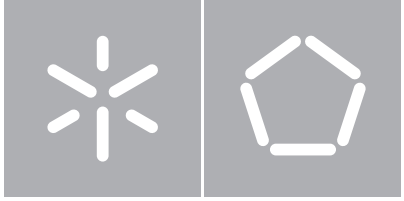
Universidade do Minho
Escola de Engenharia

Luís Duarte Dias Machado

Televisão Interativa: Recomendação de Conteúdos Multimédia

Outubro de 2016



Universidade do Minho
Escola de Engenharia
Departamento de Informática

Luís Duarte Dias Machado

Televisão Interativa: Recomendação de Conteúdos Multimédia

Dissertação de Mestrado
Mestrado em Engenharia Informática

Trabalho realizado sob orientação de

Professor Paulo Novais

Outubro de 2016

Agradecimentos

Quero agradecer a todos os meus amigos e familiares, que indiretamente contribuíram para a realização desta dissertação, e deixar uma sentida homenagem à minha querida avó materna que partiu deste mundo no decorrer desta dissertação.

Também agradeço ao meu orientador, o Professor Paulo Novais, pelos conselhos e sugestões dadas, e sobretudo força e incentivo que transmitiu.

Agradeço ao Davide Carneiro, o apoio através das suas ideias, discussões e experiência transmitida.

Agradeço ainda, a todos os colegas que fui conhecendo ao longo do percurso académico na Universidade do Minho, em especial para aqueles que sempre me deram apoio e me ajudaram com a correção ortográfica deste documento, Cátia Braga, Nuno Oliveira, Dalila Teixeira e Nuno Amorim.

Por fim, o agradecimento mais importante e muito especial, aos meus pais, Maria da Conceição Machado e José Luís Machado, pela dedicação, apoio e incentivo que sempre me deram nas decisões e nos momentos difíceis deste último ano e em toda a minha vida.

Abstract

Nowadays, television offers a large set of contents. When the viewers have a big range of content, it is difficult to choose something they really like. The time spent selecting the content may even consume all the time they have to watch the selected content. This fact influences the level of stress on people. To try to answer this problem, this project uses artificial intelligence techniques to create an intelligent system, providing a better TV experience and contributing to the decrease of stress levels on users, reducing the time consumed.

Keywords: Stress, Multimedia Content, Recommender Systems, Interactive Television.

Resumo

A televisão de hoje em dia disponibiliza um enorme lote de conteúdos. Quando existe um leque de conteúdos de grande dimensão é difícil optar pela solução que mais nos agrada ou que melhor satisfaz as exigências dos espetadores. O tempo consumido na seleção pode mesmo esgotar o tempo que o espetador dispõe para visualizar o conteúdo selecionado. Este facto influencia o nível de stresse nas pessoas. Para tentar responder a este problema, recorreu-se a técnicas de inteligência artificial para a criação de um sistema inteligente que seja capaz de proporcionar ao utilizador uma melhor experiência televisiva, contribuindo para o seu bem estar e diminuição dos seus níveis de stress, reduzindo o tempo gasto.

Palavras-Chave: Stress, Conteúdos Multimédia, Sistemas de Recomendação, Televisão Interativa.

Nota Informativa

O escritor deste documento é portador de dislexia, por esse motivo é possível que surjam alguns erros ortográficos. Caso sejam detetados, seria uma boa prática entrar em contacto com o escritor do documento para que o erro seja corrigido. Caro leitor, agradece-se desde já a sua compreensão.

O que é dislexia?

Segundo a definição mais consensual da Associação Internacional da Dislexia e do National Institute of Child Health and Human Development, a dislexia é uma incapacidade específica de aprendizagem cuja origem é neurológica. Caracteriza-se por dificuldades no reconhecimento exato e fluente das palavras e por uma capacidade deficiente de as soletrar e compreender. Nuno Lobo Antunes, neuropediatra e Director Clínico do Centro de Apoio ao Desenvolvimento Infantil, em Cascais, define a dislexia como "uma dificuldade da leitura e/ou escrita que é desproporcionada ao grau de inteligência da criança e à oportunidade que lhe foi dada de aprender. Assim, para se fazer o diagnóstico é necessário demonstrar uma velocidade de leitura que é pelo menos dois anos académicos abaixo do esperado". A dislexia existe em todas as línguas e culturas. Alguns professores têm dificuldade em perceber como é que crianças com um quociente de inteligência (QI) elevado sofrem de dislexia, sendo por isso acusadas de serem preguiçosas ou desmotivadas. De acordo com Octávio Moura, "não existe uma relação direta entre o QI e o aparecimento da dislexia". Quando os disléxicos se debatem com a capacidade de leitura, encaram uma verdadeira batalha com o cérebro – cérebro esse que simplesmente não está "formatado" para a leitura.

Um número cada vez maior de provas localiza a dislexia numa falha no circuito cerebral, o que torna a leitura extremamente difícil. Através da ressonância magnética, identificaram-se três áreas no hemisfério cerebral esquerdo que estão envolvidas na leitura, duas na parte posterior do cérebro e uma à frente[1].

Para uma maior informação sobre este assunto aconselha-se a vista ao portal da dislexia em www.dislexia.pt

Conteúdo

Lista de Figuras

Lista de Tabelas

1 Introdução

1.1 Motivação

Atualmente, muitas pessoas não dispõem de tempo para assistir televisão, e sempre que dispõem, gastam muito desse tempo na procura de algo que valha a pena ser visto, deixando apenas uma porção desse tempo para a visualização do conteúdo selecionado, não permitindo, muitas vezes, visualizar a sua totalidade. Dado este facto, surgiu a ideia de criar um sistema inteligente de recomendação de conteúdos para fazer face a este problema.

Nos dias de hoje, o tempo livre é cada vez menor, vivemos numa sociedade extremamente ocupada e competitiva em que ferramentas de seleção e recomendação de conteúdos têm um papel cada vez mais importante. Com um sistema de recomendação, o tempo gasto na procura de conteúdos que se adequem ao utilizador será drasticamente reduzido, tornando a experiência televisiva muito mais agradável.

Para além da questão temporal, adequar os conteúdos a cada utilizador pode fazer com que este tenha um momento de lazer totalmente descontraído, sem que tenha a preocupação de procurar um conteúdo para visualizar no enorme labirinto que se pode tornar uma base de dados repleta de conteúdo. Usando a terminologia anglo-saxónica, *big data* pode provocar stress nos utilizadores e em vez de terem um momento de prazer, acabam por ter um momento de "tortura".

Quando não se sabe o que se procura, é muito mais difícil encontrar algo que agrade, mas se existir algo ou alguém que seja capaz de sugerir algo de acordo com o perfil do utilizador, é muito mais fácil encontrar os conteúdos adequados. Por conseguinte, o utilizador não se apercebe do gigantesco volume de dados disponível e encontra facilmente um conteúdo que pode ver durante uma pausa na sua vida stressante, aproveitando este período temporal para relaxar.

Se for possível saber qual o nível de stress do indivíduo, será sempre possível fazer recomendações com base neste facto e apresentar soluções que sejam capazes de baixar o nível de stress e satisfazer os gostos do mesmo. Além de tudo o que já foi apresentado, recorrendo ao perfil do utilizador também será possível direcionar melhor a publicidade, com o intuito de melhorar os resultados da mesma, fazendo com que os produtos publicitados seja exibidos ao seu público

alvo, sem que este pense que a publicidade é enfadonha.

1.2 Âmbito da Dissertação

A sociedade atual vive a um ritmo frenético, sem tempo a perder e o tempo gasto desnecessariamente numa tarefa provoca um aumento nos níveis de stress, como por exemplo o tempo desperdiçado no trânsito. O mesmo se passa quando um utilizador não encontra nenhum conteúdo que satisfaça as suas preferências, em tempo útil, no lote de conteúdos multimédia disponibilizados pelo seu serviço de TV. Para resolver este problema, os sistemas de recomendação são amplamente utilizados com resultados satisfatórios. Estas técnicas são usadas na recomendação de conteúdos como o You Tube ou a Netflix.

Neste projeto, são estudadas algumas técnicas de inteligência artificial. Após uma análise cuidada destas técnicas, foi criado uma aplicação protótipo, a Zen TV, que implementa algumas das técnicas estudadas, que tentam reduzir o tempo despendido pelos utilizadores na seleção dum conteúdo multimédia. Além da disso, também tenta perceber o comportamento e gostos dos utilizadores.

1.3 Projeto ISLab

Este trabalho foi desenvolvido ao abrigo do projeto CAMCoF - Context Aware Multimodal Communication Framework, financiado por Fundos FEDER através do Programa Operacional Fatores de Competitividade - COMPETE e por Fundos Nacionais através da FCT - Fundação para a Ciência e a Tecnologia no âmbito do projeto FCOMP-01-0124-FEDER-028980. E foi elaborado no Laboratório de Sistemas Inteligentes (ISLAB) da Universidade do Minho (UM).

O principal objetivo do CAMCoF é desenvolver uma framework capaz de modelar o ambiente, focado no estado de stress dos seus utilizadores, fornecendo essas informações a um ambiente virtual, permitindo desenvolver processos de comunicação mais ricos. Estes processos de comunicação permitirão aos utilizadores comunicarem de uma forma mais próxima de uma comunicação cara-a-cara. A estrutura será não-intrusiva, a fim de facilitar uma monitorização mais precisa e frequente. Assim, a estimativa do nível de stress será baseada numa análise transparente do comportamento do utilizador.

1.4 Objetivos

Com este trabalho, espera-se que o tempo despendido pelos utilizadores na procura de um conteúdo seja drasticamente reduzido, e que os conteúdos sugeridos sejam o mais próximo dos gostos do utilizador. Tendo em conta quais são os interesses do utilizador, será mais fácil saber qual a publicidade que lhe será dirigida. Desta forma, o utilizador não a verá com algo enfadonho, mas algo que lhe suscite o interesse, com esta noção o mesmo conteúdo multimédia poderá ter vários anúncios associados, cada um dirigido a diferentes perfis.

Abaixo, apresenta-se a lista de objetivos a alcançar:

- Reduzir o tempo de procura por conteúdos;
- Reduzir os níveis de stress do utilizador;
- Aumentar os níveis de satisfação do utilizador;
- Perceber e tentar melhorar os atuais sistemas de recomendação de conteúdos;
- Tentar apontar os anúncios publicitários ao seu público alvo;
- Descobrir perfis de utilizador.

1.5 Plano de Trabalho

Os trabalhos a desenvolver obedecerão às seguintes fases:

- Levantamento do estado da arte sobre Sistemas e Técnicas de Recomendação;
- Análise e conceção de uma solução com base na literatura revista;
- Desenvolvimento de um protótipo capaz de sugerir conteúdos de uma forma personalizada e utilizando o conhecimento adquirido;
- Redação da dissertação de mestrado.

1.6 Metodologia de Investigação

Para a realização dos objetivos anteriormente apresentados, foi seguida uma metodologia ação-pesquisa [2]. Esta metodologia passa pela identificação do problema em causa, para que este

possa ser formulado e desenvolvido. Durante todo o processo, a informação é recolhida, avaliada e organizada de forma contínua, construindo um suporte para a resolução do problema. Por fim, a análise e validação dos resultados obtidos durante a investigação devem desencadear conclusões sobre o problema em causa. Para seguir esta metodologia de investigação, serão seguidos os seguintes passos:

1. Especificar o problema assim como as suas características;
2. Atualização constante e incremental do estado de arte;
3. Modelação e implementação do sistema;
4. Análise de resultados e formulação de conclusões;
5. Validação do sistema.

1.7 Estrutura do Documento

Além do presente capítulo, este documento integra mais seis capítulos, nomeadamente:

- **Capítulo 2 - Televisão Interativa**. Neste capítulo, é feita uma breve abordagem à televisão iterativa, aqui é abordada uma forma de desenhar um serviço de televisão iterativa e como deve ser construída a sua interface. Também é abordada a temática da publicidade, nomeadamente a melhor forma de a apresentar aos utilizadores. Por fim, faz-se uma análise ao panorama português.
- **Capítulo 3 - Stress**. Neste capítulo, é apresentada uma breve noção da teoria de stress e é apresentada uma possível forma de medir o nível de stress dum individuo duma forma não invasiva.
- **Capítulo 4 - Aprendizagem e Recomendação**. Neste capítulo, é feita uma abordagem a diversas temáticas utilizadas na aprendizagem nos sistemas de recomendação. Aqui são descritos os principais método de cálculo de clusters, com especial atenção ao k-means e ao k-medoids. Também é abordada a temática da geração de regras de associação e a descoberta de padrões frequentes. Por fim, é aprofundado o estudo nos sistemas de recomendação, o k-nearest neighbors.

- **Capítulo 5 - Trabalhos Relacionados**. Aqui são apresentados alguns trabalhos que estão relacionados com o tema em análise. É analisado o comércio eletrónico pois muitas das técnicas aqui estudadas também são utilizadas em massa nessa área.

- **Capítulo 6 - Zen TV**. Neste capitulo, é descrito o protótipo resultante do projeto, são expostas as tecnologias utilizadas, como foram obtidos os dados para a validação da aplicação desenvolvida, o modelo de dados utilizado e as técnicas de inteligência artificial usadas, bem como foram implementadas.

- **Capítulo 7 - Conclusões**. Neste último capítulo, apresentam-se as conclusões sobre o trabalho efetuado e algumas diretrizes a seguir de forma a continuar a desenvolver o trabalho realizado no âmbito desta dissertação

2 Televisão Interativa

”A Televisão Interativa é um paradoxo. Mas por outro lado, a televisão é um ponto forte na cultura e um terreno fértil nas conversas do dia-a-dia, que é sem dúvida a interação mais agradável que as pessoas têm. Devemos tentar aproveitar o poder da televisão, e criar um canal que permita ao público fornecer conteúdo, controlo ou apenas uma pequena conversa”.— Dan O’Sullivan, Interactive Telecommunication program New York University, Tisch School of the Arts [3]

A televisão interativa utiliza uma coleção de técnicas que permitem que os telespetadores interajam com os conteúdos visualizados na televisão, permitindo a interatividade. Para tal, os equipamentos televisivos e/ou os conversores digitais devem ser capazes de receber e executar aplicações. Embora o conceito de interatividade é habitualmente mencionado na linguagem quotidiana e na literatura quando se refere às novas tecnologias, nem sempre é claro o que se entende por ’interatividade’. Uma série de trabalhos tem tentado fornecer uma definição do conceito de interatividade, muitas vezes dividindo-o em várias dimensões. McMillan e Downes propõem uma melhor definição conceptual de interatividade baseado em seis dimensões: direção de comunicação, flexibilidade de horário, o sentido de lugar, o nível de controlo, capacidade de resposta, e perceção do propósito de comunicação[4].
No entanto, existe uma necessidade para uma definição mais básica do conceito, para tal recorremos a uma definição apresentada por Laurence Habib[5], em que o conceito de interatividade pode ser dividido em quatro sub-conceitos, a saber:

- **Interatividade Transmissional –** é uma funcionalidade que permite ao utilizador selecionar um conteúdo, de entre vários que são transmitidos num fluxo contínuo de informações, onde não existe um canal de retorno que inibe o utilizador da possibilidade de fazer pedidos. (por exemplo, difusão de dados, multidifusão, teletexto).
- **Interatividade sob Consulta –** é uma funcionalidade que permite ao utilizador, através de um pedido, aceder a um conteúdo num conjunto de vários conteúdos recorrendo a um

canal bidirecional. Exemplo deste tipo de interatividades são os protocolos de internet (FTP, HTTP).

- **Interatividade de Conversação –** é uma funcionalidade que permite ao utilizador produzir e colocar os seus conteúdos no sistema de Multimédia, sendo possível também interagir com outros utilizadores. Estes conteúdos podem ficar armazenados ou serem trocados em tempo real (exemplo vídeo-conferência, chat, e-mail).

- **Interatividade de Registo de Informação –** é uma funcionalidade que permite registar informações e, assim, adaptar e/ou responder a uma dada necessidade ou ação, que traduz uma escolha explícita do utilizador e permite ao sistema automaticamente adaptar-se. (por exemplo, sistemas de vigilância, agentes inteligentes, interfaces inteligentes, etc).

2.1 Conteúdo Multimédia

Quando se fala num conteúdo multimédia, inclui-se, numa só palavra, uma combinação de texto, áudio, imagens estáticas, animação, vídeo ou formas de conteúdo de interatividade. Conteúdos multimédia são geralmente gravados e reproduzidos, exibidos ou acedidos por dispositivos de processamento de conteúdo de informação, tais como dispositivos informáticos e eletrónicos, mas também podem ser parte de uma transmissão ao vivo.[6] Neste trabalho, um conteúdo multimédia refere-se apenas a vídeo ou filmes digitais.

2.2 Desenho de um Serviço de Televisão Interativa

O objetivo do projeto descrito no artigo *Interactive TV service design*[7] é defender os utilizadores comuns do futuro da televisão interativa. Tal como muitos dos produtos interativos de hoje, a televisão interativa causa problemas de usabilidade para utilizadores com baixos índices tecnológicos. A televisão interativa pode ser particularmente interessante para especialistas em usabilidade, dado que quase todas as pessoas têm, pelo menos, uma televisão na sua habitação. Pelas razões apresentadas, os utilizadores comuns assumem o papel mais importante no projeto que essencialmente tenta responder a duas questões, ”o que oferecer?”e ”como operar?”, no que toca aos serviços interativos e no que diz respeito ao aspeto da usabilidade dos futuros sistemas de televisão interativa.

2.2.1 O que oferecer?

Assume-se que as motivações que levam os humanos a assistir televisão esta relacionado com o prazer e hedonismo adquirido durante esse período. Acredita-se que, em torno desta suposição, duas hipóteses formam o universo provável do futuro da televisão interativa.

A primeira é que a televisão interativa pode favorecer o desenvolvimento individual e coletivo. A outra hipótese é que a forma atual de televisão interativa deixará de ser visualizada apenas nos lares dos utilizadores e passará a ter uma configuração móvel ou nómada, o que já é possível verificar hoje em dia.

No designer de televisão interativa, deve-se ter em conta que os serviços estão centrados no entretenimento, dado que não existe uma solução única para o futuro da televisão interativa. Para satisfazer as necessidades dos utilizadores, múltiplas soluções devem ser propostas. Os fatores sociológicos são um dos aspetos importantes da análise de tendências em conjunto com a criatividade.

2.2.2 Como operar?

Uma inovação no atual sistema de entrada de dados será necessário, pois a simples mudança de paradigma entre o sistema de televisão tradicional e os novos sistemas de televisão interativa torna a operabilidade mais complexa. A partir desta perspetiva, observou-se que os sujeitos se queixavam, em geral, da navegação entre as opções que lhes eram apresentadas, sendo complexo navegar entre elas apenas com o tradicional controlo remoto.

Várias vezes, foi observada vergonha por não se saber como chegar a uma zona especifica. A partir desta experiência, percebeu-se que a inovação é necessária num novo sistema de interação. Para fazer face a este problema, surgem três opções de controlo: controlo por voz, controlo através dum dispositivo com touchscreen e um sistema hibrido que combina os dois anteriores.

No entanto, será sempre preciso ter em conta que a experiência de assistir TV não é simples: é frequentemente uma atividade de grupo, de modo que o ruído pode intervir no ambiente de interação, o que não é o melhor para comando de voz. Quando não é o caso, o controlo de voz pode proporcionar uma visualização de televisão num clima muito relaxante. É impossível concluir qual dos dois modos de interação é preferível, contudo a melhor opção é fornecer interfaces multimodais.

2.3 Interface

Preece, em 2000[8], apresentou um conjunto básico de princípios de design e estratégia, entre os quais destaca-se o design focado na usabilidade, porque as pessoas precisam de executar tarefas de forma fácil e eficaz.

Ao adicionar novas funcionalidades aos programas de TV, haverá uma respetiva interface visível no ecrã, o que pode levar ao risco das interações interromperem o gozo de assistir o conteúdo, mudando o foco para a interface. Esta é uma questão importante no caso da TV, pois os espetadores estão a familiarizar-se com um conjunto de técnicas audiovisuais estabelecidas que mantêm a área de vídeo limpa de distrações visuais.[3]

Para resolver este problema, a melhor forma será apresentar a informação num novo ecrã. Os dispositivos móveis já permitem aceder a muitos conteúdos enquanto o utilizador esta fora de casa e até pode agendar gravações no seu serviço de TV. Com a utilização dum dispositivo móvel, os menus seriam apresentados no seu ecrã, mantendo a área de visualização limpa e sem possibilidade de distrações para os outros utilizadores.

Utilizar um dispositivo móvel em substituição do tradicional controlo remoto também permite ao utilizador encontrar o que procura muito mais facilmente. Exemplo disso seria o acesso a um determinado canal que o utilizador pretende assistir mas que não se recorda qual é a posição do canal. Com o tradicional controlo remoto, teria de fazer *zapping* até o encontrar, mas caso utilizasse um dispositivo móvel, ser-lhe-ia apresentada uma lista com todos os canais, em que o utilizador só teria de navegar nesaa lista e selecionar o canal pretendido. De forma análoga, acontece exatamente o mesmo quando o utilizador procura uma conteúdo gravado na sua *box*. O utilizador, com o controlo remoto, tem de navegar sequencialmente pelo menu, mas se utilizasse um dispositivo móvel para fazer esta tarefa, seria muito mais rápido encontrar o que procura. [9]

2.4 Publicidade Personalizada

A televisão iterativa combina a audiência massiva que a televisão tradicional tem com capacidade interativa que a web disponibiliza, o que oferece aos telespetadores uma experiência de entretenimento mais ativa. Uma das consequências mais importantes da interatividade é a possibilidade de personalização do serviço de TV por parte dos utilizador.

Nos meios de comunicação tradicionais, tanto o fornecedor do canal publicitário com o anunciante têm de procurar as informações dos clientes em outros lugares, como por exemplo empresas de pesquisa de mercado ou inquéritos aos consumidores diretos, a fim de personalizar os seus serviços e/ou os seus anúncios.

A televisão iterativa oferece às empresas novas oportunidades para conhecer melhor a audiência e tentar servir melhores conteúdos publicitários ao seu publico alvo, de modo a tornar os espetadores em clientes. Uma vez que num serviço de televisão iterativa estão guardadas informações sobre o perfil do telespetadores, com o uso de técnicas de extração de conhecimento é possível conhecer melhor o publico. Desta forma, seria possível criar conteúdos publicitários para determinados grupos e converter mais facilmente espetadores em clientes.[10]

A escolha da publicidade pode levantar muita controvérsia, pois a publicidade televisiva é irritante para muitas pessoas: uma pesquisa mostrou que 30% das pessoas mudam de canal durante o intervalo publicitário. Por outro lado, alguns estudos propõem que a publicidade pode ser uma solução para os problemas relacionados com os custos dos direitos de autor dos conteúdos, permitindo disponibilizar os conteúdos de forma gratuita aos utilizadores.

De acordo com estudos sobre publicidade, a eficácia da publicidade é melhorada através duma melhor segmentação dos espetadores. A necessidade de dados precisos colide com a preocupação dos utilizadores de invasão de privacidade. Com o reconhecimento da necessidade de proteger a privacidade dos utilizadores, é necessário encontrar uma solução que satisfaça tanto os anunciantes com os utilizadores.[11]

2.5 Panorama Português

Na edição de março de 2015 da revista Exame Informática [12], foi publicado um estudo onde foram analisados os três serviços de televisão líderes em Portugal: Meo, Nos e Vodafone. Estes serviços têm vindo a evoluir, acrescentando novas funcionalidades e, não menos importante, interfaces cada vez mais intuitivas. Esta análise comparou os serviços em termos de experiência de utilização, com especial ênfase na interface. Neste estudo, não foram analisados outros aspetos como os canais disponíveis ou os tarifários.

Nesta primeira fase do estudo, apenas foram focados os seguintes aspetos técnicos:

- Interface;
- Funcionalidades;
- Navegabilidade;
- Usabilidade;
- Controlo remoto;

- Box(tempos de resposta, consumos de energia, funcionalidades).

Os operadores portugueses estão entre os mais sofisticados do mercado global, como demonstram os prémios internacionais que têm conseguido. E em todos os três, foram encontrados pontos fortes e ponto fracos.

Na Meo, os pontos fortes encontrados foram uma interface apelativa e intuitiva, uma velocidade de resposta agradável, sugestões ao utilizador, como por exemplo, as apps mais utilizadas ou programas, e por fim, a possibilidade de jogar no Meo Jogos. O único ponto fraco encontrado foi o facto do controlo remoto ser pouco ergonómico.

No que toca à Nos, representada pelo seu serviço Iris, tem como pontos fortes uma interface intuitiva, comando simples e possibilidade de gravação dos conteúdo multimédia diretamente na cloud da Nos. Os pontos fracos apontados foram a falta de performance do sistema e a falta de botões de acesso rápido no controlo remoto.

Por fim, o último sistema a ser analisado foi o da Vodafone, que apresentou como pontos fortes uma velocidade global muito satisfatória, um controlo remoto "premium" com botões de acesso rápido, comandos por voz e live on tv, que é um serviço da Vodafone que permite aos seus clientes ver ao vivo e em direto, na sua TV da Vodafone, os vídeos que os seus amigos capturam no smartphone, tablet ou câmaras GoPro Hero3 e Hero4. O serviço da Vodafone apresentou como pontos fracos uma interface fora do contexto e alguns menus lentos.

Na edição seguinte (abril de 2015)[13], foi feito um teste inédito em Portugal, juntando 99 subscritores dos operadores analisados, 33 de cada operador. Estas pessoas testaram os três serviços de televisão ao longo de quatro dias.

Os envolvidos no teste foram convidados a analisar a experiência de utilização dos sistemas de TV em análise em três grandes categorias: Interface, Desempenho e Funcionalidades/Serviços. Todos os colaboradores executaram, com a ajuda da equipa da Exame Informática, um guião predefinido, que consistia na execução de tarefas comuns, como por exemplo, mudar de canal, aceder ao videoclube, pesquisar conteúdos e gestão da conta de utilizador.

No final, alguns deles deixaram sugestões bastantes interatuantes, como por exemplo:

- "Personalizar os menus com as opções que mais me interessam." Susana Pedro, 32 anos.
- "A minha TV do futuro confunde-se totalmente com a Net e vai estar disponível onde eu estiver." Pedro Luiz Castro, 65 anos.
- "Sabe os conteúdos que vejo e as horas que gosto de ver. Sugere conteúdos por temática." Marcos Filipe Sousa, 27 anos.

- ”Sistema verdadeiramente interativo, capaz de adaptar-se a cada utilizador e de interagir com outros dispositivos.” Jorge Campeão de Freitas, 56 anos.
- ”Interface semelhante aos smartphones e tablets” Luís Miguel Almeida da Costa
- ”Acesso direto a conteúdos como séries sem passar por canais.” Gustavo Guimarães de Carvalho, 28 anos.

Os intervenientes neste estudo gostaram particularmente das ferramentas de recomendações e pesquisa de conteúdos relacionados. As pessoas que participaram no estudo também responderam a um inquérito onde as perguntas e respostas mais relevantes são mencionadas em seguida.

O que mudaria no serviço de TV?

- 42% - Mais recomendações de conteúdos
- 50% - Mais conteúdos
- 8% - Outras funcionalidades

No futuro, onde mais vai ver TV?

- 15% - No telemóvel
- 20% - No computador
- 65% - No tablet

Se a interface de TV fosse apenas um campo de pesquisa?

- 25% - Seria ótimo, não preciso duma lista de canais
- 40% - Acho um ”salto”demasiado grande
- 35% - Não gosto

Valoriza recomendações de acordo com o seu perfil?

- 23% - Nem por isso
- 36% - Sim, mas apenas ocasionais
- 41% - Sim, com aprendizagem dos meus padrões

3 Stress

3.1 Teoria do Stress

Uma das primeiras definições de stress foi proposta por Selye (1956) [14]. De acordo com Selye, o stress pode ser visto como uma resposta não específica do corpo a solicitações externas, denominando-se stressors. Selye também foi o primeiro a documentar as mudanças químicas e hormonais que ocorrem no corpo devido ao stress.

No entanto, a definição de stress ainda não é consensual na comunidade científica, mantendo-se como um tópico aberto. Na verdade, o stress envolve uma multiplicidade de fatores, muitos deles subjetivos, levando a múltiplas interpretações que tornam difícil ser objetivamente definido.[15]

Enquanto alguns investigadores afirmam que o stress é inclassificável, outros optam por não fornecer uma definição atual, até a obtenção de uma visão mais precisa e consensual do que é o stress.

Na tentativa de resolver essa questão, os investigadores começaram a lidar com o stress de um ponto de vista empírico, concentrando-se na análise dos efeitos cognitivos e comportamentais, vendo o stress como um problema tanto da mente como do corpo, ou seja, um fenómeno psicofisiológico. Nessa linha de análise, uma visão mais recente do stress foca-se numa excitação psicofisiológica que ocorre na resposta do corpo como resultado a estímulos externos.[16]

O stress é um processo biológico pelo qual o organismo atravessa, de forma a adaptar-se a algum desafio, mobilizando as suas energias na luta contra uma doença e/ou na luta pela sua sobrevivência.[14] Segundo Selye, o stress pode afetar o sistema imunológico, sistema gastrointestinal e glândulas supra-renais.[17]

Os estudos sobre o trabalho com computador, e que avaliam os níveis de stress dos trabalhadores, levam às seguintes conclusões gerais:

- O uso do computador produz, direta e indiretamente, um aumento dos níveis de stress. Os efeitos indiretos são mediados principalmente por fatores do projeto de trabalho;

- O stress provocado por tarefas executadas no computador pode produzir reações fisiológicas, tais como, alterações na frequência cardíaca, na pressão arterial, nível de catecolaminas[1] e atividade das ondas cerebrais;
- As estratégias utilizadas para implementar novas tecnologias computacionais podem afetar o nível de stress dos funcionários;
- Os trabalhadores menos qualificados têm maiores níveis de stress do que os quadros superiores;
- Quando há uma transição para uma nova tecnologia, os trabalhadores menos qualificados também apresentam níveis de stress mais elevado devido à nova tecnologia do que os quadros superiores.

3.2 Deteção de Stress de uma Forma não Invasiva

O estudo descrito no artigo *Multimodal Behavioural Analysis for Non-invasive Stress Detection*[15] apresenta uma abordagem que permite medir os níveis de stress em humanos através da análise dos seus padrões comportamentais, quando interagem com dispositivos tecnológicos, tendo em atenção algumas características comportamentais dos indivíduos analisados. O estudo teve a colaboração de 19 pessoas e a informação foi recolhida ao longo de várias fases, com diferentes níveis de stress induzido.

Aos dados recolhidos, é aplicado um teste de hipóteses estatísticas não paramétrico para determinar quais as características que demonstram diferenças estatisticamente significativas, para cada pessoa, quando está sob stress.

Este estudo foi desenvolvido com o intuito de acrescentar uma nova funcionalidade à plataforma de resolução de conflitos UMCourt. Esta plataforma tenta fornecer às partes envolvidas em disputas jurídicas mais informação necessária para que elas possam tomar decisões mais racionais, permitindo acordos que sejam mais vantajosos. Para que tal seja possível, as partes são informadas de qual é o seu melhor e pior cenário e qual é o possível desenlace do litígio, bem como todos os outros possíveis desenlaces [19].

A fim de identificar quais os fatores que variam de acordo com o stress e em que magnitude, um jogo foi desenvolvido. O principal objetivo do jogo é que o utilizador execute cálculos mentais

[1] Catecolamina é um super-grupo que engloba dopamina, noradrenalina e epinefrina. Catecolaminas são moléculas que podem ser encontradas no sistema nervoso central, no cérebro, em neurónios e na medula supra-renal. [18]

usando as quatro operações aritméticas básicas e um grupo de números dado aleatoriamente, a fim de chegar o mais próximo possível de um dado número.

A recolha dos dados foi organizada em duas fases. Numa primeira fase, o utilizador foi solicitado a jogar o jogo por algumas rodadas sem qualquer fonte de stress. Nesta versão, o jogo não tem limite de tempo, nem vibração ou bipes irritantes. Nesse sentido, o utilizador joga com calma, com tempo suficiente para pensar sobre as diferentes possibilidades. Esta fase permite recolher alguns dados sobre a forma como o utilizador normalmente se comporta quando não está sob stress. Isto permite estabelecer uma linha de base para comparação.

Posteriormente, os mesmos dados são coletados quando o utilizador está jogar o jogo com os stressors, ou seja, na segunda fase de recolha de dados. Nesta fase, a cada ronda consecutiva do jogo, o utilizador tem de efetuar um cálculo, no entanto, agora existem fatores de stress que tornam visíveis os seus efeitos, nomeadamente um limite de tempo é imposto a cada ronda, são ativadas as vibrações e sons desagradáveis surgem.

Como dito antes, o jogo é desenvolvido para induzir o stress no utilizador e para determinar de que forma cada utilizador é afetado. No estudo, são analisados 8 parâmetros para cada utilizador: aceleração, a intensidade máxima do toque, intensidade mínima do toque, a duração do toque, quantidade de movimento, pontuação, precisão do toque e a quantidade de toques classificados como stressados.

3.2.1 Funcionalidades Analisadas

Para se poder analisar se um dado indivíduo está sobre stress, é necessário recorrer a sensores. Para a realização do estudo supra mencionado, recorreu-se aos equipamentos tecnológicos presentes no cotidiano, tais como smartphones, tablets e computadores. De entre todas as funcionalidades presentes em tais equipamentos, recorreu-se a touchscreens com a capacidade de detetar a intensidade do toque, câmaras e acelerómetros, analisando-se as seguintes características:

- **Padrão de Toque –** representa a maneira como um utilizador toca no dispositivo, representa uma variação de intensidade ao longo de um período de tempo;
- **Precisão do Toque –** uma comparação entre toques em áreas ativas e toques em áreas vazias nas quais não faz sentido tocar;
- **Intensidade do Toque –** representa a quantidade de força que o utilizador está a colocar em cada toque e é analisada a intensidade máxima, mínima e média de cada toque;
- **Duração do Toque –** representa o intervalo de tempo entre o início e o final do toque;

- **Quantidade de movimento –** representa como e quanto o utilizador se move dentro do ambiente. Uma estimativa da quantidade de movimento é construída a partir da câmara de vídeo. O processamento de imagem usa técnicas de diferença entre imagens para calcular a quantidade de movimento entre duas frames consecutivas;

- **Aceleração –** a aceleração é medida a partir de acelerómetros em dispositivos móveis. É útil para a construção de uma estimativa de quanto o utilizador se move e como o está a fazer, por exemplo, perceber se o utilizador realiza movimentos bruscos. Para além disso, a informação a partir do acelerómetro é utilizada para suportar a estimativa da intensidade do toque.

Os resultados do teste mostram que as funcionalidades influenciadas pelo stress são a aceleração e a intensidade média e máxima do toque. Também é possível apurar que cada individuo é afetado pelo stress de uma maneira específica. Além disso, todo o processo é realizado de uma forma não invasiva.

4 Aprendizagem e Recomendação

No decorrer deste capitulo, utilizou-se como principal diretriz o livro *Data Mining: Concepts and Techniques* [20]. Este livro é a principal fonte de conhecimento utilizada. Para facilitar o acesso às informações por ele transmitidas, vão ser feitas múltiplas referências ao mesmo, com a indicação das partes utilizadas em cada tema abordado.

4.1 Clusters

O processo de agrupamento de objetos físicos ou abstratos em classes ou grupos é denominado de clustering. A segmentação é feita de modo que os objetos dentro de um grupo têm alta similaridade em comparação com outros elementos do grupo, mas são muito dissimilares quando comparados com objetos de outros grupos, designados de clusters. Um cluster pode ser tratado com um único objeto, por isso pode ser considerado como uma forma de compressão de dados.

A análise de clusters é uma importante atividade humana. Logo no início da infância, aprendemos a distinguir entre cães e gatos, entre animais e plantas, melhorando continuamente e de forma subconsciente os esquemas de clustering.

A avaliação da semelhança entre objetos tem por base os valores dos atributos que descrevem os objetos. Estes atributos são usados no cálculo da distância entre dois objetos, ou seja, quanto mais próximos estão, mais semelhantes são os objetos.

Quando comparado com o clustering, pode-se dizer que a classificação também é um meio eficaz para distinguir grupos ou classes de objetos, mas requer a recolha de um grande conjunto de treino e a rotulagem da cada possível grupo ou classe, utilizado para modelar cada grupo. Esta tarefa por muitas vezes é dispendiosa ou até mesmo impraticável o que obriga a fazer o processo em sentido inverso, ou seja, numa primeira fase, segmentar o conjunto de dados em grupos com base na semelhança de dados, e depois atribuir etiquetas a um número relativamente pequeno de grupos. Além disso, o processo de clustering é adaptável a mudanças.

A análise de clusters é amplamente utilizada em inúmeras aplicações, como por exemplo, estudos de mercado, reconhecimento de padrões, análise de dados ou até em processamento de

imagem.

No comércio, o clustering pode ajudar os comerciantes a descobrir grupos distintos nas suas bases de dados de clientes e caracterizar os seus clientes com base em padrões de compra.

O clustering também pode ser usado em aplicações de deteção de fraudes com cartões de crédito e no acompanhamento das atividades criminosas no comércio eletrónico.

Em biologia, pode ser utilizado para derivar taxonomia de plantas e animais, categorias de genes com funcionalidade semelhante e ter uma visão sobre estruturas inerentes nas populações.

Em *machine learning*, o clustering é um exemplo de aprendizagem não supervisionada. Ao contrário da classificação, o clustering utiliza aprendizagem não supervisionada, não dependendo de classes predefinidas, como já foi referido. Por esta razão, o clustering é uma forma de aprendizagem por observação, em vez de aprendizagem com base em exemplos. [21]

4.1.1 Tipo de Dados na Análise de Clusters

Nesta secção, analisam-se os tipos de dados que frequentemente ocorrem na análise de clusters e como pré-processá-los em tal análise. Supondo que um conjunto de dados a ser agrupados contém n objetos, que podem representar pessoas, casas, compras, animais, entre outros, os principais algoritmos de clustering normalmente utilizam as seguintes estruturas dados[21]:

- **Matriz de dados:** A matriz representa os n objetos, com p variáveis (medições ou atributos), tais como a idade, altura, peso, género, e assim por diante. A estrutura de dados pode estar na forma de uma tabela relacional, ou uma matriz de n por p (n objetos $\times$ p variáveis) como está ilustrado na matriz a baixo (4.1).

$$\begin{pmatrix} x_{1,1} & \cdots & x_{1,f} & \cdots & x_{1,p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{i,1} & \cdots & x_{i,f} & \cdots & x_{i,p} \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ x_{n,1} & \cdots & x_{n,f} & \cdots & x_{n,p} \end{pmatrix} \tag{4.1}$$

- **Tabela de Contingência:** Esta matriz armazena o conjunto de proximidades disponíveis para todos os pares de objetos. Muitas vezes, é representada por uma matriz n por n, um

exemplo é apresentado na matriz 4.2.

$$\begin{pmatrix} 0 & & & & \\ d(2,1) & 0 & & & \\ d(3,1) & d(3,2) & 0 & & \\ \vdots & \vdots & \vdots & 0 & \\ d(n,1) & d(n,2) & \cdots & \cdots & 0 \end{pmatrix} \tag{4.2}$$

Na matriz 4.2, $d(i,j)$ representa a distância ou dissemelhança entre os objetos i e j. Geralmente, $d(i,j)$ resulta num número não negativo que está próximo de 0, quando objetos i e j são muito semelhantes ou "próximos" um do outro, e torna-se maior quanto mais eles diferem. Uma vez que $d(i,j) = d(j,i)$ e $d(i,i) = 0$, temos uma matriz simétrica, ou seja, a matriz é igual à sua transposta ($A = A^T$)[22].

Variáveis Contínuas

Variáveis intervalares são medições contínuas numa escala aproximadamente linear, onde é possível quantificar as distâncias entre as medições, no entanto não existe um ponto nulo natural. Exemplos típicos são o peso e altura, latitude e longitude e a temperatura.

A unidade de medida utilizada pode afetar a análise dos clusters. Um exemplo é alterar a unidade de medida de metro para polegada, ou de quilómetros para milhas, o que pode levar a um agrupamento dos dados muito diferente. Geralmente, expressar uma variável em unidades mais pequenas conduzirá a um maior leque de intervalos, e como consequência, um maior efeito sobre a organização dos objetos nos clusters resultantes. Para evitar a dependência da escolha de medida, os dados devem ser normalizados. A normalização das medições procura dar a todas as variáveis um peso igual. Isto é particularmente útil quando não se tem conhecimento prévio dos dados. No entanto, em algumas aplicações, pode dar-se, intencionalmente, mais peso a um determinado conjunto de variáveis do que a outros. Para tal, adicionam-se esses pesos na função de distancia, algo que será abordado mais à frente neste documento.

Para normalizar as medições, uma opção é converter as medições originais para variáveis sem unidade, onde os valores para uma variável f, são convertidos da seguinte forma:

- Calcular o desvio médio absoluto, S_f:

$$S_f = \frac{1}{n}(|x_{1f} - m_f| + |x_{2f} - m_f| + \cdots + |x_{nf} - m_f|), \tag{4.3}$$

onde $x_{1f}, ..., x_{nf}$ são n valores dos vários objetos para o atributo f, e m_f é o valor médio de f , ou seja, $m_f = \frac{1}{n}(x_{1f} + x_{2f} + \cdots + x_{nf})$.

- Calcular o valor normalizado:

$$Z_{if} = \frac{x_{if}}{S_f}. \tag{4.4}$$

O desvio médio absoluto, S_f, é mais robusto para valores discrepantes do que o desvio padrão, σ_f. Quando se calcula o desvio médio absoluto, os desvios da média, ou seja $|x_{if} - m_f|$, não são quadráticos. Assim, o efeito dos valores discrepantes é relativamente reduzido.

Após a manipulação das variáveis, a dissimilaridade (ou semelhança) entre os objetos decretos é tipicamente calculada com base na distância entre cada par de objetos. A medida mais popular é a distância Euclidiana, definida por:

$$d(i, j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots + (x_{in} - x_{jn})^2}, \tag{4.5}$$

onde $i = (x_{i1}, x_{i2}, \cdots, x_{in})$ e $j = (x_{j1}, x_{j2}, \cdots, x_{jn})$ são dois objetos com n dimensões.
Outra métrica bem conhecida e muito utilizada é distância Manhattan, definida por:

$$d(i, j) = |x_{i1} - x_{j1}| + |x_{i2} - x_{j2}| + \cdots + |x_{in} - x_{jn}|. \tag{4.6}$$

Tanto a distância Euclidiana como a distância de Manhattan satisfazem os seguintes requisitos de uma função de distância:

1. $d(i, j) \geq 0$: A distância é sempre um número não negativo.
2. $d(i, i) = 0$: A distância de um objeto a ele mesmo é 0.
3. $d(i, j) = d(j, i)$: A distância é uma função simétrica.
4. $d(i, j) \leq d(i, h) + d(h, j)$: Ir diretamente do objeto i para o objeto j no espaço não é mais do que fazer um desvio sobre qualquer outro objeto h (desigualdade triangular).

Variáveis Binárias

As variáveis binárias possuem apenas dois estados: 0 ou 1, em que 0 significa que a variável está ausente, e 1 significa que está presente. Dada a variável ”febre”, por exemplo, para um paciente com a variável a 1 indica que o doente tem febre, enquanto que 0 indica que o doente não tem febre. Tratar as variáveis binárias como se fossem contínuas ou intervalares pode levar a resultados falsos na organização dos clusters. Portanto, é necessário utilizar métodos específicos para dados binários para um cálculo correto das dissimilaridades.

Para calcular a dissimilaridade entre duas variáveis binárias, utiliza-se duma tabela de contingência a partir das variáveis binárias fornecidas.

Se todas as variáveis binárias são consideradas como tendo o mesmo peso, temos uma tabela de contingência 2 por 2 a partir da tabela 4.1, onde $\mathbf{q}$ é o número de variáveis que são iguais a 1 para ambos os objetos $\mathbf{i}$ e $\mathbf{j}$, $\mathbf{r}$ é o número de variáveis que são iguais a 1 para o objeto $\mathbf{i}$, mas que são iguais a 0 para o objeto $\mathbf{j}$, $\mathbf{s}$ é o número de variáveis que são iguais a 0 para o objeto $\mathbf{i}$ mas iguais a 1 para o objeto $\mathbf{j}$, e por fim, $\mathbf{t}$ é o número de variáveis que são iguais a 0 para ambos os objetos $\mathbf{i}$ e $\mathbf{j}$. O número total de variáveis $\mathbf{p}$ é dado pela soma de todas as analisadas, ou seja,

$$p = q + r + s + t. \tag{4.7}$$

Objeto i \ Objeto j	1	0	Soma
1	q	r	q + r
0	s	t	s + t
Soma	q + s	r + t	p

Tabela 4.1: Tabela de Contingência

As variáveis binárias estão divididas em **simétricas** e **assimétricas**. Uma variável binária é **simétrica** se ambos os estados têm o mesmo valor e o mesmo peso, ou seja, não há preferência se o estado é codificado como 0 ou 1. Exemplo disso poderia ser o género, tendo os estados masculino e feminino, onde é indiferente o masculino ser representado por 1 e feminino por 0, ou masculino ser representado por 0 e feminino por 1. A dissimilaridade ou distância entre os objetos $\mathbf{i}$ e $\mathbf{j}$ é definida na equação (4.8).

$$d(i, j) = \frac{r + s}{q + r + s + t} \tag{4.8}$$

Uma variável binária é **assimétrica** se os resultados dos estados não são igualmente importantes, tais como os resultados positivos e negativos de um teste de doença. Por convenção, vamos codificar o resultado mais importante, o que geralmente é o mais raro, com o valor 1 (por exemplo, diabetes positivo) e o outro por 0 (diabetes negativo). Dadas duas variáveis binárias assimétricas, a ocorrência de dois valores 1 (um resultado positivo) é então considerada mais significativa do que a de dois valores 0. Como o número de correspondências negativas, $\mathbf{t}$, é considerado pouco importante, é ignorado no cálculo, como se mostra na equação 4.9

$$d(i,j) = \frac{r+s}{q+r+s} \tag{4.9}$$

Complementarmente, pode-se medir a distância entre duas variáveis binárias baseadas no conceito de semelhança em vez de dissimilaridade. A **similaridade binária assimétrica** entre os objetos **i** e **j** é denotada por $sim(i,j)$. O coeficiente $sim(i,j)$ é denominado de **coeficiente de Jaccard**, calculado da seguinte forma:

$$sim(i,j) = \frac{q}{q+r+s} = 1 - d(i,j). \tag{4.10}$$

Variáveis nominais

Uma variável nominal é uma generalização das variáveis binárias, em que pode assumir mais do que dois estados. Um exemplo disso é o mapa de cores onde cada variável pode ter cinco estados: vermelho, amarelo, verde, rosa e azul. Suponha-se que o número de estados de uma variável nominal é M. Os estados podem ser denotados por letras, símbolos ou um conjunto de inteiros, tal como $1, 2, \ldots, M$. De notar que tais números inteiros são usados apenas para manipulação dos dados e não têm qualquer ordem específica.

A dissimilaridade entre dois objetos **i** e **j** pode ser calculada com base na relação de incompatibilidades:

$$d(i,j) = \frac{p-m}{p}, \tag{4.11}$$

em que $\mathbf{m}$ é o número de correspondências (isto é, o número de variáveis em que **i** e **j** têm o mesmo estado), e $\mathbf{p}$ representa o número total de variáveis.

Variáveis Ordinais

Uma variável ordinal discreta assemelha-se a uma variável nominal, a diferença está no facto dos $\mathbf{M}$ estados encontrarem-se ordenados numa sequência significativa. As variáveis ordinais

são muito úteis para registar as avaliações subjetivas de qualidades que não podem ser medidas objetivamente. Por exemplo, as notas de uma aluno podem ser enumeradas numa ordem sequencial, como ”Não Satisfaz”, ”Satisfaz”, ”Satisfaz Bastante”e ”Excelente”. Uma variável ordinal contínua parece-se com um conjunto de dados contínuos de uma escala desconhecida; isto é, a ordem relativa dos valores é essencial, mas a sua magnitude real não é. Por exemplo, a classificação relativa num desporto olímpico (ouro, prata, bronze) é muitas vezes mais importante do que os valores reais de uma determinada medida alcançada num salto em comprimento. As variáveis ordinais também podem ser obtidas a partir da discretização das quantidades em intervalo, dividindo a faixa de valor em um número finito de classes ou escalões. Vamos ilustrar com um exemplo: suponha-se que uma variável ordinal f tem M_f estados ordenados que definem uma hierarquia de 1, ... , M_f .

O tratamento das variáveis ordinais é bastante semelhante ao das variáveis contínuas, no que diz respeito ao cálculo da dissimilaridade entre objetos. Suponha-se que f é uma variável a partir de um conjunto de variáveis ordinais que descrevem n objetos. O cálculo da dissimilaridade relativa a f envolve os seguintes passos:

1. O valor de f para o *(i)*nesimo objeto é x_{if}, e f tem M_f estados ordenados, ordenados de 1 até M_f (1, ... , M_f). Substitui-se cada x_{if} pela sua classificação correspondente (rank), $r_{if} \in \{1, ..., Mf\}$.

2. Uma vez que cada variável ordinal pode ter um número diferente de estados, muitas vezes é necessário normalizar a escala de cada variável no intervalo [0,1] de modo que cada variável possua o mesmo peso. Isto pode ser alcançado através da substituição de rank r_{if} por:

$$Z_{if} = \frac{r_{if} - 1}{M_f - 1}. \tag{4.12}$$

3. Por fim, a dissimilaridade pode ser calculada utilizando qualquer uma das medidas de distância usadas para as variáveis contínuas (secção 4.1.1), para tal é usado o fator Z_{if} em substituição do valor de f para todas as variáveis ordinais.

Variáveis de vários tipos

Até este ponto, apenas foi abordado o cálculo de dissimilaridade entre variáveis de um único tipo. No entanto, em muitas bases de dados reais, os objetos são compostos por mistura de tipos de atributos. Em geral, uma base de dados pode conter todos os tipos de atributos já abordados.

Para calcular a dissimilaridade entre objetos com vários tipos de atributos, uma abordagem possível é agrupar cada tipo num conjunto, realizando o cálculo dos clusters, separado para cada tipo de atributo. Isso é viável, se a análise obter resultados compatíveis. No entanto, em aplicações reais, é improvável que o cálculo dos clusters separados por tipo de atributos gere resultados compatíveis, impossibilitando a utilização dos resultados obtidos.

Uma abordagem preferível seria processar todos os tipos de atributos em conjunto, realizar uma única análise, utilizando uma técnica que combina as diferentes variáveis numa única tabela de contingência, normalizando toda a informação. O valor de todos os atributos deverá estar compreendido no intervalo [0.0, 1.0].

Suponha-se que o conjunto de dados contém p atributos de tipos variados. A dissimilaridade $d(i, j)$ entre os objetos i e j é defendida por:

$$d(i,j) = \frac{\sum_{f=1}^{p} \delta_{ij}^{(f)} d_{ij}^{(f)}}{\sum_{f=1}^{p} \delta_{ij}^{(f)}}, \tag{4.13}$$

em que o indicador $\delta_{ij}^{(f)} = 0$ se qualquer (1) x_{if} ou x_{jf} está ausente (isto é, não existe medição para o atributo f no objeto i ou no objeto j); ou (2) $x_{if} = x_{jf} = 0$ e f é um atributo binário assimétrico; caso contrário, $\delta_{ij}^{(f)} = 1$. A contribuição do atributo f para a dissimilaridade entre i e j, isto é, d_{ij} , é calculada dependendo do tipo. De seguida, mostra-se como calcular este fator:

- Se f é um atributo contínuo: $d_{ij}^{(f)} = \frac{|x_{if} - x_{jf}|}{max_h x_{hf} - min_h x_{hf}}$ onde h é determinado percorrendo todos os objetos onde o atributo f está presente;
- Se f é um atributo binário ou nominal: $d_{ij}^{(f)} = 0$ se $x_{if} = x_{jf}$; caso contrário $d_{ij}^{(f)} = 1$;
- Se f é um atributos ordinal: calcular os ranks r_{if} e $z_{if} = \frac{r_{if}-1}{M_f-1}$, e tratar z_{if} como contínuo.

Os passos acima são idênticos aos que foram referidos anteriormente para cada um dos tipos individuais. A única diferença relevante está no facto dos atributos contínuos estarem normalizados, assim, a dissimilaridade entre objetos pode ser facilmente calculada, mesmo quando os atributos que descrevem os objetos são de diferentes tipos.

4.1.2 Principais Métodos de Cálculo de Clusters

É difícil fornecer uma única categoria aos algoritmos de clustering mais referidos na literatura, porque as categorias podem sobrepor-se, de modo que um método pode ter características de várias categorias. No entanto, é útil apresentar uma imagem relativamente organizada dos diferentes métodos. Nas seguintes secções, são apresentados os principais métodos de clustering.

Métodos de Particionamento

Dada uma base de dados com n objetos, é construído um método com k partições, em que cada partição representa um cluster, e $k \leq n$, ou seja, o método divide os objetos em k em grupos, que satisfazem os seguintes requisitos:

1. cada grupo tem de conter, pelo menos, um objeto;

2. cada objeto só pode estar presente num único grupo.

Dado o número de partições a construir, k, o método de segmentação cria as primeiras k partições. Em seguida, utiliza uma técnica de medição iterativa que tenta melhorar a divisão efetuada, movendo objetos de um grupo para outro. Considera-se que uma segmentação é boa, quando se comparam objetos do mesmo cluster, estes estão ”próximos” ou relacionados uns com os outros, e quando se comparam objetos de diferentes clusters, estes estão ”distantes” ou são muito diferentes.

Para alcançar o ótimo global, com um método de particionamento, seria necessário uma enumeração exaustiva de todas as possíveis partições. Mas em vez disso, a maioria das aplicações utiliza um dos métodos heurísticos mais populares, como por exemplo, k-means ou k-medoids. O k-means representa o seu centro recorrendo ao valor médio de todos os objetos do cluster. No caso do k-medoids, o valor central é representado por um objeto localizado perto do centro do cluster. Nas secções 4.1.3 e 4.1.4, será, respetivamente, aprofundado o estudo destes algoritmos.

Métodos Hierárquicos

Um método hierárquico cria uma decomposição hierárquica do conjunto de objetos fornecido como parâmetro. Um método hierárquico pode ser classificado como de aglomeração ou de divisão, com base na decomposição hierárquica. A abordagem de aglomeração, também conhecida com abordagem de baixo para cima, começa com segmentos formados por um único objeto. A cada iteração, vai sucessivamente juntando grupos que estão próximos uns dos outros, até que todos os grupos estejam fundidos num único super grupo (nível mais alto da hierarquia), ou até que uma condição de paragem seja alcançada. A abordagem de divisão, também conhecida por abordagem de cima para baixo, começa com todos os objetos no mesmo cluster. Em cada iteração, um cluster é dividido em clusters menores, até que, eventualmente, cada objeto forma um cluster, ou até que uma condição de paragem seja alcançada.

Nos métodos hierárquicos, cada etapa (fusão ou divisão) nunca poderá ser desfeita. Esta rigidez é útil, porque permite reduzir os custos de computação, devido ao facto de não ser necessário

preocupar em todas as combinações de objetos possíveis. No entanto, é impossível corrigir decisões erradas. No entanto, existem duas abordagens para melhorar a qualidade do agrupamento hierárquico. A primeira realiza uma análise cuidadosa de todas as ligações possíveis entre os clusters em cada divisão hierárquica. A segunda integra a fusão hierárquica utilizando primeiro um algoritmo de fusão para agrupar objetos em microclusters, e, em seguida, realizar um outro método de agrupamento, como por exemplo a realocação iterativa, onde se desloca um objeto de um cluster para outro, afim de obter melhores resultados.

Métodos Baseados na Densidade

A maioria dos métodos de clustering agrupa os objetos com base na distância entre eles, estes métodos podem encontrar apenas os clusters de forma esférica e dificilmente irão descobrir clusters com formas arbitrárias. Para fazer face a este problema, foram desenvolvidos métodos com base na noção de densidade. A ideia geral é fazer crescer o cluster até uma determinada densidade (número de objetos máximos) na ”zona” se excede o limiar fixado, ou seja, para cada ponto dentro dum dado cluster, a vizinhança de um dado raio tem de conter um número mínimo de pontos. Este método pode ser utilizado para filtrar o ruído (outliers) e como já foi referido descobrir clusters de formas arbitrárias.

Métodos Baseados no Modelo

Os métodos baseados no modelo criam um modelo para cada um dos clusters e encontram a melhor distribuição dos objetos para o modelo dado. Para tal, os algoritmos baseados em modelos constroem os clusters através duma função de densidade que reflete a distribuição espacial dos objetos dados, e determinam automaticamente o número de clusters com base em estatísticas, levando o ”ruído”ou valores extremos em conta, produzindo clusters mais robustos.

A escolha do algoritmo de segmentação depende do tipo de dados disponíveis e da finalidade da aplicação. Se a análise de clusters é utilizada como ferramenta descritiva ou exploratória, é possível usar vários algoritmos sobre os mesmos dados, e depois fazer uma análise ao que os dados revelaram.

4.1.3 K-means

O algoritmo k-means tem como parâmetro de entrada o número de partições a criar, k, e o conjunto de n objetos. O algoritmo irá distribuir os n objetos em k clusters, de modo a que a semelhança intracluster seja elevada, mas a semelhança extracluster seja baixa. A similaridade

do cluster é medida com base no valor médio dos objetos de um cluster, a este ponto médio dá-se o nome de **centróide**.

O k-means, numa primeira fase, seleciona aleatoriamente k objetos, cada um dos quais representa inicialmente o centróide dum cluster e é ao mesmo tempo elemento do cluster. Para cada um dos restantes objectos, é atribuído ao cluster com o centróide mais semelhante, com base na distância entre o objeto e o centróide. Em seguida, são cálculados novos centróides para cada cluster. Este processo repete-se até que a função de paragem convirja. Tipicamente, é usada a função do erro quadrático, definida por:

$$E = \sum_{i=1}^{k} \sum_{p \in C_i} |p - m_i|^2, \tag{4.14}$$

em que E é a soma do erro quadrático para todos os objetos no conjunto, p é o ponto no espaço que representa um determinado objeto e m_i é a média do cluster C_i (ambos p e m_i são multi-dimensionais). Por outras palavras, para cada objeto, em cada cluster, a distância do objeto ao centróide do cluster a que pertence, é elevada ao quadrado, depois são somadas todas as distâncias. Este critério tenta fazer com que os k clusters resultantes sejam compactos e que estejam tão separados quanto possível.

Para a melhor compreensão do funcionamento do k-means, é apresentado um resumo do algoritmo abaixo[21, 23].

Input:

- k: o número de clusters,
- D: conjunto dados que contém n objetos.

Output: Um conjunto de k clusters.
Método:

1. arbitrariamente escolhe-se k objetos de D como centróides iniciais;
2. **repetir**
3. (re)atribui-se cada objecto ao cluster cujo centróide é mais similar ao objeto;
4. atualizar os centróides, isto é, recalcular o valor médio de todos os objetos para cada cluster;

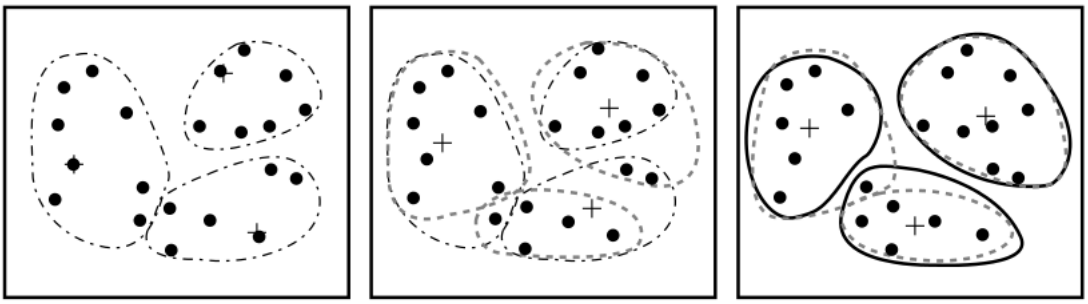


Figura 4.1: Segmentação dum conjunto de objetos recorrendo ao k-means

5. **até** que não ocorra nenhuma mudança;

De acordo com a figura 4.1, selecionaram-se arbitrariamente três objetos para centróides iniciais. Os centróides estão marcados por um "+" em cada iteração. Após selecionar os centróides, cada objeto é distribuído pelos clusters, com base no centróide que lhe é mais próximo. A distribuição dos objetos está organizada nos clusters que estão delimitados pelas linhas a tracejado, tal como mostra a figura 4.1(a).

Na próxima iteração, os centróides são atualizados, ou seja, o valor médio de cada conjunto é recalculado com base nos atuais objetos que compõem o cluster. Utilizando os novos centróides, os objetos são redistribuídos pelos clusters, medindo a distância de cada objeto a todos os centróides e alocando o objeto ao cluster cujo centróide lhe está mais próximo, como é possível observar na figura 4.1(b), onde a redistribuição é representada pela fronteiras a tracejado mais claro.

A iteração seguinte leva-nos a análise da figura 4.1(c). O processo de reafetação iterativa dos objetos com o intuito de melhorar a divisão é denominado por deslocalização iterativa. Este processo termina quando numa iteração não ocorre nenhuma deslocalização, ou seja, quando nenhum objeto muda de cluster. Após o fim da deslocalização iterativa, o algoritmo termina, retornando os clusters resultantes.

O algoritmo tenta determinar as k partições, minimizando a função quadrática do erro, e produz resultados aceitáveis quando os clusters são compactos e bem separados uns dos outros. O método é relativamente escalável e eficiente no processamento de grandes conjuntos de dados, porque a complexidade computacional do algoritmo é dada por $O(nkt)$, onde n é o número total de objectos, k é número de clusters e t é o número de iterações. Normalmente, $k < n$ e $t < n$, e geralmente o método termina num ótimo local.

O método k-means, no entanto, só pode ser aplicado quando a média de um cluster é definível. Isto pode impedir a sua utilização em algumas aplicações, pois quando os atributos são nominais, torna-se muito difícil de usar. A necessidade de especificar os k clusters, também pode impedir a sua aplicação. O método k-means não é adequado para a descoberta de clusters com formas não convexas ou de tamanho muito diferente. Além disso, é sensível ao ruído e a objetos *outliers* porque podem influenciar substancialmente o valor médio.

4.1.4 K-medoids

Como já foi referido, o algoritmo k-means é sensível a *outliers*, pois os objetos com um valor extremamente grande podem distorcer substancialmente a distribuição de dados. Este efeito é particularmente agravado devido à utilização da função do erro quadrático (4.14).

Para resolver este problema, modificou-se o algoritmo de forma a diminuir a sensibilidade a *outliers*. Em vez de se tomar o valor médio dos objetos (centróide) como referência do cluster, selecionam-se objetos reais para representar os clusters. Após a seleção dos objetos que serão o ponto de referência de forma aleatória, a próxima fase será colocar cada objeto restante no cluster cujo objeto representativo lhe está mais próximo. O método de divisão baseia-se na minimização da soma das diferenças entre cada objeto e o seu ponto de referência correspondente. Para que tal seja possível é usada a função do erro absoluto, definida por:

$$E = \sum_{j=1}^{k} \sum_{p \in C_j} |p - o_j|, \tag{4.15}$$

em que E é o resultado da soma do erro absoluto para todos os objetos no conjunto de dados, p é o ponto no espaço que representa um determinado objeto no cluster C_j e o_j é o objeto representativo, de ora avante designado por medóide, de C_j.

Analisemos de uma forma mais aprofundada o k-medoids. Os medóides iniciais (ou sementes) são escolhidos arbitrariamente. O processo iterativo de substituição dos medóides por outros objetos continua enquanto a qualidade do clusters resultante é melhorada. Esta qualidade é estimada recorrendo a uma função de custo que mede a dissimilaridade média entre um objeto e o medóide do seu cluster. Para determinar se um objeto não medóide, o_{random} , é um bom substituto para o actual medóide, o_j , os quatro casos seguintes são examinados para cada um dos objetos não medóides, p.

- Caso 1: p atualmente pertence ao cluster representado pelo medóide, o_j. Se o_j for substituído por o_{random} como novo medóide, e p está mais próximo de um dos outros medóides,

o_i ,$i \neq j$, então p é atribuído ao cluster representado pelo medóide o_i .

- Caso 2: p atualmente pertence ao cluster representado pelo medóide, o_j. Se o_j for substituído por o_{random} como novo medóide, e p está mais próximo de o_{random}, então p é atribuído ao cluster representado pelo medóide o_{random}.

- Caso 3: p atualmente pertence ao cluster representado pelo medóide, o_i e $i \neq j$. Se o_j for substituído por o_{random} como novo medóide, e p continua mais próximo de o_i, então não há alterações.

- Caso 4: p atualmente pertence ao cluster representado pelo medóide, o_i e $i \neq j$. Se o_j é substituído por o_{random} como novo medóide, e p está mais próximo de o_{random} , então p é atribuído ao cluster representado pelo medóide o_{random}.

Cada vez que ocorre uma mudança, o erro absoluto, E, é recalculado com o candidato a novo medóide. O custo total da troca é a soma dos erros para todos os objetos. Se o custo total é negativo, então o_j é substituído por o_{random} uma vez que o erro absoluto E é reduzido. Se o custo total for positivo, o medóide, o_j é considerado aceitável, e nada será alterado na iteração.

Para a melhor compreensão do funcionamento do k-medoids, é apresentado um resumo do algoritmo abaixo[21].

Input:

- k: o número de clusters,

- D: conjunto dados que contém n objetos.

Output: Um conjunto de k clusters.
Método:

1. arbitrariamente escolhe-se k objetos de D como medóides iniciais;

2. **repetir**

3. atribui-se cada objeto ao cluster cujo medóide está mais próximo do objeto;

4. selecionar aleatoriamente um objeto que não seja medóide, o_{random};

5. calcular o custo total, S, de trocar o medóide, o_j , por o_{random};

6. se $S < 0$ então troca-se o_j por o_{random} para formar o novo conjunto de k medoides;
7. **até** que não ocorra nenhuma mudança;

Comparando o k-medoids com o k-means, é possível afirmar que o k-medoids é mais robusto do que o k-means na presença de ruído e de *outliers*, porque um medóide é menos influenciável por valores extremos do que um centróide. No entanto, o custo de processamento é mais elevado do que o método k-means. Ambos os métodos requerem que o utilizador especifique o número de clusters (k) o que, por vezes, pode ser visto como uma desvantagem.

4.2 Regras de Associação

Uma das técnicas de *data mining* mais vulgarmente utilizadas no comércio eletrónico é a descoberta de regras de associação entre um conjunto de produtos correlacionados. Essencialmente, esta técnica procura descobrir associações entre conjuntos de produtos de tal modo que a presença de alguns produtos numa determinada transação implica que os produtos provenientes do outro conjunto também estejam presentes na mesma transação. A qualidade das regras de associação é normalmente avaliada, consoante o seu suporte e confiança. Por outras palavras, tenta-se encontrar padrões frequentes, associações, correlações ou estruturas ocasionais em conjuntos de dados.

O suporte é uma medida que mede a frequência com que os elementos surgem no conjunto de dados.

A confiança é uma medida de certeza que corresponde à percentagem de ocorrências do antecedente juntamente com o consequente.

Uma regra que tem um alto nível de confiança é frequentemente muito importante, pois ela fornece uma previsão exata do resultado em questão. O suporte de uma regra também é importante, uma vez que as regras com muito baixo suporte são muitas vezes desinteressantes, uma vez que eles não descrevem suficientemente a população[24].

4.2.1 Descoberta de Padrões Frequentes

Os padrões frequentes são padrões (conjuntos de itens, subsequências ou subestruturas) que aparecem num conjunto de dados com frequência.

Por exemplo, o leite e o pão, que aparecem com frequência juntos num conjunto de dados de transação, constituem um conjunto de itens frequentes.

Se considerarmos a compra dum computador, em seguida uma câmara digital, e por fim um cartão de memória, se esta sequência ocorrer com frequência numa base de dados de compras, então é um padrão sequencial frequente.

Uma subestrutura pode referir-se a formas estruturais diferentes, tais como sub-árvores, sub-grafos ou sub-redes, que podem ser combinadas com conjuntos de itens ou subsequências. Se uma subestrutura ocorrer com frequência, ela é denominada de padrão estruturado frequente. Encontrar tais padrões frequentes desempenha um papel essencial na descoberta de regras de associação, correlações e muitos outros relacionamentos interessantes entre os dados. Além disso, ajuda na classificação de dados, clustering, e outras tarefas de data mining.

Suponha-se que $I = \{I_1, I_2, \ldots, I_m\}$ é um conjunto de itens e D é o conjunto de transações da base de dados onde cada transação T é um conjunto de itens de tal modo que $T \subseteq I$. Cada transação está associada a um identificador, denominado TID. Seja A um conjunto de itens. Uma transação T contém A se e só se $A \subseteq T$. Uma regra de associação é uma implicação da forma $A \Rightarrow B$, onde $A \subseteq I$, $B \subseteq I$, e $A \cap B = \phi$. A regra $A \Rightarrow B$ é obtida no conjunto de transações D com suporte s, onde s é a percentagem de transações em D que contêm $A \cup B$, ou seja, as transações que contenham todos os elementos dos conjuntos A e B. Isto é exatamente a probabilidade $P(A \cup B)$. A regra $A \Rightarrow B$ tem confiança c no conjunto de transação D, onde c é a percentagem de transações em D contenham A que também contêm B. Isto é a probabilidade condicional, $P(B|A)$. Dados tais factos pode-se dizer que,

$$suporte(A \Rightarrow B) = P(A \cup B), \tag{4.16}$$

$$confianca(A \Rightarrow B) = P(B|A). \tag{4.17}$$

As regras que satisfazem o limite mínimo de suporte (sup_min) e o limiar mínimo de confiança (conf_min) são denominadas de regras fortes. Por convenção, escrevem-se os valores do suporte e da confiança de modo a ocorrer entre 0% e 100%, ou 0 e 1,0.

Um conjunto de itens é referido como um itemset (termo utilizado na literatura de investigação, onde "itemset" onde é mais utilizado do que conjunto de itens). Um itemset que contém k items é um k-itemset, por exemplo o conjunto $\{$computador, antivírus$\}$ é um 2-itemset. A frequência de ocorrência dum itemset é dado pelo número de transações que contêm o conjunto de itens, também denominado simplesmente de frequência, contador do suporte ou contador do itemset. É de notar que o suporte definido na equação (4.16) é por vezes referido como o suporte relativo, ao passo que a frequência de ocorrência é denominada de suporte absoluto. Se o suporte relativo

dum itemset I satisfaz um limiar mínimo de suporte, pré-especificado, (ou seja, o suporte absoluto de I satisfaz o limite de frequência mínima correspondente), então I é um conjunto de itens frequentes. O conjunto k-itemset frequente é geralmente denotada por L_k. A partir da equação (4.17), tem-se:

$$confianca(A \Rightarrow B) = P(B|A) = \frac{suporte\_count((A \cup B)}{suporte\_count((A)} = \frac{suporte(A \cup B)}{suporte(A)}. \quad (4.18)$$

A equação (4.18) mostra que a confiança da regra $A \Rightarrow B$ pode ser facilmente deduzida a partir da frequência de A e $A \cup B$, isto é, uma vez encontradas as frequências de A, B e $A \cup B$, é simples derivar as regras de associação correspondentes $A \Rightarrow B$ e $B \Rightarrow A$ e verificar se elas são fortes. Assim, o problema de descobrir regras de associação pode ser reduzido à extração de conjuntos de itens frequentes.

Em geral, a descoberta de regras de associação pode ser vista como um processo de duas etapas:

1. Encontrar todos os conjuntos de itens frequentes: por definição, cada um desses conjuntos de itens irá ocorrer, pelo menos, tão frequentemente quanto uma frequência de ocorrências mínima, predeterminada, denotada por min_sup (suporte mínimo).

2. Gerar regras de associação fortes a partir dos conjuntos de itens frequentes: por definição, essas regras devem satisfazer o suporte mínimo e confiança mínima, critérios definidos a priori.

Um dos maiores desafios na descoberta de itemsets frequentes de um grande conjunto de dados é o facto de que tal descoberta, muitas vezes, gera um grande número de conjuntos de itens que satisfazem o suporte mínimo (min_sup), especialmente quando o min_sup é definido como um valor baixo. Esta situação verifica-se pois se um conjunto de itens é frequente, cada um dos seus sub-grupos é também frequente. Um itemset longo conterá um número de sub-conjuntos frequentes mais curtos, recorrendo as diferentes combinações possíveis entre os elementos do itemset para formar tais sub-conjuntos. Por exemplo, um conjunto de itens frequente com 100 elementos, $\{a_1, a_2, \ldots, a_{100}\}$, contém $\binom{100}{1}$ = 100 1-itemsets frequentes: $a_1, a_2, ..., a_{100}$. O número total de sub-conjuntos $\binom{100}{2}$ 2-itemsets frequentes: $(a_1, a_2), (a_1, a_3), ..., (a_{99}, a_{100})$, e assim por diante. O número total de sub-conjuntos frequentes é de,

$$\binom{100}{1} + \binom{100}{2} \cdots + \binom{100}{100} = 2^{100} - 1 \approx 1.27 \times 10^{30} \quad (4.19)$$

Este é um número muito grande de conjuntos para qualquer computador calcular ou armazenar. Para superar esta dificuldade, utiliza-se os conceitos de itemset frequente fechado e itemset de frequência máxima.

Um itemset X é fechado num conjunto de dados S se não existe um super-itemset Y adequado, de tal modo que Y tem a mesma frequência de ocorrência de X dentro de S. Um itemset X é itemset frequente fechado no conjunto S se X é ao mesmo tempo fechado e frequente em S. Um itemset X é um itemset de frequência máxima (ou max-itemset) no conjunto S se X é frequente, e não existe nenhum super-itemset Y de tal modo que $X \subset Y$ e Y é frequente em S.

Suponha-se que C é o conjunto de itemsets frequentes fechados num conjunto de dados S que satisfazem o limiar mínimo de suporte, min_sup. Suponha-se que M é o conjunto de frequência máxima em S que satisfaz o min_sup. Suponha-se que se tem a frequência de ocorrências de cada itemsets em C e M. Note-se que C e a sua informação de frequência podem ser utilizados para obter o conjunto completo de itemsets frequentes. Assim, dizemos que C contém informações completas sobre os seus itemsets frequentes. Por outro lado, M regista apenas o suporte absoluto para o itemset de frequência máxima.

No entanto, geralmente não são contidas as informações completas sobre o suporte de cada itemset frequente correspondente. Ilustram-se esses conceitos com o seguinte exemplo:

Imagine-se que numa base de dados de transações existem apenas duas transações: $\{\langle a_1, a_2, \ldots, a_{100}\rangle; \langle a_1, a_2, \ldots, a_{50}\rangle\}$. O limite mínimo de suporte absoluto é min_sup = 1. Encontram-se dois itemsets frequentes fechados e as suas respetivas frequências de ocorrência, $C = \{\{a_1, a_2, \ldots, a_{100}\} : 1; \{a_1, a_2, \ldots, a_{50}\} : 2\}$. Neste caso, existe apenas um itemset de frequência máxima: $M = \{\{a_1, a_2, \ldots, a_{100}\} : 1\}$, pois não é possível incluir $\{a_1, a_2, \ldots, a_{50}\}$ como um itemset de frequência máxima, porque ele tem um super-conjunto frequente $\{a_1, a_2, \ldots, a_{100}\}$.

O conjunto de itemsets frequentes fechados contém informações completas sobre os itemsets frequentes. Por exemplo, a partir de C , podemos derivar:

1. $\{a_2, a_{45} : 2\}$ uma vez que $\{a_2, a_{45}\}$ é um sub-conjunto de itens do itemsets $\{a_1, a_2, \ldots, a_{50} : 2\}$;

2. $\{a_8, a_{55} : 1\}$ uma vez que $\{a_8, a_{55}\}$ não é um sub-conjunto de itens do itemsets anterior mas do itemsets $\{a_1, a_2, \ldots, a_{100} : 1\}$.

No entanto, a partir do itemsets de frequência máxima, só é possível afirmar que ambos os itemsets ($\{a_2, a_{45}\}$ e $\{a_8, a_{55}\}$) são frequentes, no entanto não é possível afirmar qual é o seu suporte real.

4.2.2 Algoritmo Apriori: Encontrar Itemsets Frequentes através da Geração de Candidatos

O algoritmo Apriori é um algoritmo seminal proposto por R. Agrawal e R. Srikant em 1994, utilizado na descoberta de itemsets frequentes, que serão usados posteriormente na descoberta de regras de associação booleanas. O algoritmo utiliza conhecimento prévio das propriedades frequentes do itemset e utiliza uma pesquisa iterativa conhecida como *level-wise search*, onde os k-itemsets são usados para descobrir os $(k+1)$-itemsets[25].

O primeiro passo é encontrar o conjunto de todos os 1-itemsets frequentes, fazendo uma procura pela base de dados e, em simultâneo, uma contagem do número de ocorrências de cada item. No final, recolhem-se os itens que satisfazem o suporte mínimo. O conjunto resultante é denotado por L_1. No próximo passo, L_1 é usado para localizar L_2, o conjunto de itemsets frequentes compostos por conjuntos de 2 itens, que é utilizado no terceiro passo, para encontrar L_3, e assim sucessivamente, até que não seja possível encontrar k-itemsets mais frequentes. Em cada iteração, para determinar o L_k correspondente, é necessário verificar por completo a base de dados.

Para melhorar a eficiência da geração *level-wise* de itemsets frequentes, é usada uma propriedade importante denominada por propriedade apriori, que é utilizada para reduzir o espaço de procura.[25]

Propriedade apriori: Todos os subconjuntos não vazios de um conjunto de itens frequentes também são frequente.

A propriedade apriori baseia-se na seguinte observação: por definição, se um conjunto de itens I não satisfaz o limiar de suporte mínimo, então I não é frequente; ou seja, $P(I) < min\_sup$. Se um item A for adicionado ao conjunto de itens I, o itemset resultante (ou seja, $I \cup A$) não pode ocorrer com maior frequência do que I. Portanto, o itemset $I \cup A$ também não é frequente, pois $P(I \cup A) < min\_sup$.

A propriedade apriori é uma propriedade anti-monótona, uma vez que se um conjunto falhar num teste, todos os seus superconjuntos falharão o mesmo teste também.

Para a melhor compreensão de como a propriedade apriori é utilizada no algoritmo, veja-se como L_{k-1} é usado para descobrir L_k para $k \geq 2$. Para tal, seguem-se os dois passos seguintes:

1. **Passo de fusão:** Para encontrar L_k, um conjunto de k-itemsets candidatos é gerado através da junção de L_{k-1} com ele próprio ($L_{k-1} \bowtie L_{k-1}$). Este conjunto de candidatos é denotado por C_k. Sejam l_1 e l_2 itemsets contidos por L_{k-1}. A notação $l_i[j]$ refere-se ao

item de l_i na posição j, por exemplo, $l_1[k-2]$ refere-se ao antepenúltimo item de l_1). Por convenção, assume-se que os itens dentro dum itemset estão ordenados segundo a sua ordem lexicográfica, ou seja, para o $(k-1)$-itemset, os itens de l_i, são ordenados da seguinte forma, $l_i[1] < l_i[2] < \cdots < l_i[k-1]$. A fusão, $L_{k-1} \bowtie L_{k-1}$, é realizada, onde os membros de L_{k-1} são aglomeráveis, ou seja, se os seus primeiros $(k-2)$ itens estão em comum. Isto é, os membros l_1 e l_2 de L_{k-1} são aglomerados se $(l_1[1] = l_2[1]) \wedge (l_1[2] = l_2[2]) \wedge \ldots \wedge (l_1[k-2] = l_2[k-2]) \wedge (l_1[k-1] < l_2[k-1])$ A condição $l_1[k-1] < l_2[k-1]$ simplesmente garante que não há duplicados. O itemset resultante da fusão de l_1 e l_2 é $l_1[1], l_1[2], \ldots, l_1[k-2], l_1[k-1], l_2[k-1]$.

2. **Passo de poda:** C_k é um superconjunto de L_k, isso é, os seus elementos podem ser ou não frequentes, mas todos os k-itemsets frequentes estão incluídos em C_k . É feita uma pesquisa na base de dados com o intuito de determinar se cada candidato em C_k satisfaz o suporte mínimo. O resultado será o conjunto de itemsets, L_k, onde estão todos os candidatos que satisfazem o suporte mínimo e por definição, são frequentes. Entretanto C_k, poderá ser enorme, o que obrigará a uma computação pesada. Para reduzir o tamanho de C_k, recorre-se à prioridade apriori. Qualquer $(k-1)$-itemset que não é frequente não pode ser um subconjunto de um k-itemset frequente. Consequentemente, se qualquer $(k-1)$-subconjunto de um k-itemset candidato não pertence a L_{k-1} , então o candidato não pode se frequente e, por isso, pode ser removido de C_k . Este teste pode ser feito rapidamente mantendo uma hash tree com todos os itemsets frequentes.

Exemplo

Para uma melhor perceção do exposto até agora, analise-se o seguinte exemplo, tendo como referência a base de dados de transações de itens, D, da tabela 4.2. Aqui é possível encontrar nove transações, ou seja, a cardinalidade é de, $|D| = 9$. A figura 4.2 ilustra, de forma esquemática, como o algoritmo apriori encontra os itemsets frequentes em D.

1. Na primeira iteração do algoritmo, todos os itens são membros do conjunto de candidatos 1-itemsets, C_1 . O algoritmo simplesmente percorre todas as transações e vai contando o número de ocorrências de cada item.

2. Suponha-se que o suporte absoluto mínimo requerido é 2, ou seja, $min\_sup = 2$ (corresponde a um suporte relativo mínimo de 2/9 = 22%). O conjunto de 1-itemsets frequentes, L_1, é constituído pelos 1-itemsets candidatos que satisfazem o suporte mínimo. No exemplo, todos os candidatos contidos em C_1 satisfazem o suporte mínimo, logo $L_1 = C_1$.

TID	Lista de item IDs
T100	I1, I2, I5
T200	I2, I4
T300	I2, I3
T400	I2, I4
T500	I1, I3
T600	I2, I3
T700	I1, I3
T800	I1, I2, I3, I5
T900	I1, I2, I3

Tabela 4.2: Base de dados de transações de itens

3. Para descobrir o conjunto de 2-itemsets frequentes, L_2 , o algoritmo utiliza a junção $L_1 \bowtie L_1$ para gerar um conjunto de 2-itemsets candidatos, C_2. De notar que não há candidatos removidos de C_2 durante o passo de poda, porque cada subconjunto dos candidatos também é frequente.

4. De seguida, percorre-se novamente D e conta-se o numero de ocorrências de cada itemset candidato pertencente a C_2, como mostra a tabela do meio na segunda linha da figura 4.2

5. O conjunto de 2-itemsets frequentes, L_2 , é formado por todos os 2-itemsets candidatos, pertencentes em C_2, que satisfazem o suporte mínimo especificado.

6. A geração do conjunto de 3-itemsets candidatos, C_3, está detalhada a seguir. Através do passo de fusão, obtém-se $C_3 = L_2 \bowtie L_2 = \{\{I1, I2, I3\}, \{I1, I2, I5\}, \{I1, I3, I5\}, \{I2, I3, I4\}, \{I2, I3, I5\}, \{I2, I4, I5\}\}$. Com base na propriedade apriori, sabe-se que todos os sub-conjuntos dum itemset frequente também são frequentes, o que permite perceber que quatro dos candidatos não podem ser frequentes, e portanto podem ser removidos de C_3, poupando o esforço de procura em D de mais quatro itemsets e assim, determina-se $L3$ mais rapidamente. De notar que, dado um k-itemset candidato, apenas é necessário verificar se os $(k-1)$-subconjuntos são frequentes, uma vez que, o algoritmo apriori utiliza uma estratégia de pesquisa *level-wise*. O C_3 resultante do passo de poda é apresentado no primeiro quadro da linha do fundo da figura 4.2

a) Fusão: $C_3 = L_2 \bowtie L_2 = \{\{I1, I2\}, \{I1, I3\}, \{I1, I5\}, \{I2, I3\}, \{I2, I4\}, \{I2, I5\}\} \bowtie \{\{I1, I2\}, \{I1, I3\}, \{I1, I5\}, \{I2, I3\}, \{I2, I4\},$

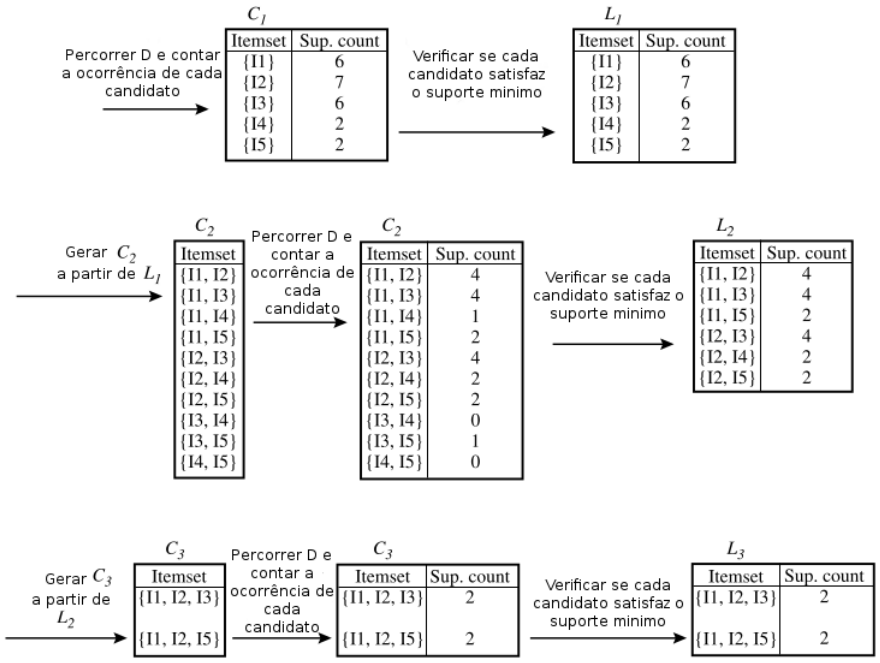


Figura 4.2: Geração de itemsets candidatos e descoberta de itemsets frequentes, quando o suporte absoluto é de 2

$\{I2, I5\}\}$ = $\{\{I1, I2, I3\}, \{I1, I2, I5\}, \{I1, I3, I5\}, \{I2, I3, I4\}, \{I2, I3, I5\}, \{I2, I4, I5\}\}$.

b) Podar usando a propriedade apriori: todos os subconjuntos não vazios de um itemset frequente também devem ser frequentes. Será que algum dos candidatos tem um subconjunto que não é frequente?

- Os subconjuntos de cardinalidade 2 (2-subconjuntos) de $\{I1, I2, I3\}$ são $\{I1, I2\}$, $\{I1, I3\}$ e $\{I2, I3\}$. Todos os 2-subconjuntos de $\{I1, I2, I3\}$ são membros de L_2, portanto, mantêm-se $\{I1, I2, I3\}$ em C_3 .
- Os 2-subconjuntos de $\{I1, I2, I5\}$ são $\{I1, I2\}$, $\{I1, I5\}$, e $\{I2, I5\}$. Mais uma vez, todos os 2-subconjuntos de $\{I1, I2, I5\}$ são membros de L_2, logo, mantêm-se $\{I1, I2, I5\}$ em C_3.
- Os 2-subconjuntos de $\{I1, I3, I5\}$ são $\{I1, I3\}$, $\{I1, I5\}$, e $\{I3, I5\}$. O subconjunto, $\{I3, I5\}$, não é membro de L_2 , e por consequência, não é frequente, logo remove-se $\{I1, I3, I5\}$ de C_3.

- Os 2-subconjuntos de $\{I2, I3, I4\}$ são $\{I2, I3\}$, $\{I2, I4\}$, e $\{I3, I4\}$. O subconjunto $\{I3, I4\}$ não é membro de L_2 , ou seja, não é frequente, e $\{I2, I3, I4\}$ é removido de C_3.
- Os 2-subconjuntos de $\{I2, I3, I5\}$ são $\{I2, I3\}$, $\{I2, I5\}$, e $\{I3, I5\}$. O subconjunto $\{I3, I5\}$ não é frequente, pois não é membro de L_2, portanto, remove-se $\{I2, I3, I5\}$ de C_3.
- Os 2-subconjuntos de $\{I2, I4, I5\}$ são $\{I2, I4\}$, $\{I2, I5\}$, e $\{I4, I5\}$. O subconjunto $\{I4, I5\}$ não é membro de L_2 , o que indica que não é frequente, assim, remove-se $\{I2, I4, I5\}$ de C_3.

c) No final do processo de poda, o C_3 resultante será, $C_3 = \{\{I1, I2, I3\}, \{I1, I2, I5\}\}$.

7. As transações em D são percorridas com o intuito de determinar L_3 , verificando quais os 3-itemsets candidatos em C_3 que satisfazem os suporte mínimo (Figure 4.2).

8. O algoritmo utiliza $L_3 \bowtie L_3$ para gerar o conjunto de candidatos 4-itemsets, C_4 . Embora o resultado da fusão seja o itemset, $\{\{I1, I2, I3, I5\}\}$, sera removido no passo de poda, uma vez que o sub-conjunto $\{\{I2, I3, I5\}\}$ não é frequente. Sendo assim, $C_4 = \phi$, e por consequência, o algoritmo termina, encontrando todos os itemsets frequentes.

No próximo quadro, é apresentada uma implementação do algoritmo Apriori, recorrendo a pseudo código

Input:

- D, base de dados de transações;
- min_sup, limite de suporte mínimo.

Output: L, itemsets frequentes em D.
Método:

```
1 Set apriori(data_base D, float mun_sup)
2 {
3     Set L1;
```

```
4
5       L_1 = find_frequent_1-itemsets(D);
6       for( k = 2; L_{k-1} = φ ;k++){
7           C_k = apriori_gen(L_{k-1});
8           foreach (transaction ∈ D) { // scan D for counts
9               C_t = subset(C_k, transaction); // get the subsets of
                    transaction that are candidates
10              foreach candidate ∈ C_t{
11                  candidate.count++;
12              }
13              L_k = {candidate ∈ C_k |c.count ≥ min_sup}
14          }
15      }
16      return L = ∪_k L_k;
17  }
18
19  procedure apriori_gen(L_{k-1}){ //request (k-1)-itemsets
20      foreach (itemset l_1 ∈ L_{k-1})
21          foreach (itemset l_2 ∈ L_{k-1}
22              if (l_1[1] = l_2[1]) ∧ (l_1[2] = l_2[2]) ∧ ··· ∧ (l_1[k − 2] = l_2[k − 2])
23              ∧(l_1[k − 1] < l_2[k − 1]) then \{
24                  c = l_1 ⋈ l_2 ; // join step: generate candidates
25                  if has infrequent subset(c, L_{k-1} ) then
26                      delete c; // prune step: remove unfruitful
                            candidate
27                  else add c to C k ;
28              }
29      return C_k;
30  }
31  procedure has infrequent subset(c /*candidate k-itemset*/, L_{k-1}
        /*frequent (k-1)-itemsets*/){ // use prior knowledge
32      foreach (k-1)-subset s of c
33          if s ∈ L_{k-1} then
34          return TRUE;
35      return FALSE;
```

```
36 }
```

O quadro acima apresenta pseudo código relativo ao algoritmo Apriori. O primeiro passo do algoritmo consiste em encontrar os 1-itemsets frequentes, L_1. Do passo 2 até ao 10, L_{k-1} é usado para gerar candidatos C_k de forma a encontrar L_k para $k \geq 2$. O método **apriori_gen** gera os candidatos e usa a propriedade apriori para eliminar candidatos que têm pelo menos um subconjunto que não é frequente. Uma vez gerados todos os candidatos, a base de dados é varrida. Para cada operação, a função **subset** é usada para encontrar todos os subconjuntos da transação que são candidatos (que estão presentes em C_k) e a contagem da frequência de cada um destes candidatos é acumulada. Finalmente, todos os candidatos que satisfazem o suporte mínimo são colocados no conjunto de itemsets frequentes, L. O L resultante pode ser utilizado na geração de regras de associação a partir dos conjuntos de itemsets frequentes.

O método **apriori_gen** efetua ambos os passos, tanto o passo de fusão como o passo de poda. Na componente de junção, procede-se à junção natural do conjunto de itemsets frequentes da etapa anterior, L_{k-1}, com ele mesmo, ou seja, $L_{k-1} \bowtie L_{k-1}$. Deste modo, são gerados os potenciais candidatos. Na componente de poda, é utilizada a propriedade apriori para remover os candidatos que têm pelo menos um subconjunto que não é frequente. O teste que verifica se os candidatos são ou não removidos fica a cargo do método **infrequent_subset**.

4.2.3 Geração de Regras de Associção a partir de Itemsets Frequentes

Uma vez encontrados os itemsets frequentes através da análise da base de dados D, é muito simples gerar regras de associação fortes a partir destes itemsets. Para que as regras associação sejam fortes, têm de satisfazer tanto o suporte mínimo como a confiança mínima. Caso não satisfaçam um ou os dois limites mínimos, serão descartadas. Para se determinar a confiança, recorre-se à equação (4.18), que se volta a relembrar aqui:

$$confianca(A \Rightarrow B) = P(B|A) = \frac{suporte\_count(A \cup B)}{suporte\_count(A)}.$$

A probabilidade condicional é expressa em termos do suporte absoluto do itemsets , onde $(A \cup B)$ é o numero de transações que contém o itemset $A \cup B$, e $suporte\_count(A)$ é o numero

de transações que contém o itemset A. Com base nesta equação, as regras de associação podem ser geradas da seguente forma:

- Para cada itemset frequente, l, gerar todos os subconjunto não vazios;
- Para todos os subconjuntos gerados, s, obtém-se a regra$s \Rightarrow (l - s)$ se $\frac{suporte\_count(l)}{suporte\_count(s)} \geq min\_conf$, onde min_conf é o limite mínimo para a confiança.

Como as regras são gerados a partir de itemsets frequentes, elas satisfazem automaticamente o suporte mínimo. Os itemsets frequentes podem ser armazenados em tabelas hash juntamente com o seu suporte relativo e suporte absoluto, o que permite um acesso mais rápido, otimizando assim a geração de regras de associação.

Para perceber melhor a geração de regras de associação, volta-se a analisar a tabela 4.2, utilizada no seguinte exemplo:

Suponha-se que, após a procura de itemsets frequentes, o resultado obtido foi o itemset l = {I1, I2, I5}. Para se poder gerar as regras de associação, é necessário encontrar primeiro todos os subconjuntos não vazios de l. Os subconjuntos não vazios de l são {I1, I2}, {I1, I5}, {I2, I5}, {I1}, {I2}, e {I5}. Como é sabido, graças à propriedade apriori todos os subconjuntos de um conjunto frequente também são frequentes, e por este motivo todos eles satisfazem o suporte mínimo. As regras de associação resultantes possíveis são:

I1 $\wedge$ I2 $\Rightarrow$ I5, confiança = 2/4 = 50%
I1 $\wedge$ I5 $\Rightarrow$ I2, confiança = 2/2 = 100%
I2 $\wedge$ I5 $\Rightarrow$ I1, confiança = 2/2 = 100%
I1 $\Rightarrow$ I2 $\wedge$ I5, confiança = 2/6 = 33%
I2 $\Rightarrow$ I1 $\wedge$ I5, confiança = 2/7 = 29%
I5 $\Rightarrow$ I1 $\wedge$ I2, confiança = 2/2 = 100%

Se o limite mínimo de confiança for de 70%, as únicas regras de associação consideradas fortes são a segunda, a terceira e a última, pois são as únicas a satisfazer o limite mínimo no que toca à confiança, e por isso serão guardadas. As outras que não satisfazem o limite mínimo de confiança são descartadas.

4.3 Sistemas de Recomendação

Um sistema de recomendação combina várias técnicas computacionais para selecionar itens personalizados com base nos interesses dos utilizadores e conforme o contexto no qual estão inseridos. Os itens podem variar entre livros, filmes, notícias, músicas, vídeos, anúncios, links patrocinados, páginas de internet, produtos de uma loja virtual, entre outros. Empresas como Amazon, Netflix e Google são reconhecidas pelos ótimos sistemas de recomendação. Nos dias de hoje, são bastante utilizados no comércio eletrónico[24]. Os sistemas de recomendação fazem sugestões sobre determinados artigos a um utilizador. Por exemplo, os sistemas de recomendação podem prever que um utilizador está interessado em ver um filme. Pode ser utilizada a técnica dos vizinhos mais próximos para fazer recomendações ao utilizador com base nas suas últimas visualizações.[26]

4.3.1 K-Nearest Neighbors(k-NN)

O k-Nearest Neighbors (k-NN) é um dos métodos mais antigos e mais simples de classificação de padrões. Frequentemente produz resultados competitivos combinados com conhecimento prévio. o seu desempenho depende da métrica de distância utilizada para identificar os k vizinhos mais próximos.

Na ausência de conhecimento prévio, a maior parte das implementações do kNN usa a distância euclidiana para medir a diferença entre objetos, representando por um vetor de variáveis que compõem o objeto em análise.[27]

O k-Nearest Neighbors foi descrito pela primeira vez no início dos anos cinquenta do século passado, mas como o k-NN é um método que exige um poder computacional considerável para um conjunto de treino de grandes dimensões, só ganhou popularidade nos anos sessenta, quando os computadores alcançaram as exigências computacionais que o k-NN necessita. Desde então, tem sido amplamente utilizado na área de reconhecimento de padrões.

O k-NN baseia-se na aprendizagem por analogia, ou seja, através da comparação de um dado objeto com objetos do conjunto de treino que lhe são similares. Estes objetos são descritos por n atributos, onde cada objeto representa um ponto num espaço n-dimensional. Isto permite guardar todos os objetos de treino num espaço n-dimensional. Dado um objeto desconhecido, z, o k-NN procura no espaço pelos k objetos mais próximos de z, ou seja, estes k objetos são os k vizinhos mais próximos de z. A proximidade é defendida graças a uma função de distância, esta função é defendida através da distância euclidiana, onde a distancia euclidiana entre dois objetos, X_1 e X_2, onde, $X_1 = (x_{11}, x_{12}, ..., x_{1n})$ e $X_2 = (x_{21}, x_{22}, ..., x_{2n})$, é dada pela equação 4.20:

$$dist(X_1, X2) = \sqrt{\sum_{i=1}^{n}(x_{1i} - x_{2i})^2}. \tag{4.20}$$

Traduzindo para palavras a equação 4.20, para cada atributo, é calculada a diferença entre os valores correspondentes desse atributo no objeto X_1 e no objeto X_2, calculando o quadrado da diferença e acumulando o resultado. Por fim, é calculada a raíz quadrada do total acumulado.

Para um cálculo mais preciso, os atributos devem ter os seus valores normalizados antes de se proceder ao cálculo da distância euclidiana. Isto ajuda a prevenir que atributos com intervalos de valores inicialmente maiores influenciem mais a distância que atributos com intervalos inicialmente menores. Por exemplo, sem normalização, o atributo de idade influenciaria bastante a função de distância quando comparado com o atributo género. A normalização do valor numérico v dum atributo A para o valor v', no intervalo [0, 1], poderia ser feita da seguinte forma:

$$v' = \frac{v - min_A}{max_A - min_A}, \tag{4.21}$$

onde max_A e min_A representam o valor máximo e o mínimo que o atributo A pode tomar.

O k-NN atribui o objeto desconhecido à classe mais comum entre os seus k vizinhos mais próximos. Quando k = 1, o objeto desconhecido é atribuído à classe do objeto mais próximo no espaço composto por todos os objetos do conjunto de treino. O método k-NN também pode ser utilizado na previsão do valor dum determinado objeto desconhecido. Neste caso, o resultado da classificação é o valor médio das classificações dos k vizinhos mais próximos do objeto desconhecido.

No caso do atributo ser nominal e não numérico, uma forma simples de resolver o problema é comparar o valor correspondente do atributo no objeto X_1 com o correspondente no objeto X_2. Se são iguais, então a distância entre os dois é considerada 0, se os dois são diferentes, então a distância é considerada 1. Para a melhor compreensão, suponha-se que existem dois objetos, X_1 e X_2, onde o atributo a ser comparado é a cor. Se X_1 e X_2 são ambos azuis, então a distância entre eles, no que toca ao atributo cor, é de 0. Por outro lado, se X_1 é azul e X_2 é vermelho, a distância é 1.

Outros métodos mais sofisticados podem ser usados para resolver este problema, como por exemplo, a criação de uma matriz de distância entre os elementos, sendo que a cor vermelha está mais próxima da cor magenta do que da cor azul, por esemplo. Em última instância, é possível utilizar uma função de conversão: ao tomar o código RGB da cor, coloca-se cada cor num eixo (vermelho, verde, azul) e utiliza-se a distância euclidiana para medir a distância entre as cores,

sendo que é extremamente aconselhável que esta distância esteja compreendida no intervalo[0, 1].

Outro problema que pode ocorrer é o valor estar em falta no atributo em análise num ou noutro elemento em analise, ou até mesmo em ambos. Quando tal acontecer, assume-se que a distância é a máxima possível. Suponha-se que cada um dos atributos foi mapeado no intervalo de [0, 1]. Caso o atributo em análise, A, seja nominal, toma-se o valor da diferença como sendo 1, se qualquer um dos objetos, X_1 e X_2, ou ambos tem o valor de A em falta. Se A for um atributo numérico em falta tanto no objeto X_1 como no objeto X_2, a diferença entre eles também é será de 1. Mas se o valor só estiver apenas em falta num dos objetos e no outro estiver presente e normalizada, então calcula-se a distância como sendo igual a $|1 - v'|$ ou $|0 - v|$, escolhendo o resultado que apresenta o maior valor, onde v' é o valor presente e normalizada.

Determinar o valor de k é uma tarefa iterativa, começando com $k = 1$ e utilizando o conjunto de teste para testar a taxa de erro da classificação. Incrementa-se k, aumentando o número de vizinhos, e volta-se a testar a taxa de erro, até que esta esteja no limite desejado. O valor k é fornecido pela iteração que minimiza o valor da taxa de erro mínima. Em geral, quanto maior for o número de objetos do conjunto de treino, maior será o valor de k. À medida que o número de objetos do conjunto da treino se aproximado infinito e k = 1, a taxa de erro não poderá ser pior do que o dobro do valor mínimo teórico da taxa de erro (taxa de erro de Bayes). Se k também se aproximar do infinito, então a taxa de erro aproxima-se da taxa de erro de Bayes[28].

5 Trabalhos Relacionados

5.1 Comércio eletrónico

Os maiores sites de comércio eletrónico oferecem milhões de produtos para venda. Escolher entre tantas opções é um desafio para os consumidores. Os sistemas de recomendação surgiram em resposta a este problema. Um sistema de recomendação para um site de comércio eletrónico recebe informações de um consumidor sobre os produtos em que ele está interessado e recomenda produtos que são adequados às suas necessidades. Hoje em dia, os sistemas de recomendação são implantados em centenas de sites diferentes, servindo milhões de consumidores. Estes necessitam de recomendações em que podem confiar e que os ajudam a encontrar produtos que gostam. Os sistemas de recomendação, como outros sistemas de busca, têm dois tipos de erros característicos: falsos negativos, ou seja, produtos que não são recomendados, embora o consumidor tenha interesse neles; e falsos positivos, que são produtos recomendados, embora o consumidor não goste deles. Se o consumidor confiar num sistema de recomendação, compra um produto sugerido sem pensar muito no assunto, mais tarde percebe que na realidade não gosta do produto e será pouco provável que volte a utilizar o sistema de recomendação. No domínio do comércio eletrónico, os erros mais importantes são os falsos positivos, uma vez que irão deixar os consumidores irritados. Como geralmente a oferta de produtos num site de comércio eletrónico é grande, o problema de falsos negativos não é grave, desta forma não há razão para arriscar recomendar algo que o consumidor não goste. Em alguns aspetos, estes dois desafios estão em conflito, uma vez que quanto menos tempo um algoritmo gasta na procura de recomendações, mais escalável poderá ser, mas irá piorar a sua qualidade de recomendação. Por estas razões, é importante tratar os dois desafios em simultâneo de modo que as soluções encontradas sejam úteis e práticas. Tal como pode ser visto no artigo *Analysis of Recommendation Algorithms for E-Commerce*[24], investigam-se várias técnicas para análise de compra em grande escala e dados de preferências, com a finalidade de produzir recomendações úteis para os clientes. Em particular, aplica-se um conjunto de algoritmos e técnicas tradicionais de *data mining*, com o intuito de reduzir a dimensão do problema, tentando responder ao desafio de

melhorar a qualidade das recomendações.

5.2 Movie-Map

O Movie-Map é um site de recomendação de filmes, que recomenda aos seus utilizadores filmes, com base num mapa que foi previamente construído, utilizando a técnica do vizinho mais próximo. O site é um projeto da Gnod, que por sua vez é um projeto de Marek Gibney. Além do site[29], não foi encontrada mais informação acerca deste projeto.

5.3 MovieLens

O MovieLens é um sistema de recomendação online, que convida os seus utilizadores a classificar filmes, numa escala de meia estrela até cinco estrelas. Em troca, faz recomendações personalizadas de filmes que o utilizador ainda não classificou, e prevê o resultado que o utilizador daria a estes filmes.

O MovieLens é um projeto do grupo de investigação GroupLens, este grupo faz parte do Departamento de Ciências da Computação e Engenharia da Universidade de Minnesota.

O MovieLens é um dos sites de recomendação de filmes sem fins lucrativos mais populares do mundo. Prova disso. são as noticias publicadas no *The New York Times*, *ABC News Nightline* e *The New Yorker*. A 30 de Abril de 2006, já contava com 13 milhões classificações divididas por 9043 filmes. Essas classificações provinham de pouco mais de 100 mil utilizadores, onde cerca de 15000 utilizavam o site com regularidade[30].

Para determinar quais as melhores recomendações para os seus utilizadores, o MovieLens utiliza filtros colaborativos que analisam as opiniões do utilizador e de toda a comunidade de utilizadores, onde o pressuposto subjacente é que aqueles que concordaram no passado tendem a concordar novamente no futuro. O algoritmo combina utilizadores com opiniões similares em relação aos filmes que classificaram, criando uma ”vizinhança” de utilizadores com uma mente semelhante. As recomendações personalizadas são geradas a partir desta vizinhança.

O MovieLens encoraja os seus utilizadores a classificar os filmes que eles viram, ajudando a melhorar o perfil, o que permite melhores recomendações, uma vez que o algoritmo dispõe de mais informação sobre o utilizador. Esta prática também adiciona mais informação à base de dados o que ajuda a melhorar a globalidade do sistema [30].

5.4 Youtube

O YouTube foi fundado por Chad Hurley, Steve Chen e Jawed Karim, ex-funcionários da PayPal, tendo sido lançado oficialmente em junho de 2005.

O YouTube tentava eliminar as barreiras técnicas para a maior parte dos utilizadores que queriam partilhar os seus vídeos online. Nessa altura, era disponibilizada uma interface bastante simples e integrada, dentro da qual o utilizador podia fazer o *upload*, publicar e assistir vídeos em *streaming* sem necessidade de grande conhecimento técnico. O YouTube não estabeleceu limites para o número de vídeos que cada utilizador poderia colocar online, apenas um limite máximo de duração para cada vídeo. Oferecia também funções básicas de partilha, dando a possibilidade do utilizador e gerava URLs e códigos HTML que permitiam que os vídeos fossem facilmente colocados noutros sites.

Em outubro de 2006, a Google pagou cerca de 1,3 mil milhões de euros pelo YouTube. Em novembro de 2007, era o site de entretenimento mais popular do Reino Unido, com o site da BBC em segundo e no começo de 2008, estava entre os dez sites mais visitados do mundo. Em abril de 2008, o YouTube já hospedava 85 milhões de vídeos, um número que representa um aumento dez vezes maior em comparação ao ano anterior e que continua a crescer exponencialmente[31].

O YouTube é uma plataforma e um agregador de conteúdo, embora não seja um produtora de conteúdo em si. O YouTube tem utilizadores que produzem conteúdos com regularidade, no seu espaço ou canal de YouTube. É neste ponto que o YouTube se assemelha a televisão interativa. Para suportar os criadores de conteúdos, o YouTube divide os lucros arrecadados com a publicidade com os criadores de tais conteúdos. A principal função do YouTube é a difusão dos conteúdos e o seu respetivo armazenamento. De notar que as práticas comerciais do YouTube têm-se mostrado particularmente controversas, tanto em relação aos velhos meios de comunicação como junto de alguns dos membros mais ativos de sua comunidade. O seu sucesso é inegável, gerando milhões de euros em receitas publicitarias[32].

O YouTube cresceu rapidamente e tornou-se no site de partilha de vídeos mas popular do mundo. Os utilizadores utilizam o YouTube para descobrir, visualizar e partilhar vídeos criados pela comunidade. O sistema de recomendação do YouTube sugere aos utilizadores um conjunto de vídeos baseado na atividade recente no site, onde são analisados os vídeos vistos, os vídeos adicionados aos favoritos e os filmes que o utilizador gostou.

Para que tais recomendações sejam possíveis, todos os vídeos visualizados por todos os utilizadores são utilizados como sementes na procura dos elementos que vão formar o conjunto de vídeos recomendados. Para recomendar estes vídeos, são utilizadas regras de associação e a contagem de co-visitação. Para um determinado período de tempo, normalmente as últimas

24 horas, conta-se para cada par de vídeos (v_i, v_j) quantas vezes foram co-visionados dentro de todas as sessões. Depois de se ter obtido, o número de ocorrências dos dois vídeos, é utilizada uma função que permite calcular a popularidade global tanto da semente como do candidato a elemento do conjunto de recomendação.

Em seguida, são escolhidos os primeiros N vídeos do conjunto de vídeos candidatos, gerados a partir dum determinado vídeo semente. Este conjunto de vídeos candidatos está ordenado de acordo com a classificação da sua popularidade global. Para calcular as recomendações personalizadas, combinam-se as regras de associação geradas com a atividade do utilizador, onde são incluídos os vídeos vistos, os vídeos adicionados aos favoritos e os filmes que utilizador gostou ou adicionados a uma *playlist*.

Neste estágio, os vídeos voltam a ser ordenados segundo três parâmetros: qualidade do vídeo, especificações do utilizador e diversidade. Os vídeos com melhor classificação são apresentados ao utilizador. [33]

5.5 Netflix

No passado, a Netflix era um serviço de aluguer de filmes online que permitia aos utilizadores alugar filmes por uma mensalidade fixa. Os utilizadores disponham duma lista de prioridades de filmes que desejavam ver, isto é, lista de filmes desejados. Os filmes podiam ser enviados pelo correio ou entregues por via eletrónica através da Internet. Se o utilizador assistisse ao filme em formato DVD, quando terminasse de assistir o filme seria devolvi-d pelo correio e o próximo DVD seria automaticamente enviado por correio quando fosse recebida a devolução, sem custos acrescidos para o utilizador.

O período de subscrição estava relacionado com o número de filmes que os utilizadores assistiam e gostavam. Se os subscritores não conseguissem encontrar filmes que lhe interessavam, tinham tendência a abandonar o serviço. Perceber que filmes seriam mais de acordo com a preferência dos utilizadores era importante tanto para a empresa como para os subscritores. Com foco neste propósito, a Netflix incentivou os seus subscritores a classificar o que mais e o que menos gostavam nos filmes que assistiram. Em 2007, a Netflix tinha recolhido cerca de 1,9 mil milhões de classificações vindas de mais de 11,7 milhões de subscritores em mais de 85 mil títulos.

A Netflix recorreu ao sistema de recomendação da Cinematch para analisar a informação acumulada até então em relação aos filmes. Este sistema de recomendação fazia inúmeras previsões personalizadas aos subscritores da Netflix, por dia, com base nos seus gostos particulares.

O sistema de recomendação da Cinematch analisava automaticamente as classificações dos filmes adquiridas na última semana, utilizando uma função que media a correlação entre os filmes e utilizadores, para determinar a lista de filmes similares e prever o nível de satisfação de cada utilizador[34].

A Netflix não estava contente com os resultados do sistema de recomendação em uso e por isso, em Outubro de 2006, lançou uma competição onde o primeiro a alcançar uma redução de 10% da raíz quadrada da média do erro seria recompensado com um prémio de 1.000.000$[35].

O vencedor da competição foi o projeto BellKor's Pragmatic Chaos [36]. Para resolver o problema, o sistema de recomendação utiliza filtros colaborativos. Ao contrário dos outros algoritmos, este tenta filtrar os utilizadores que apenas dão classificações máximas a todos os filmes e os utilizadores que só dão classificações mínimas, e não se limita a criar a vizinhança dos utilizadores, usando também uma matriz com diversos fatores, como por exemplo, fator temporal das qualificações em análise, mudanças de opinião, frequência que o utilizador classifica filmes, entre outros. Após o cálculo da vizinhança, o algoritmo produz recomendações orientadas à vizinhança em análise. Esta recomendação de filmes vem acompanhada com a classificação expectável que este conjunto de utilizadores daria a cada recomendação.

6 ZenTV

O ZenTV é o protótipo desenvolvido no decorrer desta dissertação. No ZenTV, foram implementadas algumas da técnicas de inteligência artificial estudadas.

Foi implementada uma forma de obter regras de associação através da geração de candidatos a itemsets frequentes. Os utilizadores foram divididos em clusters, sendo que nesta divisão foi implementado o k-means.

O sistema de recomendação desenvolvido recorreu a uma implementação do algoritmo k-NN, onde todos os filmes que o utilizador ainda não tenha validado têm uma aceitação esperada.

Neste protótipo, também foi tomado em conta o conhecimento adquirido nas secções Interface e Avaliação de Stress, na tentativa de construir uma interface simples e que não induzisse stress no utilizador.

No desenvolvimento deste protótipo, foi utilizada a framework Ruby on Rails, o que foi um desafio, pois foi necessário aprender uma nova linguagem e implementar as técnicas de inteligência artificial já referidas numa linguagem de script. A ideia inicial era comprovar que era possível implementar facilmente estas técnicas neste tipo de linguagem.

6.1 Tecnologias Utilizadas

6.1.1 Ruby on Rails

No desenvolvimento de software, existiu, por muito tempo, uma lacuna entre as abordagens rápidas mas pouco limpas, como o PHP e as abordagens mais lentas, mas limpas e sólidas, como Java.

Muitos programadores gostariam de ter o melhor dos dois mundos: uma abordagem rápida, produtiva, que produz aplicações limpas e confiáveis mais rapidamente, usando menos código. Para colmatar este problema, surgiu o Ruby on Rails.

O Ruby on Rails é uma framework baseada em Ruby, uma linguagem de script orientada a objetos, semelhante ao Perl. O Ruby on Rails (ou apenas ”Rails”para simplificar a linguagem)

utiliza pacotes integrados e código predefinido, conhecidos como convenções, concebidos para estar completo e pronto para usar imediatamente, sem necessidade de configuração.

Os defensores de Rails dizem que é o mais adequado para a construção de infraestruturas que exibam informações de uma base de dados numa aplicação Web, tais como as que são usadas no comércio online e comunidades online.[37]

O Ruby on Rails é uma framework de desenvolvimento web escrita na linguagem de programação Ruby. Desde o seu lançamento em 2004, o Ruby on Rails rapidamente tornou-se uma das ferramentas mais poderosas e populares para a construção de aplicações web dinâmicas. O Rails é utilizado por empresas, como por exemplo a Airbnb, Basecamp, Disney, GitHub, Hulu, Kickstarter, Shopify, Twitter ou a Yellow Pages.

O que torna o Rails tão fantástico é o facto de ser 100% *open-source*. O Rails deve muito do seu sucesso ao seu design elegante e compacto e pode-se afirmar que se trata duma linguagem de domínio específica para escrever aplicações web. Como resultado, muitas tarefas comuns de programação web, como por exemplo, a geração de HTML, criação dos modelos de dados, routing dos URLs são fáceis com Rails e o código da aplicação resultante é conciso e legível.

O Rails beneficiou de uma comunidade extraordinariamente entusiasta e diversificada, que inclui centenas de colaboradores que disponibilizam as suas contribuições, conferências, um número enorme de *gems* (soluções independentes para problemas específicos, como paginação e upload da imagem), uma rica variedade de blogs informativos e uma infinidade de fóruns de discussão. O grande número de programadores de Rails também torna mais fácil para lidar com os inevitáveis erros das aplicações: o algoritmo de ”pesquisar no Google a mensagem de erro”quase sempre retorna um post dum blog ou dum tópico de discussão relevante.[38]

Ruby

O Ruby tem sido objeto de muita atenção desde que surgiu o Ruby on Rails, a framework que permite desenvolver aplicações web escritas em Ruby. O Ruby já existe há quase tanto tempo como Java, mas com pouca atenção fora do Japão até ao ano 2000. Nos últimos anos, a sua popularidade tem crescido constantemente, e por boas razões.

David Heinemeier Hansson, inventor de Ruby on Rails, disse algo bastante sábio: ”As pessoas aprendem por mudar uma pequena coisa, recarrega, e observar mudanças”.

Matz, o criador do Ruby, queria criar a sua própria linguagem de programação desde o secundário, queria criar uma linguagem de script, mas também que fosse orientada a objetos. Tal como em Java, o Ruby possui classes, que armazenam dados nas suas variáveis, constantes e métodos, que são coleções compactas de código que permitem executar operações. As classes

podem derivar informações a partir de outras, mas apenas uma de cada vez, através de herança. Isso permite que se reutilize o código, o que significa menos tempo gasto a corrigir ou a fazer *debugging* ao código.

Devise

O Devise[39] é uma solução de autenticação flexível para Rails e utiliza o MVC de uma forma completa, permitindo vários modelos autenticados ao mesmo tempo. É extremamente modular, ou seja, utilizam-se somente os componentes necessários. Em seguida, faz-se uma breve analise aos módulos mais importantes:

Autenticação, Base de Dados: criptografa e armazena uma senha na base de dados para validar a autenticidade de um utilizador durante a inscrição. A autenticação pode ser feito tanto por meio de solicitações POST ou autenticação básica HTTP.

Confirmável: envia um email com instruções de confirmação e verifica se a conta está confirmada antes de fazer o login.

Recuperável: permite ao utilizador redefinir a sua senha, enviam as instruções par o seu email.

Registável: trata de inscrever os utilizadores através de um processo de registo, também permitindo-lhes editar e apagar a sua conta.

Rememberable: cria e gera um token que permite lembrar o utilizador, a partir de um cookie.

Timeoutable: termina as sessões que não foram acedidas por um determinado período de tempo.

Verificável: fornece validações para o email e para a senha. Também é possível personalizar e criar validações.

Bloqueável: bloqueia uma conta após um determinado número de tentativas de login falhadas, pode ser desbloqueada via email ou depois de um período de tempo especificado.

Cancan

O CanCan é uma biblioteca de autorização para Ruby on Rails, que restringe os recursos que um determinado utilizador tem permissão para aceder. Todas as permissões são definidas num único local (a classe Ability) e não há necessidade de duplicar controladores, views e consultas a base de dados.[40]

Paperclip

O Paperclip é uma biblioteca que se destina a anexar ficheiros duma forma fácil através do Active Record. A intenção do Paperclip é manter as configurações tão fáceis quanto possível e tratar os ficheiros como se fossem outros atributos. Isso significa que os ficheiros não são guardados na sua localização final no disco, nem são apagados se forem definidos como nulo, até evocado ActiveRecord::Base#save. As validações são geridas com base no tamanho e presença dos ficheiros, se for necessário verificar alguma destas restrições. O Paperclip pode transformar a sua imagem carregada em outras de tamanhos diferentes, se necessário, para tal ser possível, é necessário instalar o ImageMagick. Os ficheiros anexados são guardados no sistema de arquivo referenciado por uma especificação facilmente compreensível e alterável.[41]

Font-awesome-rails

O Font Awesome dá ícones vetoriais escaláveis que podem ser facilmente personalizados, tanto em tamanho, cor, sombra, e/ou qualquer outro atributo, podendo ser alterado com o poder da CSS. [42] A biblioteca font-awesome-rails fornece todas as funcionalidades do Font Awesome de uma forma simples a ajustada ao Ruby on Rails ,tornado a sua utilização mais simples na produção de páginas web. [43]

Faker

Esta biblioteca é utilizada na produção de dados falsos, sendo muito úteis durante o processo de desenvolvimento. Com os dados fornecidos, é possível executar testes de performance, escalabilidade, robustez da aplicação desenvolvida. É igualmente muito útil na criação e organização das *views*, permitindo uma melhor distribuição e organização dos dados a apresentar ao utilizador. No entanto, o Faker não pode garantir a unicidade dos dados gerados, dado que gera os dados de forma aleatória.[44]

Will paginate

O will_paginate é uma biblioteca de paginação que se integra com Ruby on Rails, tem uma coleção de extensões para a camada de dados que permitem criar paginadas e visualizar a informa-ção contida em cada uma delas. Na camada view pode-se ajustar-se o estilo da CSS. Com esta biblioteca pode-se assim dividir a informa-ção em varias paginas web, com um numero defendível de elementos por pagina, em vez de apresentar toda a informação numa única pagina com informa-ção ”infinita”, facilitada, assim, a navegabilidade entre a informação. [45]

Google Charts

O Google Chart é uma ferramenta que permite facilmente criar um gráfico a partir de alguns dados e incorporá-lo em uma página da web. O Google cria uma imagem PNG do gráfico a partir dos parâmetros de formatação na requisição HTTP. Para incluir os gráficos gerados numa página da web, basta utilizar a tag de imagem do HTML com a imagem PNG fornecida pelo Google. [46]

Bootstrap

O Bootstrap torna o desenvolvimento do front-end para a web mais rápido e mais fácil. É feito para pessoas de todos os níveis, dispositivos de todas as formas e projetos de todos os tamanhos.

O Bootstrap usa no seu código fonte dois dos pré-processadores de CSS mais populares, Less e Sass, o que permitem a utilização de macros no CSS.

Escala de forma eficiente, sites e aplicações, com um único código fonte, tanto para smartphones, como para computadores com as mais diversas resoluções de ecrã.

O Bootstrap vem acompanhado de uma documentação extensa para elementos HTML comum, dezenas de personalizações, componentes CSS, e plugins jQuery diversificados.[47]

6.1.2 JavaScript

O JavaScript é uma linguagem orientada a objetos que permite a realização de cálculos e manipulação de objetos num ambiente de *runtime*. Nos últimos anos, o JavaScript ganhou muita importância, e a explicação para tal reside no facto da linguagem se ter imposto como a linguagem de script da Web. Sendo maioritariamente usada em aplicações Web, desempenha um papel importante do lado do servidor e praticamente indispensável, do lado do cliente[48]

A esmagadora maioria dos sites modernos usam JavaScript e todos os navegadores modernos tanto em computadores, consolas, tablets e smartphones incluem interpretadores de JavaScript, tornando o JavaScript a linguagem de programação mais omnipresente na história. O JavaScript faz parte do tridente de tecnologias que todos os desenvolvedores Web devem aprender: HTML para especificar o conteúdo, CSS para especificar a apresentação e JavaScript para especificar o comportamento das páginas web.

Para quem já está familiarizado com outras linguagens de programação, pode ajudar saber que o JavaScript é uma linguagem de alto nível, dinâmica, sem tipo, é interpretada e não compilada, permite um estilo de programação orientado a objetos, onde JavaScript deriva a sua sintaxe de Java.

Para ser útil, uma linguagem deve ter uma plataforma ou biblioteca padrão ou API de funções para a realização de, coisas como por exemplo *input* e *output* básico. O núcleo da linguagem JavaScript, define uma API mínima para trabalhar com texto, matrizes, datas e expressões regulares, mas não inclui qualquer funcionalidade de input ou output. O *input* e o *output* (assim como recursos mais sofisticados, tais como armazenamento e gráficos) são da responsabilidade do ”ambiente de host”dentro do qual está incorporado o JavaScript.[49]

6.1.3 HTML 5

O HTML foi publicado pela primeira vez como um esboço da internet em 1993. Os anos 90 tiveram uma enorme atividade em torno do HTML, com a versão 2.0, as versões 3.2 e 4.0 (no mesmo ano) e finalmente, em 1999, a versão 4.01. No decorrer do seu desenvolvimento, a World Wide Web Consortium (W3C) assumiu o controle da especificação.

Nesta época, o HTML era primitivo e não existiam ferramentas disponíveis. As aplicações web quase não existiam, exceto alguns scripts de processamento de texto primitivos. As páginas web eram codificadas à mão, usando um editor de text,o e raramente eram atualizadas.

Após a evolução rápida destas quatro versões, o HTML foi amplamente considerado um beco sem saída. O foco do desenvolvimento web deslocou-se para o XML e para o XHTML. O HTML foi colocado em stand by. Neste período de tempo, o HTML recusou-se a morrer e a maioria de conteúdo na web continuou a ser criado usando o HTML. Com as necessidades exigidas pelas novas aplicações web, foi necessário criar novas características e especificações para HTML.

Com o intuito de levar a plataforma web para um novo nível, um pequeno grupo de pessoas criou o Web Hypertext Application Working Group (WHATWG), em 2004. Este grupo foi o responsável pela especificação do HTML5, trabalhando em novos recursos específicos para as necessidades da novas aplicações Web e foi nesta época que o termo Web 2.0 foi cunhado. Realmente era como se uma nova web aparecesse, onde sites estáticos deram lugar a sites mais dinâmicos e sociais que exigiam mais recursos e muito mais características.

Percorreu-se um longo caminho nos últimos quinze anos, muitos esforços têm sido feitos para tornar o HTML5 mais seguro e dinâmico. Cada parte da especificação do HTML5 tem secções sobre segurança, pois a segurança é um ponto primordial. O HTML5 introduz um novo modelo de segurança baseado na origem, fácil de usar, utilizado de forma consistente por diferentes APIs.[50]

6.1.4 MySQL

O MySQL é um servidor de base de dados SQL (Structured Query Language) muito rápido, robusto, *multithreaded* e multi-utilizador. O MySQL é utilizado em sistemas informáticos que tenham necessidade de armazenar uma quantidade de dados considerável e complexa.

Uma base de dados é uma coleção de dados estruturados. Pode ser qualquer coisa, desde uma simples lista de compras a uma galeria de imagens ou a grande quantidade de informações numa rede distribuída. Para adicionar, aceder e processar dados armazenados numa base de dados, é necessário um sistema para gerir os dados e o MySQL Server assume esta função.

O MySQL é um sistema de gestão de base de dados relacional. Uma base de dados relacional armazena os dados em tabelas separadas em vez de colocar todos os dados numa só tabela gigantesca, o que proporciona uma maior velocidade e flexibilidade no acesso aos dados. Tais tabelas estão interligadas com regras relacionais, que consistem em associar um ou vários atributos de uma tabela com um ou vários atributos de outra tabela, tornando possível combinar dados de várias tabelas, mediante um pedido.

O MySQL é open source. Uma licença open source permite que qualquer pessoa faça o download do software, utilizá-lo, estudar o código fonte e alterá-lo para atender às suas necessidades, sem qualquer custo.

O servidor de base de dados MySQL é muito rápido, confiável e fácil de usar. O MySQL também tem um conjunto de recursos muito práticos desenvolvidos em estreita cooperação com os utilizadores.

O servidor MySQL foi desenvolvido originalmente para lidar com grandes bases de dados muito mais rápidas do que as soluções existentes e tem sido usado com sucesso em ambientes de produção de alta requisição de dados por diversos anos, apesar de estar em constante desenvolvimento.

O MySQL Database Software é um sistema cliente/servidor que consiste de um servidor SQL *multithreaded* que suporta acessos diferentes *backends*, diversos programas clientes e bibliotecas, ferramentas administrativas e uma grande variedade de APIs.[51]

6.2 Obtenção de Dados

A ZenTV tem dois ambientes de navegação, utilizador comum e administrador, que para as funcionalidades que o utilizador dispões não sejam as funcionalidades de administração. Um utilizador pode ser promovido a administrador por um dos administradores atuais. O primeiro administrador é promovido, modificando o atributo na base de dados relativo as permissões de

administração.

Uma vez que os filmes estão protegidos por direitos de autor, foi impossível colocar o ficheiro multimédia. Com a ausência do ficheiro que continha o filme, também seria difícil angariar utilizadores suficientes para se poder proceder ao desenvolvimento da aplicação. Uma forma para ultrapassar esta situação seria usar um conjunto de teste, mas infelizmente não foi encontrado nenhum que satisfizesse as necessidades desta aplicação. A forma encontrada para ultrapassar problema foi criar uma script que gerasse aleatoriamente **mil** utilizadores com idades entre os zero e os oitenta anos, o género seria aleatório. Para gerar o nome e o email, usa-se a *gem* faker, já presentada. A associação ao concelho em que reside e a sua profissão é feita aleatoriamente entre as previamente carregadas para a base de dados. Por uma questão de facilidade de acesso, todos os utilizadores gerados aleatoriamente têm a sua palavra-passe igual ao seu email. A seleção das categorias que gosta bem como os filmes que gosta e não gosta é feita aleatoriamente entre os presentes na base de dados na altura.

Para facilitar o processo de desenvolvimento, também foram gerados aleatoriamente duzentos atores, cinquenta realizadores, cinquenta escritores, quarenta palavras-chave e cem filmes. Desta forma, foi possível desenvolver e testar mais facilmente todas a técnicas de inteligência artificial utilizadas neste projeto.

6.3 Modelo de Dados Utilizado

Na figura 6.1, é apresentado o modelo de dados utilizado, duma forma simplificada.

Para a melhor compreensão do modelo de dados, é apresentada uma descrição das tabelas que o compõem:

- **A tabela de atores** é constituída por nome, data de nascimento, foto e pelo seu respetivo índice;
- **A tabela de realizadores** é formada pelos campos que representam o nome, a sua data de nascimento e por uma foto que o representa;
- **A tabela de escritores** também é formada pelos campos que representam o nome, a sua data de nascimento e por uma foto que o representa;
- **A tabela de categorias** é formada por cor e designação da categoria;

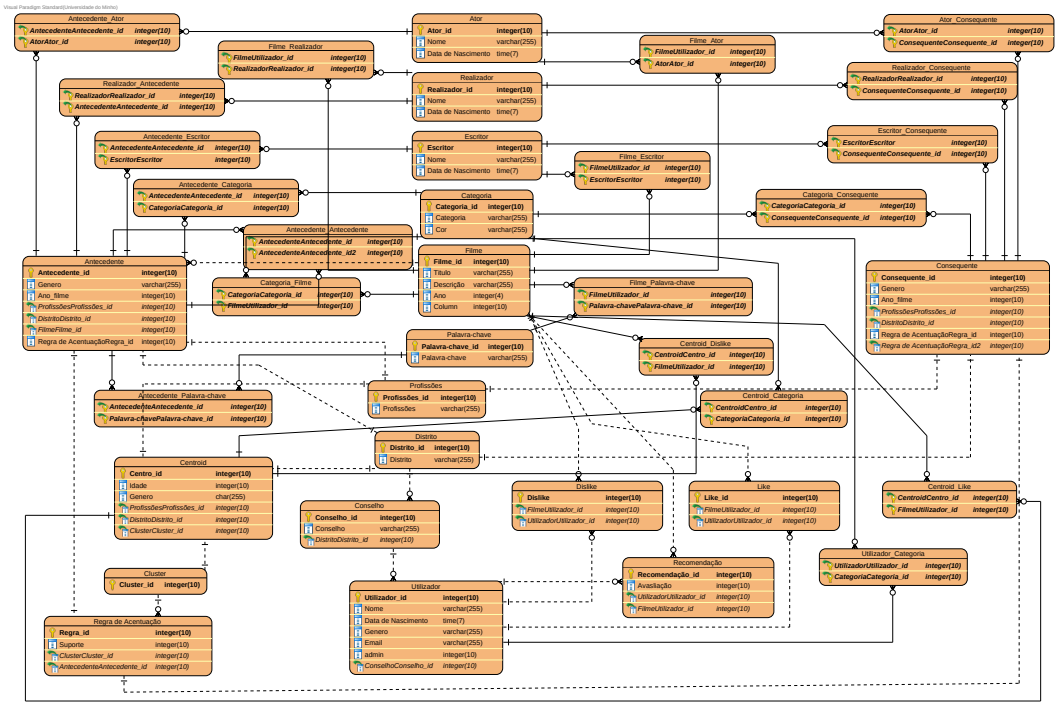


Figura 6.1: Modelo de Dados Utilizada

- **A tabela de filmes** é formada pelo seu título, duração, ano, foto e ficheiro multimédia com o conteúdo do filme. A tabela de filmes está ligada à tabela de atores, categorias, escritores, realizadores e palavras-chave;
- **A tabela das palavras-chave** é formada pela designação da palavra-chave;
- **A tabela de profissões** é formada pela designação da profissão;
- **A tabela de distrito** é formada pela designação do distrito;
- **A tabela de concelhos** é formada pela designação do concelho e pela conexão ao distrito a que pertence;
- **A tabela de clusters** tem associadas as regras de associação de cada cluster e os centróides de cada cluster. Guarda ainda a informação das categorias, likes e dislikes em tabelas auxiliares;
- **A tabela que representa os centróides** é composta pela média de idades, sexo predominante, cluster a que pertence, distrito predominante e profissão predominante;

- **A tabela de utilizadores** é formada pelos campos que representam o nome, a sua data de nascimento, por uma foto que o representa, género, email, uma palavra-passe encriptada, um token, permissão de administrador, profissão, concelho onde reside e por fim a que cluster está associado. Um utilizador também tem associada uma lista de categorias pelas quais nutre um interesse especial;

- **A tabela de likes** faz a ponte entre os utilizadores e os filmes, onde cada utilizador está relacionado com os filmes que gosta;

- **A tabela de dislikes** faz a ponte entre os utilizadores e os filmes, à semelhança da tabela de likes, mas neste caso cada utilizador está relacionado com os filmes que não gosta ou não nutre um particular interesse;

- **A tabela de regras de associação** é definida pelo suporte, confiança, antecedente, consequente e pelo cluster a que está associada. Como se pode verificar, a tabela de regras de associação tem associado uma antecedente e um consequente. Para representar toda a informação que pode surgir, é necessário criar duas tabelas que permitam armazenar tal uma informação, ambas com os mesmos campos, mas cada uma representando um dos lados da implicação;

- **A tabela de antecedentes** está ligada à tabela de regras de associação e pode ter um género, um ano dum filme, uma profissão, um distrito, um filme, um conjunto de atores, realizadores, escritores e/ou categorias;

- **A tabela de consequentes** está ligada à tabela de regras de associação e pode ter um género, um ano dum filme, uma profissão, um distrito, um filme, um conjunto de atores, realizadores, escritores e/ou categorias;

- **A tabela de recomendações** representa os possíveis filmes que os utilizadores podem gostar, acompanhados com a sua respetiva avaliação.

Em tabelas auxiliares, é armazenada a informação que permite fazer a ligação de ”muitos para muitos”, permitindo fazer a ponte entre a tabelas com outras tabelas com informação relevante. Estas tabelas têm habitualmente apenas dois campos, um representa o índice do elemento na tabela de origem e o outro representa o índice da tabela de destino. Desta forma, é possível guardar mais informação, gastando menos espaço em disco.

6.4 Vista Geral Sobre a Aplicação

6.4.1 Diagrama de Use Case - ZenTV

Os Use Cases são uma importante técnica de levantamento de requisitos, tendo sido muito utilizados em engenharia de software nos últimos anos. Os diagramas de Use Case constituem uma lista de interações entre o ator e uma parte de um sistema ou de uma unidade funcional, com a finalidade de atingir um certo objetivo e são muito usados como meio de descobrir e registar requisitos.

Abaixo, na figura 6.2, é apresentado o diagrama de Use Case utilizado na implementação do ZenTV:

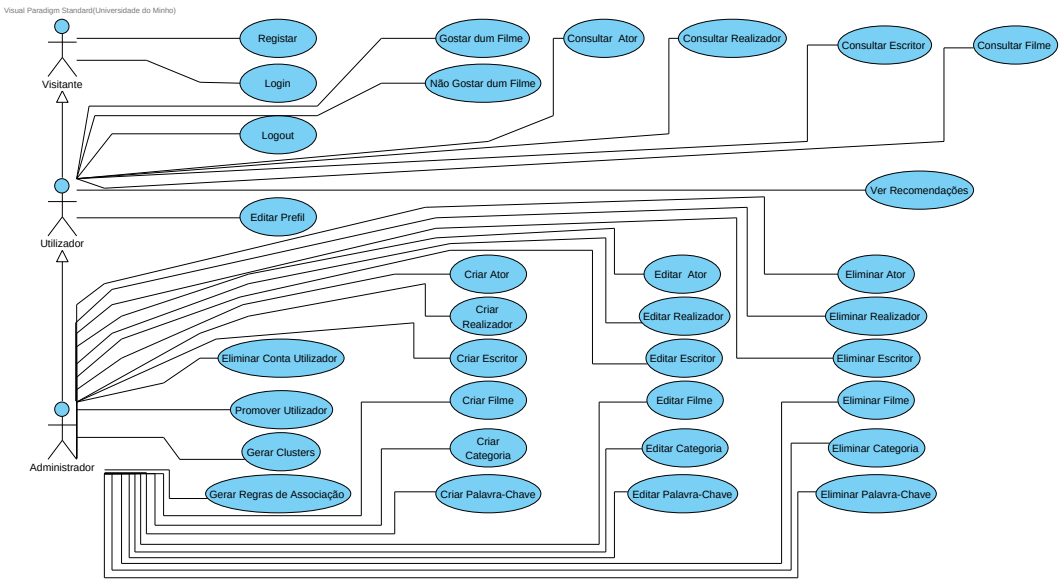


Figura 6.2: Diagrama de Use Case - ZenTV

Como se pode ver no diagrama da figura 6.2, os utilizadores estão divididos em três grupos: visitantes, utilizadores comuns e administradores. Em seguida, faz-se uma breve descrição das funcionalidades de cada grupo.

- **Visitante –** As únicas funcionalidades a que um visitante tem acesso são o registo, onde lhe é permitido criar uma nova conta no ZenTV, e o login numa conta previamente existente;

- **Utilizador –** O utilizador herda todas as funcionalidades a que um visitante em acesso. Para além destas funcionalidades, um utilizador tem asseguradas as seguintes funcionalidades:
  - Editar Perfil – O utilizador tem a possibilidade de alterar as suas informações pessoais;
  - Logout – O utilizador pode desconectar-se do ZenTV a qualquer momento;
  - Gostar de um Filme – É permitido ao utilizador indicar que gosta dum determinado filme;
  - Não Gostar de um Filme – É permitido ao utilizador indicar que não gosta dum determinado filme;
  - Consultar Ator – O utilizador tem a capacidade de visualizar a informação dum determinado ator;
  - Consultar Realizador – O utilizador tem a capacidade de visualizar a informação dum determinado realizador;
  - Consultar Escritor – O utilizador tem a capacidade de visualizar a informação dum determinado escritor;
  - Consultar Filme – O utilizador tem a capacidade de visualizar a informação dum determinado filme;
  - Ver Recomendações – O sistema, com base nas preferências do utilizador, calcula a lista de recomendações e apresenta o resultado ao utilizador.
- **Administrador –** O administrador herda todas as funcionalidades a que um utilizador em acesso e por transitividade herda todas a funcionalidades a que um visitante em acesso. A estas funcionalidades um administrador soma as seguintes funcionalidades:
  - Eliminar Conta de Utilizador – O administrador tem a possibilidade de banir utilizadores, se assim o entender;
  - Promover Utilizador – O administrador pode promover utilizadores a administradores;
  - Gerar Clusters – O administrador tem a possibilidade de eliminar os clusters atuais e gerar novos;
  - Gerar Regras de Associação – O administrador tem a possibilidade de eliminar as regras de associação atuais e gerar novas;

- Criar Ator – O administrador pode criar novos atores;
- Criar Realizador – O administrador pode criar novos realizadores;
- Criar Escritor – O administrador pode criar novos escritores;
- Criar Filme – O administrador pode criar novos filmes;
- Criar Categoria – O administrador pode criar novas categorias;
- Criar Palavra-chave – O administrador pode criar novas palavras-chave;
- Editar Ator – O administrador pode editar atores;
- Editar Realizador – O administrador pode editar realizadores;
- Editar Escritor – O administrador pode editar escritores;
- Editar Filme – O administrador pode editar filmes;
- Editar Categoria – O administrador pode editar categorias;
- Editar Palavra-Chave – O administrador pode editar palavras-chave;
- Eliminar Ator – O administrador pode eliminar atores;
- Eliminar Realizador – O administrador pode eliminar realizadores;
- Eliminar Escritor – O administrador pode eliminar escritores;
- Eliminar Filme – O administrador pode eliminar filmes:
- Eliminar Categoria – O administrador eliminar categorias;
- Eliminar Palavra-Chave – O administrador pode eliminar palavras-chave.

6.4.2 Modelo de Domínio - ZenTV

O modelo de domínio é um glossário do projeto, pois identifica os termos utilizados e representa as relações existentes entre esses mesmos termos. Para criar o modelo de domínio, utiliza-se um diagrama de classes com as entidades pertencentes ao domínio do problema e com as relações entre elas.

Não se deve confundir o modelo de dados com o modelo de domínio. Embora os diagramas possam parecer semelhantes, representam conceitos diferentes, pois uma tabela relaciona um conjunto de dados enquanto uma classe é um agregado de dados e comportamentos, ou seja, um objeto representa uma instância duma entidade.

Na figura 6.3, é apresentado o diagrama com o modelo de domínio utilizado na implementação do ZenTV:

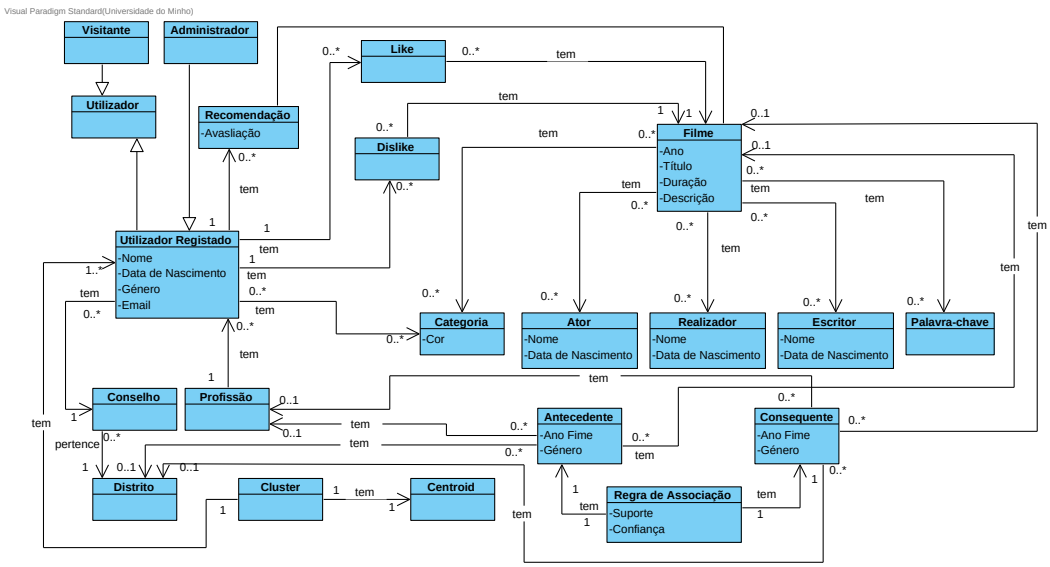


Figura 6.3: Modelo de Domínio - ZenTV

Para a melhor compreensão do diagrama da figura 6.3, é feita uma breve descrição de cada classe do diagrama de classes que representa o modelo de domínio.

- **Visitante –** Trata-se duma ideia abstrata para que não sejam esquecidas as funcionalidades para os utilizador não registados, por isso é considerada uma subclasse de utilizador;
- **Utilizador –** É a superclasse de utilizador, que tem como subclasses Visitante e Utilizador Registado;
- **Utilizador Registado –** Representa um utilizador que tem acesso a uma conta no ZenTV. É composto por nome, data de nascimento, género, email, profissão, concelho, lista de categorias pelas quais nutre interesse, lista de filmes que gosta, lista de filmes que não gosta e pela sua lista de recomendações;
- **Administrador –** Um administrador é subclasse dum utilizador registado. Em termos de implementação, é um utilizador registado, com a diferença da variável de administração, que neste caso possui o valor verdadeiro;

- **Concelho –** Representa um concelho e pertence a um distrito;
- **Distrito –** Representa um distrito;
- **Profissão –** Representa uma profissão;
- **Recomendação –** Representa uma recomendação ao utilizador. Dela fazem parte um filme e a sua possível aceitação por parte do utilizador a que é destinado;
- **Like –** Representa o interesse positivo que o utilizador nutre por um filme;
- **Dislike –** Representa o interesse negativo que o utilizador nutre por um filme;
- **Filme –** Representa um filme e é composto por ano, título, duração, uma descrição, elenco, lista de realizadores, lista de escritores, lista de palavras-chave e pela lista de categorias a que pertence;
- **Categoria –** Representa uma categoria, composta por uma cor, para que seja possível identificar a categoria mais rapidamente, e pela respetiva designação;
- **Ator –** Representa um ator e é composto pelo seu nome, data de nascimento e a lista de filmes em que participou como ator.
- **Realizador –** Representa um realizador e é composto pelo seu nome, data de nascimento e a lista de filmes que realizou;
- **Escritor –** Representa um escritor e é composto pelo seu nome, data de nascimento e a lista de filmes que escreveu;
- **Palavra-chave –** Representa uma palavra-chave.
- **Cluster –** É um conjunto de utilizadores que estão próximos. Além do conjunto de utilizadores, um cluster também é composto por um centróide;
- **Centróide –** Representa o valor médio de todos os elementos que compõem o conjunto de utilizadores dum cluster.
- **Regra de Associação –** Representa uma regra de associação, composta por um antecedente e por um consequente, bem como pelo seu suporte e pela sua confiança;

- **Antecedente –** Representa o lado esquerdo da implicação presente na regra de associação e pode ser composto por ano do filme, género, lista de atores, lista de realizadores, lista de escritores, lista de categorias e lista de palavras-chave, que foram encontradas no processo de descoberta de regras de associação;

- **Consequente –** Representa o lado direito da implicação presente na regra de associação, e pode ser composto por ano do filme, género, lista de atores, lista de realizadores, lista de escritores, lista de categorias e lista de palavras-chave, que foram encontradas no processo de descoberta de regras de associação;

Um aspeto importante a referir é o facto de as linhas que ligam tanto o antecedente como o consequente às categorias, atores, realizadores, escritores e palavras-chave foram removidas para uma melhor leitura do diagrama.

6.5 Interface Utilizada

A ZenTV é uma aplicação web e como tal foi utilizada uma interface que permita ao utilizador aceder a toda a informação necessária em menos de 4 clicks. Como as pessoas têm a necessidade de executar tarefas de forma fácil e eficaz, a interface tenta ser minimalista, intuitiva e fácil de manusear pelo utilizador. Como se pode ver na figura 6.4, a interface tenta ser simples para todos os estratos sociais e para todos os níveis intelectuais e assim tenta induzir o mínimo possível de stress em todos os tipos de utilizadores.

6.6 Implementação do Algoritmo K-Means

O k-means recebe como parâmetro o número de partições em que se vão dividir os espaços de dados em análise. O primeiro passo do algoritmo é criar as k partições (designadas por clusters), e para cada um dos clusters, é selecionado aleatoriamente um utilizador do conjunto de todos os utilizadores, passando a ser membro e o centróide temporário do cluster a que foi alocado. Os utilizadores selecionados são removidos do conjunto onde se encontram todos os utilizadores.

O próximo passo é percorrer o conjunto de utilizadores restantes e alocar cada utilizador ao cluster cuja distância entre ele e o centróide é menor. A função de distância utilizada na medição é descrita na secção 6.6.1.

Após a distribuição dos utilizadores pelos clusters, é calculado um novo centróide para cada um dos clusters. Os novos centróides são calculados da seguinte forma para cada componente

Figura 6.4: Página de Entrada da ZenTV

do centróide:

- **Idade:** é obtida através da média aritmética de todos os elementos do cluster, $\frac{1}{n}\sum_{i=1}^{n} x_i$, onde x_i é a idade do utilizador i e n é o número total de utilizadores;
- **Género:** o total de elementos de cada género é guardado em duas variáveis. No final da contagem, o centróide guarda o género com maior número bem como a percentagem de elementos desse género. Por exemplo, um cluster que contenha 100 elementos do género masculino e 200 do género feminino, o centróide guarda a informação {género = Feminino, percentagem = 66.7% };
- **Profissão:** de forma semelhante ao que é feito na variável $género$, também são contadas as ocorrências de cada profissão, mas neste caso cada profissão é agrupada numa tabela de hash, onde a chave será a profissão. No final, percorre-se a hash e guarda-se no centróide a profissão predominante e a sua percentagem;
- **Distrito:** o processo é exatamente igual ao que é feito na variável $profissão$, mas analisando a variável $distrito$. No final da contagem, também se percorre a hash e guarda-se no centróide o distrito predominante e a sua percentagem;
- **Categorias:** para cada categoria encontrada, guarda-se numa hash o número de ocorrências da categoria utilizada como chave. No final, para cada categoria encontrada na análise, guarda-se no centróide a respetiva categoria e a percentagem de ocorrências da mesma;

- **Likes:** aqui analisam-se os filmes que os utilizadores gostam e utiliza-se um processo semelhante ao implementado nas categorias, guardando numa hash o número de ocorrências de um determinado filme que está marcado com ”like”. No final, para cada filme encontrado na análise, guarda-se no centróide o respetiva filme na secção de likes e a percentagem de ocorrências do mesmo;

- **Dislikes:** o processo é o mesmo do que foi apresentado para os likes, mas neste caso são analisados os filmes que os utilizadores não gostam. No final, guarda-se a informação sobre os filmes e a percentagem de ocorrências de cada um na secção de dislikes.

Após o cálculo dos novos centróides é recalculada a distância entre os utilizadores e os centróides. No caso da distância entre o utilizador e o centróide de um outro cluster ser menor do que a distância ao centróide do cluster que é membro, move-se o utilizador para o cluster cuja a distância é menor e assinala-se que nesta iteração ocorreram alterações. No final da avaliação de todo os espaços em análise, são recalculadaos os novos centróides. Este procedimento repete-se até que a função de erro convirja, ou seja, quando não ocorrem mais alterações. Em seguida, é apresentada mais exaustivamente a função de distância utilizada na implementação do k-means.

6.6.1 Função de Distância

A função de distância é o que permite estimar a que distância se encontram dois objetos, esta função é essencial para o algoritmo de clustering. Nesta função, assume-se que que a idade máxima dum utilizador é de 123, pois a maior idade autenticada que um humano já viveu é de 122 anos e 164 dias. A pessoa responsável por esta marca foi a senhora Jeanne Louise Calment, francesa, que nasceu a 21 de fevereiro de 1875 e faleceu a 4 de Agosto de 1997[52].

Para calcular a idade do utilizador, é necessário saber qual é a data atual, depois faz-se a diferença entre a data atual e a data de nascimento do utilizador para saber a idade.

Para calcular a distância entre a idade de dois utilizadores, recorre-se à distancia euclidiana, com as idades normalizadas. Por outras palavras, trata-se da raíz quadrada do quadrado da diferença entra a idade do utilizador um e o a idade do utilizador dois, ambas as idades dividida pela idade máxima defendida (123 anos) ou seja:

$$d_{idade}(i,j) = \sqrt{\left(\frac{idade_i}{idade_{max}} - \frac{idade_j}{idade_{max}}\right)^2}, \tag{6.1}$$

A normalização obriga a que todos os valores das variáveis estejam entre 0 e 1, o que permite dar pesos a cada variável e assegurar que a função final não é afetada por variáveis de valor elevado.

Exemplo 6.6.1. Por exemplo, suponha-se que estão dois utilizadores em análise:

Utilizador i:

- **Nome:** António João Cardoso Simões
- **Idade:** 75
- **Sexo:** Masculino
- **Distrito:** Lisboa
- **Profissão:** Professor

Utulizador j:

- **Nome:** Maria Victória Alves Gomes
- **Idade:** 29
- **Sexo:** Feminino
- **Distrito:** Braga
- **Profissão:** Enfermeira

Sem a utilização da normalização, a distância em relação à idade seria:

$$d_{idade}(i,j) = \sqrt{(75-29)^2} = 46,$$

e com normalização o valor encontrado seria:

$$d_{idade}(i,j) = \sqrt{\left(\frac{75}{123} - \frac{29}{123}\right)^2} \approx 0.373984,$$

Como se pode verificar, o valor normalizado permite aproximar a distância em relação à idade ao valor da distância das outras variáveis.

Para calcular a distância entre dois utilizadores em relação à variável que representa o género, recorre-se a tabela de contingência e a equação (4.8) apresentada na secção 4.1.1. Voltemos a recordar a equação (4.8),

$$d(i,j) = \frac{r+s}{q+r+s+t}$$

onde,

- **q** é o número de variáveis que são igual a 1 para ambos os objetos **i** e **j**:
- **r** é o número de variáveis que são igual a 1 para o objecto **i**, mas que são igual a 0 por objecto **j**:
- **s** é o número de variáveis que são igual a 0 para o objeto **i** mas igual a 1 para o objeto **j**:
- **t** é o número de variáveis que são igual 0 para ambos os objetos **i** e **j**.

Também é de referir que se tomou a decisão que o género masculino seria representado pelo valor um (1) e o feminino seria representado pelo valor zero (0).

Recordando o exemplo 6.6.1 como referência, a tabela de contingência seria:

Utulizador i \ Utlizador j	Masculino	Feminino	Soma
Género			
Masculino	q = 0	r = 1	q + r
Feminino	s = 0	t = 0	s + t
Soma	q + s	r + t	p

Tabela 6.1: Tabela de Contingência

Usando a informação obtida através da tabela 6.1 é possível substituir as variáveis, na equação (4.8), pelos respetivos valores e obter a distância entre os dois utilizadores, da seguinte forma:

$$d_{genero}(i,j) = \frac{1+0}{0+1+0+0} = 1$$

Como se pode verificar, a distância é máxima, pois os valores de $d(i,j)$ têm de estar compreendidos entre 0 e 1, $d(i,j) \geq 0$ e $d(i,j) \leq 1$, onde 0 indica proximidade máxima e 1 indica

distância máxima. No exemplo apresentado, como apenas existe uma variável binária simétrica, apenas ocorreram distâncias com o valor 0, quando os utilizadores são do mesmo género, ou com o valor 1, quando os utilizadores são de géneros diferentes.

A decisão da utilização e implementação total deste método para calcular as variáveis binárias foi tomada com a consciência de que apesar de se ter atualmente apenas uma variável binária simétrica em análise, no futuro poderão surgir outras em análise que sejam necessárias para o problema, como por exemplo o turno de trabalho (diurno ou noturno).

O maior lote de variáveis da implementação são variáveis binárias assimétricas. Deste lote, fazem parte as categorias pelas quais o utilizador nutre interesse, os filmes que o cativam e os filmes que não aprecia.

Para cada um destes três conjuntos de variáveis, designados por categorias, likes e dislikes, surgiu a necessidade de criar três equações de distância, $d_{categ}(i,j)$, $d_{like}(i,j)$ e $d_{dislike}(i,j)$, para que ser possível depois calcular a distância ponderada com todas as variáveis em análise. Para se calcular a distância para cada um destes conjuntos de variáveis, recorre-se à equação 4.9 e não ao coeficiente de Jaccard 4.10, pois até agora foi calculada a distância entre objetos e não a sua semelhança de proximidade.

Recorde-se novamente a equação 4.9, que foi usada para o cálculo de distancia, $d_{categ}(i,j)$, $d_{like}(i,j)$ e $d_{dislike}(i,j)$.

$$d(i,j) = \frac{r+s}{q+r+s}$$

Neste projeto, são utilizadas duas variáveis nominais, a profissão e o distrito a que pertence o utilizador. Para o cálculo da distância destas duas variáveis é relembrada a equação 4.11:

$$d(i,j) = \frac{p-m}{p},$$

onde $\mathbf{m}$ é o número de variáveis em que tanto $\mathbf{i}$ como em $\mathbf{j}$ têm o mesmo valor e $\mathbf{p}$ representa o número total de variáveis. Utilize-se o exemplo 6.6.1 para demonstrar o que acabou de ser descrito.

Como o António é de **Lisboa** e a Maria é de **Braga**, a variável $distrito$ não tem o mesmo valor, logo não irá incrementar as variável $\mathbf{m}$. Como ambos são de profissões diferentes, um utilizador é **professor** e outro **enfermeiro**, a variável $profissão$ também não irá incrementar $\mathbf{m}$.

Neste momento, tem-se $\mathbf{m} = 0$ e $\mathbf{p} = 2$. Recorrendo-se a equação 4.11 para obter o valor da distância, tem-se:

$$d_{distProf}(i,j) = \frac{2-0}{2} = 1.$$

Mais uma vez, a distância é máxima, como seria expectável.

A distância total será um somatório de todas as distâncias até agora apresentadas, multiplicadas pelo peso que cada componente deve assumir. Sabendo que a distância tem de estar ente 0 e 1, é produzida a seguinte equação 6.2:

$$\begin{aligned} d_{total}(i,j) = p_1 \times d_{idade}(i,j) + p_2 \times d_{genero}(i,j) + p_3 \times d_{distProf}(i,j) \\ + p_4 \times d_{categ}(i,j) + p_5 \times d_{like}(i,j) + p_6 \times d_{dislike}(i,j), \end{aligned} \tag{6.2}$$

onde p_z representa o peso para cada um dos membros da equação de distância total, $d_{total}(i,j)$. Estes pesos podem ser fornecidos como parâmetro, de forma a adequar-se às preferências do administrador, como a restrição de obedecer à seguinte regra:

$$p_1 + p_2 + p_3 + p_4 + p_5 + p_6 = 1. \tag{6.3}$$

6.7 Implementação das Regras de Associação

Para se poderem encontrar as regras de associação, é necessário receber como parâmetro um valor que representa o suporte mínimo e outro valor que representa a confiança mínima. Estes dois valores tem de estar compreendidos entre 0 e 100.

Para cada cluster, é encontrado o conjunto de itemsets frequentes e depois para um deles são derivadas as regras de associação fortes. Para que tal seja possível, é criada uma estrutura de dados auxiliar com a informação associada a cada cluster. Mais uma vez, são usadas tabelas de hash para otimizar todo o processo de procura que é feito no procura de itemsets frequentes.

Como já foi referido anteriormente na secção 4.2, recorre-se ao algoritmo apriori para serem encontrados os itemsets frequentes. Começa-se por encontrar todos os 1-itemsets, C_1, que satisfazem o suporte mínimo. Depois segue-se o passo de poda onde é obtido L_1. Com a junção $L_1 \bowtie L_1$ obtém-se C_2. Este processo repete-se para todos os C_k e L_k, tendo como caso de paragem $C_k = d$, onde d representa a toda a informação contida na base de dados, ou quando $C_k = \phi$, ou seja, quando C_k é um conjunto vazio. Os elementos dos itemsets podem ser idade, género, profissão, distrito, categorias, filmes, atores, realizadores e escritores.

Depois de serem encontrados os itemsets frequentes, procede-se à construção das regras de associação. Para obter regras de associação fortes, estas têm de satisfazer tanto o suporte mínimo como a confiança mínima. O suporte mínimo está assegurado pelo algoritmo apriori. Para determinar a confiança, recorre-se à equação 4.18, que é relembrada aqui:

$$confianca(A \Rightarrow B) = P(B|A) = \frac{suporte\_count(A \cup B)}{suporte\_count(A)}.$$

A probabilidade condicional é expressa em termos do suporte absoluto do itemset, onde $(A \cup B)$ é o número de transações que contêm o itemset $A \cup B$, e $suporte\_count(A)$ é o número de transações que contêm o itemset A. As regras de associação são geradas da seguinte formas: para cada itemset frequente, l, são gerados todos os subconjuntos não vazios. Para todos os subconjuntos gerados, s, obtém-se a regra "$s \Rightarrow (l - s)$"se a confiança mínima for assegurada. Por fim, todas as regras de associação fortes são guardadas na base de dados e exibidas ao administrador.

6.8 Implementação do K-Nearest Neighbors

O k-Nearest Neighbors é utilizado em dois cenários distintos: quando se regista um novo utilizador e quando o utilizador aciona o botão de like dum filme.

Quando o utilizador aciona o botão de like, o k-Nearest Neighbors utiliza a função de distância entre filmes, descrita na secção 6.8.2, para medir a distância de todos os filmes que estão presentes na base de dados, com a exceção dos que estão presentes na lista de likes e na lista de dislikes, aos filmes que estão presentes na lista de likes do utilizador. Por outras palavras, é medida a distância de cada um dos filmes a todos os filmes que o utilizador gosta. Depois, é calculada a média das distâncias e o resultado é guardado na base de dados de recomendação. Os filmes que não fazem parte das recomendações são os filmes que o utilizador manifestou interesse ou desinteresse, por este motivo os filmes que fazem parte da lista de likes e da lista de dislikes não são incluídos no cálculo. Neste caso, o k-Nearest Neighbors é utilizado para estimar a aceitação do utilizador a cada filme: quanto mais distante da sua lista de likes, menos provável será que o utilizador goste do filme. O valor de k, nesta situação, está dependente do tamanho da lista de likes, onde o k será igual à sua cardinalidade.

Quando se regista um novo utilizador, o k-Nearest Neighbors utiliza a função de distância entre utilizadores, descrita na secção 6.8.1, pois o sistema não tem informação relativa aos gostos do utilizador e, para fazer face a este facto, é necessário procurar os k utilizadores vizinhos do novo utilizador. O k é defendido pelo administrador e pode ser alterado quando necessário. Depois de

encontrados os k utilizadores vizinhos, utiliza-se a lista de likes de cada um deles para estimar a lista de likes do novo utilizador. Depois de estimada a lista de likes, realiza-se o mesmo procedimento que ocorre quando o utilizador aciona o botão de like.

O resultado do k-Nearest Neighbors é visualizado na página de recomendações, que no caso da ZenTV é a pagina de entrada. As recomendações estão ordenadas segundo o cálculo da média de distâncias de cada filme à lista de likes do utilizador, ou na ausência desta, à lista de likes estimada.

6.8.1 Função de Distância Entre Utilizadores

A função de distância utilizada no k-Nearest Neighbors é semelhante à utilizada na implementação do k-means, com a diferença que neste caso só é utilizada a distância euclidiana, como referido na secção 4.3.1, ou seja a equação 4.20 relembrada abaixo:

$$dist(X_1, X2) = \sqrt{\sum_{i=1}^{n}(x_{1i} - x_{2i})^2}.$$

Para utilizar esta função, recorre-se às adaptações referidas na secção 4.3.1, que serão descritas no decorrer desta secção. A normalização dos valores utiliza a equação 4.21 , também descrita na secção 4.3.1, relembrada a seguir:

$$v' = \frac{v - min_A}{max_A - min_A},$$

Como já foi referido, a normalização obriga a que todos os valores das variáveis estejam entre 0 e 1, o que permite a função final não seja afetada por variáveis de valor elevado.

Para guardar o resultado do somatório de todas as distâncias dos atributos, é utilizada a variável *sigma* que representa o Σ da equação 4.20. Nesta função de distância, assume-se que a idade máxima de um utilizador é de 123, pois como já foi referido, a maior longevidade dum humano registada é de 122 anos e 164 dias e a idade mínima é de 0 anos.

Dado que o registo que se tem é a data de nascimento do utilizador e não a sua idade, é necessário calcular a idade atual para cada utilizador. Para tal, faz-se a diferença entre a data atual do sistema e a data de nascimento do utilizador para saber a sua idade. Para calcular a distância entre a idade de dois utilizadores recorre-se à equação 4.21, para normalizar as idades, onde min_{idade} é igual a 0, e por isso não é considerado, e max_{idade} é igual a 123. A normalização é feita da seguinte forma:

$$idade' = \frac{idade}{max_{idade}},$$

Depois de normalizadas as idades, o resultado de

$$(idade'_{utilizador1} - idade'_{utilizador2})^2$$

é acumulado na variável *sigma*. De seguida, é calculado o valor do próximo atributo em análise, o género. Para calcular a distância entre dois utilizadores em relação ao género, recorre-se à tabela de contingência e à equação (4.8), apresentada na secção 4.1.1. A distância em relação ao género é dada por:

$$dist_{genero}(i,j) = \frac{r+s}{q+r+s+t}.$$

O resultado de $dist_{genero}(i,j)$ volta a ser acumulado na variável *sigma*.

Como será expectável, só as variáveis binárias assimétricas relativas às categorias pelas quais o utilizador nutre interesse são analisadas, uma vez que não faz sentido avaliar a distância entre as outras variáveis binárias assimétricas, analisadas na implementação do k-means, pois o utilizador recém registado não dispõe de filmes que gosta ou de filmes que não gosta. Para se calcular a distância para os conjuntos de variáveis com as categorias, recorre-se à equação 4.9.

Após o cálculo de $dist_{categ}(i,j)$,

$$dist_{categ}(i,j) = \frac{r+s}{q+r+s},$$

multiplica-se o resultado obtido pelo número de variáveis analisadas e é acumulado na variável *sigma*.

Por fim, avaliam-se as duas variáveis nominais, a profissão e o distrito do utilizador. Para o cálculo da distância destas duas variáveis, recorre-se à equação 4.11.

O resultado de $dist_{prof\_distri}(i,j)$,

$$dist_{prof\_distri}(i,j) = \frac{p-m}{p},$$

é multiplicado por 2, pois neste passo são analisadas duas variáveis, e depois é acumulado na variável *sigma*. O resultado final da função de distância é dado pela raíz quadrada do valor acumulado na variável *sigma*, $\sqrt{sigma}$. Desta forma tem-se a equação:

$$dist(X_1, X2) = \sqrt{\sum_{i=1}^{n}(x_{1i} - x_{2i})^2}.$$

6.8.2 Função de Distância Entre Filmes

Tal como no algoritmo de clustering, a função de distância é essencial para o cálculo do vizinho mais próximo. No caso da função que mede a distância entre filmes, volta-se a utilizar a equação 4.20,

$$dist(X_1, X2) = \sqrt{\sum_{i=1}^{n}(x_{1i} - x_{2i})^2}.$$

Para utilizar esta função, recorre-se às adaptações referidas na secção 4.3.1, que serão descritas no decorrer desta secção. Para a normalização dos valores, volta-se a usar a equação 4.21,

$$v' = \frac{v - min_A}{max_A - min_A},$$

Para guardar o resultado do somatório de todas as distancia dos atributos é, também, utilizada a variável *sigma* que representa o Σ da equação 4.20.

Na função, o primeiro atributo analisado é a duração do filme. Para obter o valor máximo e mínimo deste atributo, procura-se na base de dados o filme com maior duração e o filme com menor duração. Estes valores passam a ser, respetivamente, o $max_{duracao}$ e o $min_{duracao}$, para a normalização dos dados. O valor da duração normalizado, $duracao'$, é dado por:

$$duracao' = \frac{duracao - min_{duracao}}{max_{duracao} - min_{duracao}}.$$

Depois de ser feita a normalização da duração, o resultado de

$$(duracao'_{filme1} - duracao'_{filme2})^2$$

é acumulado na variável *sigma*.

O próximo atributo em analise é o ano do filme. Para obter o valor máximo e mínimo deste atributo, procura-se na base de dados o filme mais recente e o filme mais antigo. Estes valores passam a ser, respetivamente, o max_{ano} e o min_{ano}, para normalização dos dados. O valor da

duração normalizado, ano', é dado por:

$$ano' = \frac{ano - min_{ano}}{max_{ano} - min_{ano}}.$$

Depois de normalizadar a duração, o resultado de,

$$(ano'_{filme1} - ano'_{filme2})^2$$

é acumulado na variável *sigma*.

As variáveis binárias assimétricas analisadas foram as categorias dos filme, as palavras-chave, o elenco, os escritores e os realizadores do filme. Para se calcular a distância para cada conjunto de variáveis, recorre-se à equação 4.9. Para cada um dos cinco conjuntos de variáveis. é utilizada a função de distancia. Como são todas semelhantes, podem ser representadas pela função $d(i, j)$,

$$d(i, j) = \frac{r + s}{q + r + s}.$$

O resultado obtido na análise de cada conjunto de variáveis é multiplicado pelo número de variáveis analisadas e é acumulado na variável *sigma*.

O resultado final da função de distância é dado pela raíz quadrada do valor acumulado na variável *sigma*, $\sqrt{sigma}$. Desta forma tem-se o resultado a equação:

$$dist(X_1, X2) = \sqrt{\sum_{i=1}^{n} (x_{1i} - x_{2i})^2}.$$

7 Conclusões

7.1 Análise de Resultados

A televisão iterativa é uma área muito fértil, com inúmeras áreas a estudar e por vezes até labirínticas. Desenvolver um novo serviço de televisão iterativa pode levar a perguntas como ”o que oferecer?” ou ”como operar?”. As respostas a estas perguntas podem ser encontradas na secção 2.2.

Relativamente à pergunta ”como tornar o serviço de televisão iterativa sustentável”?, a resposta, segundo o analisado, seria possível de duas formas: através duma subscrição, como é o caso dos serviços de TV pagos em Portugal como Meo, Nos e Vodafone, ou ainda Netflix, onde os utilizadores têm de pagar pelo serviço prestado; ou através de publicidade como é o caso do YouTube.

Em relação à publicidade, é possível concluir que a maior parte dos utilizadores a consideram enfadonha e se tiverem oportunidade de a ignorar, fazem-no. No entanto, se esta publicidade for da sua área de interesse, têm todo o prazer de assistir até ao final e é muito mais provável comprarem o produto. Por este motivo, faz todo o sentido personalizar a publicidade e dirigi-la ao seu público alvo.

No que toca à interface, esta pode induzir stress aos utilizadores se não for simples e intuitiva. Num serviço de televisão iterativa, faz todo o sentido disponibilizar várias interfaces, como é o caso do serviço de TV da Vodafone, que permite aos seus utilizadores interagirem com o sistema através do controlo remoto, ou através do comando por voz. Como foi possível verificar, a prática da criação de vários atalhos também é uma boa medida no desenho da interface, permitindo assim uma melhor experiência ao utilizador e reduzindo os seus níveis de stress.

Foi referido na secção 2.4 que o panorama português está num bom caminho, como demonstram os prémios arrecadados no setor. No entanto, os utilizadores não estão totalmente satisfeitos e pedem melhores sistemas de recomendação, e que sejam introduzidos lentamente, pois, como já foi referido, mudanças bruscas são sempre criadoras de stress, principalmente para os utilizadores com dificuldades no manuseamento das tecnologias.

Neste projeto, existia o desafio de utilizar técnicas de inteligência numa linguagem de script, o Ruby. Sendo uma linguagem desconhecida no arranque da criação da ZenTV, o desafio foi superado, tendo sido concluído que apesar de ser possível a implementação nesta linguagem, não é uma prática recomendável dada a falta de performance registada na análise de um grande volume de dados, como é o caso da implementação do k-means ou na geração de candidatos a itemsets frequentes nas regras de associação ou na execução do k-Nearest Neighbors.

Relativamente ao sistema de recomendação, a combinação de diferentes técnicas resultou numa forma interessante de analisar o problema, capaz de dar recomendação face ao perfil do utilizador em análise.

Desta forma, os objetivos definidos na secção 1.4 foram alcançadas.Para uma melhor perceção, é feito um paralelismo com os objetivos e o que foi alcançado abaixo:

- **Reduzir o tempo de procura por conteúdos -** Com o sistema de recomendação, o tempo de procura por conteúdos é reduzido, desta forma o utilizador dispõe de mais tempo para visualizar o conteúdo selecionado;

- **Reduzir os níveis de stress do utilizador -** Para reduzir o stress do utilizador, ou pelo menos para não aumentar o stress do utilizador, é utilizada uma interface que permite ao utilizador aceder a todas as partes da aplicação de uma forma rápida e com pouco esforço. Como o sistema de recomendação permite ao utilizador poupar tempo, os níveis de stress deste não eram aumentados durante a pesquisa de um conteúdo que satisfaça as suas preferências. Desta forma, o utilizador não necessita de navegar no enorme volume de dados que pode ser a base de dados de conteúdos;

- **Aumentar os níveis de satisfação do utilizador -** Com tudo o que foi apresentado nos pontos anteriores, é possível melhorar os níveis de satisfação do utilizador, uma vez que o sistema de recomendação tenta satisfazer as suas preferências;

- **Perceber e tentar melhorar os atuais sistemas de recomendação de conteúdos -** Com a análise das técnicas estudadas, foi possível perceber como a maioria dos sistemas de recomendação funciona. Não foi possível distinguir se as técnicas utilizadas no ZenTV produzem melhores resultados que as técnicas utilizadas em outros sistemas de recomendação;

- **Tentar apontar os anúncios publicitários ao seu público alvo -** As regras de associação, bem como a divisão dos utilizadores em clusters, permite analisar o público alvo de um

determinado produto, permitindo orientar a publicidade aos potenciais compradores dum determinado produto;

- **Descobrir perfis de utilizador -** Com a análise do espaço de utilizadores, é possível desenvolver um conjunto de utilizadores que partilham características semelhantes, o que posteriormente permite sugerir conteúdos multimédia a utilizadores com base num conjunto de perfis semelhantes.

7.2 Trabalho Futuro

Como em qualquer projeto, existe sempre algo que poderia ser acrescentado, alterado ou melhorado, de modo a ficar mais consistente e/ou eficiente.

Um aspeto que foi impossível implementar, dadas as inúmeras temáticas abordadas, foi a capacidade de medir o nível de stress do utilizador. A melhor forma de o fazer seria de uma forma não invasiva e, se possível, de um modo transparente para o utilizador, como é caso do método descrito na secção 3.2. Isto seria útil para recomendar algo a um utilizador, com o intuito de lhe reduzir o nível de stress.

Como foi percetível, muitas técnicas de inteligência artificial ficaram por estudar e seria útil estudar mais e melhores formas de recomendação de conteúdos, elaborar novas combinações e perseguir o objetivo de melhorar os sistemas de recomendação de conteúdos multimédia.

Para otimizar a solução atual ou futuras, é extremamente recomendável otimizar o sistema de recomendação colocando *multithread*, ou até multi-máquina, de forma a responder aos pedidos sem dificuldade.

Para uma maior e melhor perceção da precisão do sistema, seria necessário realizar um estudo com conteúdos e pessoas reais, sendo necessário um parceiro que disponibilize os conteúdos duma forma legal, para evitar problemas com os direitos de autor.

Referências

[1] Seleccoes quando ler é um quebra-cabeças. www.seleccoes.pt/quando_ler_e_um_quebra_cabecas.html. Último acesso: 19-02-2014.

[2] Chris Kyriacou. Action research: a methodology for change and development - by bridget somekh. *British Journal of Educational Studies*, 55(4):468–469, 2007.

[3] Konstantinos Chorianopoulos and George Lekakos. Introduction to social tv: Enhancing the shared experience with interactive tv. *Intl. Journal of Human–Computer Interaction*, 24(2):113–120, 2008.

[4] S.J. McMillan and E.J. Downes. Defining interactivity: A qualitativeidentification of key dimensions. *New Media and Society*, pages 157–179, 2000.

[5] Laurence Habib. Interactive digital television — a literature review. *unknown*, pages 2–3, 2002.

[6] Yao Wang, Zhu Liu, and Jin-Cheng Huang. Multimedia content analysis. *unknown*, 1995.

[7] D. Lim, C. Bouchard, A. Aoussat, and Asian Society for the Science of Design. Interactive tv service design - rnrt (the french national network of researches in telecommunication) project. *unknown*, 2003.

[8] Jenny Preece. Online communities: designing usability, supporting sociability. *Industrial Management & Data Systems*, 100(9):459–460, 2000.

[9] Alexander Gruenstein Bo-June Paul Hsu James, Glass Stephanie Seneff, Lee Hetherington Scott Cyphers Ibrahim Badr, and Chao Wang Sean Liu. A multimodal home entertainment interface via a mobile device. In *Workshop on Mobile Language Processing*, page 1, 2008.

[10] Theodoros Bozios, Georgios Lekakos, Victoria Skoularidou, and Kostas Chorianopoulos. Advanced techniques for personalized advertising in a digital tv environment: the imedia system. In *Proceedings of the eBusiness and eWork Conference*, pages 1025–1031, 2001.

[11] Konstantinos Chorianopoulos, George Lekakos, and Diomidis Spinellis. Intelligent user interfaces in the living room: usability design for personalized television applications. In *Proceedings of the 8th international conference on Intelligent user interfaces*, pages 230–232. ACM, 2003.

[12] Sérgio Magno. Estudo, melhor experiência televisão, parte 1. *Exame Informática 237*, 2015.

[13] Sérgio Magno. Estudo, melhor experiência televisão, parte 2. *Exame Informática 238*, 2015.

[14] H Selye. The stress of life. *unknown*, 1956.

[15] Davide Carneiro, José Carlos Castillo, Paulo Novais, Antonio Fernández-Caballero, and José Neves. Multimodal behavioural analysis for non-invasive stress detection. *unknown*, 2012.

[16] Fábio Catalão. Analysis of the influence of stress on the interaction with the computer. *unknown*, 2013.

[17] Michael J Smith, Frank T Conway, and Ben-Tzion Karsh. Occupational stress in human computer interaction. *Industrial health*, 37(2):157–173, 1999.

[18] D. Gardner and D. Shoback. *Greenspan's Basic and Clinical Endocrinology, Ninth Edition*. LANGE Clinical Medicine. McGraw-Hill Education, 2011.

[19] Davide Carneiro, Ângelo Costa, Paulo Novais, Francisco Andrade, and José Neves. Providing relevant knowledge in disputes: Umcourt project. In *ODR*, pages 63–78, 2010.

[20] Jiawei Han and Micheline Kamber. *Data Mining: Concepts and Techniques*. Elsevier, 2006.

[21] Jiawei Han and Micheline Kamber. Cluster analysis. In *Data Mining: Concepts and Techniques*, pages 383–467. Elsevier, 2006.

[22] Arthur Cayley. A memoir on the theory of matrices. *Philosophical transactions of the Royal society of London*, pages 17–37, 1858.

[23] John A Hartigan and Manchek A Wong. Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, 28(1):100–108, 1979.

[24] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. Analysis of recommendation algorithms for e-commerce. *unknown*, 2000.

[25] Jiawei Han and Micheline Kamber. Mining frequent patterns, associations, and correlation. In *Data Mining: Concepts and Techniques*, pages 227–250. Elsevier, 2006.

[26] Chumki Basu, Haym Hirsh, and William Cohen. Recommendation as classification: Using social and content-based information in recommendation. *unknown*, 1998.

[27] Kilian Q Weinberger, John Blitzer, and Lawrence K Saul. Distance metric learning for large margin nearest neighbor classification. In *Advances in neural information processing systems*, pages 1473–1480, 2005.

[28] Jiawei Han and Micheline Kamber. Mining frequent patterns, associations, and correlation. In *k-Nearest-Neighbor Classifiers*, pages 348–350. Elsevier, 2006.

[29] Movie-map - find similar movies. http://www.movie-map.com/. Último acesso: 23-02-2014.

[30] Yan Chen, F Maxwell Harper, Joseph Konstan, and Sherry Xin Li. Social comparisons and contributions to online communities: A field experiment on movielens. *The American economic review*, 100(4):1358–1398, 2010.

[31] Jean Burgess and Joshua Green. *YouTube: Online video and participatory culture*. John Wiley & Sons, 2013.

[32] Jean Burgess and Joshua Green. Youtube e a revolução digital. *São Paulo: Aleph*, 2009.

[33] James Davidson, Benjamin Liebald, Junning Liu, Palash Nandy, Taylor Van Vleet, Ullas Gargi, Sujoy Gupta, Yu He, Mike Lambert, Blake Livingston, et al. The youtube video recommendation system. In *Proceedings of the fourth ACM conference on Recommender systems*, pages 293–296. ACM, 2010.

[34] James Bennett and Stan Lanning. The netflix prize. In *Proceedings of KDD cup and workshop*, volume 2007, page 35, 2007.

[35] Robert M Bell and Yehuda Koren. Lessons from the netflix prize challenge. *ACM SIGKDD Explorations Newsletter*, 9(2):75–79, 2007.

[36] Yehuda Koren. The bellkor solution to the netflix grand prize. *Netflix prize documentation*, 81:1–10, 2009.

[37] David Geer. Will software developers ride ruby on rails to success? *Computer*, 39(2):18–20, 2006.

[38] Michael Hartl. *Ruby on Rails Tutorial: Learn Web Development with Rails*. Pearson Education, 2012.

[39] Devise. https://github.com/plataformatec/devise. Último acesso: 09-05-2014.

[40] Cancan. https://github.com/ryanb/cancan. Último acesso: 23-06-2014.

[41] Paperclip. https://github.com/thoughtbot/paperclip. Último acesso: 07-06-2014.

[42] Font awesome. http://fortawesome.github.io/Font-Awesome. Último acesso: 15-05-2014.

[43] Font-awesome-rails. https://github.com/bokmann/font-awesome-rails. Último acesso: 15-05-2014.

[44] Faker. https://github.com/stympy/faker. Último acesso: 12-10-2014.

[45] Will paginate. https://github.com/mislav/will_paginate. Último acesso: 12-10-2014.

[46] Google chart. https://developers.google.com/chart. Último acesso: 17-03-2015.

[47] Bootstrap. http://getbootstrap.com. Último acesso: 17-06-2014.

[48] Luís Abreu. *JavaScript*. FCA, 2011.

[49] David Flanagan. *JavaScript: the definitive guide*. "O'Reilly Media, Inc.", 2006.

[50] Peter Lubbers, Brian Albers, Frank Salim, and Tony Pye. *Pro HTML5 programming*. Springer, 2011.

[51] Michael Widenius and David Axmark. *MySQL reference manual: documentation from the source*. "O'Reilly Media, Inc.", 2002.

[52] Oldest person ever. http://www.guinnessworldrecords.com/world-records/oldest-person. Último acesso: 20-11-2014.

8 Anexo

8.1 Povoamento da Base de Dados

Em seguida, é apresentada uma lista com toda a informação previamente carregada para a base de dados:

Lista de distritos utilizados:

- Aveiro
- Beja
- Braga
- Bragança
- Castelo Branco
- Coimbra
- Évora
- Faro
- Guarda
- Leiria
- Lisboa
- Portalegre
- Porto
- Santarém

- Setúbal
- Viana do Castelo
- Vila Real
- Viseu
- R. A. da Madeira
- R. A. dos Açores

Concelhos que pertencem ao distrito de Aveiro:

- Águeda
- Albergaria-a-Velha
- Anadia
- Arouca
- Aveiro
- Castelo de Paiva
- Espinho
- Estarreja
- Santa Maria da Feira
- Ílhavo
- Mealhada
- Murtosa
- Oliveira de Azeméis
- Oliveira do Bairro
- Ovar

- São João da Madeira
- Sever do Vouga
- Vagos
- Vale de Cambra

Concelhos que pertencem ao distrito de Beja:

- Aljustrel
- Almodôvar
- Alvito
- Barrancos
- Beja
- Castro Verde
- Cuba
- Ferreira do Alentejo
- Mértola
- Moura
- Odemira
- Ourique
- Serpa
- Vidigueira

Concelhos que pertencem ao distrito de Braga:

- Amares
- Barcelos

- Braga
- Cabeceiras de Basto
- Celorico de Basto
- Esposende
- Fafe
- Guimarães
- Póvoa de Lanhoso
- Terras de Bouro
- Vieira do Minho
- Vila Nova de Famalicão
- Vila Verde
- Vizela

Concelhos que pertencem ao distrito de Bragança:

- Alfândega da Fé
- Bragança
- Carrazeda de Ansiães
- Freixo Espada à Cinta
- Macedo de Cavaleiros
- Miranda do Douro
- Mirandela
- Mogadouro
- Torre de Moncorvo

- Vila Flor
- Vimioso
- Vinhais

Concelhos que pertencem ao distrito de Castelo Branco:

- Belmonte
- Castelo Branco
- Covilhã
- Fundão
- Idanha-a-Nova
- Oleiros
- Penamacor
- Proença-a-Nova
- Sertã
- Vila de Rei
- Vila Velha de Ródão

Concelhos que pertencem ao distrito de Coimbra:

- Arganil
- Cantanhede
- Coimbra
- Condeixa-a-Nova
- Figueira da Foz
- Góis

- Lousã
- Mira
- Miranda do Corvo
- Montemor-o-Velho
- Oliveira do Hospital
- Pampilhosa da Serra
- Penacova
- Penela
- Soure
- Tábua
- Vila Nova de Poiares

Concelhos que pertencem ao distrito de Évora:

- Alandroal
- Arraiolos
- Borba
- Estremoz
- Évora
- Montemor-o-Novo
- Mora
- Mourão
- Portel
- Redondo

- Reguengos de Monsaraz
- Vendas Novas
- Viana do Alentejo
- Vila Viçosa

Concelhos que pertencem ao distrito de Faro:

- Albufeira
- Alcoutim
- Aljezur
- Castro Marim
- Faro
- Lagos
- Loulé
- Monchique
- Olhão
- Portimão
- São Brás de Alportel
- Silves
- Tavira
- Vila do Bispo
- Vila Real Sto António

Concelhos que pertencem ao distrito de Guarda:

- Aguiar da Beira

- Almeida
- Celorico da Beira
- Fig. Castelo Rodrigo
- Fornos de Algodres
- Gouveia
- Guarda
- Manteigas
- Meda
- Pinhel
- Sabugal
- Seia
- Trancoso
- Vila Nova de Foz Côa

Concelhos que pertencem ao distrito de Leiria:

- Alcobaça
- Alvaiázere
- Ansião
- Batalha
- Bombarral
- Caldas da Rainha
- Castanheira de Pêra
- Figueiró dos Vinhos

- Leiria
- Marinha Grande
- Nazaré
- Óbidos
- Pedrógão Grande
- Peniche
- Pombal
- Porto de Mós

Concelhos que pertencem ao distrito de Lisboa:

- Alenquer
- Arruda dos Vinhos
- Azambuja
- Cadaval
- Cascais
- Lisboa
- Loures
- Lourinhã
- Mafra
- Oeiras
- Sintra
- Sobral de Monte Agraço
- Torres Vedras

- Vila Franca de Xira
- Amadora
- Odivelas

Concelhos que pertencem ao distrito de Portalegre:

- Alter do Chão
- Arronches
- Avis
- Campo Maior
- Castelo de Vide
- Crato
- Elvas
- Fronteira
- Gavião
- Marvão
- Monforte
- Nisa
- Ponte de Sor
- Portalegre
- Sousel

Concelhos que pertencem ao distrito de Porto:

- Amarante
- Baião

- Felgueiras
- Gondomar
- Lousada
- Maia
- Marco de Canaveses
- Matosinhos
- Paços de Ferreira
- Paredes
- Penafiel
- Porto
- Póvoa de Varzim
- Santo Tirso
- Valongo
- Vila do Conde
- Vila Nova de Gaia
- Trofa

Concelhos que pertencem ao distrito de Santarém:

- Abrantes
- Alcanena
- Almeirim
- Alpiarça
- Benavente

- Cartaxo
- Chamusca
- Constância
- Coruche
- Entroncamento
- Ferreira do Zezêre
- Golegã
- Mação
- Rio Maior
- Salvaterra de Magos
- Santarém
- Sardoal
- Tomar
- Torres Novas
- Vila Nova da Barquinha
- Ourém

Concelhos que pertencem ao distrito de Setúbal:

- Alcácer do Sal
- Alcochete
- Almada
- Barreiro
- Grândola

- Moita
- Montijo
- Palmela
- Santiago do Cacém
- Seixal
- Sesimbra
- Setúbal
- Sines

Concelhos que pertencem ao distrito de Viana do Castelo:

- Arcos de Valdevez
- Caminha
- Melgaço
- Monção
- Paredes de Coura
- Ponte da Barca
- Ponte de Lima
- Valença
- Viana do Castelo
- Vila Nova de Cerveira

Concelhos que pertencem ao distrito de Vila Real:

- Alijó
- Boticas

- Chaves
- Mesão Frio
- Mondim de Basto
- Montalegre
- Murça
- Peso da Régua
- Ribeira de Pena
- Sabrosa
- Sta Marta de Penaguião
- Valpaços
- Vila Pouca de Aguiar
- Vila Real

Concelhos que pertencem ao distrito de Viseu:

- Armamar
- Carregal do Sal
- Castro Daire
- Cinfães
- Lamego
- Mangualde
- Moimenta da Beira
- Mortágua
- Nelas

- Oliveira de Frades
- Penalva do Castelo
- Penedono
- Resende
- Santa Comba Dão
- São João da Pesqueira
- São Pedro do Sul
- Sátão
- Sernancelhe
- Tabuaço
- Tarouca
- Tondela
- Vila Nova de Paiva
- Viseu
- Vouzela

Concelhos que pertencem ao distrito de R. A. da Madeira:

- Calheta
- Câmara de Lobos
- Funchal
- Machico
- Ponta do Sol
- Porto Moniz

- Ribeira Brava
- Santa Cruz
- Santana
- São Vicente
- Porto Santo

Concelhos que pertencem ao distrito de R. A. dos Açores:

- Vila do Porto
- Lagoa
- Nordeste
- Ponta Delgada
- Povoação
- Ribeira Grande
- Vila Franca do Campo
- Angra do Heroísmo
- Praia da Vitória
- Santa Cruz da Graciosa
- Velas
- Lajes do Pico
- Madalena
- São Roque do Pico
- Horta
- Lajes das Flores

- Santa Cruz das Flores
- Corvo

Profissões

- Advogado
- Arquiteto
- Pescador
- Biólogo
- Cozinheiro
- Dietista
- Docente do Ensino Superior
- Educador de Infância
- Eletricista
- Enfermeiro
- Engenheiro Civil
- Engenheiro de Materiais
- Engenheiro do Ambiente
- Engenheiro Eletrotécnico
- Engenheiro Florestal
- Engenheiro Geógrafo
- Engenheiro Informático
- Engenheiro Mecânico
- Engenheiro Metalúrgico e de Materiais

- Engenheiro Naval
- Engenheiro Químico
- Enólogo
- Farmacêutico
- Fisioterapeuta
- Maquinista
- Marinheiro
- Mecânico
- Médico
- Mergulhador
- Motorista
- Nutricionista
- Professor
- Psicólogo
- Soldador
- Solicitador
- Técnico
- Terapeuta da fala
- Terapeuta
- Economista

Categorias

- Comédia
- Terror

- Thriller
- Romance
- Ação
- Crime
- Drama
- Mistério
- Ficção Científica
- Aventura
- Documentário
- Biografia
- Desporto
- Fantasia
- História
- Musical
- Western
- Familiar
- Animação
- Policial
- Guerra

8.2 Diagramas em Tamanho A4

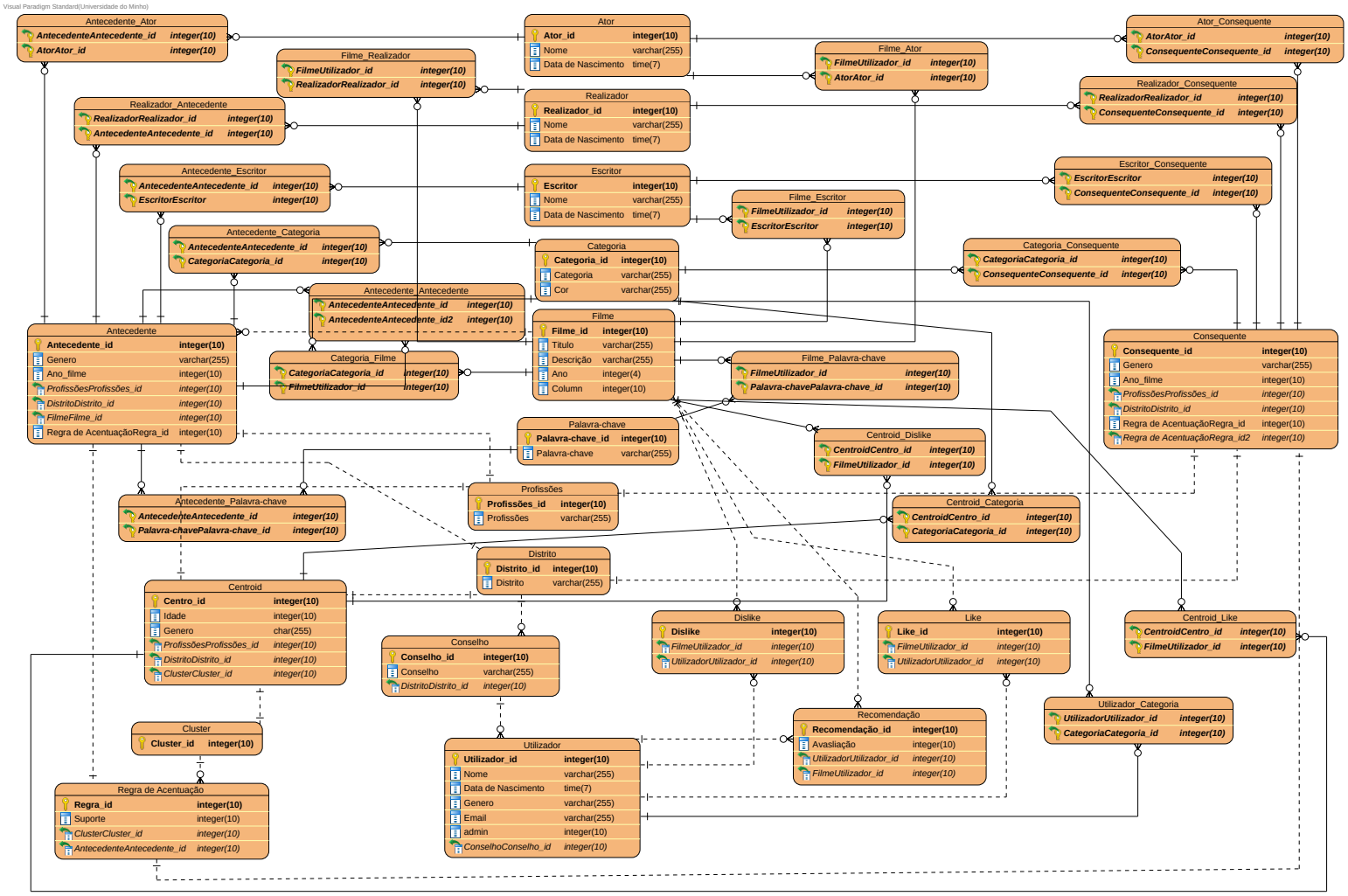


Figura 8.1: Modelo de Dados Utilizada (A4)

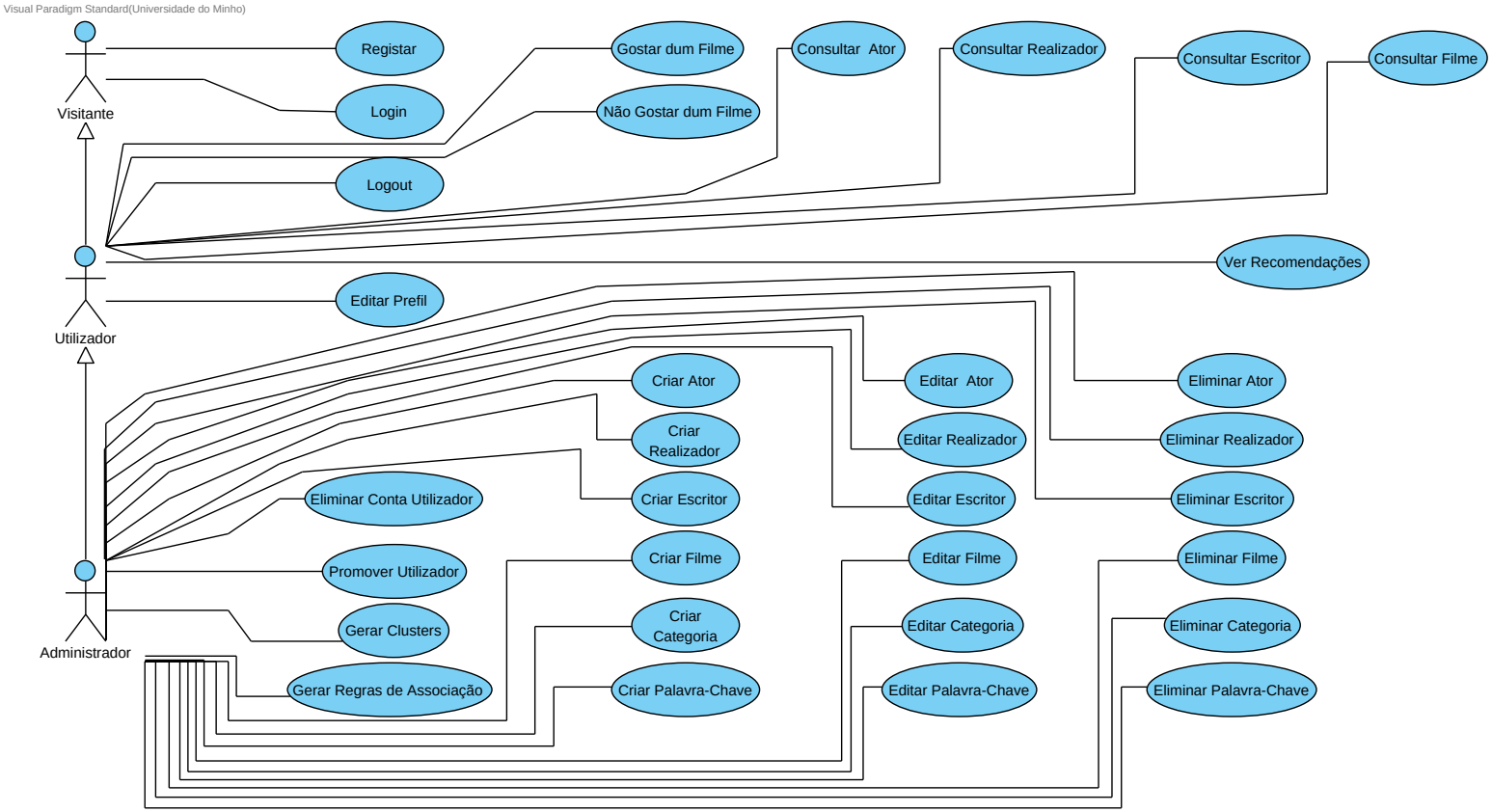


Figura 8.2: Diagrama de Use Case - ZenTV (A4)

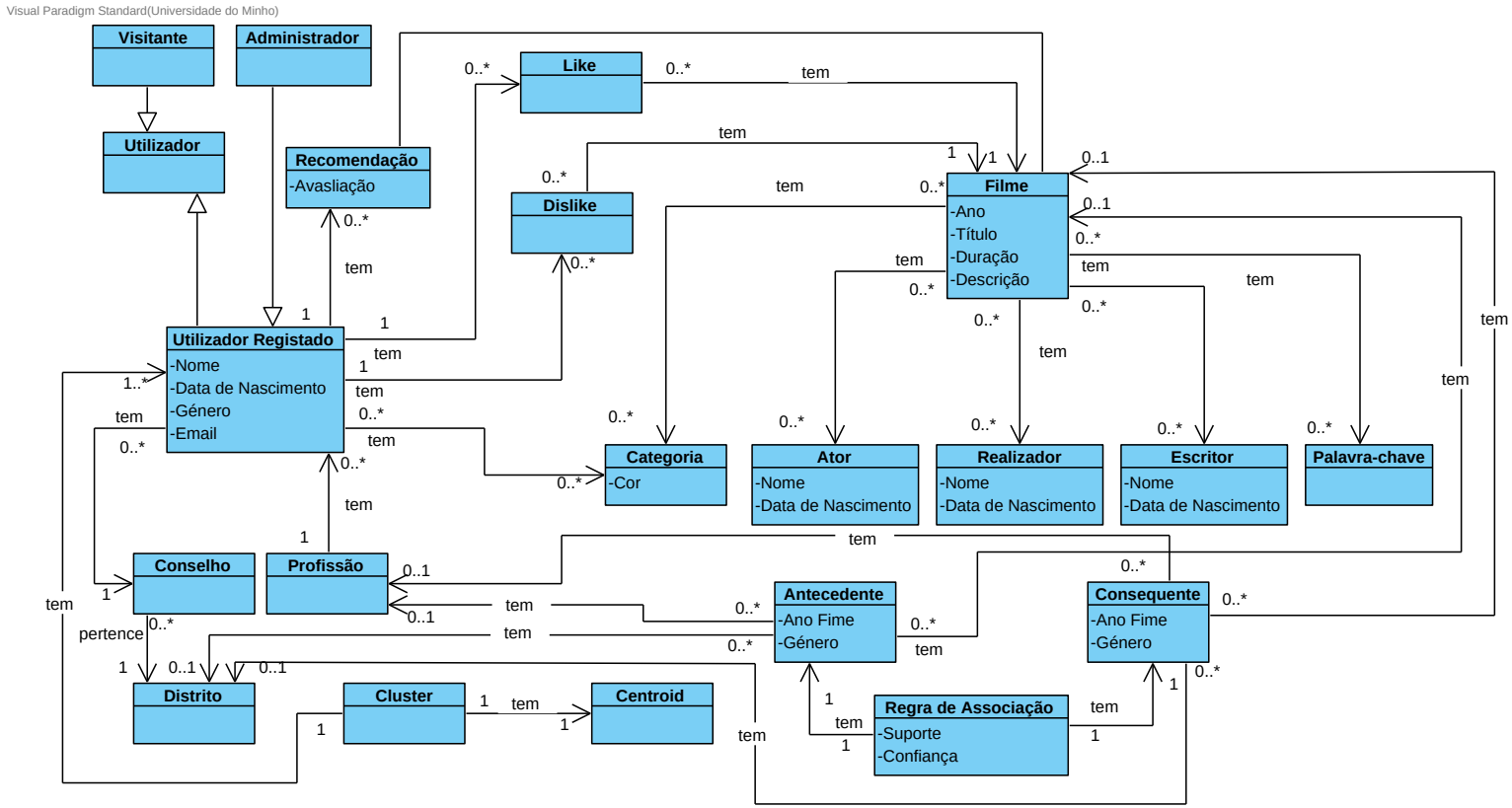


Figura 8.3: Modelo de Domínio - ZenTV (A4)

8.3 Manual de instalação

Para que seja possível colocar o projeto a funcionar, é necessário um sistema operativo Ubuntu Server, ou similar. No sistema operativo, é necessário instalar alguns componentes. Abaixo seguem os passos de instalação.

8.3.1 Instalar Ruby on Rails

O projeto foi desenvolvido em Ruby on Rails 4. Para que seja possível executá-lo, é necessário instalar o Rails e para tal devem-se executar os seguintes comandos no terminal:

```
sudo apt-get update
sudo apt-get install git-core curl zlib1g-dev build-essential libssl-d

wget http://ftp.ruby-lang.org/pub/ruby/2.2/ruby-2.2.3.tar.gz
tar -xzvf ruby-2.2.3.tar.gz
cd ruby-2.2.3/
./configure
make
sudo make install
ruby -v

sudo gem install bundler
sudo apt-get install -y nodejs

sudo gem install rails -v 4.0.13
```

8.3.2 Instalar MySQL

O projeto foi desenvolvido utilizando a base de dados MySQL. Para que seja possível executá-lo, é necessário instalar esta base de dados e para tal devem-se executar os seguintes comandos no terminal:

```
sudo apt-get install mysql-server mysql-client libmysqlclient-dev

sudo apt-get install imagemagick
```

8.3.3 Sublime Text

Para a leitura do código fonte, é recomendado o uso do Sublime Text, para o instalar é necessário seguir os seguintes passos:

```
sudo add-apt-repository ppa:webupd8team/sublime-text-3
sudo apt-get update
sudo apt-get install sublime-text-installer
```

8.3.4 Colocar a aplicação em funcionamento

Para colocar a aplicação em funcionamento, é necessário ir para a pasta onde está o código fonte e depois executar os seguintes comandos no terminal:

```
bundle install

rake db:setup

rails s
```