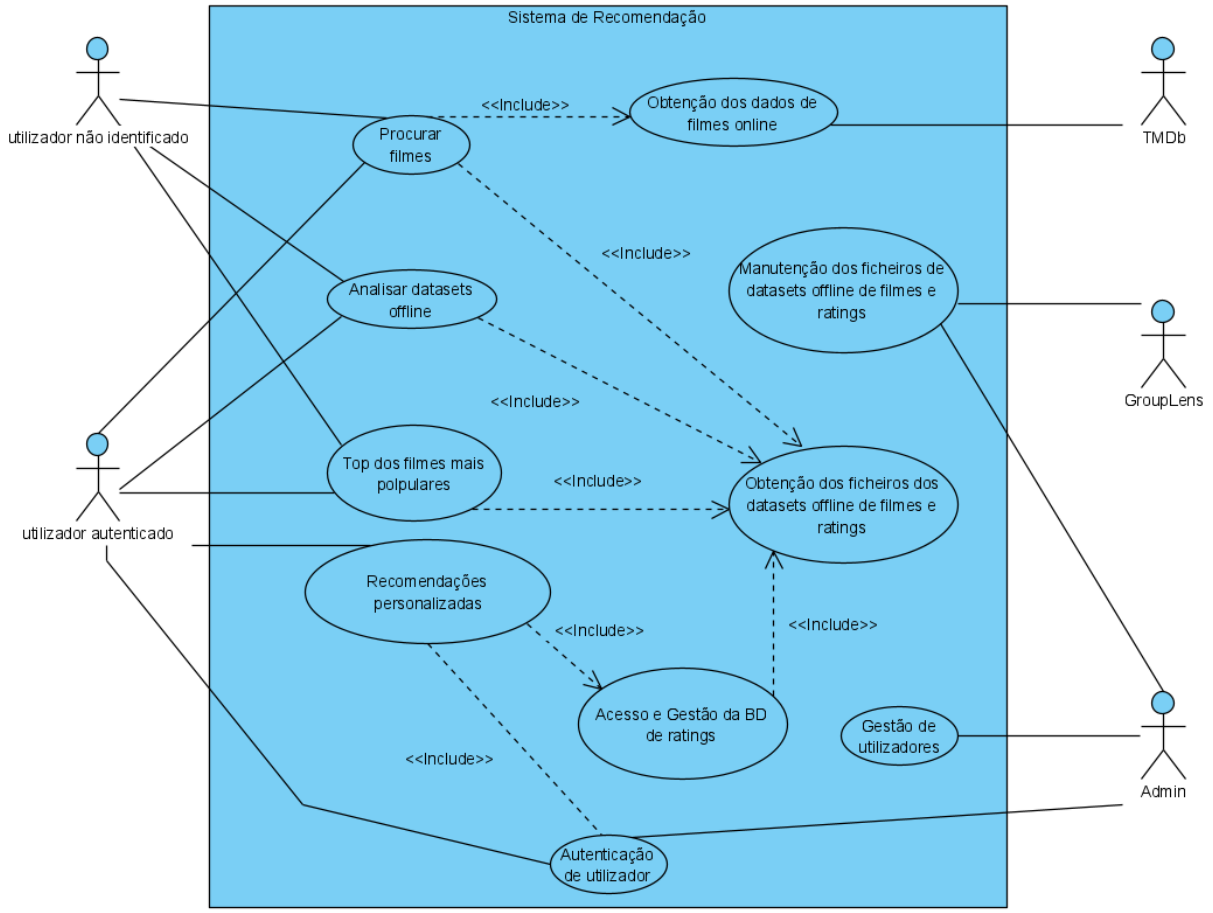

Figura 5-13: Diagrama de casos de uso do protótipo do sistema de recomendação

Os utilizadores autenticados, para além da funcionalidade já enunciada, podem obter recomendações personalizadas de filtragem colaborativa, baseada em conteúdo ou híbrida. Existe ainda um utilizador administrador do sistema, que para além da gestão da base de dados de *ratings* interna e de utilizadores do sistema, efetua ainda a manutenção manual dos *MovieLens Datasets offline*. Para uma melhor compreensão do protótipo do sistema de recomendação, os principais casos de uso podem ser verificados no diagrama que consta da Figura 5-13.

5.1.1 Sistema de avaliação de filmes

O sistema de avaliação de filmes implementado baseou-se nos *MovieLens Datasets* disponibilizados de forma gratuita pelo *GroupLens*. Caso não se tivesse optado por um conjunto de dados disponível neste momento, as experiências, testes, recomendações e avaliações só podiam ser realizada com dados simulados (Herlocker et al., 2004).
Assim, começou-se por uma investigação do funcionamento do *site* web do *MovieLens*[9]. Este só é utilizável, se se criar uma conta no sistema. Quando o utilizador faz o registo é-lhe solicitado, com carácter de obrigatoriedade, a seleção de 3 grupos de filmes de 9 disponíveis. Cada um desses grupos agrega 3 géneros de filmes, eventualmente relacionados (*e.g.*, ação, cómicos e efeitos especiais; clássico, obra prima e cotado). Só após esta caracterização inicial é possível começar a utilizar o sistema, receber recomendações e avaliar filmes. Em termos de recomendações, aparecem 5 listas de filmes com as pontuações previstas, baseado nas seleções iniciais. As listas são: *top picks*, lançamento recentes, mais classificados, favoritos do ano anterior e novas adições à base de dados. À medida que se vão classificando filmes, as recomendações vão sendo ajustadas, baseada nos géneros dos filmes.
As classificações têm uma escala numérica, entre 0.5 e 5, com incrementos de 0.5, representada por estrelas e meias estrelas. Quanto mais estrelas possuir melhor é a avaliação dos utilizadores. É este o *site* a partir do qual são criados os *MovieLens Datasets* disponibilizados pelo *GroupLens*, que constituirão os *datasets offline* a utilizar neste protótipo de sistema de recomendação. Os *datasets* são constituídos pelo conjunto de ficheiros *csv* (*comma-separated values*) a utilizar. A escala de classificações a utilizar é a adotada pelo *MovieLens*.
O GroupLens disponibiliza para a comunidade vários conjuntos de dados (GroupLens, 2020b) para utilização académica, desenvolvimento e investigação. Neste trabalho utilizou-se o último conjunto de dados (*dataset*) disponibilizados para utilização académica e desenvolvimento. Este conjunto de dados foi atualizado em setembro de 2018.
Segundo o ficheiro *README* contido, este conjunto de dados descreve a avaliação de 5 estrelas e a atividade livre de marcação de texto (*tagging*) do *site MovieLens*. Este contém 100 836 classificações e 3 683 *tags* em 9 742 filmes. Estes dados foram criados por 610 utilizadores, entre 29 de março de 1996 e 24 de setembro de 2018. O conjunto de dados foi gerado em 26 de setembro de 2018. Os utilizadores incluídos foram aleatoriamente selecionados. Todos os utilizadores selecionados avaliaram pelo menos 20 filmes. Não é incluída nenhuma informação demográfica, sendo cada utilizador representado exclusivamente por um *userId*. Os dados estão contidos nos seguintes ficheiros: *links.csv*, *movies.csv*, *ratings.csv* e *tags.csv*.
As avaliações (*ratings*) estão contidas no ficheiro *ratings.csv*. Cada linha deste, após a linha do cabeçalho, representa uma avaliação de um filme por um utilizador e tem o seguinte

[9] Site de investigação do *GroupLens* que utiliza tecnologia de filtragem colaborativa para fazer recomendações de filmes (*https://movielens.org/*).

formato: *userId, movieId, rating, timestamp*. As linhas estão ordenadas, primeiro por *userId* e, em seguida, para cada utilizador, por *movieId*. As avaliações são feitas numa escala de 5 estrelas. O *timestamp* representa os segundos desde a meia-noite do 1º de janeiro de 1970 UTC (Tempo Universal Coordenado ou *Coordinated Universal Time*).
As *tags* estão contidas no ficheiro *tags.csv*. Cada linha após a linha do cabeçalho representa uma *tag* aplicada a um filme por um utilizador e tem o seguinte formato: *userId, movieId, tag, timestamp*. Em termos de ordenação e do campo *timestamp*, aplica-se o já referido para o ficheiro de *ratings.csv*.
As informações do filme estão contidas no ficheiro *movies.csv*. Cada linha deste após a linha do cabeçalho representa um filme e tem o seguinte formato: *movieId, title, genres*. O título (*title*) do filme foi manualmente introduzido ou importado do sítio do *TMDb*, incluindo o ano de lançamento entre parênteses. Segundo o *README*, podem ocorrer erros e inconsistências no título. Os géneros (*genres*) são listas separadas por "|" selecionadas a partir dos seguintes: *Action, Adventure, Animation, Children's, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, Musical, Mystery, Romance, Sci-Fi, Thriller, War, Western* e *(no genres listed)*.
Os identificadores de filme que podem ser usados para aceder (*link*) a outras fontes de dados de filmes estão contidos no ficheiro *links.csv*. Cada linha deste, após a linha do cabeçalho representa um filme e tem o seguinte formato: *movieId, imdbId, tmdbId*. O *movieId* é um identificador de filmes usado no *site* web do *MovieLens* (*e.g.*, o filme *Toy Story* tem o *link https://movielens.org/movies/1*). O *imdbId* é um identificador de filmes usado no *site* web do pelo *IMBb* (*e.g.*, o mesmo filme *Toy Story* tem o *link http://www.imdb.com/title/tt0114709/*). O *tmdbId* é um identificador de filmes usado no *site* web *TMDb* (*e.g.*, ainda o filme *Toy Story* tem o *link https://www.themoviedb.org/movie/862*).

5.1.2 Modelação de dados

Para a implementação da base de dados de *ratings* a manter no âmbito deste sistema de recomendação, a abstração de dados, foi baseada nos *MovieLens Datasets*. O diagrama de classes do sistema é o constante da Figura 5-14. Este descreve as classes, relacionamentos entre elas e atributos a implementar neste protótipo. Não são representadas quaisquer operações (ou métodos) nas classes, uma vez que não existe nenhuma operação específica, para além das características *CRUD* (*Create, Read, Update, and Delete*).
Esta base de dados, vai sendo criada e mantida à medida que os utilizadores identificados vão efetuando classificações (*ratings*) de filmes. O administrador do sistema (*admin*) pode efetuar a manutenção de toda a base de dados.
Os filmes (*Movie*) contêm os atributos constantes de *movies.csv*, incluindo os identificadores *imdbId* e *tmdbId*, que constam de *links.csv* e que permitem integrar com as bases de dados já descritas.

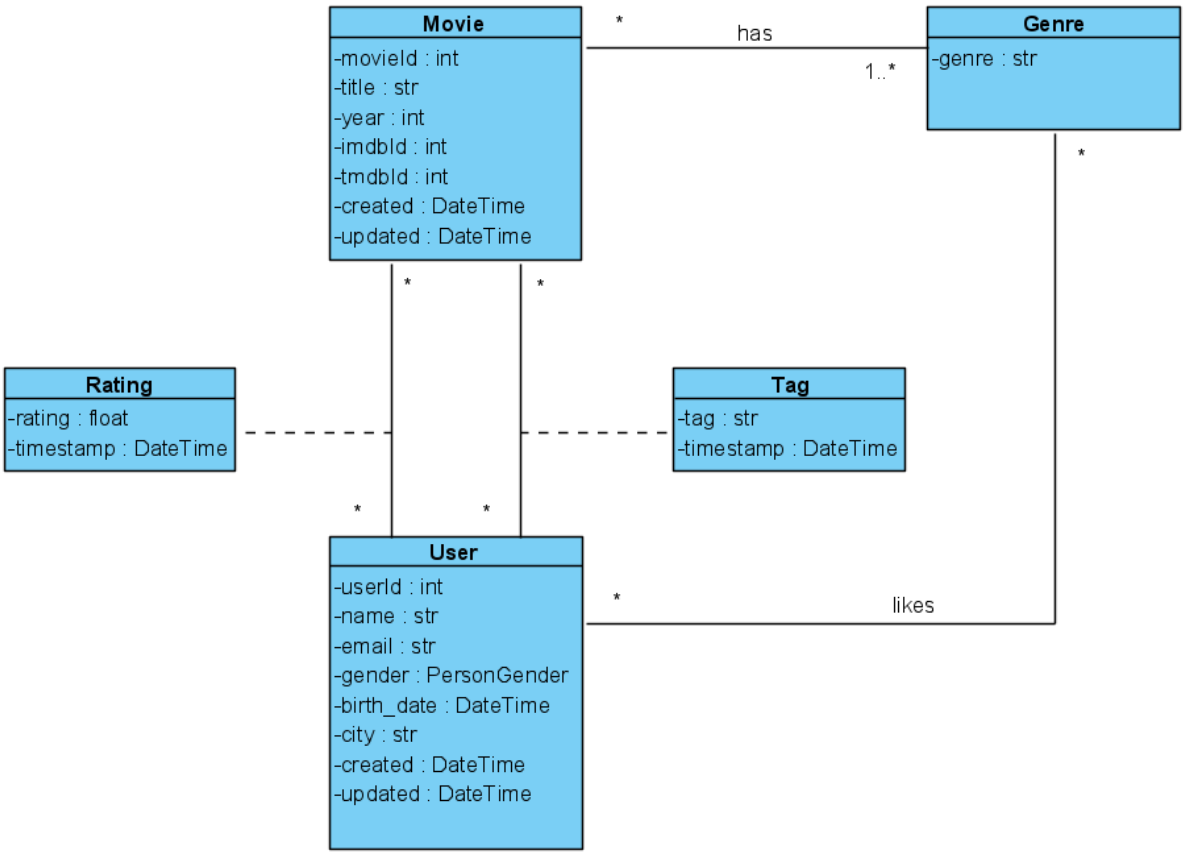

Figura 5-14: Diagrama de classes do protótipo do sistema de recomendação

É criado um tipo género de filme (*Genre*) que permite agregar os géneros de filmes, presentes no *movies.csv*. Além de, naturalmente, estarem associados aos filmes que serão mantidos na base de dados de *ratings* deste sistema de recomendação, serão ainda associados aos utilizadores, para criação dos seus perfis nos processos de recomendação.
Embora nos *MovieLens datasets*, o utilizador só está caracterizado com *userId*, aqui, para o Utilizador (*User*), serão salvaguardados outros atributos característicos, como nome, endereço de correio eletrónico e, alguns dados demográficos, para orientar futuras recomendações (*e.g.*, género, data de nascimento, localidade onde habita).

5.2 Implementação da Funcionalidade do protótipo

A interface web com o utilizador foi desenvolvida em *Django*, constituindo um *site web* responsivo. A página *web* inicial (*Home*) é a que se apresenta na Figura 5-15.

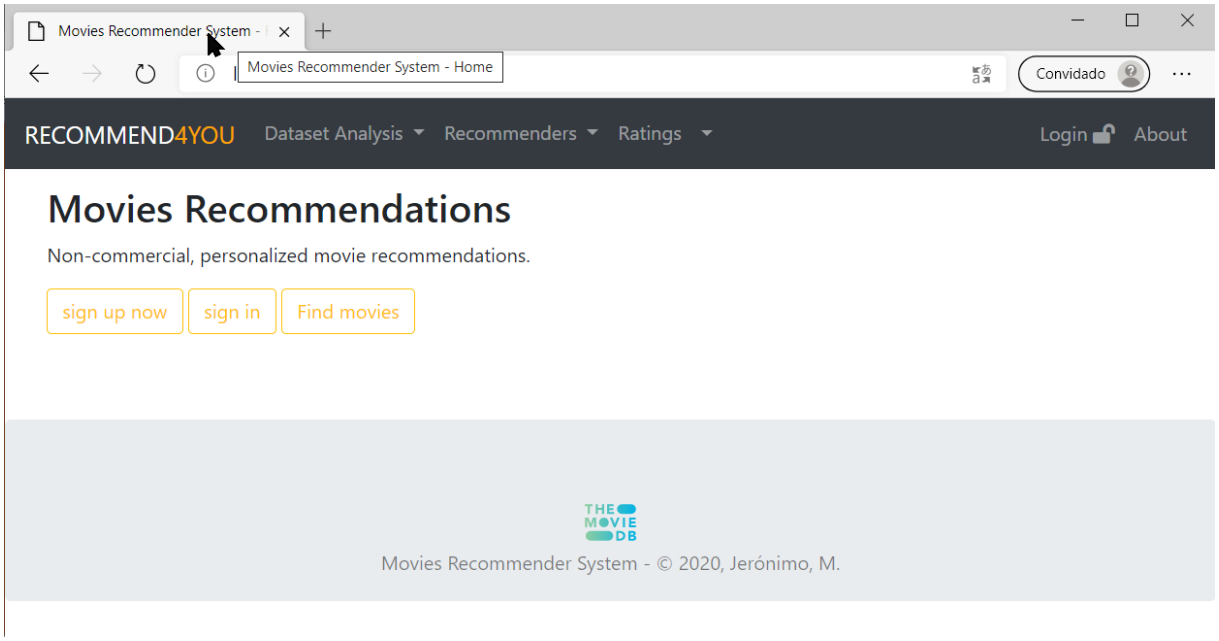

Figura 5-15: *Home* do protótipo do sistema de recomendação

Esta página permite a um utilizador não autenticado obter alguma informação sobre filmes, criar um utilizador ou autenticar-se para classificar filmes e obter recomendações. Este processo é facilitado, através de botões: o registo no sistema ("*sign up now"*), a autenticação, para poder efetuar classificações ("*sign in"* ou "*Login"* na barra de navegação) ou ainda procurar filmes por título/parte do título existentes no *MovileLens Datasets offline* ("*find movies"*). Quando o utilizador está autenticado, a página principal, apresentará uma recomendação do tipo *Top-N*, com a aplicação da técnica híbrida.

No topo da página é apresentado um menu de navegação, que operacionaliza toda a funcionalidade do *site*. Do lado esquerdo, aparecem as opções relacionadas com o sistema de recomendação de filmes. Da esquerda para a direita, é apresentado o botão de navegação *brand*, que permite aceder à página inicial (*Home*); um menu de acesso a estatísticas e gráficos dos dados preparados e limpos do *MovileLens Datasets offline*; um menu de acesso aos recomendadores implementados (*Content-Based, Collaborative, Hybrid*); bem como um menu de manutenção dos *ratings* atribuídos pelo utilizador autenticado. Do lado direito, aparecem as opções relacionadas com a manutenção dos dados do utilizador, acesso à administração do sistema de recomendação e informação sobre o sistema (*About*).

Caso, o acesso seja feito a partir de um dispositivo com um ecrã de menores dimensões, o *site* é responsivo, adaptando o menu de navegação em conformidade e permitindo o acesso às opções de navegação a partir de um botão único (conhecido por *"hamburger menu"*), conforme Figura 5-16.

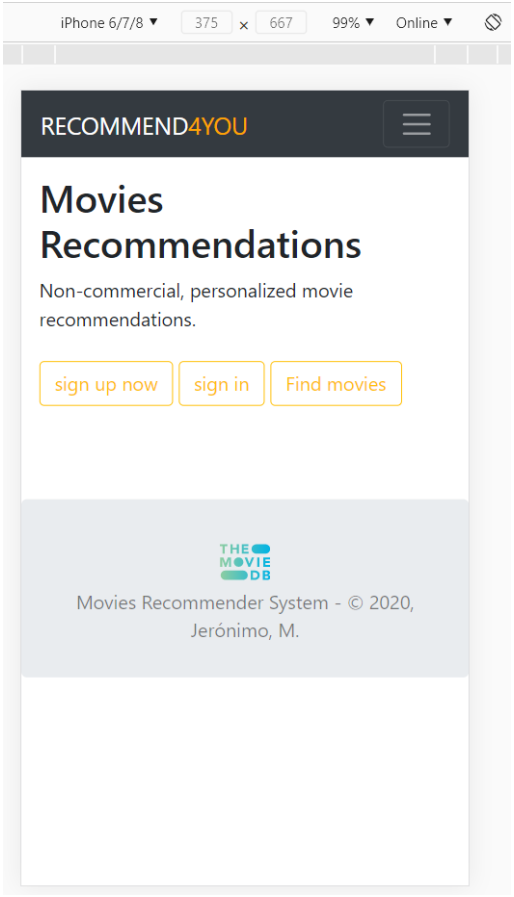

Figura 5-16: *Home* do protótipo do sistema de recomendação a partir de um *smartphone*

Embora seja pretendido a autenticação do utilizador para poder obter recomendações personalizadas, sem efetuar a autenticação, um utilizador não identificado, pode utilizar o sistema para procurar, no *MovileLens Datasets offline*, informações sobre: filmes (botão *"Find movies*"), estatísticas e gráficos dos dados preparados e limpos (submenu *"Dataset Analysis ➨"*), obter uma recomendação baseada exclusivamente na semelhança textual de filmes (submenu *"Recommenders ➨ Content-Based ➧ Description of content"*) ou obter uma recomendação colaborativa simples e não personalizada do *Top* de popularidade dos filmes (submenu *"Recommenders ➨ Collaborative ➧ Top of Popularity"*).
Conforme a Figura 5-17, um utilizador autenticado por seu turno, pode obter recomendações personalizadas, baseadas no *MovieLens Dataset offline*, melhoradas com dados constantes na BD de *ratings* (submenu *"Recommenders ➨ Content-Based ➧ User rating content*", *"Recommenders ➨ Collaborative ➧ User based"* ou *"Recommenders ➨ Hybrid"*); efetuar e manter as suas classificações (submenu *"Ratings ➨*") bem como manter os dados do seu perfil de utilizador (submenu *"User ➨*").
Em termos de *output*, a procura de filmes (botão *"Find movies*" ou submenu *"Recommenders ➨ Content-Based ➧ Description of content"*) permite obter até 3 filmes cujo título mais se aproximam do texto de procura. Estes são apresentados como *Bootstrap's cards*, cujo conteúdo é obtido a partir do site do *TMDb*, como por exemplo a imagem dos cartazes, presentes na Figura 5-18. A partir dos cartões é possível obter mais informação sobre o filme,

obter recomendações de filmes baseadas exclusivamente na semelhança textual com este em termos de: géneros, título ou título e géneros ou, ainda, atribuir um *rating* ao filme em questão (tem de ser um utilizador identificado e autenticado).

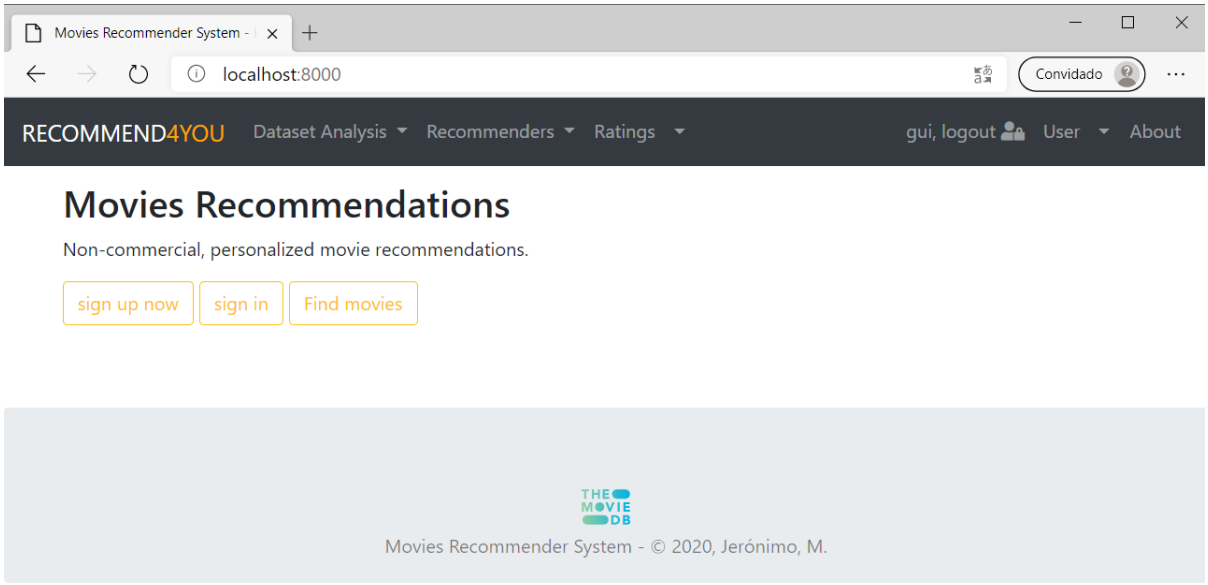

Figura 5-17: *Home page* do sistema de recomendação para um utilizador autenticado

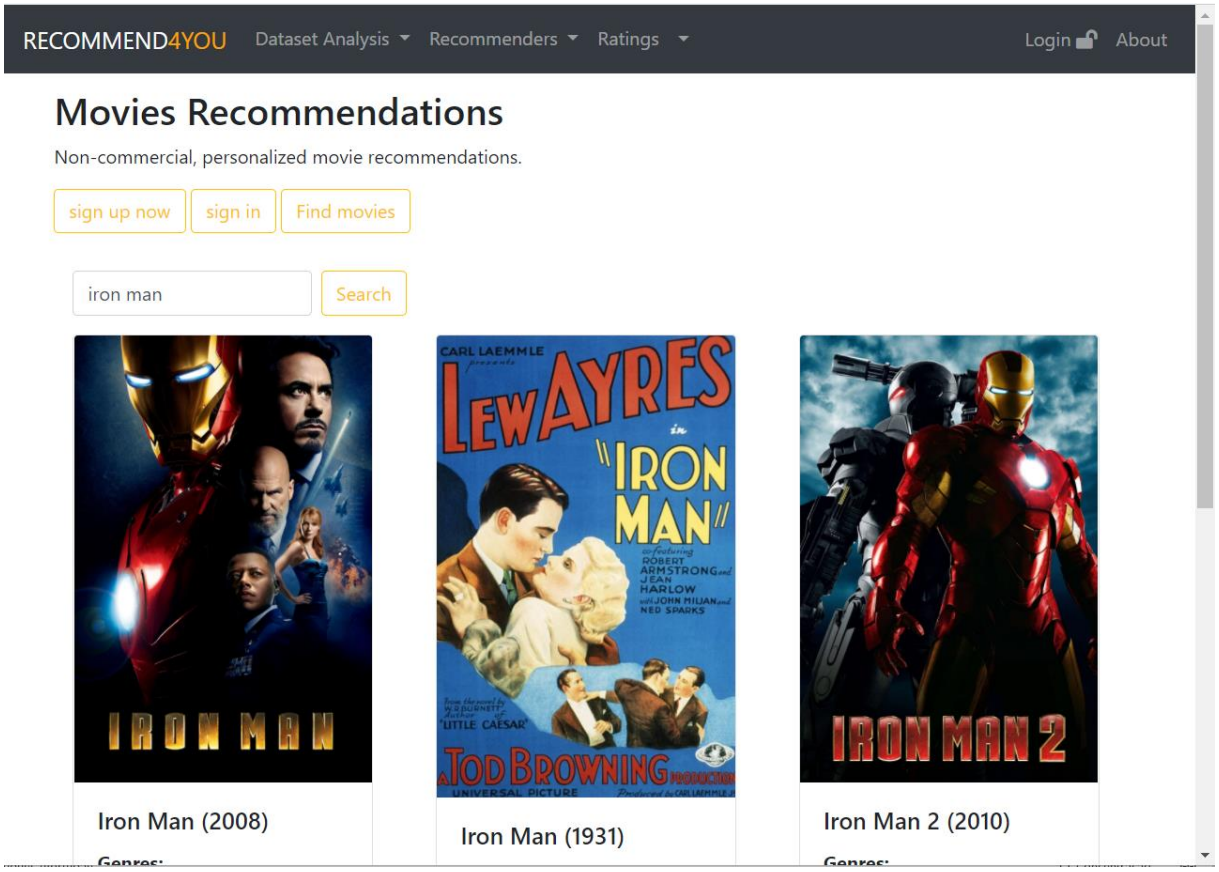

Figura 5-18: Resultado da procura de filmes por título

Para a preparação e limpeza dos dados do *MovileLens Dataset offline*, elaboração de gráficos, bem como suportar a implementação dos algoritmos de recomendação foram utilizadas as bibliotecas de *Python pandas* e *numpy*.
A biblioteca *pandas* (NumFOCUS, n.d.) é uma ferramenta de manipulação e análise de dados de código aberto rápida, poderosa, flexível e fácil de usar, construída com base na linguagem de programação *Python*. As principais facilidades são: a representação de dados de forma tabular com indexação (*Dataframe*) ou como séries indexadas (*Serie*), leitura/escrita facilitada de dados entre as estruturas de dados em memória e ficheiros em diferentes formatos: *csv*, *excel*, *bases de dados SQL e HDF5;* bem como inúmeras facilidades de manipulação dos *Dataframes* e *Series*.
A biblioteca *numpy* (*NumPy*, 2020) é um módulo fundamental para computação científica em *Python*. As principais facilidades são: implementação de *arrays* multidimensionais, ferramentas matemáticas de computação.
Todas as recomendações, são apresentadas como listas do tipo *Top-N*, sendo cada linha composta, pelo menos, por um número de ordem, o título, a previsão de classificação (se aplicável), bem como botões para obter mais informação, obter outros filmes com similaridades textuais baseadas em título e género e, ainda, classificar o filme (ver Figura 5-19).

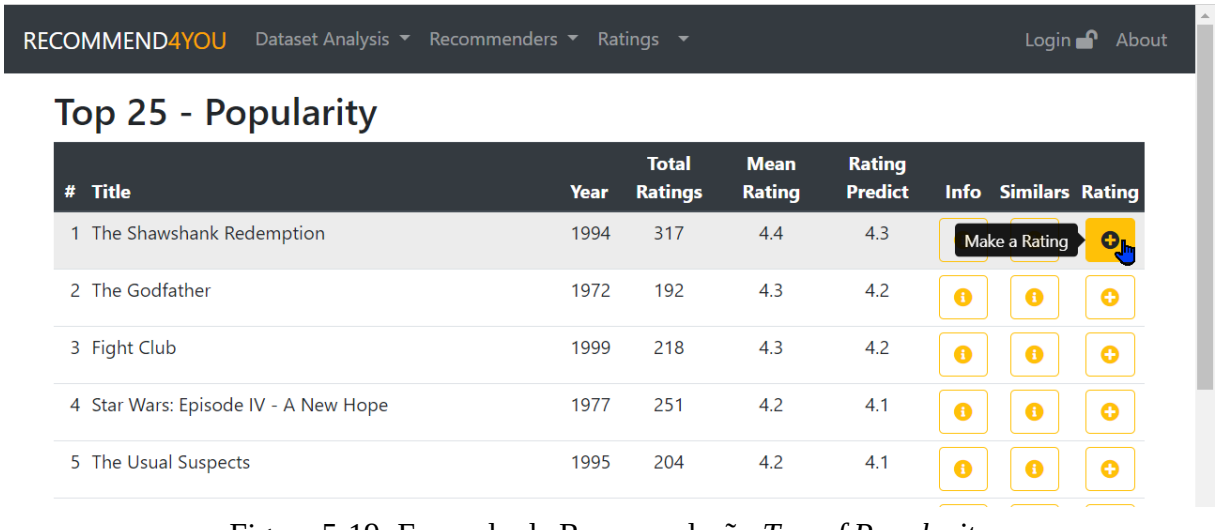

Figura 5-19: Exemplo de Recomendação *Top of Popularity*

5.3 Obter estatísticas e gráficos de dados de filmes

Para poder exibir estatísticas e gráficos dos dados de filmes, começa-se por carregar em *pandas Dataframes*, limpar e pré-processar os dados. Estas operações foram operacionalizadas em funções, para poderem ser utilizadas nas implementações dos algoritmos de recomendação. Estas funções permitem criar *Dataframes* limpos, consistentes e pré-processados de filmes (*movies_df*) e classificações (*ratings_df*), por exemplo, a partir dos respetivos ficheiros *csv* que constituem o *MovieLens Dataset offline*. A título de exemplo no

caso dos filmes, o ano é retirado do título, como uma nova coluna do respetivo *Dataframe*, enquanto nas classificações, a coluna *timestamp* é convertida para o tipo *datetime* do *pandas*.
Para apresentar estatísticas fundamentais dos dados, estes são limpos, corrigidos e são removidas inconsistências.
O conjunto de estatísticas descritivas apresentadas, correspondem a algumas medidas calculadas sobre o *dataframe* de classificações (*ratings*) que constitui o *MovieLens Dataset*. Essas estatísticas são a taxa de esparsidade[10] dos dados (razão entre o número de classificações atribuídas e o número de classificações possíveis, esta última traduzida no produto entre o número de utilizadores e número de *ratings*), o número de utilizadores, o *userId* do utilizador com maior número de *ratings* e respetiva frequência, o número de filmes, *movieId* do filme mais classificado e respetiva frequência, a média, a classificação mínima, o 1º quartil, o 2º quartil, o 3º quartil, o máximo, o mínimo e o desvio padrão das classificações (*ratings*) e *timestamp*.
São ainda, filtrados os *Top* dos filmes mais classificados, dos filmes mais recentes e dos filmes mais antigos. Para o efeito o *Dataframe* de classificações é agrupado por filme, em termos do número de classificações atribuídas e média das classificações atribuídas.
Para os filmes mais classificados, é obtido o *Top*, pela ordenação do agrupamento efetuado, por ordem decrescente do número de classificações e média das classificações atribuídas (em primeiro lugar por total de classificações e dentro do total de classificações, por média das classificações).
Para os filmes mais recentes, é obtido o *Top*, pela ordenação do agrupamento efetuado, por ordem decrescente do ano, número de classificações e média das classificações atribuídas.
Para os filmes mais antigos, é obtido o *Top*, pela ordenação do agrupamento efetuado, por ordem crescente do ano, decrescente do número de classificações e média das classificações atribuídas.
Em relação à criação de gráficos de visualização dos dados *MovieLens Dataset offline* é utilizada a biblioteca de *Python Matplotlib*, que é uma biblioteca abrangente para a criação de visualizações (gráficos) estáticas, animadas e interativas em *Python* (The Matplotlib development Team, 2021).
Um dos primeiros gráficos implementados permite visualizar o número de filmes lançados por ano. A ideia é contar o número de filmes lançados por ano, assim, como os *ratings* efetuados por ano, criando um gráfico com a evolução destas duas séries. Este gráfico foi implementado de forma interativa, permitindo o refinamento da visualização, pela alteração do ano inicial e final a considerar para a visualização.
O segundo gráfico implementado, permite constatar o histograma de classificações (*ratings*). Neste é possível perceber as frequências de classificação por elemento da escala (de 0,5 a 5 estrelas).

[10] Uma caraterística deste *dataset* é a esparsidade dos mesmos, uma vez que há muitos itens sem classificações, assim como o a distribuição de ratings (classificações) não é uniforme, ou seja, enquanto uma pequena parte dos itens tendem a obter a maioria das classificações dos utilizadores, uma outra parte, tende a receber poucas classificações por item.

O terceiro gráfico permite verificar a existência de *long tail*, pela visualização de classificações por filme. Para além da escala linear, uma vez que podem existir variações significativas, foi implementada a opção de visualização do gráfico na escala logarítmica, permitindo, verificar a tendência de forma mais refinada.
Tendo em vista a análise dos dados em termos de conteúdo, concretamente, em termos da distribuição de filmes e *ratings* por géneros, foram implementados mais três gráficos.
Foi assim, implementado um gráfico de áreas cumulativo, que permite verificar a evolução anual (ano de lançamento) do número total de filmes por género. Este permite verificar a evolução do número de filmes no total e por género ao longo do tempo, permitindo perceber os géneros predominantes nos filmes. Naturalmente que, uma vez que cada filme pode ser caracterizado por mais que um género, a curva apresentada não está de acordo, com a totalidade dos filmes lançados em cada ano, assim, foi acrescentada uma série comparativa, que permite observar a evolução do lançamento dos filmes, independentemente dos géneros. Este gráfico, permite constatar a tendência de caracterização dos filmes em termos de géneros. Esta implementação também foi implementada de forma interativa, para permitir refinar a visualização. Nesta, para além da possibilidade de variar o limite temporal, também é possível selecionar os géneros a visualizar, permitindo comparar individualmente e em períodos temporais definidos, o número de lançamento de filmes relativamente aos géneros pretendidos. As seleções efetuadas no formulário implementado, permitem personalizar os outros dois gráficos implementados: gráfico de barras de comparação do número de filmes lançados, caracterizados por géneros e histogramas de *ratings vs.* géneros. Neste último, sendo consideradas as frequências em termos relativos às classes a visualizar (*frequency density*), é possível comparar frequências *ratings* por género e filme.

5.4 Recomendações não personalizadas

Estas são todas as recomendações do tipo *Top-N*, com eventual previsão de classificação, cuja filtragem não obriga à identificação/autenticação do utilizador. Todas as implementações efetuadas neste âmbito, salvo, as estatísticas descritivas do *MovieLens Datasets offline* e os gráficos de visualização de dados, resultaram numa lista do tipo *Top-N*, a qual permite efetuar classificações (*ratings*) desde que o utilizador se autentique no sistema.

5.4.1 Procurar informação sobre filmes

Antes de obter informação sobre filmes a partir do site *TMDb*, é necessário encontrar esse mesmo filme no *MovieLens Dataset offline*, por forma a obter o *tmdbId*. Para isso é necessário procurar a ocorrência de uma *string* (texto de procura) nos títulos dos filmes constante da base de dados de filmes. Para o efeito, é utilizada uma técnica de identificação de um padrão de forma aproximada, designada por *fuzzy string matching*. Isto é, a

correspondência de *fuzzy string matching* é um tipo de pesquisa que encontra correspondências mesmo quando os utilizadores falham na ortografia ou introduzam apenas palavras parciais para a pesquisa. Também é conhecido como correspondência aproximada de *strings*. É mais uma técnica de *NLP* (*Natural Language Processing*).
Existe uma implementação de um biblioteca para *python* que utiliza a distância de *Levenshtein* (SeatGeek, 2011) e foi a empregue neste protótipo. Neste, é construído um mapa reverso dos títulos dos filmes para *movieId*, ao qual uma função de procura baseada na técnica *fuzzy string matching* é aplicada, a qual devolve um determinado número de *movieIds*, que mais se aproximam da frase de pesquisa.
Com esta lista, é utilizada a API de acesso à base de dados do *TMDb*, para serem obtidos os atributos pretendidos dos filmes (*e.g.*, cartaz, elenco, *trailer*). Este site permite a solicitação de uma chave de forma gratuita para utilizar, desde que seja publicitada a origem dos dados como sendo do *TMDb*. Foi o que foi feito, razão pela qual se colocou o *logo* do *TMDb* no rodapé do protótipo. Este pormenor, bem como um exemplo da utilização desta funcionalidade, podem ser constatados nas Figura 5-17 e Figura 5-18.

5.4.2 Obter filmes similares exclusivamente pela análise da descrição

As implementações efetuadas, permitem obter o *Top-N* de filmes similares a um determinado filme, com base no conteúdo (descrição textual de um item). No âmbito do *MovieLens Dataset offline*, estamos a falar das características dos filmes, concretamente: título, géneros ou, eventualmente, o ano de lançamento (incluído no título entre parênteses, podendo não estar presente).
Efetuou-se uma implementação que pressupõe propor os filmes similares baseado nos géneros e utilizando como medida de similaridade, a similaridade do cosseno entre géneros de filmes (ver secção 4.3.2). Neste caso é criado um *dataframe* de géneros, cujas colunas correspondem aos géneros dos filmes, com representação binária para os géneros presentes (um, se o filme possui o género ou zero, se não possui). Este *dataframe* é indexado por *movieId*. Este é o *dataframe*, sobre a qual são calculadas as similaridades do cosseno, sendo os filmes representados por vetores de géneros (características/atributos). Para encontrar os itens similares de um determinado filme, basta ordenar a coluna ou linha da matriz de similaridades, correspondente a um determinado filme, por ordem decrescente, constituindo a recomendação do tipo *Top-N*.
Para as outras implementações baseadas em filtragem baseada em conteúdo, utilizou-se a classe *TfidfVectorizer* do sub-módulo *sklearn.feature_extraction.text* para *Python* (scikit-learn developers, 2020a). Este sub-módulo é parte da biblioteca *Machine Learning in Python scikit-learn*, que implementa várias ferramentas de *ML* para *Python* (Pedregosa et al., 2011). Esta classe permite vetorizar um "saco de termos" ("*bag of words*") das características dos itens e construir um conjunto de vetores *TF-IDF* caracterizadores dos itens (ver secção 4.5). Chama-se a atenção que os vetores *TF-IDF* estão normalizados (módulo dos vetores é 1), pelo que para obter a matriz de similaridades (similaridade do cosseno), basta calcular o produto

interno entre os vetores *TF-IDF* normalizados. A forma de encontrar os itens similares de um determinado filme, já foi apresentada no final da descrição da implementação anterior.
Foram feitas implementações para "sacos de termos" constituídos por: géneros como palavras, palavras constantes do título e a conjugação das duas (frases caracterizadoras dos itens, que incluem palavras do título e os géneros como palavras).
A utilização das palavras constantes do título permite a procura de filmes similares baseados no título. Esta permite encontrar similaridades entre títulos de filmes, caracterizadores de sequelas ou outras eventuais palavras-chave. Naturalmente, que antes da criação do "saco de termos", os títulos foram pré-processados, por forma a eliminar determinadas palavras designadas por *stop-words*[11] em inglês. Para além disso, também foram alterados alguns termos característicos dos filmes, com a retirada dos parênteses, espaços após a *sub-string* "Part " e termos do tipo "I", "II", "III", "IV", "V", entre outros.

5.4.3 Obter o Top de Popularidade dos filmes

Esta é a primeira implementação de filtragem colaborativa, isto é, com base os *ratings* de outros utilizadores constantes de *MovileLens Datasets offline*, no âmbito da técnica designada como Filtragem Colaborativa simples, uma vez que não é personalizada, nem, no caso do deste protótipo, exige a identificação/autenticação do utilizador (ver secção 4.4).
Esta permite calcular uma previsão de classificação a atribuir a cada um dos filmes, com base nas classificações conhecidas. É de salientar que na implementação, foi considerado um percentil de 90, o que permitiu considerar apenas os filmes com um número de votos superior a pelo menos 90% dos filmes. No caso presente, estamos a falar de 10% do *MovileLens Datasets offline*.
Estes são os candidatos a serem recomendados, pelo que é a este conjunto que é aplicada a equação (4-5). Esta lista, é então ordenada por ordem decrescente da média pondera calculada para cada filme, sendo esta a previsão de classificação a atribuir pelo utilizador não identificado. A seleção do *Top-N* desta lista ordenada, constitui o *Top-N* de popularidade. Um exemplo desta recomendação para o *MovileLens Datasets offline* adotado, consta da Figura 5-19.

5.5 Recomendações personalizadas

Estas correspondem a todas as recomendações do tipo *Top-N*, com previsão de classificação, cuja filtragem obriga à identificação/autenticação do utilizador. Todas as implementações efetuadas neste âmbito, resultaram numa lista do tipo *Top-N*, a qual permite efetuar classificações (*ratings*) por parte de utilizadores autenticados.

[11] No âmbito do NLP (processamento de linguagem natural) são palavras a eliminar antes do processamento, uma vez que se referem às palavras mais comuns num idioma (*e.g.*, em inglês *The*, *A*, *An*).

Para obter recomendações personalizadas é fundamental criar perfis de utilizadores, em que os dados podem ser divididos em itens que um utilizador gosta e itens que não gosta. As recomendações podem ser apresentadas a um utilizador ativo, prevendo classificações de itens que um utilizador não viu e construindo uma lista de itens ordenados pelas suas preferências. O mecanismo de predição usa classificações de outros utilizadores (ou de itens) ou informações de conteúdo e, em seguida, prevê o quanto o utilizador gostaria de determinado item. Outra forma de fazer recomendações é entregar recomendações seletivamente ao utilizador através de uma lista ordenada de itens, as recomendações *Top-N* (Cremonesi et al., 2010; Ghazanfar & Prügel-Bennett, 2010; Gupta & Gadge, 2015).
Para construir o modelo do utilizador, o seu perfil conterá a informação de todos os filmes que avaliou, haverá ainda o registo da preferência em termos de géneros dos filmes, ou outros atributos de conteúdo, por *feedback* implícito, através da sua interação com o sistema, havendo dedução dos géneros de preferência do utilizador, com base na ponderação das classificações que efetuou nos itens.
A aprendizagem dos interesses de cada utilizador é efetuada com base no conjunto de avaliações realizadas, podendo, assim, assumir com algum grau de confiança que o utilizador demostra interesse por itens (filmes) que apresentem características idênticas.

5.5.1 Obter filmes similares pela construção de perfis de itens e utilizador

A implementação efetuada permite obter o *Top-N* de filmes preferidos por um utilizador identificado (ativo), com base no perfil de utilizador criado a partir dos perfis de filmes que ele já classificou, incluindo previsão da classificação a atribuir. Tal como já foi descrito na secção 5.4.2, no âmbito do *MovieLens Dataset offline*, para a criação dos perfis de filmes (itens) são consideradas as características dos filmes, concretamente: título, géneros ou, eventualmente, o ano de lançamento (incluído no título entre parênteses, podendo não estar presente). Na implementação efetuada, foram considerados os géneros dos filmes, para a prova de conceito. Aqui também foi implementada a técnica *TF-IDF* para determinar a similaridade entre utilizadores e itens (ver secção 4.5.2).
Para a construção do perfil de item, também é criado um *dataframe* de géneros, cujas colunas correspondem aos géneros dos filmes, com representação binária para os géneros presentes (um, se o filme possui o género ou zero, se não possui). Este é normalizado (cada célula é dividida pela raiz quadrada da soma da linha). Este *dataframe* é indexado por *movieId*, constituindo os vetores *TF* (*Term Frequency*), caracterizadores de cada filme.
Para o cálculo dos valores de *IDF* (*Inverse Document Frequency*), calculam-se os respetivos *DF* (*Data Frequency*), por género. Tendo em atenção o número total de filmes, aplica-se a equação (4-8), para calcular o *IDF*, por género. Multiplicado, cada vector *TF* pelo vetor *IDF*, género a género, obtemos os vetores *TF-IDF*, caracterizadores de cada filme. Estes vetores são normalizados (o módulo dos vários vetores é igual a um), constituindo os perfis dos itens (filmes).

Para a criação do perfil do utilizador é necessário considerar as classificações (*ratings*) já efetuadas pelo utilizador ativo. Para ser possível a previsão da classificação a atribuir a filmes ainda não classificados, são consideradas as classificações efetivas e não a classificação binária (1, para gosta e -1, para não gosta). Preenchendo as classificações ainda não efetuadas a zero, para o utilizador ativo, é criado um vetor (série) de classificações. Para a criação do perfil de utilizador, obtém-se o *TF-IDF* do utilizador, por género, efetuando o produto interno entre o vetor formado por cada coluna, que constitui os perfis dos filmes e o vetor de classificações, obtendo assim, o vetor *TF-IDF*, que constitui o perfil do utilizador, após normalização.
Para obter a similaridades entre o utilizador e todos os itens do sistema, é criada a lista de similaridades, efetuando o produto interno entre o perfil do utilizador e cada perfil de item. Para determinar a similaridade entre o utilizador e os itens ainda não classificados e obter as similaridades entre o utilizador e os filmes já classificados, é necessário retirar os filmes já classificados desta série de similaridades, criando uma série de similaridades do utilizador com os itens já classificados, ambas indexadas por *movieId*.
Para obter a recomendação do tipo *Top-N*, a série de itens ainda não classificados é ordenada por ordem decrescente de similaridade e obtidos os *N* primeiros elementos.
Para calcular a previsão da classificação à aplicada a equação (4-11). Naturalmente, que esta é única, uma vez que o cálculo é efetuado exclusivamente com base nos *ratings* já efetuados, não considerando *ratings* de outros utilizadores.

5.5.2 Obter filmes com base na similaridade entre utilizadores

A implementação efetuada, permite obter o *Top-N* de filmes preferidos por um utilizador identificado (ativo), com base nas classificações atribuídas pelos utilizadores na sua vizinhança, incluindo as previsões de classificação a atribuir. Tal como foi descrito na secção 4.6.1, o processo divide-se nas fases de cálculo das semelhanças entre utilizadores, determinação das vizinhanças para os utilizadores e previsão das classificações para itens ainda não classificados pelo utilizador ativo (incluindo o *Top-N* de preferências), utilizando, para o efeito, o *MovieLens Dataset offline*, combinado com os *ratings* já atribuídos pelos utilizadores deste sistema (BD de *ratings* deste sistema de recomendação).
Os itens já classificados permitem selecionar um conjunto de *k* "vizinhos" do utilizador identificado, o qual é constituído pelos utilizadores com valores de semelhança mais elevados em relação a este. Quanto mais similar se encontrar um utilizador, ou vizinhança (*k-nearest neighbour*) de utilizadores do utilizador ativo, que é aquele a quem se está a fazer a predição, maior será a sua influência (peso) na recomendação a efetuar.
Começa-se por criar um *dataframe* a partir dos *ratings*, obtidos do *MovieLens Dataset offline*. As classificações constantes da *BD* de *ratings*, também são adicionados ao mesmo *dataframe*. São calculadas as médias dos utilizadores, a partir do *dataframe* criado. Para cada classificação constante do *dataframe*, é subtraída a média de cada utilizador, permitindo criar um *rating* ajustado.

É criada uma matriz de classificações utilizador-item, com os *ratings* ajustados (*dataframe user_feature*s). Uma vez que existem *ratings* ajustados com o valor zero, o preenchimento dos *ratings* ajustados que ainda não têm classificação (valores a *"Nan"*), poderão ser preenchidos de duas formas: média dos *ratings* ajustados do utilizador, isto é, média das linhas da matriz ou média dos *ratings* ajustados do item, isto é média das colunas da matriz (Pathak et al., 2019). Foram implementadas as duas aproximações, para serem comparadas experimentalmente.

Para calcular a similaridade entre utilizadores, utilizando a similaridade do cosseno, é utilizado o *dataframe user_features*. Como se subtraiu aos *ratings* a média de classificação do utilizador, antes de calcular a similaridade do cosseno, esta medida é equivalente à correlação de *Pearson*, quando os utilizadores classificam os mesmos *itens* (Eksnd et al., 2010).

Este *dataframe* de similaridades é a base para a criação dos *k-nearest neighbour* do utilizador ativo.

Para o cálculo da previsão da classificação do utilizador ativo, para um item ainda não classificado por este, é aplicada a equação (4-12), após a seleção dos *ratings* ajustados e das similaridades (correlações entre utilizadores), dos *k-nearest neighbour* deste.

Para a recomendação do tipo *Top-N*, são selecionados todos os filmes ainda não classificados pelo utilizador ativo, mas que foram classificados pelos seus vizinhos, constituindo a lista de filmes a considerar. Para estes são calculadas as previsões de classificação, conforme o parágrafo anterior, ordenadas por ordem decrescente de previsão e obtidos os *N* primeiros elementos e respetivas previsões.

5.5.3 Obter filmes com base em filtragem híbrida

A implementação efetuada procedeu à combinação das técnicas de CF com a CBF, onde foram atribuídos pesos a cada técnica, dando origem a um sistema de HF Ponderado. Esta, permite obter o *Top-N* de filmes preferidos por um utilizador identificado (ativo), incluindo as previsões de classificação a atribuir.

Procedeu-se a uma implementação adaptada a partir da técnica *Dynamic Weighted*, apresentada em (Do et al., 2020), a qual combina, de forma linear, os resultados das técnicas descritas nas secções 5.5.2 (Obter filmes com base na similaridade entre utilizadores) e 5.5.1 (Para obter recomendações personalizadas é fundamental criar perfis de utilizadores, em que os dados podem ser divididos em itens que um utilizador gosta e itens que não gosta. As recomendações podem ser apresentadas a um utilizador ativo, prevendo classificações de itens que um utilizador não viu e construindo uma lista de itens ordenados pelas suas preferências. O mecanismo de predição usa classificações de outros utilizadores (ou de itens) ou informações de conteúdo e, em seguida, prevê o quanto o utilizador gostaria de determinado item. Outra forma de fazer recomendações é entregar recomendações seletivamente ao utilizador através de uma lista ordenada de itens, as recomendações *Top-N* (Cremonesi et al., 2010; Ghazanfar & Prügel-Bennett, 2010; Gupta & Gadge, 2015).

Para construir o modelo do utilizador, o seu perfil conterá a informação de todos os filmes que avaliou, haverá ainda o registo da preferência em termos de géneros dos filmes, ou outros atributos de conteúdo, por *feedback* implícito, através da sua interação com o sistema, havendo dedução dos géneros de preferência do utilizador, com base na ponderação das classificações que efetuou nos itens.
A aprendizagem dos interesses de cada utilizador é efetuada com base no conjunto de avaliações realizadas, podendo, assim, assumir com algum grau de confiança que o utilizador demostra interesse por itens (filmes) que apresentem características idênticas.

Obter filmes similares pela construção de perfis de itens e utilizador). Estes constituem Filtragem Colaborativa e Filtragem de Conteúdo, respetivamente.
O cálculo da previsão $P_{u,i}^{(CF)}$ é efetuado nos moldes descritos em 5.5.2. O cálculo da previsão $P_{u,i}^{(CBF)}$ é efetuado nos moldes descritos em 5.5.1. Efetua-se ainda o cálculo do α_u pela aplicação da equação (4-17).
Para o cálculo da previsão da classificação do utilizador ativo, para um item ainda não classificado por este, é aplicada a equação (4-16).
Para a recomendação do tipo *Top-N*, são selecionados todos os filmes ainda não classificados pelo utilizador ativo, mas que foram classificados pelos seus vizinhos, constituindo a lista de filmes a considerar. Para estes são calculadas as previsões de classificação, conforme o parágrafo anterior, ordenadas por ordem decrescente de previsão e obtidos os *N* primeiros elementos e respetivas previsões.

6. Avaliação experimental

O protótipo implementado permite obter informação, estatísticas textuais e gráficas, bem como recomendações sobre filmes e classificações (*ratings*), a partir do *MovieLens Latest Datasets* pequeno, constituído por 100 836 *ratings*, aplicado a 9 742 filmes, por 610 utilizadores, atualizado em setembro de 2018 (GroupLens, 2020b).
Como forma de investigar o *MovieLens Latest Datasets* em questão, são apresentados e analisados os *datasets*. Para utilizadores não identificados, as recomendações disponibilizadas sobre o *MovieLens Latest Datasets* são apresentadas e analisadas, sendo a avaliação efetuada de forma empírica.
Para utilizadores identificados, na elaboração de experiências, testes e avaliação dos algoritmos de recomendação implementados, foi utilizado o ambiente de desenvolvimento científico baseado em *python, Spyder*[12], sobre o *MovieLens Latest Datasets* pequeno. O *dataset* de *ratings* é subdividido em dois: *dataset* de treino ou base para as recomendações e *dataset* de teste, que permite aferir a qualidade das recomendações.
As secções seguintes permitem explorar todos estes aspetos.

6.1 Análise do conjunto de dados utilizado

Os dados de classificações (*ratings*) constantes do *MovieLens Latest Datasets*, permitem reconhecer a informação estatística constante da Figura 6-20. Um aspeto importante tem a ver com a esparsidade das classificações que é de 98,3%. Fundamentalmente a matriz de classificações (matriz utilizador-item), só tem ≈ 1,7% dos *ratings*, o que a torna esparsa.

[12] É um ambiente científico gratuito e *open source* desenvolvido em *python* para *python* para cientistas, engenheiros e analistas de dados (*https://www.spyder-ide.org/*).

Classificaram filmes, 610 utilizadores, sendo que o utilizador com *userId* 414, classificou 2 698 filmes. Foram classificados 9 724 dos 9 742 filmes, havendo 18 filmes que não têm qualquer classificação. O filme mais classificado é o que possui o *movieId* 356, o qual foi classificado 329 vezes (trata-se do filme *"Forrest Gump (1994)"*, conforme pode ser constatado mais à frente no *Top 10 – Most Rated Movies*, na Figura 6-26).

Descriptive statistics

Total Ratings: 100,836
Sparsity: 98.3 %

Descriptive	UserId	MovieId	Rating	Timestamp
unique	610	9724	nan	
top	414	356	nan	
freq	2698	329	nan	
mean			3.502	2008/03/19 05:03:27
min			0.500	1996/03/29 06:03:55
25%			3.000	2002/04/18 09:04:46
50%			3.500	2007/08/02 08:08:02
75%			4.000	2015/07/04 07:07:44
max			5.000	2018/09/24 02:09:30
std			1.043	

Figura 6-20: Informação descritiva do *MovieLens Latest Datasets* pequeno

No que diz respeito à distribuição dos *ratings*, a média de classificação é 3,502, com um desvio padrão de 1,043, tendo sido classificados numa escala entre 0,5 e 5,0. Em termos de quartis, verifica-se que até 25% dos utilizadores, classificaram os filmes com pelo menos 3,0, ou seja, há muito poucas classificações de filmes considerados maus. Até 50% dos utilizadores classificaram os filmes com pelo menos 3,5 de rating e 75% dos utilizadores classificaram os filmes até 4,0 de rating, logo 25% dos utilizadores classificaram com 4,5 e 5 os filmes.

Na distribuição de datas de classificação, os filmes foram classificados entre 1996/03/29 e 2018/09/24. Pode ainda ser constatado que pelo menos 25% dos filmes foram classificados em ≈ 6 anos (até 2002/04), 50% em ≈ 11 anos (até 2007/08) e 75% em ≈ 19 anos (até 2015/07).

Numa primeira análise pretende-se verificar, por ano, o número de filmes lançados *vs.* o número de ratings efetuados, em termos relativos. Concretamente, pretende-se saber o período de disponibilização de filmes e *ratings* efetuados, assim como, perceber o crescimento das duas variáveis em termos temporais. O resultado é o constante na Figura 6-21.

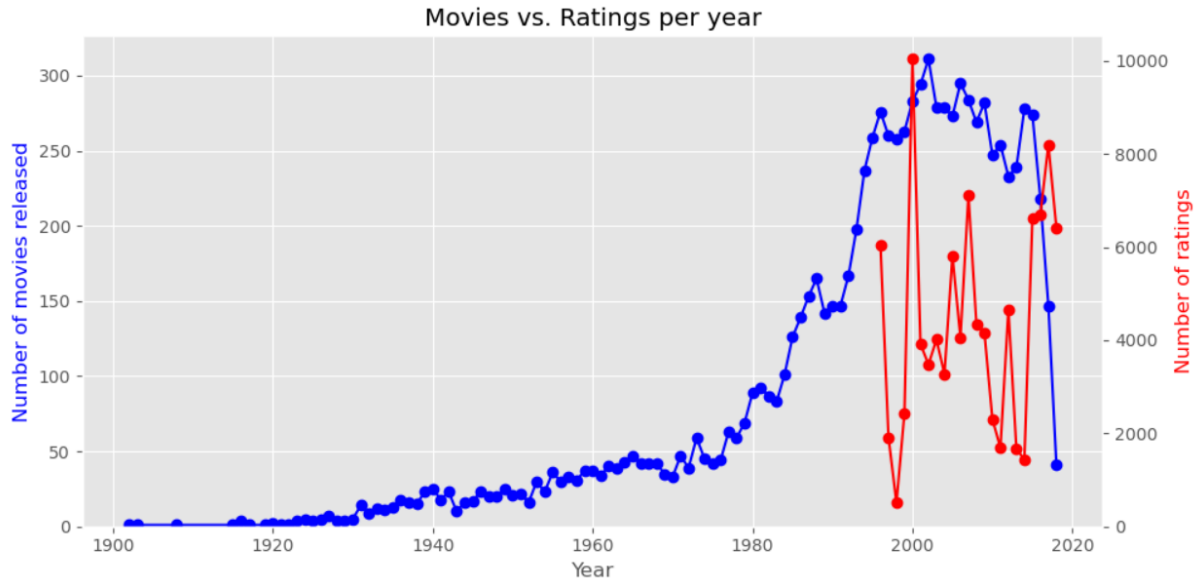

Figura 6-21: Comparação entre o número de filmes lançados e número de *ratings* por ano

Assim, o período de lançamento de filmes é entre 1902 e 2018, tendo sido classificados, entre 1996 e setembro de 2018, conforme já descrito.

O número de filmes lançados foi aumentando de forma exponencial até 2001, tendo estabilizado nos anos seguintes até 2014, ano em que existiu uma quebra significativa. Em termos relativos, a atribuição de *ratings* aumentou significativamente entre 1998 e 2000, tendo, inclusivamente ultrapassado o número de lançamentos, tendo abrandado logo e mantido e ao longo dos anos seguintes.

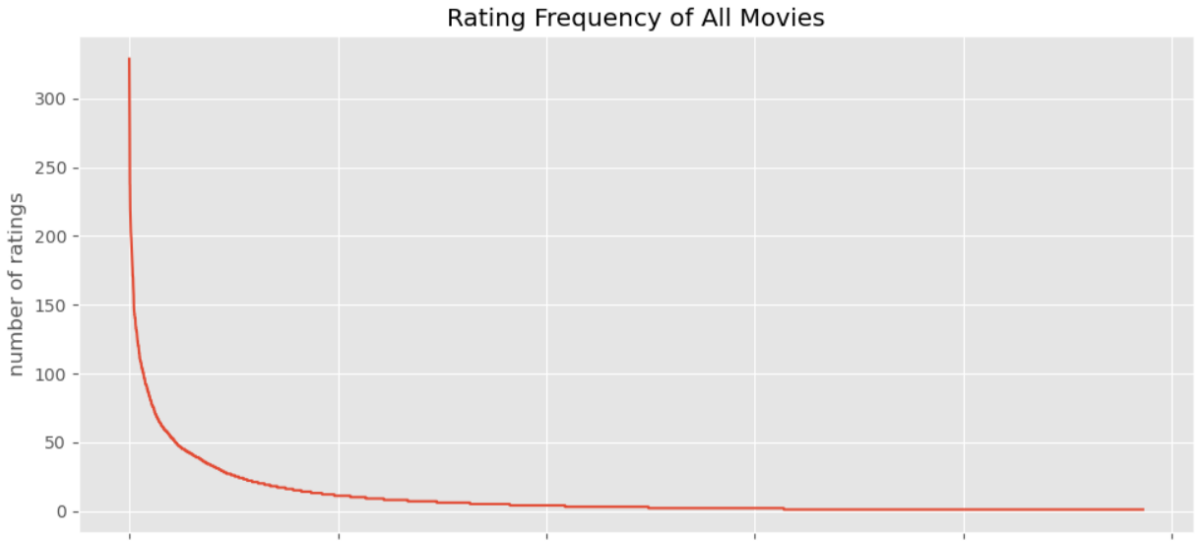

Figura 6-22: Efeito de *long tail* do *MovieLens Latest Datasets* pequeno

Pela análise do gráfico da Figura 6-22, é possível verificar que a distribuição de *ratings* (classificações) não é uniforme, ou seja, enquanto uma pequena parte dos itens tendem a obter a maioria das classificações dos utilizadores, uma outra parte, tende a receber poucas

classificações por item. Esta distribuição do gráfico, é conhecido por *long tail* (Anderson, 2004, 2006).

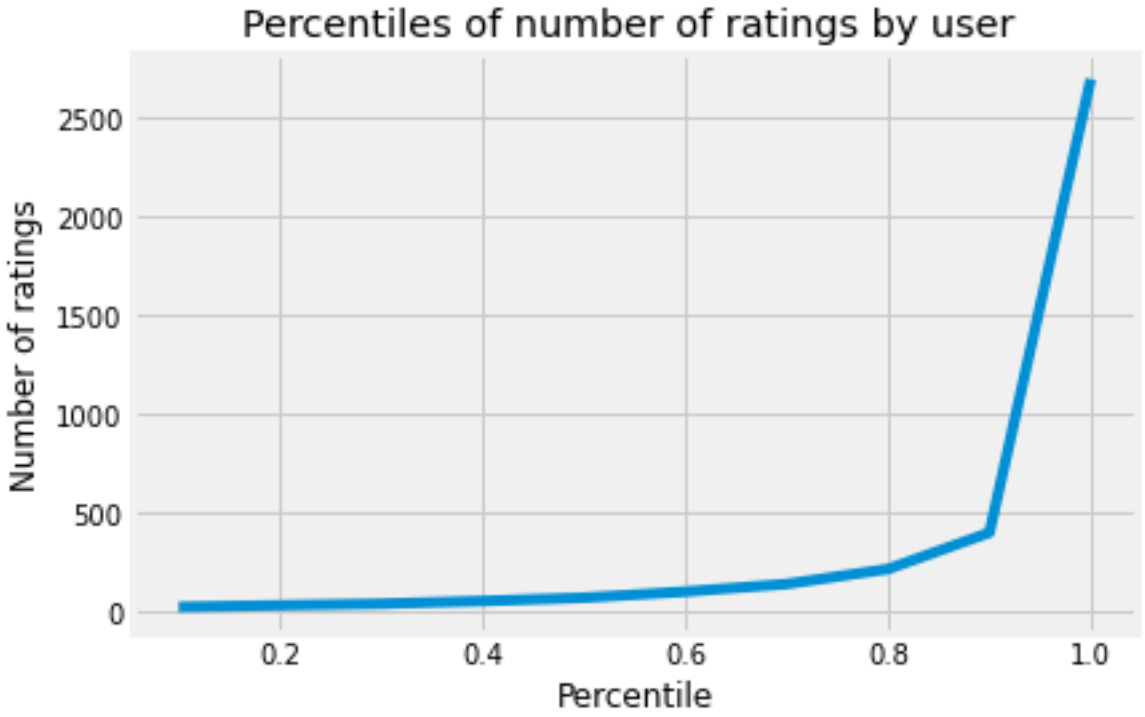

Figura 6-23: Percentis do número de avaliações por utilizador

Ao analisar o gráfico da Figura 6-23 pode verificar-se que há cerca de 40% dos utilizadores que fizeram até 54 classificações, só um utilizador fez 2 698 classificações e 90% fez até 400 classificações.

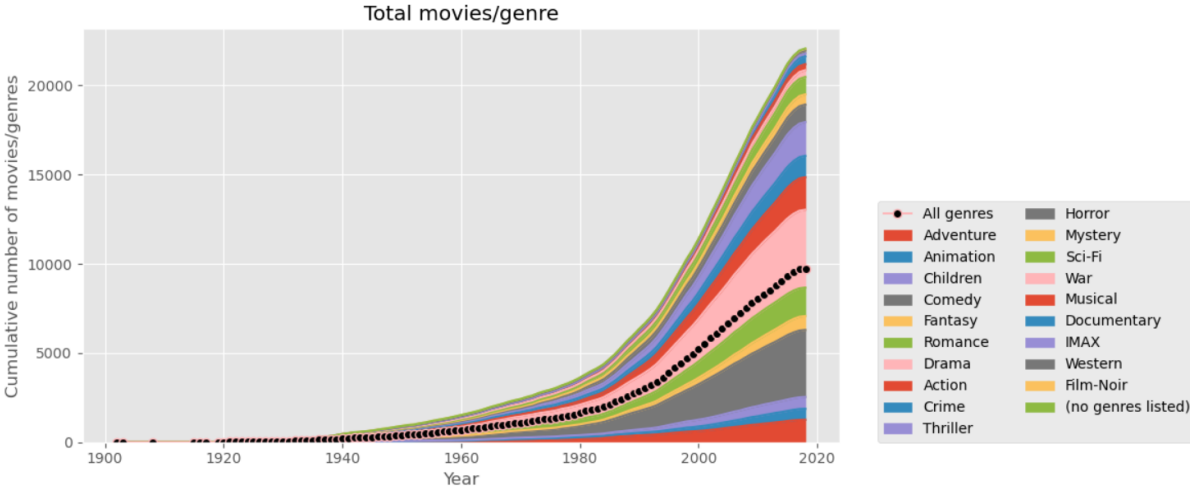

Figura 6-24: Número cumulativo de filmes por géneros *vs.* número total de filmes

Em relação ao número de filmes disponibilizados por ano, pretende-se conhecer os géneros dos mesmos e a sua ocorrência por filme. Pelo gráfico da Figura 6-24 conclui-se que em

média os filmes têm pelo menos dois géneros, sendo os géneros com ocorrência em maior número a comédia e o drama.

Conhecer a distribuição de *ratings* por género *vs.* distribuição de *ratings* em geral e em termos relativos (ver gráfico da Figura 6-25), permite identificar a consistência de *ratings*, ou eventuais desvios (*e.g*. os géneros comédia e crime têm mais classificações em geral). Pode-se concluir que todos os géneros mostram uma distribuição similar (distribuição normal, com ligeira inclinação para a direita, ou seja com tendência para a existência de classificações mais elevadas), exceto o género terror que é ligeiramente fletido à esquerda, logo com classificações mais baixas.

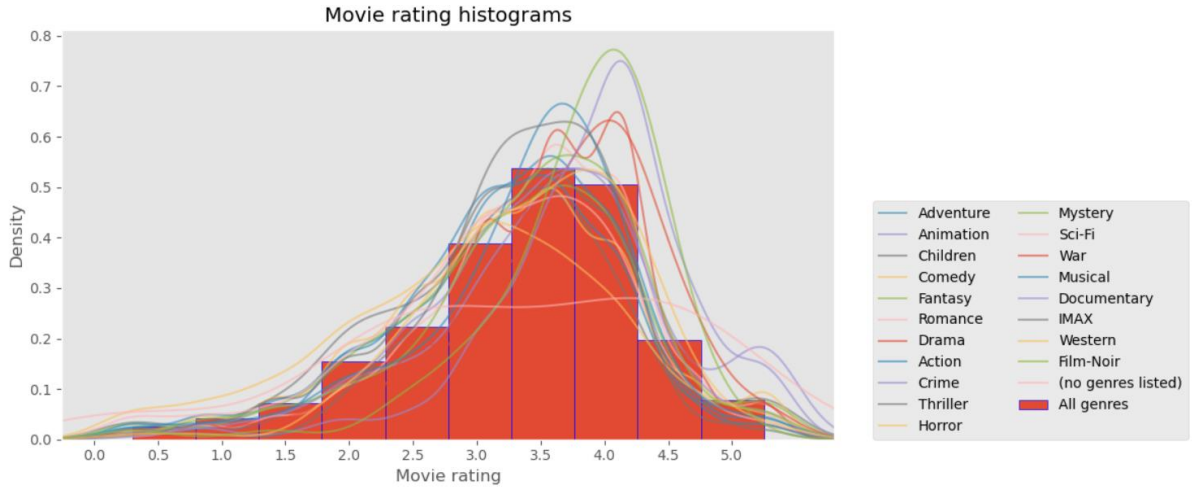

Figura 6-25: Distribuição por géneros, sobre a distribuição do total de ratings

Foram ainda verificados os filmes mais classificados, mais recentes e mais antigos. Para o efeito, foram criadas listas de *Top-N* com esta informação, permitindo facultar acesso a mais informação dos filmes listados, bem como permitir obter recomendações baseadas em conteúdo, de filmes similares, ou ainda, efetuar classificações, após autenticação no sistema. A título de exemplo na Figura 6-26 é apresentado o *Top 10* dos filmes mais classificados de sempre. Estes *outputs*, poderão ser considerados uma das formas de ultrapassar o problema de *cold start*, para novos utilizadores do sistema, uma vez que constituem "recomendações" do tipo *Top-N* não personalizadas.

Top 10 - Most Rated Movies

#	Title	Year	Total Ratings	Mean Rating	Info	Similars	Rating
1	Forrest Gump	1994	329	4.2			
2	The Shawshank Redemption	1994	317	4.4			
3	Pulp Fiction	1994	307	4.2			
4	The Silence of the Lambs	1991	279	4.2			
5	The Matrix	1999	278	4.2			
6	Star Wars: Episode IV - A New Hope	1977	251	4.2			
7	Jurassic Park	1993	238	3.8			
8	Braveheart	1995	237	4.0			
9	Terminator 2: Judgment Day	1991	224	4.0			
10	Schindler's List	1993	220	4.2			

Figura 6-26: *Top 10* dos filmes mais classificados de sempre e respetivos *ratings* médios

6.2 Recomendações não personalizadas de filmes

No âmbito das recomendações baseadas exclusivamente em conteúdo para utilizadores não identificados, fizeram-se experiências, para obtenção de listas *Top-N* que apresentassem filmes semelhantes por géneros, por título e por uma combinação de género e título. No caso das experiências efetuadas por géneros, exclusivamente, os resultados demonstram que os géneros são respeitados, mas eventuais sequelas, podem não ser identificadas como similares. No caso das experiências efetuadas exclusivamente por título, os resultados demonstram que as sequelas são normalmente identificadas, mas os géneros não são respeitados.

As experiências efetuadas, com a combinação de título e géneros, normalmente, tanto os géneros são respeitados, como as sequelas são identificadas e recomendadas. Por exemplo, para o filme *"Toy Story (1995)"*, o resultado obtido é apresentado na Figura 6-27.

Nas recomendações de filtragem colaborativa para utilizadores não identificados, obteve-se um *Top 25* de popularidade, baseada nas classificações constantes do *MovileLens Datasets offline*. Como já foi descrito, para cada filme desta lista, foi calculada a previsão de classificação, através de uma pontuação ponderada. Verificou-se que, o segundo filme com o maior número de *ratings* no *Top 10* de filmes mais classificados (ver Figura 6-26), passou para número um, neste *Top 25*, uma vez que possui uma média de *ratings* superior (Figura 5-19).

Top 15 - Similar Movies by genre and title (cosine similarity)

Title: Toy Story
Year: 1995
Genres: ['Adventure', 'Animation', 'Children', 'Comedy', 'Fantasy']

#	Title	Year	Genres	Info	Similars	Rating
1	Toy Story 2	1999	['Adventure', 'Animation', 'Children', 'Comedy', 'Fantasy']			
2	Toy Story 3	2010	['Adventure', 'Animation', 'Children', 'Comedy', 'Fantasy', 'IMAX']			
3	The Toy	1982	['Comedy']			
4	We're Back! A Dinosaur's Story	1993	['Adventure', 'Animation', 'Children', 'Fantasy']			
5	The Story of Us	1999	['Comedy', 'Drama']			
6	Toy Soldiers	1991	['Action', 'Drama']			
7	Up	2009	['Adventure', 'Animation', 'Children', 'Drama']			
8	The NeverEnding Story	1984	['Adventure', 'Children', 'Fantasy']			
9	L.A. Story	1991	['Comedy', 'Romance']			
10	The Wild	2006	['Adventure', 'Animation', 'Children', 'Comedy', 'Fantasy']			
11	A Christmas Story	1983	['Children', 'Comedy']			

Figura 6-27: Parte da Recomendação *TOP 15* para o filme "*Toy Story (1995*)"

6.3 Recomendações personalizadas de filmes

Para os testes e análise de resultados das recomendações personalizadas, o *dataset* de *ratings*, foi subdividido em treino e teste, permitindo efetuar as recomendações sobre o conjunto de treino e avaliar os algoritmos, pela confrontação com o conjunto de testes. Aqui, é introduzida uma variável que determina a percentagem de dados utilizados nos conjuntos de treino e teste, chamada *x*. Um valor de *x=0,8*, indica que foram utilizados 80% dos dados para treino e 20% para testes. O processo passa por utilizar as equações de previsão de classificação, sobre os *ratings* que constituem o conjunto de teste para calcular as respetivas previsões. Para efetuar esta subdivisão é utilizado o módulo *model_selection* da biblioteca *sklearn* (scikit-learn developers, 2020b). Esta permite subdividir *arrays* ou matrizes de forma aleatória em conjuntos de treino e teste.

Naturalmente, que os algoritmos de previsão trabalham sobre o conjunto de dados de teste para aferir o erro entre as previsões efetuadas e os *ratings* efetivos, através de métricas de precisão estatística.

As métricas de precisão estatística utilizadas para aferirem a precisão das recomendações e, consequentemente, dos algoritmos implementados foram: o *MAE* (*Mean Absolute Error*) e o *RMSE* (*Root Mean Square Error*).

Nas secções seguintes serão avaliados e analisados os resultados dos algoritmos implementados, podendo no final concluir pelo melhor, dependendo das condições de aplicação.

6.3.1 Recomendação pela construção de perfis de itens e utilizadores

Os perfis de utilizadores, são criados com base nas classificações que estes fizeram e que constam do *dataset* de treino. Como já foi referido, a previsão depende integralmente das classificações já efetuadas pelo utilizador ativo (simulado com os que pertencem ao *dataset* de teste e que servem para efetuar as previsões), pelo que, independentemente do filme a classificar, a previsão é semelhante, conforme pode ser constatado pela equação (4-11).
O único parâmetro que poderá fazer variar a precisão das previsões é a variável *x*, que determina as dimensões dos *datasets* de treino e teste. Pretende-se assim, verificar experimentalmente, qual o melhor rácio a considerar nos testes seguintes, avaliando a sensibilidade desta variável ou parâmetro.

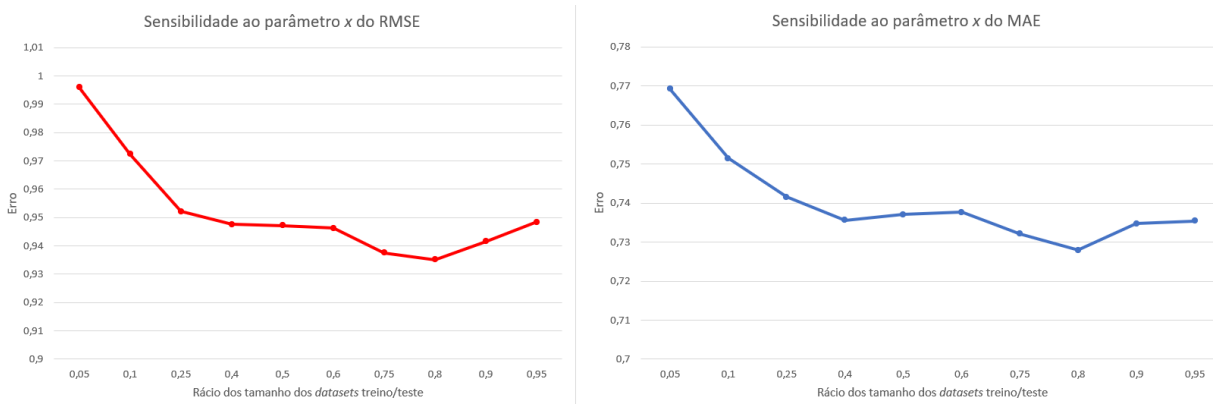

Figura 6-28: Sensibilidade ao rácio das dimensões dos *datasets* treino/teste da CBF

Conforme pode ser verificado pelo gráfico da Figura 6-28, o valor mais conveniente para a variável *x=0,8*, uma vez que é aquele que, tanto no *RMSE*, como no *MAE*, apresenta o menor valor e por conseguinte melhor precisão.
Os resultados obtidos em (Do et al., 2020), pelo algoritmo de recomendação baseado em conteúdo, para *x=0,8*, apresenta um *RMSE* ≈ 1,2. Logo neste estudo, aparentemente, foi conseguida uma melhor recomendação (*RMSE* ≈ *0,938* para *x=0,8*).

6.3.2 Recomendação colaborativa baseada na similaridade entre utilizadores

Conforme descrito na secção 5.5.2, foram efetuadas implementações para a substituição dos valores *"Nan"*, na matriz de classificações utilizador-item (itens ainda não classificados), considerando, para a primeira implementação, a média dos *ratings* ajustados do utilizador e, para a segunda implementação, a média dos *ratings* ajustados do item.
Efetuando todas as experiências com a variável *x=0,8*, foi testada a sensibilidade dos erros de previsão, relativamente ao tamanho da vizinhança. Os resultados obtidos constam dos gráficos da Figura 6-29.

No caso do preenchimento dos *ratings* ainda não efetuados na matriz utilizador-item, com a média dos *ratings* ajustados do utilizador (*média do utilizador* nos gráficos), verifica-se que o número de vizinhos a considerar para as recomendações, situa-se entre os 10 e 40 vizinhos (quer no *RMSE* quer no *MAE*).

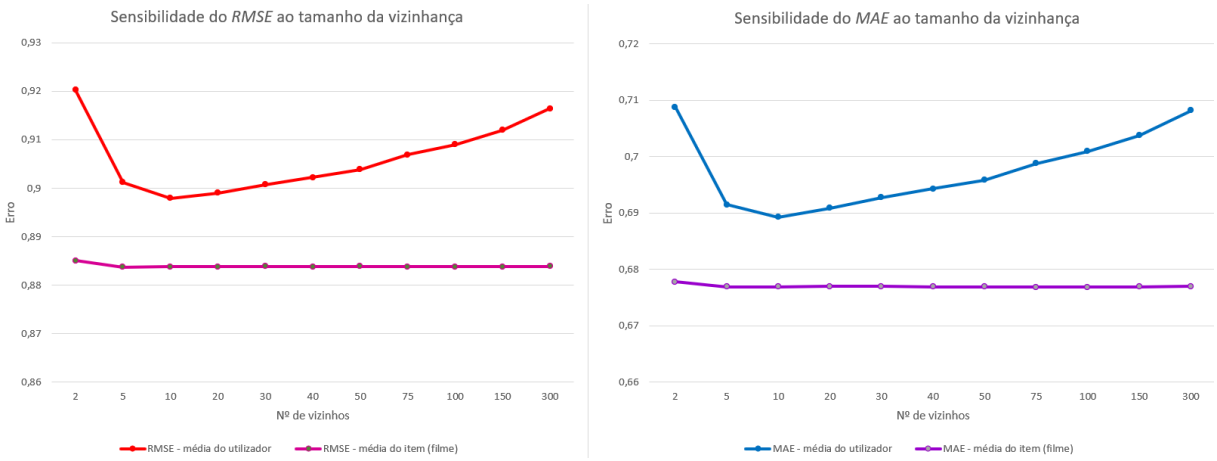

Figura 6-29: Sensibilidade ao número de vizinhos da CF

No caso do preenchimento dos *ratings* ainda não efetuados na matriz utilizador-item, com a média dos *ratings* ajustados do item (*média do item* nos gráficos), verifica-se que o sistema é pouco sensível ao número de vizinhos.

Verificou-se ainda que a implementação baseada na *média do item*, produziu melhor precisão do SR CF. Considerando entre 10 e 40, vizinhos, o $MAE \approx 0{,}69$ na implementação baseada na *média do utilizador vs.* $MAE \approx 0{,}67$ na implementação baseada na *média do item*.

Verificou-se, ainda, que para além da vizinhança, se for utilizada a média dos *ratings* ajustados do item, para preencher os *ratings* ainda não efetuados, as recomendações são melhores e pouco sensíveis ao número de vizinhos.

6.3.3 Recomendação híbrida

Com o objetivo de verificar a possibilidade de melhorar as recomendações, fez-se na implementação descrita na secção 5.5.3.

Tendo em atenção os *datasets* utilizados para os testes, a técnica implementada considerou sempre um fator de confiança $\alpha_u = 0.9$ para a predição CF. Todos os utilizadores, classificaram pelo menos 14 itens. Assim, com a exceção dos testes efetuados com 2 vizinhos, os resultados traduzem que a implementação HF, acompanha a implementação CF, com um erro ligeiramente superior e, por conseguinte, a implementação CBF, não contribui para a melhoria das recomendações. Contudo, quando considerados apenas 2 vizinhos, a implementação CBF contribui para a melhoria das recomendações, conforme Figura 6-30.

Estes resultados, confirmam que os SR CBF são bons ao encontrarem itens semelhantes e não precisam de muitas informações sobre os *ratings* do utilizador ativo, podendo ser utilizados quando ainda não há um perfil de utilizador com muitas preferências.

É assim uma realidade que o problema de *cold start*, pode ser ultrapassado com a contribuição das técnicas CBF. Contudo, no *dataset* utilizado, essa evidência, na implementação HF, resume-se à experiência com menos de 5 vizinhos.

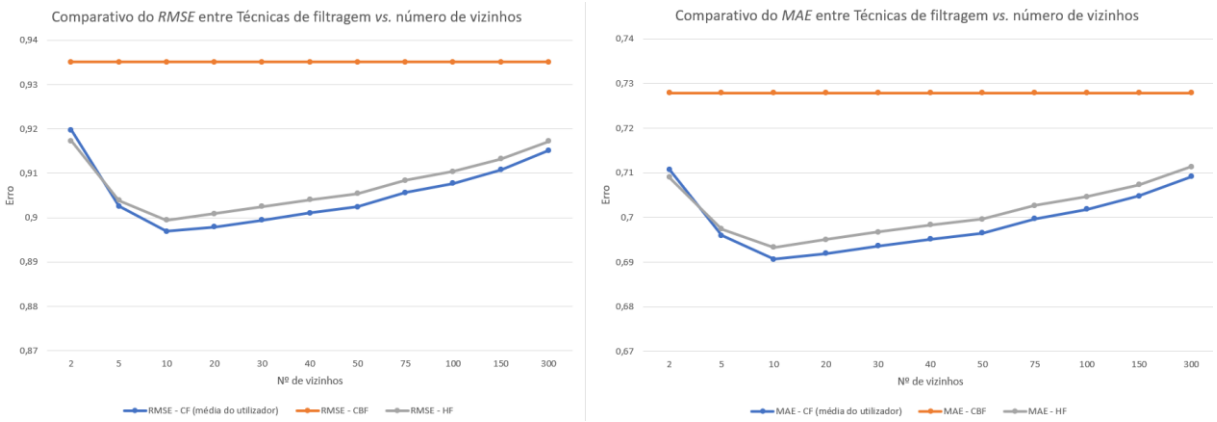

Figura 6-30: Comparação entre todas as técnicas implementadas

Os resultados obtidos, confirmam a qualidade das recomendações, tendo em atenção (Do et al., 2020). Nesse estudo, os erros *RMSE* para todas as técnicas ultrapassaram sempre 1.1, enquanto neste estudo, todos se situaram abaixo de 0.94.

7. Conclusão

Este trabalho constitui a dissertação do Mestrado em Sistemas e Tecnologias de Informação para as Organizações, do Departamento de Informática, da Escola Superior de Tecnologia e Gestão de Viseu, do Instituto Politécnico de Viseu, cujo tema é Sistemas de Recomendação para Conteúdos de Aplicações web. Os objetivos traçados, correspondem ao estudo do desenvolvimento de Sistemas de Recomendação com técnicas baseadas em conteúdo (CBF), colaborativas (CF) e na combinação das mesmas (HF). Foi criado um protótipo de um site web, com especial enfoque no *back-end*, de recomendação de filmes. Este permite ser um intermediário para informação disponibilizada *online* pelo *TMDb* e efetuar recomendações não personalizadas e personalizadas, para utilizadores registados. O SR utiliza os *MovieLens Datasets offline* de *ratings* disponibilizados, pelo *GroupLens*, como base para as recomendações. Estes mesmos *datasets*, permitiram a validação do protótipo de SR desenvolvido.

Começou-se por definir o problema a solucionar, em termos do desenvolvimento de SRs, nomeadamente o *cold start*, o *sparsity problema* a escalabilidade do sistema, dos utilizadores dos tipos *gray sheep*, *white sheep* e *black sheep*, sinonímia, o *long tail* e a vizinhança.

Também foi apresentado o estado da arte, nomeadamente com definições e conceitos no âmbito dos SR, tipos e trabalhos relacionados.

No seguimento, são descritas as metodologias de desenvolvimento a utilizar no SR a implementar, bem como as métricas de avaliação a utilizar para a validação do SR.

O desenvolvimento do sistema foi descrito. Foi feita a análise e conceção do sistema, a descrição da funcionalidade implementada do protótipo, bem como a explicação dos algoritmos de recomendação implementados.

Por fim foi feita uma avaliação experimental, com a descrição e análise dos resultados obtidos.

O protótipo implementado permite obter informação, estatísticas textuais e gráficas, assim como obter recomendações sobre filmes, a partir do *MovieLens Latest Datasets* pequeno, constituído por 100 836 *ratings*, aplicado a 9 742 filmes, por 610 utilizadores, atualizado em setembro de 2018.

As recomendações não personalizadas permitem filtrar filmes semelhantes em termos de conteúdo e apresentar o top de popularidade a partir dos *ratings* efetuados por outros utilizadores.

As recomendações personalizadas permitem filtrar filmes semelhantes em termos de conteúdo, tendo em atenção as classificações já efetuadas pelo utilizador ativo **e filtrar filmes de utilizadores semelhantes**, tendo por base as classificações atribuídas por estes, **bem como a combinação numa recomendação híbrida** das duas anteriores. **Estas recomendações contêm ainda as previsões de classificações a atribuir pelo utilizador ativo aos filmes que ainda não classificou**. **Estas recomendações foram validadas** a partir do *MovieLens Latest Datasets*, pela subdivisão destes, em *dataset* de treino/teste, aos quais foram aplicadas as métricas de precisão estatística que permitiram avaliar a precisão, comparando as classificações previstas diretamente com a classificação real do utilizador, no *dataset* de teste. As métricas utilizadas foram o MAE e o RMSE, para validar a consistência dos resultados.

Verificou-se que os *MovieLens Latest Datasets* utilizados, permitiram concluir as tendências já apresentadas em estudos semelhantes, mas devido à dimensão dos mesmos, podem não ter esclarecido completamente algumas condições de aplicação das técnicas. Concretamente, com base nos *datasets* utilizados, uma vez que, os utilizadores classificaram sempre, pelo menos 20 filmes, não foi identificado grande contributo do CBF para a recomendação híbrida, a não ser quando considerados 2 vizinhos.

Pelos testes realizados ao protótipo e resultados da validação, **as recomendações obtidas, revelaram-se interessantes do ponto de vista empírico e válidas**, com *RMSE* inferior a 0.94 e *MAE* inferior 0,73, em todas as técnicas implementadas.

Em termos de trabalho futuro, poder-se-á utilizar um *MovieLens Datasets* de maior dimensão, outros *datasets* similares (*e.g.*, do *Netflix*, *IMDb*) ou automatizar o processo de obtenção dos *datasets*, a partir do *site* do *GroupLens*, comparando os resultados com os já obtidos. Uma vez que só foram estudadas técnicas baseadas em memória, poder-se-á aplicar outras técnicas baseadas em modelo, incluindo, várias de ML.

Apesar de ter sido utilizado uns *datasets* de menor dimensão, foi constatada a complexidade computacional, principalmente na validação dos resultados, pelo que se poderá explorar outras técnicas de suporte *offline* de recomendações ou, eventualmente, *online* baseado em *Cloud Computing*.

REFERÊNCIAS

Adomavicius, G., & Tuzhilin, A. (2005). *Towards the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions*.

Aggarwal, C. C. (2016). Recommender Systems: The Textbook. In *Wirtschaftsinformatik*. Springer International Publishing. https://doi.org/10.1007/978-3-319-29659-3

Alam, M., & Fekpe, E. (2000). Application of Dimensionality Reduction in Recommender System - A Case Study. *Transportation Research Record*, *1625*, 173–183. https://doi.org/10.3141/1625-22

Amazon.com. (n.d.). *Amazon.com: Online Shopping for Electronics, Apparel, Computers, Books, DVDs & more*. Retrieved November 8, 2019, from https://www.amazon.com/

Anderson, C. (2004). *LONG TAIL Why the Future of Business Is Selling Less of More*.

Anderson, C. (2006). *Long_Tail_Chris_Anderson_Motamem_Org*. 238. http://dl.motamem.org/long_tail_chris_anderson_motamem_org.pdf

B.Thorat, P., M. Goudar, R., & Barve, S. (2015). Survey on Collaborative Filtering, Content-based Filtering and Hybrid Recommendation System. *International Journal of Computer Applications*, *110*(4), 31–36. https://doi.org/10.5120/19308-0760

Balabanovic, M. (1997). *Fab: Content-Based, Collaborative Recommendation*. https://doi.org/10.1145/245108.245124

Barragáns-Martínez, A. B., Costa-Montenegro, E., Burguillo, J. C., Rey-López, M., Mikic-Fonte, F. A., & Peleteiro, A. (2010). A hybrid content-based and item-based collaborative filtering approach to recommend TV programs enhanced with singular value decomposition. *Information Sciences*, *180*(22), 4290–4311. https://doi.org/10.1016/j.ins.2010.07.024

Berry, M. W., Drmač, Z., & Jessup, E. R. (1999). Matrices, vector spaces, and information retrieval. *SIAM Review*, *41*(2), 335–362. https://doi.org/10.1137/S0036144598347035

Bostandjiev, S., O'Donovan, J., & Höllerer, T. (2012). *TasteWeights*. 35. https://doi.org/10.1145/2365952.2365964

Bouraga, S., Jureta, I., Faulkner, S., & Herssens, C. (2014). Knowledge-based recommendation systems:A survey. *International Journal of Intelligent Information Technologies*, *10*(2), 1–19. https://doi.org/10.4018/ijiit.2014040101

Burke, R. (2002). *Hybrid Recommender Systems: Survey and Experiments 1*. http://www.google.com

Carvalho, P. K. R. M. L. de. (2018). *Os Sistemas de Recomendação e Big Data - IGTI Blog*. http://igti.com.br/blog/os-sistemas-de-recomendacao-e-big-data/

Cazella, S. C., Augusta, M., Nunes, S. N., & Reategui, E. B. (2010). *A Ciência da Opinião: Estado da arte em Sistemas de Recomendação*.

Chen, F., Deng, P., Wan, J., Zhang, D., Vasilakos, A. V, & Rong, X. (2015). *Review Article Data Mining for the Internet of Things: Literature Review and Challenges*. https://doi.org/10.1155/2015/431047

Chen, M., Mao, S., & Liu, Y. (2014). Big data: A survey. *Mobile Networks and Applications*, *19*(2), 171–209. https://doi.org/10.1007/s11036-013-0489-0

Christakou, C., Lefakis, L., Vrettos, S., & Stafylopatis, A. (2005). *A Movie Recommender System Based on Semi-supervised Clustering*. www.imdb.com

Christian, H., Agus, M. P., & Suhartono, D. (2016). Single Document Automatic Text Summarization using Term Frequency-Inverse Document Frequency (TF-IDF). *ComTech: Computer, Mathematics and Engineering Applications*, *7*(4), 285.

https://doi.org/10.21512/comtech.v7i4.3746
Cremonesi, P., Koren, Y., & Turrin, R. (2010). Performance of recommender algorithms on top-N recommendation tasks. *RecSys'10 - Proceedings of the 4th ACM Conference on Recommender Systems*, 39–46. https://doi.org/10.1145/1864708.1864721
De Campos, L. M., Fernández-Luna, J. M., Huete, J. F., & Rueda-Morales, M. A. (2010). Combining content-based and collaborative recommendations: A hybrid approach based on Bayesian networks. *International Journal of Approximate Reasoning*. https://doi.org/10.1016/j.ijar.2010.04.001
Do, H. Q., Le, T. H., & Yoon, B. (2020). Dynamic Weighted Hybrid Recommender Systems. *International Conference on Advanced Communication Technology, ICACT*, *2020*, 644–650. https://doi.org/10.23919/ICACT48636.2020.9061465
EBay. (n.d.). *Eletrônicos, Automóveis, Moda, Colecionáveis, Cupons e muito mais | eBay*. Retrieved November 8, 2019, from https://www.ebay.com/
Eksnd, tra M. D., Riedl, J. T., & Konstan, J. A. (2010). Collaborative filtering recommender systems. In *Foundations and Trends in Human-Computer Interaction* (Vol. 4, Issue 2, pp. 81–173). https://doi.org/10.1561/1100000009
Falk, K. (2019). *Practical Recommender Systems*. Manning Publications Co.
Fayyad, U., Piatetsky-shapiro, G., & Smyth, P. (1996). *From Data Mining to Knowledge Discovery in*. *17*(3), 37–54.
Ge, M., Delgado-Battenfeld, C., & Jannach, D. (2010). Beyond Accuracy: Evaluating Recommender Systems by Coverage and Serendipity. *The 16Th International Conference*, 257. https://doi.org/10.1145/1864708.1864761
Gemmell, J., Schimoler, T., Mobasher, B., & Burke, R. (2012). Resource recommendation in social annotation systems: A linear-weighted hybrid approach. *Journal of Computer and System Sciences*. https://doi.org/10.1016/j.jcss.2011.10.006
Gennari, J. H., Musen, M. A., Fergerson, R. W., Grosso, W. E., Crubézy, M., Eriksson, H., Noy, N. F., & Tu, S. W. (2003). *The Evolution of Protégé: An Environment for Knowledge-Based Systems Development*.
Ghazanfar, M. A., & Prugel-Bennett, A. (2010). A scalable, accurate hybrid recommender system. *3rd International Conference on Knowledge Discovery and Data Mining, WKDD 2010*, *2*, 94–98. https://doi.org/10.1109/WKDD.2010.117
Ghazanfar, M. A., & Prügel-Bennett, A. (2010). An improved switching hybrid recommender system using Naive Bayes classifier and collaborative filtering. *Proceedings of the International MultiConference of Engineers and Computer Scientists 2010, IMECS 2010*, 493–502.
Goldberg, D., Nichols, D., Oki, B., & Terry, D. (1992). *Using Collaborative Filtering to Weave an. Information Tapestry*.
GroupLens. (2020a). *MovieLens*. https://movielens.org/
GroupLens. (2020b). *MovieLens Datasets*. https://grouplens.org/datasets/movielens/
Gupta, J., & Gadge, J. (2015). Performance analysis of recommendation system based on collaborative filtering and demographics. *Proceedings - 2015 International Conference on Communication, Information and Computing Technology, ICCICT 2015*, 1–6. https://doi.org/10.1109/ICCICT.2015.7045675
Herlocker, J. L., Konstan, J. A., Terveen, L. G., & Riedl, J. T. (2004). *Evaluating Collaborative Filtering Recommender Systems*.
Hofmann, T. (2003). *Collaborative Filtering via Gaussian Probabilistic Latent Semantic Analysis*.
Huang, A. (2008). *Similarity Measures for Text Document Clustering*.
Huang, Z., Chen, H., & Zeng, D. (2004). Applying Associative Retrieval Techniques to

Alleviate the Sparsity Problem in Collaborative Filtering Associative Retrieval Techniques for the Sparsity Problem. In *ACM Transactions on Information Systems* (Vol. 22, Issue 1).

Iaquinta, L., De Gemmis, M., Lops, P., Semeraro, G., Filannino, M., & Molino, P. (2008). Introducing serendipity in a content-based recommender system. *Proceedings - 8th International Conference on Hybrid Intelligent Systems, HIS 2008*, 168–173. https://doi.org/10.1109/HIS.2008.25

IMDb Help Center - Ratings FAQ. (2021). https://help.imdb.com/article/imdb/track-movies-tv/ratings-faq/G67Y87TFYYP6TWAV#calculatetop

Insight, C. (2014). *Dean - Big Data and Data Mining - 2015*. 289.

Isinkaye, F. O., Folajimi, Y. O., & Ojokoh, B. A. (2015). Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal*, *16*(3), 261–273. https://doi.org/10.1016/j.eij.2015.06.005

Jiang, L., Cai, Z., Wang, D., & Jiang, S. (2007). Survey of improving K-nearest-neighbor for classification. *Proceedings - Fourth International Conference on Fuzzy Systems and Knowledge Discovery, FSKD 2007*, *1*, 679–683. https://doi.org/10.1109/FSKD.2007.552

Jin, R., & Si, L. (2004). A study of methods for normalizing user ratings in collaborative filtering. *Proceedings of Sheffield SIGIR - Twenty-Seventh Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 568–569. https://doi.org/10.1145/1008992.1009124

Karypis, G., Konstan, J., & Riedl, J. (2015). Incremental Singular Value Decomposition Algorithms for Highly Scalable Recommender Systems Dimensionality Reduction for. *Science*, *81*(4), 717–733.

Kim, H. N., Ji, A. T., Ha, I., & Jo, G. S. (2010). Collaborative filtering based on collaborative tagging for enhancing the quality of recommendation. *Electronic Commerce Research and Applications*, *9*(1), 73–83. https://doi.org/10.1016/j.elerap.2009.08.004

Kleinberg, J. M. (1999). *Authoritative Sources in a Hyperlinked Environment*. www.harvard.edu

Koren, Y. (2010). Factor in the neighbors: Scalable and accurate collaborative filtering. *ACM Transactions on Knowledge Discovery from Data*, *4*(1), 1–24. https://doi.org/10.1145/1644873.1644874

Lang, K. (1995). *NewsWeeder: Learning to Filter Netnews (To appear in ML 95)*.

Last.fm. (n.d.). *Last.fm | Ouça músicas, encontre canções e descubra artistas*. Retrieved November 7, 2019, from https://www.last.fm/pt/

Learning, S., Trees, D., Golden, R. M., & Sciences, B. (2001). *Identification , Characterization , and Modeling*.

Lee, H., & Lee, J. (2019). Scalable deep learning-based recommendation systems. *ICT Express*. https://doi.org/10.1016/j.icte.2018.05.003

Lee Herlocker, J. (2000). *Understanding and Improving Automated Collaborative Filtering Systems*.

Lee Rodgers, J., & Alan Nice Wander, W. (1988). Thirteen ways to look at the correlation coefficient. *American Statistician*, *42*(1), 59–66. https://doi.org/10.1080/00031305.1988.10475524

Leskovec, J., Rajaraman, A., & Ullman, J. D. (2014). *Mining of Massive Datasets*. Cambridge University Press. https://doi.org/10.1017/CBO9781139924801

Linden, G., Smith, B., & York, J. (2003). Amazon.com recommendations: Item-to-item collaborative filtering. *IEEE Internet Computing*, *7*(1), 76–80. https://doi.org/10.1109/MIC.2003.1167344

Luk, K. (2019). *Introduction to TWO approaches of Content-based Recommendation System*.

https://towardsdatascience.com/introduction-to-two-approaches-of-content-based-recommendation-system-fc797460c18c

Ma, H., Zhou, D., Liu, C., Lyu, M. R., & King, I. (2011). *Recommender Systems with Social Regularization*.

Marcelino, V. F. (2014). *Sistema de Recomendação - Filtragem Coaborativa*.

Mazeh, I., & Shmueli, E. (2019). A personal data store approach for recommender systems: enhancing privacy without sacrificing accuracy. *Expert Systems with Applications*. https://doi.org/10.1016/j.eswa.2019.112858

Mooney, R. J., & Roy, L. (2000). Content-based book recommending using learning for text categorization. *Proceedings of the ACM International Conference on Digital Libraries*, 195–204. https://doi.org/10.1145/336597.336662

Mukund, D., & George, K. (2004). Item-based top- N recommendation algorithms. *ACM Transactions on Information Systems*, *22*(1), 143–177. https://doi.org/1046-8188/04/0100-0143

Netflix Prize. (2009). https://www.netflixprize.com/

Ning, X., Desrosiers, C., & Karypis, G. (2015). A comprehensive survey of neighborhood-based recommendation methods. *Recommender Systems Handbook, Second Edition*, 37–76. https://doi.org/10.1007/978-1-4899-7637-6_2

NumFOCUS. (n.d.). *pandas - Python Data Analysis Library*. Retrieved November 25, 2020, from https://pandas.pydata.org/

NumPy. (2020). https://numpy.org/

Official Google Blog: Introducing the Knowledge Graph: things, not strings. (n.d.). Retrieved November 15, 2019, from https://googleblog.blogspot.com/2012/05/introducing-knowledge-graph-things-not.html

Oussous, A., Benjelloun, F.-Z., Ait Lahcen, A., & Belfkih, S. (2018). Big Data technologies: A survey. *Journal of King Saud University - Computer and Information Sciences*, *30*(4), 431–448. https://doi.org/10.1016/J.JKSUCI.2017.06.001

Palmisano, C., Tuzhilin, A., & Gorgoglione, M. (2008). Using context to improve predictive modeling of customers in personalization applications. *IEEE Transactions on Knowledge and Data Engineering*, *20*(11), 1535–1549. https://doi.org/10.1109/TKDE.2008.110

Papagelis, M., & Plexousakis, D. (2005). Qualitative analysis of user-based and item-based prediction algorithms for recommendation agents. *Engineering Applications of Artificial Intelligence*. https://doi.org/10.1016/j.engappai.2005.06.010

Pasquinelli, M. (2009). Google's PageRank Algorithm: A diagram of cognitive capitalism and the rentier of the common intellect. *Deep Search: The Politics of Search Beyond Google*, *May 2012*, 152–162.

Pathak, A., Mandava, C., & Patel, R. (2019). *Recommendation Systems : User-based Collaborative Filtering using N Nearest Neighbors*. Medium. https://medium.com/sfu-cspmp/recommendation-systems-user-based-collaborative-filtering-using-n-nearest-neighbors-bf7361dc24e0

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine Learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830. https://scikit-learn.org/stable/index.html

Pham, M. C., Cao, Y., Klamma, R., & Jarke, M. (2011). *A Clustering Approach for Collaborative Filtering Recommendation Using Social Network Analysis*. http://www10.epinions.com/

protégé - A free, open-source ontology editor and framework for building intelligent systems.

(2020). https://protege.stanford.edu/
Puglisi, S., Parra-Arnau, J., Forné, J., & Rebollo-Monedero, D. (2015). On content-based recommendation and user privacy in social-tagging systems. *Computer Standards and Interfaces*, *41*, 17–27. https://doi.org/10.1016/j.csi.2015.01.004
Ramos, J. G. (2010). *Algoritmos Colaborativos para Sistemas de Recomendação*.
Reis, L. F. M. dos. (2012). *Sistema de Recomendação Baseado em Conhecimento*. Faculdade de Ciências e Tecnologia Universidade de Coimbra.
Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994). *GroupLens: An Open Architecture for Collaborative Filtering of Netnews*.
Resnick, P., & Varian, H. R. (1997). Recommender Systems. In *COMMUNICATIONS OF THE ACM* (Vol. 40, Issue 3).
RevistaUsenet.com. (2019). https://revistausenet.com/
Ricci, F., Rokach, L., Shapira, B., Kantor, P. B., & Ricci, F. (2011). Recommender Systems Handbook. In *Recommender Systems Handbook*. https://doi.org/10.1007/978-0-387-85820-3
Salton, G., & Buckley, C. (1988). Term-weighting approaches in automatic text retrieval. *Information Processing & Management*, *24*(5), 513–523. https://doi.org/10.1016/0306-4573(88)90021-0
Sarwar, B., Karypis, G., Konstan, J., & Riedl, J. (2001). Item-Based Collaborative Filtering Recommendation Algorithms. *GroupLens Research Group/Army HPC Research Center Department of Computer Science and Engineering*, 286–295.
Schafer, J. Ben, Konstan, J., & Riedl, J. (1999). *Recommender Systems in E-Commerce*. www.reel.com
Schwab, I., Kobsa, A., & Koychev, I. (2004). Learning User Interests through Positive Examples Using Content Analysis and Collaborative Filtering. *User Modeling and User-Adapted Interaction - UMUAI*, *14*(5), 469–475.
scikit-learn developers. (2020a). *sklearn.feature_extraction.text.TfidfVectorizer*. https://scikit-learn.org/stable/modules/generated/sklearn.feature_extraction.text.TfidfVectorizer.html#sklearn.feature_extraction.text.TfidfVectorizer
scikit-learn developers. (2020b). *ssklearn.model_selection.train_test_split*. https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html
SeatGeek. (2011). *FuzzyWuzzy: Fuzzy String Matching in Python*. https://chairnerd.seatgeek.com/fuzzywuzzy-fuzzy-string-matching-in-python/
Shani, G., & Gunawardana, A. (2011). Evaluating Recommendation Systems. *Recommender Systems Handbook*, 257–297. https://doi.org/10.1007/978-0-387-85820-3_8
Sharifi, Z., Rezghi, M., & Nasiri, M. (2013). New algorithm for recommender systems based on singular value decomposition method. *Proceedings of the 3rd International Conference on Computer and Knowledge Engineering, ICCKE 2013*, *Iccke*, 86–91. https://doi.org/10.1109/ICCKE.2013.6682799
Sharma, A. (2020). *Beginner Tutorial: Recommender Systems in Python*. https://www.datacamp.com/community/tutorials/recommender-systems-python
Sobre a Netflix. (n.d.). Retrieved November 12, 2019, from https://media.netflix.com/pt_pt/about-netflix
Su, X., & Khoshgoftaar, T. M. (2009). A Survey of Collaborative Filtering Techniques. *Advances in Artificial Intelligence*. https://doi.org/10.1155/2009/421425
Takács, G., Pilászy, I., & Németh, B. (2009). Scalable Collaborative Filtering Approaches for Large Recommender Systems. In *Journal of Machine Learning Research* (Vol. 10). http://sifter.org/
Terveen, L., Hill, W., Amento, B., McDonald, D., & Creter, J. (1997). A System for Sharing

Recommendations. *Communications of the ACM*, *40*(3), 59–62. https://doi.org/10.1145/245108.245122

The Matplotlib development Team. (2021). *Matplotlib: Visualization with Python*. https://matplotlib.org/

TheFork - Reserve nos melhores restaurantes da Europa. (n.d.). Retrieved November 7, 2019, from https://www.thefork.pt/

TripAdvisor. (2020). *TripAdvisor*. https://www.tripadvisor.pt/?fid=dcbfa773-47e7-47e9-9f5b-96dac5f7ad07

Ying, R., He, R., Chen, K., Eksombatchai, P., Hamilton, W. L., & Leskovec, J. (2018). *Graph Convolutional Neural Networks for Web-Scale Recommender Systems*. *11*. https://doi.org/10.1145/3219819.3219890

Yu, S. J. (2012). The dynamic competitive recommendation algorithm in social network services. *Information Sciences*, *187*(1), 1–14. https://doi.org/10.1016/j.ins.2011.10.020

Zanker, M. (2008). A collaborative constraint-based meta-level recommender*. *RecSys'08: Proceedings of the 2008 ACM Conference on Recommender Systems*, 139–145. https://doi.org/10.1145/1454008.1454032

Zuva, T., Ojo, S. O., Ngwira, S. M., & Zuva, K. (2012). A Survey of Recommender Systems Techniques , Challenges and Evaluation Metrics. *International Journal of Emerging Technology and Advanced Engineering*, *2*(11), 382–386.