UNIVERSIDADE DO ALGARVE

Sistemas de Recomendação para o SocialDB

Tiago Miguel Cruz Torres

Mestrado em Engenharia Informática

Tese orientada por:
Professor Fernando Lobo

2013

UNIVERSIDADE DO ALGARVE

Sistemas de Recomendação para o SocialDB

Tiago Miguel Cruz Torres

Mestrado em Engenharia Informática

Tese orientada por:
Professor Fernando Lobo

2013

Sistemas de Recomendação para o SocialDB

Declaração de autoria de trabalho

Declaro ser o autor deste trabalho, que é original e inédito. Autores e trabalhos consultados estão devidamente citados no texto e constam da listagem de referências incluída.

Agradecimentos

À Direção Geral de Ensino Superior, pela bolsa de estudos que permitiu o meu desenvolvimento académico, na Licenciatura e no Mestrado.

À *Inesting*, pela colaboração e experiência, à Senhora Laurinda e ao Senhor Rui da *Inesting* pelo apoio dado.

Ao Professor Fernando Lobo, pela disponibilidade para ser o orientador e pelas sugestões apresentadas.

Aos meus colegas de Mestrado pelo companheirismo, à Denise e ao Mosab, pelos templates e a ajuda que me deram.

À Sandy pela paciência e pelo apoio dado nos piores e nos melhores momentos.

Ao meu pai e às minhas irmãs pela paciência e acompanhamento, à minha avó, além de tudo, por me ajudar a corrigir a tese.

À minha mãe, em especial, pelo carinho e dedicação dado ao longo da minha vida.

Muito Obrigado.

Resumo

O *SocialDB* é um *website* de caracter social, que dá a oportunidade às instituições sem fins lucrativos de se registarem no *website* a fim de receberem donativos. Quando um sponsor tem uma campanha de publicidade que quer promover, pode fazê-lo através do *SocialDB*. Essa campanha será distribuída a utilizadores que se enquadram no perfile de possíveis clientes. Quando um utilizador vê uma campanha, uma percentagem do valor pago pelo sponsor pela visualização dessa campanha, vai para uma instituição em forma de donativo.

Este trabalho pretende utilizar os *sistemas de recomendação* com a intensão de entender quais as preferências individuais dos utilizadors e com base nessas preferências dar recomendações das campanhas mais adequadas para esses utilizadores. Quantas mais campanhas forem visualizadas com um verdadeiro interesse por parte do utilizador, mais motivação haverá por parte dos sponsors para continuarem a anunciar as suas campanhas no *SocialDB*, o que leva à distribuição de mais donativos para as instituições sem fins lucrativos.

Palavras chave: Sistemas de Recomendação, Filtragem Colaborativa, Web Marketing.

Abstract

The *SocialDB* is a website with social awareness, that gives the opportunity to nonprofit institutions to register in the website to receive donations. When a sponsor has an advertising campaign to promote, he can do it through the *SocialDB*. This campaign will be distributed to users who are considered to be potential customers. When a user sees a campaign, a percentage of the amount paid by the sponsor to advertise this campaign goes to an institution as a donation.

This work intends to use *recommendation systems* with the purpose of understanding what are the preferences of the users, and based on those preferences give recommendations of campaigns that are more appropriated for those users. The more campaigns that are viewed with real interest of the user, the more motivation there will be for sponsors to continue advertising their campaigns in the *SocialDB*, which ends up resulting in more donations to nonprofits institutions.

Keywords: Recomendation Systems, Collaborative Filtering, Web Marketing.

Conteúdo

Lista de Figuras

Lista de Tabelas

Capítulo 1

Introdução

Os *sistemas de recomendação* são ferramentas que analisam os utilizadores ou itens num conjunto de dados, de forma a encontrar semelhanças e produzir uma recomendação com base na informação obtida.

Neste trabalho é proposto a utilização dos sistemas de recomendação para dar recomendações ao *website* desenvolvido pela empresa *Inesting*: o *SocialDB*. O *SocialDB* é um *website* que pretende criar uma plataforma de marketing com três tipos de registo.

- O registo dos sponsors para que possam anunciar campanhas de publicidade.

- O registo das intuições sem fins lucrativos para estarem aptas a receber donativos.

- O registo dos utilizadores que podem escolher da lista de instituições aquelas que querem apoiar e assim, quando recebem uma campanha de publicidade, uma parte do custo pago pelo sponsor para a visualização da sua campanha reverte para uma instituição escolhida pelo utilizador.

Quando um utilizador recebe uma campanha, vai ter a oportunidade de votar se a campanha recebida foi interessante para ele numa escala de 1 (não gostou nada) a 5 (gostou

muito) e cada campanha tem uma ou mais categorias. Assim, é possível identificar ao longo do tempo quais as categorias que agradam mais a um dado utilizador, sendo possível recolher logo algumas categorias de interesse do utilizador quando se regista, se o mesmo for feito através do *Facebook* e se o utilizador autorizar a visualização das páginas que fez *like* no *Facebook*.

Neste trabalho são propostos três tipos de recomendação. A primeira é para os novos utilizadores que fizeram o registo com acesso ao *Facebook*, assim é possível saber algumas categorias de interesse para o utilizador mas sem valor numérico associado a cada categoria, neste caso seria dada a recomendação de um valor numérico para cada categoria de interesse deste utilizador. A segunda recomendação seria para um utilizador com algum histórico de visualização de campanhas, e assim teria algumas categorias de interesse com valor numérico associado, então a recomendação a ser feita seria uma nova categoria que poderia ser potencialmente interessante para o utilizador. A terceira recomendação seria também para um utilizador com algum histórico de visualização de campanhas, mas neste caso era dada a recomendação de uma campanha que ainda não foi visualizada pelo utilizador mas que, pode vir a têr muito interesse para o mesmo.

1.1 Motivações

Uma das motivações que me levou ao desenvolvimento deste trabalho foi, porque ao realiza-lo teria a possibilidade de adquirir experiência profissional ao colaborar com uma empresa, mas também porque queria realizar um trabalho na área da *inteligência artificial*. Quando surgiu a oportunidade de desenvolver um projeto com a *Inesting*, chamado *The Social Database*, que consistia basicamente na construção de um *website* para distribuir campanhas publicitárias, tive a ideia de aplicar uma componente de inteligência artificial ao projeto e assim tentar melhorar os tipos de campanhas que um utilizador pode vir a visualizar. Assim, tive a oportunidade de realizar um trabalho na área de inteligência artificial e também adquirir uma maior experiencia profissional ao colaborar com a *Inesting*.

1.2 Objetivos e Limitações

O objetivo deste trabalho é ajudar no desenvolvimento de um *website* com a *Inesting*, a servir de plataforma para os sponsors anunciarem as suas campanhas, para os utilizadores receberem campanhas do seu interesse e ajudar instituições no processo, sendo que para as instituições é a oportunidade de receberem donativos para ajudar nas suas causas. Mas o objetivo principal é a realização de recomendações para os utilizadores.

O que é pretendido é tentar prever quais as categorias que os utilizadores mais gostam quando se registam pelo *Facebook*, tentar recomendar categorias que o utilizador ainda não mostrou interesse mas que podem ser do seu agrado, e também dar a recomendação de campanhas que um utilizador ainda não visualizou mas que pode vir a ser do seu interesse.

Infelizmente até à data da realização deste trabalho o *website* ainda não teve o seu lançamento comercial, impossibilitando o estudo dos utilizadores e a realização de recomendações, sendo necessário recorrer a dados simulados para representar os utilizadores do *SocialDB*. A simulação destes dados será explicada com mais detalhe na secção 4.1.

1.3 Resumo dos Capítulos

No Capítulo 2, será explicado como se desenvolveu a colaboração com a *Inesting* e as fases do projeto. Irá também ser explicado com mais detalhe o *SocialDB*, quais os seus objetivos, e quais as diferenças entre uma campanha e uma categoria. Ainda no Capítulo 2 será explicado que tipo de recomendações se pretende obter.

No Capítulo 3, será feito a revisão da literatura sobre os sistemas de recomendação, iniciando com uma descrição geral, mas com o intuito de aprofundar mais sobre a *filtragem colaborativa* que é um tipo de sistema de recomendação. Na filtragem colaborativa serão revistos alguns do algoritmos com *base na memória* e *baseados em modelos*.

No Capítulo 4, será explicado como podem ser interpretados os dados que vão ser utilizados para simular os dados do *SocialDB*. Será também explicado como será feita cada recomendação e os algoritmos utilizados para a aplicação dessas recomendações.

No Capítulo 5, será feita a análise dos dados recolhidos usando os métodos descritos no Capítulo 4, assim como uma comparação entre os algoritmos.

Por fim no Capítulo 6, será descrita a melhor forma de dar uma recomendação aos diferentes utilizadores com base no estudo efetuado, será também referido o que pode ser melhorado como trabalho futuro.

Capítulo 2

The Social Database

Neste capítulo será descrito o tipo de colaboração que houve com a empresa *Inesting* para o desenvolvimento do *website SocialDB*. Será explicado qual o papel dos sponsors, das instituições e dos utilizadores no *SocialDB*, qual a diferença entre campanhas e categorias e como será feita a sua avaliação. Por fim, será explicado o tipo de recomendações que podem ser dadas aos novos utilizadores e também a recomendação de categorias e de campanhas.

2.1 Colaboração com a Inesting

A ideia de realização de um projeto de tese com uma empresa surgiu uns meses antes de começar a procurar tópicos para a tese, numa palestra do *Best Engineering Week*, a decorrer na Universidade do Algarve em que a empresa *Inesting* era uma das participantes no evento. Durante a palestra foi referido que havia projetos de tese, o que despertou o meu interesse.

Algumas semanas mais tarde entrei em contacto com a *Inesting* para agendar uma possível reunião, a fim de obter mais informações sobre os projetos de tese possíveis, do que resultou uma reunião com o Senhor Rui Brás da *Inesting*. Um dos projetos apresentados foi o *SocialDB*, mas também referi que tinha interesse na área de inteligência artificial. Foi então que surgiu a ideia de criar recomendações para os utilizadores do *SocialDB* através de inteligência artificial, adicionando algo de novo ao projeto. Falei do

projeto ao Professor Fernando Lobo a quem pedi que fosse orientador da tese de Mestrado, tendo o mesmo concordado.

Passaram-se algumas semanas com alguns acertos entre a Universidade do Algarve e a *Inesting* de forma a poder realizar a tese com uma empresa. Quando finalmente se deu início ao desenvolvimento do *SocialDB*, ficou acordado a realização de reuniões semanais de forma a coordenar o trabalho realizado.

A Senhora Laurinda da *Inesting* foi a responsável pela minha aprendizagem em *ASP.NET MVC* (`http://www.asp.net/`) a plataforma utilizada no *Microsoft Visual Web Developer 2010 Express* para desenvolvimento *web*. Para além do desenvolvimento do *front end* do *website*, fiquei também responsável pela investigação e implementação de uma integração do site com a *API* do *Facebook* (`http://developers.facebook.com/web/`), tornando possível aos utilizadores fazer o registo através do *Facebook*, mas também recolher dados que podem ajudar a perceber as preferências dos utilizadores como as páginas em que fizeram *likes*. Depois do desenvolvimento do site e dos métodos de login, a Senhora Laurinda ficou responsável pelo *backoffice* de modo a poder focar-me mais na componente de inteligência artificial, mais especificamente, na aplicação dos sistemas de recomendação.

2.2 SocialDB

The Social Database é o nome do projeto desenvolvido pela Inesting e que tem como objetivo criar uma plataforma para três tipos de registo: os sponsors, as instituições e os utilizadores (Fig. A.1). As figuras no Apêndice A são facultativas e ilustram alguns exemplos das páginas do *SocialDB*.

2.2.1 Sponsors

Depois do sponsor fazer o pedido de registo (Fig. A.3), tem a possibilidade de começar a distribuir campanhas, para tal tem de escolher em quais categorias a sua campanha se encaixa, pode escolher o seu público-alvo com base na localização geográfica, se é masculino ou feminino, e a idade. As campanhas de publicidade serão enviadas por

e-mail para um dado número de utilizadores, dependendo do plafond gasto pelo sponsor.

2.2.2 Instituições

As instituições sem fins lucrativos podem candidatar-se a fazer parte do *SocialDB* através da página de registo (Fig. A.4). Quando uma instituição é aceite, tem de fornecer mais informações sobre a sua causa e os seus objetivos, pode também adicionar imagens que representem as suas causas (Fig. A.5). Com este registo efetuado, as instituições estão aptas a receber donativos que serão pagos por transferência bancária, cada vez que o saldo dessa instituição atingir um determinado valor.

2.2.3 Utilizadores

Os utilizadores podem fazer o registo no *SocialDB* (Fig. A.6) e se tiverem uma conta de *Facebook*, o registo pode ser ainda mais fácil, porque acede diretamente às informações do utilizador que estão no seu *Facebook*. Para além disso, são recolhidas informações adicionais como as páginas de *Facebook* que o utilizador gosta, de forma a tentar perceber os seus interesses. Quando se registarem podem escolher algumas instituições sem fins lucrativos, que gostariam de apoiar, dentro da lista de instituições disponíveis.

Quando o utilizador registado estiver apto a receber campanhas de publicidade por e-mail e cada vez que o utilizador visualizar uma campanha de e-mail, uma percentagem do valor pago pelo sponsor, para a visualização dessa campanha, vai para uma das instituições escolhidas pelo utilizador. Cada e-mail que recebe faz um ciclo entre as instituições que apoia de forma a dar contributo a todas as instituições apoiadas pelo utilizador.

2.3 Campanhas e Categorias

Quando um sponsor quer anunciar uma campanha, tem de escolher quais os tópicos da sua campanha, ou seja, as categorias. Essas categorias vão ser as mesmas categorias recolhidas pelo *Facebook* no acto de registo de um utilizador, de forma a obter-se um padrão no tipo de categorias existentes. Quando um utilizador recebe uma campanha

por e-mail, pode fazer uma avaliação da campanha numa escala de 1 (nenhum interesse) a 5 (interesse muito elevado) (Fig. A.7). Quando essa avaliação é feita, é adicionado à média da categoria da campanha visualizada, o valor da avaliação feita pelo utilizador, dado pela seguinte fórmula:

$$m_1 = \frac{nm + a}{n + 1} \tag{2.1}$$

onde m_1 é o novo valor médio, n é o número de campanhas visualizadas de uma dada categoria, m é o valor médio dado à categoria e a é a avaliação de uma campanha.

Por exemplo o utilizador U viu a campanha de e-mail X e deu uma avaliação de 5, e a campanha X tem como categoria Z. Imaginemos que o utilizador U já tinha avaliado 19 campanhas com a categoria Z, com uma média de avaliações de 4,3158. Neste caso o novo valor médio de avaliações para a categoria Z será 4,35.

2.4 Recomendação para o Utilizador

Neste trabalho é proposto três tipos de recomendação aos utilizadores do *SocialDB*: as recomendações para novos utilizadores, a recomendação de novas categorias de interesse e a recomendação de uma campanha. O objetivo será que quando um sponsor quer publicitar uma campanha, possa recorrer a estas recomendações para possivelmente obter clientes mais interessados nas suas campanhas.

Um exemplo seria arranjar 1000 utilizadores com a melhor média de avaliação para a categoria em causa. Ao só existirem 800 utilizadores com uma boa avaliação dessa categoria, poder-se-ia se recorrer aos sistemas de recomendação para tentar encontrar mais 200 utilizadores que provavelmente também iriam gostar de campanhas com este tipo de categoria. Outro exemplo, um sponsor que quisesse enviar uma campanha para 2000 utilizadores, podia escolher os primeiros 1000 com base em dados demográficos e idade e aguardar pela avaliação de alguns desses utilizadores. Recorrer-se-ia então aos sistemas de recomendação, para encontrar os outros 1000 utilizadores que ficariam potencialmente agradados com essa campanha.

2.4.1 Novos Utilizadores

Se um utilizador se registar através do *Facebook* será possível recolher categorias de interesse para esse utilizador, mas essas categorias não têm valor numérico associado. Ou seja, podem recolher-se as categorias de interesse através do *Facebook*, mas não se sabe quais dessas categorias o utilizador está realmente interessado e quais as que têm pouca relevância para o utilizador.

Com os sistemas de recomendação poderá ser feita essa avaliação e tentar perceber quais as categorias retiradas do *Facebook*, que têm realmente interesse para o utilizador, associando um valor numérico a cada categoria.

2.4.2 Recomendação de Categorias

Quando há falta de utilizadores com uma boa avaliação média de uma dada categoria, é possível recorrer aos sistemas de recomendação para encontrar utilizadores que nunca deram uma avaliação a essa categoria mas que se dessem, seria provavelmente uma boa avaliação.

2.4.3 Recomendação de Campanhas

Quando se trata de uma campanha, é possível procurar utilizadores que não viram essa campanha mas que se vissem, possivelmente, dar-lhe-iam uma boa avaliação.

2.5 Sumário

Neste capítulo foi explicado como se desenvolveu a colaboração com a *Inesting*. Qual o objetivo e os tipos de registos possíveis no *SocialDB*, assim como, qual o papel dos sponsors das instituições e dos utilizadores. Foi explicado quais as diferenças entre campanhas e as categorias e por fim, que tipo de recomendações vão ser dadas aos utilizadores.

No próximo capitulo será feita a revisão da literatura em sistemas de recomendação.

Capítulo 3

Revisão da Literatura

A revisão da literatura vai servir como base para a utilização dos sistemas de recomendação.

Neste capítulo será dada uma introdução aos sistemas de recomendação e algumas referências históricas. Serão apresentados diferentes tipos de sistemas de recomendação mas, o foco principal vai ser na filtragem colaborativa, onde será explicado algumas das suas características mas também dificuldades gerais. Serão também apresentados algoritmos com base na memória e algoritmos baseados em modelos.

3.1 Introdução aos Sistemas de Recomendação

Sistemas de recomendação consistem em técnicas para recolher informação sobre um utilizador e tentar prever quais são os seus interesses. Estas recomendações podem ser feitas através de várias técnicas, como estudar os padrões de um utilizador ou até comparar com outros utilizadores parecidos e dar recomendações com base nessas semelhanças.

A primeira aplicação de sistemas de recomendação conhecida foi *Grouplens* em 1992 e *Firefly* em 1994, mais tarde, o *Yahoo* e *Barnesandnoble* também passaram a utilizar a tecnologia do *Firefly*. Mas a ideia de filtragem colaborativa foi introduzida, em 1998, pelo vendedor de livros *Amazon.com* para um vasto número de pessoas, que mais tarde adaptaram o sistema de *BookMatcher* para outros itens [3].

Um fator importante a ter em conta com os sistemas de recomendação, é como reu-

nir a informação necessária para fornecer as recomendações para os utilizadores. Pode dividir-se a recolha de informação em dois grupos, recolha *explícita* e recolha *implícita* de dados. Na recolha explícita, o utilizador expressa a sua preferência em relação a um item, o que pode dar uma boa estimativa do que os utilizadores gostam mais, mas para este tipo de recolha de dados é necessária a cooperação do utilizador, o que pode não ser sempre a opção mais viável. A recolha implícita por outro lado tenta obter informação dos utilizadores de forma mais subtil. Pode ser feito com base no histórico de visualizações dos produtos ou compras dos mesmos, ou até através de um estudo do comportamento do utilizador. Os problemas deste tipo de recolha de informação, para além dos conflitos moraís, há também a possibilidade do utilizador estar a comprar por outra pessoa ou querer oferecer uma prenda a alguém [10].

3.2 Diferentes Sistemas de Recomendação

Nos sistemas de recomendação existem diversos tipos de filtragens. De seguida irão ser apresentadas algumas características destes tipos de filtragens [3].

Filtragem Colaborativa Compara os interesses e desinteresses de um utilizador com outros utilizadores para tentar prever as suas preferências, baseado nas avaliações subjetivas dos outros utilizadores. Um exemplo do sistema que usa a filtragem colaborativa é o *Amazon.com* onde um grande número de utilizadores registados, dão avaliações aos itens que compraram (com base nos utilizadores), mas também dão sugestões com base no perfil do utilizador (com base nos itens).

Filtragem Social É outra forma, ou por vezes apenas um sinónimo de filtragem colaborativa. Filtragem social é muitas vezes comparada a filtragem colaborativa com base nos itens.

Filtragem com base em Cliques É uma outra variante de filtragem colaborativa. Esta filtragem usa um grupo de pessoas que são semelhantes na forma de pensar como indicador, para as preferências do utilizador.

Filtragem Adaptativa É uma combinação entre filtragens com base nos utilizadores e com base nos itens. A ideia é que o sistema vai aprendendo ao longo do tempo, pedindo avaliações ao utilizador e monitoriza o seu comportamento e o que faz no sistema.

Filtragem com base em Características Centra-se na ideia de ser possível capturar as características do que um utilizador gosta e não gosta em relação a um item, podendo assim recomendar novos itens que tenham as características que o utilizador mais gosta.

Filtragem com base em Conteúdos Recomenda itens para os utilizadores com base na relação, entre os conteúdos de um item e as preferências do utilizador.

Filtragem com base em Palavras-chave É uma versão mais simples da filtragem com base em conteúdos. É limitada aos tipos de conceitos que podem ser expressos em termos de palavras-chave.

Filtragem de Perfil Esta é a filtragem mais simples. Os utilizadores simplesmente escolhem de uma lista de palavras-chave as que mais interessam e o sistema rejeita qualquer item que não tenha uma das palavras-chave escolhidas.

3.3 Filtragem Colaborativa

Filtragem colaborativa é a técnica mais utilizada nos sistemas de recomendação. A ideia base pretende fazer recomendações com base nas avaliações que os utilizadores deram aos itens [6]. Estas recomendações podem ser feitas com base nos utilizadores ou com base nos itens. As recomendações com base nos utilizadores representam os algoritmos de recomendações mais populares, devido a sua simplicidade e excelentes qualidades de recomendações, mas a complexidade computacional é incrementada linearmente com o número de utilizadores, o que pode gerar problemas de escalabilidade, enquanto que as recomendações com base nos itens, foram desenvolvidos para analisar a relação utilizador-item para identificar diferenças nas avaliações do itens de forma a dar recomendações aos

utilizadores [3, 6].

Um aspeto importante a considerar na filtragem colaborativa são os desafios que os algoritmos podem ter devido à falta de informação, porque normalmente um utilizador não avalia todos os itens num sistema. Outro problema é saber se o sistema consegue dar uma resposta em tempo útil ao utilizador, devido ao facto de poder ter muita informação para processar antes de dar uma recomendação ao utilizador. De seguida vão ser apresentados alguns exemplos de desafios para os algoritmos de filtragem colaborativa [1].

Dispersão de Dados Normalmente algoritmos de filtragem colaborativa são usados em sistemas com um grande número de itens. Os utilizadores que avaliam esses itens, por norma, apenas avaliam uma pequena percentagem de todos os itens disponíveis. Neste caso, podemos considerar a dispersão de dados como um desafio se não houver informação suficiente sobre um determinado utilizador. O mesmo acontece no *arranque frio*, que ocorre quando um novo utilizador ou um novo item é introduzido no sistema, gerando um desafio para a obtenção de recomendações, porque não existe informação sobre o mesmo.

Escalabilidade Quando o número de utilizadores e itens existentes crescem demasiado, os algoritmos de filtragem colaborativa podem sofrer graves problemas de escalabilidade, com os recursos computacionais a chegarem a níveis para além do aceitável. Este desafio é muito mais relevante se o sistema tiver de dar uma resposta imediata a um utilizador, o que acontece em muitos sistemas de compras online.

Sinónimos Refere-se à tendência de um número de itens que são similares ou iguais mas que têm nomes diferentes. Muitos sistemas não são capazes de reconhecer esta similaridade e acabam por tratar como se fossem diferentes.

Ovelha Cinzenta Quando um utilizador não concorda nem discorda com o resto dos utilizadores no sistema, é difícil proceder a alguma recomendação para este utilizador.

Ataque por Manipulação No caso onde qualquer pessoa pode dar uma avaliação, alguns comerciantes aproveitam este facto para dar avaliações positivas aos seus itens e avaliações negativas aos itens dos seus concorrentes.

Os algoritmos de filtragem colaborativa podem dividir-se em duas classes, os algoritmos com base na memória e os algoritmos baseados em modelos [1]. Nas próximas secções vão ser descritos alguns algoritmos de ambas as classes.

3.4 Algoritmos com base na Memória

Os algoritmos com base na memória usam a base de dados completa de forma a formalizar uma recomendação para os utilizadores. Cada utilizador faz parte de um grupo com interesses semelhantes. Muitos destes algoritmos tentam encontrar os utilizadores que são semelhantes ao *utilizador ativo* (utilizador a que se pretende dar uma recomendação) de forma a oferecerem previsões ou recomendações [10].

Vão ser apresentadas algumas técnicas representativas da filtragem colaborativa com base na memória.

3.4.1 Nearest Neighbor

O *Nearest Neighbor* é uma técnica de filtragem colaborativa que pode ser utilizada tanto para utilizadores como para itens e utiliza os seguintes passos: calcular o *fator de semelhança* ou alguma forma de peso de forma a relacionar dois utilizadores ou dois itens, escolher o *top-N* ou seja os N utilizadores ou itens mais semelhantes ao utilizador ou item ativo, e calcular a média ou uma média com pesos do *top-N* de forma a se obter uma recomendação para o utilizador ou item ativo [1].

3.4.2 Pearson Correlation

O *Pearson Correlation* pode ser utilizado para obter o fator de semelhança a ser utilizado no *Nearest Neighbor*. Para tal é calculado a correlação entre o utilizador ativo e um outro

utilizador, com a seguinte fórmula [10]:

$$w(a,i) = \frac{\sum_j (v_{a,j} - \bar{v}_a)(v_{i,j} - \bar{v}_i)}{\sqrt{\sum_j (v_{a,j} - \bar{v}_a)^2 \sum_j (v_{i,j} - \bar{v}_i)^2}} \tag{3.1}$$

onde o somatório de j são todos os itens que o utilizador a e o utilizador i deram alguma avaliação. $v_{a,j}$ é a avaliação que o utilizador a deu ao item j, e $\bar{v}_a$ é a média de todos os itens avaliados pelo utilizador a.

3.4.3 Default Voting

Default Voting é uma extensão para o algoritmo de *Pearson Correlation*. O seu objetivo é dar uma avaliação por defeito d aos itens que o utilizador ativo não deu avaliação mas o outro utilizador a ser comparado deu, ou vice-versa. Assim, no caso de um dos utilizadores ter poucas avaliações, pode ser feita uma comparação mais extensiva. Ou seja, se o conjunto de todos os itens avaliados por o utilizador ativo for I_a e o conjunto de todos os itens avaliados por um outro utilizador for I_j. Então usando o *Pearson Correlation*, são comparados itens em que ambos deram alguma avaliação ($I_a \cap I_j$) mas com o Default Voting, são comparados os itens na união dos utilizadores ($I_a \cup I_j$). É possível também assumir a mesma avaliação por defeito d para um número k de itens adicionais que nenhum dos utilizadores deu avaliação. Na maior parte dos casos, o valor por defeito d, reflete uma preferência neutra ou até negativa, para os itens não avaliados, utilizando a fórmula [10]:

$$w(a,i) = \frac{(n+k)(\sum_j v_{a,j}v_{i,j} + kd^2) - (\sum_j v_{a,j} + kd)(\sum_j v_{i,j} + kd)}{\sqrt{((n+k)(\sum_j v_{a,j}^2 + kd^2) - (\sum_j v_{a,j} + kd)^2)((n+k)(\sum_j v_{i,j}^2 + kd^2) - (\sum_j v_{i,j} + kd)^2)}} \tag{3.2}$$

onde o somatório em j vai até à união dos itens avaliados ($I_a \cup I_j$) e $n = |I_a \cup I_j|$.

3.4.4 Vector Similarity

Vector Similarity é um outro método de se calcular o fator de semelhança. Foi inicialmente desenvolvido para destacar a frequência de palavras entre dois documentos, tratando cada documento como um vetor com a frequência de cada apalavra.

Este princípio pode ser utilizado na filtragem colaborativa se um documento for considerado como um utilizador, uma palavra como um item e a sua frequência como uma avaliação, pode-se então utilizar a fórmula [10]:

$$w(a,i) = \sum_j \frac{v_{a,j}}{\sqrt{\sum_{k \in I_a} v_{a,k}^2}} \frac{v_{i,j}}{\sqrt{\sum_{k \in I_i} v_{i,k}^2}} \tag{3.3}$$

onde j é um item que o utilizador a e i deram alguma avaliação e $k \in I_a$ é o conjunto de itens que o utilizador a deu alguma avaliação e o termo ao quadrado no denominador serve para normalizar os votos, para que um utilizador que dê mais avaliações não seja à partida mais parecido do que outros utilizadores que fizeram menos avaliações.

3.4.5 Inverse User Frequency

Inverse User Frequency é uma extensão para o algoritmo *Vector Similarity*. A ideia é reduzir o peso dos itens mais comuns, porque normalmente os itens mais comuns não são tão uteis para capturar a semelhança de dois utilizadores como os itens menos comuns. É definido f_j como $\log \frac{n}{n_j}$ onde n_j é o número de utilizadores que votaram no item j e n é o número total de utilizadores. Se todos os utilizadores votarem no item j, então f_j é zero. Para aplicar o Inverse User Frequency ao Vector Similarity, basta multiplicar a avaliação original pelo fator f_i [1].

3.4.6 Cosine-Based Similarity

Cosine-based Similarity é um algoritmo com base nos itens e utiliza dois itens como se fossem dois vetores. A semelhança entre dois itens é calculada medindo o cosseno do angulo entre os dois vetores. Se m x n é a matriz utilizador-item, então a semelhança

entre dois itens, i e j é definido pela seguinte fórmula [1]:

$$w(a,i) = \cos(\vec{i},\vec{j}) = \frac{\vec{i} \bullet \vec{j}}{||\vec{i}|| * ||\vec{j}||} \tag{3.4}$$

onde o "$\bullet$" é denominado como o produto vetorial dos dois vetores. Por exemplo se o vetor $\vec{A} = \{x_1 + y_1\}$, e o vetor $\vec{B} = \{x_2 + y_2\}$, então a semelhança entre $\vec{A}$ e $\vec{B}$ é dado por:

$$w(A,B) = \cos(\vec{A},\vec{B}) = \frac{\vec{A} \bullet \vec{B}}{||\vec{A}|| * ||\vec{B}||} = \frac{x_1 x_2 + y_1 y_2}{\sqrt{x_1^2 + y_1^2}\sqrt{x_2^2 + y_2^2}} \tag{3.5}$$

3.4.7 Adjusted Cosine Similarity

O Adjusted Cosine Similarity tem algumas semelhanças com o Pearson Correlation, mas o Pearson Correlation com base nos utilizadores calcula a semelhança através das linhas da matriz. No caso de Adjusted Cosine Similarity que é com base nos itens, calcula a semelhança através das colunas da matriz, para dar o fator de semelhança entre o item i e o item j [14]:

$$w(i,j) = \frac{\sum_u (v_{u,i} - \bar{v}_u)(v_{u,j} - \bar{v}_u)}{\sqrt{\sum_u (v_{u,i} - \bar{v}_u)^2 \sum_u (v_{u,j} - \bar{v}_u)^2}} \tag{3.6}$$

onde o somatório de u refere-se a todos os utilizadores que deram alguma avaliação aos itens i e j, e $\bar{v}$ é o valor médio das avaliações feitas pelo utilizador u.

3.5 Algoritmos baseados em Modelos

Os algoritmos baseados em modelos usam estimativas e modelos matemáticos, que são então utilizados para dar as recomendações aos utilizadores. Estes modelos permitem a um sistema aprender e reconhecer padrões nos dados, melhorando a eficiência em sistemas que utilizam um grande número de utilizadores e itens, mas como desvantagem podem perder-se dados importantes no processo [1]. De seguida irá ser feita a referência a alguns dos algoritmos baseados em modelos.

3.5.1 Clustering

Um *Cluster* é uma coleção de objetos que são semelhantes entre eles no mesmo *cluster* e dissemelhantes de objetos em outros *clusters*. A medida de similaridade entre estes objetos pode ser medida com métricas como *Minkowski distance*, *Pearson Correlation* entre outras [1].

3.5.2 Bayesian Network

Um modelo probabilístico para a filtragem colaborativa é a *Bayesian Network*, com "nós" que correspondem a cada item no seu domínio. Depois de ser feita a aprendizagem da *Bayesian Network* para os dados de treino, o algoritmo procura por vários modelos em termos de dependência para cada item, de modo a originar uma rede em que cada item irá ter um conjunto de itens "pais" que são os melhores indicadores para uma recomendação [10].

3.5.3 Eigentaste

O *Eigentaste* é um algoritmo que utiliza o *Principal Components Analysis* [11] para redimensionar os dados de forma a criar um *cluster*. A partir daqui o *cluster* é dividido em quatro partes iguais, e para cada subdivisão que tem a origem como um dos seus vértices, divide-se em quatro partes outra vez, e continua-se a repetir até chegar a uma profundidade desejada.

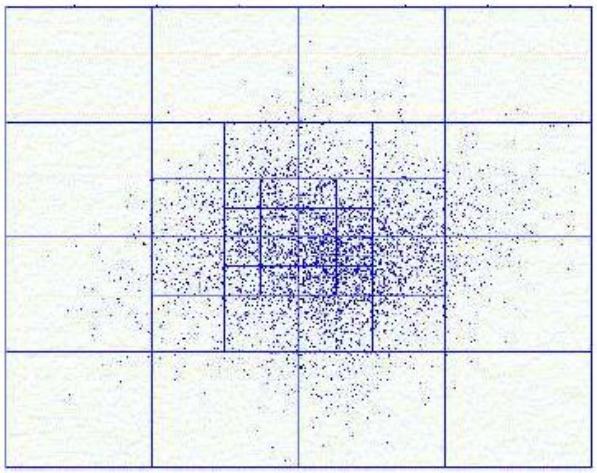

Figura 3.1: Cluster com 4 níveis de recursão, formando um total de 40 clusters [4].

Quando é necessário dar uma recomendação a um utilizador ativo, escolhem-se os utilizadores que estão no mesmo cluster, faz-se a média de todos os itens que o utilizador ativo não deu avaliação, e escolhe-se o que tem uma maior avaliação média para dar como recomendação ao utilizador ativo [4].

3.5.4 Slope One

O *Slope One* é um algoritmo baseado em modelos mas pode utilizar toda a base de dados de forma a chegar a uma recomendação, e assim não perde qualidade de recomendação como alguns outros algoritmos baseados em modelos, que usam estimativas para melhorar a sua performance. Para se chegar a uma recomendação, é calculada a diferença entre um item que queremos dar a recomendação e um segundo item de outro utilizador, depois é adicionada essa diferença ao "segundo" item do utilizador ativo para se obter a recomendação. Na Figura 3.2 é ilustrado um exemplo.

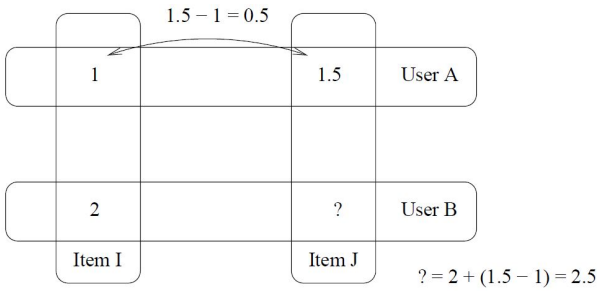

Figura 3.2: Exemplo de uma recomendação utilizando o *Slope One* [15]

Este exemplo é realizado apenas para um item secundário e um outro utilizador, mas o objetivo é fazer este cálculo para todos os itens a que o utilizador ativo deu alguma avaliação e para todos os utilizadores que deram avaliação ao item a ser recomendado e ao item secundário. Para calcular a diferença média entre dois itens é utilizado a fórmula [15]:

$$\bar{d}(i,j) = \frac{\sum_u v_{u,i} - v_{u,j}}{n(i,j)} \tag{3.7}$$

onde $\bar{d}(i,j)$ é a diferença média entre o item i e o item j, o somatório de u são todos os utilizadores que deram alguma avaliação aos itens i e j, $v_{u,i}$ é a avaliação que o utilizador u deu ao item i e $n(i,j)$ é o número de utilizadores que deram alguma avalição aos itens i e j. Agora é possível proceder a uma recomendação com a seguinte formula [15]:

$$r_{a,i} = \frac{\sum_j \bar{d}(i,j) + v_{a,j}}{\sum_j n(i,j)} \tag{3.8}$$

onde $r_{a,i}$ é a recomendação dada ao item i do utilizador ativo a e o somatório de j é para todos os itens que o utilizador ativo deu alguma avaliação.

Um problema com o *Slope One* é que o número de avaliações não é levado em consideração, assim é possível que um item tenha sido avaliado por 2000 utilizadores e outro tenha sido avaliado por apenas 20, mas ambos têm o mesmo peso para recomendação a efetuar. Mas com o *Weighted Slope One* é definido um peso de forma ter em conta itens

avaliados por mais utilizadores, o que pode ser visto na seguinte fórmula [15]:

$$r_{a,i} = \frac{\sum_j (\bar{d}(i,j) + v_{a,j}) n(i,j)}{\sum_j n(i,j)} \tag{3.9}$$

a única diferença em relação ao *Slpoe One* normal é o $n(i,j)$ que representa o número de utilizadores que avaliou os itens i e j. Outra variação do *Slope One* é o *Bi-Polar Slope One*, em que divide as avaliações que os utilizadores gostam e as avaliações que os utilizadores não gostam. A escolha de uma avaliação que se gosta ou não pode ser feita com o ponto médio, ou seja, se a escala de avaliações for de 0 a 10, então o ponto médio seria 5, assim os itens com avaliações superiores a 5 entravam na categoria do gosto e inferiores a 5 na categoria do não gosto. O problema é que por vezes as avaliações não são distribuídas equilibradamente, o 5 pode passar a ser um mau medidor de gosto e não gosto. Uma solução passa por usar como limite a média das avaliações dos utilizadores em vez de usar uma média global [15].

3.6 Sumário

Neste capítulo foi feita a revisão da literatura em que foi dada uma introdução aos sistemas de recomendação, dando ênfase à filtragem colaborativa. Por fim, foram descritos diversos algoritmos de filtragem colaborativa com base na memória e baseados em modelos.

No próximo capítulo será discutido como será feita a aplicação dos sistemas de recomendação ao *SocialDB*, assim como, quais os dados utilizados e os tipos de algoritmos que foram testados.

Capítulo 4

Aplicação de Sistemas de Recomendação

Neste capítulo será explicado como se procedeu à implementação dos sistemas de recomendação. Serão apresentados os dados simulados e de que forma representam as categorias e as campanhas do *SocialDB*. Será explicado o tipo de recomendações que foram feitas e quais os algoritmos escolhidos para cada tipo de recomendação. Por fim será explicado a aplicação do *Nearest Neighbor* e do *Slope One*, algoritmos utilizados para a realização das recomendações.

4.1 Dados para Teste

O *SocialDB* ainda não teve o seu lançamento comercial, impossibilitando a aplicação dos algoritmos aos dados do *website*, sendo necessário a simulação destes mesmos dados. Na prática vai ser necessário simular dois tipos de dados: as categorias e as campanhas referidas na secção 2.3.

4.1.1 Simulação das Categorias

Os dados necessários para as categorias são um ID de utilizador, um ID da categoria e o valor numérico associado a este utilizador e categoria, este valor será um número real

entre 1,0 e 5,0, assim, para a representação das categorias é necessário um conjunto de dados que tivesse um número total de itens relativamente pequeno (entre 50 a 200) e que as avaliações desses itens fossem números reais. Para fazer a simulação das categorias foi utilizado o *Jester dataset* (`http://eigentaste.berkeley.edu/dataset/`), que é um conjunto de dados que consiste em avaliações de piadas. Existem 150 piadas que os utilizadores podem ver e dar avaliações entre -10,0 e 10,0. Se considerarmos cada piada como se fosse uma categoria este conjunto de dados pode servir como simulação para as categorias do *SocialDB*. O ID de utilizador do *Jester* passa a ser um ID de utilizador do *SocialDB*, o ID de uma piada passa a ser o ID de uma categoria, e o valor da avaliação pode ser normalizado de -10,0 a 10,0 para 1,0 a 5,0.

4.1.2 Simulação das Campanhas

No caso da campanha é necessário um ID de utilizador, um ID de campanha e um número inteiro entre 1 e 5 que representa a avaliação do utilizador a essa campanha. Para a simulação das campanhas foi utilizado o *MovieLens dataset* (`http://www.grouplens.org/node/73`) em que os utilizadores dão avaliações entre 1 e 5 aos filmes visualizados de entre 4000 filmes. Se for considerado que cada filme representa uma campanha de e-mail (ou seja, existem 4000 campanhas de e-mail) que um utilizador pode avaliar, então o ID de utilizador no *MovieLens* passa a ser o ID de utilizador do *SocialDB*, o ID de um filme passa a ser o ID de uma campanha, e o valor da avaliação de um filme entre 1 e 5 mantem-se o mesmo e representa a avaliação a uma campanha.

4.2 As Recomendações Efetuadas

As recomendações que são necessárias para o *SocialDB*, não são muito afetadas pela escalabilidade do número de utilizadores. Claro que quantos mais utilizadores e itens, mais tempo demora para dar uma recomendação, mas estas recomendações não são dadas de imediato aos utilizadores, o que faz com que não seja tão importante a complexidade computacional de cada algoritmo. O mais importante neste caso é a qualidade das recomendações, e para tal foram escolhidos algoritmos que não sacrificam a qualidade das

recomendações por uma melhor performance.

A linguagem de programação utilizada para aplicar das recomendações foi o *Java*. A escolha efetuada não provém de nenhum requerimento necessário na programação das recomendações, foi apenas uma escolha pessoal.

Nas próximas secções vai ser explicado como serão aplicadas as recomendações referidas na secção 2.4, utilizando o *Nearest Neighbor* e o *Slope One*.

4.3 Nearest Neighbor

O primeiro passo na aplicação do *Nearest Neighbor* (referido na secção 3.4.1) é a escolha de quantos *vizinhos* (*Neighbors*) se quer encontrar, em que cada vizinho representa um utilizador que é de alguma forma semelhante ao utilizador ativo. Então é necessário definir o quão parecido um utilizador é do utilizador ativo e escolher quantos utilizadores parecidos queremos juntar, ou seja o número de vizinhos. Os testes realizados com *Nearest Neighbor* foram para 1, 10, 100 e 1000 vizinhos. Para se achar o fator de semelhança entre o utilizador ativo e outro utilizador foi utilizado o *Vector Similarity* para a recomendação de novos utilizadores e o *Pearson Correlation* para a recomendação de categorias e de campanhas.

Vector Similarity Para a recomendação realizada aos novos utilizadores é utilizado o *Vector Similarity* referido na secção 3.4.4, mas como os novos utilizadores não têm nenhum valor associado as suas categorias, é dado o valor de 1 para cada categoria do utilizador ativo. Para os outros utilizadores também é necessário fazer o mesmo, seja qual for o valor que deram a uma categoria o seu valor passa a 1. O que se pretende é que, se um utilizador tiver exatamente as mesmas categorias que o utilizador ativo então esse utilizadores tem um fator de semelhança de 1,0 (idênticos). Mas se um utilizador não tiver nenhuma das categorias que o utilizador ativo tem então passa a ter um fator de semelhança de 0,0 (totalmente diferentes). Dependendo do número de vizinhos, são então escolhidos os utilizadores mais semelhantes ao utilizador ativo com base no *Vector Similarity*.

	I_1	I_2	I_3	I_4	I_5
U_1	2,3		4,1		3,2
U_2	3,4	2,7		4,3	
U_3		1,2		3,6	4,6
A_1	?	?		?	

Tabela 4.1: Exemplo de recomendação para um novo utilizador

Utilizando *Vector Similarity* para o exemplo da Tabela 4.1, obtém-se um fator de semelhança entre o utilizador U_1 e o utilizador ativo A_1 de 0,3333, entre o U_1 e A_1 um fator de semelhança de 1,0 (tem as mesmas categorias de interesse que o utilizador ativo) e entre o U_3 e A_1 um fator de semelhança de 0,6667. Assim, são encontrados os utilizadores mais semelhantes ao utilizador ativo e depois serão feitas as recomendações para cada uma das categorias de interesse.

Pearson Correlation Para fazer a recomendação de uma categoria ou de uma campanha, foi utilizado o *Pearson Correlation* referido na secção 3.4.2. Foi ponderado a utilização do *Default Voting* referido na secção 3.4.3 mas devido à instabilidade dos resultados e à dificuldade de escolher o melhor valor por defeito para os diferentes tipos de recomendação [13], foi optado por usar apenas o *Pearson Correlation* que aparentemente tinha resultados mais estáveis. Quando um utilizador é comparado ao utilizador ativo usando o *Pearson Correlation* é dado um fator de semelhança entre -1,0 e 1,0 em que 1,0 significa que ambos os utilizadores têm as mesmas avaliações para os mesmos itens. Ao escolher o número de vizinhos, serão escolhidos os utilizadores com melhor fator de semelhança com base no *Pearson Correlation*.

	I_1	I_2	I_3	I_4	I_5
U_1	1,7	3,1	1,9		
U_2	4,3	2,9			4,1
U_3		4,1	2,9	3,8	
A_1	3,7	?	3,1	3,4	4,5

Tabela 4.2: Exemplo de recomendação para uma categoria

Utilizando o *Pearson Correlation* para o exemplo da Tabela 4.2, obtém-se um fator de semelhança entre U_1 e A_1 de 0,4927, um fator de semelhança de 0,5554 entre

U_2 e A_1 e um fator de semelhança de 0,7489 entre U_3 e A_1. Assim, é encontrado o fator de semelhança entre o utilizador ativo e os outros utilizadores.

Depois de serem encontrados os vizinhos usando o *Vector Similarity* ou o *Pearson Correlation*, é necessário proceder à recomendação para o utilizador ativo. Para se calcular uma recomendação é utilizado um sistema de pesos em que os pesos são o fator de semelhança, dado pela fórmula [9]:

$$r_j = \frac{\sum_i^n s_i a_{i,j}}{\sum_i^n s_i} \tag{4.1}$$

onde r é o valor da recomendação dada ao item j para o utilizador ativo, n é o número de vizinhos que deram alguma avaliação ao item j, s é o fator de semelhança calculado através do Vector Similarity ou do Pearson Correlation para o utilizador i e a é a avaliação dada ao item j pelo utilizador i. Esta fórmula é feita para cada item a ser recomendado ao utilizador ativo, que no caso de um novo utilizador, são todas as categorias que o utilizador tem interesse.

Se no exemplo da Tabela 4.1, forem escolhidos 2 vizinhos, então os utilizadores escolhidos seriam U_2, com um fator de semelhança de 1,0, e U_3 com um fator de semelhança de 0,6667. Para se efectuar as recomendações para os itens I_1, I_2 e I_3 do A_1, utiliza-se a fórmula 4.1, para os dois vizinhos mais semelhantes (U_2 e U_3), e obtém-se $I_1 = 3{,}4$, $I_2 = 2{,}1$ e $I_4 = 4{,}02$.

Se for feita a recomendação para o exemplo da Tabela 4.2, escolhendo os 2 vizinhos mais semelhantes, neste caso, seria escolhido o utilizador U_2 com um fator de semelhança de 0,5554 e o utilizador U_3 com um fator de semelhança de 0,7489. A recomendação efectuada ao item I_2 do utilizador ativo A_1, para os 2 vizinhos mais semelhantes (U_2 e U_3), com base na fórmula 4.1, é 3,589.

4.4 Slope One

O *Slope One* é um algoritmo com base nos itens como foi referido na secção 3.5.4, mais especificamente o *Weighted Slope One*, que foi o algoritmo utilizado para dar a recomen-

dação de uma campanha ou categoria. O *Slope One* não foi utilizado para a recomendação de novos utilizadores porque os novos utilizadores não têm nenhum valor associado às suas categorias, então não faria sentido utilizar um algoritmo que usa as avaliações prévias do utilizador ativo para chegar a sua recomendação. Foi utilizado o *Weighted Slope One* em vez do *Slope One* normal porque este usa um sistema de pesos para ter em conta o número de avaliações observadas e assim dar resultados mais equilibrados.

Não foi utilizado o *Bi-Polar Slope One* pela mesma razão que não foi utilizado o *Default Voting*. Encontrar um valor onde se pode dizer que o utilizador gosta ou não gosta pode ser subjetivo e este valor pode alterar conforme a situação, então com o *Bi-Polar Slope One* parecia que os resultados eram mais instáveis, e que o *Weighted Slope One* de modo geral apresentava melhores resultados. Assim o algoritmo escolhido para fazer os testes para as recomendações de categorias e campanhas foi o *Weighted Slope One*.

O processo para fazer a recomendação de um item é mais simples do que o processo utilizado no *Nearest Neighbor*, porque não é necessário escolher o número de vizinhos e a recomendação é feita diretamente ao item pretendido. Para a aplicação do *Weighted Slope One* a um utilizador ativo, remove-se um dos seu itens e tenta-se prever qual seria o valor que esse utilizador daria ao item removido utilizando o *Weighted Slope One*. Depois é comparado o valor da recomendação com o valor real de forma a obter uma estimativa do erro.

Se o *Slope One* for utilizado para dar a recomendação ao exemplo da Tabela 4.2, então é necessário calcular a diferença média entre o item que se pretende dar a recomendação e os outros itens que o utilizador ativo deu alguma avaliação, utilizando a fórmula 3.7. A diferença média entre o item I_2 e I_1 é 0,0 com base em 2 utilizadores, a diferença média entre I_2 e I_3 é 1,2 com base em 2 utilizadores, entre I_2 e I_4 é 0,3 com base em 1 utilizador e entre I_2 e I_5 é -1,2 com base em 1 utilizador.

$$\frac{((0{,}0+3{,}7)2)+((1{,}2+3{,}1)2)+((0{,}3+3{,}4)1)+((-1{,}2+4{,}5)1)}{2+2+1+1} = 3,8333$$

Utilizando a fórmula 3.9, obtém-se a recomendação do I_2 para o utilizador ativo A_1 de 3,8333.

4.5 Sumário

Neste capítulo foi explicado de que forma os dados do *Jester* e do *MovieLens*, podem ser simulados como categorias e campanhas para o *SocialDB*. Foi explicado quais os algoritmos que foram utilizados para cada tipo de recomendação e como foi aplicado o *Nearest Neighbor* e o *Slope One*.

No próximo capítulo vão ser discutidos os resultados obtidos para cada tipo de recomendação.

Capítulo 5

Análise de Resultados

Para cada algoritmo testado, foi baralhado o conjunto de dados e foi dividido em dez partes equivalentes em que a primeira parte servia como conjunto de dados a testar e o resto servia como conjunto de treino. Depois de obtido o resultado, era usada a segunda parte como conjunto de teste e o resto como conjunto de treino, e assim sucessivamente, até à obtenção dos resultados de cada uma das dez partes, como conjunto de teste. Este processo é chamado de *10 fold cross-validation*. Dos dez resultados obtidos é feita a média e obtém-se a primeira tentativa. Volta a repetir-se este processo até se obterem dez tentativas. Correndo cada algoritmo testado num total de 100 vezes, usando *10 times 10 fold cross-validation* [16]. As tabelas com os resultados de cada uma das tentativa podem ser consultadas no Apêndice B.

A métrica de erro utilizado para a análise de resultados foi o *Mean Absolute Error*, em que dá em valor absoluto o erro entre uma avaliação prevista por uma algoritmo e a avaliação real de um utilizador, em que a avaliação real é retirada ao utilizador no decorrer do algoritmo, e no fim é comparado com a recomendação efetuada de forma a dar uma estimativa do erro.

Nas secções seguintes serão analisadas as recomendações para os novos utilizadores, a recomendação de uma categoria e a recomendação de uma campanha.

5.1 Recomendação de Categorias para Novos Utilizadores

Para a recomendação feita para os novos utilizadores foi utilizado o *Nearest Neighbor* com 1, 10, 100, 1000 vizinhos. Na Figura 5.1 pode-se observar que o menor erro é obtido quando o número de vizinhos é 10, isto porque quando é usado o *Vector Similarity* os vizinhos que têm as categorias de interesse mais parecidas vão ser os que têm o melhor fator de semelhança. Assim, parece que escolhendo apenas 1 vizinho seria a melhor solução porque era o utilizador mais parecido ao utilizador ativo, mas na realidade pode ser pior porque esse utilizador pode apenas ter tido sorte e parecer idêntico mas não o ser, ou pode dar uma recomendação muito alta a um item que o resto dos vizinhos mais parecidos deram uma recomendação baixa, ou vice-versa [9].

Assumindo que o resultado das experiências seguem uma distribuição normal, podemos ver que na Tabela 5.1 que o melhor erro médio tem o valor de 0,7171 com um intervalo de confiança de 95% entre 0,7082 a 0,726. Isto quer dizer que se voltasse a repetir o teste dos novos utilizadores, baralhando completamente o conjunto de dados, dividindo em dez partes e utilizar uma das partes como conjunto de teste e o resto como conjunto de treino, e se utilizasse 10 vizinhos, o erro médio iria encontrar-se entre 0,7082 e 0,726 com 95% de confiança.

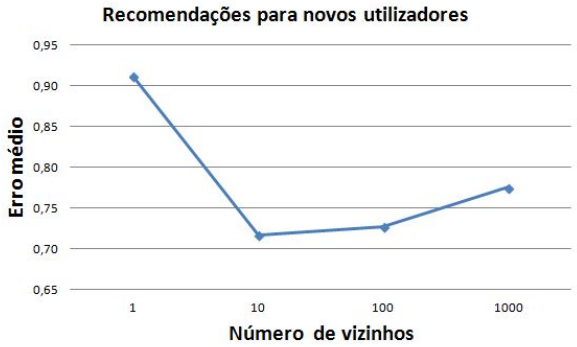

Figura 5.1: Erro médio da recomendação para novos utilizadores para diferentes números de vizinhos

	Erro médio	Int. Confiança 95%	
1 vizinho	0,9117	0,9013	0,9221
10 vizinhos	**0,7171**	**0,7082**	**0,726**
100 vizinhos	0,7277	0,7184	0,737
1000 vizinhos	0,7754	0,7655	0,7853

Tabela 5.1: Erro médio e intervalo de confiança para os algoritmos utilizados com os novos utilizadores

5.2 Recomendação de Categorias para Utilizadores Existentes

Para as recomendações de categorias é utilizado o *Nearest Neighbor* com 1, 10, 100, 1000 e também o *Slope One*. Na Figura 5.2 dá para reparar numa alteração em relação à Figura 5.1, agora o erro mais baixo encontra-se quando o número de vizinhos é 100. Neste caso o fator de semelhança foi obtido através do *Pearson Correlation* e é dado apenas uma categoria como recomendação. Quando se escolhe apenas 1 vizinho o erro normalmente é grande. Quantos mais vizinhos forem utilizados para chegar a uma recomendação, menor será o erro até chegar a um ponto em que o erro volta a aumentar porque se aproxima cada vez mais da média geral, em vez de ser uma seleção de vizinhos semelhantes ao utilizador ativo e neste caso o número ideal de vizinhos é a volta dos 100.

Agora também foi utilizado o *Slope One* e como se pode ver, na Tabela 5.2 este algoritmo obtém um erro médio de 0,6663 com uma confiança de 95% entre 0,6495 a 0,6831. Tendo um melhoramento considerável em comparação ao *Nearest Neighbor*.

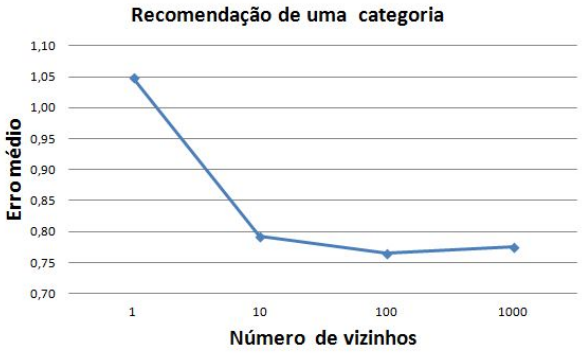

Figura 5.2: Erro médio da recomendação de uma categoria para diferentes números de vizinhos

	Erro médio	Int. Confiança 95%	
1 vizinho	1,0489	1,0233	1,0745
10 vizinhos	0,7922	0,7737	0,8106
100 vizinhos	0,765	0,7476	0,7823
1000 vizinhos	0,7751	0,7577	0,7925
Slope One	**0,6663**	**0,6495**	**0,6831**

Tabela 5.2: Erro médio e intervalo de confiança para os algoritmos utilizados para a recomendação de uma categoria

5.3 Recomendações de Campanhas

Os resultados para as recomendações de campanhas são bastante parecidas às recomendações de categorias, visto que os algoritmos utilizados foram os mesmos e que a única diferença é que no caso das campanhas, existem muitos mais itens por utilizador do que nas recomendações de categorias. O objetivo era mesmo esse, ver qual o desempenho dos algoritmos quando existem muitos mais itens que podem ser recomendados, neste caso as campanhas.

Pela Tabela 5.3 pode ver-se que o erro médio tem o mesmo comportamento do que as recomendações para as campanhas, mas com uma média de erro um pouco superior, devido a haver um número maior de itens, ou seja, uma maior dispersão de dados. Mas pode ver-se que mais uma vez o menor erro médio obtido foi com o *Slope One*, com um erro médio de 0,7298 com uma confinaça de 95% entre 0,6845 a 0,7752.

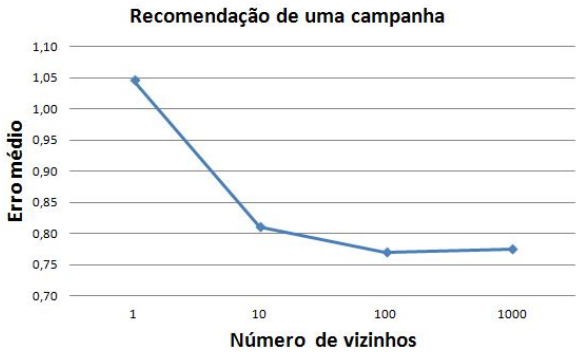

Figura 5.3: Erro médio da recomendação de uma campanha para diferentes números de vizinhos

	Erro médio	Int. Confiança 95%	
1 vizinho	1,0464	0,9711	1,1217
10 vizinhos	0,8112	0,7618	0,8606
100 vizinhos	0,7704	0,7235	0,8172
1000 vizinhos	0,7754	0,7286	0,8222
Slope One	**0,7298**	**0,6845**	**0,7752**

Tabela 5.3: Erro médio e intervalo de confiança para os algoritmos utilizados para a recomendação de uma campanha

Capítulo 6

Sumário e Conclusão

Como finalização da dissertação, irá ser apresentada a revisão dos capítulos, as conclusões dos resultados obtidos e o que poderá vir a ser melhorado no futuro.

6.1 Revisão dos Capítulos

Nesta dissertação foi dada uma introdução ao tema e quais as motivações que levaram à realização deste projeto. Quais os objetivos a realizar e as limitações provenientes da falta de dados para a realização de testes. Foi falado de como se procedeu à colaboração com a *Inesting*, o que é o *SocialDB* e quais os tipos de registos que se pode efetuar. As diferenças entre as categorias e as campanhas e o tipo de recomendações que se podem obter. Foi feita a revisão da literatura em que foi falado dos sistemas de recomendação, mas com ênfase na filtragem colaborativa. Dentro da filtragem colaborativa foram referidos algoritmos com base na memória e algoritmos baseados em modelos e foram apresentados alguns algoritmos de ambas as classes. Foram apresentado os dados usados para simular as categorias e as campanhas do *SocailDB* e foi explicado como foi feita a aplicação do *Nearest Neighbor* e do *Slope One*, para as recomendações pretendidas. Por fim, foi explicado como foram testados os algoritmos e o tipo de erro obtido para a comparação de desempenho.

6.2 Conclusão

Quando os sistemas de recomendação forem aplicados ao *SocialDB*, os utilizadores que se registrarem através do *Facebook*, vão ter categorias marcadas como interessantes mas sem valor numérico associado. Usando o *Nearest Neighbor* com um pequeno número vizinhos (por volta dos 10) será possível dar um valor numérico a cada uma dessa categorias e assim será possível ter uma estimativa de quais as categorias que um utilizador mais gosta.

Quando for necessário enviar uma campanha com uma determinada categoria, em que não existem suficientes utilizadores com uma boa avaliação para essa categoria, será possível usar o *Slope One* para encontrar utilizadores que nunca viram uma campanha com essa categoria, mas que possivelmente teriam interesse no mesmo.

Quando um sponsor quiser distribuir uma campanha em duas ou mais remessas, pode utilizar a primeira remessa para se verificar a avaliação de alguns utilizadores e nas remessas seguintes será possível usar o *Slope One*, para escolher os utilizadores que ainda não viram essa campanha mas que provavelmente davam uma boa avaliação se a vissem.

Com este tipo de recomendações, é esperado melhorar o tipo de conteúdo que cada utilizador é exposto quando recebe campanhas por e-mail, através do *SocialDB*.

6.3 Trabalho Futuro

Como trabalho futuro ainda falta a aplicação dos sistemas de recomendação, ao sistema real do *SocialDB* e para este processo, possivelmente, será necessário realizar afinações e otimizações. Será necessário testar qual o impacto dos algoritmos no sistema real e comparar com os resultados deste trabalho. Também como trabalho futuro, existe a possibilidade de se testar um maior número de algoritmos, com o intuito de obter resultados significativamente melhores, do que os resultados apresentados neste trabalho.

Bibliografia

[1] Xiaoyuan Su and Taghi M. Khoshgoftaar (2009). *A Survey of Collaborative Filtering Techniques.* Adv. in Artif. Intell. 2009, 19 pages.

[2] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl (2000). *Application of Dimensionality Reduction in Recommender System - A Case Study.* WebKDD-2000 Workshop, 2000.

[3] Sonja Kangas (2002). *Collaborative Filtering and Recommendation Systems.* Research Report TTE4-2001-35, VTT Information Technology, 2002.

[4] Ken Goldberg, Theresa Roeder, Dhruv Gupta and Chris Perkins (2001). *Eigentaste: A Constant Time Collaborative Filtering Algorithm.* Inf. Retr. 4, 2 (July 2001), pp. 133-151.

[5] Tavi Nathanson, Ephrat Bitton and Ken Goldberg (2007). *Eigentaste 5.0: Constant-Time Adaptability in a Recommender System Using Item Clustering.* In Proceedings of the 2007 ACM Conference on Recommender Systems (RecSys '07). ACM, New York, NY, USA, pp. 149-152.

[6] CaiNicolas Ziegler, Sean M. McNee, Joseph A. Konstan, Georg Lausen (2005). *Improving Recommendation Lists Through Topic Diversification.* In Proceedings of the 14th International Conference on World Wide Web (WWW '05). ACM, New York, NY, USA, pp. 22-32.

[7] Francesco Ricci, Lior Rokach and Bracha Shapira (2011). *Introduction to Recommender Systems Handbook.* Springer 2011, pp. 1-35.

[8] Yehuda Koren, Robert Bell and Chris Volinsky (2009). *Matrix Factorization Techniques for Recommender Systems*. Computer 42, 8 (August 2009), pp. 30-37.

[9] Toby Segaram (2007). *Programing Collective Intelligence*. O'Reilly Media 2007, pp. 1-28.

[10] John S. Breese, David Heekerman and Carl Kadie (1998). *Empirical Analysis of Predictive Algorithms for Collaborative Filtering*. In Proceedings of the Fourteenth Conference on Uncertainty in Artificial Intelligence (UAI'98), Gregory F. Cooper and Serafín Moral (Eds.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 43-52.

[11] Lindsay I Smith (2002). *A tutorial on Principal Components Analysis*. `www.cs.otago.ac.nz/cosc453/student_tutorials/principal_components.pdf`, 2002.

[12] Greg Linden, Brent Smith and Jeremy York (2003). *Amazon.com Recommendations Item-to-Item Collaborative Filtering*. IEEE Internet Computing, 2003, pp. 76-80.

[13] Lommatzsch A., Mehlitz M. and Albayrak S. (2007). *Assessing the Value of Unrated Items in Collaborative Filtering*. Digital Information Management, 2007, pp. 212-216.

[14] Badrul Sarwar and George Karypis and Joseph Konstan and John Riedl (2001). *Itembased Collaborative Filtering Recommendation Algorithms*. Proc. 10th International Conference on the World Wide Web, 2001, pp. 285-295.

[15] Daniel Lemire and Anna Maclachlan (2005). *Slope One Predictors for Online Rating-Based Collaborative Filtering*. Proceedings of SIAM Data Mining (SDM'05), pp. 471-475.

[16] Ian H. Witten, Eibe Frank (2005). *Data Mining: Practical Machine Learning Tools and Techniques, Second Edition*. Morgan Kaufmann Publishers Inc. 2005, pp. 143-184.

Apêndice A

Imagens do SocialDB

Neste apêndice será mostrado algumas imagens retiradas do website:
`http://www.thesocialdatabase.org/`.

Figura A.1: Página inicial do SocialDB

Figura A.2: O projecto SocialDB

Figura A.3: Página de registo dos sponsors

Figura A.4: Página de registo das instituições

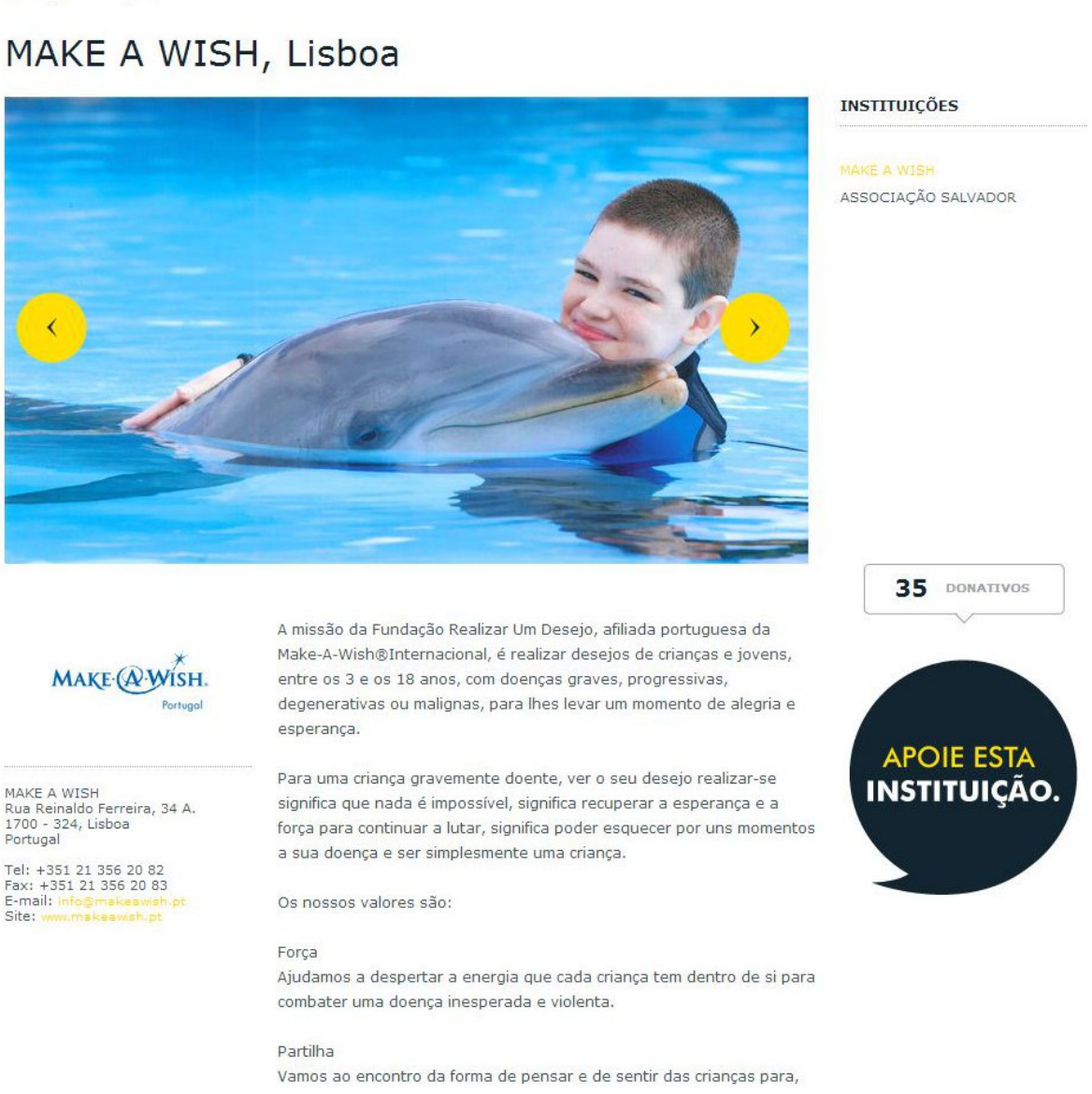

Figura A.5: Exemplo de uma instituição

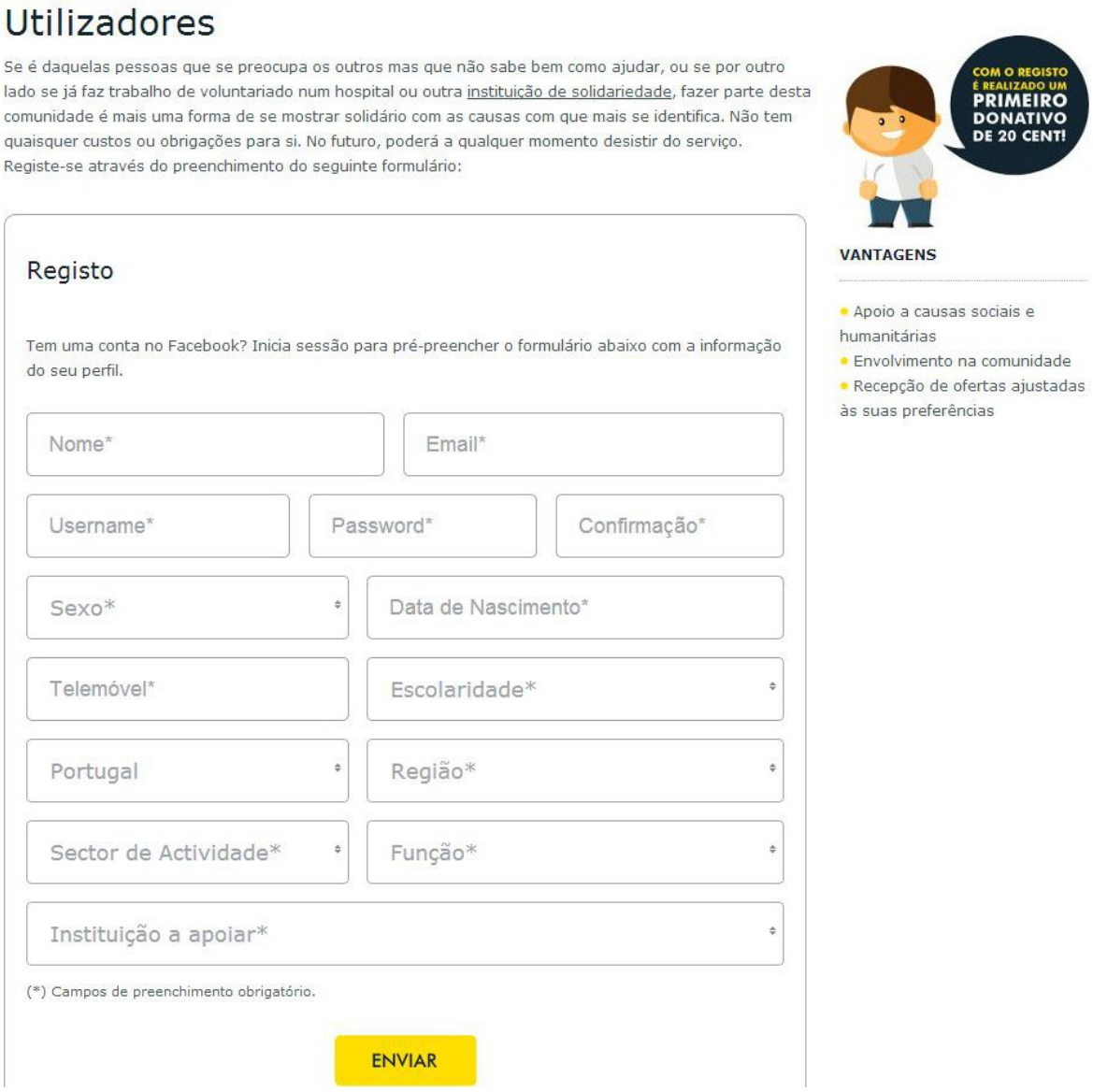

Figura A.6: Página de registo dos utilizadores

Figura A.7: Exemplo de uma página para avaliação de uma campanha

Apêndice B

Tabelas de Resultados

Neste apêndice será mostrado o resultado de cada tentativa para os diferentes algoritmos. Em cada tentativa é dado o erro médio e desvio padrão médio do *10 fold cross-validation*, apresnetado no Cap. 5.

	Erro médio	Desvio Padrão médio
1ª Tentativa	0,9337	0,3604
2ª Tentativa	0,9049	0,3376
3ª Tentativa	0,9159	0,3357
4ª Tentativa	0,911	0,332
5ª Tentativa	0,8949	0,3292
6ª Tentativa	0,9126	0,3336
7ª Tentativa	0,9363	0,3588
8ª Tentativa	0,8962	0,3334
9ª Tentativa	0,9146	0,347
10ª Tentativa	0,8965	0,3306

Tabela B.1: Novos utilizadores com 1 vizinho

	Erro médio	Desvio Padrão médio
1ª Tentativa	0,7163	0,2902
2ª Tentativa	0,7157	0,2897
3ª Tentativa	0,7198	0,2923
4ª Tentativa	0,7181	0,2916
5ª Tentativa	0,7164	0,2889
6ª Tentativa	0,7188	0,2923
7ª Tentativa	0,7146	0,289
8ª Tentativa	0,7189	0,291
9ª Tentativa	0,7161	0,2887
10ª Tentativa	0,716	0,2875

Tabela B.2: Novos utilizadores com 10 vizinhos

	Erro médio	Desvio Padrão médio
1ª Tentativa	0,7276	0,3038
2ª Tentativa	0,7275	0,304
3ª Tentativa	0,7277	0,3037
4ª Tentativa	0,7277	0,3036
5ª Tentativa	0,7276	0,3037
6ª Tentativa	0,7277	0,3036
7ª Tentativa	0,7278	0,3037
8ª Tentativa	0,7278	0,3041
9ª Tentativa	0,7276	0,3036
10ª Tentativa	0,7284	0,3046

Tabela B.3: Novos utilizadores com 100 vizinhos

	Erro médio	Desvio Padrão médio
1ª Tentativa	0,7754	0,3238
2ª Tentativa	0,7756	0,324
3ª Tentativa	0,7754	0,324
4ª Tentativa	0,7754	0,3239
5ª Tentativa	0,7754	0,3239
6ª Tentativa	0,7753	0,3239
7ª Tentativa	0,7754	0,3239
8ª Tentativa	0,7754	0,324
9ª Tentativa	0,7753	0,3239
10ª Tentativa	0,7754	0,324

Tabela B.4: Novos utilizadores com 1000 vizinhos

	Erro médio	Desvio Padrão médio
1ª Tentativa	1,0483	0,8372
2ª Tentativa	1,0554	0,8358
3ª Tentativa	1,0445	0,8356
4ª Tentativa	1,0482	0,8361
5ª Tentativa	1,053	0,8366
6ª Tentativa	1,0405	0,8317
7ª Tentativa	1,0547	0,8404
8ª Tentativa	1,0477	0,834
9ª Tentativa	1,0535	0,8387
10ª Tentativa	1,0432	0,8286

Tabela B.5: Recomendação de uma categoria com 1 vizinho

	Erro médio	Desvio Padrão médio
1ª Tentativa	0,7892	0,5994
2ª Tentativa	0,7877	0,6012
3ª Tentativa	0,7906	0,6011
4ª Tentativa	0,7903	0,5999
5ª Tentativa	0,7927	0,603
6ª Tentativa	0,7963	0,6049
7ª Tentativa	0,7892	0,6002
8ª Tentativa	0,7972	0,6077
9ª Tentativa	0,7936	0,6045
10ª Tentativa	0,7949	0,6047

Tabela B.6: Recomendação de uma categoria com 10 vizinhos

	Erro médio	Desvio Padrão médio
1ª Tentativa	0,7638	0,5651
2ª Tentativa	0,7689	0,57
3ª Tentativa	0,7669	0,5682
4ª Tentativa	0,7664	0,5664
5ª Tentativa	0,7631	0,5677
6ª Tentativa	0,7688	0,5688
7ª Tentativa	0,7597	0,5642
8ª Tentativa	0,7628	0,5652
9ª Tentativa	0,7649	0,5691
10ª Tentativa	0,7644	0,5681

Tabela B.7: Recomendação de uma categoria com 100 vizinhos

	Erro médio	Desvio Padrão médio
1ª Tentativa	0,7701	0,5666
2ª Tentativa	0,7727	0,5703
3ª Tentativa	0,7772	0,5707
4ª Tentativa	0,7807	0,5714
5ª Tentativa	0,7798	0,571
6ª Tentativa	0,7755	0,5695
7ª Tentativa	0,7713	0,5674
8ª Tentativa	0,7728	0,5674
9ª Tentativa	0,7765	0,5726
10ª Tentativa	0,7744	0,5685

Tabela B.8: Recomendação de uma categoria com 1000 vizinhos

	Erro médio	Desvio Padrão médio
1ª Tentativa	0,6655	0,546
2ª Tentativa	0,6668	0,5481
3ª Tentativa	0,666	0,5487
4ª Tentativa	0,6664	0,551
5ª Tentativa	0,6673	0,552
6ª Tentativa	0,665	0,5472
7ª Tentativa	0,6672	0,5505
8ª Tentativa	0,6635	0,5474
9ª Tentativa	0,6666	0,5483
10ª Tentativa	0,6686	0,5515

Tabela B.9: Recomendação de uma categoria com o *Slope One*

	Erro médio	Desvio Padrão médio
1ª Tentativa	1,031	0,9493
2ª Tentativa	1,0391	0,9404
3ª Tentativa	1,079	0,9699
4ª Tentativa	1,0371	0,9306
5ª Tentativa	1,0409	0,9394
6ª Tentativa	1,0621	0,9407
7ª Tentativa	1,0606	0,9447
8ª Tentativa	1,0374	0,9452
9ª Tentativa	1,0401	0,9444
10ª Tentativa	1,0364	0,9382

Tabela B.10: Recomendação de uma campanha com 1 vizinho

	Erro médio	Desvio Padrão médio
1ª Tentativa	0,8071	0,6175
2ª Tentativa	0,8105	0,6186
3ª Tentativa	0,7934	0,6112
4ª Tentativa	0,8279	0,633
5ª Tentativa	0,8126	0,6208
6ª Tentativa	0,8129	0,6267
7ª Tentativa	0,8182	0,6197
8ª Tentativa	0,8131	0,6112
9ª Tentativa	0,8078	0,6189
10ª Tentativa	0,8086	0,6187

Tabela B.11: Recomendação de uma campanha com 10 vizinhos

	Erro médio	Desvio Padrão médio
1ª Tentativa	0,7753	0,5949
2ª Tentativa	0,7684	0,5849
3ª Tentativa	0,7599	0,5793
4ª Tentativa	0,7598	0,5755
5ª Tentativa	0,7766	0,6021
6ª Tentativa	0,7779	0,6005
7ª Tentativa	0,7689	0,5797
8ª Tentativa	0,7739	0,5939
9ª Tentativa	0,7704	0,5837
10ª Tentativa	0,7724	0,5749

Tabela B.12: Recomendação de uma campanha com 100 vizinhos

	Erro médio	Desvio Padrão médio
1ª Tentativa	0,7701	0,5749
2ª Tentativa	0,77	0,5817
3ª Tentativa	0,7818	0,5883
4ª Tentativa	0,7796	0,5926
5ª Tentativa	0,7722	0,5882
6ª Tentativa	0,777	0,5767
7ª Tentativa	0,7611	0,5813
8ª Tentativa	0,774	0,5835
9ª Tentativa	0,7804	0,5972
10ª Tentativa	0,7878	0,6006

Tabela B.13: Recomendação de uma campanha com 1000 vizinhos

	Erro médio	Desvio Padrão médio
1ª Tentativa	0,7186	0,5591
2ª Tentativa	0,7338	0,5708
3ª Tentativa	0,7316	0,5703
4ª Tentativa	0,7296	0,5644
5ª Tentativa	0,73	0,5724
6ª Tentativa	0,7323	0,5705
7ª Tentativa	0,7311	0,5713
8ª Tentativa	0,7258	0,577
9ª Tentativa	0,7324	0,5598
10ª Tentativa	0,7332	0,5674

Tabela B.14: Recomendação de uma campanha com o *Slope One*