TÉCNICO LISBOA

Motor de Recomendação de Eventos

Jorge Miguel Lopes Branco de Oliveira

Dissertação para obtenção do Grau de Mestre em

Engenharia de Telecomunicações e Informática

Orientador(es): Prof. Pável Pereira Calado
Prof. Bruno Emanuel da Graça Martins

Júri

Presidente: Prof. Luís Manuel Antunes Veiga
Orientador: Prof. Pável Pereira Calado
Vogal: Prof. João Magalhães

Novembro 2016

À minha família

Agradecimentos

Quero agradecer aos professores Pável Pereira Calado e Bruno Emanuel da Graça Martins, investigadores do grupo de Sistemas de Informação e Suporte à Decisão (IDSS) do INESC-ID e orientador e co-orientador, respectivamente, por todo o apoio e ajuda prestadas durante a elaboração desta dissertação. Quero também agradecer à minha família por todo o apoio que me deram ao longo da elaboração desta dissertação.

Resumo

Actualmente, com o crescimento da quantidade de informação referente a eventos existentes na Internet, tornou-se cada vez mais difícil para os utilizadores encontrar aqueles que melhor encaixem nas suas preferências e gostos pessoais. Nesse sentido, os sistemas de recomendação vieram ajudar os utilizadores nessa tarefa, ao reduzir a sobrecarga de informação sobre estes através da recomendação de eventos que possam ser da sua preferência. Contudo, diferentemente do problema de recomendação clássico, que envolve itens como filmes ou livros, os eventos padecem do denominado *new item cold-start problem*. Uma vez que os eventos têm quase sempre lugar no futuro, existe uma ausência de *feedback* por parte dos utilizadores, bem como uma falta de registo de participação destes nos eventos. Nestas situações, é imperativo considerar não só informação acerca dos eventos e dos utilizadores, como também informação contextual. Tendo em conta o problema apresentado, foi implementado um motor de recomendação de eventos, o qual tem, por objetivo fundamental, prever os eventos em que os utilizadores possam eventualmente estar interessados, tendo como base eventos em que os mesmos participaram no passado, informação demográfica acerca dos utilizadores, e quais os eventos que os utilizadores viram e com os quais interagiram no passado, por exemplo através de cliques numa aplicação. Por forma a obter o melhor desempenho de recomendação possível, foram criadas várias *features*, sendo a recomendação de eventos feita com recurso a um classificador *Random Forest*. Testes efectuados permitiram atestar a boa eficácia e desempenho de recomendação da solução implementada.

Palavras-chave: Sistemas de Recomendação, Recomendação de Eventos, Contexto, Personalização.

Abstract

Nowadays, with the growing ammount of information related to existing events on the Internet, it has become increasingly difficult for users to find those that best fit their preferences and personal tastes. In this regard, recommender systems came to help users in this task by reducing the overload of information felt by the users, by recommending events that users may like. However, differently from the classical recommender problem, which envolves items such as movies or books, events suffer from the so called *new item cold-start problem*. Since events often take place in the future, there's an absence of *feedback* from the users, as well as a lack of records about the user's attendance in the events. In such situations, it's necessary to consider, not only information about the events and the users, but also contextual information. Having in mind the presented problem, it was implemented an event recommendation engine, which has, for primary objective, to predict the events in which the users may be, eventualy, interested. The recommendation is made by considering information regarding previously attended events by the users, demographic information about them, and what events they've seen and interacted with, e.g., through clicks in an application. In order to get the best possible recommendation performance, several *features* were created, being the event recommendation made with a *Random Forest* classifier. Several tests carried out have allowed to certify the good efficacy and performance of the recommender solution developed.

Keywords: Recommender Systems, Event Recommender Systems, Context, Personalization.

Conteúdo

Lista de Tabelas

Lista de Figuras

Capítulo 1

Introdução

1.1 Motivação

Os sistemas de recomendação configuram-se como uma ferramenta de particular importância no panorama atual da Internet, tanto para os utilizadores, como para as empresas que os aplicam. Com efeito, todos os dias são lançadas na Internet quantidades colossais de informação, tornando difícil para os utilizadores encontrar items que melhor se encaixem nas suas preferências e gostos pessoais. Nesse sentido, os sistemas de recomendação auxiliam os utilizadores nessa tarefa, ao reduzir a sobrecarga de informação que sobre estes paira, recomendando-lhes items que estes possam, com grande probabilidade, vir a gostar no futuro, e.g., livros e filmes. Por outro lado, ao recomendar items de possível interesse para os utilizadores, estes sistemas configuram-se também como uma fonte de lucro para as empresas de comércio que deles dispõem, oferecendo, além disso, um serviço adicional e personalizado aos utilizadores que permite conquistar a confiança e a fidelização destes [1].

Contudo, a tarefa de recomendação não se aplica apenas e só a items como filmes e livros. Com efeito, enormes quantidades de informação acerca dos mais variados tipos de eventos, e.g., concertos, festivais de música, palestras de divulgação científica, etc., são também lançadas diariamente na Internet. A sobrecarga de informação daí decorrente torna difícil a tarefa de encontrar eventos que se encaixem nas preferências dos utilizadores. Nesse sentido, o uso de sistemas de recomendação fornece uma preciosa ajuda, recomendando aos utilizadores eventos que possam ser da sua preferência.

A tarefa de recomendar eventos é, no entanto, um problema diferente da tarefa de recomendação clássica. Nos problemas de recomendação clássicos, os items objeto de recomendação foram já previamente consumidos e avaliados por muitos utilizadores, com exceção dos items recentemente adicionados ao sistema, os quais possuem insuficiente informação a esse respeito. Este problema é denominado de *new item cold-start problem*. Já nos problemas de recomendação de eventos, os items a ser recomendados (i.e., eventos, também denominados de *one-and-only items*) têm, tipicamente, um período de vida curto e têm sempre lugar no futuro, por definição [2]. Como consequência, os utilizadores não podem participar nos eventos nem avaliar os mesmos antes destes ocorrerem. Deste modo, a recomendação de eventos tem de lidar constantemente com o *new item cold-start problem*, tornando

a tarefa mais complicada que a tarefa de recomendação clássica. Não obstante, os utilizadores podem expressar a sua intenção de participar ou não em eventos, proporcionando desta forma informação útil para mitigar o *new item cold-start problem* [2].

Por forma a proporcionar aos utilizadores recomendações precisas, os sistemas de recomendação de eventos têm em consideração um leque de informações alargado, que compreende informação acerca dos utilizadores (e.g., data de nascimento, sexo, localização), dos eventos (e.g., tipo de evento, descrição do evento, localização do evento), e informação contextual (e.g., distância entre um utilizador e um evento, co-participação de utilizadores em eventos, as preferências temporais de um utilizador em termos de frequência dos eventos, etc.).

1.2 Proposta de Dissertação

A presente proposta de dissertação tem, por objetivo, a implementação de um sistema de recomendação que possa prever os eventos em que os utilizadores possam eventualmente estar interessados, tendo como base eventos a que os mesmos responderam no passado, informação demográfica acerca dos utilizadores, e quais os eventos que os utilizadores viram e com os quais interagiram no passado, através, por exemplo, de cliques numa aplicação.

Para cada utilizador, o sistema deve de retornar uma lista de eventos. Estes devem estar ordenados por relevância, desde os eventos em que o sistema preveja que o utilizador possa estar mais interessado ou mostrar um maior interesse, até àqueles em que o utilizador possa demonstrar ter menos interesse.

1.3 Contributos

No decurso da realização da presente dissertação, foi implementado um motor de recomendação de eventos, o qual faz uso de informação acerca dos utilizadores, informação acerca dos eventos, e informação contextual. Foram também realizados vários testes, por forma a aferir a eficácia e desempenho de recomendação da solução de recomendação implementada. Os contributos mais importantes da dissertação a que este relatório se refere são sumarizados em seguida:

1. Foi implementado um motor de recomendação de eventos, tal como descrito anteriormente, o qual faz uso de um total de 45 *features*. Este motor de recomendação de eventos faz uso de um classificador *Random Forest*, por forma a gerar a lista de eventos a recomendar, para cada utilizador considerado. Para este motor de recomendação de eventos, foi obtido um valor de precisão média superior ao estado da arte;

2. Após a implementação e teste do motor de recomendação de eventos descrito anteriormente, foi executado um procedimento de selecção de *features* baseado no algoritmo SFS (*Sequencial Forward Selection*). Com base na execução deste procedimento, foi obtido um subconjunto de 15 *features*, a partir do conjunto original de 45 *features*. Estas foram então aplicadas no motor

de recomendação de eventos, por forma a gerar as listas de eventos recomendados para cada utilizador. Desta forma, foi obtida uma melhoria de 4.16% na precisão média.

A implementação do presente motor de recomendação de eventos permitiu testar:

- A importância do uso de informação contextual no processo de recomendação de eventos, além do uso de informação relacionada com os utilizadores e os eventos;
- A importância do uso de um processo de selecção de *features*, por forma a (1) reduzir a dimensionalidade do problema de recomendação, e (2) aumentar a eficácia e desempenho de recomendação da solução desenvolvida.

1.4 Organização do Relatório

O restante conteúdo deste relatório encontra-se organizado da seguinte forma: No Capítulo 2 são apresentados conceitos fundamentais e discutidos trabalhos relacionados na área dos sistemas de recomendação de eventos. No Capítulo 3 é descrita a solução de recomendação de eventos implementada, em particular: quais as *features* usadas pelo sistema de recomendação de eventos, o modelo de classificação usado, a arquitectura do sistema e o processo de selecção de *features* usado, por forma a seleccionar um subconjunto de features relevantes a serem usadas pelo sistema de recomendação de eventos. No Capítulo 4 são descritos o *dataset* utilizado e a métrica de avaliação considerada. São também apresentados e discutidos os resultados obtidos com recurso ao sistema de recomendação de eventos implementado. Por fim, no Capítulo 5 são apresentadas as principais conclusões, bem como o trabalho futuro a desenvolver.

Capítulo 2

Conceitos e Trabalhos Relacionados

O presente capítulo tem, por objetivo, dar a conhecer os conceitos mais importantes na área dos sistemas de recomendação, aplicáveis na tarefa de recomendação de eventos. Ao mesmo tempo, são dados a conhecer um conjunto de trabalhos relacionados com o tema em questão.

2.1 Conceitos

São apresentados em seguida alguns conceitos importantes no contexto dos sistemas de recomendação.

Filtragem Colaborativa (*Collaborative Filtering*)

Filtragem Colaborativa (FC) é um método de filtragem em que as recomendações para cada utilizador são feitas com base na informação (e.g., classificações atribuídas a um dado conjunto de items, tais como filmes e livros) providenciada por outros utilizadores que apresentem uma elevada similaridade com o utilizador-alvo de recomendação [3]. De uma forma informal, a ideia-chave é a de que, se dois utilizadores tiveram preferências semelhantes no passado, irão ter preferências semelhantes no futuro [1]. Podemos distinguir entre dois tipos principais de filtragem colaborativa [4]:

- **User-based Collaborative Filtering**, em que a predição da classificação de um dado item por um dado utilizador é feita através da agregação das classificações atribuídas a esse mesmo item por parte de utilizadores similares ao utilizador em questão;
- **Item-based Collaborative Filtering**, em que os items são recomendados com base na informação acerca de outros items que o utilizador previamente classificou. Neste tipo de filtragem colaborativa, os items recomendados a um dado utilizador são classificados agregando as similaridades entre cada item candidato e os items que o utilizador classificou.

Em FC, dispomos de uma matriz de utilizadores versus items, em que, para cada utilizador, dispomos da classificação por este atribuída a um dado conjunto de items. Com base nesta matriz, podemos, por exemplo, determinar a similaridade entre dois utilizadores recorrendo à Correlação de Pearson:

$$\mathrm{sim}(a,b) = \frac{\sum_{p \in P}(r_{a,p} - \overline{r}_a)(r_{b,p} - \overline{r}_b)}{\sqrt{\sum_{p \in P}(r_{a,p} - \overline{r}_a)^2}\sqrt{\sum_{p \in P}(r_{b,p} - \overline{r}_b)^2}} \tag{2.1}$$

Na Equação 2.1, a e b são utilizadores, $r_{a,p}$ é a classificação do utilizador a para o item p, P é o conjunto dos items classificados por a e b, e $\overline{r_a}$ e $\overline{r_b}$ correspondem às médias das classificações dos utilizadores a e b.

A predição da classificação que um utilizador daria a um item ainda não classificado por este é dada por:

$$\mathrm{Pred}(a,p) = \overline{r_a} + \frac{\sum_{b \in N} \mathrm{sim}(a,b)(r_{b,p} - \overline{r}_b)}{\sum_{b \in N} \mathrm{sim}(a,b)} \tag{2.2}$$

O algoritmo mais usado neste tipo de abordagem baseia-se na pesquisa pelos *k Nearest Neighbors* (*kNN*, ou k Vizinhos mais Próximos) [3].

Filtragem Baseada no Conteúdo (*Content-Based Filtering*)

Filtragem Baseada no Conteúdo (FBC) é um método de filtragem em que as recomendações para cada utilizador são feitas tendo como base a similaridade entre items que um utilizador consumiu e classificou no passado, de forma positiva, com novos items ainda não vistos pelo utilizador [3]. De uma maneira informal, a ideia-chave deste método é recomendar novos items a um dado utilizador com base nas suas escolhas passadas (e.g., se um dado utilizador leu e gostou de romances policiais no passado, o sistema tentará recomendar ao utilizador outros romances policiais ainda não lidos por este).

Nesta abordagem, o conteúdo dos items é analisado (e.g., descrição dos items, palavras-chave, género, etc.), sendo o resultado dessa análise utilizado no estabelecimento de similaridades entre items [3]. Uma forma simples de determinar a similaridade entre um item ainda não visto pelo utilizador e o perfil deste é obtida através do uso do Coeficiente de Dice [1], em que o cálculo da similaridade é feito com base na sobreposição de palavras-chave:

$$\mathrm{sim}(b_i, b_j) = \frac{2 * \mid \text{palavras-chave}(b_i) \cap \text{palavras-chave}(b_j) \mid}{\mid \text{palavras-chave}(b_i) \mid + \mid \text{palavras-chave}(b_j) \mid} \tag{2.3}$$

Contudo, a representação em palavras-chave simples apresenta alguns problemas, tais como o facto de nem todas as palavras extraídas do conteúdo dos items terem igual importância. Assim, a medida padrão usada neste tipo de metodologia de recomendação é a medida TF-IDF (*Term Frequency-Inverse Document Frequency*), em que os documentos (conteúdos dos items) são codificados como um vetor de termos pesados:

$$\mathrm{TF}(i,j) = \frac{\mathrm{freq}(i,j)}{\mathrm{maxOthers}(i,j)} \tag{2.4}$$

$$\mathrm{IDF}(i) = \log\left(\frac{N}{n(i)}\right) \tag{2.5}$$

$$\text{TF-IDF}(i,j) = \text{TF}(i,j) \times \text{IDF}(i) \tag{2.6}$$

Classificação Supervisionada

Classificação supervisionada consiste num tipo de aprendizagem automática (*Machine Learning*) no qual o objectivo principal assenta na construção de um modelo conciso de distribuição de classes (*labels*), através do uso de um conjunto de características (ou *features*) [5]. Por forma a gerar um modelo de classificação supervisionada, são fornecidos um conjunto de exemplos de treino, dos quais são conhecidos os valores das suas *features*, bem como as classes de cada exemplo de treino considerado. O modelo de classificação supervisionada resultante é depois aplicado a um conjunto de exemplos de teste, dos quais apenas são conhecidos os valores das suas *features*, sendo a classe de cada exemplo de teste desconhecida. O modelo de classificação supervisionada, gerado com base nos dados de treino fornecidos anteriormente, deverá ser capaz de prever a classe para cada exemplo de teste fornecido. A Figura 2.1 [5] descreve, de uma forma simplificada, o processo de classificação supervisionada.

Como nota adicional, o motor de recomendação de eventos implementado no âmbito da presente dissertação faz uso de um classificador supervisionado (classificador *Random Forest* - ver Capítulo 3, Secção 3.2), por forma a gerar as listas de eventos recomendados para cada utilizador considerado.

2.2 Trabalhos Relacionados

Existem na literatura vários trabalhos cujo foco incide sobre o problema de recomendação de eventos. A maior parte destes trabalhos tem por objetivo a proposta de soluções precisas e eficientes para solucionar este problema, nas mais diversas situações, e.g. em *Event-based Social Networks* [2], ou em contexto de eventos académicos [6], entre outros. Contudo, também encontramos na literatura artigos cujo foco se centra na avaliação de vários algoritmos de recomendação genéricos que podem ser empregues na tarefa de recomendação de eventos [7, 8].

Os trabalhos apresentados e discutidos na presente dissertação dividem-se em 3 grandes grupos:

1. Abordagens simples ao problema da recomendação de eventos, ou seja, soluções para o problema de recomendação de eventos que façam uso de abordagens baseadas numa única técnica de recomendação (e.g., abordagens de recomendação que façam uso apenas de filtragem colaborativa);

2. Abordagens híbridas como forma de abordar a recomendação de eventos, ou seja, soluções que consistam na combinação de vários dados de entrada e/ou várias abordagens de recomendação individuais, e.g. FC e FBC;

3. Comparação de abordagens para a recomendação de eventos.

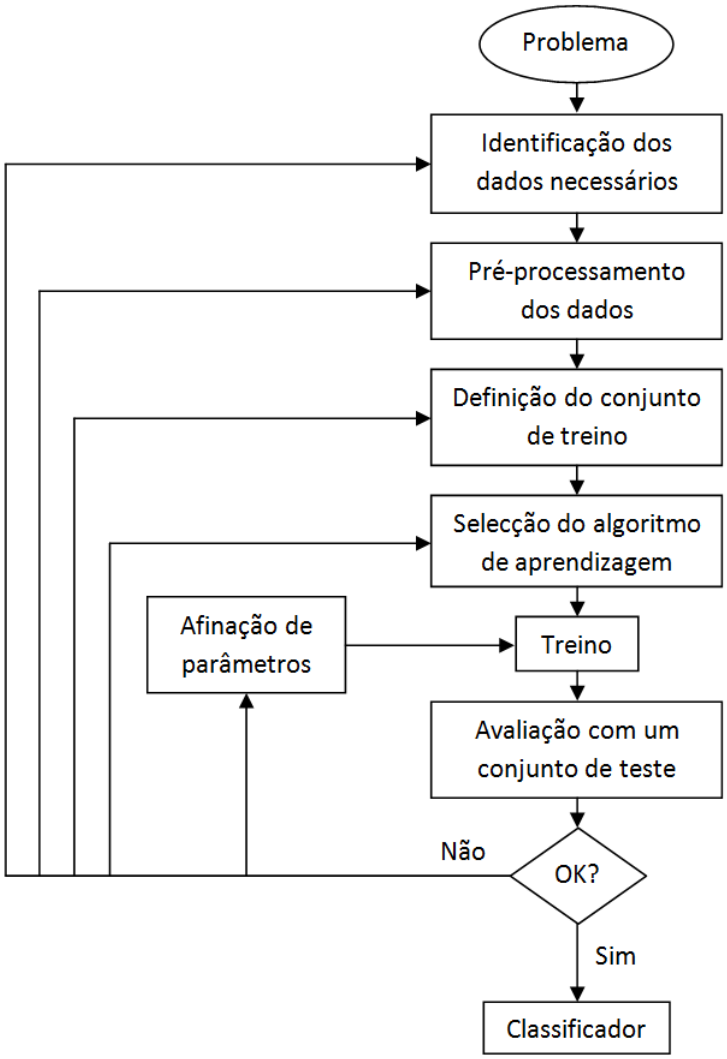

Figura 2.1: Diagrama ilustrativo do processo de classificação supervisionada [5].

Existem na literatura, além disso, vários trabalhos destinados a fornecer aos leitores uma pespetiva geral dos sistemas de recomendação [1, 4, 3]. O presente texto irá incidir apenas no problema de recomendação de eventos.

Abordagens Simples ao Problema da Recomendação de Eventos

Existem na literatura soluções para o problema de recomendação de eventos que fazem uso de abordagens baseadas numa única técnica de recomendação (e.g., abordagens de recomendação que façam uso apenas de filtragem colaborativa).

No trabalho de [9] é abordado o problema da implementação de um sistema de recomendação de eventos em tempo real, capaz de agregar grandes quantidades de *online media* a partir de canais heterogéneos, sumarizar estes em eventos, descobrir associações significativas através da ligação entre eventos, e de gerar uma *sequence map* dos eventos que providencie uma imagem de como os eventos interagem com outros eventos ao longo do tempo. Por forma a solucionar o problema

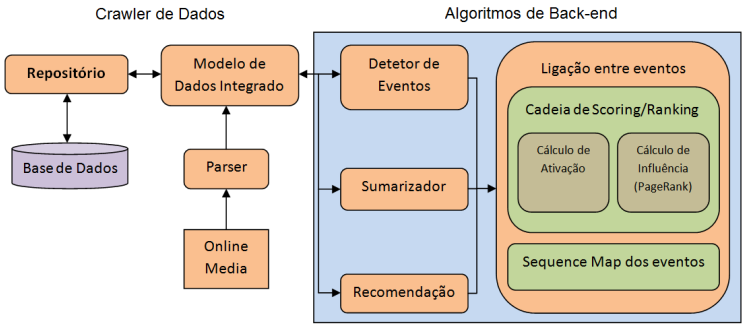

Figura 2.2: A arquitetura do sistema EVENTERA ([9]).

apresentado, foi desenvolvido o EVENTERA[1], um sistema de recomendação de média passivo, mas personalizado, capaz de lidar com os problemas resultantes de atividades de *media browsing* intensas, tais como a publicação de grandes quantidades de notícias, tweets e outras mensagens em redes sociais, vídeos, etc.

A arquitetura do sistema EVENTERA é a apresentada na Figura 2.2. A recomendação de eventos é implementada no bloco de recomendação, onde um algoritmo de filtragem colaborativa é aplicado sobre a lista de palavras-chave dos eventos, por forma a gerar recomendações. O funcionamento deste algoritmo é análogo ao apresentado na Secção 2.1 do corrente capítulo. O sistema consegue ultrapassar o *new user cold-start problem*, recomendando eventos globalmente famosos a novos utilizadores, até que tenham sido gerados os historiais de participação destes na base de dados do sistema. Para os demais utilizadores, são recomendados eventos relevantes, desde que os utilizadores possuam informação de perfil contendo palavras-chave dos eventos.

Por forma a testar o sistema desenvolvido, foi usado um cenário em que um evento do mundo real, denominado *"Psy won top vídeo of the year"* ocorre, e o sistema EVENTERA consegue detetar este mesmo evento se a frequência do mesmo aumentar, de forma rápida, ao longo de uma certa janela temporal, recomendando o evento a um utilizador que se encontre interessado no mesmo. O sistema conseguiu, além disso, criar uma *sequence map* dos eventos ocorridos no caso da experiência efetuada, a qual apresenta os vários eventos ordenados pela ordem em que surgiram nos diversos *media channels*, os diferentes *media channels* em que os mesmos surgiram pela primeira vez, e a relação entre os vários tipos de eventos, através da aglomeração de sub-eventos em grupos.

Com a execução da experiência mencionada, pôde-se também construir um *interaction map of media*, a qual mostrou ao utilizador que, no caso do evento *Psy*, a maior interação entre os diferentes tipos de *media channels* ocorreu entre canais de notícias e canais de *social media*. Assim, fica claro que ainda que sendo distintos, vários eventos podem apresentar traços em comum entre si. Esta característica permite a aglomeração de vários sub-eventos em famílias de eventos, permitindo aos utilizadores perceber qual a relação entre diversos eventos. Contudo, o sistema não toma em consideração informação contextual importante, tanto acerca dos utilizadores, como acerca dos eventos, e.g., distâncias geográficas, co-participação, etc. Assim, uma grande quantidade de informação útil

[1] http://www.cs.cmu.edu/ dongyeok/project/Eventera/

para o processo de recomendação perde-se, podendo levar o sistema a ter um desempenho inferior.

Abordagens Híbridas como Forma de Abordar a Recomendação de Eventos

As abordagens híbridas são, de longe, as mais comummente discutidas na literatura. Uma abordagem híbrida consiste na combinação de vários dados de entrada e/ou várias abordagens de recomendação individuais, e.g. FC e FBC. A abordagem de recomendação daí resultante permite conjugar as vantagens das várias abordagens de recomendação usadas, e possui a vantagem de fazer uso de diferentes dados de entrada. Em conjunto, estes fatores permitem às abordagens híbridas obter melhores resultados que aqueles obtidos com recurso a abordagens de recomendação individuais.

Hibridização Monolítica

A hibridização monolítica consiste no uso de uma única componente de recomendação, a qual combina as diferentes características e estratégias de recomendação das várias abordagens empregues.

O trabalho de [6] aborda o problema de recomendação de eventos, tomando o caso particular de conferências científicas. Por forma a solucionar o problema apresentado, os autores apresentam uma abordagem do tipo *content-based*, por forma a recomendar eventos a utilizadores, tendo como base as escolhas anteriores destes por eventos passados e as descrições dos eventos, i.e., as suas características (*features*). Simultaneamente, os autores propõem uma extensão colaborativa, designada LowRank, através da decomposição dos parâmetros dos utilizadores em componentes partilhadas e individuais.

Cada evento e_j é representado como um vetor de *m features*, x_j. Seguidamente, foram criados dois esquemas de representação de eventos. O primeiro esquema (ou esquema de base) corresponde a uma contagem pesada de palavras TF-IDF, em que cada coordenada do vetor de *features* x_j corresponde à ocorrência de uma palavra. O segundo esquema (alternativo) corresponde à identificação de tópicos a partir dos anúncios dos seminários, inferindo a distribuição de tópicos através de métodos como o *Latent Dirichlet Allocation* [10]. Nesta situação, cada coordenada do vetor de *features* x_j corresponde à frequência com que um dado tópico foi usado na geração do anúncio de um seminário.

Na abordagem *content-based* apresentada pelos autores, são consideradas funções de classificação lineares, onde cada vetor de características dos eventos x_j é mapeado para uma pontuação $\theta \,.\, x_j$. O objetivo é o de encontrar parâmetros θ tais que a função de classificação capture, da melhor forma, o *feedback* do utilizador acerca de eventos passados, para cada utilizador u, de acordo com uma formulação RANKSVM [11]:

$$\begin{aligned} &\text{minimizar} \quad \frac{1}{2} \parallel \theta_u \parallel^2 + C \sum_{jk} \xi_{jk} \\ &\text{sujeito a} \quad \theta_u \,.\, x_j \geq \theta_u \,.\, x_k + 1 - \xi_{jk} \,, \forall j,k : e_j \in \mathrm{E_P^+}(T_i,u) \wedge e_k \in \mathrm{E_P^-}(T_i,u) \end{aligned}$$

considerando uma janela temporal T_i. $\mathrm{E_P^+}(T_i,u)$ e $\mathrm{E_P^-}(T_i,u)$ denotam, respetivamente, o subconjunto de eventos que o utilizador u gostou ou nos quais participou, e o subconjunto de eventos que o utilizador

u não gostou. e_j e e_k são eventos.

Na extensão colaborativa apresentada pelos autores (LOWRANK), o problema de estimação pode ser definido de forma semelhante à da formulação RANKSVM. A diferença encontra-se agora no facto de que os parâmetros de transformação U e V são estimados ao longo dos utilizadores.

$$\text{minimizar} \quad \frac{1}{2} \parallel U \parallel_F^2 + \frac{1}{2} \parallel V \parallel_F^2 + C \sum_{ujk} \xi_{ujk}$$

$$\text{sujeito a} \quad [UVx_j]_u \geq [UVx_k]_u + 1 - \xi_{ujk}, \forall u, j, k : e_j \in \mathrm{E}_\mathrm{P}^+(T_i, u) \wedge e_k \in \mathrm{E}_\mathrm{P}^-(T_i, u)$$

considerando uma janela temporal T_i. V é uma matriz de parâmetros, de dimensão $k \times m$, usada para mapear as descrições dos eventos, x_j, num sub-espaço de dimensão k, ${x_j}' = Vx_j$, com $k \ll m$. U é uma matriz de parâmetros, de dimensão $N \times k$, correspondente aos parâmetros dos utilizadores, ${\theta_u}'$, com $u = 1, 2, ..., N$.

Por fim, foi conduzida uma experiência, por forma a testar e comparar o método LOWRANK com o método RANKSVM. Nesta experiência, foram recolhidas as preferências dos utilizadores, em termos de conferências científicas. No estudo efetuado, foi criado um cenário realista, que consistiu em anunciar seminários científicos em várias instituições, semanalmente, através de e-mail. Aos participantes, foi pedido que escolhessem em qual dos seminários científicos estariam interessados, tendo sido obtidas predições dos eventos que cada utilizador poderia vir a gostar no futuro, através de uma lista ordenada.

Comparando o desempenho das duas soluções testadas, para cada um dos esquemas de representação de eventos acima, pôde concluir-se que: (1) A abordagem RANKSVM obteve resultados comparáveis para ambos os esquemas de representação de eventos, e (2) a abordagem LOWRANK obteve melhores resultados que a abordagem RANKSVM, usando a representação de eventos TF-IDF, e resultados comparáveis à abordagem RANKSVM, usando a representação de eventos LDA. Contudo, apesar de demonstrar ter um bom desempenho quando comparada com uma solução puramente *content-based*, a solução desenvolvida coloca de parte informação contextual como, por exemplo, informação respeitante às localizações de utilizadores e eventos, entre outras. Além disso, a abordagem desenvolvida requer dos utilizadores o fornecimento de *feedback* explícito.

Já o trabalho de [2] aborda o problema da recomendação de eventos em *event-based social networks* (EBSNs). Em particular, dado um utilizador e um conjunto de sinais contextuais, os autores procuram determinar quais os eventos com maior probabilidade de serem do agrado desse mesmo utilizador. Assim, foi proposta uma abordagem *context-aware* para recomendação de eventos, a qual, além dos conjuntos de utilizadores e eventos, explora vários sinais contextuais, tais como as preferências temporais do utilizador em termos de frequência dos eventos, o conjunto de grupos a que os utilizadores se podem juntar, as preferências dos utilizadores em termos de distâncias geográficas, e o conteúdo textual dos eventos. Foram seguidamente desenvolvidos modelos de recomendação contextuais adaptados a cada um dos sinais contextuais acima. Por fim, foi desenvolvida uma abordagem híbrida, do tipo *Learning to Rank*, a qual recebe, como *input features*, cada um dos modelos de recomendação *context-aware* desenvolvidos.

No que ao aspeto social diz respeito, foram desenvolvidos dois modelos. O primeiro é um modelo de

frequência de grupo, sendo o segundo um modelo multi-relacional. No modelo de frequência de grupo, a ideia fundamental é a de que, quantos mais eventos um utilizador frequentar num dado grupo, maior é a probabilidade deste continuar a frequentar eventos desse mesmo grupo. Contudo, este modelo não toma em consideração as relações entre os utilizadores e os grupos nos quais estes se encontram inseridos, nem toma em consideração as relações entre os grupos e os eventos por eles criados. No modelo multi-relacional, essas interações são tidas em conta. O objetivo é tentar perceber, por exemplo, se utilizadores associados a um grupo são propensos a frequentar eventos criados por esse grupo.

No que ao conteúdo textual dos eventos diz respeito, o objetivo é tentar perceber qual o grau de similaridade entre um dado utilizador e um dado evento. Assim, ao tomar em consideração o conteúdo textual de um dado evento, bem como as palavras extraídas dos eventos frequentados por um dado utilizador no passado, é possível aferir este valor de similaridade.

No que ao contexto da localização diz respeito, foi proposta uma abordagem de estimação de densidade do tipo *kernel-based*, por forma a modelar os padrões de mobilidade dos utilizadores como distribuições das distâncias geográficas entre os eventos frequentados. A relevância de um dado evento para um utilizador baseia-se na probabilidade agregada desse mesmo evento se encontrar localizado nalguma das regiões nas quais decorreram eventos frequentados pelo utilizador.

No que diz respeito ao contexto temporal, o objetivo é descobrir, para cada utilizador, quais as suas preferências temporais em termos de frequência de eventos (e.g., se um dado utilizador gosta mais de frequentar eventos durante a manhã, ao início da tarde, etc.).

Por forma a aferir a eficiência da solução proposta, foram recolhidos dados da plataforma *Meetup*, de três cidades americanas, i.e., Phoenix, Chicago e San Jose, compreendidos entre Janeiro de 2010 a Abril de 2014. O desempenho da solução foi comparado com outros algoritmos estado de arte, em particular: *Most Popular*, *Bayesian Personalized Ranking-Matrix Factorization* (BPR-MF), e BPR-NET [12]. Neste último algoritmo, usam-se dois tipos de redes sociais, i.e., uma rede social baseada nos grupos partilhados de utilizadores, e uma rede social inferida a partir da co-participação em eventos, como termos de regularização de um modelo BPR-MF.

Com base nas experiências efetuadas, concluiu-se que a abordagem de recomendação proposta apresenta maior desempenho e robustez em relação aos algoritmos estado de arte da literatura. Concluiu-se também que os modelos de recomendação baseados em características contextuais, como o contexto social e o conteúdo dos eventos, apresentam a maior eficácia de recomendação. Os algoritmos baseados nos contextos geográficos e temporais são os menos eficazes. Contudo, fica bem patente a importância de se empregar o contexto na tarefa de recomendação de eventos. O motor de recomendação de eventos desenvolvido faz uso de informação contextual, por forma a providenciar recomendações precisas aos utilizadores.

O trabalho de [13] abordou a importância do uso de informação contextual como forma de elaborar predições de recomendação. O trabalho em questão aborda ainda o método *multiverse recommendation* [14], argumentando que este apresenta duas principais desvantagens: (1) a sua complexidade computacional é exponencial no número de variáveis de contexto e polinomial no tamanho da fatorização, i.e., apresenta uma complexidade de $O(k^m)$, e (2) apenas consegue modelar variáveis ca-

tegóricas (*categorical variables*), não sendo possível o uso de variáveis de conjuntos categóricos (*set categorical variables*), ou mesmo variáveis reais (*real-valued variables*). Além disso, é também abordado o problema da existência de uma grande variedade de métodos de recomendação que falham em considerar informação contextual.

Por forma a solucionar os problemas apresentados, os autores propõem a aplicação de *Factorization Machines* (FMs) na tarefa de modelar informação contextual e providenciar recomendações que tenham em consideração o contexto. As *Factorization Machines* (FMs) são uma abordagem genérica que combina a elevada precisão de predição dos modelos de fatorização com a flexibilidade da denominada *feature engineering*.

Seja S um conjunto de tuplos da forma $(\mathbf{x}, y)$, onde $\mathbf{x} \in \Re^n$ é um vetor de *features* que representa o utilizador e um item candidato, e onde y denota o alvo de predição. Uma FM modela todas as interações aninhadas, até uma dada ordem d, entre o número n de variáveis de entrada contidas em $\mathbf{x}$, através do uso de parâmetros de interação fatorizados [15]. A Equação 2.7 apresenta o modelo de predição de uma FM de ordem $d = 2$ [13]:

$$\hat{y}(x) := w_0 + \sum_{i=1}^{n} w_i \, x_i + \sum_{i=1}^{n} \sum_{j=i+1}^{n} \hat{w}_{i,j} \, x_i \, x_j \tag{2.7}$$

Na Equação 2.7, $\hat{w}_{i,j}$ denota os parâmetros de interação fatorizados entre pares, de acordo com a Equação 2.8:

$$\hat{w}_{i,j} := \langle \mathbf{v}_i, \mathbf{v}_j \rangle = \sum_{f=1}^{k} v_{i,f} \, . \, v_{j,f} \tag{2.8}$$

Nesta segunda equação, k denota a dimensionalidade da fatorização. Os parâmetros do modelo, $\Theta = \{w_0, w_1, ..., w_p, v_{1,1}, ..., v_{n,k}\}$, têm os seguintes domínios:

$$w_0 \in \Re, \quad \mathbf{w} \in \Re^n, \quad \mathbf{V} \in \Re^{n \times k} \tag{2.9}$$

A primeira parte do modelo FM descrito pela Equação 2.7 contém as interações unárias de cada variável de entrada x_i com o alvo, de forma semelhante a um modelo de regressão linear convencional. A segunda parte contém todos os pares de interações de variáveis de entrada, i.e., $x_i \, x_j$.
Por forma a obter modelos de predição FM, necessitamos primeiro de proceder ao seu treino. Existem vários algoritmos disponíveis para o efeito, tais como SGD (*Stochastic Gradient Descent*), ALS (*Alternating Least Squares*), MCMC (*Markov Chain Monte Carlo*), e ASGD (*Adaptative SGD*). As FMs são facilmente aplicáveis a uma grande variedade de contextos, unicamente através da especificação dos dados de entrada.

Seguidamente, procedeu-se à comparação do método *Factorization Machines* com o método *Multiverse Recommendation*. Três diferentes *datasets* foram empregues: (1) *Food*, (2) *Adom.*, e (3) *Yahoo! Webscope*. Desta comparação, ficou demonstrado que o método *Factorization Machines* detém a melhor performance em termos de complexidade computacional, quando comparado com o método *Multiverse Recommendation*. Por outro lado, demonstrou-se também que os métodos *Factorization*

Machines e *Multiverse Recommendation* apresentam uma qualidade de predição comparável, nos *datasets Food* e *Adom.*, enquanto que, para o *datset Yahoo! Webscope*, a solução proposta ultrapassa largamente o método *Multiverse Recommendation*. Conclui-se, deste modo, que a proposta apresentada [13] constitui assim uma ótima alternativa a considerar aquando da elaboração de soluções de recomendação que tenham em conta informação contextual, a qual se mostra de grande importância aquando da elaboração de predições de recomendação.

Hibridização em Paralelo

Na hibridização em paralelo, a pontuação de recomendação final é obtida através das pontuações de recomendação resultantes de várias abordagens de recomendação, por meio de uma combinação linear ou mecanismo de votação.

O trabalho de [16] propõe uma abordagem híbrida construída sobre a Web Semântica. A abordagem proposta combina um sistema de filtragem baseada no conteúdo enriquecido com *Linked Data* [17], e um sistema de filtragem colaborativa, por forma a envolver o aspeto social. Por fim, o sistema híbrido foi melhorado através da integração de um modelo de diversidade do utilizador. O uso de *Linked Data* como forma de enriquecer uma abordagem *Content-based* tem, como função, enriquecer o perfil de um item, facilitando assim a comparação com o perfil do utilizador.

Os valores de similaridade entre eventos são usados na obtenção de uma lista ordenada de eventos recomendados, de acordo com a seguinte fórmula:

$$\mathrm{rank}_{cb}(u, e_i) = \frac{\sum_{e_j \in E_u} \sum_{p \in P} \alpha_p \mathrm{sim}^p(e_i, e_j)}{\mid P \mid * \mid E_u \mid} \tag{2.10}$$

Nesta equação, e_i é o evento a recomendar, P é o conjunto de propriedades partilhadas entre dois eventos, E_u é o conjunto de eventos em que um utilizador u participou no passado, e α_p é o peso atribuído a cada propriedade, o qual reflete a contribuição de cada uma no que ao perfil dos utilizadores diz respeito. Foram selecionadas propriedades relacionadas com a localização, assunto, e agentes envolvidos. Contudo, a similaridade dos eventos numa abordagem *content-based* pode ser influenciada pela diversidade de tópicos dos eventos em que os utilizadores participaram no passado. Por forma a mitigar este impacto, foi introduzido um termo β na Equação 2.10, que denota os pesos dos eventos, dependendo se estes estão ou não incluídos nos picos de interesse dos utilizadores. Estes picos de interesse são detetados com recurso à técnica de modelação de tópicos *Latent Dirichlet Allocation* [10]. A Equação 2.10 é então reescrita da seguinte forma:

$$\mathrm{rank}_{cb++}(u, e_i) = \frac{\sum_{e_j \in E_u} \sum_{p \in P} \alpha_p \beta_p \mathrm{sim}^p(e_i, e_j)}{\mid P \mid * \mid E_u \mid} \tag{2.11}$$

A abordagem colaborativa não só considera a similaridade entre utilizadores, como também a contribuição de um grupo de amigos, sendo a predição de um utilizador u_i participar num evento e dada pela seguinte fórmula:

$$\mathrm{rank}_{cf}(u_i, e) = \frac{\sum_{j \in C} a_{i,j}}{\mid C \mid} * \frac{\mid E_i \cap (\cup_{j \in C} E_j) \mid}{\mid E_i \mid} \tag{2.12}$$

Nesta equação, C é o conjunto de utilizadores que vão co-participar num evento e juntamente com o utilizador u_i, E_i denota o conjunto de eventos em que o utilizador u_i participou, e $a_{i,j}$ denota a fração de eventos em comum entre os utilizadores u_i e u_j, na cardinalidade de E_j.

Por fim, a combinação das abordagens acima descritas, numa abordagem híbrida, é conseguida através da seguinte combinação linear das pontuações de cada uma das abordagens descritas anteriormente:

$$\mathrm{rank}(u, e) = \mathrm{rank}_{cb++}(u, e) + \alpha_{cf} \mathrm{rank}_{cf}(u, e) \tag{2.13}$$

O parâmetro α_{cf} é o peso da componente de filtragem colaborativa.

Por forma a testar o sistema, foram usados *datasets* obtidos de três grandes diretórios públicos de eventos: *Last.fm*, *Eventful*, e *Upcoming*. Com base nestes *datasets*, foram testadas as seguintes abordagens de recomendação: *User-based CF* tradicional, UBExtended (*Probability based Extended Profile Filtering*) [18], e a abordagem proposta pelos autores do presente artigo (*CF+CB-based++*). A eficácia da solução híbrida proposta foi aferida com recurso às métricas de precisão e cobertura (recall). Por fim, demonstrou-se que os valores de precisão e cobertura (recall) são maiores para a abordagem híbrida desenvolvida (*CF+CB-based++*), em comparação com as outras abordagens testadas. Estes resultados realçam a importância do uso de abordagens híbridas como forma de desenvolver sistemas de recomendação de eventos eficazes e com bom desempenho.

No trabalho de [19] são propostas três abordagens de recomendação de eventos, baseadas na similaridade semântica (*Similarity Based Approach*, ou SBA), nas relações entre utilizadores (*Relationship Based Approach*, ou RBA), e no historial de participação dos utilizadores em eventos (*History Based Approach*, ou HBA). Estas três abordagens de recomendação são depois combinadas numa abordagem híbrida, a qual usa uma soma pesada de cada uma das pontuações de recomendação de cada uma das três abordagens anteriores, por forma a calcular a similaridade entre um utilizador e um evento.

Na abordagem baseada na similaridade semântica (SBA), são recomendados a um utilizador os eventos que apresentem uma elevada similaridade com este, com base na similaridade de distribuição de tópicos. Para cada documento representativo de um evento é gerada a distribuição de tópicos usando a técnica *Latent Dirichlet Allocation* [10]. Para esta abordagem, a pontuação de recomendação de um evento e_j para um utilizador u_i é dada pela seguinte fórmula, a qual corresponde à similaridade do coseno:

$$S_1(u_i, e_j) = \mathrm{sim}(u_i, e_j) = \cos(\overrightarrow{\theta}_{u_i}, \overrightarrow{\theta}_{e_j}) = \frac{\overrightarrow{\theta}_{u_i} . \overrightarrow{\theta}_{e_j}}{\| \overrightarrow{\theta}_{u_i} \| \| \overrightarrow{\theta}_{e_j} \|} \tag{2.14}$$

Na Equação 2.14, $\overrightarrow{\theta}_{u_i}$ e $\overrightarrow{\theta}_{e_j}$ denotam os vetores normalizados para a distribuição de tópicos, respetivamente para um utilizador u_i e para um evento e_j.

Na abordagem baseada no relacionamento (RBA), a recomendação de eventos é feita com base no princípio de que utilizadores com interesses semelhantes tendem a frequentar os mesmos eventos. A similaridade do coseno é normalmente utilizada para o cálculo da similaridade entre as distribuições de tópicos, para dois utilizadores u_i e $u_{i'}$. A pontuação de recomendação para esta abordagem é dada pela Equação 2.15:

$$S_2(u_i, e_j) = \frac{\sum_{u_k \in F(u_i) \cap A(e_j)} \mathrm{sim}(u_i, u_k)}{\mid F(u_i) \cap A(e_j) \mid} \tag{2.15}$$

Na equação acima, $F(u_i) \cap A(e_j)$ denotam os amigos do utilizador u_i que participaram no evento e_j.

Na abordagem baseada no histórico de participação de cada utilizador (HBA), a recomendação é vista como um problema de classificação, sendo treinado um modelo de regressão logística para cada utilizador, utilizando, para tal, a distribuição de tópicos de eventos em que o mesmo participou no passado. A pontuação de recomendação para esta abordagem é dada pela Equação 2.16:

$$S_3(u_i, e_j) = f_{u_i} = \frac{1}{1 + e^{-z}} \tag{2.16}$$

Na Equação 2.16, f_{u_i} denota o output da função de regressão logística sobre um evento futuro, e z é dado pela Equação 2.17, em que k denota o número de tópicos, θ_{e_j} denota o vetor de distribuição de tópicos para o evento e_j, $\theta_{e_j}^{(t)}$ denota o valor para o tópico t no vetor de distribuição de tópicos, e $\overrightarrow{\beta} = [\beta_0, \beta_1, ..., \beta_k]^T$ são os parâmetros para o modelo de regressão logística de um utilizador em particular.

$$z = \beta_0 + \beta_1 \theta_{e_j}^{(1)} + ... + \beta_k \theta_{e_j}^{(k)} \tag{2.17}$$

Para todas as três abordagens acima, os resultados são ordenados por ordem decrescente das pontuações obtidas.

Por fim, as três abordagens acima são combinadas numa abordagem híbrida, sendo a pontuação de recomendação final dada pela Equação 2.18:

$$S(u_i, e_j) = \omega_1 S_1(u_i, e_j) + \omega_2 S_2(u_i, e_j) + \omega_3 S_3(u_i, e_j) \tag{2.18}$$

O vetor de parâmetros $\overrightarrow{\omega} = [\omega_0, \omega_1, ..., \omega_k]^T$ denota os pesos de cada uma das três abordagens anteriores.

Por forma a testar a solução desenvolvida, foram usados dois *datasets*, nomeadamente um para efeitos de recomendação de eventos académicos, que podem ser conferências ou workshops (obtido através de *Linked Data Enablement*), e um para testar a recomendação de eventos no Facebook. Para cada um dos *datasets*, foram testadas as três abordagens de recomendação de eventos propostas e a abordagem híbrida resultante da combinação das três abordagens, em conjunto com uma estratégia de recomendação aleatória. Com base nos testes efetuados em cada um dos *datasets*, pôde-se concluir que cada uma das três abordagens de recomendação propostas apresenta um desempenho superior à abordagem de recomendação aleatória, sendo que a abordagem híbrida resultante da combinação das

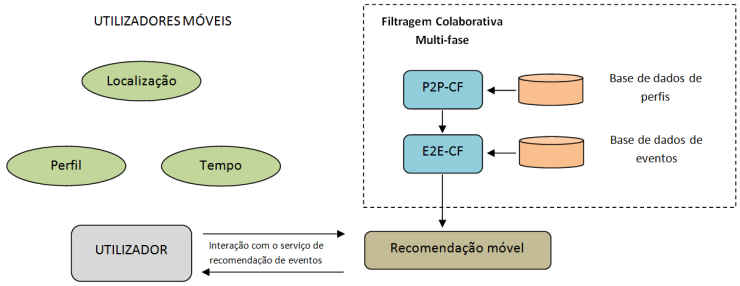

Figura 2.3: A arquitetura do sistema de recomendação baseado em filtragem colaborativa multi-fase [20].

três abordagens de recomendação supera todas as outras abordagens de recomendação testadas. Fica assim clarificada a importância do uso de estratégias híbridas como forma de obter uma maior eficácia e desempenho de recomendação.

Hibridização em Pipelining

Na hibridização em *pipelining*, a pontuação de recomendação final é obtida à custa de várias abordagens de recomendação montadas em série. Cada uma delas efetua algum tipo de pré-processamento sobre os dados de entrada que recebeu, sendo o resultado do pré-processamento passado ao próximo bloco de recomendação, até se chegar a uma pontuação de recomendação final.

No trabalho de [20] é discutida a importância de serviços que sejam personalizados e baseados na localização, argumentando que a maioria dos provedores de serviços móveis não consideram as necessidades dos utilizadores, em termos da sua localização e participação em eventos. Tal facto pode levar o provedor de serviços móveis a perder oportunidades de oferecer melhores serviços aos seus utilizadores, levando a uma consequente perda de lucros. Tendo em conta este problema, os autores do artigo apresentam um processo de Filtragem Colaborativa Multi-fase (MSCF), por forma a providenciar recomendações de eventos com base na localização e perfil dos utilizadores de serviços móveis, levando ainda em conta o fator temporal.

O processo desenvolvido apresenta duas fases. Na primeira fase, é efetuada uma filtragem colaborativa do tipo *User-to-User*, cujo objetivo consiste em aglomerar utilizadores vizinhos com perfis e preferências semelhantes, com recurso a uma rede ART (*Adaptative Resonance Theory*). Na segunda fase, é aplicada uma filtragem colaborativa do tipo *Item-to-Item*, cujo objetivo consiste em descobrir os padrões de participação de eventos dos utilizadores móveis, gerando por fim regras sequenciais. A arquitetura da solução proposta é a apresentada na Figura 2.3.

O processo de recomendação desenrola-se da seguinte forma: em primeiro lugar, procede-se à identificação do utilizador móvel, seguindo-se a obtenção da informação do mesmo. O passo seguinte consiste na obtenção das regras sequenciais. Neste ponto, convém realçar o facto de que podem existir no sistema dois tipos de utilizadores móveis: (1) utilizadores móveis recentemente registados e (2) restantes utilizadores móveis. Assim, para se proceder à obtenção das regras sequenciais para utilizadores recentemente registados no sistema (e, consequentemente, com um histórico de participação em

eventos reduzido), o sistema obtém as regras sequenciais de utilizadores similares, através do uso das regras sequenciais do aglomerado de utilizadores ao qual o utilizador recentemente registado pertence. Para os restantes utilizadores no sistema, são obtidas as regras sequenciais individuais dos mesmos, bem como as regras sequenciais do aglomerado ao qual o utilizador pertence, tal como no caso dos utilizadores recentemente registados no sistema.

As regras sequenciais obtidas permitem prever as próximas localizações possíveis dos utilizadores móveis. Ao cruzar esta informação com informação respeitante a eventos, tal como contida numa base de dados de eventos, são obtidas correspondências entre as localizações futuras dos utilizadores e os eventos, permitindo descobrir eventos que possam ser de interesse para os utilizadores móveis. O passo final consiste em ordenar os eventos a recomendar, de acordo com as preferências dos utilizadores móveis.

Por forma a validar a solução desenvolvida, foi montada uma experiência que envolveu a aquisição de informação de 1452 utilizadores móveis em Taiwan, tendo sido pedido a cada utilizador que preenchesse o seu perfil pessoal, o qual incluía os seguintes atributos: sexo, estado civil, data de nascimento, e as preferências de eventos. Estes atributos foram usados para determinar a idade e informação demográfica de cada um dos utilizadores. O objetivo da experiência consistiu em avaliar o desempenho de recomendação, através da comparação entre a abordagem proposta no artigo (MSCF) e as abordagens *user-based CF* e *item-based CF* tradicionais. Foram então obtidos resultados experimentais, para os casos de utilizadores registados à já algum tempo (ORMU) e para utilizadores recentemente registados (NRMU). Em ambos os casos, mostrou-se que a solução proposta (MSCF) apresentava os melhores resultados, em comparação com as abordagens *user-based CF* e *item-based CF*.

Assim, ficam bem patentes as vantagens da hibridização, bem como o uso de informação contextual, em particular no que diz respeito à localização e perfis dos utilizadores e aos fatores temporais. O uso de regras sequenciais, por forma a aferir os padrões de movimentação dos utilizadores móveis, mostrou-se igualmente importante na tarefa de recomendação de eventos.

Comparação de Abordagens para a Recomendação de Eventos

O tema da recomendação de eventos tem gerado, ao longo dos anos, várias pesquisas e trabalhos de investigação, cujo objetivo último consiste na criação de abordagens de recomendação de eventos eficientes e eficazes, sejam elas híbridas ou não. Contudo, com o aparecimento de várias abordagens e soluções de recomendação de eventos, tornou-se necessária a sua avaliação, por forma a estabelecer quais os melhores algoritmos e abordagens de recomendação existentes.

O trabalho de [8] investiga e discute características importantes das denominadas *event-based social networks* (EBSN), tais como a análise dos RSVPs[2], o tempo de vida dos eventos, a co-participação de utilizadores em eventos, e dados geográficos. Estas características podem afetar a forma como os sistemas de recomendação de eventos são desenhados, bem como a sua eficácia. Simultaneamente, são também testados vários algoritmos de recomendação (*random*, *most-popular*, *location-*

[2]Do francês *répondez s'il vous plaît*. Os RSVPs contém informação acerca da intenção de um dado utilizador participar ou não num dado evento.

aware, *Bayesian personalized ranking-matrix factorization* (BPR-MF), *user-KNN*, *item-KNN*, e regressão logística), por forma a tirar conclusões acerca do seu desempenho e respetivas limitações.

Para este estudo, foram recolhidos dados referentes a três cidades americanas: Phoenix, Chicago e San José, a partir do *Meetup*, uma EBSN. Com base nos dados recolhidos, foram obtidos todos os utilizadores, eventos, e pares utilizador-evento (RSVPs), tendo sido em seguida analisados: O número de RSVPs positivos, o tempo de vida dos eventos, se as respostas aos eventos (RSVPs) têm lugar mais longe ou mais perto da data em que os eventos ocorrem, a distribuição da co-participação em eventos por dois utilizadores distintos, e os dados referentes às distâncias entre os utilizadores e os eventos.

O estudo começou por investigar a relevância dos RSVPs. Quando um evento é criado, os vários utilizadores podem manifestar a sua intenção de participar ou não nesse evento. Tal informação é dada pelos RSVPs dos eventos. Considera-se que um utilizador que manifeste a sua intenção de participar num evento tem maior probabilidade de participar, de facto, nesse mesmo evento, do que um outro utilizador que não manifeste vontade em participar, ou nem sequer responda se pretende ou não participar. Assim, analisando os RSVPs dos eventos, podemos inferir qual a taxa de participação aproximada para um dado evento.

No que se refere à análise do tempo de vida dos eventos, os autores consideram o tempo de vida de um evento como sendo o período de tempo entre a criação do mesmo (neste caso, no *Meetup*) e a sua ocorrência. Os dados obtidos a partir do *Meetup*, para cada uma das três cidades consideradas, mostram que a maioria dos eventos possui um período de vida que varia entre os 5 e os 100 dias.

Foi também investigado quando é que se dá a ocorrência de RSVPs positivos durante o período de vida dos eventos. As conclusões tiradas, tendo em conta os dados recolhidos, revelam que os eventos recebem uma maior quantidade de RSVPs positivos à medida que a data de ocorrência dos mesmos se aproxima. Note-se que, aquando do lançamento de um novo evento no *Meetup*, haverá pouca informação, em termos de RSVPs, que possa ser usada por abordagens de filtragem colaborativa, dando-se assim preferência às abordagens de filtragem baseada no conteúdo. O uso de abordagens de filtragem colaborativa ou híbridas é favorecido à medida que mais RSVPs sejam fornecidos.

Foi também investigada a co-participação de dois utilizadores diferentes num mesmo evento, em termos de RSVPs positivos. A análise efetuada revelou que apenas cerca de 30% dos utilizadores co-participam em dois ou mais eventos, para todas as cidades consideradas no estudo.

A análise da distribuição das distâncias entre as moradas dos utilizadores e a localização dos eventos foi também investigada. Pôde concluir-se, a partir dos dados obtidos, que 50% dos utilizadores manifestaram a sua intenção de participar em eventos num raio de 10 Km das suas moradas, ao passo que nenhum utilizador respondeu se pretendia comparecer ou não em eventos que distassem mais do que 100 Km das suas moradas.

Por fim, foram testados os vários algoritmos de recomendação anteriormente citados. Pôde concluir-se que o algoritmo *user-KNN* apresentou o melhor desempenho na tarefa de recomendar eventos aos utilizadores. Uma explicação possível para o bom desempenho deste algoritmo, o qual é relativamente simples, pode prender-se com o facto de que, em muitos casos, utilizadores que participem no mesmo evento são amigos ou conhecidos, sendo exercida, como tal, influência mútua na seleção de eventos

futuros. Constatou-se também que os algoritmos *item-KNN* e regressão logística possuem um desempenho comparável ao do algoritmo *user-KNN*.

O trabalho de [7] focou-se na elaboração de um estudo, centrado nos utilizadores, com vista a encontrar um algoritmo de recomendação de eventos que melhore a satisfação dos utilizadores, para um website de eventos culturais belga. Foram testados cinco algoritmos diferentes neste estudo: *Random* (Recomendação aleatória), *User-based Nearest Neighbor Colaborative Filtering* (UBCF), *Singular Value Decomposition* (SVD), *Content-Based Filtering* (CB), e uma abordagem híbrida (UBCF + CB).

Neste estudo, foi recolhido *feedback* explícito (i.e., feedback sob a forma de classificações explícitas) e implícito (i.e., sob a forma de cliques e navegação através das páginas de informação dos eventos) dos utilizadores, durante um período de 41 dias. Este *feedback* foi usado como input para os cinco diferentes algoritmos de recomendação mencionados anteriormente. A cada um dos utilizadores foi atribuído, de forma aleatória, um algoritmo diferente de entre os 5 disponíveis. Foi então pedido aos mesmos que respondessem a um inquérito online, composto por 14 perguntas, por forma a avaliarem a recomendação gerada pelo algoritmo de recomendação que lhe fora atribuído, em termos de: (1) precisão da recomendação, (2) novidade, (3) diversidade, (4) satisfação e (5) confiança no sistema. As respostas foram dadas segundo uma escala de Likert de 5 pontos, desde *discordo totalmente* (1) a *concordo totalmente* (5).

Das 14 perguntas do inquérito, as seguintes 8 foram selecionadas pelos autores do trabalho como sendo as mais relevantes para o seu estudo, no que diz respeito aos vários aspetos dos sistemas de recomendação: (1) Os items que me foram recomendados combinam com os meus interesses; (2) Alguns dos items recomendados são-me familiares; (3) O sistema de recomendação ajudou-me a descobrir novos produtos; (4) Os items que me foram recomendados apresentam semelhanças entre eles; (5) Não entendi porque me foram recomendados os items; (6) Globalmente, estou satisfeito com o sistema de recomendação; (7) O sistema de recomendação é de confiança; (8) Iria participar em alguns dos eventos que me foram recomendados, dada a oportunidade para tal.

Com base nos dados recolhidos e nas respostas obtidas por parte dos utilizadores, pôde concluir-se que: (1) o algoritmo CB + UBCF superou todos os restantes algoritmos, exceto no que à diversidade (pergunta 4) diz respeito; (2) o algoritmo de recomendação aleatória (RAND) obteve a melhor classificação em termos da diversidade (pergunta 4); (3) os dois aspetos qualitativos que mais se correlacionaram com a satisfação dos utilizadores (pergunta 6) foram a precisão da recomendação (pergunta 1) e a transparência (pergunta 5). Por outro lado, concluiu-se que a diversidade (pergunta 4) não se encontra correlacionada com a satisfação dos utilizadores (pergunta 6), confiança (pergunta 7), ou qualquer outro aspeto qualitativo estudado; (4) o algoritmo SVD surgiu em último lugar, em conjunto com o algoritmo RAND, exceto no que à diversidade (pergunta 4) diz respeito.

Este estudo mostra, mais uma vez, o potencial das abordagens de recomendação híbridas, em termos de eficácia e desempenho de recomendação, em relação a outras abordagens alternativas. Mostra também a importância de se efetuarem estudos orientados à satisfação dos utilizadores, no que às várias características dos sistemas de recomendação de eventos diz respeito.

2.3 Sumário

Neste capítulo foram apresentados, em primeiro lugar, uma série de conceitos importantes no âmbito dos sistemas de recomendação, nomeadamente:

- Filtragem colaborativa (*Collaborative Filtering*);
- Filtragem baseada no conteúdo (*Content-Based Filtering*);
- Classificação supervisionada (*Supervised Learning*).

Por fim, foram abordados e discutidos trabalhos relacionados no âmbito dos sistemas de recomendação (em particular, no âmbito dos sistemas de recomendação de eventos), em particular:

- Abordagens simples ao problema da recomendação de eventos;
- Abordagens híbridas como forma de abordar a recomendação de eventos;
- Comparação de abordagens para a recomendação de eventos.

Capítulo 3

Recomendação de Eventos

Neste capítulo serão apresentados e discutidos todos os aspectos relativos à solução de recomendação de eventos implementada. Em particular, irão ser abordados os seguintes aspectos:

- Quais as *features* desenvolvidas, assim como uma breve explicação de cada uma delas;
- Qual o modelo de classificação aplicado, por forma a gerar as listas de recomendação para cada utilizador;
- Qual a arquitectura do motor de recomendação de eventos implementado;
- A importância da selecção de *features* na obtenção de motores de recomendação de eventos eficazes, bem como o procedimento de selecção de *features* aplicado no caso concreto.

Por fim, serão sumarizados os aspectos mais relevantes do corrente capítulo.

3.1 Features

Por forma a implementar um motor de recomendação de eventos eficaz e preciso, é necessária a criação e implementação de um conjunto de *features*. Estas têm, por objectivo, modelar cada par utilizador-evento em termos de um conjunto de características próprias, a fim de se proceder à aplicação de um classificador, o qual irá determinar a lista de eventos recomendados para cada utilizador tratado.

O processo de criação/implementação de *features* apresenta-se como um processo complexo e incremental, que envolve uma análise completa e exaustiva dos dados que serão mais tarde usados na realização de experiências (ver Capítulo 4, Secção 4.1). Assim, e após de uma análise exaustiva do *dataset* a usar na presente dissertação[1], foram implementadas 45 *features*, as quais, por sua vez, se dividem em 6 grupos distintos:

1. Dados demográficos e de localização;

[1] Na presente dissertação, foi aplicado o *dataset* do *Kaggle Event Recommendation Engine Challenge*, uma competição que teve lugar na plataforma *Kaggle* em 2013. Consultar a Secção 4.1.1 do Capítulo 4 da presente dissertação para uma descrição mais detalhada acerca deste *dataset*.

2. Dados dos utilizadores;

3. Dados temporais;

4. Dados relacionados com o aspecto social;

5. Dados colaborativos;

6. Dados de similaridade.

Em seguida, são enumeradas as várias *features* constituintes de cada um dos grupos acima apresentados.

3.1.1 Dados Demográficos e de Localização

Este grupo compreende todas as *features* que explorem informação demográfica e de localização relativa aos utilizadores e aos eventos, sendo composto pelas seguintes *features*:

1. **Distância, em quilómetros, entre a morada do utilizador alvo de recomendação e a localização do evento a recomendar**: A obtenção das distâncias geográficas, medidas em quilómetros, entre as moradas dos utilizadores e as localizações dos eventos é feita com recurso às coordenadas geodésicas (i.e., latitude e longitude) das localizações dos eventos, bem como às localizações dos utilizadores, na forma de uma string representativa da sua localização, com o formato $<$ Cidade, País $>$. A obtenção das coordenadas geodésicas referentes às moradas dos utilizadores, bem como o cálculo das distâncias entre as moradas dos mesmos e as localizações dos eventos, é feita com recurso à biblioteca de geocodificação *geopy*[2]. A probabilidade de um utilizador participar num dado evento é, em princípio, tanto maior quanto menor for a distância entre a morada do utilizador e a localização do evento;

2. **Se a cidade e país do utilizador alvo de recomendação são iguais à cidade e país do evento a recomendar**: São obtidas as localizações dos utilizadores e dos eventos, na forma de uma string representativa da sua localização, com o formato $<$ Cidade, País $>$. Seguidamente, são obtidas as coordenadas geodésicas das localizações dos utilizadores e dos eventos, com recurso à biblioteca de geocodificação *geopy*. Esta *feature* valerá 1 se a cidade e país do utilizador alvo de recomendação forem as mesmas das do evento a recomendar, e 0 caso contrário. A probabilidade de um utilizador participar num dado evento é, em princípio, tanto maior se o evento a recomendar se situar na mesma cidade e país de onde o utilizador alvo de recomendação é proveniente.

3.1.2 Dados dos Utilizadores

Este grupo compreende todas as *features* que explorem informação relativa aos utilizadores, sendo composto pelas seguintes *features*:

[2]https://pypi.python.org/pypi/geopy, https://github.com/geopy/geopy

3. **Número de eventos nos quais o utilizador alvo de recomendação participou no passado**: Se o número de eventos nos quais um dado utilizador participou no passado for grande, então é provável que esse mesmo utilizador venha a participar em muitos eventos no futuro. Por outras palavras, se um utilizador participou em muitos eventos no passado, então estaremos perante um utilizador que exibe um perfil de actividade social de grande intensidade, pelo que a probabilidade deste vir a frequentar eventos futuros é, em princípio, maior que aquela que se apresenta para um utilizador com um perfil de actividade social média ou de pouca intensidade;

4. **Número de eventos nos quais o utilizador alvo de recomendação talvez tenha participado no passado**: Pode suceder que, por algum motivo, um dado utilizador pretenda participar num dado evento, mas não tenha certezas definitivas acerca da sua participação nesse mesmo evento. Nesses casos, o utilizador pode expressar a sua intenção de talvez participar nesse mesmo evento. A motivação é análoga àquela apresentada para a *feature* 3;

5. **Número de eventos para os quais o utilizador alvo de recomendação foi convidado a participar no passado**: Motivação análoga àquela apresentada para a *feature* 3;

6. **Número de eventos nos quais o utilizador alvo de recomendação não participou no passado**: Motivação análoga àquela apresentada para a *feature* 3;

7. **Se o utilizador alvo de recomendação é do sexo masculino**: Todos os eventos, pela sua natureza, são dirigidos a um determinado público-alvo. Dependendo do evento a considerar, este pode ter, como público-alvo, utilizadores do sexo masculino, feminino, ou poderá ser dirigido a pessoas de ambos os sexos. Utilizadores dos sexos masculino ou feminino terão tendência para participar em eventos dirigidos ao público masculino e feminino, respectivamente. Contudo, é improvável que uma pessoa do sexo masculino frequente um evento dirigido ao público feminino, ou vice-versa. A presente *feature* pretende detectar estas situações;

8. **Se o utilizador alvo de recomendação é do sexo feminino**: Motivação análoga àquela apresentada para a *feature* anterior;

9. **Idade do utilizador**: Assim como os eventos podem ser dirigidos a pessoas de diferentes géneros, podem ser também dirigidos a pessoas de diferentes idades. Com a presente *feature*, pretende-se capturar esta informação, tornando-a útil no processo de recomendação de eventos. A determinação das idades para cada utilizador é feita em relação ao ano de 2016.

3.1.3 Dados Temporais

Este grupo compreende todas as *features* que explorem informação temporal como factor de relevo na recomendação de eventos, sendo composto pela seguinte *feature*:

10. **Diferença de tempo, em segundos, entre o instante no qual o evento a recomendar vai ter lugar e o instante no qual o utilizador alvo de recomendação visualizou o evento a recomendar no sistema**: Seja t_{evento} o instante no qual o evento a recomendar irá decorrer no futuro,

e $t_{\text{visualização}}$ o instante no qual o utilizador alvo de recomendação viu o evento a recomendar no sistema. A ideia por detrás do emprego desta *feature* advém do trabalho de [8], o qual investigou, entre outras coisas, a ocorrência de RSVPs positivos durante o período de vida dos eventos, em EBSNs, tendo concluído que os eventos recebem uma maior quantidade de RSVPs positivos à medida que a data de ocorrência dos mesmos se aproxima. Assim, pode-se usar esta informação para induzir se um dado utilizador vai ou não participar num dado evento futuro. A probabilidade de um dado utilizador participar num dado evento é tanto maior quanto menor for a diferença entre t_{evento} e $t_{\text{visualização}}$.

3.1.4 Dados Relacionados com o Aspecto Social

Este grupo compreende todas as *features* que explorem informação relativa ao aspecto social, sendo composto pelas seguintes *features*:

11. **Número de eventos nos quais os amigos do utilizador alvo de recomendação participaram no passado**: Nos casos apresentados nas *features* 3 a 6, usou-se o perfil de actividade social dos utilizadores (sob a forma do número de eventos nos quais estes participaram, talvez tenham participado, foram convidados a participar, ou não participaram no passado) como forma de prever se os mesmos poderão ou não vir a participar em eventos futuros. De forma análoga, podemos considerar que um utilizador que possua um círculo de amizades no qual os amigos apresentem um perfil de actividade social de grande intensidade (definido de forma análoga ao apresentado nas *features* 3 a 6) tenha mais propensão a participar em eventos futuros que um outro que possua um círculo de amizades no qual os amigos apresentem um perfil de actividade social médio ou de pouca intensidade;

12. **Número de eventos nos quais os amigos do utilizador alvo de recomendação talvez tenham participado no passado**: Motivação análoga àquela apresentada para a *feature* 11;

13. **Número de eventos para os quais os amigos do utilizador alvo de recomendação foram convidados a participar no passado**: Motivação análoga àquela apresentada para a *feature* 11;

14. **Número de eventos nos quais os amigos do utilizador alvo de recomendação não participaram no passado**: Motivação análoga àquela apresentada para a *feature* 11;

15. **Se o utilizador alvo de recomendação foi ou não convidado a participar no evento a recomendar**: Sempre que um dado utilizador é convidado a participar num dado evento, existe uma possibilidade de que a pessoa que enviou o convite de participação conheça os gostos e preferências do utilizador a quem o convite se dirigiu. Assim, a probabilidade de um dado utilizador participar num dado evento a recomendar será, em princípio, tanto maior se o mesmo tiver sido convidado a participar no evento a recomendar;

16. **Número de amigos do utilizador alvo de recomendação**: Contrariamente ao que sucede nas *features* 3 a 6 (nas quais o perfil de actividade social dos utilizadores era medido em termos do

número de eventos nos quais os utilizadores tivessem participado, talvez tivessem participado, tivessem sido convidados a participar, ou não tivessem participado), e nas *features* 11 a 14 (nas quais o perfil de actividade social dos utilizadores era medido em termos do número de eventos nos quais os amigos dos utilizadores tivessem participado, talvez tivessem participado, tivessem sido convidados a participar, ou não tivessem participado), na presente *feature* o perfil de actividade social dos utilizadores é medido em termos do tamanho do círculo de amizades destes. Como consequência, utilizadores que possuam um círculo de amizades de grande dimensão serão considerados utilizadores de grande actividade social e, como tal, serão mais propensos a frequentar eventos futuros que os utilizadores que possuam um círculo de amizades de dimensão reduzida;

17. **Número de amigos do utilizador alvo de recomendação que vão participar no evento a recomendar**: Além do aspecto referido na *feature* 16, há que considerar também o grau de similaridade entre os utilizadores e o seu círculo de amizades. Se uma grande parcela dos amigos de um dado utilizador decidir participar num dado evento futuro, então a probabilidade desse mesmo utilizador poder vir a frequentar esse mesmo evento futuro será grande, uma vez que o utilizador e o seu círculo de amizades terão, em princípio, gostos e preferências similares;

18. **Rácio entre o número de amigos do utilizador alvo de recomendação que vão participar no evento a recomendar e o número de amigos do utilizador alvo de recomendação**: Esta *feature* constitui uma representação alternativa à informação apresentada na feature 17;

19. **Número de amigos do utilizador alvo de recomendação que talvez participem no evento a recomendar**: Motivação análoga àquela apresentada para a *feature* 17;

20. **Rácio entre o número de amigos do utilizador alvo de recomendação que talvez participem no evento a recomendar e o número de amigos do utilizador alvo de recomendação**: Esta *feature* constitui uma representação alternativa à informação apresentada na feature 19;

21. **Rácio entre o número de amigos do utilizador alvo de recomendação que talvez participem no evento a recomendar e o número de amigos do utilizador alvo de recomendação que vão participar no evento a recomendar**: Trata-se de mais uma medida de popularidade dos eventos, medida em função do número de amigos do utilizador que nele talvez participe, em relação a todos os amigos do utilizador que nele efectivamente vão participar. Nesta situação, considera-se que um utilizador que expressou a sua vontade de talvez participar num dado evento futuro poderá vir, com grande probabilidade, a consumar a sua participação nesse mesmo evento futuro. Os diferentes valores que este rácio pode tomar, para esta *feature*, bem como os respectivos significados, são descritos na Equação 3.1:

$$\begin{cases} \text{Feature } 21 > 1 & \text{Popularidade elevada} \\ \text{Feature } 21 = 1 & \text{Popularidade neutra} \\ \text{Feature } 21 < 1 & \text{Popularidade reduzida} \end{cases} \tag{3.1}$$

Sendo a popularidade dos eventos um dos factores determinantes no processo de recomendação destes a utilizadores, considera-se que a probabilidade de um dado utilizador participar num dado evento a recomendar será tanto maior quanto mais popular for esse mesmo evento a recomendar, medido de acordo com esta *feature*;

22. **Número de amigos do utilizador alvo de recomendação que foram convidados a participar no evento a recomendar**: Motivação análoga àquela apresentada para a *feature* 17;

23. **Rácio entre o número de amigos do utilizador alvo de recomendação que foram convidados a participar no evento a recomendar e o número de amigos do utilizador alvo de recomendação**: Esta *feature* constitui uma representação alternativa à informação apresentada na feature 22;

24. **Rácio entre o número de amigos do utilizador alvo de recomendação que foram convidados a participar no evento a recomendar e o número de amigos do utilizador alvo de recomendação que vão participar no evento a recomendar**: Motivação análoga àquela apresentada para a *feature* 21;

25. **Número de amigos do utilizador alvo de recomendação que não vão participar no evento a recomendar**: Motivação análoga àquela apresentada para a *feature* 17;

26. **Rácio entre o número de amigos do utilizador alvo de recomendação que não vão participar no evento a recomendar e o número de amigos do utilizador alvo de recomendação**: Esta *feature* constitui uma representação alternativa à informação apresentada na feature 25;

27. **Rácio entre o número de amigos do utilizador alvo de recomendação que não vão participar no evento a recomendar e o número de amigos do utilizador alvo de recomendação que vão participar no evento a recomendar**: De forma análoga ao que foi já descrito nas *features* 21 e 24, considera-se que a popularidade de um evento será tanto menor quanto maior for o número de amigos do utilizador que nele não vá participar, em relação a todos os amigos do utilizador que nele efectivamente vão participar. Os diferentes valores que este rácio pode tomar, para esta *feature*, bem como os respectivos significados, são descritos na Equação 3.2:

$$\begin{cases} \text{Feature } 27 > 1 & \text{Popularidade reduzida} \\ \text{Feature } 27 = 1 & \text{Popularidade neutra} \\ \text{Feature } 27 < 1 & \text{Popularidade elevada} \end{cases} \tag{3.2}$$

    A motivação da presente *feature* é análoga àquela apresentada para as *features* 21 e 24.

28. **Se o utilizador que criou o evento a recomendar é ou não amigo do utilizador alvo de recomendação**: Dado que um utilizador e o seu círculo de amizades podem, com grande probabilidade, apresentar os mesmos gostos e preferências, a probabilidade de um dado utilizador participar num dado evento a recomendar será, em princípio, tanto maior se o evento a recomendar tiver sido criado por um dos amigos do utilizador alvo de recomendação.

3.1.5 Dados Colaborativos

Este grupo compreende todas as *features* que explorem informação referente à participação em eventos por parte da generalidade dos utilizadores do sistema, sendo composto pelas seguintes *features*:

29. **Número de utilizadores que vão participar no evento a recomendar**: Um evento será tanto mais popular quanto maior for o número de utilizadores que nele vai participar. Assim, a probabilidade de um dado utilizador participar num dado evento a recomendar será, em princípio, tanto maior quanto mais popular for esse mesmo evento a recomendar;

30. **Número de utilizadores que talvez participem no evento a recomendar**: Motivação análoga àquela apresentada para a *feature* 29;

31. **Rácio entre o número de utilizadores que talvez participem no evento a recomendar e o número de utilizadores que vão participar no evento a recomendar**: A motivação da presente *feature* é análoga àquela apresentada para as *features* 21, 24 e 27. Os valores do rácio determinado por esta *feature*, bem como os respectivos significados, são em tudo idênticos àqueles determinados para as *features* 21 e 24, sendo agora o seu cálculo feito com base no universo de todos os utilizadores, e não com base no universo restrito de amigos do utilizador alvo de recomendação;

32. **Número de utilizadores que foram convidados a participar no evento a recomendar**: Motivação análoga àquela apresentada para a *feature* 29;

33. **Rácio entre o número de utilizadores que foram convidados a participar no evento a recomendar e o número de utilizadores que vão participar no evento a recomendar**: Motivação análoga àquela apresentada para a *feature* 31;

34. **Número de utilizadores que não vão participar no evento a recomendar**: Motivação análoga àquela apresentada para a *feature* 29;

35. **Rácio entre o número de utilizadores que não vão participar no evento a recomendar e o número de utilizadores que vão participar no evento a recomendar**: A motivação da presente *feature* é análoga àquela apresentada para a *feature* 31. Os valores do rácio determinado por esta *feature*, bem como os respectivos significados, são em tudo idênticos àqueles determinados para a *feature* 27, sendo agora o seu cálculo feito com base no universo de todos os utilizadores, e não com base no universo restrito de amigos do utilizador alvo de recomendação;

36. **Número de utilizadores do mesmo género que o do utilizador alvo de recomendação que vão participar no evento a recomendar**: Nas *features* 7 e 8 foi abordado o facto de os eventos serem sempre dirigidos a um determinado público-alvo, podendo este ser determinado de acordo com o género dos utilizadores que nele poderão vir a participar (utilizadores do sexo masculino, feminino, ou de ambos os sexos). Foi abordada também a importância do uso de informação referente ao género dos utilizadores como factor determinante no processo de recomendação de

eventos. Além desta informação, é também importante tentar perceber qual a popularidade de um dado evento junto do seu público-alvo. Tome-se, como exemplo, um dado evento futuro cujo público-alvo foi definido como sendo utilizadores do sexo masculino. Este evento será tão mais popular quanto mais intenções de participação tiver por parte de utilizadores do sexo masculino. Tomando esta informação em conta, considera-se que a probabilidade de um dado utilizador participar num dado evento futuro a recomendar será tanto maior quanto maior for o número de utilizadores do mesmo género que o do utilizador alvo de recomendação que vão participar no evento futuro a recomendar;

37. **Rácio entre o número de utilizadores que vão participar no evento a recomendar e o número de amigos do utilizador alvo de recomendação**: Além da informação descrita pelas *features* anteriores, é ainda possível perceber qual a dimensão do universo de utilizadores que vão participar num dado evento, em relação ao universo de amigos do utilizador. Trata-se, como tal, de mais uma medida de popularidade dos eventos, definida de acordo com a Equação 3.3:

$$\begin{cases} \text{Feature } 37 > 1 & \text{Popularidade elevada} \\ \text{Feature } 37 = 1 & \text{Popularidade neutra} \\ \text{Feature } 37 < 1 & \text{Popularidade reduzida} \end{cases} \tag{3.3}$$

A probabilidade de um dado utilizador frequentar um dado evento futuro é tanto maior quanto maior for o valor do rácio determinado por esta *feature*.

3.1.6 Dados de Similaridade

Este grupo compreende todas as *features* que explorem informação relativa à similaridade entre eventos.

Seja $E = \{e_1, e_2, e_3, ...\}$ o conjunto de todos os eventos a considerar. Cada evento $e_x \in E$ pode ser representado em termos de um vector $\overrightarrow{v}_{ws}$ de contagens de *word stems*[3]. Por forma a se determinar estes vectores $\overrightarrow{v}_{ws}$, começa-se por determinar as 100 *word stems* mais comuns (obtidas através de *Porter Stemming*), cuja ocorrência se dê no nome ou na descrição de um grande subconjunto aleatório de eventos $E_R \in E$. Após a obtenção dessas mesmas *word stems*, é então determinado, para cada evento $e_x \in E$, o respectivo vector $\overrightarrow{v}_{ws}$ de contagens de *word stems*, cuja estrutura é a indicada abaixo:

$$\overrightarrow{v}_{ws} = (\text{count}_1, \text{count}_2, \text{count}_3, \text{count}_4, ..., \text{count}_{100}) \tag{3.4}$$

Na equação, count_n, $n \in [1, 100]$, é um inteiro que denota o número de vezes que a n-ésima *word stem* mais comum aparece no nome ou na descrição do evento $e_x \in E$.

Após a obtenção, para cada evento $e_x \in E$, dos respectivos vectores $\overrightarrow{v}_{ws}$ de contagens de *word stems*, pode proceder-se então à determinação das *features* indicadas em seguida:

[3]O processo de extracção de *word stems* visa a remoção de quaisquer afixos de uma dada palavra, com vista à obtenção do seu radical (*stem*) [21]. Este processo torna possível a aglomeração, num mesmo grupo, de palavras distintas que tenham tido, na base da sua formação, o mesmo radical e, consequentemente, possuam significados semelhantes.

38. **Similaridade do cosseno entre o evento a recomendar e os eventos nos quais o utilizador alvo de recomendação participou no passado**: Por forma a determinar este valor de similaridade do cosseno, necessitamos primeiro de determinar:

    (a) $\overrightarrow{V}_{re}$, o vector de pesos TF-IDF correspondente ao evento a recomendar;

    (b) $\overrightarrow{V}_{ae}$, o vector de pesos TF-IDF correspondente a todos os eventos nos quais o utilizador alvo de recomendação participou no passado.

    Por forma a se determinar $\overrightarrow{V}_{re}$, necessitamos primeiro de obter o vector $\overrightarrow{v}_{ws}$ de contagens de *word stems* do evento a recomendar. Após a obtenção do vector $\overrightarrow{v}_{ws}$, procede-se então à determinação dos pesos TF-IDF, de acordo com as Equações 2.4, 2.5 e 2.6.

    Por forma a se determinar $\overrightarrow{V}_{ae}$, necessitamos primeiro de obter o vector $\overrightarrow{v}_{ws\_total}$ de contagens de *word stems* de todos os eventos nos quais o utilizador alvo de recomendação participou no passado. A obtenção do vector $\overrightarrow{v}_{ws\_total}$ é feita de acordo com o seguinte procedimento:

    (a) Obtenção dos vectores $\overrightarrow{v}_{ws}$ de contagens de *word stems* de todos os eventos nos quais o utilizador alvo de recomendação participou no passado;

    (b) Obtenção, a partir dos vectores $\overrightarrow{v}_{ws}$ obtidos anteriormente, de um vector $\overrightarrow{v}_{ws\_total}$, o qual apresenta a estrutura seguinte:

    $$\overrightarrow{v}_{ws\_total} = (\mathrm{count\_t}_1, \mathrm{count\_t}_2, \mathrm{count\_t}_3, \mathrm{count\_t}_4, ..., \mathrm{count\_t}_{100}) \tag{3.5}$$

    Na equação, $\mathrm{count\_t_n}$, $n \in [1, 100]$ é um inteiro que denota o número de vezes que a n-ésima *word stem* mais comum aparece no nome ou na descrição de todos os eventos nos quais o utilizador alvo de recomendação participou no passado, obtido de acordo com a Equação 3.6:

    $$\mathrm{count\_t_n} = \sum_{i=1}^{N} \mathrm{count_n}(i) \tag{3.6}$$

    Na Equação 3.6, N denota o número total de eventos nos quais o utilizador alvo de recomendação participou no passado, e $\mathrm{count_n}(i)$ denota o número de vezes que a n-ésima *word stem* mais comum aparece no nome ou na descrição do evento i.

A Figura 3.1 ilustra um exemplo simples da aplicação deste procedimento na determinação do vector $\overrightarrow{v}_{ws\_total}$ numa situação com $N = 2$. Após a obtenção do vector $\overrightarrow{v}_{ws\_total}$, procede-se então à determinação dos pesos TF-IDF, de acordo com as Equações 2.4, 2.5 e 2.6.

Após a determinação dos vectores $\overrightarrow{V}_{re}$ e $\overrightarrow{V}_{ae}$, pode então prosseguir-se para a determinação do valor de similaridade do cosseno entre o evento a recomendar e os eventos nos quais o utilizador alvo de recomendação participou no passado.

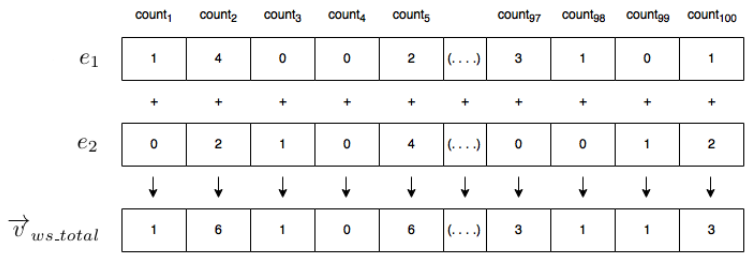

Figura 3.1: Exemplo da determinação do vector $\overrightarrow{v}_{ws\_total}$ numa situação com $N = 2$.

A probabilidade de um dado utilizador participar num dado evento a recomendar será, em princípio, tanto maior quanto maior for a similaridade do cosseno entre o evento a recomendar e os eventos nos quais o utilizador participou no passado;

39. **Similaridade do cosseno entre o evento a recomendar e os eventos nos quais o utilizador alvo de recomendação talvez tenha participado no passado**: Por forma a determinar o valor de similaridade do cosseno para esta *feature*, é empregue a mesma metodologia descrita na *feature* 38.

    A probabilidade de um dado utilizador participar num dado evento a recomendar será, em princípio, tanto maior quanto maior for a similaridade do cosseno entre o evento a recomendar e os eventos nos quais o utilizador talvez tenha participado no passado;

40. **Similaridade do cosseno entre o evento a recomendar e os eventos para os quais o utilizador alvo de recomendação foi convidado a participar no passado**: Por forma a determinar o valor de similaridade do cosseno para esta *feature*, é empregue a mesma metodologia descrita na *feature* 38.

    A probabilidade de um dado utilizador participar num dado evento a recomendar será, em princípio, tanto maior quanto maior for a similaridade do cosseno entre o evento a recomendar e os eventos para os quais o utilizador alvo de recomendação foi convidado a participar no passado;

41. **Similaridade do cosseno entre o evento a recomendar e os eventos nos quais o utilizador alvo de recomendação não participou no passado**: Por forma a determinar o valor de similaridade do cosseno para esta *feature*, é empregue a mesma metodologia descrita na *feature* 38.

    A probabilidade de um dado utilizador participar num dado evento a recomendar será, em princípio, tanto menor quanto maior for a similaridade do cosseno entre o evento a recomendar e os eventos nos quais o utilizador alvo de recomendação não participou no passado;

42. **Similaridade do cosseno entre o evento a recomendar e os eventos nos quais os amigos do utilizador alvo de recomendação participaram no passado**: Começa-se por determinar quais os amigos do utilizador alvo de recomendação. Seguidamente, para cada um dos amigos do utilizador alvo de recomendação, são obtidos os eventos nos quais estes participaram no passado.

É depois aplicada a mesma metodologia empregue na *feature* 38, por forma a determinar o valor de similaridade do cosseno para esta *feature*.

A probabilidade de um dado utilizador participar num dado evento a recomendar será, em princípio, tanto maior quanto maior for a similaridade do cosseno entre o evento a recomendar e os eventos nos quais os amigos do utilizador alvo de recomendação participaram no passado;

43. **Similaridade do cosseno entre o evento a recomendar e os eventos nos quais os amigos do utilizador alvo de recomendação talvez tenham participado no passado**: Por forma a determinar o valor de similaridade do cosseno para esta *feature*, é empregue a mesma metodologia descrita na *feature* 42.

    A probabilidade de um dado utilizador participar num dado evento a recomendar será, em princípio, tanto maior quanto maior for a similaridade do cosseno entre o evento a recomendar e os eventos nos quais os amigos do utilizador alvo de recomendação talvez tenham participado no passado;

44. **Similaridade do cosseno entre o evento a recomendar e os eventos para os quais os amigos do utilizador alvo de recomendação foram convidados a participar no passado**: Por forma a determinar o valor de similaridade do cosseno para esta *feature*, é empregue a mesma metodologia descrita na *feature* 42.

    A probabilidade de um dado utilizador participar num dado evento a recomendar será, em princípio, tanto maior quanto maior for a similaridade do cosseno entre o evento a recomendar e os eventos para os quais os amigos do utilizador alvo de recomendação foram convidados a participar no passado;

45. **Similaridade do cosseno entre o evento a recomendar e os eventos nos quais os amigos do utilizador alvo de recomendação não participaram no passado**: Por forma a determinar o valor de similaridade do cosseno para esta *feature*, é empregue a mesma metodologia descrita na *feature* 42.

    A probabilidade de um dado utilizador participar num dado evento a recomendar será, em princípio, tanto menor quanto maior for a similaridade do cosseno entre o evento a recomendar e os eventos nos quais os amigos do utilizador alvo de recomendação não participaram no passado

3.1.7 Casos Particulares

As situações descritas em seguida constituem alguns casos particulares aplicáveis às *features* abaixo indicadas:

- **Feature 1**: Nalguns casos, o cálculo dos valores de distância, em quilómetros, entre a morada de um utilizador e a localização de um evento não pode ser efectuado, em virtude dos seguintes factores:

    1. Informação acerca das localizações dos utilizadores inexistente;

2. Informação acerca das localizações dos eventos inexistente;
3. Inexistência de ambas as informações mencionadas nos pontos 1 e 2.

Nestas situações, procede-se ao cálculo do valor médio de distância, em quilómetros, entre a morada de um utilizador e a localização de um evento, sendo o mesmo aplicado a todos os pares utilizador-evento para os quais não foi possível determinar valores de distância geográfica determinados por esta *feature*;

- **Features 38, 39, 40, 41, 42, 43, 44, 45**: Nalguns casos, o cálculo dos valores de similaridade do cosseno determinados por cada uma das *features* acima mencionadas não é possível, em virtude dos seguintes factores:

  1. Inexistência de informação relativa ao evento a recomendar;
  2. Inexistência de informação relativa a todos os eventos frequentados pelo utilizador alvo de recomendação (*feature* 38) ou pelos amigos deste (*feature* 42);
  3. Inexistência de informação relativa a todos os eventos talvez frequentados pelo utilizador alvo de recomendação (*feature* 39) ou pelos amigos deste (*feature* 43);
  4. Inexistência de informação relativa a todos os eventos para os quais o utilizador alvo de recomendação (*feature* 40), ou os amigos deste (*feature* 44) foram convidados a participar;
  5. Inexistência de informação relativa a todos os eventos não frequentados pelo utilizador alvo de recomendação (*feature* 41) ou pelos amigos deste (*feature* 45);
  6. Inexistência de participações em eventos por parte do utilizador alvo de recomendação (*feature* 38) ou por parte dos amigos deste (*feature* 42);
  7. Inexistência de intenções de participação em eventos por parte do utilizador alvo de recomendação (*feature* 39) ou por parte dos amigos deste (*feature* 43);
  8. Inexistência de convites de participação em eventos dirigidos ao utilizador alvo de recomendação (*feature* 40) ou aos amigos deste (*feature* 44);
  9. Inexistência de recusas de participação em eventos por parte do utilizador alvo de recomendação (*feature* 41) ou por parte dos amigos deste (*feature* 45);

  Nestas situações, procede-se ao cálculo do valor médio de similaridade do cosseno para as *features* 38, 39, 40, 41, 42, 43, 44, 45. Os valores médios de similaridade do cosseno determinados são aplicados a todos os pares utilizador-evento onde se verifiquem as seguintes condições:

  1. O valor de similaridade do cosseno para alguma das *features* acima deve ser inexistente;
  2. O número de participações (para as *features* 38 e 42), intenções de participação (para as *features* 39 e 43), convites de participação (para as *features* 40 e 44), ou recusas de participação (para as *features* 41 e 45) deve ser nulo.

3.2 Modelo de Classificação

Uma vez criadas/implementadas todas as *features*, torna-se necessário o emprego de uma metodologia de classificação que delas possa fazer uso, por forma a gerar, para cada utilizador, a respectiva lista de eventos recomendados.

Várias metodologias de classificação foram testadas, tais como *Factorization Machines* (FMs) e *Random Forests*, tendo em vista a obtenção da melhor pontuação de recomendação possível (medida em termos de $\mathrm{MAP@200}$). De entre todas elas, aquela que se relevou mais eficaz quando aplicada no processo de recomendação de eventos foi a metodologia de classificação *Random Forest*, motivo pelo qual a mesma foi a escolhida por forma a gerar as listas de eventos recomendados, para cada utilizador considerado. Outro dos factores que também contribuíram para a escolha desta metodologia de classificação está relacionado com o facto de a mesma ter sido usada por um dos participantes do *Kaggle Event Recommendation Engine Challenge*[4], Andrei Olariu. Este mesmo participante apresentou-se como um dos melhores nesta competição da plataforma *Kaggle*, tendo terminado a mesma em 7º lugar.[5]. De modo semelhante, algumas das *features* descritas na Secção 3.1 do presente capítulo basearam-se naquelas idealizadas por Olariu, dado que este participante descreveu, em detalhe, a sua estratégia para a participação no *Kaggle Event Recommendation Engine Challenge*[6].

Esta metodologia de classificação supervisionada consiste de um *ensemble* de árvores de decisão. Assim, por forma a compreender melhor o funcionamento de um classificador *Random Forest*, torna-se necessário, em primeiro lugar, conhecer e compreender o funcionamento de uma árvore de decisão.

3.2.1 Árvores de Decisão

Autores como [5] abordam várias metodologias de classificação supervisionada, nas quais as árvores de decisão se encontram inseridas. Uma árvore de decisão apresenta-se como uma metodologia de classificação supervisionada, na qual a classificação de uma dada amostra é feita através da análise dos valores das *features* dessa mesma amostra. A Figura 3.2 [5] apresenta um exemplo simples de uma árvore de decisão. Cada nó da árvore representa uma *feature* numa dada instância objecto de classificação, sendo cada ramo da árvore representativo dos vários valores que cada nó da árvore pode assumir.

Por forma a construir-se uma árvore de decisão, é necessário, em primeiro lugar, encontrar a *feature* que melhor divida os dados de treino fornecidos, a qual irá integrar o nó-raiz da árvore de decisão. O mesmo procedimento é aplicado repetidamente, em cada nível da árvore de decisão, por forma a descobrir quais as várias *features* que melhor dividam os dados de treino num dado nível da árvore de decisão. Metodologias como o ganho de informação [22] ou o índice de Gini [23] são usadas por forma a descobrir as *features* que melhor dividam os dados de treino, em cada nível de uma árvore de decisão. O algoritmo C4.5 [24] configura-se como o mais conhecido algoritmo empregue na construção de árvores de decisão, o qual se apresenta como uma extensão ao algoritmo ID3 [25].

[4] https://www.kaggle.com/c/event-recommendation-engine-challenge
[5] Pontuação proveniente da tabela privada de pontuações (ver Secção 4.3.3 do Capítulo 4 do presente relatório).
[6] http://webmining.olariu.org/event-recommendation-contest-on-kaggle/

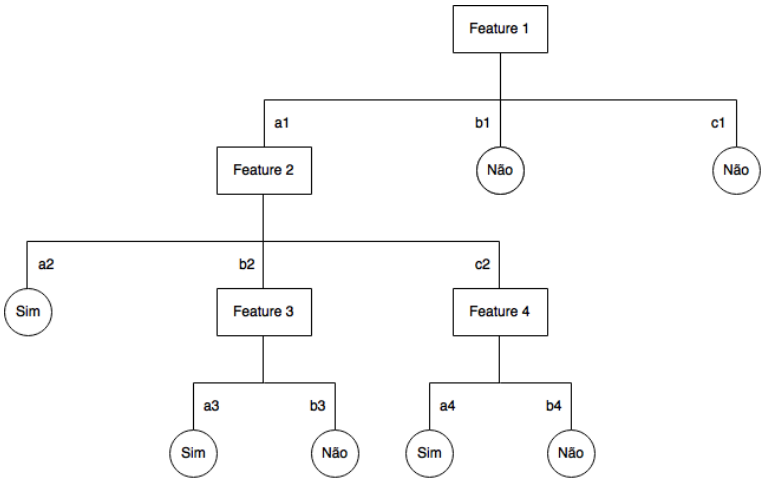

Figura 3.2: Um exemplo simples de uma árvore de decisão [5].

Uma das vantagens que as árvores de decisão apresentam sobre outros métodos de classificação supervisionada prende-se com o facto de a forma de classificação ser explícita. Contrariamente a outras metodologias de classificação supervisionada, em que o sistema final apresenta-se como uma ”caixa preta”, numa árvore de decisão é facilmente perceptível o porquê da atribuição de uma dada classe a uma dada instância.

3.2.2 Floresta Aleatória (*Random Forest*)

De acordo com [26], uma floresta aleatória (*Random Forest*) apresenta-se como uma metodologia de classificação supervisionada, a qual consiste num *ensemble* de árvores de decisão $\{h(\mathbf{x}, \Theta_k), k = 1, ...\}$, e onde $\{\Theta_k\}$ são vectores aleatórios i.i.d. que denotam subconjuntos aleatórios dos dados de treino originais, usados na construção de cada árvore de decisão constituinte do classificador *Random Forest*. A classificação é efectuada por meio de um mecanismo de votação, em que cada árvore constituinte do classificador *Random Forest* vota numa dada classe, sendo a classificação final atribuída a uma dada amostra correspondente à classe mais votada entre todas as árvores de decisão.

O classificador *Random Forest* empregue no motor de recomendação de eventos implementado no contexto da presente dissertação é aquele implementado na biblioteca *scikit-learn*[7]. Neste classificador *Random Forest*, cada árvore de decisão é construída de acordo com o seguinte procedimento:

1. A partir do conjunto de treino original, escolher um subconjunto aleatório de exemplos de treino, com reposição (*bagging* [27]);

2. A partir do subconjunto aleatório de exemplos de treino obtidos anteriormente, escolher um subconjunto aleatório de *features*;

3. Usar este subconjunto de exemplos de treino com um subconjunto aleatório de *features* para construir cada árvore de decisão.

[7] http://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

De uma forma simplificada, o processo de construção de um classificador *Random Forest* envolve o uso da técnica de *Bootstrap aggregating*, ou *Bagging* [27], em conjunto com selecção aleatória de *features*. O uso da técnica de *bagging* em conjunto com selecção aleatória de *features* possibilita a construção de classificadores *Random Forest* de maior exactidão [26].

Além da simplicidade do conceito associado aos classificadores *Random Forest*, uma outra vantagem que estes apresentam deve-se ao facto de que estes não apresentam *overfitting*, em virtude da Lei dos Grandes Números [26].

3.3 Arquitectura do Sistema

A arquitectura geral do sistema de recomendação de eventos implementado é a apresentada na Figura 3.3.

Numa primeira fase, procede-se ao treino de um classificador *Random Forest* através de um conjunto S de exemplos de treino na forma $(\mathbf{x}, y)$, onde $\mathbf{x} \in \Re^n$ é um vetor de 45 *features* que representa o utilizador e um item (evento) candidato, e onde y denota o alvo de predição. As 45 *features* contidas no vetor $\mathbf{x}$ encontram-se ordenadas da seguinte forma:

$$\mathbf{x} = [1, 38, 3, 39, 4, 40, 5, 41, 6, 42, 11, 43, 12, 44, 13, 45, 14, 15, 10, 16, 17, 18, \\ 19, 20, 21, 22, 23, 24, 25, 26, 27, 29, 30, 31, 32, 33, 34, 35, 28, 7, 8, 36, 37, 2, 9] \quad (3.7)$$

Após o treino do classificador *Random Forest*, um modelo de recomendação de eventos é gerado, sendo o mesmo aplicado no recomendador, por forma a gerar predições de recomendação de eventos para cada utilizador a considerar. As predições são obtidas a partir de um conjunto de exemplos de teste na forma de um vector de *features* $\mathbf{x}$, tal como apresentado anteriormente. Para cada exemplo de treino fornecido ao classificador *Random Forest* treinado, é retornado um valor de probabilidade $P \in [0, 1]$, o qual denota a probabilidade de um dado utilizador alvo de recomendação poder vir a frequentar um dado evento a recomendar.

Após a obtenção das listas de eventos recomendados para cada utilizador no conjunto de teste fornecido, torna-se necessário ordenar as predições de eventos por ordem decrescente de relevância. Desta forma, garante-se que os eventos que surjam nos primeiros lugares na lista correspondam, de facto, àqueles que o sistema de recomendação de eventos considerou serem os mais relevantes. A ordenação das listas de recomendação é feita com base nos valores de probabilidade retornados pelo classificador *Random Forest*.

3.4 Selecção de Features

Tal como tratado na Secção 3.1 do presente capítulo, o processo de criação/implementação de *features* configura-se como um dos mais importantes aspectos no processo de aprendizagem automática, uma vez que permite descrever cada amostra inserida nos conjuntos de treino ou de teste em função de um

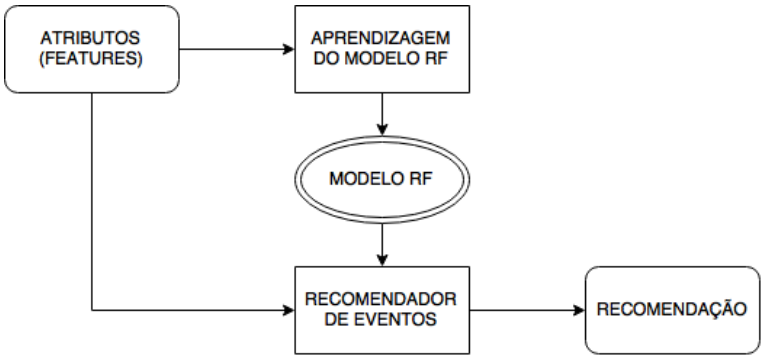

Figura 3.3: A arquitectura do sistema de recomendação de eventos implementado.

dado conjunto de atributos, por forma a possibilitar o seu uso por parte de algoritmos de classificação automática (em particular, de classificação supervisionada). De um modo geral, podem caracterizar-se *features* segundo 3 grupos [28]:

1. ***Features* relevantes**: São constituintes deste grupo todas as *features* que possuam influência sobre o resultado de classificação, não podendo o seu papel no processo de classificação ser assumido por outras *features*;

2. ***Features* irrelevantes**: Constituem este grupo todas as *features* que não possuam influência sobre o resultado de classificação;

3. ***Features* redundantes**: Constituem este grupo todas as *features* que possam tomar o papel de outras já existentes.

De acordo com esta caracterização, rapidamente nos apercebemos da importância do uso de *features* relevantes, em detrimento daquelas irrelevantes ou redundantes, as quais possuem pouco ou nenhum poder preditivo, não contribuem para o processo de aprendizagem automática e podem mesmo levar à introdução de ruído neste. Torna-se, como tal, imprescindível o emprego de metodologias no sentido de determinar quais as *features* relevantes para um dado problema de aprendizagem automática, descartando ao mesmo tempo *features* que possam ser irrelevantes ou redundantes. Este processo, denominado ”selecção de *features*”(*feature selection*) tem, por objectivo, servir este mesmo propósito, através da determinação de um subconjunto de tamanho reduzido de *features* relevantes, de entre todas as *features* existentes, as quais, quando aplicadas num processo de aprendizagem automática, possam levar à criação de sistemas de classificação mais eficazes e de melhor desempenho [28].

As vantagens do emprego de uma metodologia de selecção de *features* são as seguintes [28]:

1. Redução da dimensionalidade do espaço de *features*, tornando mais rápida a execução de algoritmos e facilitando a leitura dos dados;

2. Remoção de informação irrelevante, redundante, ou ruidosa;

3. Redução do tempo de execução dos algoritmos de aprendizagem empregues;

4. Aumento da qualidade dos dados;

5. Aumento da exactidão dos modelos resultantes;

6. Aumento de desempenho, com o correspondente ganho em termos de exactidão de predição;

Convém salientar que, idealmente, o processo de selecção de *features* passaria pelo teste de todas as combinações possíveis de *features* existentes, por forma a ter a certeza absoluta de qual o subconjunto de *features* relevantes para um dado problema de aprendizagem automática. Contudo, muitas das vezes, este processo mostra-se inviável devido à grande quantidade de *features* existentes (o número de combinações possíveis seria igual a $n!$, sendo n o número total de *features*).

Não obstante o problema apresentado anteriormente, existem diversas metodologias de selecção de *features* que permitem obter resultados bastante satisfatórios ou até mesmo próximos do resultado considerado óptimo. No contexto da presente dissertação, foi empregue a metodologia ***Sequential Forward Selection*** (ou **SFS**). Esta metodologia de selecção de *features* é explicada em maior detalhe em seguida.

3.4.1 Sequential Forward Selection (SFS)

O algoritmo SFS (*Sequential Forward Selection*) integra o grupo dos denominados *Greedy Search Algorithms*, sendo também o mais simples de entre eles. Uma descrição em pseudo-código deste algoritmo é apresentada no Algoritmo 1.

Algoritmo 1 Sequential Forward Selection (SFS)

Input: Conjunto de treino $\mathrm{S_{train}} = \{\mathbf{x}_i, y_i\}_{i=1}^{M}$, conjunto de teste $\mathrm{S_{test}} = \{\mathbf{x}_i\}_{i=1}^{N}$
Output: Conjunto de *features* seleccionadas $\mathrm{S_{sel\_features}} = \{\mathrm{F_1}, \mathrm{F_2}, ..., \mathrm{F_z}\}$

$\mathrm{S_{sel\_features}} \leftarrow \{\phi\}$
$\mathrm{best\_score} = 0$
for $i = 0 \rightarrow \mathrm{num\_features} - 1$ **do** ▷ $\mathrm{num\_features}$: Número total de *features*
    $\mathrm{F_{best}} = \mathrm{NULL}$
    for $\mathrm{n} = 0 \rightarrow \mathrm{num\_features} - 1$ **do**
        if $\mathrm{F_n}$ in $\mathrm{S_{sel\_features}}$ **then**
            goto 5
        end if
        $\mathrm{S_{feature\_subset}} \leftarrow \mathrm{S_{sel\_features}} + \mathrm{F_n}$
        Treinar modelo com base em $\mathrm{S_{train}}$ e $\mathrm{S_{feature\_subset}}$
        Testar modelo com base em $\mathrm{S_{test}}$ e $\mathrm{S_{feature\_subset}}$
        $\mathrm{score} = \mathrm{Score}\,(\mathrm{modelo})$
        if $\mathrm{score} > \mathrm{best\_score}$ **then**
            $\mathrm{best\_score} = \mathrm{score}$
            $\mathrm{F_{best}} = \mathrm{F_n}$
        end if
    end for
    if $\mathrm{F_{best}} \,!= \mathrm{NULL}$ **then**
        $\mathrm{S_{sel\_features}} \leftarrow \mathrm{S_{sel\_features}} \cup \mathrm{F_{best}}$
    else
        break
    end if
end for
return $\mathrm{S_{sel\_features}}$

Inicialmente, considera-se o conjunto de *features* relevantes como sendo o conjunto vazio. Por cada iteração do algoritmo SFS, é adicionada ao conjunto de *features* relevantes, de forma sequencial, a *feature* $\mathrm{F_x}$ que, quando combinada com as *features* já seleccionadas, maximize uma dada métrica de avaliação. Diferentes métricas de avaliação podem ser usadas, sendo aquela usada no contexto da presente dissertação a métrica $\mathrm{MAP@200}$ (ver a Secção 4.1.2 do Capítulo 4 do presente relatório).

A execução do algoritmo é interrompida quando (1) o número de *features* seleccionadas é igual ao número total de *features* existentes, ou (2) quando a máxima pontuação obtida na iteração actual seja menor que a máxima pontuação obtida na iteração anterior.

3.5 Sumário

Neste capítulo foram apresentados todos os aspectos relativos ao motor de recomendação de eventos implementado no contexto da dissertação a que o presente relatório se refere, nomeadamente:

- Quais as *features* implementadas, juntamente com uma explicação resumida de cada uma delas e a motivação respectiva;
- Qual o modelo de classificação empregue, por forma a gerar as listas de eventos recomendados para cada utilizador considerado;
- Qual a arquitectura do motor de recomendação de eventos implementado;
- A importância da selecção de *features* na obtenção de motores de recomendação de eventos eficazes, bem como o procedimento de selecção de *features* aplicado no caso concreto.

Capítulo 4

Avaliação

Neste capítulo são apresentados todos os aspectos relativos à avaliação da solução de recomendação de eventos implementada. Em particular, serão abordados os seguintes tópicos:

- Qual o *dataset* empregue, por forma a suportar a realização de experiências;
- Qual a métrica de avaliação usada para avaliar os resultados obtidos;
- Quais os resultados obtidos no decurso da avaliação do sistema de recomendação de eventos implementado.

Será feita também uma discussão dos resultados obtidos no decurso das experiências efectuadas. Por fim, serão sumarizados os aspectos abordados no presente capítulo.

4.1 Datasets e Métricas

A presente secção destina-se a fornecer uma pespectiva detalhada acerca do *dataset* usado para suportar a realização de experiências, bem como a métrica de avaliação usada para avaliar os resultados obtidos.

4.1.1 Dataset

Por forma a suportar a realização de experiências, foi usado o *dataset* do *Kaggle Event Recommendation Engine Challenge*[1], uma competição que teve lugar na plataforma Kaggle, em 2013. As estatísticas mais relevantes respeitantes a este *dataset* podem ser consultadas na Tabela 4.1, sendo que o conjunto de dados disponibilizado se refere a um total de 38209 utilizadores e 3137972 eventos.

Além dos conjuntos de treino e teste, este *dataset* é também composto pelos seguintes 4 subconjuntos de dados:

1. **users:** Subconjunto de dados com informação referente aos utilizadores;

[1] https://www.kaggle.com/c/event-recommendation-engine-challenge

2. **user_friends:** Subconjunto de dados com informação de cariz social referente aos utilizadores (i.e., para cada utilizador, quais os seus amigos);

3. **events:** Subconjunto de dados com informação referente aos eventos;

4. **event_attendees:** Subconjunto de dados com informação de participação em eventos por parte de utilizadores.

Tabela 4.1: Caracterização estatística do conjunto de dados a utilizar.

	Mínimo	Máximo	Média	Mediana
Número de amigos por utilizador	0	4964	795.41	481
Número de participações de utilizadores por evento	0	10000	34.42	16
Número de intenções de participação de utilizadores por evento	0	2500	21.57	12
Número de convites dirigidos a utilizadores por evento	0	9983	390.11	97
Número de recusas de participação de utilizadores por evento	0	1495	19.65	3
Número de participações em eventos por utilizador	1	252	1.28	1
Número de intenções de participação em eventos por utilizador	1	321	1.23	1
Número de convites de participação em eventos por utilizador	1	854	3.45	1
Número de recusas de participação em eventos por utilizador	1	524	1.84	1

4.1.2 Métrica de Avaliação

Uma boa metodologia de avaliação do trabalho realizado é importante para se poder aferir, com rigor, o desempenho e precisão da solução desenvolvida. Autores como [29] e [30] abordam a avaliação de sistemas de recomendação. No Capítulo 2 deste relatório foram apresentados trabalhos cujo foco incide na avaliação e comparação de sistemas de recomendação de eventos [7, 8].

A escolha de uma dada métrica (ou métricas) de avaliação deve ser feita de forma cuidadosa e de acordo com aquilo que se pretende medir. Neste caso concreto, por forma a avaliar o desempenho e precisão do motor de recomendação de eventos implementado, foi usada a métrica de avaliação *Mean Average Precision* na posição n ($\mathrm{MAP@n}$), com $n = 200$. A escolha desta métrica de avaliação foi decidida tendo em conta os seguintes aspectos:

- O objectivo final do motor de recomendação de eventos implementado consiste na geração, para cada utilizador considerado, das respectivas listas de eventos recomendados, ordenados por ordem decrescente de relevância. Desta forma, pretende-se que, para um dado utilizador, os eventos mais relevantes surjam nos primeiros lugares da lista, estando os eventos menos relevantes colocados no fim desta. Como tal, a qualidade de uma lista de eventos recomendados, para um

dado utilizador, é medida em termos da ordem pela qual os vários eventos recomendados surgem nesta, justificando-se, como tal, o emprego desta métrica de avaliação;

- Por ter sido a métrica empregue no decurso do *Kaggle Event Recommendation Engine Challenge*, permite a comparação dos resultados obtidos pelos participantes desta competição do Kaggle com aqueles obtidos através do motor de recomendação de eventos implementado no decurso da dissertação a que o presente relatório se refere.

Por forma a calcular o valor de $\mathrm{MAP@n}$, são primeiramente obtidos os valores de *Average Precision* na posição n ($\mathrm{AP@n}$), com $n = 200$, através do cálculo da média dos valores de precisão obtidos nas posições da lista das *top-n* recomendações em que um evento relevante foi recomendado, para um dado utilizador, de acordo com a Equação 4.1:

$$\mathrm{AP@n} = \frac{\sum_{k=1}^{n}[\mathrm{Precisão}(k) \times \mathrm{Relevância}(k)]}{\min(m, n)} \tag{4.1}$$

Na equação, $\mathrm{Precisão}(k)$ denota o valor de Precisão na posição k (também designado de $\mathrm{P@k}$), m denota o número total de eventos relevantes, e $\mathrm{Relevância}(k)$ é uma função que pode tomar os seguintes valores:

$$\mathrm{Relevância}(k) = \begin{cases} 1 & \text{, se o k–ésimo evento é relevante} \\ 0 & \text{, se o k–ésimo evento não é relevante} \end{cases} \tag{4.2}$$

Por fim, o valor de $\mathrm{MAP@n}$ é calculado à custa dos valores de $\mathrm{AP@n}$, de acordo com a Equação 4.3, em que $\mathrm{AP@n}_u$ denota o valor de $\mathrm{AP@n}$ para cada utilizador u, de acordo com a Equação 4.1, e onde N denota o número total de utilizadores.

$$\mathrm{MAP@n} = \frac{\sum_{u=1}^{N} \mathrm{AP@n}_u}{N} \tag{4.3}$$

4.2 Resultados

Na presente secção são apresentados os resultados obtidos com recurso ao motor de recomendação de eventos implementado. O procedimento de avaliação aplicado encontra-se dividido em duas fases distintas:

1. Numa primeira fase, procedeu-se à obtenção de resultados através da aplicação de todas as 45 *features* no processo de recomendação de eventos;

2. Numa segunda fase, procedeu-se à obtenção de resultados através da aplicação de um subconjunto de *features* relevantes, seleccionadas a partir do conjunto original de todas as 45 *features* implementadas, no processo de recomendação de eventos.

Todos os valores de $\mathrm{MAP@200}$ apresentados neste relatório foram obtidos com recurso à ferramenta trec_eval, uma ferramenta de avaliação da Text Retrieval Conference[2]. Seguidamente, são descritas, em maior detalhe, cada uma das fases constituintes do procedimento de avaliação adoptado.

4.2.1 1ª Fase: Utilização de todas as Features Implementadas

Numa primeira fase do processo de avaliação do motor de recomendação de eventos implementado, importa perceber qual a pontuação obtida (em termos de $\mathrm{MAP@200}$) decorrente da aplicação de todas as 45 *features* no motor de recomendação de eventos implementado. Tendo em vista a obtenção dos melhores resultados possíveis, foi empregue, no motor de recomendação de eventos implementado, um classificador *Random Forest* composto por 1400 árvores.

A partir da execução da presente experiência, foi possível obter um valor de $\mathrm{MAP@200} = 0.7059$.

4.2.2 2ª Fase: Selecção de Features

Tal como referido na Secção 3.4 do Capítulo 3 do presente relatório, as *features* podem classificar-se como (1) relevantes, (2) irrelevantes, ou (3) redundantes. Foi também abordada a importância do uso de metodologias de selecção de *features* como forma de seleccionar *features* consideradas relevantes, rejeitando todas as demais que possam ser consideradas irrelevantes ou redundantes. A selecção de *features* relevantes e a sua futura inclusão no processo de recomendação de eventos permitem obter melhores resultados que aqueles que seriam obtidos sem a aplicação destas metodologias.

Assim, numa segunda fase do processo de avaliação do motor de recomendação de eventos implementado, importa perceber quais as *features* verdadeiramente relevantes no processo de recomendação de eventos, bem como as pontuações obtidas (em termos de $\mathrm{MAP@200}$) decorrentes da aplicação das mesmas no motor de recomendação de eventos implementado. Tendo em vista estes objectivos, foi aplicado o seguinte procedimento de avaliação:

1. Começou-se por criar 45 subconjuntos de *features* a partir do conjunto original de todas as 45 *features* implementadas. Cada um dos 45 subconjuntos de *features* criados contém um número variável de *features*, de acordo com a seguinte disposição:

   (a) O 1º subconjunto contém a 1ª *feature* do conjunto original de *features*;

   (b) O 2º subconjunto contém a 1ª e 2ª *features* do conjunto original de *features*;

   (c) O 3º subconjunto contém a 1ª, 2ª e 3ª *features* do conjunto original de *features*;

   (d) (...)

   (e) O 44º subconjunto contém as primeiras 44 *features* do conjunto original de *features*;

   (f) O 45º subconjunto contém todas as 45 *features* implementadas.

   A ordem de inserção das *features* em cada um dos vários subconjuntos é feita de acordo com a ordem pela qual estas se apresentam no conjunto original de todas as 45 *features* implementadas;

[2] http://trec.nist.gov/

2. Após a geração de todos os 45 subconjuntos de *features* anteriormente descritos, procede-se à aplicação de um algoritmo de selecção de *features* SFS a cada um dos 45 subconjuntos de *features*. Deste modo, podem ser obtidas, para cada um dos 45 subconjuntos de *features* considerados, todas as *features* consideradas relevantes, bem como os valores de $\mathrm{MAP@200}$ referentes à aplicação das *features* seleccionadas no motor de recomendação de eventos implementado. O algoritmo de selecção de *features* SFS empregue nesta fase do processo de avaliação faz uso de um classificador *Random Forest* com 1200 árvores. Os resultados obtidos no decurso deste passo do procedimento de avaliação são apresentados na Tabela 4.2;

3. Após a conclusão do passo anterior, procede-se à escolha do subconjunto de *features* seleccionadas que der origem ao valor mais elevado de $\mathrm{MAP@200}$;

4. Partindo das *features* já seleccionadas no passo anterior, procede-se novamente à aplicação de um algoritmo de selecção de *features* SFS sobre todas as 45 *features* implementadas. Pretende-se desta forma seleccionar, de entre todas as *features* não incluídas no subconjunto de *features* seleccionadas no passo anterior, mais algumas *features* que possam ser consideradas relevantes pelo algoritmo, as quais são depois adicionadas ao conjunto de todas as *features* seleccionadas no passo anterior. Por forma a tentar encontrar o melhor valor possível de $\mathrm{MAP@200}$, o número de árvores do classificador *Random Forest* empregue pelo algoritmo de selecção de *features* SFS foi sendo alterado, variando entre as 600 e as 1600 árvores, em incrementos de 100 árvores. Os resultados obtidos no decurso deste passo do procedimento de avaliação são apresentados nas Tabelas 4.3 e 4.4;

5. Mais uma vez, procede-se à escolha do subconjunto de *features* seleccionadas que der origem ao valor mais elevado de $\mathrm{MAP@200}$;

6. Repetir novamente os passos 4 e 5 até que o valor máximo de $\mathrm{MAP@200}$ obtido na iteração actual seja inferior ao obtido na iteração anterior;

Por via da execução do procedimento de selecção de *features* descrito anteriormente, foram seleccionadas as seguintes *features*:

***Feature* 30:** Número de utilizadores que talvez participem no evento a recomendar;

***Feature* 29:** Número de utilizadores que vão participar no evento a recomendar;

***Feature* 10:** Diferença de tempo entre o instante no qual o evento a recomendar vai ter lugar e o instante no qual o utilizador alvo de recomendação visualizou o evento a recomendar no sistema;

***Feature* 45:** Similaridade do cosseno entre o evento a recomendar e os eventos nos quais os amigos do utilizador alvo de recomendação não participaram no passado;

***Feature* 20:** Rácio entre o número de amigos do utilizador alvo de recomendação que talvez participem no evento a recomendar e o número de amigos do utilizador alvo de recomendação;

***Feature* 1:** Distância, em quilómetros, entre a morada do utilizador alvo de recomendação e a localização do evento a recomendar;

***Feature* 22:** Número de amigos do utilizador alvo de recomendação que foram convidados a participar no evento a recomendar;

***Feature* 12:** Número de eventos nos quais os amigos do utilizador alvo de recomendação talvez tenham participado no passado;

***Feature* 17:** Número de amigos do utilizador alvo de recomendação que vão participar no evento a recomendar;

***Feature* 41:** Similaridade do cosseno entre o evento a recomendar e os eventos nos quais o utilizador alvo de recomendação não participou no passado;

***Feature* 13:** Número de eventos para os quais os amigos do utilizador alvo de recomendação foram convidados a participar no passado;

***Feature* 4:** Número de eventos nos quais o utilizador alvo de recomendação talvez tenha participado no passado;

***Feature* 2:** Se a cidade e país do utilizador alvo de recomendação são iguais à cidade e país do evento a recomendar;

***Feature* 26:** Rácio entre o número de amigos do utilizador alvo de recomendação que não vão participar no evento a recomendar e o número de amigos do utilizador alvo de recomendação;

***Feature* 14:** Número de eventos nos quais os amigos do utilizador alvo de recomendação não participaram no passado.

As mesmas foram depois aplicadas no motor de recomendação de eventos implementado, tendo o mesmo empregue um classificador *Random Forest* composto de 700 árvores (ver Tabela 4.3). Deste modo, foi possível obter um valor de $\mathrm{MAP@200} = 0.7475$.

4.3 Discussão

Na Secção 4.2 do presente capítulo, foram apresentados os resultados obtidos a partir do motor de recomendação de eventos implementado. Não obstante, apenas uma análise e discussão dos resultados obtidos permite aferir, com rigor, o desempenho e eficácia de recomendação da solução de recomendação de eventos implementada, face a outras soluções de recomendação de eventos existentes.

A análise e discussão dos resultados obtidos incidirá sobre os seguintes aspectos:

1. *Features* aplicadas no processo de recomendação de eventos;

Tabela 4.2: Selecção de *features* sobre cada um dos 45 subconjuntos de *features* obtidos a partir do conjunto original de todas as 45 *features* implementadas.

Número de *features* de cada subconjunto considerado	*Features* seleccionadas pelo algoritmo SFS	Valor de MAP@200 correspondente às *features* seleccionadas
1	1	0.4409
2	38,1	0.4840
3	38,1	0.4840
4	38,1,39,3	0.5189
5	38,1,4,3	0.5412
6	38,1,40	0.5467
7	38,1,40	0.5467
8	38,1,41,5	0.5540
9	38,1,41,5	0.5540
10	38,1,41,5	0.5540
11	38,1,41,5	0.5540
12	38,1,41,5	0.5540
13	38,1,41,5	0.5540
14	38,1,41,5	0.5540
15	38,1,41,5	0.5540
16	38,1,41,5	0.5540
17	38,1,41,5	0.5540
18	38,1,41,5	0.5540
19	10,1,41,43,42,11,13	0.6491
20	10,1,41,43,42,11,13	0.6491
21	10,1,17,41,15,6	0.6435
22	10,1,18,15,6,41,17	0.6409
23	19,4,39,3	0.5249
24	19,4,39,3	0.5249
25	19,4,39,3	0.5249
26	19,4,39,3	0.5249
27	19,4,39,3	0.5249
28	19,4,39,3	0.5249
29	19,4,39,3	0.5249
30	19,4,39,3	0.5249
31	19,4,39,3	0.5249
32	29,1,10,22,20,45,12,14,15,24,25	0.7245
33	**30,29,10,45,20,1,22,12,17,41,13,4**	**0.7357**
34	31,30,10,44,12,5,20	0.6941
35	31,30,10,44,12,5,20	0.6941
36	33,30,10,32,44,3,29,1,31,21,27,25	0.7227
37	33,30,10,32,44,3,29,1,31,21,27,25	0.7227
38	35,34,10,33,23,29,20,1,45	0.7193
39	35,34,10,33,23,29,20,1,45	0.7193
40	35,34,10,33,23,29,20,1,45	0.7193
41	35,34,10,33,23,29,20,1,45	0.7193
42	35,34,10,33,23,29,20,1,45	0.7193
43	35,34,10,33,23,29,20,1,45	0.7193
44	35,34,10,33,23,29,20,1,45	0.7193
45	35,34,10,33,23,29,20,1,45	0.7193

Tabela 4.3: Selecção de *features* a partir das *features* já seleccionadas anteriormente, com um número variável de árvores para o classificador *Random Forest* (1ª iteração). As novas *features* acrescentadas posteriormente pelo algoritmo SFS encontram-se assinaladas a vermelho.

Número de árvores do classificador *Random Forest* usado pelo algoritmo SFS	*Features* seleccionadas pelo algoritmo SFS	Valor de MAP@200 correspondente às *features* seleccionadas
600	30,29,10,45,20,1,22,12,17,41,13,4,2	0.7395
700	**30,29,10,45,20,1,22,12,17,41,13,4,2,26,14**	**0.7475**
800	30,29,10,45,20,1,22,12,17,41,13,4,2,31	0.7413
900	30,29,10,45,20,1,22,12,17,41,13,4,32,40	0.7414
1000	30,29,10,45,20,1,22,12,17,41,13,4,32,40	0.7415
1100	30,29,10,45,20,1,22,12,17,41,13,4,32,40	0.7409
1200	30,29,10,45,20,1,22,12,17,41,13,4,32,40	0.7412
1300	30,29,10,45,20,1,22,12,17,41,13,4,34,26	0.7436
1400	30,29,10,45,20,1,22,12,17,41,13,4,34,26,40	0.7451
1500	30,29,10,45,20,1,22,12,17,41,13,4,39	0.7331
1600	30,29,10,45,20,1,22,12,17,41,13,4,31,38	0.7371

Tabela 4.4: Selecção de *features* a partir das *features* já seleccionadas anteriormente, com um número variável de árvores para o classificador *Random Forest* (2ª Iteração). As novas *features* acrescentadas posteriormente pelo algoritmo SFS encontram-se assinaladas a vermelho.

Número de árvores do classificador *Random Forest* usado pelo algoritmo SFS	*Features* seleccionadas pelo algoritmo SFS	Valor de MAP@200 correspondente às *features* seleccionadas
600	30,29,10,45,20,1,22,12,17,41,13,4,2,26,14,40	0.7392
700	30,29,10,45,20,1,22,12,17,41,13,4,2,26,14,40	0.7406
800	30,29,10,45,20,1,22,12,17,41,13,4,2,26,14,36,6	0.7372
900	30,29,10,45,20,1,22,12,17,41,13,4,2,26,14,40	0.7438
1000	30,29,10,45,20,1,22,12,17,41,13,4,2,26,14,6,28	0.7442
1100	30,29,10,45,20,1,22,12,17,41,13,4,2,26,14,36,3	0.7425
1200	30,29,10,45,20,1,22,12,17,41,13,4,2,26,14,36	0.7442
1300	30,29,10,45,20,1,22,12,17,41,13,4,2,26,14,36,40	0.7451
1400	30,29,10,45,20,1,22,12,17,41,13,4,2,26,14,36,3	0.7450
1500	30,29,10,45,20,1,22,12,17,41,13,4,2,26,14,36,3,40	0.7460
1600	**30,29,10,45,20,1,22,12,17,41,13,4,2,26,14,36,3,40**	**0.7464**

2. A importância dos procedimentos de selecção de *features* na obtenção de bons resultados de recomendação;

3. Comparação de resultados face àqueles reportados pelos participantes do *Kaggle Event Recommendation Engine Challenge*.

Cada um dos três aspectos anteriores é desenvolvido em maior detalhe em seguida.

4.3.1 Features Aplicadas no Processo de Recomendação de Eventos

O tipo de *features* empregues no processo de recomendação de eventos constitui um dos primeiros aspectos que deve ser analisado e discutido no âmbito da avaliação de sistemas de recomendação de eventos. Dado que a qualidade e desempenho de recomendação estão estritamente dependentes

das *features* empregues, este constitui um dos aspectos mais importantes e relevantes no âmbito da avaliação de sistemas de recomendação de eventos, devendo ser analisado e discutido o quanto antes.

De acordo com os procedimentos experimentais efectuados no decurso da avaliação do motor de recomendação de eventos implementado, o máximo valor de $\mathrm{MAP@200}$ é atingido através do uso de um subconjunto de 15 *features*, seleccionadas de entre todas as 45 *features* implementadas. Da análise destas, percebe-se de imediato que as mesmas apresentam informação contextual de grande relevância no processo de recomendação, tais como a distância entre a morada do utilizador e a localização do evento, ou a similaridade do cosseno entre o evento a recomendar e os eventos nos quais os amigos do utilizador não participaram no passado.

No Capítulo 2, tinham sido já foram abordados trabalhos focados no problema de recomendação de eventos, os quais realçam a importância do uso de informação contextual no processo de recomendação de eventos, tendo em vista a obtenção de sistemas de recomendação de eventos de maior precisão e com desempenho de recomendação elevado [2, 13, 20, 8]. Assim, ao fazer uso, não só de informação referente aos utilizadores e aos eventos, mas também de informação contextual, o motor de recomendação de eventos implementado apresenta um desempenho e precisão de recomendação superiores àquilo que seria expectável caso este tipo de informação fosse descartada do processo de recomendação de eventos.

4.3.2 A Importância dos Procedimentos de Selecção de Features na Obtenção de Bons Resultados de Recomendação

Na Secção 3.4 do Capítulo 3 do presente relatório, foi abordada a temática da selecção de *features* e em como é imprescindível o emprego de metodologias no sentido de determinar quais as *features* relevantes para um dado problema de aprendizagem automática, descartando ao mesmo tempo *features* que possam ser irrelevantes ou redundantes. O uso, num sistema de aprendizagem automática, de *features* relevantes em detrimento das irrelevantes ou redundantes, permite a criação de sistemas de classificação mais eficazes e de melhor desempenho [28]. Como tal, um outro aspecto de grande importância que vale a pena analisar e discutir é o da selecção de *features*.

Tal como mencionado na Secção 3.1 do Capítulo 3 do presente relatório, um conjunto de 45 *features* foi implementado, tendo em vista não só a obtenção dos melhores resultados possíveis, como também a extracção da maior quantidade de informação possível dos dados disponíveis. Não obstante, algumas destas *features* configuram-se como irrelevantes ou redundantes, não contribuindo, como tal, no processo de recomendação de eventos. A execução do procedimento de selecção de *features* descrito na Secção 4.2.2 permite corrobar esta afirmação, dado que, do conjunto inicial de 45 *features*, apenas um subconjunto de 15 *features* foi seleccionado. Verificou-se também que a aplicação destas no processo de recomendação de eventos dá origem ao valor mais elevado de $\mathrm{MAP@200}$ registado. Fica, desta forma, demonstrada a importância da aplicação de metodologias de selecção de *features* em problemas de classificação automática no geral (e em problemas de recomendação de eventos, em particular).

4.3.3 Comparação de Resultados Face Àqueles Reportados Pelos Participantes do Kaggle Event Recommendation Engine Challenge

O *dataset* empregue, por forma a suportar a realização de experiências, foi aquele utilizado no contexto do *Kaggle Event Recommendation Engine Challenge* (ver Secção 4.1.1 do presente capítulo). No contexto desta competição, a qual decorreu na plataforma *Kaggle*, em 2013, participaram vários concorrentes, cada um dos quais desenvolveu a sua própria solução de recomendação de eventos. Assim, e tendo em consideração o facto de que o *dataset* empregue na presente dissertação é idêntico ao do *Kaggle Event Recommendation Engine Challenge*, bem como a métrica de avaliação usada na avaliação dos resultados obtidos ($\mathrm{MAP@200}$), um último aspecto que vale a pena referir é o da comparação dos resultados obtidos na dissertação a que o presente relatório se refere, face àqueles obtidos pelos participantes do *Kaggle Event Recommendation Engine Challenge*.

No decurso do *Kaggle Event Recommendation Engine Challenge*, duas tabelas de pontuações foram usadas: (1) uma tabela de pontuações pública (também designada de *Public Leaderboard*) e (2) uma tabela de pontuações privada (também designada de *Private Leaderboard*):

- No caso da tabela de pontuações pública (*Public Leaderboard*), os resultados apresentados foram obtidos com base em 197 dos 10237 exemplos de teste fornecidos;
- No caso da tabela de pontuações privada (*Private Leaderboard*), os resultados apresentados foram obtidos com base em $10237 - 197 = 10040$ exemplos de teste fornecidos, i.e., todos os exemplos de teste que não foram usados na obtenção das pontuações da tabela de pontuações pública. O vencedor desta competição foi também determinado com base nos resultados apresentados nesta tabela.

Na Tabela 4.5 são apresentados os resultados obtidos pelos participantes do *Kaggle Event Recommendation Engine Challenge* colocados nos 10 primeiros lugares. As pontuações nela exibidas correspondem àquelas exibidas na tabela de pontuações pública[3] disponível na plataforma *Kaggle*. Idealmente, a comparação de resultados seria feita com base nas pontuações exibidas na tabela de pontuações privada[4]. Contudo, dado que as políticas adoptadas pela plataforma *Kaggle* inviabilizam o fornecimento das soluções integrais dos *datasets* empregues em cada uma das competições realizadas nesta plataforma, a única forma de comparação de resultados será por via da tabela de pontuações pública, e não por via da tabela de pontuações privada, como seria desejável.
Foram igualmente usadas, nesta competição, duas diferentes *baselines* de comparação:

- Na primeira, denominada *Event Popularity Benchmark*, os eventos que são recomendados a cada utilizador encontram-se ordenados por popularidade, i.e., desde os eventos com maior quantidade de respostas positivas dos utilizadores, até aos eventos com menor número de respostas positivas por parte destes. A esta *baseline* está associado um valor de $\mathrm{MAP@200} = 0.49503$;

[3] https://www.kaggle.com/c/event-recommendation-engine-challenge/leaderboard/public
[4] https://www.kaggle.com/c/event-recommendation-engine-challenge/leaderboard/private

Tabela 4.5: Resultados obtidos pelos participantes do *Kaggle Event Recommendation Engine Challenge* colocados nos 10 primeiros lugares (tabela de pontuações pública).

Nome do participante	$\mathrm{MAP@200}$
DataLab	0.72876
uneventful	0.72809
srinivasan4u	0.72707
Chase & IntermediateLuck	0.71794
Andrei Olariu	0.71734
D3PO	0.71531
jsf	0.71447
Sergey	0.71117
n˙m & IM	0.70745
Montblanc	0.70728

- Na segunda, denominada *Random Order Benchmark*, os eventos são recomendados numa ordem aleatória. A esta *baseline* está associado um valor de $\mathrm{MAP@200} = 0.41938$.

O vencedor desta competição foi Josef Feigl (*jsf*), o qual obteve um valor de $\mathrm{MAP@200} = 0.73621$[5].

Com base nos resultados apresentados na Tabela 4.5, o participante *DataLab* foi aquele que obteve a pontuação mais elevada, tendo obtido um valor de $\mathrm{MAP@200} = 0.72876$. Uma comparação deste valor com aquele obtido com recurso ao motor de recomendação de eventos implementado no contexto da dissertação a que o presente relatório se refere, revela que a pontuação obtida por via deste é superior àquela obtida pelo participante *DataLab* ($\mathrm{MAP@200} = 0.7475 > 0.72876$). Fica, deste modo, demonstrada a eficácia e desempenho de recomendação da solução de recomendação de eventos implementada, face àquelas implementadas pelos participantes do *Kaggle Event Recommendation Engine Challenge*.

4.4 Sumário

Neste capítulo foram apresentados todos os aspectos relativos à avaliação do motor de recomendação de eventos implementado no contexto da dissertação a que o presente relatório se refere, nomeadamente:

- Qual o *dataset* empregue, por forma a suportar a realização de experiências;
- Qual a métrica de avaliação empregue, por forma a avaliar os resultados obtidos;
- Quais os resultados obtidos no decurso da avaliação do sistema de recomendação de eventos implementado.

Procedeu-se também a uma discussão dos resultados obtidos no decurso das experiências efectuadas.

[5] Valor obtido a partir da tabela privada de pontuações. Tal como referido neste relatório, foi a partir das pontuações constantes da tabela privada de pontuações que se determinou o vencedor desta competição. Os valores contidos na tabela pública de pontuações diferem daqueles contidos na tabela privada de pontuações. Se tivermos em conta as pontuações exibidas na tabela pública de pontuações, o mesmo participante obteve um valor de $\mathrm{MAP@200} = 0.71447$, tendo terminado em 7º lugar.

Capítulo 5

Conclusões e Trabalho Futuro

Neste capítulo serão apresentadas as principais conclusões decorrentes da implementação do motor de recomendação de eventos no contexto da dissertação a que o presente relatório se refere. Serão também lançadas as bases para um trabalho futuro no âmbito do motor de recomendação de eventos implementado.

5.1 Conclusões

O presente relatório abordou a temática dos sistemas de recomendação, e como estes são importantes no actual panorama da Internet, tanto para os utilizadores, como para as entidades que deles fazem uso. Foi igualmente abordado o problema da recomendação de eventos, e em como este novo problema de recomendação é diferente da recomendação tradicional. Foram apresentados os principais artigos da literatura na área, por forma a aferir, em maior detalhe, quais os problemas inerentes à recomendação de eventos, e quais as soluções adoptadas pelos diversos autores, com vista à resolução deste problema.

Com base na análise do trabalho relacionado, foi implementada uma solução de recomendação de eventos. Esta tem, por objectivo, providenciar, para cada utilizador tratado, uma lista de eventos recomendados, ordenados por ordem decrescente de relevância. O uso de vários tipos de informação no processo de recomendação, em particular, de (1) informação acerca dos utilizadores, (2) informação acerca dos eventos, e (3) informação contextual, bem como a natureza híbrida da solução de recomendação desenvolvida, conferem ao motor de recomendação de eventos implementado uma eficácia e desempenho de recomendação elevadas. Por via de uma análise exaustiva do conjunto de dados utilizado no suporte de realização de experiências, foram criadas 45 *features*, por forma a poder descrever, com o maior rigor e detalhe possíveis, cada par utilizador/evento tratado. Estas foram, por sua vez, aplicadas num classificador *Random Forest*, por via do qual as listas de recomendação, para cada utilizador considerado, foram geradas.

Por forma a aumentar ainda mais o rigor e desempenho de recomendação do motor de recomendação de eventos implementado, foi aplicada uma metodologia de selecção de *features* com base no algoritmo SFS, tendo em vista a selecção de todas as *features* consideradas relevantes, em detrimento da-

quelas consideradas irrelevantes ou redundantes. Por via da aplicação deste procedimento de selecção de *features*, foi possível obter um subconjunto de tamanho reduzido, composto por apenas 15 das 45 *features* originalmente concebidas, as quais foram depois aplicadas no processo de recomendação de eventos. Tal como esperado, o desempenho e precisão de recomendação resultantes aumentaram, em relação àqueles obtidos por via da aplicação de todas as 45 *features* implementadas no processo de recomendação de eventos e sem a aplicação de metodologias de selecção de *features*.

Foi, por fim, feita uma comparação dos resultados obtidos por via do motor de recomendação de eventos implementado com aqueles obtidos pelos participantes do *Kaggle Event Recommendation Engine Challenge*, dado que o *dataset* empregue na presente dissertação, bem como a métrica de avaliação utilizada na avaliação dos resultados obtidos ($\mathrm{MAP@200}$) são idênticas às utilizadas nesta competição da plataforma *Kaggle*. Através desta análise, demonstrou-se que a solução de recomendação de eventos implementada superou todas as outras implementadas pelos participantes desta competição da plataforma *Kaggle*.

5.2 Trabalho Futuro

Um dos aspectos que deve, por fim, ser analisado e tido em consideração, prende-se com o possível trabalho futuro a ser elaborado no contexto do motor de recomendação de eventos implementado. Este divide-se, fundamentalmente, em três partes:

1. Criação/implementação de outras *features*, além daquelas já implementadas e em uso pelo motor de recomendação de eventos;

2. Teste de outras metodologias de aprendizagem automática, além daquelas já testadas durante a implementação do motor de recomendação de eventos;

3. Emprego de outras metodologias de selecção de *features*, além daquela aplicada no contexto desta dissertação.

Na Secção 3.1 do Capítulo 3 do presente relatório, a criação/implementação de *features* constitui-se como um processo demorado, o qual requer uma análise exaustiva dos dados empregues, tendo em vista a extracção da maior quantidade de informação possível destes. Embora tenha sido criado, com base nesta análise, um conjunto de 45 *features*, não se garante que toda a informação relevante para o processo de recomendação tenha sido extraída dos dados utilizados. Neste aspecto, uma análise posterior dos dados empregues, tendo em em vista a implementação de *features* adicionais, constitui claramente um alvo de um eventual trabalho futuro.

Na Secção 3.2 do Capítulo 3 foi descrito o modelo de classificação utilizado na geração das listas de eventos recomendados, para cada utilizador considerado (modelo de classificação *Random Forest*). A escolha do mesmo teve em consideração o seu desempenho e eficácia de recomendação, face a outros modelos de classificação testados. Contudo, dada a quantidade e diversidade de modelos de classificação existentes, seria de todo incorrecto afirmar que os melhores resultados de recomendação,

no caso concreto do motor de recomendação de eventos implementado, se obtém por via de uma metodologia de classificação *Random Forest*. Para se ter a certeza absoluta acerca da melhor metodologia de classificação a aplicar, ter-se-iam de testar, de forma relativamente exaustiva, um grande conjunto de metodologias de classificação. Este constitui, como tal, um outro aspecto alvo de um eventual trabalho futuro.

Por fim, mas não menos importante, convém salientar as questões relacionadas com o processo de selecção de *features*. Na presente dissertação, foi aplicada uma metodologia de selecção de *features* com base no algoritmo SFS (ver Secção 4.2.2 do Capítulo 4 do presente relatório). Embora a metodologia de selecção de *features* descrita se tenha revelado eficaz, não podemos afirmar que se trate, de facto, da melhor metodologia de selecção de *features* existente, dada a grande quantidade e diversidade de metodologias de selecção de *features* existentes. Tal como o que sucede com a metodologia de classificação empregue na geração das listas de eventos recomendados, para cada utilizador considerado, apenas podemos ter a certeza acerca da melhor metodologia de selecção de *features*, no caso concreto da presente dissertação, através da aplicação e teste de várias outras metodologias de selecção de *features*. Este constitui, portanto, outro dos aspectos alvo de um eventual trabalho futuro.

Bibliografia

[1] D. Jannach and G. Friedrich. Tutorial: Recommender systems. In *Proceedings of the International Joint Conference on Artificial Intelligence*, 2013.

[2] A. Q. Macedo, L. B. Marinho, and R. L. Santos. Context-aware event recommendation in event-based social networks. In *Proceedings of the ACM Conference on Recommender Systems*, 2015.

[3] J. Bobadilla, F. Ortega, A. Hernando, and A. Gutiérrez. Recommender systems survey. *Knowledge-Based Systems*, 46(0), 2013.

[4] Y. Shi, M. Larson, and A. Hanjalic. Collaborative filtering beyond the user-item matrix: A survey of the state of the art and future challenges. *ACM Computing Surveys*, 47(1), 2014.

[5] S. B. Kotsiantis. Supervised machine learning: A review of classification techniques. In *Proceedings of the Conference on Emerging Artificial Intelligence Applications in Computer Engineering: Real Word AI Systems with Applications in eHealth, HCI, Information Retrieval and Pervasive Technologies*, 2007.

[6] E. Minkov, B. Charrow, J. Ledlie, S. Teller, and T. Jaakkola. Collaborative future event recommendation. In *Proceedings of the ACM International Conference on Information and Knowledge Management*, 2010.

[7] S. Dooms, T. De Pessemier, and L. Martens. A user-centric evaluation of recommender algorithms for an event recommendation system. In *Proceedings of the RecSys Workshop on Human Decision Making in Recommender Systems and User-Centric Evaluation of Recommender Systems and Their Interfaces*, 2011.

[8] A. Q. Macedo and L. B. Marinho. Event recommendation in event-based social networks. In *Proceedings of the International Workshop on Social Personalization*, 2014.

[9] D. Kang, D. Han, N. Park, S. Kim, U. Kang, and S. Lee. EVENTERA: Real-time event recommendation system from massive heterogeneous online media. In *Proceedings of the IEEE International Conference on Data Mining Workshops*, 2014.

[10] D. M. Blei, A. Y. Ng, and M. I. Jordan. Latent dirichlet allocation. *Journal of Machine Learning Research*, 3, 2003.

[11] T. Joachims. Optimizing search engines using clickthrough data. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 2002.

[12] Z. Qiao, P. Zhang, C. Zhou, Y. Cao, L. Guo, and Y. Zhang. Event recommendation in event-based social networks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2014.

[13] S. Rendle, Z. Gantner, C. Freudenthaler, and L. Schmidt-Thieme. Fast context-aware recommendations with factorization machines. In *Proceedings of the International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2011.

[14] A. Karatzoglou, X. Amatriain, L. Baltrunas, and N. Oliver. Multiverse recommendation: n-dimensional tensor factorization for context-aware collaborative filtering. In *Proceedings of the ACM Conference on Recommender Systems*, 2010.

[15] S. Rendle. Factorization machines with libFM. *ACM Transactions on Intelligent Systems and Technology*, 3(3), 2012.

[16] H. Khrouf and R. Troncy. Hybrid event recommendation using linked data and user diversity. In *Proceedings of the ACM Conference on Recommender Systems*, 2013.

[17] C. Bizer, T. Heath, and T. Berners-Lee. Linked data - the story so far. *International Journal on Semantic Web and Information Systems*, 5(3), 2009.

[18] T. D. Pessemier, S. Coppens, K. Geebelen, C. Vleugels, S. Bannier, E. Mannens, K. Vanhecke, and L. Martens. Collaborative recommendations with content-based filters for cultural activities via a scalable event distribution platform. *Multimedia Tools and Applications*, 58(1), 2012.

[19] Y. Zhang, H. Wu, V. S. Sorathia, and V. K. Prasanna. Event recommendation in social networks with linked data enablement. In *Proceedings of the International Conference on Enterprise Information Systems*, 2013.

[20] L.-H. Li, F.-M. Lee, Y.-C. Chen, and C.-Y. Cheng. A multi-stage collaborative filtering approach for mobile recommendation. In *Proceedings of the International Conference on Ubiquitous Information Management and Communication*, 2009.

[21] P. Willett. The porter stemming algorithm: then and now. *Program*, 40(3), 2006.

[22] E. B. Hunt, J. Marin, and P. J. Stone. *Experiments in Induction*. 1966.

[23] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone. *Classification and Regression Trees*. 1984.

[24] J. R. Quinlan. *C4.5: Programs for Machine Learning*. 1993.

[25] J. R. Quinlan. Discovering rules by induction from large collections of examples. In *Expert Systems in the Micro-Electronic Age*. 1979.

[26] L. Breiman. Random forests. In *Machine Learning*, 2001.

[27] L. Breiman. Bagging predictors. In *Machine Learning*, 1996.

[28] L. Ladha and T. Deepa. Feature selection methods and algorithms. *International Journal of Advanced Trends in Computer Science and Engineering*, 3(5), 2011.

[29] J. L. Herlocker, J. A. Konstan, L. G. Terveen, and J. T. Riedl. Evaluating collaborative filtering recommender systems. *ACM Transactions on Information Systems*, 22(1), 2004.

[30] G. Shani and A. Gunawardana. Evaluating recommendation systems. In F. Ricci, L. Rokach, B. Shapira, and P. B. Kantor, editors, *Recommender Systems Handbook*. 2011.