

**FUNDAÇÃO GETULIO VARGAS
ESCOLA BRASILEIRA DE ECONOMIA E FINANÇAS**

RAYANA DE SOUZA LAURIA

APLICAÇÃO DE SISTEMAS DE RECOMENDAÇÃO EM FINANÇAS

Rio de Janeiro

2020

RAYANA DE SOUZA LAURIA

APLICAÇÃO DE SISTEMAS DE RECOMENDAÇÃO EM FINANÇAS

Dissertação para obtenção do grau de mestre apresentada
à Escola Brasileira de Economia e Finanças

Área de concentração: Data Science

Orientador: Genaro Dueire Lins

Rio de Janeiro

2020

Lauria, Rayana de Souza

Aplicação de sistemas de recomendação em finanças / Rayana de Souza
Lauria. – 2020.

135 f.

Dissertação (mestrado) - Fundação Getulio Vargas, Escola Brasileira
de Economia e Finanças.

Orientador: Genaro Dueire Lins.

Inclui bibliografia.

1. Comércio eletrônico. 2. Ciência - Processamento de dados.
3. Aprendizado do computador. 4. Bancos - Automação. 5. Finanças - Inovações
Tecnológicas. I. Lins, Genaro Dueire. II. Fundação Getulio Vargas Escola
Brasileira de Economia e Finanças. III. Título.

CDD – 332

RAYANA DE SOUZA LAURIA


"APLICAÇÃO DE SISTEMAS DE RECOMENDAÇÃO EM FINANÇAS".

Tese apresentado(a) ao Curso de Mestrado Profissional em Economia Empresarial e Finanças do(a) EPGE
Escola Brasileira de Economia e Finanças - FGV EPGE para obtenção do grau de Mestre(a) em Economia
Empresarial e Finanças.

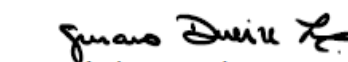
Data da defesa: 08/12/2020

ASSINATURA DOS MEMBROS DA BANCA EXAMINADORA

Presidente da Comissão Examinadora: Prof^o/a Genaro Dueire Lins


Genaro Dueire Lins
Orientador


por Edson Daniel Lopes Gonçalves
Membro Interno


por Rafael Martins de Souza
Membro Interno

Nos termos da Lei nº 13.979 de 06/02/20 - DOU nº 27 de 07/02/20 e Portaria MEC nº 544 de 16/06/20 - DOU nº 114 de 17/06/20 que dispõem sobre a suspensão temporária das atividades acadêmicas presenciais e a utilização de recursos tecnológicos face ao COVID-19, as apresentações das defesas de Tese e Dissertação, de forma excepcional, serão realizadas de forma remota e síncrona, incluindo-se nessa modalidade membros da banca e discente.



Ricardo de Oliveira Cavalcanti
Coordenador



Antonio de Araujo Freitas Junior
Pró-Reitor de Ensino, Pesquisa e Pós-Graduação FGV

Antonio Freitas, PhD
Pró-Reitor de Ensino, Pesquisa e Pós-Graduação
Fundação Getúlio Vargas

Instrução Normativa nº 01/19, de 09/07/19 - Pró-Reitoria FGV

Em caso de participação de Membro(s) da Banca Examinadora de forma não-presencial*, o Presidente da Comissão Examinadora assinará o documento como representante legal, delegado por esta I.N.

*Skype, Videoconferência, Apps de vídeo etc

A Deus, por me proporcionar mais essa oportunidade única, e aos meus pais, Mirian e Victor, que estiveram sempre incansáveis ao meu lado vibrando a cada conquista.

AGRADECIMENTOS

O aprendizado ao longo de um mestrado profissional é construído por um conjunto de elementos. Apesar da dedicação do aluno ser um fator essencial para o processo, definitivamente não é a única.

Por esse motivo, não poderia deixar de agradecer a todos os professores que me transferiram parte de seus conhecimentos nesta trajetória e também durante toda a minha vida acadêmica, pois cada um deles teve seu grau de contribuição para que eu chegasse até aqui. Portanto, agradeço ao grupo de professores da EPGE e, em especial, ao meu orientador, Professor Genaro Lins, pela dedicação ao me orientar no presente trabalho.

Agradeço também aos meus colegas de trabalho, que me apoiaram nesta caminhada difícil de conciliação da vida profissional e acadêmica. Sem seu suporte, seria impossível concluir este projeto.

Por fim, agradeço aos meus pais, Mirian e Victor, pelo apoio e dedicação nessa jornada e em todas as outras da minha vida. Vocês são o alicerce que me incentiva a seguir em frente sempre e nunca desistir dos meus sonhos.

RESUMO

O presente trabalho busca associar algoritmos de recomendação, tão presentes na atual era do *e-commerce*, a bases de produtos financeiros das plataformas digitais em um projeto de ciência de dados. Neste estudo, portanto, será realizada uma revisão bibliográfica dos sistemas já existentes e presentes no cotidiano dos consumidores virtuais para produtos normais, como livros e filmes, e das aplicações existentes de tais sistemas para recomendar produtos financeiros. A ideia central é a incorporação dos parâmetros de finanças que tornam um determinado produto em finanças recomendável ao perfil do investidor, dado seu padrão de consumo na base de transações. Para isso, serão aplicados sistemas de recomendação de aprendizado de máquina a bases transacionais de produtos financeiros. Os resultados obtidos permitem uma avaliação preliminar desta aplicação, abrindo portas para estudos futuros relacionados ao tema.

Palavras-chave: Ciência de Dados. Sistemas de Recomendação. *Machine Learning*. Banco Digital. Produtos Financeiros.

ABSTRACT

This work attempts to associate recommendation system algorithms, often present at the e-commerce era, to digital bank's financial products datasets in a data science project. Therefore, this study will make a survey on existing recommendation systems, normally seen and present on customers lives to normal products, as books and movies for instance, and the existing applications at financial products. The main idea is to incorporate the financial parameters, which make a specific financial product interesting to investor's profile, known your consume pattern. With this purpose, machine learning recommendation system algorithms will be applied to financial exchange datasets. The results allow to preliminarily evaluate this kind of application, paving the way for further studies related to the subject.

Keywords: Data Science. Recommendation Systems. Machine Learning. Digital Bank. Financial Products.

LISTA DE ILUSTRAÇÕES

Figura 1: Evolução transações bancárias por canal. Fonte: Pesquisa FEBRABAN de Tecnologia Bancária 2019.....	26
Figura 2 : Contas com uso de mobile banking e internet banking (*contas com movimentação nos últimos 6 meses). Fonte: Pesquisa FEBRABAN de Tecnologia Bancária 2019	26
Figura 3: Composição dos gastos com tecnologia por setor no Brasil e principais áreas de investimentos do setor bancário. Fonte: Pesquisa FEBRABAN de Tecnologia Bancária 2019.....	27
Figura 4: Evolução da quantidade de pessoas físicas na BM&FBovespa (valores atualizados até maio/20).	29
Figura 5: Volume de Assets under Custody por plataforma em bilhões de reais. Órama e BTG Digital não divulgaram seus valores. Fonte: Valor Investe (dados de julho/2019).....	29
Figura 6: Portfolio produtos de prateleira ex-home broker das plataformas fora dos grandes bancos. Dados de julho de 2019. Fonte: Valor Investe.	30
Figura 7: Clusters de ativos no mercado financeiro. Fonte: A autora	34
Figura 8: Caracterizando um ativo específico no mercado financeiro. Fonte: A autora.	34
Figura 9: Ilustração de filtro baseado em conteúdo. Fonte: Google.....	39
Figura 10: Ilustração de filtro colaborativo. Fonte: Google.	41
Figura 11: Exemplo de tabela input	45
Figura 12: Resumo abordagens sistemas de recomendação. Fonte: a autora, baseada na descrição de Shah et al.	50
Figura 13: Resumo da abordagem específica utilizada no caso Amazon.com conforme classificação do capítulo 3. Fonte: a autora	53
Figura 14: Exemplo mini-web. Fonte: Rpubs, professor Adriano Azevedo Filho (USP).....	55
Figura 15: Matrizes P e π . Fonte: Rpubs, professor Adriano Azevedo Filho (USP).	55
Figura 16: Resumo da abordagem específica utilizada no caso Pagerank conforme classificação do capítulo 3. Fonte: a autora.	56
Figura 17: Visão atual dos gerentes de alta renda de uma instituição financeira. Fonte: Grupo Incube, 2018.....	60
Figura 18: Aplicando sistemas de recomendação à base de clientes completa. Fonte: GrupoIncube, 2018.....	60
Figura 19: Ilustração do problema da cauda longa na distribuição de produtos para sistemas de recomendação em finanças. Fonte: Site do Grupo Incube.	62
Figura 20: Resumo da abordagem utilizada pelo Grupo Incube em suas soluções. Fonte: A autora....	63
Figura 21: Composição geográfica dos autores que propuseram artigos ao The Review of Financial Studies em 2017	64
Figura 22: Resumo da abordagem utilizada no artigo de Sayyed. Fonte: A autora.	65
Figura 23: Resumo da abordagem utilizada no paper de Zhang et al (2018). Fonte: A autora.....	68
Figura 24: Resumo dos sistemas de recomendação como um serviço citados pelo Github. Fonte: a autora.....	72
Figura 25: Principais pacotes open source em Python e R de recomendação e seus métodos. Fonte: a autora com base em Jenson.	73
Figura 26: Incluindo os sistemas multimodais. Fonte: a autora.	74
Figura 27: Principais pacotes open source em outras linguagens. Fonte: a autora com base em Jenson.	75
Figura 28: Resumo da base de ações inserida no software R. Fonte: a autora.....	77
Figura 29: Resumo das observações por setor da B3. Fonte: a autora.	78
Figura 30: Código para visualizar a imagem da base. Fonte: a autora.....	78

Figura 31: Imagem da base de ações. Fonte: R.	79
Figura 32: Histograma dos ratings na base de ações. Fonte: R.	79
Figura 33: Resumo da base de fundos de investimento inserida no R. Fonte: R.....	80
Figura 34: Resumo das observações por tipo. Fonte: a autora.	81
Figura 35: Imagem da base de fundos. Fonte: R.	82
Figura 36: Histograma dos ratings na base de fundos. Fonte: R.	82
Figura 37: Criando um sistema de recomendação de vizinhança focado no usuário e calculando as medidas de desempenho para a base de ações. Usando o predict para gerar ratings. Fonte: R.....	84
Figura 38: Criando um sistema de recomendação de vizinhança focado no usuário e calculando as medidas de desempenho para a base de ações. Usando o predict para gerar top-N recomendações. Fonte: R.	84
Figura 39: Aplicando a fatoração matricial de Simon Funk na base de ações. Fonte: R.....	85
Figura 40: Criando um sistema de recomendação UBCF para a base de fundos. Função predict usada para gerar ratings. Fonte: R.	85
Figura 41: Criando um sistema de recomendação UBCF para a base de fundos. Função predict usada para gerar top-N recomendações. Fonte: R.	86
Figura 42: Aplicando a fatoração matricial de Simon Funk na base de ações. Fonte: R.....	86
Figura 43: Ilustração dos métodos de recomendação nos quais o recommenderlab se concentra. Fonte: A autora.	106
Figura 44: Histograma do número de ratings por usuário da base Movielense. Fonte: Hashler (2016).	107
Figura 45: Resumo da infraestrutura de recomendação do pacote recommenderlab. Fonte: A autora.	108
Figura 46: Linhas de comando de exemplo no R para a aplicação do recommender na base MSWeb. Fonte: Hahsler (2016).....	109
Figura 47: Resultados da aplicação do trio de funções na base MSWeb. Fonte: Hahsler.....	110
Figura 48: Gerando um sistema de recomendação híbrido. Fonte: Hahsler (2016).	111
Figura 49: Aplicando a função de fatoração matricial de Simon Funk. Fonte: Hahsler (2016).	111
Figura 50: Parâmetros dos métodos do pacote recommenderlab	114

LISTA DE TABELAS

Tabela 1: Setores presentes na base transacional de ações do presente trabalho. Fonte: B3 resumida pela autora.	77
Tabela 2: Resumo descritivo base ações. Fonte: a autora.	77
Tabela 3: Tipos de fundo presentes na base transacional do presente trabalho. Fonte: a autora.	80
Tabela 4: Resumo descritivo base fundos. Fonte: a autora.	80
Tabela 5: Exemplo de matriz construída a partir do input	111
Tabela 6: Matriz de confusão	112

SUMÁRIO

1. INTRODUÇÃO	21
2. PLATAFORMAS DIGITAIS DE INVESTIMENTO	24
2.1. AVANÇO DAS PLATAFORMAS DIGITAIS DE INVESTIMENTO NO BRASIL.....	24
2.2. VARIEDADE DAS BASES DE DADOS DE UMA PLATAFORMA DIGITAL.....	32
3. SISTEMAS DE RECOMENDAÇÃO: ORIGEM, ABORDAGENS E MÉTODOS	36
3.1. A ORIGEM: SISTEMAS DE APRENDIZADO DE MÁQUINA.....	37
3.2. ABORDAGENS DOS SISTEMAS DE RECOMENDAÇÃO.....	38
3.2.1. FILTRO BASEADO EM CONTEÚDO	39
3.2.2. FILTRO COLABORATIVO.....	41
3.2.2.1. FILTRO COLABORATIVO BASEADO EM VIZINHANÇA	42
3.2.2.2. FILTRO COLABORATIVO BASEADO EM MODELO	44
3.2.2.3. LIMITAÇÕES DO FILTRO COLABORATIVO.....	47
3.2.3. MODELOS DE CLUSTER.....	48
3.2.4. ABORDAGEM HÍBRIDA.....	49
3.3. MEDIDAS DE DESEMPENHO DOS MODELOS.....	49
4. EXEMPLOS PRÁTICOS DE REFERÊNCIA	52
4.1. O CASO AMAZON.COM	52
4.2. O CASO PAGERANK (GOOGLE INC)	54
5. APLICAÇÕES DE SISTEMAS DE RECOMENDAÇÃO EM FINANÇAS: REFERENCIAL TEÓRICO	58
5.1. DESAFIOS E BENEFÍCIOS ENVOLVIDOS NA APLICAÇÃO EM FINANÇAS	58
5.2. EXEMPLOS DE APLICAÇÕES DE REFERÊNCIA EM FINANÇAS	61
6. SISTEMAS DE RECOMENDAÇÃO DISPONÍVEIS NO MERCADO	71
6.1. SISTEMAS DE RECOMENDAÇÃO COMO UM SERVIÇO	71
6.2. SISTEMAS DE RECOMENDAÇÃO <i>OPEN SOURCE</i>	73
7. PROJETO DE CIÊNCIA DE DADOS: APLICAÇÃO PRÁTICA EM BASES TRANSAÇIONAIS.....	76
7.1. COLETA DOS DADOS, TRATAMENTO PARA <i>INPUT</i> E DESCRIÇÃO	76
7.2. APLICAÇÃO DOS MÉTODOS DE RECOMENDAÇÃO E DESEMPENHO.....	83
8. CONSIDERAÇÕES FINAIS.....	88
REFERÊNCIAS	90

APÊNDICE A - OUTROS EXEMPLOS MAIS GERAIS DE TRABALHOS NO DOMÍNIO DE AÇÕES.....	95
APÊNDICE B - OUTROS EXEMPLOS MAIS GERAIS DE TRABALHOS NO DOMÍNIO DE PORTFOLIOS.....	96
APÊNDICE C - SISTEMAS DE RECOMENDAÇÃO COMO UM SERVIÇO (GITHUB)	97
APÊNDICE D - SISTEMAS DE RECOMENDAÇÃO <i>OPEN SOURCE</i> (GITHUB)	101
APÊNDICE E - DETALHANDO O RECOMMENDERLAB: MÉTODOS, PARÂMETROS E HIPERPARÂMETROS.....	105
RECOMMENDERLAB: MÉTODOS	105
RECOMMENDERLAB: PARÂMETROS.....	113
RECOMMENDERLAB: HIPERPARÂMETROS	115
APÊNDICE F - EXEMPLOS DE APLICAÇÃO DO <i>RECOMMENDERLAB</i> DE HAHSLER (2016)	116
APÊNDICE G - APLICAÇÃO DETALHADA NA BASE DE AÇÕES	124
APÊNDICE H - APLICAÇÃO DETALHADA NA BASE DE FUNDOS.....	131

1. INTRODUÇÃO

Em uma era voltada para o *e-commerce*, na qual consumidores têm acesso a diversas plataformas e uma infinidade de produtos em cada uma delas, observa-se cada vez mais a necessidade de sugestões que se adequem de imediato ao seu perfil para ampliar o potencial das vendas. Com o aumento da complexidade das bases de dados, frequentemente os usuários nem sequer desconfiam da existência de opções interessantes que estão escondidas em um mar de informações e necessitam apenas de um filtro para serem encontradas. Foi nesse cenário que surgiram os sistemas de recomendação, possibilitando uma filtragem inteligente das alternativas do mundo virtual de acordo com as preferências do cliente.

Segundo Shah et al. (2016), o primeiro sistema de recomendação desenvolvido foi o Tapestry em 1992, um sistema de mensagens eletrônicas que permitia que os usuários as classificassem entre “gostou” ou “não gostou”. Posteriormente, surgiram novas aplicações que se enraizaram tanto no cotidiano das pessoas a ponto de se tornarem imperceptíveis, possibilitando a predição de seus gostos peculiares para aprimorar serviços e negociações.

Aplicações simples como o Microsoft Word ou os teclados dos *smartphones* que recomendam palavras a quem está escrevendo são exemplos práticos de quão comuns algoritmos de recomendação se tornaram no dia-a-dia, bem como os sites que personalizam sua página inicial de acordo com as preferências de um cliente específico ou as plataformas de *streaming* que utilizam o histórico de escolhas do usuário, localização, semelhança de perfil com outros usuários, *page ranking* entre outras abordagens a fim de realizar as melhores sugestões possíveis para um indivíduo e aprimorar sua experiência na plataforma digital.

Simultaneamente ao processo de ascensão dos algoritmos de recomendação na web, que revolucionaram o mundo das *marketplaces* e plataformas de *streaming*, observou-se uma mudança de comportamento dos consumidores brasileiros em relação a administração de suas finanças pessoais.

Segundo a pesquisa Febraban de tecnologia bancária de 2019 é possível perceber uma mudança de comportamento significativa da população brasileira, que está cada vez mais inserida no mundo *mobile*, dando preferência a transações realizadas pelo celular ou *internet banking*. É notável a evolução do chamado *mobile banking* bem como o investimento massivo das instituições financeiras em tecnologia para atender um mercado em que este é o principal

canal das movimentações. O consumidor está cada vez menos dependente das agências bancárias e cada vez mais direcionado às contas 100% digitais.

Outra mudança importante observável é o crescimento das corretoras digitais frente aos bancos tradicionais que por muito tempo foram considerados a única opção segura para investidores do varejo que buscavam um local para centralizar suas aplicações. Segundo uma reportagem da Exame de agosto de 2019, as plataformas de investimentos já atraíram mais de 10% dos 2,98 trilhões de reais dos brasileiros aplicados em fundos mútuos, ações e títulos. Tais plataformas estão reunindo esforços para se tornarem bancos digitais, concentrando em um só local serviços integrados de corretora de valores e suas aplicações com conta corrente, pagamentos de boletos e cartões de crédito, por exemplo, atraindo um novo consumidor ávido por tecnologia associada a praticidade no mundo moderno.

Neste novo cenário, o investimento massivo em tecnologia está se intensificando cada vez mais, na disputa por clientes que buscam melhores opções de investimentos e plataformas que ofereçam os melhores serviços. Foi possível verificar até mesmo uma movimentação dos bancos tradicionais para modernizar suas atividades e se adequar a uma nova realidade. A tentativa de aquisição da plataforma XP Investimentos pelo Itaú foi uma clara movimentação no sentido de direcionar sua estratégia para um novo campo, abrangendo um novo mercado, um novo perfil de cliente. O acordo de aquisição de até 35% do Modalmais pelo Credit Suisse também pode ser visto nesse sentido, uma parceria entre a tradição dos bancos tradicionais e a inovação dos bancos digitais.

Diante disso, a ideia deste estudo é associar os algoritmos de recomendação personalizados, já presentes de inúmeras formas no cotidiano dos consumidores, aos serviços das plataformas digitais financeiras, auxiliando clientes a encontrar alternativas interessantes de investimento presentes nas bases de dados e melhorar sua experiência de navegação nos aplicativos.

Para cumprir os objetivos propostos, este trabalho foi dividido em oito capítulos, além desta introdução. No capítulo 2, aborda-se o avanço das plataformas digitais de investimento no Brasil e a variedade de suas bases de dados. No capítulo 3 detalham-se os sistemas de recomendação, sua origem, suas abordagens, métodos e medidas de desempenho. No 4 são dados dois exemplos práticos de referência em sistemas de recomendação: os casos Amazon.com e Google. No quinto capítulo é descrito o problema da recomendação envolvendo finanças e no sexto capítulo são apresentados os sistemas de recomendação disponíveis no mercado. Por fim, no sétimo capítulo há um projeto de ciência de dados, no qual os métodos

abordados são aplicados a uma base transacional da bolsa e a uma de fundos de investimento e no oitavo há a conclusão da pesquisa realizada.

2. PLATAFORMAS DIGITAIS DE INVESTIMENTO

“Os investidores brasileiros têm ao menos 8 bons motivos para desbancarizar os investimentos.”

(Roberto Lee, diretor de marketing da XP Investimentos)

Conforme já descrito na introdução deste trabalho, está em curso uma mudança de rumo no comportamento do investidor brasileiro, o qual cada vez mais deixa o mundo concentrado dos grandes bancos, com sua quantidade limitada de produtos e suas agências para ingressar em um mundo digital, com grande variedade de opções de investimento. O objetivo deste capítulo é descrever o avanço das chamadas plataformas digitais de investimento no Brasil e a diversidade de opções que possibilitam.

2.1. AVANÇO DAS PLATAFORMAS DIGITAIS DE INVESTIMENTO NO BRASIL

Por décadas no mercado financeiro brasileiro vigorou uma estrutura fechada na qual bancos ofertavam apenas produtos desenvolvidos internamente. Reconhecidamente a arquitetura desse campo no Brasil é considerada limitada e concentrada nos bancos do chamado S1¹. Segundo uma matéria do Valor Investe (2020), pouco mais de 80% do total de dinheiro guardado pelos brasileiros está concentrado nos cinco grandes bancos do país, enquanto nos EUA observa-se uma situação praticamente inversa.

Contudo, é notável que este cenário vem sofrendo mudanças desde 2010, quando a XP capitaneou o surgimento das *marketplaces* de produtos financeiros, quebrando a tradicional verticalização existente e trazendo estruturas que atuam como plataformas de dois lados, aproximando ofertantes e consumidores de investimentos.

O movimento de “desbancarização”, como passou a ser chamado o processo de migração dos clientes de varejo das instituições tradicionais para as plataformas abertas, surgiu com um amplo potencial de desenvolvimento, suportado ainda por inovações tecnológicas e uma série de alterações regulatórias que favorecem o movimento, segundo informações

¹ Há uma segmentação das instituições financeiras no Sistema Financeiro Nacional que divide o setor em cinco segmentos: S1 seriam os bancos com porte igual ou acima de 10% do PIB ou internacionalmente ativos, S2 entre 1% e 10%, S3 entre 0,1% e 1%, S4 inferior a 0,1% e S5 cooperativas de crédito ou instituições não bancárias com perfil de risco simplificado.

coletadas no Ato de Concentração do CADE relativo ao caso Itaú/XP Investimentos² (2017), tais como:

- Possibilidade de realizar um cadastro digital sem comparecer a agências físicas;
- Aumento do limite do Fundo Garantidor de crédito de R\$ 70 mil para R\$ 250 mil, o que incentiva o pequeno investidor a investir em produtos de instituições de pequeno e médio porte, as quais oferecem maiores rentabilidades para um maior risco;
- Desenvolvimento do tesouro direto, que permite que a população tenha acesso aos títulos do governo sem a intermediação de um banco;
- Incentivos fiscais para aplicações em Letras de Crédito (do Agronegócio – LCAs – e da Indústria – LCIs, por exemplo);
- Avanço das plataformas digitais, *mobile* e redes sociais, o que reduz o custo das instituições financeiras, que não precisam mais manter redes físicas de atendimento.

A Pesquisa FEBRABAN de Tecnologia Bancária de 2019 em conjunto com a Deloitte reforça ainda mais essa tendência do mercado financeiro no Brasil, demonstrando que houve uma mudança de comportamento dos clientes no sentido de priorizar o *mobile banking* para realizar transações, em detrimento do *internet banking* e do comparecimento às agências bancárias. O avanço do canal digital nas transações ficou bastante claro, bem como o avanço da abertura de contas digitais.

O número de contas abertas pelo celular saltou em 56% de 2017 para 2018 e pela primeira vez o número de contas operadas por smartphones superou as contas com internet banking. As chamadas contas digitais são disponibilizadas tanto pelos principais bancos do país quanto pelas *fintechs* que passaram a integrar o atual ecossistema financeiro no Brasil. A gratuidade da operação de tais contas é garantida pela Resolução nº 3.919/10 do Banco Central do Brasil.

2 Ato de Concentração nº 08700.004431/2017-16; ver referência 19.

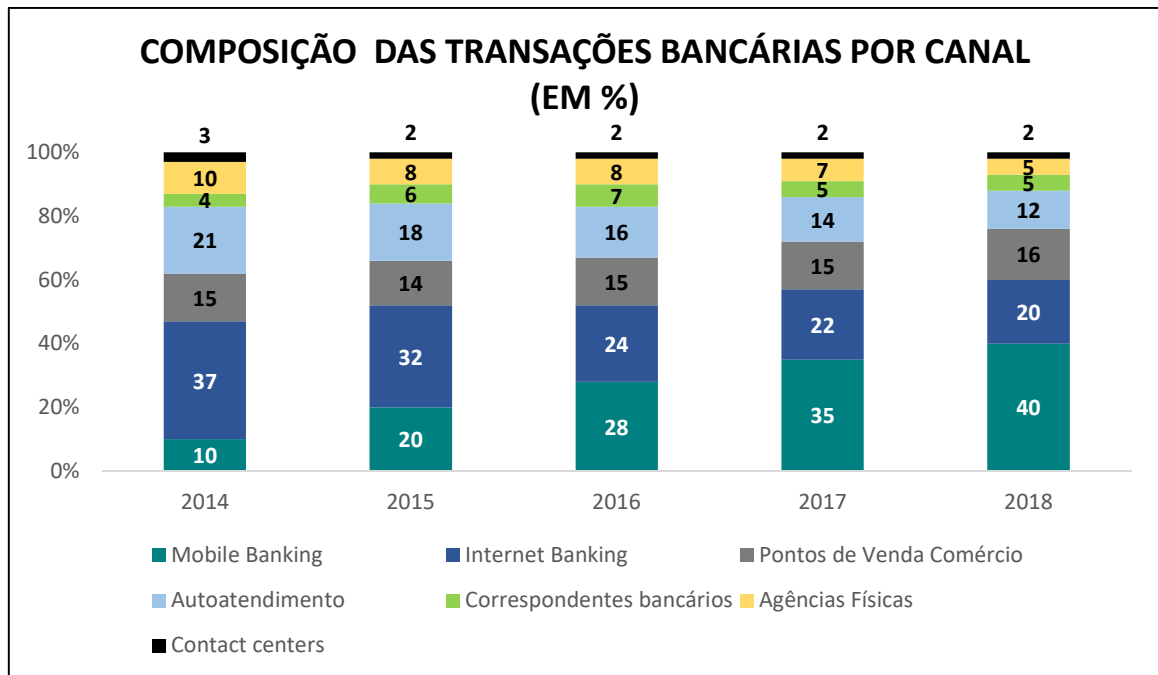


Figura 1: Evolução transações bancárias por canal. Fonte: Pesquisa FEBRABAN de Tecnologia Bancária 2019

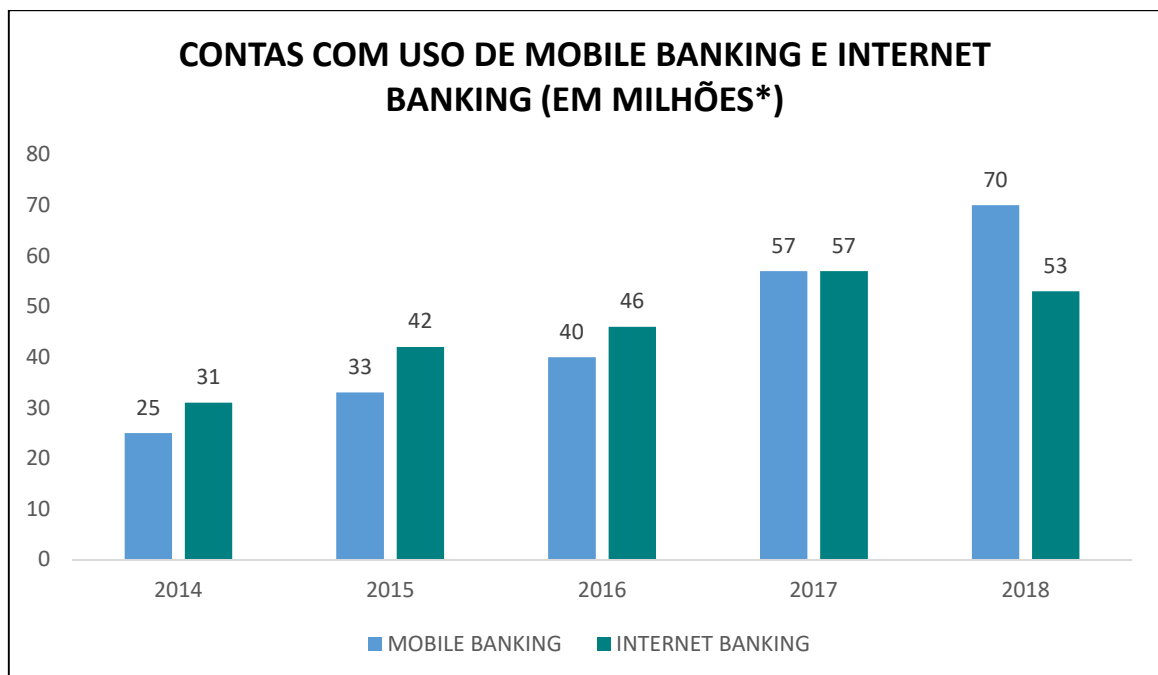


Figura 2 : Contas com uso de mobile banking e internet banking (*contas com movimentação nos últimos 6 meses). Fonte: Pesquisa FEBRABAN de Tecnologia Bancária 2019

Além disso, é visível o aumento dos investimentos dos bancos em tecnologia, principalmente em big data, *analytics* e inteligência artificial para melhorar a relação com os consumidores. Globalmente, o setor bancário é o segundo maior em investimentos de tecnologia, ficando atrás apenas dos governos e superando a indústria de telecomunicações. No Brasil, 2018 foi o primeiro ano no qual os gastos do setor bancário com tecnologia atingiram o mesmo patamar do setor público.

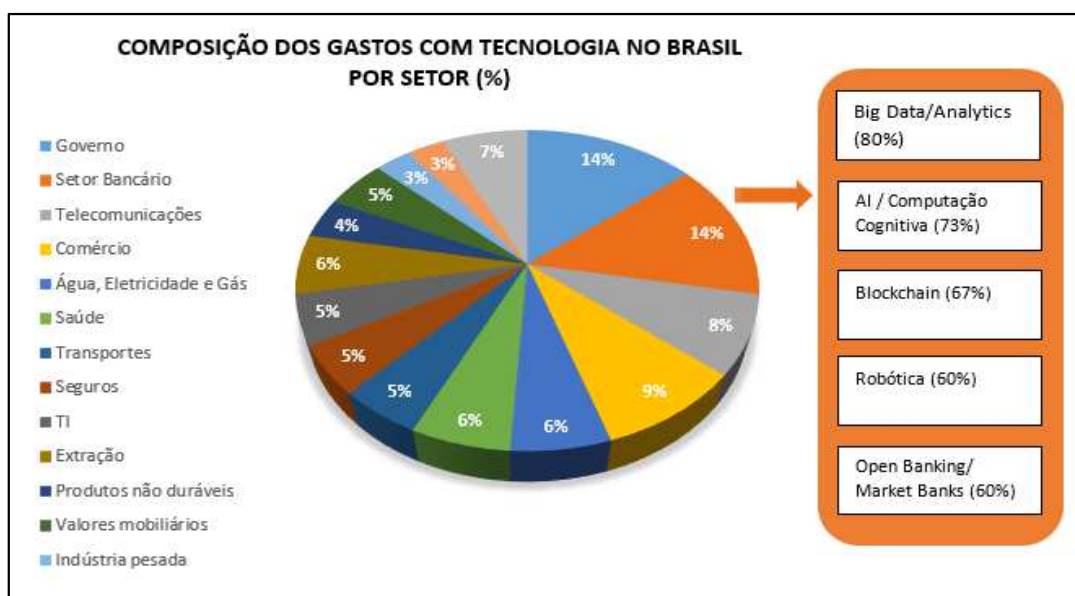


Figura 3: Composição dos gastos com tecnologia por setor no Brasil e principais áreas de investimentos do setor bancário. Fonte: Pesquisa FEBRABAN de Tecnologia Bancária 2019.

Tais investimentos visam basicamente garantir que o consumidor tenha uma melhor experiência com soluções e produtos bancários em relação a segurança e capacidade de processamento. É importante notar que os esforços de *marketing* e vendas estão mais intensamente direcionados ao *mobile banking* que ao *internet banking*, o que demonstra o potencial que as instituições observam de fortalecimento do relacionamento com os clientes por meio de celulares.

Essa revolução provocada pela digitalização das soluções bancárias promoveu ainda uma mudança completa da figura dos gerentes e dos atendentes de agência, que se tornou mais consultiva e direcionada a serviços mais complexos. Isso impactou as aplicações de recursos em treinamentos com tecnologia para os profissionais. Houve também um aumento significativo das interações com clientes via *chatbots*, robôs que utilizam linguagem natural e se aperfeiçoam quanto mais são utilizados.

A expansão das recém-chegadas plataformas de investimentos atraiu mais de 10% dos 2,98 trilhões de reais investidos por brasileiros em fundos mútuos, ações e títulos segundo a Exame. Este movimento tem incomodado bancos tradicionais como Itaú e Bradesco, que se

mantiveram por muito tempo como única alternativa para investidores de varejo interessados em centralizar conta bancária e aplicações.

Tal mudança tem sido benéfica para o mercado, no sentido de estabelecer uma pressão competitiva perceptível nas receitas de processamento de cartões, taxas de transferência de fundos entre outras tarifas. Após muito tempo engessados, os bancos tradicionais entraram em uma corrida a fim de demonstrar que podem ser ágeis, digitais e que estão abertos às mudanças recentes do ecossistema financeiro brasileiro. Isso se torna um desafio considerável levando-se em conta que tais projetos para se igualar às *fintechs* envolvem uma queda da margem de lucro dessas instituições que precisa ser sustentada simultaneamente ao funcionamento das agências.

Atitudes dos grandes bancos visando se adaptar a uma nova realidade do consumidor brasileiro incluem a aproximação e até mesmo aquisição de startups para se desenvolver, bem como o oferecimento de múltiplos serviços via aplicativos, priorizando o canal digital para transações com movimentações financeiras.

Um detalhe importante acerca das adaptações no mercado financeiro brasileiro é que as próprias atuais plataformas de investimento brasileiras disponíveis no mercado tiveram que se reinventar ao longo dos anos para explorar uma nova oportunidade. Poucas como a Órama, a Rico e a Clear, sendo que essas duas últimas foram adquiridas pela XP Investimentos, já nasceram digitais. Plataformas como a Genial, Guide, Modalmais e Easynvest eram apenas antigas corretoras que perceberam um oceano azul em expansão e a necessidade de uma transformação para prosperarem, segundo uma matéria do Valor Investe.

Aos poucos, as gestoras dos fundos de investimentos enxergaram nessas plataformas uma importante ferramenta de captação de recursos, o que possibilitou uma democratização do mercado de fundos para o pequeno investidor. Com as evidências de que a poupança não seria um investimento vantajoso e de que existiam alternativas seguras e muito mais interessantes na renda fixa acessíveis em tais plataformas, o investidor foi se “educando” e começou a explorar cada vez mais as opções disponíveis nos “*shopping centers*” de investimentos.

Há pouco tempo era difícil encontrar alternativas com aplicação mínima inferior a R\$1.000. Atualmente, com R\$ 50 é possível comprar títulos públicos. Algumas plataformas zeraram as taxas para fundos que aplicam somente no Tesouro. Outras opções como CDBs, LCAs, LCIs começaram a se tornar interessantes sob a proteção do Fundo Garantidor de Crédito. Além disso, incentivos fiscais em CRIs, CRAs e debêntures incentivadas garantiram maior diversificação do mercado de varejo.

Com a queda dos juros, a busca por maiores retornos fez com que os investidores comesçassem a olhar mais para fundos imobiliários negociados em bolsa, fundos multimercado

e de ações, além das próprias ações, enquanto a taxa básica da economia caía para um dígito e incentivava ainda mais essa busca por mais retorno, mesmo exigindo um maior risco.

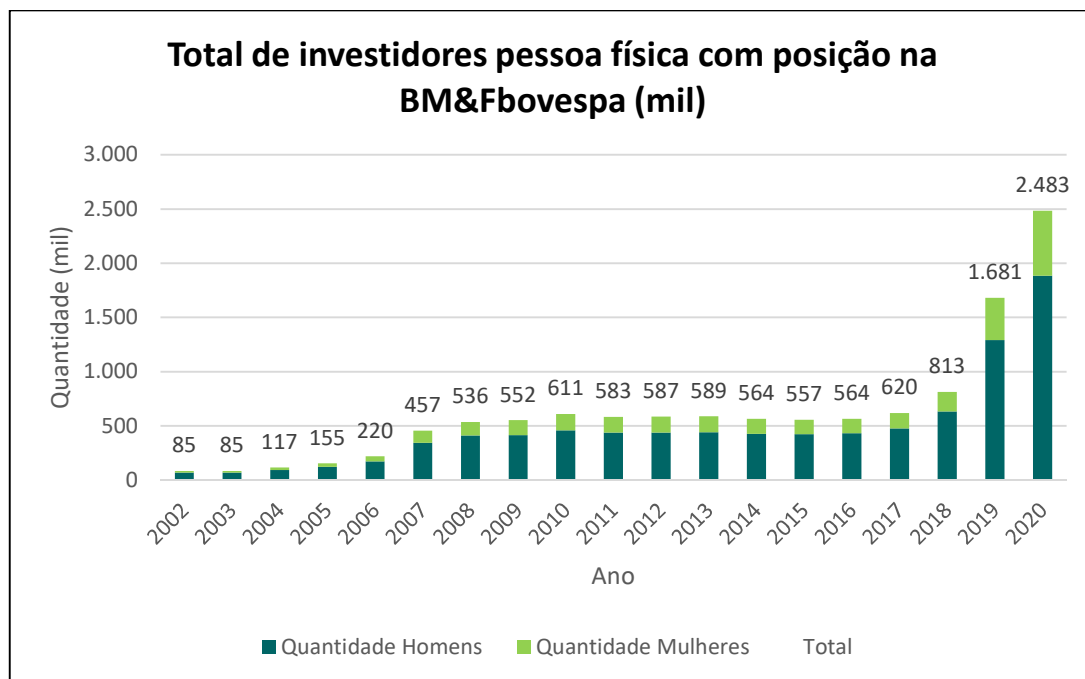


Figura 4: Evolução da quantidade de pessoas físicas na BM&FBovespa (valores atualizados até maio/20).

Para se diferenciarem, as plataformas apostaram por muito tempo na redução de taxas, mas aparentemente a fase em que Tesouro Direto com taxa de custódia zero, grande variabilidade de produtos e custos mais baixos não é mais suficiente para o consumidor moderno. Algumas prateleiras possuem mais opções do que outras, mas é viável comprar títulos públicos e privados bem como fundos e ações em praticamente todas elas.

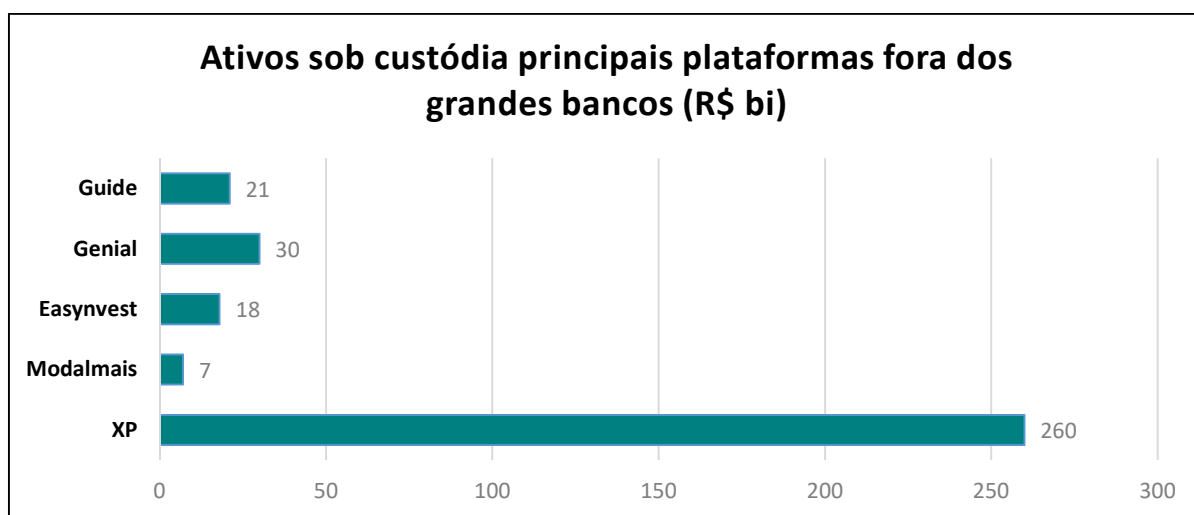


Figura 5: Volume de *Assets under Custody* por plataforma em bilhões de reais. Órama e BTG Digital não divulgaram seus valores. Fonte: Valor Investe (dados de julho/2019).

						
1,4 milhões de contas • Mais de 650 produtos, 400 fundos de investimento	1,08 milhões de contas • 650 produtos disponíveis, sendo 280 fundos	900 mil contas • Mais de 400 produtos, sendo 140 fundos	160 mil contas • 345 fundos e 271 produtos de renda fixa	82 mil contas • 400 produtos entre fundos e renda fixa	não divulgado • 400 fundos, 47 emissores bancários	não divulgado • 337 fundos e 25 emissores de renda fixa bacária

Figura 6: Portfolio produtos de prateleira *ex-home broker* das plataformas fora dos grandes bancos. Dados de julho de 2019. Fonte: Valor Investe.

Comparando as opções disponíveis é possível observar a variedade de produtos e de serviços oferecidos por essas plataformas. A grande tendência do momento é a mistura das características de uma corretora com as de um banco tradicional, o que tem sido considerado um diferencial dos chamados bancos digitais, de modo que clientes possuem acesso a diversos produtos de investimentos, incluindo tesouro direto, bolsa, fundos e renda fixa provenientes de outras casas enquanto, simultaneamente, têm a sua disposição uma conta corrente digital, a possibilidade de realizar pagamentos e transferências bem como pagar seus gastos do dia-a-dia com cartões de crédito e de débito.

No entanto, o mar de opções disponíveis tornou o portfólio da prateleira além de mais completo, mais complexo. Para tentar auxiliar o cliente a escolher os melhores produtos de acordo com seus objetivos as plataformas começaram a investir cada vez mais em consultoria e informação para os consumidores. Isso é um diferencial principalmente para os clientes do varejo, que geralmente não possuem nenhuma assessoria personalizada. Uma alternativa nesse sentido seria plugar agentes autônomos e consultores na plataforma a fim de direcionar os melhores produtos para um determinado indivíduo de acordo com sua situação no momento. Atualmente, XP, BTG Digital, Guide e Modalmais apresentam essa iniciativa.

Outra forma de atrair clientes amplamente utilizada no momento é a educação financeira por intermédio das redes sociais e dos veículos de mídia especializados. O consumidor ávido por educação financeira em um país onde a formação financeira não recebe o devido investimento desde a infância busca o apoio dos influenciadores digitais para se informar e melhorar a qualidade de seus investimentos. Tais influenciadores utilizam os seus respectivos canais do Youtube uma forma de divulgar conhecimento para aqueles que desejam aprender

mais sobre o mundo dos investimentos. A Modalmais firmou parceria com a Nathália Arcuri do canal “Me Poupe!”, mais focado para os consumidores da Renda Fixa e distribui cursos do analista Leandro Martins para aqueles que desejam se aprofundar no mundo da bolsa de valores. Simultaneamente a Rico da XP tem o influenciador Thiago Nigro do canal “O Primo Rico” e o BTG possui o apoio do seu consultor financeiro “Gustavo Cerbasi” e “André Bona” do canal “Blog de Valor”.

Segundo o Valor Investe, contudo, possuir um banco digital com serviços de banco e corretora além de apostar na educação financeira dos clientes pode não ser um diferencial daqui a um tempo, levando em conta o desenvolvimento rápido desse mercado ao longo dos últimos anos. Talvez o desenvolvimento de algoritmos para personalizar carteiras seja a nova tendência ou talvez a relação com os consultores financeiros seja o diferencial para as novas tendências do mercado.

Atualmente já existem casas que trabalham com algoritmos para personalizar os investimentos de seus clientes. Segundo a descrição no site do BTG Digital os chamados *robôs advisors* utilizam um algoritmo para criar carteiras personalizadas para seus investidores. Para isso eles usam informações fornecidas pelo usuário, tais como perfil de risco desejado, vencimentos e rendimentos. Uma das vantagens desse serviço está na taxa de administração mais baixa ou nula, já que não é necessária a participação de um profissional de *wealth management*. O robô basicamente utiliza as informações fornecidas pelo usuário de perfil de investimento e prazos para recomendar a melhor carteira possível levando em consideração o efeito diversificação. A ideia é possuir algum volume de recursos para diversificar os ativos maximizando ganhos e minimizando perdas em qualquer cenário no médio e longo prazo.

Outra alternativa para clientes com um perfil menos conservador é o chamado *robô trader*, o qual opera mais especificamente no mercado de renda variável com foco total no curto prazo. Este robô demanda um pouco mais de atenção do cliente, pois executa a estratégia definida inicialmente, mas exige um acompanhamento para que sejam feitos ajustes de rota. Tais robôs atuam por meio de análise grafista, não fundamentalista, verificando as oscilações do papel e apontando tendências segundo o histórico.

Conforme foi possível observar neste capítulo, portanto, o setor de serviços financeiros no Brasil e no mundo está passando por um ponto de inflexão. Com as visíveis mudanças tecnológicas, regulatórias e de mercado no setor, as instituições financeiras estão sendo obrigadas a mudar seus modelos de negócio e operacionais. As relações entre bancos e *fintechs*

estão evoluindo rapidamente, ora como concorrentes, ora como parceiros. Tais relações de um certo modo estão colaborando para uma abordagem mais inovadora para o setor.

No cerne desta inovação, encontra-se o investimento em inovação digital no longo prazo, para atingir um novo modelo operacional, aumentando a confiança e a eficiência dos processos bancários, gerando valor para as instituições, melhorando a gestão de dados, centralizando a infraestrutura e implementando a inteligência artificial com foco em melhorar ainda mais a experiência dos clientes. Com isso, deixa-se de lado a simples guerra de preços e coloca-se em evidência outra forma de se destacar perante os investidores, gerando lealdade, a partir do momento em que se deixa de utilizar a automatização apenas para rotinas e passa-se a empregar-la a fim de desenvolver uma melhor relação com os usuários dos serviços bancários.

Simultaneamente, novos desafios surgirão, tais como a necessidade de repensar aspectos éticos associados à aplicação de algoritmos e inteligência artificial.

2.2. VARIEDADE DAS BASES DE DADOS DE UMA PLATAFORMA DIGITAL

Conforme descrito no capítulo anterior, a medida que as plataformas digitais se desenvolveram no Brasil, seu portfólio de opções tornou-se além de mais completo, mais complexo, com uma base de ativos financeiros variada para atender a diversos perfis de investidor.

Mas afinal, em que consiste um ativo financeiro? Segundo Bodie et al (2014) seriam os meios pelos quais os indivíduos de uma economia reivindicam direitos sobre os ativos reais da economia, tais como terrenos, prédios, equipamentos, conhecimentos para produzir bens e serviços etc. Todo ativo financeiro é uma reivindicação da renda gerada por ativos reais. Por exemplo, se um indivíduo não tem renda suficiente para comprar uma fábrica de automóveis, que seria um ativo real, ele tem a possibilidade de compartilhar da renda derivada de sua produção adquirindo ações da mesma. Enquanto ativos reais geram a renda, os ativos financeiros definem a alocação desta renda entre investidores.

O termo ativo financeiro concentra um significado substancialmente amplo. Isso porque no mercado financeiro de investimentos há inúmeros ativos distintos, tais como ações, títulos, títulos da dívida externa, opções, *real estates* e contratos futuros disponíveis para negociações, de modo que é preciso notar que diferentes ativos podem conservar diferentes características. Como nem todos os ativos seriam considerados apropriados para um dado investidor, seria

desejável que tais ativos fossem estratificados em diferentes classes baseando-se em determinadas características.

Seleção de um portfólio de investimentos é um campo de estudo que começou com o modelo de Markowitz no qual retorno é quantificado como média e risco como variância. Segundo ele, a seleção de ativos para um portfólio envolve obter proporções ótimas para estruturar uma carteira que reflita as preferências do investidor.

Na maioria dos estudos, risco e retorno são considerados como dois fatores fundamentais que guiam as escolhas dos investidores. No entanto, nem toda a informação relevante para seleção de um portfólio pode ser capturada com apenas essas duas características. Os modelos de seleção de portfólio multicritério estão ganhando interesse de pesquisadores num passado recente. Desde Markowitz tem-se buscado de várias maneiras incorporar as preferências do investidor ao modelo de escolha do portfólio.

Normalmente, segundo Bodie et Al. (2014) distinguem-se três tipos de ativos financeiros: títulos de dívida, ações e derivativos.

Títulos de renda fixa ou de dívida são aqueles que prometem um fluxo fixo de renda, ou um fluxo determinado a partir de uma fórmula específica. As obrigações podem estar associadas a taxas fixas ou flutuantes. Normalmente, o desempenho desses títulos é o que menos está associado à condição financeira do emissor, mas são oferecidos com uma enorme variedade de vencimentos e fluxos de pagamento. O mercado monetário refere-se a títulos de renda fixa de curtíssimo prazo (letras do tesouro e CDBs) enquanto o mercado de capitais de renda fixa abrange títulos de longo prazo, como obrigações federais, estaduais, corporativas, que variam entre muito seguras (como os títulos do tesouro) e arriscadas (com alta rentabilidade, refletindo alto risco).

As ações, por sua vez, representam uma participação direta na propriedade da corporação. Nesse caso, não é prometido nenhum tipo de pagamento aos proprietários da ação. Os investidores recebem dividendos das empresas e possuem participação proporcional nos seus ativos reais. O desempenho desses investimentos, portanto, encontra-se diretamente associado com o sucesso da empresa e seus ativos reais. Por isso, os investimentos em ações tendem a ser mais arriscados que os investimentos em títulos de dívida.

Por fim os derivativos, tais como opções, contratos de balcão e contratos futuros, oferecem retornos que são determinados pelo valor de outros ativos, chamados ativos objetos. Desse modo, seu preço reflete a variação das expectativas acerca um ativo base em uma determinada data futura considerando as informações que o mercado possui em uma data atual.

Tais contratos são muito utilizados para proteger o capital com exposição em um fator de risco específico no futuro contra as variações do mesmo.

As figuras a seguir resumem as ramificações do mercado financeiro e as principais variáveis que as caracterizam.

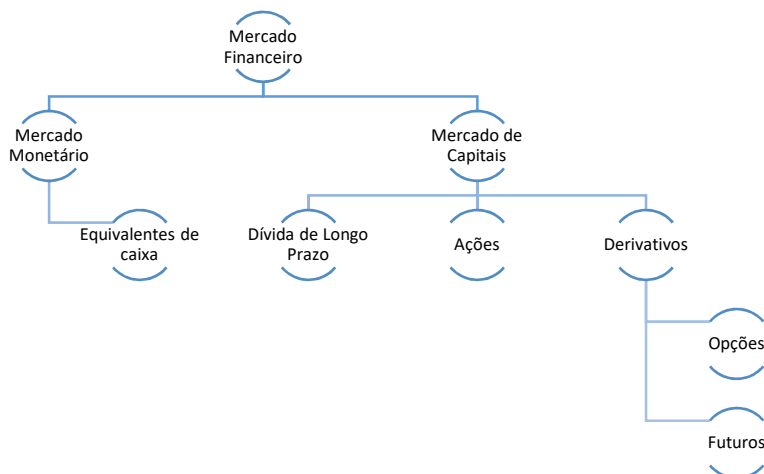


Figura 7: Clusters de ativos no mercado financeiro. Fonte: A autora

Além desses ativos, existem ainda os fundos de investimento, nos quais um gestor compõe uma carteira com ativos financeiros ou ainda outros fundos a partir da captação de recursos de investidores que confiam seu capital à gestão visando maximizar retorno e minimizar riscos de acordo com um determinado perfil de risco, mediante o pagamento de uma taxa pelo serviço.

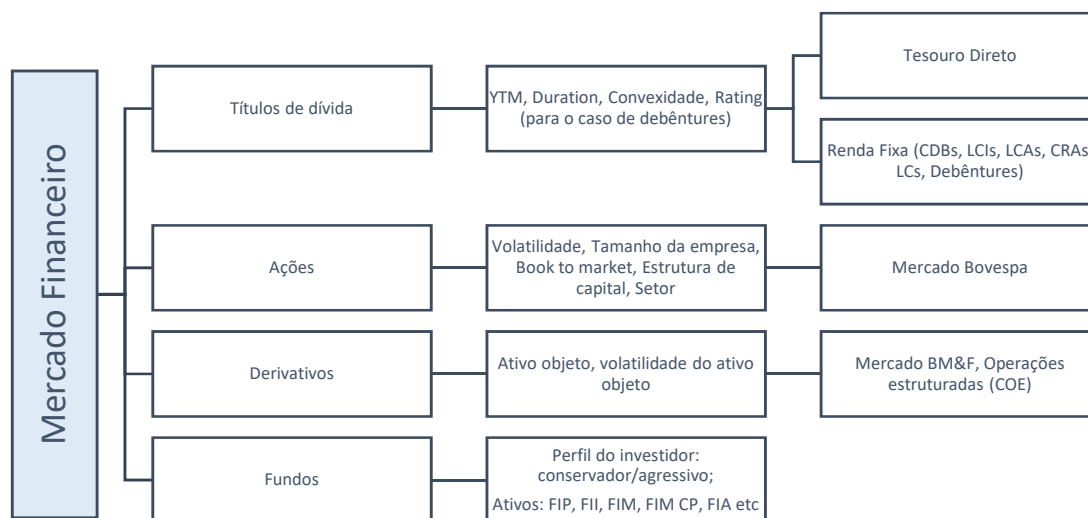


Figura 8: Caracterizando um ativo específico no mercado financeiro. Fonte: A autora.

Portanto, neste capítulo foi possível observar a evolução das plataformas digitais no Brasil nos últimos anos, as atitudes que os grandes bancos estão tomando para acompanhar a mudança de comportamento dos consumidores - que se reflete na evolução da participação de pessoas físicas na bolsa por exemplo - e até mesmo as inovações regulatórias que incentivam tal movimento de abertura de contas em plataformas digitais. Além disso, foi possível perceber o quão complexo está se tornando o portfólio de produtos disponíveis em tais canais de investimento, o que gera a necessidade de uma educação financeira mesmo para o investidor que não possui um atendimento especializado e de um filtro que possibilite melhorar a experiência do investidor na plataforma, destacando em um mar de produtos aquele que é mais interessante de acordo com seu perfil.

No capítulo a seguir serão abordados os sistemas de recomendação, já utilizados para resolver um problema semelhante ao descrito acima para produtos diversos no *e-commerce* e nas plataformas de *streaming*, como livros e filmes por exemplo. Será abordada a sua origem a partir dos sistemas de aprendizagem de máquina, suas diversas abordagens, métodos, medidas de desempenho e limitações com o objetivo de demonstrar como funcionam.

3. SISTEMAS DE RECOMENDAÇÃO: ORIGEM, ABORDAGENS E MÉTODOS

“Se eu tenho 3 milhões de clientes na web, devo ter 3 milhões de lojas na web.” (Jeff Bezos, CEO da Amazon.com)

Na era digital, em que uma grande quantidade de dados encontra-se disponível na web para ser consultada a todo momento, as opções dos usuários tornam-se cada vez mais vastas, como foi possível ver no capítulo anterior no caso da evolução das bases de plataformas digitais de investimentos. Desse modo, a necessidade de filtrar dados mostra-se constantemente necessária para apresentar ao consumidor o conteúdo que efetivamente lhe é útil.

Para isso, surgiram os chamados sistemas de recomendação, uma subparte do campo de *data mining* utilizada para dar assistência a empresas que desejam uma vantagem competitiva no sentido de incrementar *cross-selling* e promover a lealdade do consumidor preenchendo suas necessidades a partir da apresentação de itens que satisfaçam o usuário. A partir de um filtro de informações, o sistema consegue priorizar a informação que mais interessa ao usuário e descartar aquela que para ele é desnecessária. Desse modo, uma aplicação é capaz de fidelizar o cliente, considerando que um concorrente demoraria um tempo considerável para adquirir uma base de conhecimento extensa o suficiente e comparável à da aplicação original.

O termo *data mining*, segundo Schafer et al. (2001), é utilizado para descrever a coleção de técnicas de análise usadas para inferir regras ou construir modelos a partir de grandes bases de dados. Os sistemas de recomendação, como os sistemas de aprendizado de máquina em geral possuem duas fases: uma na qual “aprendem” sobre o comportamento dos clientes e outra na qual aplicam o conhecimento em tempo real.

No cotidiano é possível observar uma série de aplicações práticas, desde as mais simples tais como a recomendação de palavras a um usuário que digita no Microsoft Word ou em seu aparelho *smartphone*, até as mais complexas que vêm sendo utilizadas para recomendar produtos aos consumidores que realizam compras pela internet.

A recomendação de itens aos usuários se vê presente em *shoppings centers online* como a Amazon, plataformas de streaming de filmes e música como o Netflix e Pandora, sites de recomendação de filmes, tais como Movielens e sites de busca como o Google. Cada uma dessas empresas utiliza abordagens diferentes para melhor captar as características dos consumidores de informação e, conseqüentemente, melhor prever o que lhes interessa em um mar de dados. A origem de tais algoritmos e suas abordagens serão melhor descritas nos subcapítulos a seguir.

3.1. A ORIGEM: SISTEMAS DE APRENDIZADO DE MÁQUINA

“*Machine Learning* é a ciência (e arte) de programar computadores de modo que eles possam aprender com os dados.” (Aurélien Géron)

Conforme explicado anteriormente, os sistemas de recomendação são uma subparte do campo de aprendizado de máquina (*machine learning*). Tal campo, apesar de a princípio evocar a sensação de algo futurista, esteve presente no cotidiano por décadas, desde o reconhecimento ótico de caracteres.

Segundo Géron (2017), a primeira aplicação que teve grande visibilidade de *machine learning* foi o filtro de spam nos anos 90, que aprendia a classificar *e-mails* como *spams* a partir de exemplos já classificados pelos usuários entre *spam* e *ham* (os chamados não-*spams*). Os exemplos usados nesse “aprendizado”, seriam os chamados dados de treino, uma amostra que seria utilizada para aprender as regras do que diferencia um e-mail *spam* dos outros. A partir disso, o sistema utiliza o conhecimento obtido para classificar novos e-mails. Assim, será possível medir sua performance na classificação, a chamada acurácia do modelo.

Assim como o filtro de spam, funcionam os sistemas em geral de aprendizado de máquina e também os sistemas de recomendação.

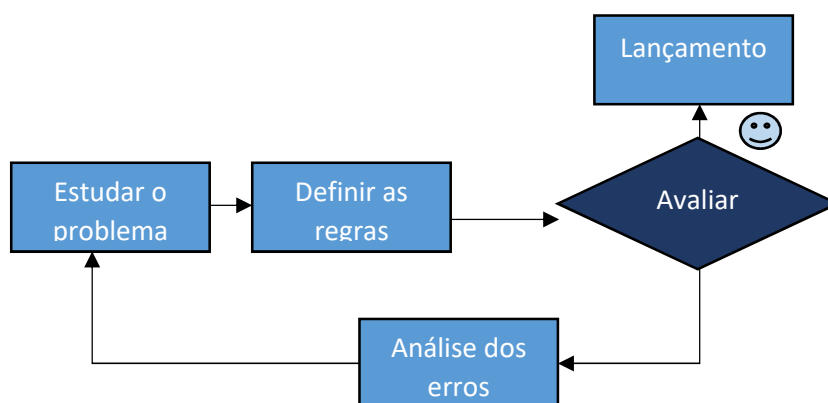


Figura 8: Fluxograma abordagem tradicional. Fonte: Géron.

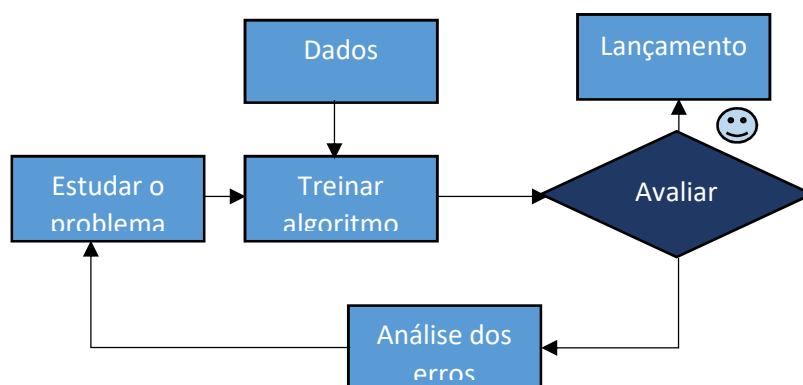


Figura 9: Fluxograma abordagem *machine learning*. Fonte: Géron.

Existem vários tipos de sistemas de aprendizado de máquina, de modo que eles podem ser classificados quanto a alguns aspectos, tais como:

- Supervisão humana: nos algoritmos supervisionados os dados de treino contêm as características desejadas, enquanto nos não-supervisionados o sistema tenta aprender sem um professor;
- Capacidade incremental de aprendizado: no sistema de lote, o algoritmo não possui capacidade incremental de aprendizado, ou seja, ele precisa ser treinado com todos os dados disponíveis, enquanto no sistema online, é possível treinar o modelo em mini-lotes;
- Capacidade de detectar padrões nos dados de treino e generalizar: no sistema baseado em exemplos, simplesmente compara-se novos exemplos com os existentes para fazer projeções enquanto no sistema baseado em modelo, detecta-se um padrão nos dados de treino para criar um modelo e realizar novas projeções.

A aplicação de algoritmos de *machine learning* em grandes quantidades de dados pode ajudar a descobrir padrões antes não tão aparentes, o que é chamado de *data mining*, campo ao qual pertencem os sistemas de recomendação, conforme já abordado na introdução do capítulo 3. No entanto, dentro do subcampo de sistemas de recomendação existem diferentes abordagens possíveis, conforme será visto no subcapítulo a seguir.

3.2. ABORDAGENS DOS SISTEMAS DE RECOMENDAÇÃO

Segundo Shah et al. (2016), existem basicamente duas abordagens possíveis em sistemas de recomendação: a não-personalizada e a personalizada. A não-personalizada é aquela na qual o sistema realiza a mesma recomendação para todos os usuários. Tal abordagem não leva em consideração, portanto, a individualidade do consumidor, sendo utilizada, por exemplo, pelo Youtube para indicar vídeos mais populares entre os usuários, ou em outros sites que recomendam produtos de acordo com o nível de popularidade destes. Como esta linha não visa interpretar o perfil do consumidor e extrair o que ele gostaria de encontrar, sua recomendação pode ou não agradar ao usuário.

A personalizada, por outro lado, considera os interesses do usuário, recomendando itens particulares para uma comunidade de indivíduos. Dentro desta abordagem, há ainda três caminhos possíveis: o baseado em conteúdo, o baseado em filtro colaborativo e o híbrido.

3.2.1. FILTRO BASEADO EM CONTEÚDO

O filtro baseado em conteúdo, também conhecido como filtro cognitivo faz recomendações considerando as características do usuário e dos itens lhe interessam. Esse perfil é criado no momento que uma conta é criada e começa a ser utilizada. A medida que o indivíduo interage com o sistema, seu perfil começa a ficar mais forte e definido.

A ideia por trás dessa abordagem é: se uma pessoa gostou deste item no passado, ela provavelmente vai gostar deste item no presente. Para construir seu perfil, é preciso coletar informações sobre suas preferências.

Há vários algoritmos disponíveis nesse tipo de filtro, muito usados principalmente na recomendação de itens baseados em texto. As técnicas mais utilizadas são TF-IDF³ e classificadores ingênuos de Bayes⁴.

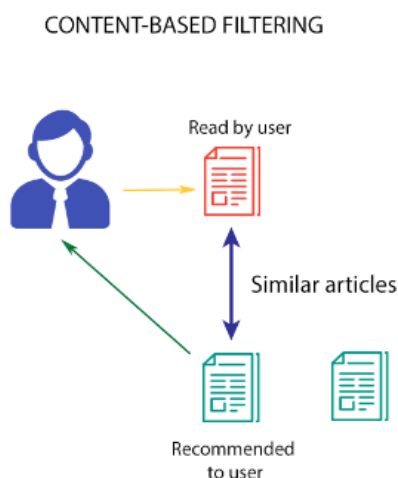


Figura 9: Ilustração de filtro baseado em conteúdo. Fonte: Google.

Segundo Shah et al. o TF-IDF considera que N é número total de documentos que podem ser recomendados para o usuário, enquanto a palavra-chave k_i aparece em n_i deles. Além disso, f_{ij} é o número de vezes que a palavra-chave aparece em um documento específico, o que seria a definição mais simples de frequência da palavra i no documento j . Existem, entretanto, formas de ajustar tal frequência, de normalizá-la pelo tamanho do documento j ou pela frequência do

³ TF-IDF (*term frequency-inverse document frequency*)

⁴ Naïve Bayes

termo mais frequente z no documento. Portanto, a frequência de um termo (TF_{ij}) pode ser definida conforme abaixo:

$$TF_{ij} = \frac{f_{i,j}}{\max_z f_{z,j}}$$

Em que, $f_{i,j}$ é o número de vezes que o termo i aparece no documento j e $\max_z f_{z,j}$ representa a frequência do termo mais frequente no documento j .

No entanto, essa definição dá importância igual a todos os termos, o que é ruim quando se considera que há termos que podem ser frequentes, porém insignificantes. Assim, torna-se necessário dar um peso menor aos termos frequentes, mas menos importantes, e um peso maior aos mais raros, a partir da frequência inversa dos documentos, definida abaixo:

$$IDF_i = \frac{N}{n_i}$$

Em que n_i representa o número de documentos no qual o termo i aparece e N representa o número total de documentos disponíveis para a recomendação.

Portanto, o TF-IDF ($w_{i,j}$) para uma palavra-chave é dado por:

$$w_{i,j} = TF_{i,j} \times IDF_i$$

Por fim, o conteúdo de um determinado documento d_j com k termos poderia ser definido como:

$$d_j = (w_{1,j}, \dots, w_{k,j})$$

Outras formas de fazer o filtro baseado em conteúdo incluem técnicas de *machine learning* tais como formação de *clusters*, classificadores ingênuos de Bayes e redes neurais. O classificador Bayesiano consiste em utilizar o teorema de Bayes assumindo independência entre os eventos para estimar a probabilidade de um conteúdo ser aprovado ou não pelo usuário. A lógica consiste em descobrir qual é a chance de um determinado conteúdo ser classificado como relevante dado um conjunto de palavras-chave, o que seria equivalente pelo teorema de Bayes ao produto da probabilidade de um conteúdo ser relevante pela probabilidade de ele possuir uma determinada característica, dado que é relevante.

Portanto a probabilidade de uma página na *web* ser considerada relevante, por exemplo, seria traduzida como a probabilidade de ser classificada como relevante (C_i) dado que possui a lista de características (k_{nj}), assumindo que tais atributos são independentes. Pelo teorema de Bayes, isso seria equivalente ao produto entre a probabilidade de ser considerado relevante $P(C_i)$ e o produtório das probabilidades individuais de possuir determinada característica x dado que é classificado como relevante $P(k_{xj}|C_i)$.

As probabilidades $P(k_{x,j}|C_i)$ e $P(C_i)$ podem ser estimadas a partir dos dados de treino do modelo.

$$P(C_i | k_{1j} \& k_{2j} \& \dots k_{nj}) = P(C_i) \prod_x^n P(k_{x,j}|C_i)$$

O filtro baseado em conteúdo, no entanto, possui algumas limitações, tais como a dificuldade em criar um perfil assim que um novo usuário entra no sistema por falta de informações acerca de suas preferências, o que pode resultar em uma recomendação pobre.

Além disso, o sistema cria uma espécie de vício em itens que o usuário classificou no passado, de modo que não recomenda itens diferentes daqueles que o indivíduo já conhece. Esse excesso de especialização torna o escopo de recomendações bastante limitado, sem sugestões novas ao usuário.

3.2.2. FILTRO COLABORATIVO

O filtro colaborativo leva em consideração a proximidade entre usuários para gerar recomendações, de modo que indica itens com base nas classificações feitas por um conjunto de indivíduos, explícita ou implicitamente, que expressam seus feedbacks em relação aos produtos consumidos. Tal filtro explora semelhanças no comportamento dos ratings marcados por um conjunto de pessoas a fim de determinar como recomendar um item.

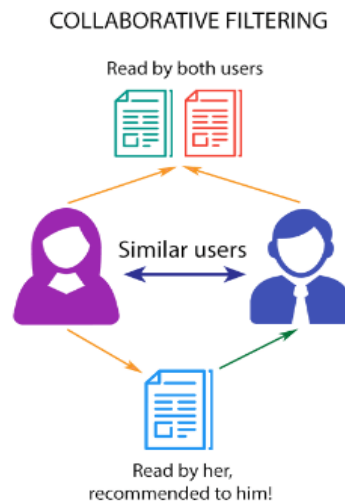


Figura 10: Ilustração de filtro colaborativo. Fonte: Google.

Tal abordagem é utilizada pela Amazon.com e pela Movielens por exemplo, e pode ser baseada em vizinhança ou modelo, de acordo com Shah et al. (2016). Tais sub-abordagens serão descritas nos sub-itens abaixo.

3.2.2.1. FILTRO COLABORATIVO BASEADO EM VIZINHANÇA

O filtro colaborativo baseado em vizinhança considera todo o conjunto de dados de todos os usuários que classificaram um determinado produto no passado. Existe uma matriz que relaciona produtos, usuários e ratings, de modo que, para alguns usuários, os ratings encontram-se vazios e precisam ser preenchidos.

O rating desconhecido de um item s para um usuário c é geralmente computado como um agregado dos N mais semelhantes usuários para o mesmo item s . Para agregar os ratings normalmente são usadas funções como a média dos ratings de usuários similares ou a média ponderada dos ratings de usuários semelhantes. É possível ainda focar no item ou no usuário, de modo que em ambos os casos se utiliza a regra do vizinho mais próximo K (KNN⁵).

Portanto, é estabelecido um usuário-alvo X , é escolhido um número K de pessoas semelhantes que serão usadas como referência e é medida a similaridade dos K usuários com X a partir de técnicas tais como distância Euclidiana, similaridade de cossenos entre outras. Por fim, é escolhida uma função de peso e é encontrado um único número.

Quando se foca no usuário, inicialmente é associado um peso para os usuários do sistema de acordo com o nível de similaridade que estes possuem com o usuário analisado. Depois, é selecionado um número k de usuários próximos, ou seja, usuários que possuem preferências parecidas com o indivíduo que se deseja analisar. Finalmente, é realizada a predição para o usuário em questão com base na função de peso e nos ratings existentes dos k usuários similares.

O método mais utilizado, segundo Shah et al (2016), para medir a similaridade entre usuários é o coeficiente de correlação de Pearson, que segue a seguinte equação:

$$w_{a,u} = \frac{\sum_{i \in I} (r_{a,i} - \bar{r}_a)(r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i \in I} (r_{a,i} - \bar{r}_a)^2 \sum_{i \in I} (r_{u,i} - \bar{r}_u)^2}}$$

Onde: $w_{a,u}$ representa a similaridade entre um usuário qualquer u e o usuário analisado a , $r_{a,i}$ o rating que o usuário a utilizou para classificar o item i , $r_{u,i}$ o rating que o usuário u

⁵ *K-nearest neighbor rule*

utilizou para classificar o item i , \bar{r}_a o rating médio de classificação de itens utilizado por a e \bar{r}_u o rating médio de classificação de itens utilizado por u . Além disso, I representa o conjunto de todos os itens que foram avaliados por ambos os usuários.

Após o cálculo dos pesos dos indivíduos do sistema, é selecionado aleatoriamente um número k de usuários para compor a análise e calcular a predição com base na média ponderada dos desvios em relação à média dos usuários próximos, conforme descrito na fórmula a seguir.

$$P_{a,i} = \bar{r}_a + \frac{\sum_{u \in K} (r_{u,i} - \bar{r}_u) \times w_{a,u}}{\sum_{u \in K} w_{a,u}}$$

Onde: $P_{a,i}$ se trata da predição para o usuário a em relação ao item i , $w_{a,u}$ é a similaridade calculada anteriormente e K é o número de usuários similares considerados na análise.

Conforme a quantidade de pessoas no sistema aumenta, a complexidade em encontrar pessoas semelhantes também aumenta. Por isso, após propor a abordagem colaborativa com foco nos usuários nos anos 90, a Universidade de Minnesota propôs em 2001 um modelo direcionado a encontrar itens semelhantes, segundo Shah et al (2016).

O processo é semelhante ao descrito anteriormente, mas focado em itens ao invés de indivíduos. Do mesmo modo calcula-se um coeficiente de correlação, só que entre itens, seleciona-se um número K de itens mais próximos e realiza-se a predição.

$$w_{i,j} = \frac{\sum_{u \in U} (r_{u,i} - \bar{r}_i)(r_{u,j} - \bar{r}_j)}{\sqrt{\sum_{u \in U} (r_{u,i} - \bar{r}_i)^2 \sum_{u \in U} (r_{u,j} - \bar{r}_j)^2}}$$

$$P_{a,i} = \frac{\sum_{j \in K} r_{a,j} w_{i,j}}{\sum_{j \in K} |w_{i,j}|}$$

Onde U consiste no conjunto de todos os usuários que classificaram os itens i e j , $r_{u,i}$ é o rating que o usuário u utilizou para classificar o item i e \bar{r}_i é o rating médio entre todos os usuários que classificaram o item i . O K corresponde à vizinhança composta pelos itens classificados pelo usuário a que são mais similares ao item i , $w_{i,j}$ consiste na similaridade entre o item analisado i e um item qualquer j e $P_{a,i}$ a previsão do rating que o usuário a daria para o item i .

Outro método também utilizado para medir a similaridade no filtro colaborativo segundo Sarwar et al (2000) é o valor do cosseno do ângulo entre vetores que representam usuários, a chamada similaridade de cossenos citada anteriormente. Nesse caso, o usuário do

sistema é visto como um vetor de dimensão N , no qual N representa o número de itens disponíveis na base. Os componentes do vetor de um usuário x são positivos quando os itens foram classificados positivamente por este ou negativos quando foram classificados negativamente pelo mesmo usuário. A proximidade entre dois usuários (x e y) seria então o cosseno do ângulo entre os dois vetores que os representam (\vec{x} e \vec{y}), e pode ser escrita a partir da equação de produto escalar entre dois vetores conforme no exemplo abaixo:

$$\text{similaridade}(\vec{x}, \vec{y}) = \cos(\vec{x}, \vec{y}) = \frac{\vec{x} \cdot \vec{y}}{\|\vec{x}\| \cdot \|\vec{y}\|}$$

Tal método também pode ser utilizado para medir a similaridade entre itens, ao invés de usuários. A lógica é a mesma, com a diferença de que, nesse caso, cada vetor representa um item e possui dimensão M , em que M representa os usuários que compraram ou avaliaram o item.

3.2.2.2. FILTRO COLABORATIVO BASEADO EM MODELO

Diferentemente do filtro colaborativo baseado em vizinhança, que realiza recomendações com base em noções estatísticas de proximidade entre usuários e itens, o filtro colaborativo baseado em modelo estima parâmetros de modelos estatísticos para os ratings dados por usuários. Os chamados modelos de fator latente consideram que a similaridade entre usuários e itens é induzida por alguma estrutura de menor dimensão escondida nos dados.

Segundo o artigo de Shah et al. (2016) do Journal Internacional de Aplicações Computacionais, esses modelos são mais eficientes no sentido de que reduzem a dimensão dos dados, baseados nas relações de dependência entre eles. Com isso, o tempo de processamento torna-se menor e a resposta sobre a preferência do usuário torna-se mais rápida. Os modelos mais comuns desse tipo são de fatoração matricial.

Conforme mencionado no subcapítulo anterior, os algoritmos de recomendação geralmente dependem de dados que podem ser representados em uma matriz, na qual uma das dimensões representa os usuários e outra representa os itens disponíveis na base. O objetivo do sistema que recomendará itens aos usuários, portanto, é prever como usuários avaliariam itens que ainda não foram avaliados. Um exemplo de como essa matriz seria pode ser observado na tabela abaixo:

	Item_1	Item_2	Item_3	...	Item_n
User_1	2	3	??	...	5
User_2	??	4	3	...	??
User_m	1	??	5	...	4

Figura 11: Exemplo de tabela input

Cada célula com um número é uma classificação dada por algum usuário a algum item específico historicamente. Os que estão destacados com pontos de interrogação são os que necessitam ser descobertos. Em algumas outras literaturas, tal problema pode ser descrito como filtro colaborativo, preenchimento da matriz.

Segundo uma dissertação de mestrado que aborda modelos de fatoração matricial para recomendação de vídeos, Figueiredo Melo e Souza (2011), a Netflix, por exemplo, coleta esses dados solicitando que os clientes avaliem itens com estrelas de 1 a 5 e tais informações geram uma base esparsa, já que nem todos os usuários avaliam. Quando tais avaliações não se encontram disponíveis explicitamente, é possível concluir a preferência dos usuários a partir de seu histórico de compras ou buscas por exemplo, o que pode ajudar no preenchimento da matriz, o que seriam as classificações implícitas mencionadas no início do capítulo.

A técnica de fatoração matricial consiste em reduzir a dimensão dos dados transformando uma matriz em um produto de matrizes, geralmente duas.

$$\overline{r_{ui}} = q_i^T p_u$$

Onde r_{ui} consiste no rating com o qual um usuário u classifica um item i . Tal rating pode ser estimado ($\overline{r_{ui}}$) pelo produto entre dois vetores, o vetor q_i^T , que mede a extensão dos fatores que o item i possui (positiva ou negativa) e o vetor p_u que mede para um dado usuário u o nível de interesse que ele possui em itens que contêm os fatores correspondentes, também negativo ou positivo. Para aprender os vetores de fator que compõem um rating o sistema minimiza o erro quadrático regularizado dos ratings já conhecidos, conforme demonstrado abaixo.

$$\min_{q^*, p^*} \sum_{(u,i) \in K} (r_{u,i} - q_i^T p_u)^2 + \lambda (\|q_i\|^2 + \|p_u\|^2)$$

Nesse caso, λ é o chamado parâmetro de regularização e K consiste no grupo de pares usuário-item para os quais o rating (r_{ui}) é conhecido.

Usar gradiente descendente pode deixar o modelo mais preciso.

$$q_i \leftarrow q_i + \gamma(e_{ui}p_u - \lambda q_i)$$

$$p_u \leftarrow p_u + \gamma(e_{ui}q_i - \lambda p_u)$$

O parâmetro γ representa o passo do aprendizado e enquanto isso o erro pode ser minimizado como:

$$e_{ui} \stackrel{\text{def}}{=} r_{ui} - q_i^T p_u$$

Particularmente o pacote recosystem, no R, foi desenvolvido por um conjunto de tailandeses na Universidade Nacional de Taiwan, Chin, Zhuang et al. (2015).

Uma técnica popular para resolver tal problema é a fatoração da matriz. A ideia basicamente é aproximar a matriz $R_{m \times n}$ pelo produto de duas matrizes $P_{k \times m}$ e $Q_{k \times n}$

De modo que $R = P'Q$. É necessário “aprender” as matrizes P , que representa os usuários, e Q , que representa os itens, para preencher os ratings faltantes. Muitas vezes, os ratings podem ser simplesmente variáveis binárias $\{-1, 1\}$, representando um “gostou” ou um “não gostou” para determinado item. Com isso, as entradas da matriz principal R devem seguir tal padrão. A biblioteca para fatoração paralela de matriz (LIBMF – *Library for Parallel Matrix Factorization*) suporta valores reais (RVMF) ou binários (BMF). O modelo RVMF foca em gerar os ratings mais exatos possíveis enquanto o BMF foca em gerar corretamente os sinais que representam as preferências, mas ambos os problemas podem ser formulados como um problema de otimização não convexo:

$$\min_{P, Q} \sum_{(u,v) \in R} \left[f(p_u, q_v; r_{u,v}) + \mu_p \|p_u\|_1 + \mu_q \|q_v\|_1 + \frac{\lambda_p}{2} \|p_u\|_2^2 + \frac{\lambda_q}{2} \|q_v\|_2^2 \right]$$

(Chin, Zhuang, et al. 2015a; Chin, Zhuang, et al. 2015b)

Nesse caso, as variáveis u e v são as entradas, $r_{u,v}$ representa o rating, $f(\cdot)$ representa a função de perda e as gregas representam os parâmetros de regularização que evitam o *overfitting*⁶. O processo de resolver as matrizes P e Q chama-se treinamento do modelo e a seleção dos parâmetros de punição chama-se ajuste de parâmetro. O pacote recosystem do software R apresenta funções para, além de cumprir esses dois objetivos, exportar as bases.

Essa técnica de fatoração matricial ganhou mais visibilidade em 2006, a partir da proposta de Brandy Webb, mais conhecido pelo pseudônimo Simon Funk, durante a competição do prêmio Netflix. Sua proposta foi associada à clássica SVD (decomposição de valor), que técnica de fatoração matricial que decompunha a matriz de input em 3 matrizes, segundo Figueiredo Melo e Souza:

⁶ Fenômeno que ocorre quando o modelo se ajusta tão bem aos dados de treino, que não consegue ser generalizado para novos dados.

$$P_{n \times m} = A_{n \times m} \Sigma_{n \times m} B_{m \times m}^T$$

Onde $\Sigma_{n \times m}$ é a matriz diagonal com k entradas e as colunas das matrizes $A_{n \times m}$ e $B_{m \times m}^T$ são autovetores à esquerda e direita de P . Pegando apenas os ratings menores que k , os valores singulares mais significativos, é possível chegar ao produto dessas três matrizes, melhor aproximação de P . Simon Funk tentou corrigir os problemas da abordagem original, e acabou propondo um algoritmo com gradiente descendente utilizando valores conhecidos da matriz de avaliações e, ao invés de decompor 3 matrizes, focou em duas matrizes características, de modo que seu método, apesar de ser conhecido como SVD é mais parecido com a fatoração inicialmente descrita nesse subcapítulo. Isso porque o fato da matriz de avaliações ser esparsa e o fato da matriz completamente preenchida com avaliações ser muito grande impossibilita a utilização da técnica original.

Há outros métodos de filtro colaborativo tais como o Bayesiano, o probabilístico relacional, a regressão linear, o máximo de entropia e a decisão sequencial de problema e propósito que utiliza processos decisivos de Markov para gerar recomendações.

3.2.2.3. LIMITAÇÕES DO FILTRO COLABORATIVO

A filtragem colaborativa de dados também pode enfrentar o problema conhecido como “*cold-start*”, assim como o filtro baseado em conteúdo. É preciso ter uma quantidade razoável de usuários no sistema para encontrar similares que possam servir de base para gerar as recomendações, o que pode não ocorrer quando um sistema se encontra em sua fase inicial.

Tal problema pode ser evitado a partir da mistura das abordagens, o que implica na criação da abordagem denominada híbrida.

Outra limitação do filtro colaborativo está no fato de que não é possível gerar recomendações de um item que não foi classificado previamente. Tal problema é conhecido como “*first rater problem*”.

Ainda é possível que a gama de produtos disponíveis na plataforma analisada seja tão extensa a ponto de ser difícil encontrar um número suficientemente grande de usuários que classificaram os mesmos itens.

Por fim, existe o problema denominado viés de popularidade (“*popularity bias*”) que consiste em não conseguir gerar recomendações para usuários com preferências diferenciadas,

o que também pode ser resolvido misturando as abordagens do filtro baseado em conteúdo e do filtro colaborativo.

3.2.3. MODELOS DE CLUSTER

Além das duas abordagens citadas, existe uma terceira baseada no agrupamento de indivíduos em clusters. Tal abordagem surgiu devido ao custo de processamento envolvido no filtro colaborativo. Com um vetor de usuários extremamente esparso, torna-se computacionalmente caro gerar recomendações com filtro colaborativo baseado em usuários, segundo o Relatório Industrial da Amazon.com de Linden et al. (2003).

De acordo com tal relatório, o filtro colaborativo focado em usuários trata-se do filtro colaborativo mais tradicional e, com a evolução do número de clientes, realmente torna-se dispendiosa a recomendação nesse caso conforme a base de usuários se expande.

Com isso, surge a necessidade de reduzir o tamanho da base de dados trabalhando-se com amostras de usuários descartando os que possuem uma base pequena de aquisições, itens muito populares ou impopulares. Outra maneira seria repartir a base gerando categorias, os chamados clusters. Infelizmente, tais métodos reduzem a qualidade da recomendação realizada, apesar de reduzir custos.

Os modelos de clusters lidam com a questão da recomendação como se fosse um problema de classificação. O objetivo é associar o usuário analisado a um segmento específico que contenha o maior número possível de usuários similares.

Os segmentos são gerados a partir de técnicas de clusterização de dados ou outros algoritmos de *machine learning*. Geralmente, inicia-se com um grupo de clusters pré-estabelecido e um grupo aleatório de usuários selecionados que começam a ser classificados entre segmentos, de modo que, conforme o processo avança, é possível criar novos clusters se necessário ou mesclar grupos.

Uma vez que o algoritmo gera os segmentos e classifica os usuários entre eles, é calculada a similaridade entre os usuários em vetores que resumem cada segmento e o usuário é inserido naquele cujo vetor mais se aproxima do seu vetor.

Tais modelos possuem maior escala online que o colaborativo, por comparar o usuário com um número limitado de segmentos ao invés da base inteira de clientes. No entanto, a qualidade da recomendação é pobre.

3.2.4. ABORDAGEM HÍBRIDA

A fim de evitar os problemas mencionados no capítulo anterior, segundo Shah et al (2016), é possível combinar as duas abordagens descritas de quatro formas diferentes basicamente:

- I. Implementando as duas abordagens separadamente e combinando suas previsões;
- II. Incorporando algumas características da abordagem baseada em conteúdo à abordagem colaborativa;
- III. Incorporando algumas características da abordagem colaborativa à abordagem baseada em conteúdo;
- IV. Construindo um modelo unificado com ambas.

É possível fazer, por exemplo uma combinação da recomendação de ambas as abordagens, utilizando pesos em uma fórmula linear para ajustar a recomendação. Outra possibilidade seria decidir a abordagem que será utilizada caso a caso, evitando os casos desvantajosos de cada abordagem. Pode-se, ainda, misturar os resultados de ambas, levando-se em consideração o histórico de um usuário e a informação acerca da preferência de outros usuários, ou usar o resultado de uma abordagem para refinar o outro. Utilizar os outputs de uma abordagem como input para outra também é comum. São muitas as possibilidades de combinação entre os dois métodos a fim de refinar os resultados de uma recomendação.

3.3. MEDIDAS DE DESEMPENHO DOS MODELOS

Existem algumas maneiras de comparar modelos dos sistemas de recomendação segundo Shah et al. (2016), tais como a precisão, a acurácia, a medida F1, que seria equivalente à média harmônica entre precisão e acurácia, o erro absoluto médio da predição em relação à resposta real do usuário e o erro quadrático médio. Abaixo, segue o detalhamento de tais medidas.

$$P = \frac{\text{número de respostas corretas fornecidas}}{\text{número de respostas fornecidas}}$$

Medida de precisão

$$R = \frac{\text{número de respostas corretas fornecidas}}{\text{número de respostas a fornecer}}$$

Medida de *recall* (também chamada de acurácia)

$$F1 = \frac{2PR}{(P + R)}$$

Média harmônica entre precisão e *recall* (*F-score*)

$$MAE = \frac{\sum_{\{u,i\}} |p_{ui} - r_{ui}|}{N}$$

Média absoluta do erro (*mean absolute error*) onde p_{ui} é a resposta predita, r_{ui} é a resposta real e N é o número de observações da base de treino.

$$RSME = \sqrt{\frac{1}{|T|} \sum_{(u,r) \in T} (s_{ur} - \hat{s}_{ur})^2}$$

Erro quadrático médio no qual T é o tamanho da base de treino, s_{ur} é o valor real e \hat{s}_{ur} é o valor calculado.

Neste capítulo, portanto, foi possível compreender o que são sistemas de aprendizado de máquina e se aprofundar mais nos chamados sistemas de recomendação e suas variadas abordagens, resumidas abaixo:

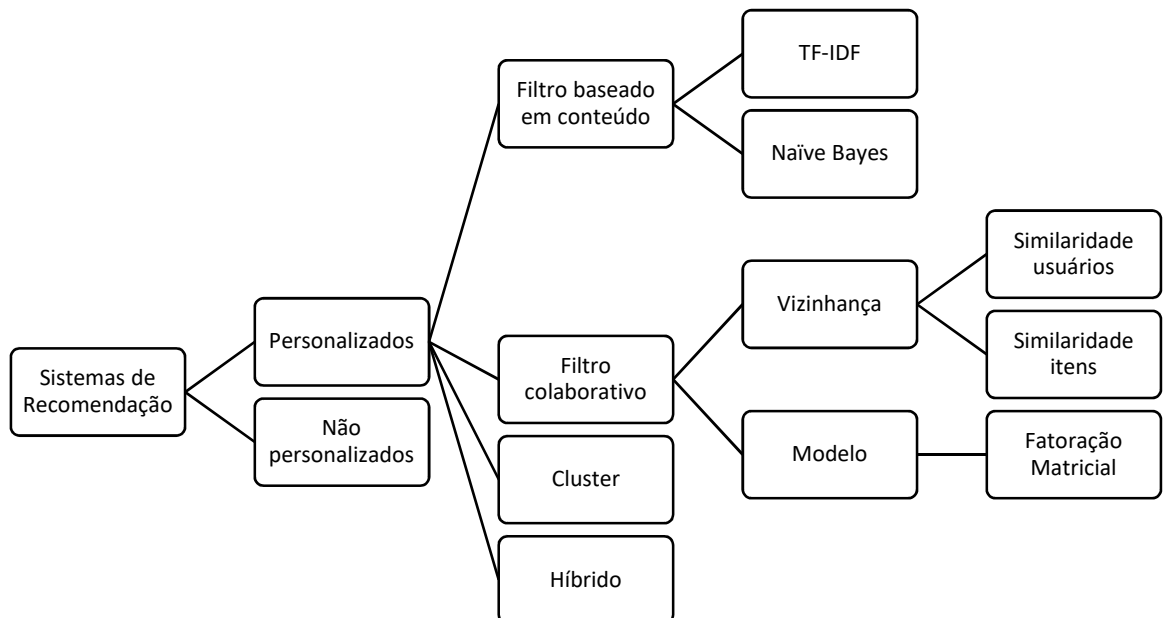


Figura 12: Resumo abordagens sistemas de recomendação. Fonte: a autora, baseada na descrição de Shah et al.

Além disso, foi possível ver as limitações envolvidas em tais sistemas, das quais as três principais estão destacadas abaixo:

- Problema *cold-start*;
- Problema do *first-rater*;
- Viés de popularidade.

Por fim foram vistas as medidas de desempenho mais comuns utilizadas para verificar resultados dos algoritmos.

No próximo capítulo, serão estudados exemplos práticos de referência no mercado de sistemas de recomendação.

4. EXEMPLOS PRÁTICOS DE REFERÊNCIA

Conforme reforçado inúmeras vezes no presente trabalho, sistemas de recomendação estão amplamente presentes no cotidiano e, muitas vezes, estão tão enraizados que sua presença se torna imperceptível em algumas aplicações. Este capítulo abordará dois casos indispensáveis sempre mencionados quando tal assunto é abordado, o caso Amazon e o caso Google.

4.1. O CASO AMAZON.COM

Os sistemas de recomendação são muito utilizados no universo dos *e-commerces online* para gerar uma lista de recomendações de itens a partir de uma série de inputs sobre os interesses dos usuários. Muitas aplicações utilizam as compras dos usuários e os ratings explícitos fornecidos por seus consumidores para classificar itens, mas também é possível utilizar outras referências tais como itens visualizados, dados demográficos, assuntos de interesse e artistas favoritos.

Um caso muito citado em artigos que abordam o tema sistemas de recomendação é o da Amazon.com, que se tornou referência no assunto em questão. No site desta companhia, algoritmos de recomendação são utilizados para personalizar a loja online de acordo com as preferências de cada consumidor. A página principal muda radicalmente de acordo com o perfil do cliente, exibindo itens que refletem suas preferências pessoais.

Segundo um Relatório Industrial da própria Amazon.com de Linden et al. (2003) os principais desafios relacionados aos algoritmos de recomendação em geral são:

- O volume intenso de informações acerca de consumidores e usuários, principalmente usuários antigos;
- A necessidade de gerar recomendações em tempo real, não mais do que meio segundo;
- A falta de informações acerca de novos usuários;
- A volatilidade das informações do usuário, de modo que cada nova informação precisa ser incorporada ao modelo rapidamente para gerar melhores previsões.

A Amazon.com utiliza o filtro colaborativo com foco em itens, de modo que identifica produtos classificados ou adquiridos pelo usuário e combina tais itens em uma lista de recomendação após calcular a similaridade de um dado produto com todos do catálogo.

A lógica pode ser resumida da seguinte maneira:

- i. Para cada item i do catálogo;
- ii. Para cada consumidor c que adquiriu i ;
- iii. Para cada produto i_2 adquirido por c ;
- iv. Decomposição de i em elementos básicos;
- v. Calcular a similaridade entre i e i_2 (para calcular a similaridade pode ser utilizado o cosseno do ângulo entre os vetores que representam os itens por exemplo).

Desse modo é formada uma tabela de itens similares e o algoritmo encontra os mais populares ou correlacionados com as compras do usuário e seus ratings.

Voltando ao resumo das abordagens do capítulo 3, o método poderia ser classificado conforme destacado em azul abaixo:

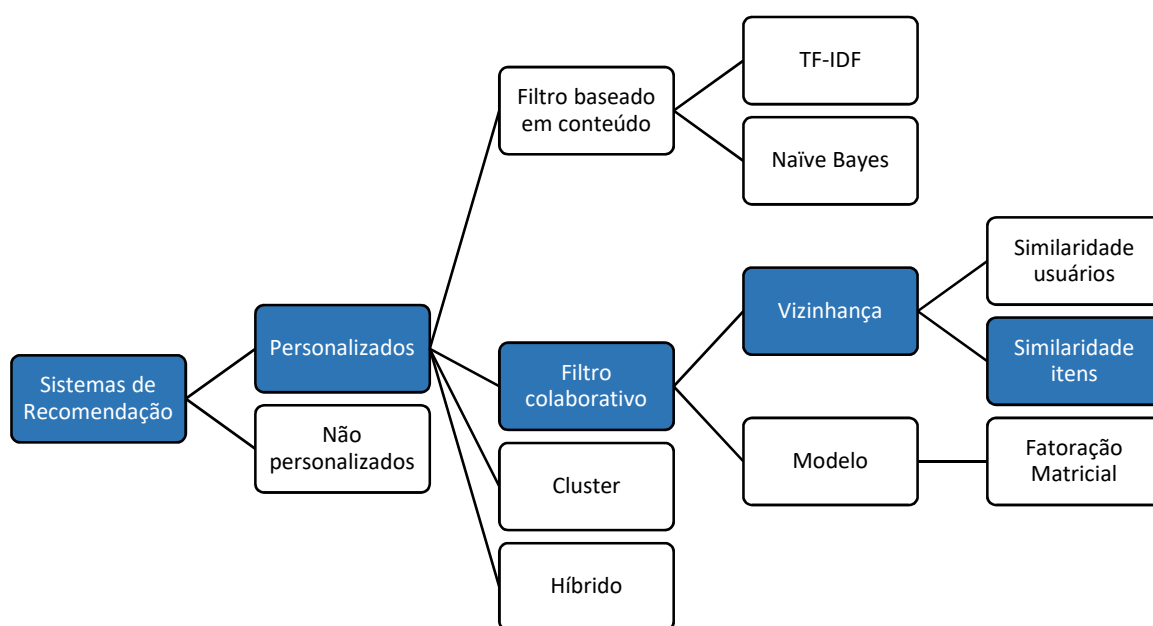


Figura 13: Resumo da abordagem específica utilizada no caso Amazon.com conforme classificação do capítulo 3. Fonte: a autora

Nesse caso, o método escolhido para o cálculo da similaridade entre itens seria o ângulo entre os cossenos dos vetores que representam tais itens com seus respectivos consumidores.

A grande questão computacional que torna o filtro colaborativo mais caro é o fato deste realizar muito pouco ou nenhum cálculo *offline*. E o cálculo *online* fica mais pesado com o aumento da base de clientes e itens. O algoritmo torna-se pouco prático em bases muito extensas, de modo a exigir uma redução de escala ou a formação de clusters, o que reduz a qualidade das recomendações. A vantagem do filtro colaborativo com foco em itens é que a

criação da tabela de itens similares é feita *offline*. A parte online depende somente do número de títulos que o usuário classificou ou comprou e não do tamanho total do catálogo ou do número total de usuários, o que aumenta consideravelmente a performance do algoritmo.

4.2. O CASO PAGERANK (GOOGLE INC)

Com a quantidade de informações disponível na web atualmente, seria praticamente impossível encontrar um resultado desejado na rede sem algum tipo de auxílio. Segundo Azevedo Filho (2020), professor da USP, até 1998 vários algoritmos tentaram encontrar a solução para esse problema, baseados em três pilares essenciais:

- Percorrer as páginas públicas com programas rastreadores cadastrando palavras-chave e links existentes;
- Cadastrar e organizar as palavras-chave em índices, ligando as palavras aos links dos sites;
- Construir um *ranking* de importância ou popularidade de cada site que possui as palavras cadastradas, a fim de proporcionar ao usuário os melhores resultados possíveis para o tema desejado.

A princípio, o buscador Altavista avançou bastante no que diz respeito aos dois primeiros pilares, mas o terceiro só foi implementado com qualidade com a vinda do Pagerank em 1998, desenvolvido em Stanford pelos fundadores da então recém-criada Google. A posição de uma página é basicamente calculada por uma constante que representa a probabilidade de um buscador seguir pela base de páginas disponíveis e encontrar o documento.

O algoritmo se baseia em noções associadas de cadeia de Markov e álgebra linear envolvendo autovalores e autovetores. O índice de popularidade de uma página de interesse é estabelecido de acordo com os sites que possuem links apontando para esta, ponderando pelo índice de popularidade deles.

De acordo com Azevedo Filho (2020), podemos considerar o exemplo abaixo para exemplificar, como uma mini-web:

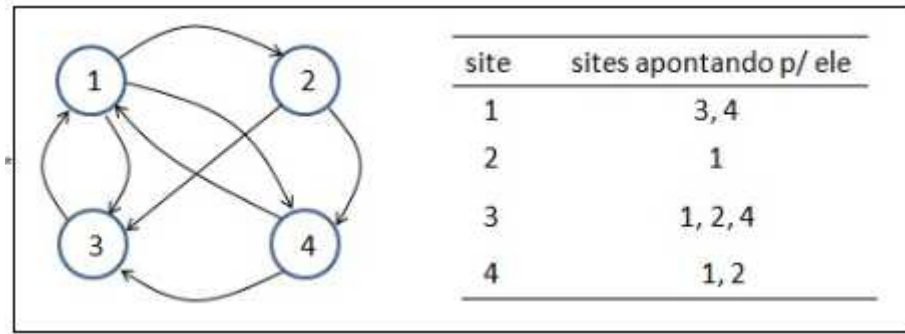


Figura 14: Exemplo mini-web. Fonte: Rpubs, professor Adriano Azevedo Filho (USP).

Considerando a lógica das aplicações que tentaram criar um *ranking* de popularidade dos sites antes de 1998, a página 3 seria considerada a mais popular entre as quatro do universo hipotético criado. No entanto, a grande ideia por trás do Pagerank está no fato de que a popularidade dos sites que apontam para outro deve ser considerada para ponderar a contagem. Cada link de um site A apontando para um site B é computado como um “voto” para B e deve ser ponderado pela popularidade do site A. Cada página aponta para s páginas, o que é computado como votos, computados como $1/s$.

Portanto, os “votos”, links entre os sites, seriam representados por uma matriz (P) de tamanho $n \times n$, onde n representa o número de páginas públicas disponíveis na rede. Tal matriz seria o autovetor 1 de outra matriz, a de popularidade de cada site (π), autovetor de P .

O sistema seria descrito, portanto, na forma matricial por $P \pi = \pi$ onde:

$$\pi = [\pi_1, \pi_2, \pi_3, \pi_4]^T$$

$$\left\{ \begin{array}{l} \pi_1 = \pi_3 + \frac{1}{2} \pi_4 \\ \pi_2 = \frac{1}{3} \pi_1 \\ \pi_3 = \frac{1}{3} \pi_1 + \frac{1}{2} \pi_2 + \frac{1}{2} \pi_4 \\ \pi_4 = \frac{1}{3} \pi_1 + \frac{1}{2} \pi_2 \end{array} \right.$$

$$\bullet \quad P \pi = \pi \text{ em que } P = \begin{pmatrix} 0 & 0 & 1 & \frac{1}{2} \\ \frac{1}{3} & 0 & 0 & 0 \\ \frac{1}{3} & \frac{1}{2} & 0 & \frac{1}{2} \\ \frac{1}{3} & \frac{1}{2} & 0 & 0 \end{pmatrix} \text{ e } \pi = \begin{pmatrix} \pi_1 \\ \pi_2 \\ \pi_3 \\ \pi_4 \end{pmatrix}$$

Figura 15: Matrizes P e π . Fonte: Rpubs, professor Adriano Azevedo Filho (USP).

A matriz P possui 4 autovalores distintos, sendo dois reais e dois complexos, de modo que um deles possui valor 1. O algoritmo basicamente descobre os autovalores e autovetores de P e o autovetor associado ao autovalor 1 é normalizado, convertido de complexo para número real e representa o índice de popularidade de cada site. No caso do exemplo proposto, o site que possui o maior índice de popularidade é o 1, apesar de visualmente o problema aparentemente apontar para 3.

No Google, antes de uma pesquisa ocorrer, os rastreadores da web coletam informações das centenas de bilhões de páginas existentes e organizam isso no índice de pesquisa. O processo se inicia a partir de uma lista de endereços de rastreamentos anteriores e os chamados *sitemaps*, arquivos em que os proprietários dos sites fornecem informações sobre os mesmos.

O *software* concentra-se em sites novos ou alterações dos já existentes. Programas definem quais sites serão rastreados e com que frequência. O Google oferece o Google Console para que os proprietários tenham instruções detalhadas de como é feito o rastreamento de seus sites, promovendo a igualdade de ferramentas oferecidas.

Os rastreadores analisam páginas disponíveis e seguem seus links, levando dados aos servidores Google, que são registrados no índice de pesquisa. Atualmente, no índice registram-se outras informações para melhorar a pesquisa.

Voltando ao resumo das abordagens do capítulo 3, portanto, o método do Pagerank poderia ser classificado conforme destacado em azul abaixo:

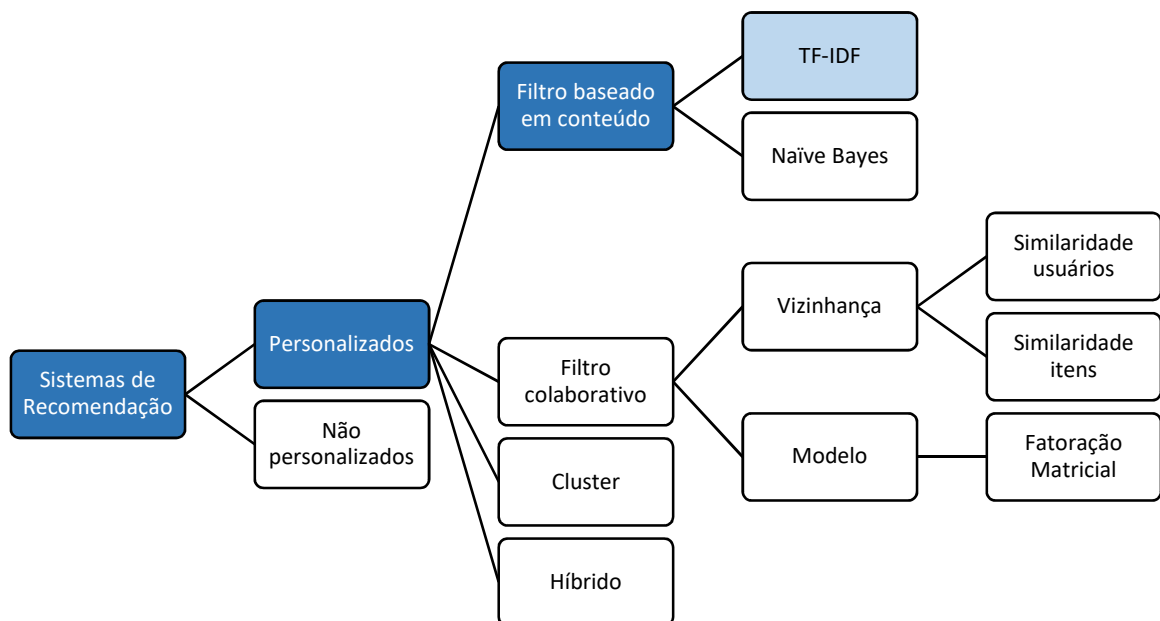


Figura 16: Resumo da abordagem específica utilizada no caso Pagerank conforme classificação do capítulo 3.

Fonte: a autora.

O método para realizar o filtro de conteúdo nesse caso dentro da abordagem de filtro baseado em conteúdo seria diferente das tradicionais mais utilizadas: TF-IDF e Naive Bayes, embora utilize um pouco da ideia de ambas na lógica de elaboração do algoritmo de filtragem de conteúdo. Para construir a matriz que resume os links entre páginas da *web*, por exemplo, a ideia seria um pouco do já descrito no capítulo 3 sobre TF-IDF, a frequência de páginas apontando para outras ponderada pela popularidade.

Neste capítulo, portanto, foi possível visualizar dois exemplos práticos importantíssimos na história dos sistemas de recomendação. Um baseado em similaridade de itens no filtro colaborativo e outro baseado na filtragem de conteúdo.

No próximo capítulo, será abordado como esse conceito de sistemas de recomendação se aplica aos produtos financeiro em uma era de bancos digitais, os desafios envolvidos nesse tema e as experiências já realizadas levando em conta a diversidade de produtos demonstrada no subcapítulo 2.2.

5. APLICAÇÕES DE SISTEMAS DE RECOMENDAÇÃO EM FINANÇAS: REFERENCIAL TEÓRICO

Como foi possível observar nos capítulos anteriores, os sistemas de recomendação são aplicados em uma série de domínios no cotidiano onde há uma vasta opção de escolha e o usuário está interessado em apenas uma pequena porção de itens. Quando são abordados tais algoritmos, automaticamente o assunto é relacionado a *e-commerce* e *streaming*, com catálogos de produtos como filmes e séries (Netflix), livros (Amazon) ou música (Spotify).

No entanto, nunca é realizada de imediato a associação desses algoritmos ao campo dos produtos financeiros, embora tal possibilidade já tenha sido abordada, testada, documentada em alguns artigos e até mesmo realizada como um serviço de *private banking* internacionalmente, o que será objeto de discussão deste capítulo dividido em duas partes.

A primeira parte abordará os desafios e benefícios envolvidos nessa aplicação segundo a literatura pesquisada referente ao assunto e a segunda descreverá na prática as aplicações realizadas nesse campo.

5.1. DESAFIOS E BENEFÍCIOS ENVOLVIDOS NA APLICAÇÃO EM FINANÇAS

São vários tanto os benefícios quanto os desafios envolvidos na aplicação de sistemas de recomendação no mundo financeiro listados na literatura. Segundo Zibriczky (2016), o maior desafio está no fato de que, comparados a esses domínios convencionais onde os sistemas são aplicados, os produtos financeiros diferenciam-se por envolverem questões que vão muito além de realizar uma utilidade imediata, tais como:

- Comprometimento de longo prazo;
- Realização da utilidade que depende de uma série de fatores externos tais como retornos de mercado, regulamentações governamentais, moedas etc.
- Risco de privacidade das informações.

Este último em especial gera o problema de *cold-start* em sistemas de recomendação. Por se tratar de uma questão delicada, usuários geralmente tendem a proteger seus dados pessoais quando o assunto é finanças, o que acaba gerando o problema destacado no capítulo 3 do presente trabalho na aplicação de sistemas de recomendação.

Ainda de acordo com Zibriczky (2016) tal tarefa torna-se ainda mais desafiadora devido aos vários domínios financeiros existentes com características distintas. Para ele, um domínio financeiro seria uma área específica em finanças que pode ser identificada, modelada e

desenvolvida com base em propriedades específicas. Cada domínio financeiro seria classificado conforme os seguintes aspectos:

- i. Heterogeneidade das propriedades de um item pertencente ao domínio;
- ii. O nível de novidade e expectativa de duração de um item do domínio;
- iii. A forma de interação com o usuário, que seria a forma como este consegue expressar sua preferência por um determinado item do domínio;
- iv. O grau de variação das preferências de um item do domínio ao longo do tempo;
- v. O risco que determina a tolerância esperada dos usuários por recomendações que não façam sentido;
- vi. O nível de demanda que existe pela explicação da recomendação de um determinado item do domínio.

Desse modo, portanto, “ações” seria um domínio financeiro enquanto “fundos de investimento” seria outro, por exemplo.

Além disso, um material do Grupo Incube⁷ de 2018 destaca outras questões importantes envolvidas nessa aplicação, tais como:

- Ausência de *feedbacks* explícitos por parte dos usuários;
- Mudança de natureza dos produtos ao longo do tempo, um dos fatores que diferencia produtos financeiros dos produtos de e-commerce por exemplo;
- Impacto financeiro que uma recomendação desse tipo pode ter na vida do usuário.

No entanto, o Grupo Incube, que trabalha desenvolvendo soluções de inteligência artificial com sua equipe de *data scientists* na Suíça, justamente nesse campo de finanças, destaca que apesar de desafiadora tal aplicação é possível e apresenta uma série de benefícios. Ela seria uma opção interessante para o segmento do varejo, por exemplo, considerando que tais clientes não costumam possuir um atendimento especializado tal como o dos clientes classificados como Alta Renda pelas instituições financeiras.

Os gerentes que atendem clientes de Alta Renda, segundo o material suíço, normalmente possuem entre 50 e 200 clientes e precisam processar informações manualmente para gerar recomendações personalizadas, de modo que seu campo de visão é restrito e eles não possuem acesso às informações de clientes de outros gerentes.

⁷ *InCube Group*, empresa suíça desenvolvedora de soluções com inteligência artificial para administração de patrimônio e indústria de seguros, adquirida em 2020 pela Finantix, empresa do mesmo ramo.

Com um sistema de recomendação, seria possível encontrar clientes semelhantes na base inteira de clientes, as recomendações realizadas teriam uma maior chance de serem aceitas de modo que a qualidade da recomendação aumentaria e tempo gasto para gera-la seria reduzido. Uma consequência interessante disso seria a inclusão dos clientes do varejo em uma classe com acesso a recomendações personalizadas, conforme ilustrado abaixo.

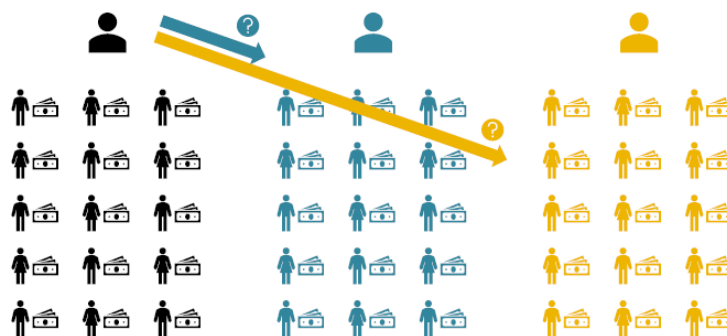


Figura 17: Visão atual dos gerentes de alta renda de uma instituição financeira. Fonte: Grupo Incube, 2018.



Figura 18: Aplicando sistemas de recomendação à base de clientes completa. Fonte: GrupoIncube, 2018.

Conforme já discutido no capítulo 2 deste trabalho, é visível o avanço dos bancos digitais no Brasil e inclusive a mudança dos bancos tradicionais para se adaptar a um mercado que engloba um novo modelo de negócio quando se fala em investimentos pessoais. Desse modo, a aplicação da tecnologia a favor de gerar um diferencial para os clientes nas plataformas de investimento seria um benefício muito vantajoso no sentido de agregar valor aos serviços fornecidos a base inteira de clientes. O Grupo Incube também deixa claro que a digitalização abre as portas para um aumento do índice de ativação de clientes nas plataformas.

Yahyapour (2008) resume algumas vantagens que os sistemas de recomendação trouxeram ao *e-commerce*. A principal delas, que se aplicaria a plataformas de investimentos,

seria a construção da lealdade do cliente, em um mundo competitivo e digital, no qual uma empresa deve investir em aprender sobre seus clientes e buscar aprimorar sua interface com eles para atender suas necessidades. Quanto mais um usuário utiliza um sistema de recomendação, mais este aprende sobre suas preferências, de modo que é construída uma lealdade de difícil replicação pelos concorrentes, que precisarão investir o mesmo tempo e a mesma energia para adquirir tal conhecimento.

No próximo subcapítulo, serão apresentadas algumas aplicações práticas encontradas em artigos e no material do Grupo Incube, que trabalha no ramo e possui uma série de publicações acerca do assunto, documentos que serviram de inspiração e referência para esse trabalho.

5.2. EXEMPLOS DE APLICAÇÕES DE REFERÊNCIA EM FINANÇAS

Conforme citado na introdução deste capítulo, a aplicação de sistemas de recomendação em finanças especificamente já foi abordada em artigos, testada, e até mesmo realizada como serviço para bancos internacionalmente.

No site do Grupo Incube, que aplica tais algoritmos em suas soluções, há uma série de publicações datadas entre 2018 e 2020 detalhando a lógica de construção dos sistemas de recomendação que foram colocados em produção para seus clientes. Suas propostas consistem basicamente em um processo construído em 5 blocos:

- i. Agrupamento dos ativos;
- ii. Filtragem colaborativa usando como base a posição histórica ou a atividade de transações como *proxy* de ratings implícitos;
- iii. Utilização de um modelo de *machine learning* complementar para gerar as explicações do modelo de filtragem colaborativa;
- iv. Seleção dos produtos individuais ranqueados dentro de cada caixa de agrupamento de ativos;
- v. Incorporação do *feedback* explícito dos clientes para retroalimentar o sistema.

Portanto, inicialmente, os sistemas do grupo usam de 10 a 20 atributos para agrupar ativos, tais como classe do ativo, moeda, setor, vencimento, estrutura de eventos, entre outras. Isso porque tal grupo aplica suas soluções em bases de dados que contêm produtos de vários

domínios distintos, de modo que fica difícil trabalhar com tantos produtos de investimentos e suas peculiaridades sem agrupá-los não apenas por classe de ativo, mas segundo características específicas dentro de uma classe em alguns casos. Lembrando que no capítulo 2 foi abordada a variedade de produtos nas bases de dados das plataformas de investimento no Brasil, por exemplo. A metodologia de agrupamento do Grupo Incube leva em consideração fatores como quão esparsa é matriz de ratings, o efeito da cauda de distribuição produtos-cliente, diversificação das recomendações entre outros. Agrupar ativos é vantajoso nesse caso porque reduz o problema de *cold-start*, reduz o efeito espalhado da matriz de *ratings* e consequentemente o problema da cauda-longa da distribuição⁸, reduz a dimensão dos dados – tornando o modelo mais prático no quesito execução –, além de permitir trabalhar com produtos que possuem atributos que variam no tempo, o que é o caso de produtos de investimento, que hoje podem ser considerados ótimos, mas amanhã nem tanto.

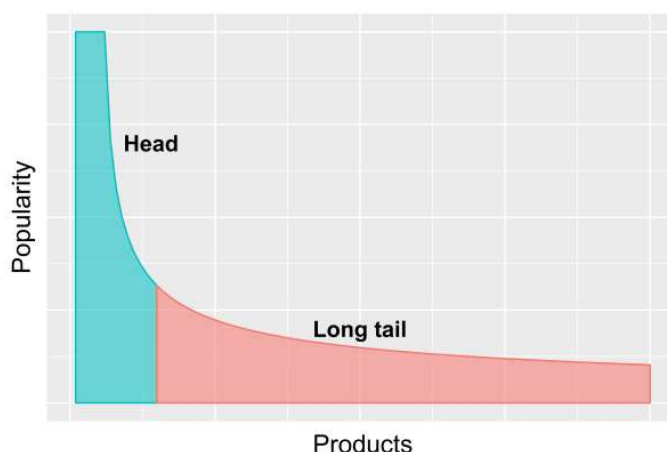


Figura 19: Ilustração do problema da cauda longa na distribuição de produtos para sistemas de recomendação em finanças. Fonte: Site do Grupo Incube.

Após o agrupamento dos ativos da base de seus clientes, o Grupo Incube parte então para a aplicação do modelo de filtragem colaborativa utilizando ratings implícitos como *proxy* para a satisfação do cliente. Para isso existem várias formas de trabalhar, como as citadas abaixo por exemplo:

- Pode-se utilizar feedbacks (implícitos) positivos quando os clientes compraram ou adquiriram o produto no passado ou neutro quando não compraram;
- Observação da posição histórica, se ela aumentou ou diminuiu ao longo do tempo.

⁸ Problema ocasionado pelo fato de que geralmente a maioria dos usuários dos sistemas de recomendação consome uma pequena porção de itens da base de produtos.

O Grupo destaca que um fator importante nesse tipo de serviço é que recomendações no ramo de finanças precisam ser confiáveis e que muitas vezes as métricas de desempenho tradicionais citadas no capítulo 3 do presente trabalho podem não ser suficientes para explicar para o investidor porque determinado produto lhe foi recomendado pelo sistema. Os clientes geralmente preferem saber qual lógica levou tal produto a ser recomendado de forma personalizada para eles na prática quando tal solução é oferecida como um serviço. No entanto, principalmente em sistemas de recomendação com filtro colaborativo baseados em modelo tal explicação não é muito óbvia. Por isso, o Grupo Incube costuma utilizar outros algoritmos de *machine learning* para construir a explicação da recomendação dada pelo filtro colaborativo de forma mais intuitiva, de modo a identificar o cluster dentro da matriz de consumo que gerou o resultado do primeiro modelo. A lógica busca gerar uma explicação na linha de pensamento dos sistemas do *e-commerce*, tal como “como você gostou de A, você provavelmente vai gostar de B”.

Por fim, o sistema de recomendação do grupo precisa sair da visão agrupada para o output final que é um produto ranqueado dentro da caixa de ativos selecionada. Para isso, segundo eles, é preciso levar em consideração também uma série de fatores, tais como restrições de produtos considerados apropriados pela instituição, restrições regulatórias, restrições do perfil de aversão a risco do cliente, país e histórico – para não recomendar um ativo que o cliente já adquiriu anteriormente, por exemplo. Além disso, é importante que o sistema seja capaz de se retroalimentar com os feedbacks explícitos dos clientes a respeito de suas recomendações.

Voltando ao resumo das abordagens do capítulo 3, portanto, a metodologia do Grupo Incube poderia ser classificada de forma simplificada conforme destacado em azul abaixo:

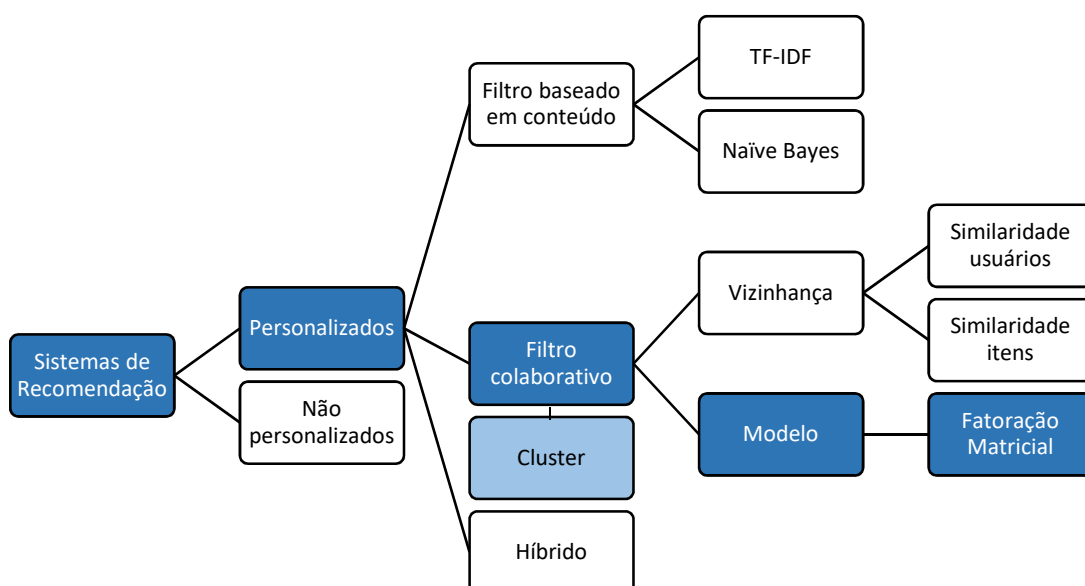


Figura 20: Resumo da abordagem utilizada pelo Grupo Incube em suas soluções. Fonte: A autora.

Além do caso prático citado anteriormente, em que uma companhia presta serviços usando seus conhecimentos de *machine learning* outras empresas, trabalhos acadêmicos envolvendo este assunto já foram feitos internacionalmente.

Segundo Goldstein et al. (2019), um aspecto importante do fenômeno das chamadas *fintechs*, que unem finanças e tecnologia, é que grande parte dele está ocorrendo fora dos EUA. A China está claramente a frente dessa tecnologia e outros países com mercados financeiros menos desenvolvidos também possuem uma atividade considerável nesse campo. Os países emergentes estão bastante voltados para estudos relacionados a tais inovações. É possível observar isso no gráfico abaixo, que demonstra a composição geográfica de autores que submeteram propostas de artigos ao The Review of Financial Studies sobre o tema em 2017.

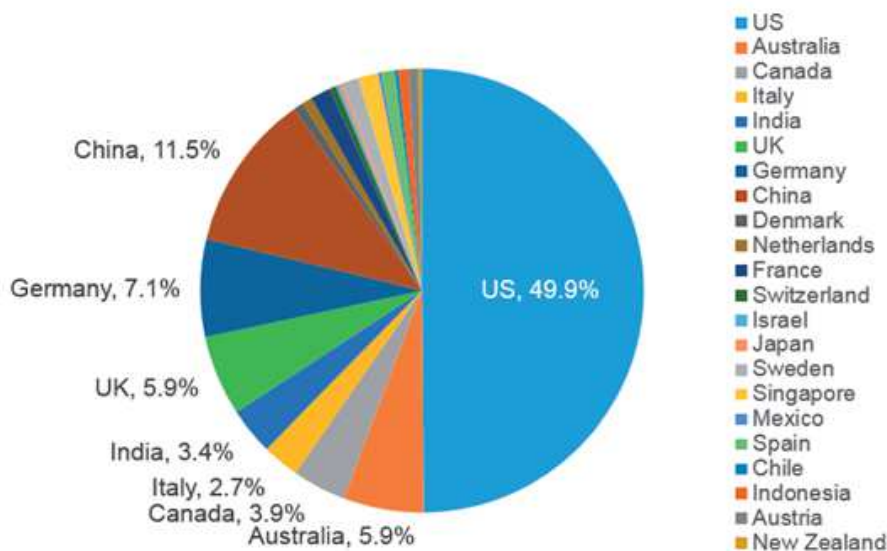


Figura 21: Composição geográfica dos autores que propuseram artigos ao The Review of Financial Studies em 2017. Fonte: Goldstein et al. (2019).

Em um *paper* da Universidade de Solapur na Índia de 2013, Sayyed et al. (2013) relata o uso de filtro colaborativo especificamente no mercado de ações para gerar recomendações a novos usuários aplicando os seguintes passos:

- i. Coleta de dados públicos de séries de preços no período de alguns meses das ações, a fim de observar seu padrão de comportamento ao longo do tempo;
- ii. Aplicação da filtragem colaborativa utilizando o software Apache Mahout⁹ nos dados transacionais e de preços das companhias a fim de observar sua similaridade.

⁹ Sistema de recomendação da Fundação Apache que utiliza recomendação personalizada de filtragem colaborativa.

A ideia nesse caso é gerar uma matriz de similaridades que indica quantas vezes um produto se comportou de maneira similar a outro ao longo do tempo. Se ações de uma companhia são compradas por dois usuários no mesmo momento e estão com retornos positivos, o sistema recomendaria tal companhia a um terceiro usuário recém-chegado prevendo que ele também teria retornos positivos no futuro.

Nesse caso, voltando ao resumo do capítulo 3, a metodologia utilizada pode ser resumida conforme destacado em azul no diagrama abaixo:

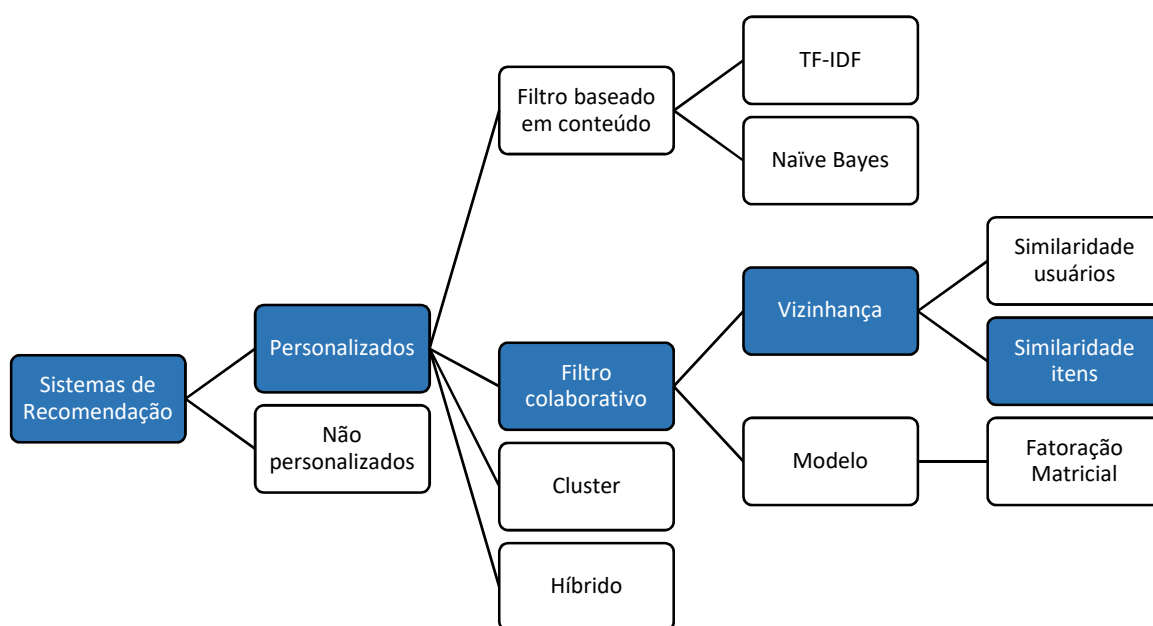


Figura 22: Resumo da abordagem utilizada no artigo de Sayyed. Fonte: A autora.

Outro trabalho interessante também associado indiretamente ao domínio das ações é o de Zhang et al. (2018), que aborda um sistema de recomendação de fundos de ações combinando técnicas de *transfer learning*¹⁰ e função utilidade. No *paper* realiza-se uma análise comportamental a partir da criação de um perfil de fundos de ação e de um perfil de investidores sendo que, como o mercado de fundos de ação é limitado, utiliza-se o perfil dos investidores de ações como base. A ideia é transferir o conhecimento acerca do mercado de ações para o mercado de fundos de ação, que são basicamente fundos que possuem seu foco de investimento voltado para ações em uma composição definida pelo gestor em relação ao setor das companhias e nível de risco. Este artigo especificamente cria um modelo diferente dos tradicionais citados no presente trabalho e inclusive compara tal modelo com a aplicação da abordagem de similaridade tradicional vista anteriormente.

¹⁰ Transferência de conhecimentos entre mercados

Para construir um perfil de fundos de ação as características consideradas por Zhang et al. (2018) são:

- i. Alocação de ativos, ou seja, valor percentual do volume alocado em uma determinada indústria em relação ao patrimônio do fundo;
- ii. Escala de capitalização de mercado (grande, média ou pequena);
- iii. Taxa de retorno média no período do investimento e desvio padrão dos retornos para medir o risco.

Além disso, para construir o perfil dos investidores no mercado de ações, as características consideradas foram:

- i. Preferência de indústria (valor total investido em determinada indústria sobre valor total investido);
- ii. Preferência de escala (valor investido em empresas de escala x sobre o valor total investido);
- iii. Média dos ganhos dos investimentos;
- iv. Desvio padrão dos retornos do investidor;
- v. Taxas máximas de ganhos e perdas.

Posteriormente é criada uma função utilidade para cada usuário a partir desses indicadores. Supondo que e_{jk} é o valor esperado de um determinado usuário (U_j) sobre o atributo C_k e suas expectativas podem ser descritas pelo vetor $E_j = (e_{j1}, e_{j2}, \dots, e_{jm})$, sendo m o número de atributos de C_k . Além disso, sendo o vetor Q_i o representante dos reais atributos do fundo F_i , tal que $Q_i = (q_{i1}, q_{i2}, \dots, q_{im})$ a construção do vetor de decisão entre comprar ou não o fundo pode ser definido pelo vetor $D_i = (d_{i1}, d_{i2}, \dots, d_{im})$ e seus componentes definidos por:

$$d_{ik} = q_{ik} - e_{ik}$$

Assim foi construída a função de valor do estudo de Zhang et al. (2018), demonstrada abaixo:

$$V(d_{ik}) = \begin{cases} (d_{ik})^\alpha, & d_{ik} \geq 0 \\ -\Theta(d_{ik})^\alpha, & d_{ik} < 0 \end{cases}$$

$$V_i = \sum_{k=1}^m w_k V(d_{ik})$$

Onde: w_k é o peso do atributo C_k e V_i descreve o valor do fundo F_i para o investidor U_j . Quanto maior o valor de V_i , mais desejável o fundo F_i se torna para o investidor. Os coeficientes α e β são coeficientes que representam as atitudes de risco e Θ é um coeficiente de aversão a perdas.

Por fim, após a definição da função utilidade que os autores do artigo utilizam no modelo, os seguintes passos são executados para gerar as recomendações aos usuários:

- i. Passo 1: padronizar todos os dados e separar as observações em treino e teste;
- ii. Passo 2: definir o perfil do fundo;
- iii. Passo 3: definir o perfil do investidor;
- iv. Calcular a similaridade entre o investidor U_j e todos os outros fundos usando a função de similaridade:

$$sim(F_i, U_j) = \frac{1}{distância(F_i, U_j)} = \frac{1}{\sqrt{\sum_{k=1}^{n1} (F_{ik} - U_{jk})^2}}$$

- v. Depois basta ranquear os fundos de forma decrescente pela similaridade.

Sendo que o input e o output do modelo são os seguintes:

- Input: base transacional de ações e fundos, alocação de fundos e sua performance;
- Output: fundos recomendados.

Posteriormente, os autores comparam os resultados de seu modelo alternativo com os resultados da aplicação dos algoritmos de recomendação padrão de filtragem colaborativa baseada na vizinhança e concluem que seu modelo funciona melhor neste mercado específico de fundos de ações. Tal artigo de acesso aberto da Universidade Internacional de Negócios e Economia na China fornece referências para instituições financeiras recomendarem produtos e serviços para investidores que possuem uma distribuição de investimentos muito esparsa e de cauda longa, casos nos quais não é recomendável a utilização das técnicas básicas de filtro colaborativo.

Retornando ao diagrama do capítulo 3 novamente, tal método poderia ser classificado conforme destacado no diagrama abaixo:

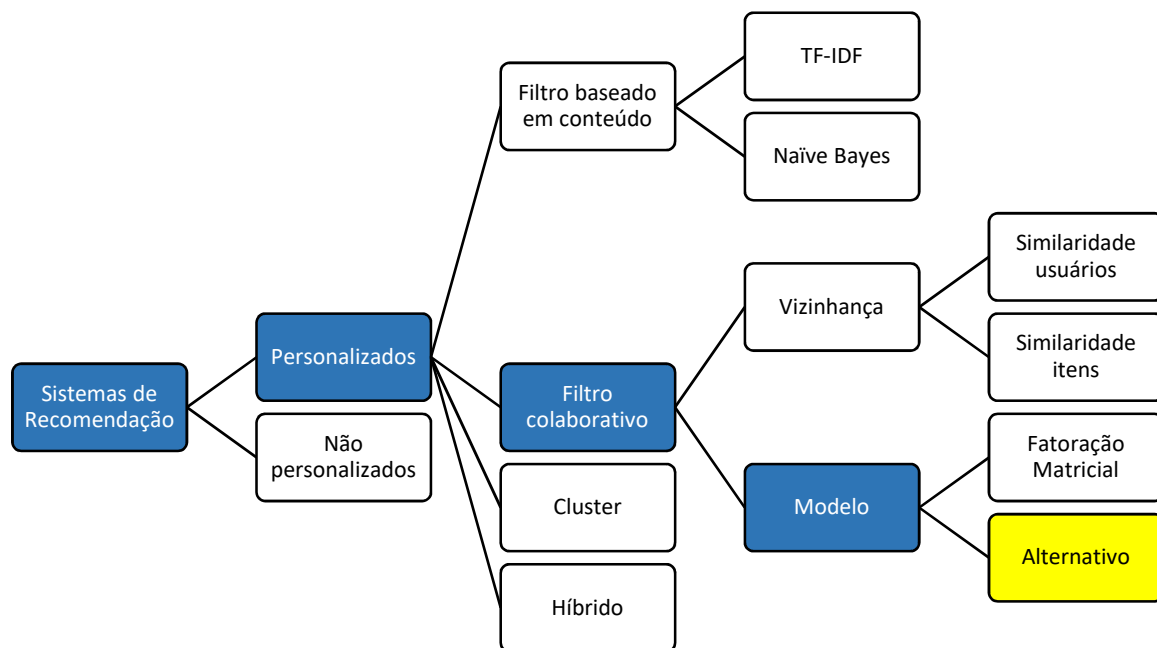


Figura 23: Resumo da abordagem utilizada no *paper* de Zhang et al (2018). Fonte: A autora.

Portanto, conforme visto acima, são vários e diversificados os trabalhos e métodos práticos realizados envolvendo este assunto. Em seu artigo, Zibriczky (2016) cita outros de maneira mais geral, buscando resumir as iniciativas de estudo nesse campo em finanças com base em sua definição de domínios do subcapítulo anterior. Inicialmente ele começa citando trabalhos mais gerais em online banking e posteriormente parte para domínios mais específicos, sem descrever muito no detalhe os métodos, mas apenas para exemplificar o tipo de trabalho que já foi feito.

No campo mais geral do *online banking* ele cita Yahyapour et al. (2008), que investigaram a introdução de sistemas de recomendação nos bancos iranianos a partir de *feedbacks* explícitos, Huecas et al. (2012) que utilizaram histórico de compras com cartões de crédito e localização geográfica para recomendar oportunidades de compra nas proximidades. Também é citado Fano et al. (2003) que sugere uma ferramenta pessoal de alocação financeira considerando os objetivos dos usuários, despesas e tempo de alocação desejado.

Posteriormente Zibriczky (2016) parte para domínios mais específicos como empréstimos, onde ele resume o problema de recomendação a encontrar o produto correto da instituição que realiza o empréstimo que atenda às necessidades do devedor e que seja mais provável de ser honrado por este. Os trabalhos desenvolvidos nessa área basicamente focam em utilizar *machine learning* para melhor formar os pares credor-devedor, considerando risco de crédito dos usuários, por exemplo.

No domínio dos seguros, ele cita que os contratos e os seguradores podem ser alvo do problema de recomendação personalizada, com base nos benefícios e no prêmio, por exemplo, que é possível modelar de várias formas a melhor recomendação para o segurado de acordo com seus objetivos.

Há, ainda, uma literatura relativamente restrita no campo do *Real Estate*, propriedades, seus recursos naturais e construções. Para isso, é preciso considerar uma série de preferências no modelo, o que leva a um problema de decisão multicritério.

Embora os domínios citados acima possuam alguma literatura, o mais citado em problemas sistemas de recomendação é o domínio das ações, que domina duas das aplicações práticas citadas no início deste subcapítulo, títulos que representam uma participação em uma companhia, seus ativos, ganhos e dividendos. A aplicação de sistemas de suporte de decisão no mercado de ações possui uma literatura ampla. A maioria das contribuições foca em desenvolver a acurácia de previsões de retornos futuros ou tendências, concedendo sinais de compra/venda ou introduzindo soluções de *trading* automático, embora ignorem o fator personalização, em sua grande maioria.

De acordo com Zibriczky (2016), um grande número de *papers* aponta que grupos possuem mais conhecimento que indivíduos isolados acerca de ações, o que é chamado de “sabedoria das multidões”. Por isso, investidores inexperientes podem se beneficiar com indicações de comunidades online, de modo que uma grande quantidade de trabalhos também foca em processos que incorpora notícias relacionadas a finanças e textos das redes sociais que possam afetar o mercado para realizar previsões. Nesse sentido, ele cita o trabalho de Ravi e Ravi (2015) como referência sobre o *data mining* de opiniões e sentimentos de mercado.

A fim de prover recomendações personalizadas, informações acerca dos indivíduos são necessárias, embora suas preferências explícitas nunca estejam disponíveis na maioria dos casos. Uma forma de superar este problema é disponibilizar uma interface na qual o usuário possa especificar suas preferências. No entanto, alguns trabalhos consideram que as características do usuário já se encontram disponíveis na base de dados. No Apêndice A do presente trabalho, outros exemplos de trabalhos citados no artigo de Zibriczky (2016) no que se refere a ações são citados.

O domínio das ações pode ser considerado heterogêneo, por apresentar companhias de vários setores, a taxa de cancelamento é rara, pois companhias deixam o mercado muito raramente, a interação com os usuários e a classificação encontra-se implícita na base de transações, a preferência do usuário é instável sempre conduzida por notícias e movimentos na

economia global. Trata-se de uma recomendação particularmente delicada e demanda explicações particulares de acordo com o autor deste artigo.

Existe também uma literatura a respeito da recomendação de portfólios, uma composição finita de ativos financeiros com determinados pesos que respeitam um certo nível de risco-retorno que se deseja atingir. Devido ao efeito diversificação, tais carteiras possuem menos riscos que as ações individuais, por exemplo. A técnica de compor um portfólio é chamada de *asset allocation* e a questão consiste em estimar pesos de ativos financeiros que reflitam as preferências individuais e a desejada relação risco-retorno.

O mais conhecido modelo de seleção de portfólios foi publicado por Markowitz e, segundo Zibriczky (2016) pode ser interpretado como um problema de recomendação em dois passos: primeiramente, portfólios bem diversificados devem fornecer a melhor relação risco-retorno para cada nível de risco e são objetos de recomendação e, em segundo lugar, um investidor é modelado a partir de sua função utilidade de aversão a risco que classifica cada oportunidade de investimento a partir de risco-retorno. Dentre os trabalhos nessa linha estaria a proposta particular de Zhang et al. (2018) citada neste subcapítulo. A ideia é selecionar portfólios que maximizem a função utilidade do investidor. Elton e Gruber (2000) argumentam que investidores podem tomar decisões irracionais de portfólio e que recomendações automáticas são vantajosas no sentido de que previnem uma decisão irracional. A lista de trabalhos citados por Zibriczky (2016) neste domínio pode ser encontrada no Apêndice B do presente trabalho.

O artigo de Zibriczky (2016), portanto, buscou listar no geral as contribuições científicas para os problemas de recomendação relacionados a serviços financeiros. É possível observar que um extenso trabalho de investigação neste campo vem sendo realizado ao longo dos anos, mas ainda há muitas oportunidades a serem exploradas com relação ao assunto. Percebe-se que já houve uma evolução nesse assunto quando se compara o artigo de 2016 ao trabalho que vem sendo realizado pelo Grupo Incube (2018), por exemplo, que possui publicações mais recentes no ramo e realiza trabalhos em uma base de dados mais generalizada que contém produtos de domínios distintos, o que exige uma lógica bem mais complexa, explicada de forma simplificada no presente trabalho.

No próximo capítulo, serão enfatizadas as ferramentas que tornam todas as aplicações práticas descritas neste subcapítulo possíveis, os softwares disponíveis do mercado que possuem sistemas de recomendação.

6. SISTEMAS DE RECOMENDAÇÃO DISPONÍVEIS NO MERCADO

No cenário de BigData explorado por este estudo, no qual é preciso analisar grandes volumes de informação para geração de recomendações para usuários de diversos serviços, formas de recomendação vêm sendo exploradas por serviços provedores de recursos da internet e portais de *e-commerce*. O grande objetivo atualmente é identificar melhor os gostos dos usuários, tendo assim melhor desempenho no momento de recomendar um determinado produto aos clientes.

Segundo publicação disponível no Github¹¹, sistemas de recomendação são “pedaços” de *software*, que possuem muitos desafios, tais como processar uma grande quantidade de informações com rapidez e eficácia, memória e manter as informações do usuário seguras simultaneamente.

Basicamente existem dois tipos de sistemas de recomendação no mercado segundo Jenson: os Softwares de Sistemas de Recomendação como um serviço¹² e os Sistemas de recomendação *open source*¹³, conforme será possível ver nos subcapítulos a seguir.

6.1. SISTEMAS DE RECOMENDAÇÃO COMO UM SERVIÇO

Os chamados Softwares de Sistemas de Recomendação como um serviço, na maioria das vezes pago, auxiliam companhias que desejam melhorar seu desempenho ao recomendar produtos aos usuários, mas necessitariam de um grande investimento em seu *front* para atingir tal tecnologia. Seis o caso, por exemplo, do serviço personalizado que o Grupo Incube, adquirido pela empresa Finantix em 2020, citado nos capítulos anteriores. No Github são listados alguns serviços disponíveis nesse sentido, em que companhias em sua maioria pagam para obter essa capacidade com AI¹⁴, que tende a gerar melhoria contínua conforme a utilização dos sistemas progride. Nos sites dessas companhias não é possível obter informações detalhadas sobre como funcionam na prática seus algoritmos, mas é interessante perceber a diversidade de serviços que existem, adaptáveis a várias necessidades e áreas distintas, conforme é possível perceber no diagrama resumido abaixo. A descrição mais detalhada de tais empresas citadas

¹¹ Comunidade de desenvolvedores que dá suporte a mais de 40 milhões de pessoas, aprendendo, compartilhando e trabalhando para construir *softwares* desde 2007.

¹² *Software as a Service Recommender System (SaaS Recommender System)*

¹³ *Open Source Recommender Systems (Non SaaS Recommender System)*

¹⁴ Inteligência artificial

pelo Github e vistas um pouco mais detalhadamente aqui pode ser encontrada no Apêndice C do presente trabalho.

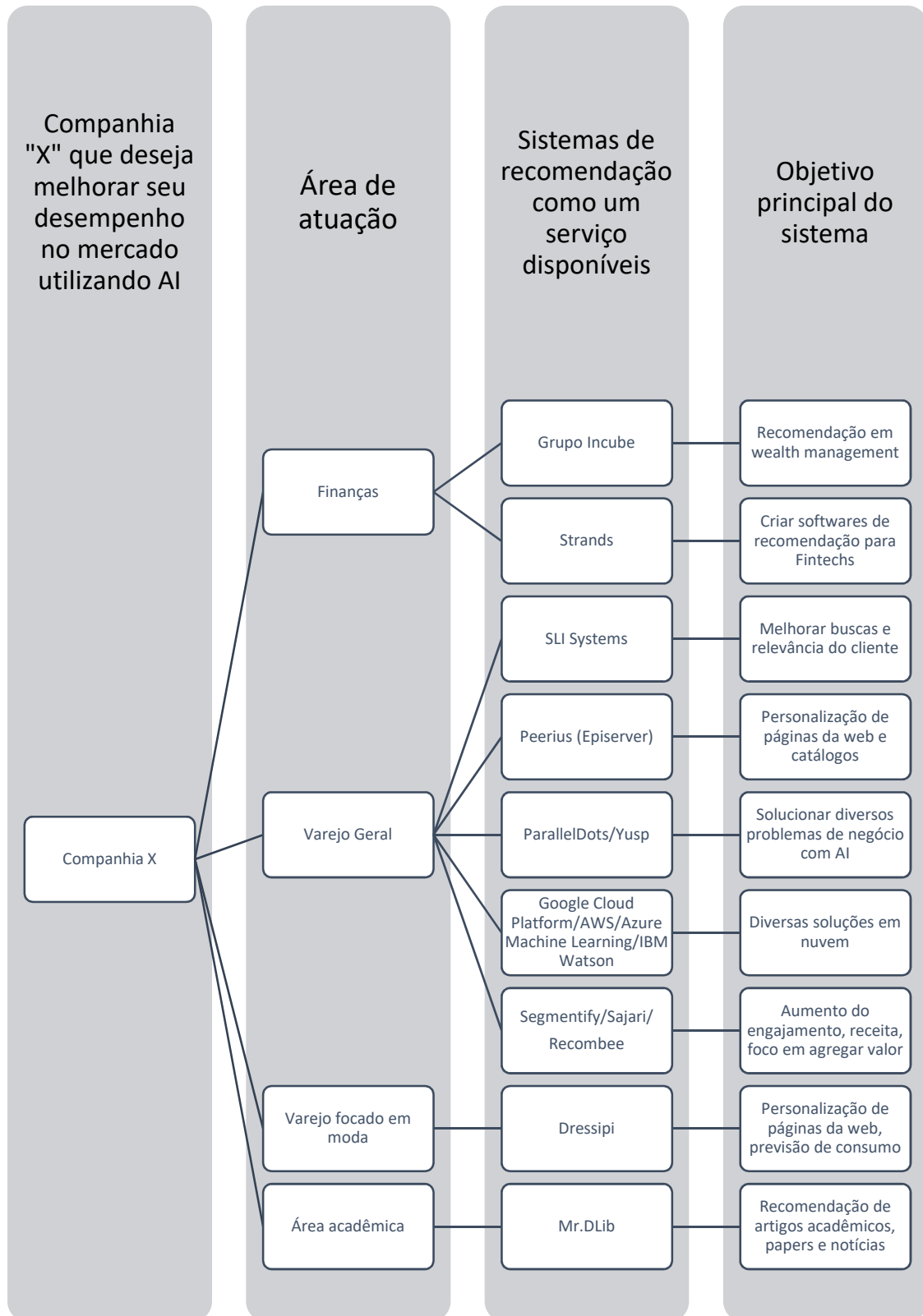


Figura 24: Resumo dos sistemas de recomendação como um serviço citados pelo Github. Fonte: a autora.

6.2. SISTEMAS DE RECOMENDAÇÃO *OPEN SOURCE*

A maioria dos sistemas de recomendação não-SaaS são *open source*, ou seja, são códigos abertos adaptáveis para diferentes fins, isso provavelmente porque tais sistemas são mais voltados para clientes e não tão facilmente convertidos em produtos. Tratam-se de softwares gratuitos que podem ser utilizados para atender a objetivos pessoais e que contam de comunidades como o Github, por exemplo, onde é possível encontrar pessoas que trabalham em conjunto para desenvolver soluções para problemas muitas vezes semelhantes aos dos usuários, de modo que não se cria dependência de um grupo de suporte em particular. Segundo Jenson, alguns exemplos de sistemas *open source* podem ser resumidos pelo diagrama abaixo.

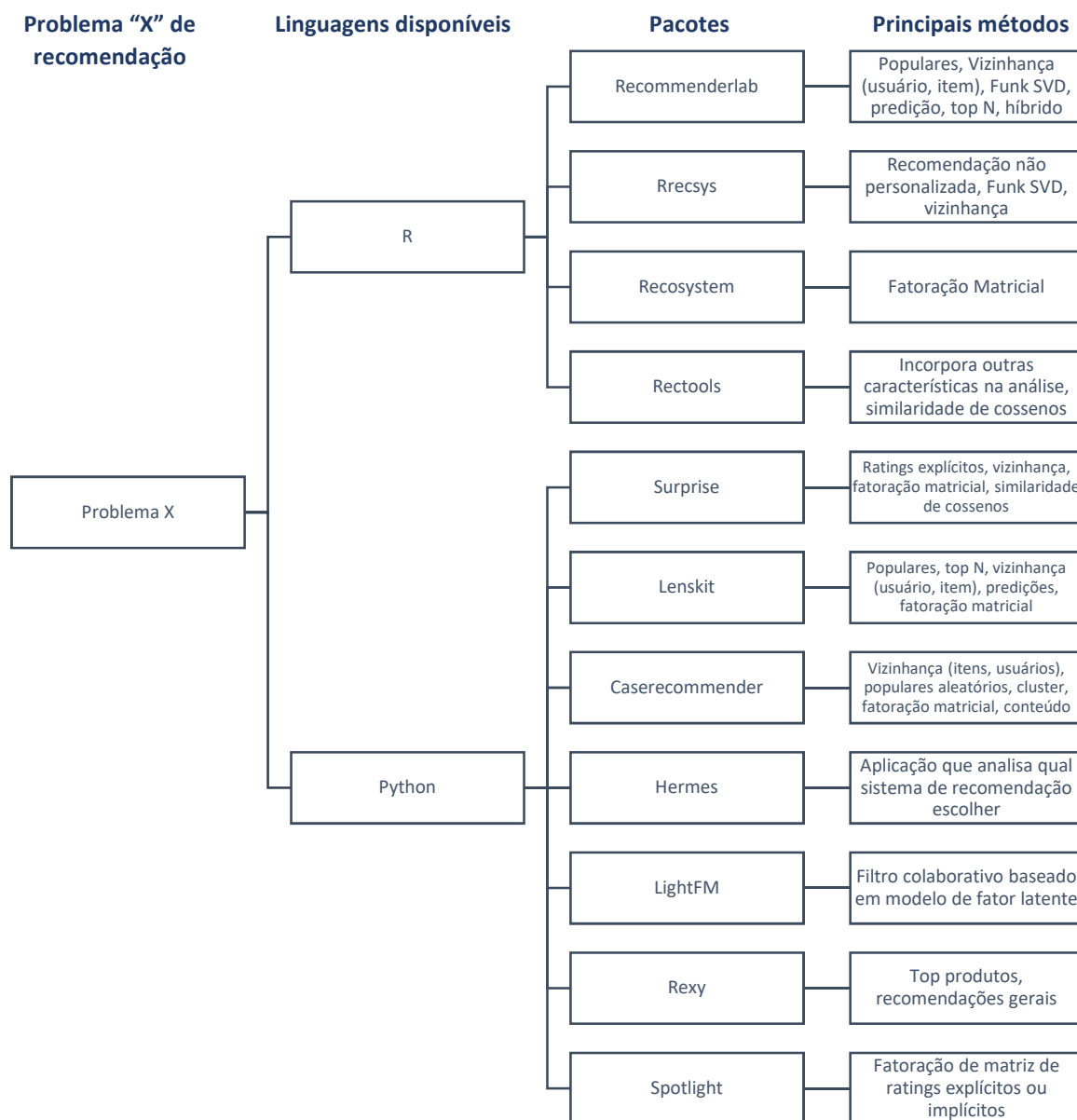


Figura 25: Principais pacotes *open source* em Python e R de recomendação e seus métodos. Fonte: a autora com base em Jenson.

Como foi possível observar, no diagrama resumido acima, a maioria dos pacotes existentes possui possibilidades variadas de métodos que podem ser utilizados. Alguns, como o *reco*system do R por exemplo, limitam-se a focar em apenas um método, nesse caso o de fatoração matricial. No entanto a maioria envolve várias possibilidades que incluem recomendações não personalizadas, personalizadas de filtro colaborativo focadas em usuário ou item, recomendação de top N produtos, predições, filtro colaborativo baseado em modelo com fatoração matricial tradicional e FunkSVD (como descrito no capítulo 3 deste trabalho). São poucos os que falam em modelos de cluster e recomendação baseada em conteúdo, como o *Caserecommender* do Python. Um caso diferente apontado entre os exemplos é o *Hermes* que, diferentemente das outras soluções consiste em um sistema de recomendação de sistemas de recomendação, indicando qual seria o mais adequado para o problema em questão a partir da análise do desempenho de vários sistemas.

É importante ressaltar que apesar de não mencionado no diagrama acima, praticamente todos os pacotes citados possuem formas de avaliar seus resultados a partir das métricas de desempenho citadas no capítulo 3. Além disso, existem outros pacotes em outras linguagens com os mesmos objetivos, conforme resumido no diagrama da figura 26. A descrição mais detalhada de todos encontra-se disponível no Apêndice D deste trabalho, embora seus métodos sejam descritos apenas de forma abrangente.

Um fator interessante que precisa ser citado, é que dentro da caixa dos sistemas de recomendação personalizados de filtro colaborativo baseados em modelo, em 2013 surgiram os sistemas multimodais, aqueles que buscam reaproveitar todo tipo de dado que possa representar um indicador das preferências do cliente para aprimorar recomendações. Dentre as aplicações da Fundação Apache é possível identificar softwares mais completos e complexos nesse sentido. Voltando ao diagrama do capítulo 3, portanto, o diagrama incluindo os multimodais ficaria da seguinte maneira:

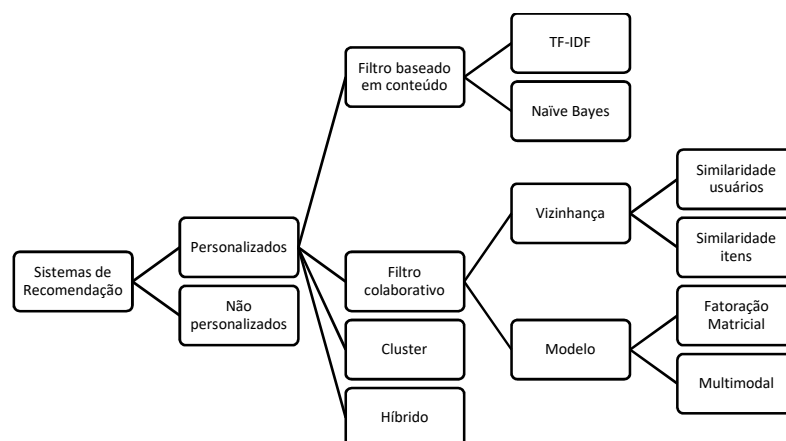


Figura 26: Incluindo os sistemas multimodais. Fonte: a autora.

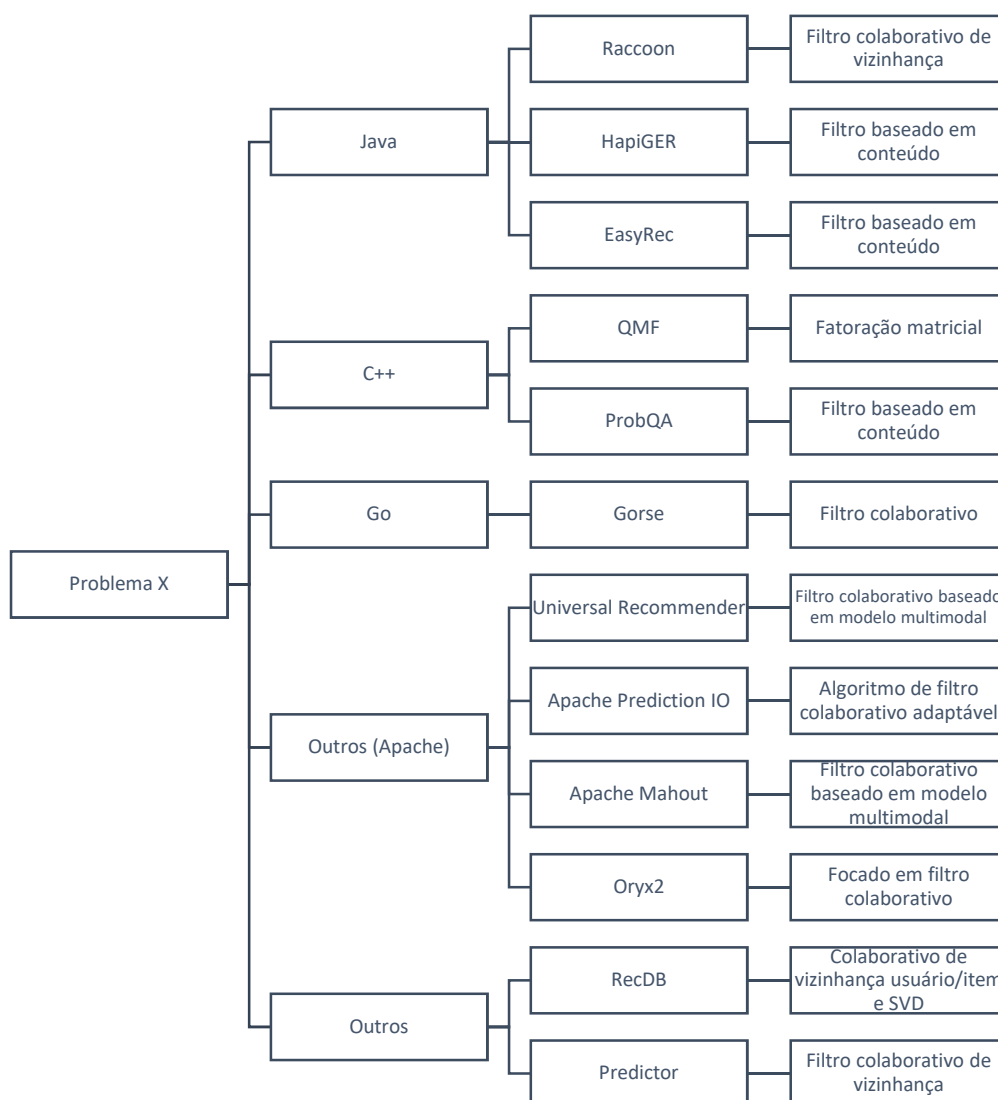


Figura 27: Principais pacotes *open source* em outras linguagens. Fonte: a autora com base em Jenson.

Portanto, nesse capítulo foi possível ver os sistemas de recomendação disponíveis no mercado, tanto os que funcionam como um serviço prestado a empresas quanto aqueles que consistem em códigos abertos, com o suporte de programadores em comunidades online desenvolvendo soluções variadas. Dentre eles foi possível observar que a maioria foca nos métodos de filtragem colaborativa de vizinhança ou em modelo. Apenas os pacotes Java e C++ parecem ser mais focados em memória/conteúdo. Nesse capítulo também foi possível observar que existem sistemas de filtragem colaborativa baseados em modelo multimodal, que buscam aproveitar todas as informações do usuário como uma possível fonte de informações de preferências a serem incorporadas no modelo.

No apêndice E do presente trabalho, será detalhado com mais ênfase o pacote *recommenderlab* do R apresentado na figura 24, seus métodos, parâmetros e hiperparâmetros.

No próximo capítulo, tal pacote será utilizado enfim para a aplicação prática do presente trabalho, em duas bases transacionais de produtos de banco digital, exemplificando o problema descrito ao longo de toda essa dissertação.

7. PROJETO DE CIÊNCIA DE DADOS: APLICAÇÃO PRÁTICA EM BASES TRANSACIONAIS

Após todo o detalhamento do que são e de como vêm crescendo no Brasil as chamadas plataformas de investimento, os *shopping centers* de produtos financeiros, quão extensas são suas bases, o que são sistemas de recomendação, seus métodos e como eles vêm sendo aplicados tanto no e-commerce quanto em problemas relacionados a finanças, chegou o momento de finalmente aplicar os conhecimentos adquiridos ao longo dos capítulos anteriores a bases reais transacionais de produtos financeiros.

O presente capítulo vai descrever a aplicação de algoritmos de recomendação do pacote *recommenderlab* do *software* R, detalhado no apêndice E deste trabalho, a duas bases transacionais basicamente, uma de ações e outra de fundos de investimento a fim de demonstrar o que foi visto ao longo desse estudo.

7.1. COLETA DOS DADOS, TRATAMENTO PARA *INPUT* E DESCRIÇÃO

Inicialmente, para a aplicação prática do trabalho em questão foram coletados dados transacionais no período de 252 dias úteis para dois produtos em especial gerando duas análises segregadas a partir de duas bases: uma de ações da bolsa de valores e outra de fundos de investimento abertos.

Posteriormente, tais bases foram trabalhadas a fim de chegar ao formato do input para serem consumidas e analisadas pelo pacote *recommenderlab*. Tanto os usuários quanto os itens foram mascarados de modo a se tornarem apenas códigos totalmente desvinculados de suas informações privadas originais, a fim de gerar uma análise segura apenas para fins de estudo.

As informações de *input* da base de ações foram trabalhadas de modo a se resumir, portanto, a um código de usuário, totalmente desvinculado de qualquer informação pessoal, um código de item, também desvinculado de qualquer informação original, um código de setor, transformado em número para também preservar informações e ao retorno calculado para o papel no período de 252 dias úteis, que serviu como referência de rating implícito na base.

O metadado de setor utilizado na base para os itens nesse caso foi gerado a partir de informações da B3. A lista de setores presentes na base pode ser conferida abaixo. Os códigos da base não necessariamente seguem essa ordem.

Tabela 1: Setores presentes na base transacional de ações do presente trabalho. Fonte: B3 resumida pela autora.

Setores base ações
Saúde
Financeiro
Consumo não Cíclico
Consumo Cíclico
Outros
Utilidade Pública
Bens Industriais
Materiais Básicos
Petróleo, Gás e Biocombustíveis
Tecnologia da Informação
Comunicações

Resumindo a base de ações, após as manipulações realizadas para chegar ao formato desejado de input para o *recommenderlab*, foi possível obter as seguintes informações:

```
'data.frame': 293151 obs. of 4 variables:
 $ USUARIO : int 1 1 1 1 1 3 19 19 19 19 ...
 $ ITEM : int 110 166 50 211 3 27 154 107 126 88 ...
 $ SETOR : int 8 11 7 4 3 2 4 8 2 6 ...
 $ RETORNO252: num 0.18 -0.52 0.03 -0.35 -0.01 3.3 0.71 0.09 0.43 1.58
 ...
```

USUARIO	ITEM	SETOR	RETORNO252
Min. : 1	Min. : 1.0	Min. : 1.000	Min. : -0.8600
1st Qu.: 8953	1st Qu.: 73.0	1st Qu.: 2.000	1st Qu.: 0.0000
Median :18505	Median :126.0	Median : 4.000	Median : 0.3300
Mean :18787	Mean :129.5	Mean : 5.077	Mean : 0.6888
3rd Qu.:28199	3rd Qu.:183.0	3rd Qu.: 7.000	3rd Qu.: 0.6500
Max. :40748	Max. :254.0	Max. :11.000	Max. :20.0800

Figura 28: Resumo da base de ações inserida no software R. Fonte: a autora.

Coluna Input Modelo	Tipo Variável	Número total de observações	Valor mínimo	Valor Máximo
Usuário	Inteiro	40.748	1	40.748
Item	Inteiro	254	1	254
Setor	Inteiro	11	1	11
Retorno Item 252 dus - Rating Implícito	Real	293.951	-0,86	20,08

Tabela 2: Resumo descritivo base ações. Fonte: a autora.

Desse modo, foi possível observar que ela possui 293.951 observações de ratings implícitos em 1 ano (252 dias úteis), com 40.748 usuários e 254 itens de 11 setores distintos, mascarados, com retornos que vão de -86% a 2008% no período analisado.

Analisando a composição de observações por setor, chegou-se ao seguinte gráfico:

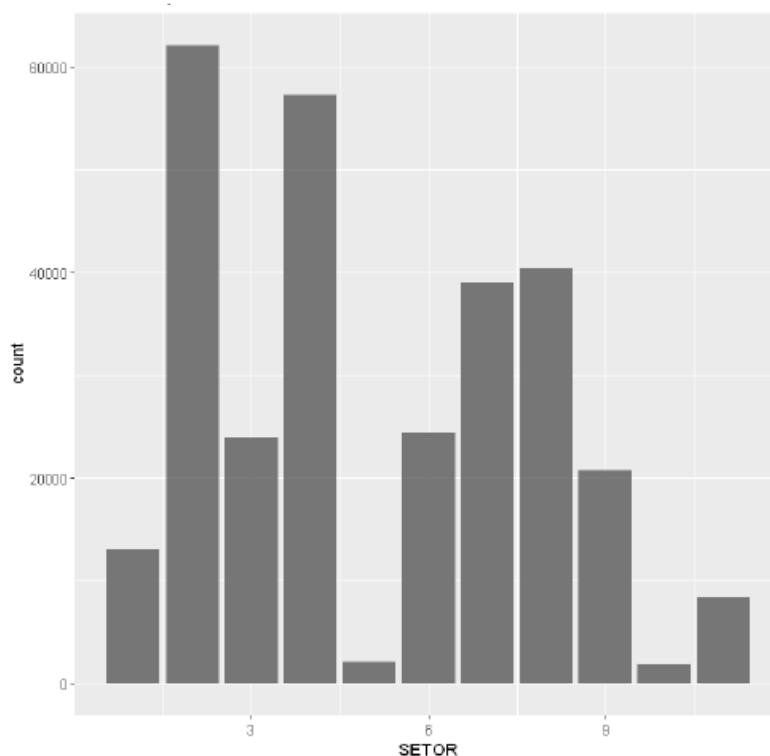


Figura 29: Resumo das observações por setor da B3. Fonte: a autora.

É possível observar que os setores 2 e 4 dominam a maior parte das observações da base, provavelmente contendo ações mais negociadas no mercado, ou pelo menos nessa amostra de mercado representada pela plataforma digital observada.

Observando os dados mais de perto é possível perceber quão esparsa é a base de *ratings* nesse caso e que o fenômeno da cauda longa é perceptível na distribuição dos mesmos. A imagem demonstra uma pequena porção dos dados apenas para visualização. Já o histograma é uma visão da base inteira.

```
library(recommenderlab)
BASEACOES_2 <- cbind(BASEACOES[,1:2],BASEACOES[,4])
BASEFINAL <- as(BASEACOES_2,"realRatingMatrix")

image(BASEFINAL[1:300,1:254]) |
```

Figura 30: Código para visualizar a imagem da base. Fonte: a autora.

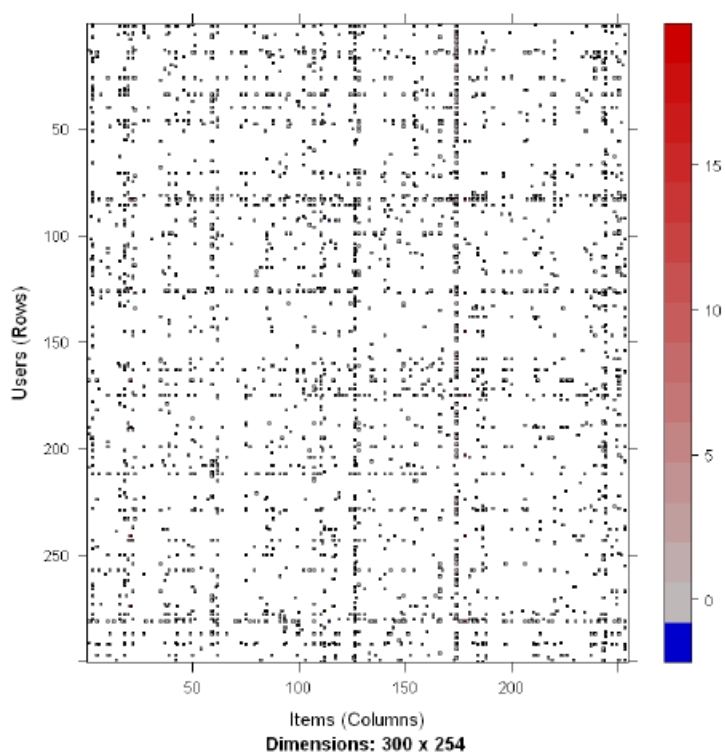


Figura 31: Imagem da base de ações. Fonte: R.

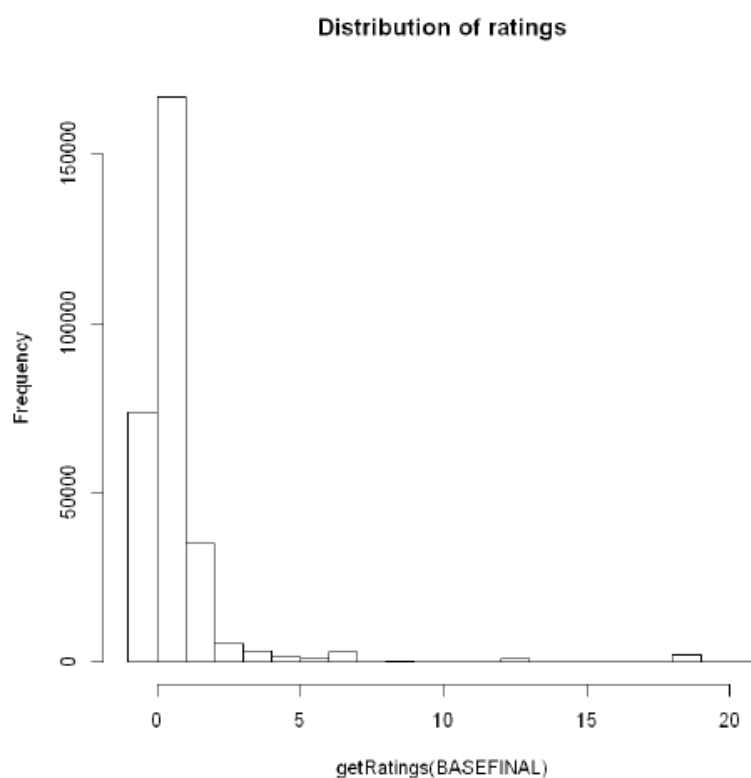


Figura 32: Histograma dos ratings na base de ações. Fonte: R.

As informações de *input* da base de fundos de investimento foram trabalhadas de modo a se resumir a algo semelhante ao que se observou na base de ações. Uma coluna de usuário, uma de item, uma de tipo e uma de retorno calculado para 252 dias úteis.

O metadado de tipo foi gerado a partir de uma classificação de 4 grupos definidos pela autora de acordo com a composição da base. Eles foram resumidos abaixo e não correspondem necessariamente aos itens da base na mesma ordem apresentada.

Tabela 3: Tipos de fundo presentes na base transacional do presente trabalho. Fonte: a autora.

Tipos de fundo

Fundo de Investimento Multimercado

Fundo de investimento em Ações

Fundo de Investimento em Renda Fixa

Fundo Cambial

Resumindo a base de dados transacional de fundos de investimento, após as manipulações para inserção no software R, foi possível obter as seguintes informações:

```
'data.frame': 38654 obs. of 4 variables:
 $ USUARIO : int 1 2 3 4 4 5 5 5 6 7 ...
 $ ITEM : int 1 2 3 4 5 2 1 4 4 4 ...
 $ TIPO : Factor w/ 4 levels "A","B","C","D": 3 3 3 3 3 3 3 3 3 3 ...
 $ RETORNO252: num 0.0729 0.1012 0.0809 0.0663 0.0952 ...
```

USUARIO	ITEM	TIPO	RETORNO252
Min. : 1	Min. : 1.00	A: 206	Min. : -0.007305
1st Qu.: 5174	1st Qu.: 9.00	B:12491	1st Qu.: 0.072469
Median :10782	Median : 15.00	C:17391	Median : 0.096670
Mean :10932	Mean : 25.63	D: 8566	Mean : 0.197163
3rd Qu.:16550	3rd Qu.: 32.00		3rd Qu.: 0.298141
Max. :22870	Max. :108.00		Max. : 0.573997

Figura 33: Resumo da base de fundos de investimento inserida no R. Fonte: R

Coluna Input Modelo	Tipo Variável	Número total de observações	Valor mínimo	Valor Máximo
Usuário	Inteiro	22.870	1	22.870
Item	Inteiro	108	1	108
Tipo	Inteiro	4	1	4
Retorno Item 252 dus - Rating Implícito	Real	38.654	-0,0073	0,57

Tabela 4: Resumo descritivo base fundos. Fonte: a autora.

Desse modo, foi possível observar que ela possui 38.654 observações de ratings de fundos de investimento abertos em 1 ano (252 dias úteis), com 22.870 usuários e 108 itens de

4 tipos, mascarados, assim como o restante da base por segurança. Os retornos vão de -0,73% a 57% no período analisado.

Na mesma linha de análise que foi realizada para as ações, é possível montar um resumo das observações por tipo de fundo, conforme abaixo:

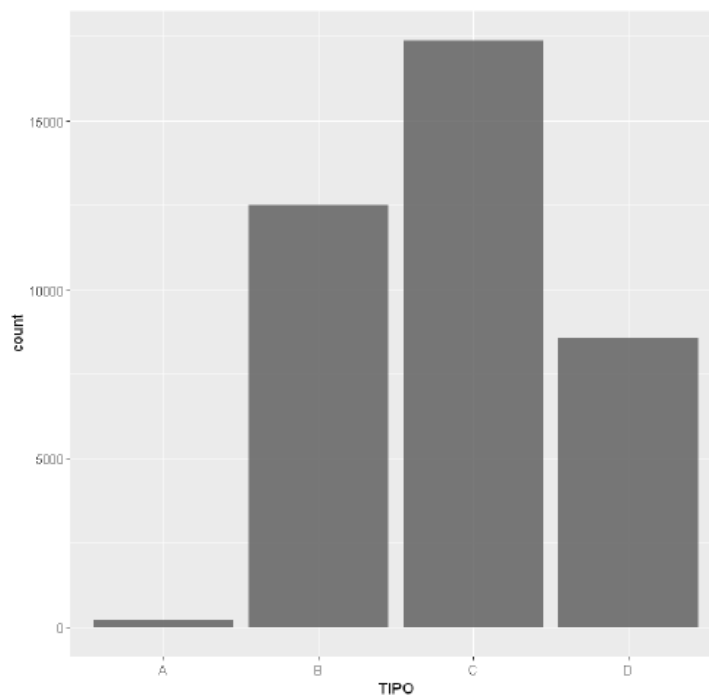


Figura 34: Resumo das observações por tipo. Fonte: a autora.

Além disso, é possível visualizar uma imagem da base de dados percebendo também quão esparsas são as observações, embora mais enxutas e concentradas do que na base de ações. Quanto a visualização dos ratings o fenômeno da cauda longa é menos observável que na base de ações.

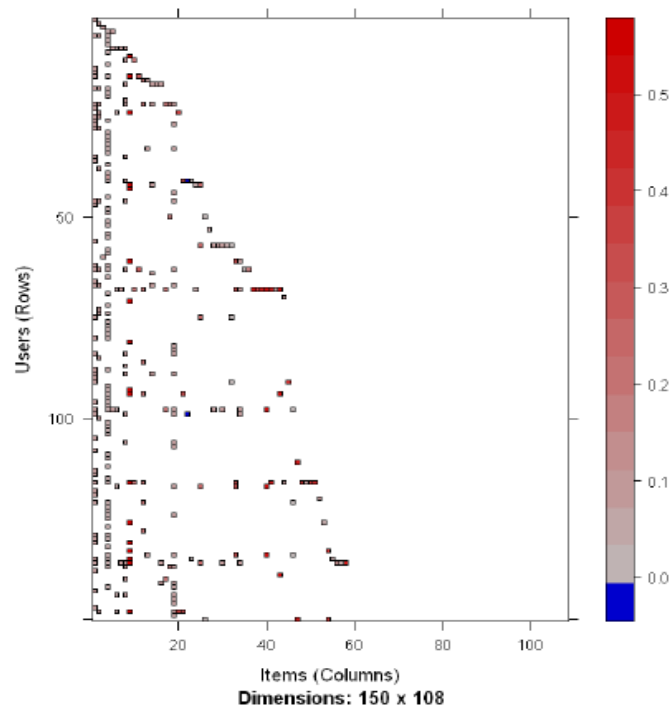


Figura 35: Imagem da base de fundos. Fonte: R.

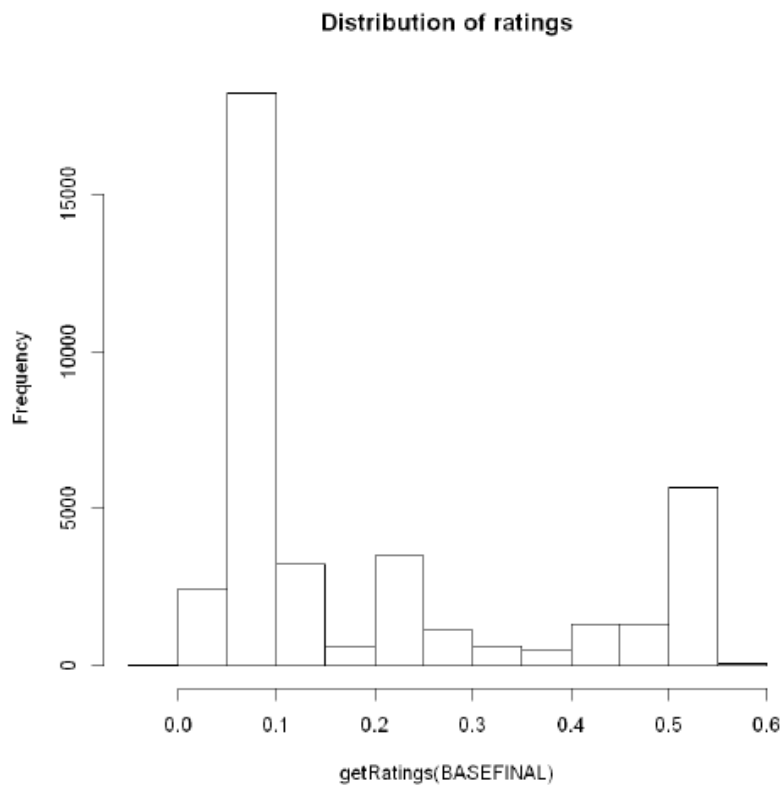


Figura 36: Histograma dos ratings na base de fundos. Fonte: R.

Com isso, finaliza-se a parte de descrição inicial das bases e passa-se para a aplicação dos métodos de sistemas de recomendação em si, no próximo subcapítulo. O

desenvolvimento detalhado dos códigos em R podem ser encontrados nos apêndices F e G do presente trabalho.

7.2. APLICAÇÃO DOS MÉTODOS DE RECOMENDAÇÃO E DESEMPENHO

Após a descrição da preparação dos inputs e da análise inicial das informações inseridas no software R seguiu-se para a aplicação dos métodos de sistemas de recomendação descritos no apêndice E que o pacote *recommenderlab* possui. A linha seguida nesta aplicação foi similar à da aplicação descrita no capítulo 5, da Universidade de Solapur na Índia, Sayyed et al. (2013), na qual se uma companhia é comprada por dois usuários diferentes no mesmo momento e está com retornos positivos, o sistema recomendaria tal companhia a um terceiro usuário prevendo que ele também teria retornos positivos no futuro.

Nesse sentido, utilizou-se a coluna de retornos calculados como referência de ratings implícitos associados aos usuários que transacionaram tais itens das bases, tanto a de ações quanto a de fundos, no período de 252 dias.

Para a base de ações, inicialmente, criou-se um esquema de avaliação, dividindo a base em treino e teste, definindo 90% para treino e 10% para teste. Definiu-se também um *goodrating* = 0.06, a fim de representar o CDI como uma referência. Os dados transacionais foram coletados de um espaço no tempo em que o CDI estava em torno de 6% a.a.

Inicialmente foi aplicado o método *recommender* em conjunto com *predict* a fim de demonstrar um método de recomendação personalizado de filtro colaborativo baseado em vizinhança com foco nos usuários (UBCF), conforme demonstrado abaixo.

```
e2 <- evaluationScheme(BASEFINAL, method = "split", train = 0.9, given=1, goodRating=0.06)
```

```
r <- Recommender(getData(e2,"train"),"UBCF")
```

```
p <- predict(r, getData(e2,"known"), type="ratings")
```

```
p
```

4075 x 254 rating matrix of class 'realRatingMatrix' with 771548 ratings.

```
calcPredictionAccuracy(p, getData(e2,"unknown"),given=1)
```

RMSE

2.50000293513906

MSE

6.2500146757039

MAE

1.05183131453812

Figura 37: Criando um sistema de recomendação de vizinhança focado no usuário e calculando as medidas de desempenho para a base de ações. Usando o *predict* para gerar ratings. Fonte: R.

```
p <- predict(r,getData(e2,"known"), type="topNList")
```

```
p
```

```
calcPredictionAccuracy(p, getData(e2,"unknown"),given=1, goodRating = 0.06)
```

Recommendations as 'topNList' with n = 10 for 4075 users.

TP

0.296687116564417

FP

7.19533742331288

FN

4.08196319018405

TN

241.426012269939

precision

0.0396003930560105

recall

0.0464732248601477

TPR

0.0464732248601477

FPR

0.0290424255734358

Figura 38: Criando um sistema de recomendação de vizinhança focado no usuário e calculando as medidas de desempenho para a base de ações. Usando o *predict* para gerar top-N recomendações. Fonte: R.

Posteriormente, foi realizada a aplicação utilizando a fatoração matricial de Simon Funk para os dados da base de ações.

```
train <- as(BASEFINAL[1:39000], "matrix")
fsvd <- funkSVD(train, verbose=TRUE)
```

```
r <- tcrossprod(fsvd$U, fsvd$V)
RMSE(train, r)
```

```
0.137476245284589
```

```
test <- as(BASEFINAL[39001:40000], "matrix")
p <- predict(fsvd, test, verbose = TRUE)
RMSE(test, p)
```

```
0.173088985609953
```

Figura 39: Aplicando a fatoração matricial de Simon Funk na base de ações. Fonte: R.

Para a base de fundos, foi realizada praticamente a mesma análise, inicialmente criando um esquema de avaliação 90% treino e 10% teste, aplicando as funções *recommender* e *predict* com o método UBCF e, posteriormente a fatoração matricial de Simon Funk.

```
e2 <- evaluationScheme(BASEFINAL, method = "split", train = 0.9, given=1, goodRating=0.06)
```

```
r <- Recommender(getData(e2, "train"), "UBCF")
```

```
p <- predict(r, getData(e2, "known"), type="ratings")
```

```
p
```

```
2284 x 108 rating matrix of class 'realRatingMatrix' with 133732 ratings.
```

```
calcPredictionAccuracy(p, getData(e2, "unknown"), given=1)
```

```
RMSE
```

```
0.295360272559267
```

```
MSE
```

```
0.0872376906062843
```

```
MAE
```

```
0.181773492891103
```

Figura 40: Criando um sistema de recomendação UBCF para a base de fundos. Função predict usada para gerar ratings. Fonte: R.

```
p <- predict(r,getData(e2,"known"), type="topNList")
p
calcPredictionAccuracy(p, getData(e2,"unknown"),given=1, goodRating = 0.06)
```

Recommendations as 'topNList' with n = 10 for 2284 users.

```
TP
0.117775831873905
FP
5.37259194395797
FN
0.786777583187391
TN
100.722854640981
precision
0.021451355661882
recall
0.10205336286841
TPR
0.10205336286841
FPR
0.0506195985063815
```

Figura 41: Criando um sistema de recomendação UBCF para a base de fundos. Função predict usada para gerar top-N recomendações. Fonte: R.

```
train <- as(BASEFINAL[1:20000], "matrix")
fsvd <- funkSVD(train, verbose=TRUE)
```

```
r <- tcrossprod(fsvd$U, fsvd$V)
RMSE(train, r)
```

```
0.0324527840981732
```

```
test <- as(BASEFINAL[20001:22000], "matrix")
p <- predict(fsvd, test, verbose = TRUE)
RMSE(test, p)
```

```
0.0455433517576075
```

Figura 42: Aplicando a fatoração matricial de Simon Funk na base de ações. Fonte: R.

Foi possível observar que os resultados das medidas de desempenho se mostraram melhores na aplicação da base de fundos, menor e com observações mais concentradas, que na base de ações, mais esparsa e maior, para o método de Simon Funk. Conclui-se isso a partir do RMSE calculado para fundos de 4,55% na base de teste contra o de 17,31% na base de ações.

Para o UBCF observa-se que para a base de ações o RSME foi menor e a precisão maior que na aplicação da base de fundos. O código mais detalhado da aplicação pode ser encontrado no apêndice F do presente trabalho.

Neste capítulo, portanto, foi possível observar uma aplicação dos métodos descritos no capítulo 7, cujo embasamento teórico pode ser visto com mais detalhes no capítulo 3 desta dissertação, em bases transacionais de plataformas digitais de investimento, respeitando a privacidade das informações e utilizando apenas a integração dos dados entre usuário, item e ratings implícitos para gerar sistemas de recomendação e avaliar seus desempenhos.

No próximo capítulo, será feita a conclusão da dissertação como um todo e um resumo dos próximos desafios no campo.

8. CONSIDERAÇÕES FINAIS

O presente trabalho se propôs a realizar uma extensa revisão bibliográfica dos conhecimentos e aplicações acerca dos sistemas de recomendação, tanto no *e-commerce* quanto nos trabalhos que já relacionaram o tema finanças aos chamados algoritmos de *machine learning*.

Ao longo dos capítulos, é possível ver o quanto as plataformas digitais de investimentos evoluíram no Brasil nos últimos anos, investindo massivamente em tecnologia, que se tornou um diferencial no mercado, e o quanto seria importante ter uma ferramenta de recomendação que atendesse ao público do varejo, o qual geralmente não possui um atendimento diferenciado semelhante a clientes Alta Renda, por exemplo.

Neste estudo, foi possível entender melhor o conceito de algoritmos de *machine learning* e dos sistemas de recomendação, amplamente utilizados no *e-commerce* e aplicações em finanças. Foi possível perceber que existem empresas que utilizam tais ferramentas como um produto para otimizar os negócios de outras empresas, entre elas também bancos. Mas também se percebeu que existem várias ferramentas *open source* no mercado que possibilitam aplicações diversas de sistemas de recomendação, como é o caso do pacote recommenderlab do R, que foi utilizado nas aplicações do presente trabalho.

Os resultados gerados pelas aplicações deste estudo foram satisfatórios, mas este trabalho ainda deixa vários desafios pendentes para próximos passos. O objetivo foi abrir as portas para estudos futuros relacionados ao tema, que procurem solucionar algumas questões tais como:

- i. Como os ativos financeiros mudam de característica rapidamente de acordo com os acontecimentos diários voláteis do mercado, qual seria a periodicidade ideal para selecionar os dados de recomendação? No presente trabalho foi utilizado o período de um ano para analisar os contatos entre usuário e item e os retornos dos itens, mas alguns produtos como opções por exemplo não poderiam ser analisados desta forma sem que fossem considerados metadados na análise;
- ii. Seria possível juntar ativos de características distintas em uma base única sem perder a qualidade da análise? Como foi visto neste estudo, o termo “ativo financeiro” concentra uma série de produtos com características distintas, o que torna o problema muito mais

complexo do que a recomendação de livros, filmes e músicas, alvos do atual sucesso dos sistemas de recomendação. Neste estudo especificamente, foram analisadas separadamente uma base de ações e outra de fundos de investimento;

- iii. Em trabalhos futuros seria interessante incluir metadados na análise. No presente trabalho, os retornos foram utilizados como *ratings*, mas outros dados como volatilidade das ações no período, setor, aversão a risco do usuário e tamanho da empresa poderiam ser incluídos na análise de ações, bem como o tipo de fundo, o risco do fundo, a aversão ao risco do usuário, o montante mínimo de aplicação e a taxa de administração do fundo poderiam ser incluídas na análise de fundos, por exemplo. No entanto é preciso estudar o impacto de processamento que isso geraria.
- iv. Outra opção interessante seria utilizar o índice de Sharpe de cada observação ao invés de apenas o retorno como *rating*, resumindo a relação risco-retorno de cada observação da base de dados.

Com isso, é possível concluir que algoritmos de recomendação podem auxiliar muito nas recomendações de investimentos das plataformas digitais, mas ainda há um vasto caminho de estudo a ser percorrido para que isso seja colocado em prática no mercado, de fato.

REFERÊNCIAS

- ¹ SHAH, Lipi. GAUDANI, Hetal. BALANI, Prem. SURVEY ON RECOMMENDATION SYSTEM. International Journal of Computer Applications. Volume 137, 2016.
- ² ZIBRICZKY, Dávid. RECOMMENDER SYSTEMS MEET FINANCE: A LITERATURE REVIEW. Budapest University of Technology and Economics. Hungary. Department of Finance, 2016.
- ³ LINDEN, Greg. SMITH, Brent. YORK, Jeremy. AMAZON.COM RECOMMENDATIONS, ITEM-TO-ITEM COLLABORATIVE FILTERING. IEE Computer Society, 2003.
- ⁴ SAYYED, F.R. ARGIDDI, R.V. APTE, S.S. GENERATING RECOMMENDATIONS FOR STOCK MARKET USING COLLABORATIVE FILTERING. Department of CSE, WIT, Solapur University, India. International Journal of Computer Engineering & Science, 2013.
- ⁵ ZHANG, Li. ZHANG, Han. HAO, SuMin. AN EQUITY FUND RECOMMENDATION SYSTEM BY COMBINING TRANSFER LEARNING AND THE UTILITY FUNCTION OF THE PROSPECT THEORY. The Journal of Finance and Data Science 4 (2018) 223-233.
- ⁶ DA COSTA JR, Newton. STOCK SELECTION BASED ON CLUSTER ANALYSIS. Department of Economics, Federal University of Santa Catarina, 2005.
- ⁷ SANKAR, Prem. Vidharaj, R. KUMAR, Satheesh. TRUST BASED STOCK RECOMMENDATION SYSTEM – A SOCIAL NETWORK ANALYSIS APPROACH. Procedia Computer Science 46 (2015) 299-305.
- ⁸ ASOSHEH, Abbas. BAGHERPOUR, Sanaz. YAHYAPOUR, Nima. EXTENDED ACCEPTANCE MODELS FOR RECOMMENDER SYSTEM ADAPTION, CASE OF RETAIL AND BANKING SERVICE IN IRAN. WSEAS Transactions on Business and Economics, 2008.
- ⁹ CRAN.R. RECOSYSTEM: RECOMMENDER SYSTEM USING MATRIX FACTORIZATION. Disponível em: < <https://cran.r-project.org/web/packages/recosystem/index.html>>. Data de acesso: 29/10/2019.
- ¹⁰ CHIN, Wei-Sheng. ZHUANG, Yong. JUAN, Yu-Chin. LIN, Chih-Jen. A LEARNING-RATE SCHEDULE FOR STOCHASTIC GRADIENT METHODS TO MATRIX FACTORIZATION. Department of Computer Science. National Taiwan University Taipei, Taiwan, 2015.
- ¹¹ CHIN, Wei-Sheng. ZHUANG, Yong. JUAN, Yu-Chin. LIN, Chih-Jen. YUAN, Bo-Wen. YANG, Meng-Yuan. ZHUANG, Yong. A LIBRARY FOR PARALLEL MATRIX FACTORIZATION IN SHARED-MEMORY SYSTEMS. Journal of Machine Learning Research 17 (2016) 1-5.

- ¹² BODIE, Zvi. KANE, Alex. MARCUS, Alan J. FUNDAMENTOS DE INVESTIMENTOS. 9ª Edição. São Paulo: AMGH Editora Ltda, 2014.
- ¹³ FEBRABAN. PESQUISA FEBRABAN DE TECNOLOGIA BANCÁRIA 2019 (ANO-BASE 2018). Disponível em: < <https://portal.febraban.org.br/pagina/3106/48/pt-br/pesquisa>>. Data de acesso: 04/01/2020.
- ¹⁴ EXAME. CORRETORAS DIGITAIS DISPUTAM INVESTIDORES COM BANCOS TRADICIONAIS. Estadão Conteúdo, 2019. Disponível em: < <https://exame.abril.com.br/seu-dinheiro/corretoras-digitais-disputam-investidores-com-bancos-tradicionais/>>. Data de acesso: 30/10/2019.
- ¹⁵ EXAME. TOLEDO, Leticia. A CORRIDA DIGITAL DOS BANCOS. Estadão Conteúdo, 2019. Disponível em: < <https://exame.abril.com.br/seu-dinheiro/corretoras-digitais-disputam-investidores-com-bancos-tradicionais/>>. Data de acesso: 30/10/2019.
- ¹⁶ Bodie, Markus and Kane, Fundamentos de Investimentos, 9ª Edição, 2014. Revista de administração Mackenzie. CRITERIA OF PORTFOLIO FORMATION OF STOCKS THROUGH HIERARCHICAL CLUSTERS. São Paulo, 2010.
- ¹⁷ SCHAFER, J. Ben. KONSTAN, Joseph A. RIEDL, John. E-COMMERCE RECOMMENDATION APPLICATIONS. GroupLens Research Project. University of Minnesota, 2001.
- ¹⁸ YAHYAPOUR, Nima. DETERMINING FACTORS AFFECTING INTENTION TO ADOPT BANKING RECOMMENDER SYSTEM – CASE OF IRAN. Lulea University of Technology, Master Thesis. 2008.
- ¹⁹ CONSELHO ADMINISTRATIVO DE DEFESA ECONÔMICA. SUPERINTENDÊNCIA-GERAL DE ANÁLISE ANTITRUSTE 2. ATO DE CONCENTRAÇÃO Nº 08700.004431/2017-16. Parecer técnico nº 24. 2017.
- ²⁰ VALOR INTESTE. PARA ONDE CAMINHAM AS PLATAFORMAS DE INVESTIMENTOS? Disponível em <<https://valorinveste.globo.com/educacao-financeira/noticia/2019/07/25/para-onde-caminham-as-plataformas-de-investimentos.ghtml>>. Data de acesso: 29/01/2020.
- ²¹ BTG DIGITAL. ROBÔ ADVISOR: O QUE É, COMO FUNCIONA, VANTAGENS E DESVANTAGENS. Disponível em <<https://www.btgpactualdigital.com/blog/investimentos/robo-advisor-o-que-e-como-funciona-vantagens-e-desvantagens>>. Data de acesso: 09/02/2020.
- ²² GÉRON, Aurélien. MÃOS À OBRA: APRENDIZADO DE MÁQUINA COM SCIKIT-LEARN E TENSORFLOW. EUA. 2017.
- ²³ JENSON, Graham. LIST OF RECOMMENDER SYSTEMS. Disponível em <https://github.com/grahamjenson/list_of_recommender_systems>. Data de acesso: 11/04/2020.

- 24 HOW GOOGLE SEARCH WORKS. Disponível em <<https://www.google.com/search/howsearchworks/>>. Data de acesso: 25/04/2020.
- 25 FILHO AZEVEDO, Adriano. INTRODUÇÃO AO ALGORITMO PAGERANK DO GOOGLE COM O R: UMA APLICAÇÃO DE AUTOVALORES/AUTOVETORES E CADEIAS DE MARKOV. Disponível em <<https://rpubs.com/adriano/PageRank>>. Data de acesso: 26/04/2020.
- 26 METHOD FOR NODE RANKING IN A LINKED DATABASE. Disponível em: <<http://patft.uspto.gov/netacgi/nph-Parser?Sect1=PTO1&Sect2=HITOFF&d=PALL&p=1&u=%2Fnetacgi%2FPTO%2Fsrchnu m.htm&r=1&f=G&l=50&s1=6285999.PN.&OS=PN/6285999&RS=PN/6285999>>. Data de acesso: 26/04/2020.
- 27 EPISERVER CUSTOMER-CENTRIC DIGITAL EXPERIENCE PLATFORM TM. Disponível em <<https://www.episerver.com/>>. Data de acesso: 01/05/2020.
- 28 EPISERVER BUYS PEERIUS TO BOOST AUTONOMOUS PERSONALIZATION Disponível em <<https://www.cms-connected.com/News-Archive/August-2016/Episerver-Buys-Peerius-to-Boost-Autonomous-Persona>>. Data de acesso: 01/05/2020.
- 29 EPISERVER ACQUIRES BUSINESS OF E-COMMERCE SOLUTION PROVIDER MEDIACHASE Disponível em <<https://www.globenewswire.com/news-release/2012/03/01/1004550/0/en/EPiServer-Acquires-Business-of-e-Commerce-Solution-Provider-Mediachase.html>>. Data de acesso: 01/05/2020.
- 30 STRANDS RETAIL COMPANY Disponível em <<http://retail.strands.com/company/>>. Data de acesso: 01/05/2020.
- 31 SLISYSTEMS OUR STORY Disponível em <<https://www.sli-systems.com/about/our-story>>. Data de acesso: 01/05/2020.
- 32 DATAPROC Disponível em <<https://cloud.google.com/dataproc?hl=pt-br>>. Data de acesso: 01/05/2020.
- 33 APACHE PROJECTS DIRECTORY Disponível em <<https://projects.apache.org/>>. Data de acesso: 01/05/2020.
- 34 APACHE PROJECTS LIST Disponível em <<http://www.apache.org/index.html#projects-list>>. Data de acesso: 01/05/2020.
- 35 PARALLELDOTS Disponível em <<https://www.paralleldots.com/>>. Data de acesso: 01/05/2020.
- 36 AWS MACHINE LEARNING Disponível em <<https://aws.amazon.com/pt/machine-learning/>>. Data de acesso: 01/05/2020.
- 37 DRESSIPI. Disponível em <<https://dressipi.com/>>. Data de acesso: 01/05/2020.

- ³⁸ LEARN HOW TO OPERATIONALIZE AI IN YOUR BUSINESS. Disponível em <<https://www.ibm.com/watson>>. Data de acesso: 01/05/2020.
- ³⁹ RECOMBEE DOCS. Disponível em <<https://docs.recombee.com/>>. Data de acesso: 01/05/2020.
- ⁴⁰ SEGMENTIFY. Disponível em <<https://www.segmentify.com/solutions/>>. Data de acesso: 01/05/2020.
- ⁴¹ MR. DLIB. Disponível em <<http://mr-dlib.org/>>. Data de acesso: 01/05/2020.
- ⁴² UNIVERSAL RECOMMENDER. Disponível em <<https://actionml.com/universal-recommender>>. Data de acesso: 02/05/2020.
- ⁴³ BUILDING CORRELATED CROSS-OCCURENCE (CCO) RECOMMENDERS WITH THE MAHOUT CLI. Disponível em <<http://mahout.apache.org/users/recommender/intro-cooccurrence-spark.html>>. Data de acesso: 02/05/2020.
- ⁴⁴ QUICK START – RECOMMENDATION ENGINE TEMPLATE. Disponível em <<http://predictionio.apache.org/templates/recommendation/quickstart/>>. Data de acesso: 02/05/2020.
- ⁴⁵ RECOMMENDATIONRACCOON. Disponível em <<https://www.npmjs.com/package/raccoon>>. Data de acesso: 02/05/2020.
- ⁴⁶ EASYREC RECOMMENDATION ENGINE. Disponível em <<http://easyrec.org/recommendation-engine>>. Data de acesso: 02/05/2020.
- ⁴⁷ MAHOUT RECOMMENDER OVERVIEW. Disponível em <<http://mahout.apache.org/docs/latest/algorithms/recommenders/>>. Data de acesso: 02/05/2020.
- ⁴⁸ LENSKIT. Disponível em <<https://lkpy.readthedocs.io/en/stable/algorithms.html#basic-algorithms>>. Data de acesso: 02/05/2020.
- ⁴⁹ ORYX PROJECT. Disponível em <<https://github.com/OryxProject/oryx>>. Data de acesso: 02/05/2020.
- ⁵⁰ PREDICTOR. Disponível em <<https://github.com/Pathgather/predictor>>. Data de acesso: 02/05/2020.
- ⁵¹ SURPRISE A PYTHON SCIKIT FOR RECOMMENDER SYSTEMS. Disponível em <<http://surpriselib.com/>>. Data de acesso: 02/05/2020.
- ⁵² LIGHTFM. Disponível em <<https://github.com/lyst/lightfm>>. Data de acesso: 02/05/2020.
- ⁵³ REXY. Disponível em <<https://github.com/kasramvd/Rexy>>. Data de acesso: 02/05/2020.

- 54 QMF – A MATRIX FACTORIZATION LIBRARY. Disponível em <<https://github.com/quora/qmf>>. Data de acesso: 02/05/2020.
- 55 TENSORREC. Disponível em <<https://github.com/jfkirk/tensorrec>>. Data de acesso: 02/05/2020.
- 56 HERMES. Disponível em <<https://github.com/lab41/hermes>>. Data de acesso: 02/05/2020.
- 57 SPOTLIGHT. Disponível em <<https://maciejkula.github.io/spotlight/interactions.html>>. Data de acesso: 02/05/2020.
- 58 CASE RECOMMENDER – A PYTHON FRAMEWORK FOR RECSYS. Disponível em <<https://github.com/caserec/CaseRecommender>>. Data de acesso: 02/05/2020.
- 59 GORSE: GO RECOMMENDER SYSTEM ENGINE. Disponível em <<https://github.com/zhenghaoz/gorse>>. Data de acesso: 02/05/2020.
- 60 HAHSLER, Michael. RECOMMENDERLAB: A FRAMEWORK FOR DEVELOPING AND TESTING RECOMMENDATION ALGORITHMS. Disponível em <<https://cran.r-project.org/web/packages/recommenderlab/index.html>>. Data de acesso: 02/05/2020.
- 61 HAHSLER, Michael. PACKAGE ‘RECOMMENDERLAB’. Disponível em <<https://cran.r-project.org/web/packages/recommenderlab/index.html>>. Data de acesso: 02/05/2020.
- 62 HISTÓRICO PESSOAS FÍSICAS. Disponível em <http://www.b3.com.br/pt_br/market-data-e-indices/servicos-de-dados/market-data/consultas/mercado-a-vista/historico-pessoas-fisicas/>. Data de acesso: 17/06/2020.
- 63 FIGUEIREDO MELO E SOUZA, Bruno. MODELOS DE FATORAÇÃO MATRICIAL PARA RECOMENDAÇÃO DE VÍDEOS. Pontifícia Universidade Católica do Rio de Janeiro, 2011.
- 64 SARWAR, Badrul. KARYPIS, George. KONSTAN, Joseph. RIEDL, John. ANALYSIS OF RECOMMENDATIONALGORITHMS FOR E-COMMERCE. University of Minnesota.
- 65 NOWAKOWSKA, Anna. RECOMMENDER SYSTEMS FOR MASS CUSTOMIZATION OF FINANCIAL ADVICE. 3rd European COST Conference on Mathematics for Industry in Switzerland. Whinterthur, 2018. <https://www.zhaw.ch/storage/engineering/institute-zentren/iamp/sp_acss/Nowakowska_2018.09_AI_in_Industry_and_Finance_InCube_Recommenders_Talk.pdf>. Data de acesso: 20/01/2020.
- 66 CHIRKINA, Aleksandra. RANKOV, Boris. INVESTMENT RECOMMENDER: GET PRODUCTION-READY WITH ME. <https://www.incubegroup.com/blog/recommender-systems-for-investment-products/>>. Data de acesso: 22/08/2020.
- 67 GOLDSTEIN, Itay. JIANG, Wei. KAROLYI, G Andrew. TO FINTECH AND BEYOND. The Review of Financial studies, Volume 32, 2019. 1647-1661 p.

APÊNDICE A - OUTROS EXEMPLOS MAIS GERAIS DE TRABALHOS NO DOMÍNIO DE AÇÕES

No artigo da Universidade de Tecnologia e Economia de Budapeste alguns trabalhos são citados propondo uma interface em que os usuários inserem suas informações para orientar o sistema de acordo com suas preferências explicitamente:

- i. Liu e Lee implementaram uma solução que analisa e indica ações baseada nas preferências especificadas pelo usuário;
- ii. Yoo et al. propõem uma interface gráfica com o usuário, que calcula recomendações personalizadas de acordo com o indicador Convergência/Divergência da Média Móvel (MACD – *Moving Average Convergence Divergence*) e interações do usuário;
- iii. Seo et al. introduz uma ferramenta de investimentos que coleta informações sobre ações e recomenda com base naquilo que o investidor já possui em sua carteira pessoal;
- iv. Chalidabhongse e Kaensar utilizam um indicador técnico estocástico nos retornos das ações, interações com os usuários e preferências explícitas para recomendar.

Alguns trabalhos, no entanto, consideram que as características do usuário e suas transações encontram-se já disponíveis na base de dados. No artigo mencionado, são citados os seguintes trabalhos:

- v. Yujun et al. propõe um algoritmo baseado no fluxo de grandes ordens. Seu argumento é de que utilizar grandes ordens de ações de baixo valor reduz a demanda computacional por algoritmos avançados. Sua base de recomendação envolve ações que foram selecionadas por usuários similares;
- vi. Taghavi et al. propõe o modelo clássico de ranquear ações e combinar técnicas híbridas de recomendação;
- vii. Sayyed et al. aplica o algoritmo padrão de filtro colaborativo descrito mais detalhadamente no capítulo 5.

APÊNDICE B - OUTROS EXEMPLOS MAIS GERAIS DE TRABALHOS NO DOMÍNIO DE PORTFÓLIOS

Alguns trabalhos são citados pelo artigo da Universidade de Tecnologia e Economia de Budapeste no campo da seleção personalizada de portfólio:

- i. Mussto et al. usam o que usuários similares selecionaram como base aplicando várias técnicas de combinação, utilizando filtro colaborativo com base em vizinhança e métodos de alocação de ativos tais como taxa média e diversidade de resultados;
- ii. Garcia-Crespo et al. e Gonzalez-Carrasco et al. utilizam lógica fuzzy para transformar um conjunto de características do usuário (nível de educação, idade, renda, aversão a risco etc) e dos portfólios (risco de mercado, taxa de retorno, liquidez, etc) em uma matriz unificada bidimensional com características psicológicas e comportamentos sociais. Os portfólios são recomendados com base na distância entre o investidor e os modelos de portfólio.
- iii. Beraldi et al. apresentam um sistema de suporte a decisão onde um investidor define seus parâmetros (caixa inicial, período, tipos de ativos e moeda). A partir destes critérios, são geradas carteiras com base na maximização do trade-off entre valor final esperado, valor em risco (VaR – Value at Risk) e um parâmetro de aversão a risco.

APÊNDICE C - SISTEMAS DE RECOMENDAÇÃO COMO UM SERVIÇO (GITHUB)

- i. Peerius (Episerver): o Episerver é um software inovador de marketing digital e e-commerce que adquiriu a Peerius em 2016 para aprimorar seus esforços em criar uma experiência de plataforma de compreensão digital nas nuvens de dados das companhias nas quais trabalha. A companhia na verdade já investia em personalização baseada em inteligência artificial em sua plataforma desde 2011, adicionando valor agregado para seus clientes. Seus serviços incluem a gestão de conteúdo, no *Epicenter Content Cloud*, personalizando páginas de *websites* para facilitar a experiência do consumidor final no site do cliente com buscas guiadas, indexação do conteúdo e identificação de padrões comportamentais. Além disso, o *Epicenter Commerce Cloud* utiliza inteligência artificial para gerir catálogos de produtos, compras, dados dos consumidores e pagamentos em diversas regiões e canais, expandindo o universo de atuação.
- ii. Strands: companhia global fundada oficialmente em 2004 cujo *core business* consiste em comercializar soluções com *BigData*, *machine learning* e personalização para aprimorar instituições financeiras e oferecer uma experiência superior aos clientes finais de um negócio a partir dos multicanais digitais de seus clientes intermediários. Foi fundada pelo Dr. Francisco Martin e Dr. Marc Torrens a partir da junção de um grupo de cientistas da Universidade de Oregon (EUA), o Instituto Avançado de Pesquisa de Inteligência Artificial (Espanha) e o Instituto Federal Suíço de Tecnologia (Suíça) para pesquisar tecnologias de personalização e recomendação em 2003.
- iii. SLISystems: companhia fundada na Nova Zelândia em 2001, a companhia começou desenvolvendo um ramo de pesquisa na internet baseado em *machine learning* para melhorar continuamente a relevância dos resultados das buscas, tecnologia que começou a vender para empresas de varejo *online*. Atualmente sua ferramenta poderosa auxilia os sites dos clientes a melhorar experiência com buscas, navegação, recomendações, *merchandising* e *mobile*.
- iv. Google *Cloud Platform* e Dataproc: uma forma de incorporar os algoritmos de clusters da rede Apache ao serviço de nuvem do Google Cloud para

gerenciamento de dados. A Fundação Apache Software é uma organização sem fins lucrativos fundada nos EUA composta por desenvolvedores de software *open source* e mantida pelo apoio de empresas como Microsoft, HP, Huawei e IBM, além de doações. Foi criada em 1999 nos EUA por um grupo de pessoas, o chamado “Grupo Apache”. Entre 1995 e 1999 o servidor de internet HTTPD, criado pelo grupo Apache, tornou-se líder do mercado e ainda é, com mais de 65% dos sites na web mantidos por ele. Com o crescimento progressivo da web, os interesses econômicos começaram a crescer e começaram a surgir novos projetos em paralelo com várias linguagens e em várias categorias. São mais de 350 iniciativas *open source*.

- v. ParallelDots: startup iniciada em 2017 que implementa inteligência artificial para solucionar problemas de negócios e criar produtos que gerem valor real para as pessoas, fornecendo também serviços de consultoria. A empresa utiliza reconhecimento ótico de caracteres, detecção de objetos e algoritmos de *deep learning* para auxiliar empresas a identificar KPIs, tais como participação de seus produtos nas estantes de supermercados, *compliance* de etiqueta de preços, além de fornecer aplicativos de análise de textos de publicações para detectar sentimentos em relação a empresa (cliente) socialmente.
- vi. Amazon Web Service (AWS): já citada neste trabalho como um exemplo prático de referência, a Amazon fornece serviços de *machine learning* e inteligência artificial para outros negócios, comercializando a mesma tecnologia de aprendizado profundo que alimenta a Amazon.com com recomendações, previsões, análise de imagem e vídeo, análise de texto, tradução, transcrição, pesquisa empresarial, detecção de fraudes entre outros.
- vii. Azure Machine Learning (Microsoft): serviço de *machine learning* para empresas que constrói, treina e implanta modelos rapidamente.
- viii. Gravity R&D¹⁵ (Yusp): empresa que foca em ciência de dados desde 2006, pesquisando e desenvolvendo algoritmos de recomendação e sua aplicação em modelos de negócio. Foi construída por alguns dos ganhadores do prêmio Netflix de 2009 e possui grandes clientes, tais como Dailymotion.

¹⁵ R&D: *Research & Development*

- ix. Dressipi: empresa que utiliza aprendizado de máquina para auxiliar companhias de varejo relacionadas à moda a prever o que consumidores vão comprar e não devolver, otimizando o aproveitamento e dando a eles a melhor experiência que poderiam ter. O serviço inclui personalização da página da web, personalização de e-mails enviados aos clientes, notificações personalizadas para o cliente, entre outros.
- x. Sajari: companhia australiana que surgiu com o objetivo de criar um sistema de busca de alta performance com qualidades híbridas. A empresa provê algoritmos personalizados para maximizar as receitas de seus clientes enquanto promove melhores experiências ao consumidor final. Promove melhorias na busca de websites dos clientes intermediários, captura os interesses dos usuários, mostrando conteúdo relevante para o visitante com base em localização e perfil comportamental, além de buscar aumentar o engajamento das páginas.
- xi. IBM Watson: serviço de inteligência artificial pronto para empresas fornecido pela IBM, focado em oferecer uma vantagem competitiva aumentando a produtividade das equipes libertando-as de tarefas repetitivas e levando-as a focar em um trabalho de alto valor agregado. O serviço prepara os dados, constrói modelos e implementa em produção, com seus vários APIs.
- xii. Recombee: serviço de recomendação que pode ser usado em qualquer domínio onde haja um catálogo de produtos que interagem com usuários. Os usuários podem interagir de várias formas, visualizando o item, classificando-o, comprando-o, entre outras. A empresa disponibiliza SDKs¹⁶ para seus clientes em diversas linguagens.
- xiii. Segmentify: solução que busca melhorar conversões online e vendas a partir de recomendações personalizadas aos clientes, busca personalizada que promove o aumento do engajamento, envio de e-mails personalizados informando sobre quedas de preços de acordo com o histórico de compras de cada consumidor, notificações personalizadas, identificação de tendências em tempo real para produtos, combinar em tempo real

¹⁶ *Software Development Kit* (Kits de desenvolvimento de software)

comportamentos dos compradores e navegação em vários canais para promover uma experiência altamente personalizada.

- xiv.* Mr.DLib¹⁷: projeto *open source* sem fins lucrativos que promove recomendações de artigos acadêmicos, *papers* e notícias acadêmicas. Foi desenvolvida na Universidade da Califórnia.

¹⁷ *Machine Readable Digital Library*

APÊNDICE D - SISTEMAS DE RECOMENDAÇÃO *OPEN SOURCE* (GITHUB)

- i. Universal Recommender: leva em consideração tudo que se sabe a respeito do usuário para tornar as recomendações melhores, como ações dos usuários, perfis, contexto, metadados dos itens e tudo que possa melhorar a predição. É construído a partir de ocorrência cruzada correlacionada¹⁸ (Apache), um novo ramo multimodal de sistemas de recomendação que pode utilizar inputs de vários tipos de forma flexível. Será lançado no serviço Amazon (AWS) já citado anteriormente.
- ii. Apache PredictionIO: criado pela Fundação Apache, integrou o algoritmo de filtro colaborativo da Apache Spark MLlib como padrão, embora seja possível customizá-lo para fins específicos. Basicamente, os *inputs* necessários são o ID usuário e os itens a serem recomendados, além dos dois tipos de eventos permitidos, classificação do usuário ou aquisição. O *output* do modelo é um ranking de itens recomendados.
- iii. Raccoon: um sistema baseado em filtro colaborativo baseado em usuário, que utiliza o coeficiente de similaridade de Jaccard para determinar a similaridade entre usuários e os k vizinhos mais próximos para gerar recomendações. Aparentemente o projeto está atualmente abandonado. Utiliza Javascript (node.js¹⁹).
- iv. HapiGER Beta: código que utiliza o chamado GER²⁰, que seria um sistema de recomendação bom o suficiente, que busca eliminar a dificuldade tanto de implementar quanto de customizar um sistema de recomendação, sendo utilizável em escala, facilmente utilizável e fácil de integrar. Por padrão, utiliza eventos baseados em memória. Recomendações podem ser geradas passando o nome de um usuário ou de um item. Também utiliza JavaScript (node.js).
- v. EasyRec: as ações dos usuários são levadas ao sistema a partir do API²¹, tais como visualização, aquisição ou classificação de um item pelo usuário. As ações são guardadas na base de dados do sistema e os “analísadores” periodicamente

¹⁸ *Correlated Cross-Occurrence (CCO)*

¹⁹ JavaScript é uma linguagem leve mais conhecida como a linguagem das páginas da Web, mas também utilizada em ambientes sem browser, tais como node.js

²⁰ *Good Enough Recommendation System*

²¹ *Application Programming Interface*

analisam tais dados para identificar padrões e gerar recomendações. A linguagem utilizada é JavaScript.

- vi. Apache Mahout: mais uma criação da Fundação Apache, trata-se de um sistema de recomendação baseado em modelo, mais especificamente o multimodal, a revolução já citada anteriormente que surgiu em 2013 permitindo o processamento de todo tipo de dado que pudesse servir como indicador do gosto do usuário para aprimorar a recomendação. O algoritmo base é o mesmo já citado no Universal Recommender, ocorrência cruzada correlacionada. A aplicação coleta interações usuário/item em uma base de dados. A maneira mais fácil de fazer isso é alocar as interações dos usuários com produtos em um arquivo csv (user-id e item-id). A base vai conter as últimas n interações do usuário, o que vai formar parte da query para recomendações. A partir dessas informações será criada uma tabela de item e itens similares na forma csv. Quando for necessário gerar recomendação para um usuário específico, basta pegar as últimas interações do usuário na base histórica.
- vii. Seldon: uma solução paga que utiliza plataformas integradas para resolver problemas de *machine learning*. Segundo o Github, para sistemas de recomendação especificamente, utiliza linguagem Java baseado em tecnologias como o Apache Spark, por exemplo.
- viii. Lenskit: uma ferramenta na linguagem Python para experimentos de recomendação. Trata-se da sucessão do projeto Lenskit baseado em Java. Provê suporte para treinar, aplicar e avaliar sistemas de recomendação. As interfaces são baseadas nos padrões SciKit adaptados para estruturas do Pandas e permite várias abordagens como as básicas (predição usando usuário/item/*rating*, recomendação dos itens mais populares, top N recomendações, seleção de itens que não foram classificados como candidatos, baseado em memória), algoritmos de filtro colaborativo KNN (com foco no usuário ou no item) e fatoração de matriz.
- ix. Oryx 2: construído com base no Apache Spark e no Apache Kafka, além de outras aplicações em *machine learning* realiza recomendações com base em filtro colaborativo.

- x. RecDB: sistema de recomendação construído inteiramente dentro do PostgreSQL²², permite aos usuários montar algoritmos com base em filtro colaborativo com foco no usuário ou no item, além de SVD²³.
- xi. Crab: biblioteca de recomendação em Python que integra algoritmos de recomendação no mundo de pacotes Python (numpy, scipy, matplotlib), mas o repositório aparentemente foi abandonado.
- xii. Predictor: desenvolvido na linguagem Ruby/Redis usa o coeficiente de similaridade de Jaccard como padrão, tendo também a possibilidade de utilizar o coeficiente de Sorenson-Dice, para determinar similaridades entre itens.
- xiii. SurPRISE²⁴: é um scikit²⁵ do Python para construir e analisar sistemas de recomendação que lidam com dados de ratings explícitos. Entre os métodos disponíveis estão algoritmos básicos já descritos no Lenskit, métodos de vizinhança, fatoração de matriz, medidas de similaridade como o cosseno, por exemplo, entre outros. O pacote também possibilita medir a acurácia do modelo com indicadores como MAE e RMSE também já descritos nesse estudo.
- xiv. LightFM: trata-se de uma implementação em Python de alguns algoritmos de recomendação para bases com *feedbacks* explícitos ou implícitos. A solução torna possível incorporar metadados dos itens e usuários aos tradicionais métodos de fatoração de matriz representando cada usuário e item como um somatório das representações latentes de suas características, possibilitando que recomendações sejam generalizadas para novos itens via características dos itens e novos usuários via características dos usuários.
- xv. REXY (Rec-sy): sistema de recomendação em Python baseado em um conceito usuário-produto-tag. As abordagens incluem recomendações gerais como top produtos, recomendações baseadas em eventos (classificações) e novos produtos que podem ser considerados interessantes. Sua estrutura possui vários módulos administração (análise e visualização das características da base), principal (contém as operações principais que dão suporte aos outros módulos), e-mail (recomendações por e-mail), geral (recomendações não personalizadas), *logging*

²² Base de dados relacional *Open Source*

²³ *Singular value decomposition* – fatoração de matriz

²⁴ SurPRISE: *Simple Python Recommendation System Engine*

²⁵ Pacotes adicionais do Python desenvolvidos separadamente da principal distribuição Scipy. Todos são licenciados.

(guarda as exceções e erros do modelo), perfil (guarda os perfis de usuários, produtos etc) e busca.

- xvi. QMF: biblioteca em C++ para modelos de fatoração de matriz com *feedbacks* implícitos.
- xvii. TensorRec: trata-se de uma estrutura de recomendação no TensorFlow, plataforma de *machine learning*, com estrutura em Python. A solução consome dados do usuário, dos itens e as interações, utilizando essas três peças para aprender e ranquear recomendações. A aplicação não está mais recebendo atualizações.
- xviii. Hermes: trata-se de uma aplicação do Lab41²⁶ em Python que explora como escolher um sistema de recomendação para uma nova aplicação analisando a performance de múltiplos sistemas de recomendação em uma variedade de bases de dados.
- xix. Spotlight: estrutura implementada em Python que permite construir modelos de recomendação profundos ou rasos, de acordo com a necessidade. Permite abordagens de fatoração de matriz de ratings explícitos ou implícitos e modelos sequenciais, baseados em interações passadas dos usuários, por exemplo.
- xx. Recommenderlab: biblioteca em R para desenvolver e testar sistemas de recomendação. Permite várias abordagens e será melhor detalhada no próximo item deste estudo por ter sido escolhida como base para a aplicação no projeto de ciência de dados do último capítulo.
- xxi. CaseRecommender: trata-se de uma implementação em Python de alguns algoritmos de recomendação mais populares para *feedbacks* implícitos ou explícitos. Os algoritmos incluem várias abordagens, como vizinhança dos itens (KNN), vizinhança dos usuários (KNN), modelo de cluster, mais populares, aleatórios, baseado em conteúdo, fatoração de matriz, SVD entre outros.
- xxii. ProbQA: sistema de recomendação interativo em C++, que utiliza a abordagem bayesiana para aprender como um mapa das respostas às perguntas podem gerar as melhores recomendações para um determinado objetivo.
- xxiii. Gorse: trata-se de um sistema de recomendação off-line baseado em filtro colaborativo escrito na linguagem Go (criada pelo Google).

²⁶ Laboratório de pesquisa dos EUA em que a Inteligência Norte-americana, a academia e a indústria se unem para explorar o BigData

APÊNDICE E - DETALHANDO O RECOMMENDERLAB: MÉTODOS, PARÂMETROS E HIPERPARÂMETROS

Conforme abordado no capítulo 6 do presente trabalho, sistemas de recomendação são pedaços de *software* que aplicam técnicas estatísticas e de *machine learning* com o objetivo de solucionar problemas de recomendação em geral. Alguns deles consistem em um serviço para empresas, como uma espécie de consultoria que incorpora ciência de dados ao negócio de seus clientes, possibilitando que o consumidor final de seus itens encontre com maior facilidade aquilo que necessita em um mar de produtos. Afinal, nessa era na qual um concorrente encontra-se a apenas um clique de distância, é importante agregar valor aos serviços prestados, mesmo quando não se possui especialização técnica em sua equipe para isso. Outros deles consistem em pacotes extensivos de sistemas *open source*, de código aberto adaptável para várias finalidades, que contam com o apoio de comunidades de desenvolvedores que enriquecem ainda mais tais sistemas compartilhando suas experiências em aplicações de todo tipo. Na linha de pacotes adicionais de sistemas *open source* está o pacote recommenderlab do software R. Abaixo serão descritos seus métodos, parâmetros e hiperparâmetros.

RECOMMENDERLAB: MÉTODOS

Apesar de permitir a aplicação de mais de um método de sistemas de recomendação, pode-se dizer que o pacote recommenderlab foca em recomendações personalizadas baseadas em filtragem colaborativa, ou seja, a partir de uma lista de ratings, implícitos ou não, de produtos por usuário, é possível classificar produtos desconhecidos para um determinado usuário, além de permitir criar uma lista de top-N produtos para o mesmo.

Portanto, de acordo com o diagrama do capítulo 3, seria possível dizer que o pacote se concentra no seguinte ramo destacado abaixo:

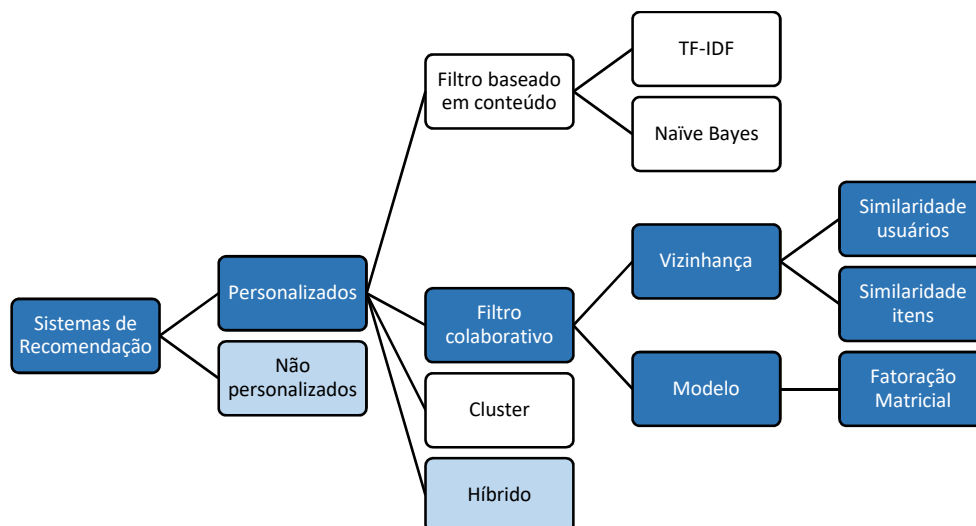


Figura 43: Ilustração dos métodos de recomendação nos quais o recommenderlab se concentra. Fonte: A autora.

O foco do pacote recommenderlab está na recomendação personalizada com base em filtro colaborativo, mas sua infraestrutura também permite realizar recomendações não personalizadas, com base em popularidade, além de possibilitar mesclar vários tipos de recomendação, gerando um sistema híbrido.

Inicialmente, é necessário importar uma matriz de ratings, a qual pode ser binária, com ratings que se resumem a 0 ou 1, ou real, com ratings que assumem qualquer outro valor real. O formato da base de importação é o de uma tabela com números representando usuários, itens e ratings, conforme demonstrado abaixo:

USUÁRIO	ITEM	RATING
1	1	5
1	2	3
2	5	1
...

Posteriormente, o sistema permite analisar os dados com funções que possibilitam contar o número de ratings da base, resumir os dados, gerar histogramas e imagens a partir da base de ratings importada a fim de analisa-la. No apêndice E do presente trabalho é possível conferir alguns testes com bases disponíveis para testes do próprio pacote do R, tal como a base de filmes do MovieLens²⁷ (base Movielense), a base de sites visitados por usuários da Microsoft (MSWeb) e a base de piadas de recomendação de piadas online Jester (Jester5k). No apêndice

²⁷ Comunidade virtual que recomenda filmes para usuários assistirem utilizando um sistema de recomendação personalizado com base em filtro colaborativo.

é possível perceber que cada uma das bases de exemplo possui uma estrutura diferente de ratings. As bases Jester e Movielense possuem ratings reais, enquanto a MSWeb se trata de uma base de ratings binária.

Na montagem do histograma da base Movielense é possível perceber o problema da cauda longa descrito no capítulo 5, no qual muitos usuários possuem poucos produtos.

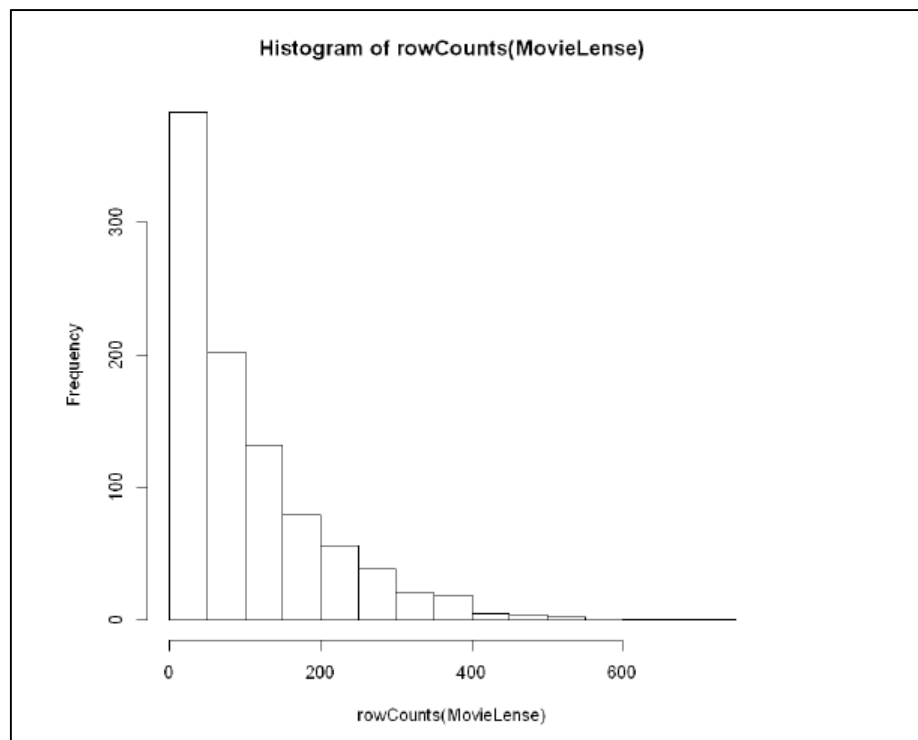


Figura 44: Histograma do número de ratings por usuário da base Movielense. Fonte: Hashler (2016).

A partir da base de ratings importada, o recommenderlab possibilita análises e, posteriormente, a aplicação de basicamente 4 métodos de recomendação, conforme resumido abaixo:

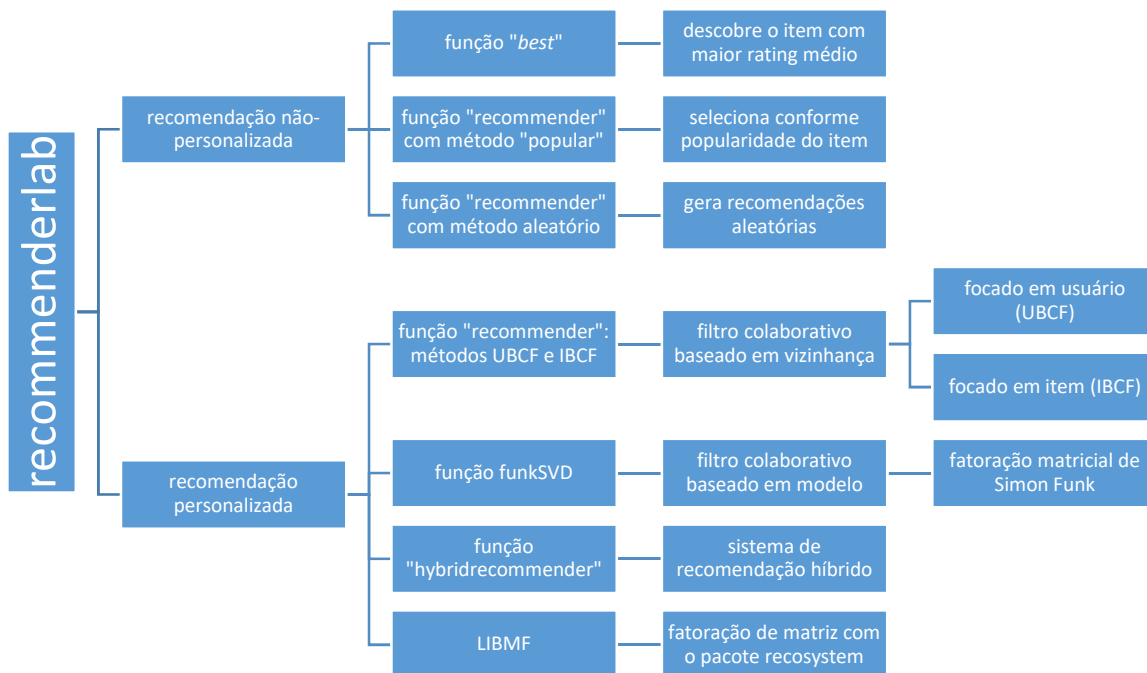


Figura 45: Resumo da infraestrutura de recomendação do pacote recommenderlab. Fonte: A autora.

A função mais abrangente do pacote, é a função ***recommender*** citada amplamente no diagrama acima, que permite criar sistemas com os mais diferentes métodos, desde recomendação aleatória ou popular até sistemas de recomendação baseados em fatoração matricial. Pelo que se observa nas demonstrações do pacote extensivo, tal função geralmente é utilizada associada a um esquema de avaliação que divide a base utilizada em base de treino e teste, função chamada ***evaluationscheme*** no sistema.

Essa divisão da base pode ser realizada de três maneiras basicamente:

- Aleatoriamente, de modo que o usuário define um percentual dos dados utilizados para treino e teste (*split*);
- Divisão em várias partes, por padrão do sistema são 10, podendo variar conforme input do usuário (k partes), de modo que em cada etapa do processo de treino $k-1$ partes são usadas para treino e as restantes são usadas para teste (*cross-validation*).
- Utilizando uma amostra com substituição a partir do número de usuários que há na base (*bootstrap*). Os dados que não estiverem na base de treino são utilizados para teste.

O ***evaluationscheme*** também permite que o usuário defina por exemplo qual rating é considerado bom pelo sistema de recomendação que está sendo criado. De modo que é

possível dizer ao sistema que se, por exemplo, a partir do *rating* = 3 os *ratings* são considerados bons para a recomendação.

Portanto, a função *evaluationcheme* define como será a divisão da base em treino e teste, a função *recommender* cria o sistema de recomendação, com o método de recomendação escolhido aplicando na base de treino e, para obter a lista de recomendações na base de teste, utiliza-se a função *predict*, que pode gerar uma lista de recomendações ou uma lista de ratings, conforme o que é definido pelo usuário na função. Por fim, para medir o desempenho do modelo, é utilizada a função *calcPredictionAccuracy*, com as medidas descritas no capítulo 3 deste trabalho.

Uma aplicação desse trio de funções é apresentada com mais detalhes no Apêndice E do presente trabalho, mas também é demonstrada resumidamente abaixo, na base de exemplo disponível no pacote, a MSWeb. Tal base possui dados da Microsoft, selecionando aleatoriamente 38.000 usuários e listando quais áreas da internet foram visitadas por eles em uma semana. Trata-se de uma base com ratings binários (0-1). Esse teste também pode ser encontrado com mais detalhes no próprio material descritivo do pacote, de Hahsler. Foi utilizado no presente trabalho a fim de melhor entender sobre o funcionamento do pacote para aplicação posterior no capítulo seguinte.

```
library(recommenderlab)
data("MSWeb")
MSWeb10 <- sample(MSWeb[rowCounts(MSWeb) > 10,],50)

e <- evaluationScheme(MSWeb10, method = "split", train=0.9,k=1,given=3)

r <- Recommender(getData(e,"train"),"UBCF")

p <- predict(r,getData(e,"known"), type="topNList",n=10)
p
calcPredictionAccuracy(p, getData(e,"unknown"),given=3)
```

Figura 46: Linhas de comando de exemplo no R para a aplicação do recommender na base MSWeb. Fonte: Hahsler (2016).

Nesse caso utiliza-se uma amostra da base para a aplicação, conforme é possível perceber a partir das linhas de comando. Abaixo, pode-se conferir os resultados que aparecem:

```

50 x 285 rating matrix of class 'binaryRatingMatrix' with 661 ratings.

Evaluation scheme with 3 items given
Method: 'split' with 1 run(s).
Training set proportion: 0.900
Good ratings: NA
Data set: 50 x 285 rating matrix of class 'binaryRatingMatrix' with 661 ra
tings.

Recommendations as 'topNList' with n = 10 for 5 users.

TP
5
FP
5
FN
5.4
TN
266.6
precision
0.5
recall
0.475
TPR
0.475
FPR
0.0183779399837794

```

Figura 47: Resultados da aplicação do trio de funções na base MSWeb. Fonte: Hahsler.

Nesse caso, foi utilizado o método de filtragem colaborativa baseada em vizinhança com foco no usuário. Como padrão, o sistema utiliza a similaridade dos cossenos para medir a proximidade dos usuários, já descrito e detalhado no capítulo 3 deste trabalho.

Alguns métodos interessantes disponíveis no pacote são funções que permitem eliminar o viés de ratings das bases, por exemplo, subtraindo a média de ratings em toda a base (função *normalize*). Além disso, é possível migrar de uma base com ratings reais para uma base binária conforme a necessidade, a partir da função *binarize*.

A função *recommender* também permite mesclar métodos gerando um sistema de recomendação híbrido, utilizando outra função, a chamada *hybridrecommender* com vários *recommenders* embutidos e pesos distintos. Novamente é preciso fazer a combinação com a função *predict*, conforme é possível observar abaixo.

```

recom <- HybridRecommender(Recommender(train, method="POPULAR"),
                           Recommender(train, method="RANDOM"),
                           Recommender(train, method="RERECOMEND"),
                           weights = c(.6,.1,.3))

getModel(recom)

as(predict(recom, test), "list")

```

Figura 48: Gerando um sistema de recomendação híbrido. Fonte: Hahsler (2016).

Além das funções já citadas e exemplificadas, que permitem aplicações de filtro colaborativo baseado em vizinhança, há uma demonstração importante baseada em modelo no material descritivo do pacote que é a de fatoração matricial de Simon Funk, também citada no capítulo 3. Nesse caso, a matriz de input primeiramente é transformada em uma matriz no seguinte formato:

Tabela 5: Exemplo de matriz construída a partir do input

Usuário/Item	1	2	3	4	5	...
1	5	2	7	N/A	N/A	...
2	N/A	3	N/A	N/A	N/A	...
...

Posteriormente, a função decompõe tal matriz gerada em duas matrizes: U, de usuários, e V, de itens. Tal decomposição é realizada utilizando otimização do gradiente descendente, o que ficou popularizado em 2006 com a aplicação de Simon Funk no prêmio Netflix, realizado a fim de obter um melhor sistema de recomendação de filtragem colaborativa para a plataforma de *streaming*. A companhia ofereceu um prêmio de US\$ 1.000.000 para quem elaborasse o melhor algoritmo.

Por fim, as duas matrizes são multiplicadas, preenchendo os valores vazios que representam itens não classificados pelos usuários e o desempenho do modelo é medido. Com a função *predict*, é possível aplicar o modelo na base de teste e novamente medir o erro gerado, comparando-o ao calculado a partir da base de treino. No material de Hahsler (2016) tal aplicação é feita na base Jester5k, as linhas de código podem ser observadas abaixo.

```

data(Jester5k)
train <- as(Jester5k[1:100], "matrix")
fsvd <- funkSVD(train, verbose=TRUE)

r <- tcrossprod(fsfd$U, fsfd$V)
RMSE(train, r)

test <- as(Jester5k[101:105], "matrix")
p <- predict(fsfd, test, verbose = TRUE)
RMSE(test, p)

```

Figura 49: Aplicando a função de fatoração matricial de Simon Funk. Fonte: Hahsler (2016).

Para todos os métodos do pacote, o cálculo do desempenho ao final da aplicação é um fator importante. Geralmente utiliza-se a função *calcPredictionAccuracy* ou funções mais diretas de erro médio, como a *RMSE*.

De todo modo, a lógica da avaliação realizada pelo sistema basicamente compara o teste realizado a partir da base de teste com a realidade observada utilizando o seguinte racional:

Tabela 6: Matriz de confusão

Realidade/Previsão	Negativo	Positivo
Negativo	a	b
Positivo	c	d

Algumas medidas de desempenho, como as já citadas anteriormente neste estudo, que o pacote utiliza para avaliar seus modelos quanto aos ratings previstos são as citadas abaixo:

- i. Desvio médio da média (MAE – *Mean Average Error*), onde K representa todos os pares usuário-item, \hat{r}_{ij} o rating previsto pelo modelo e r_{ij} o rating conhecido.

$$MAE = \frac{1}{|K|} \sum_{(i,j) \in K} |r_{ij} - \hat{r}_{ij}|$$

- ii. Raiz quadrada do erro médio (RSME – *Root Mean Square Error*), onde K representa todos os pares usuário-item, \hat{r}_{ij} o rating previsto pelo modelo e r_{ij} o rating conhecido. Essa medida penaliza erros de maior dimensão mais que a MAE e é mais útil quando erros pequenos não são tão importantes.

$$RSME = \sqrt{\frac{\sum_{(i,j) \in K} (r_{ij} - \hat{r}_{ij})^2}{|K|}}$$

Além disso, o modelo utiliza medidas de desempenho para medir as top-N recomendações geradas:

- iii. Acurácia, que em termos da tabela de confusão demonstrada mais acima pode ser representada conforme abaixo.

$$Acurácia = \frac{\text{recomendações corretas}}{\text{total de recomendações possíveis}} = \frac{a+d}{a+b+c+d}$$

- iv. MAE, que representado a partir da tabela de confusão ficaria conforme abaixo:

$$MAE = \frac{b+c}{a+b+c+d}$$

- v. Precisão, também representada pela tabela de confusão, ficaria conforme abaixo:

$$\text{Precisão} = \frac{\text{itens corretamente recomendados}}{\text{total de itens recomendados}} = \frac{d}{b+d}$$

$$\text{Recall} = \frac{\text{itens corretamente recomendados}}{\text{total de recomendações úteis}} = \frac{d}{c+d}$$

Neste capítulo, portanto, foram apresentados os métodos disponíveis no pacote recommenderlab e exemplificadas algumas aplicações possíveis a fim de demonstrar o funcionamento. No apêndice E deste estudo a demonstração das bases de exemplo de Hahsler são colocadas de forma mais completa a fim de exemplificar melhor. Nos próximos subcapítulos do capítulo 7 serão demonstrados parâmetros e hiperparâmetros envolvidos nos métodos do pacote.

RECOMMENDERLAB: PARÂMETROS

Conforme explicado no início do capítulo 3 do presente trabalho, sistemas de recomendação consistem em sistemas de aprendizado de máquina em ciências de dados. Entretanto, não foi apresentado no capítulo 3 que sistemas de aprendizado de máquina possuem variáveis características que fazem parte de seus métodos: parâmetros e hiperparâmetros.

Os chamados parâmetros seriam variáveis de um modelo que são estimadas a partir da otimização de alguma função, a partir do treinamento deste. Já os chamados hiperparâmetros são variáveis definidas pelo usuário, que não mudam de valor ao longo do processo de treinamento do modelo. Estes últimos geralmente são determinados por algum processo de validação que antecede a aplicação do método em si.

Dadas tais definições, quais seriam, por exemplo, os parâmetros envolvidos nos métodos do pacote recommenderlab? Voltando ao diagrama resumo do subcapítulo anterior é possível resumir a questão:

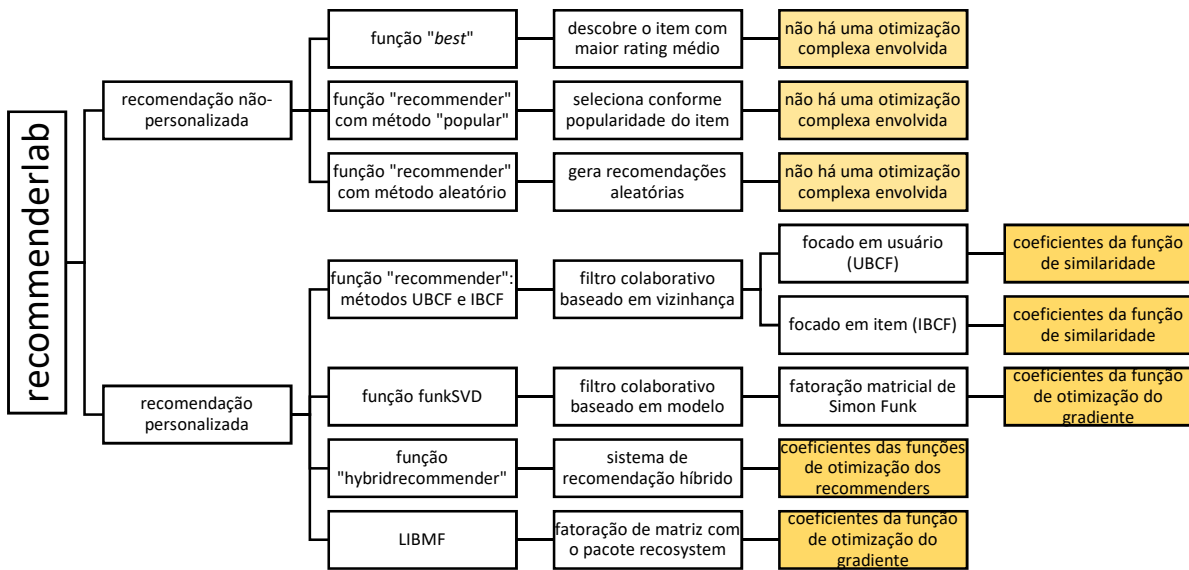


Figura 50: Parâmetros dos métodos do pacote *recommenderlab*

destacados em amarelo. Fonte: a autora.

Portanto, conforme resumido acima, os parâmetros envolvidos nas funções de otimização dos métodos do pacote *recommenderlab*, resumidamente seriam os coeficientes das funções de similaridade dos sistemas de filtro colaborativo baseados em vizinhança com foco em usuário ou item, e os coeficientes da função de otimização do gradiente no caso dos filtros colaborativos baseados em modelo de fatoração matricial. No caso do sistema híbrido, os parâmetros seriam definidos a partir dos métodos embutidos em sua combinação.

Portanto, voltando ao que foi descrito no capítulo 3, os parâmetros seriam os coeficientes que aparecem nas funções abaixo e que sofrem mudanças conforme o processamento ocorre na base consumida.

No caso dos modelos de vizinhança, por exemplo, seriam os coeficientes w , tanto na função voltada para usuários quanto na de itens.

$$w_{a,u} = \frac{\sum_{i \in I} (r_{a,i} - \bar{r}_a)(r_{u,i} - \bar{r}_u)}{\sqrt{\sum_{i \in I} (r_{a,i} - \bar{r}_a)^2 \sum_{i \in I} (r_{u,i} - \bar{r}_u)^2}}$$

$$w_{i,j} = \frac{\sum_{u \in U} (r_{u,i} - \bar{r}_i)(r_{u,j} - \bar{r}_j)}{\sqrt{\sum_{u \in U} (r_{u,i} - \bar{r}_i)^2 \sum_{u \in U} (r_{u,j} - \bar{r}_j)^2}}$$

No caso dos modelos de fatoração matricial, seriam os parâmetros de regularização da função de otimização e os parâmetros que representam os passos dentro das funções de cada gradiente (as gregas λ e γ , por exemplo).

$$\begin{aligned} \min_{q^*, p^*} \sum_{(u,i) \in k} (r_{u,i} - q_i^T p_u)^2 + \lambda (\|q_i\|^2 + \|p_u\|^2) \\ q_i \leftarrow q_i + \gamma (e_{ui} p_u - \lambda q_i) \\ p_u \leftarrow p_u + \gamma (e_{ui} q_i - \lambda p_u) \end{aligned}$$

Portanto, nesse subcapítulo foi possível compreender o que são parâmetros e hiperparâmetros de um sistema de aprendizado de máquina e descobrir quais são os parâmetros envolvidos nos métodos do pacote ***recommenderlab*** especificamente. Agora, vamos aos hiperparâmetros envolvidos nos métodos do pacote.

RECOMMENDERLAB: HIPERPARÂMETROS

Os hiperparâmetros envolvidos nos métodos do pacote ***recommenderlab*** estão particularmente relacionados à divisão da base de dados em treino e teste a partir do ***evaluationscheme***. O usuário possui a opção de dividir a base conforme descrito no subcapítulo de métodos e dessa divisão depende todo o processo de treinamento. Portanto, a partir do momento em que ele escolhe se divide a base entre treino e teste por percentual e define tal percentual ou que vai dividir a base em k partes para o treino e define essa variável k , ele está definindo um hiperparâmetro, uma variável que não muda com o processo de otimização e depende do usuário. Isso vale para todos os métodos em que há um processo de otimização envolvido, como os de filtro colaborativo baseado em vizinhança e os métodos de filtro colaborativo baseados em modelo, por exemplo. A definição de um ***goodrating***, um rating considerado bom pelo modelo a partir do usuário também seria um hiperparâmetro envolvido, também definido no ***evaluationscheme***.

No caso do sistema de recomendação híbrido, seriam os pesos definidos pelo usuário para cada ***recommender*** dentro do ***hybridrecommender***.

Nesse apêndice, portanto, foram resumidos os métodos, parâmetros e hiperparâmetros do pacote ***recommenderlab*** do software R. No apêndice F deste trabalho serão demonstrados exemplos disponíveis no pacote para testagem segundo Hahsler (2016).

APÊNDICE F - EXEMPLOS DE APLICAÇÃO DO *RECOMMENDERLAB* DE HAHSLER (2016)

A fim de melhor compreender o funcionamento do pacote extensivo *recommenderlab*, é possível explorar algumas bases de exemplo pré-existentis disponíveis no *software* R para testes. As bases disponíveis para testes no *software* são antigas e estão listadas abaixo:

- ***Jester5k***, com dados de 5.000 usuários que classificaram piadas no Jester Online entre abr/99 e mai/03. Trata-se de uma base com *ratings* reais, ou seja, não binários.
- ***MSWeb***, com dados da Microsoft de 38.000 usuários anônimos e as áreas da Web visitadas por eles em uma semana de fev/98. Trata-se de uma base com *ratings* binários.
- ***MovieLense***, com dados coletados entre set/97 e abr/98. São 100.000 observações de 943 usuários que classificaram 1664 filmes. Cada qual classificou um número diferente de filmes, mas o total de itens disponível na base é 1664. A base ***MeveLenseMeta*** contém metadados que podem acrescentar informações a análise.

Hahsler no arquivo descritivo do pacote exemplifica a utilização dos métodos envolvidos nos sistemas de recomendação do ***recommenderlab*** utilizando essas três bases. Tais exemplos serão também registrados aqui nesse apêndice a nível de demonstração. Tais funções foram utilizadas como referência para as aplicações práticas do presente trabalho.

Utilizando a ***Jester5k***, descrevendo a base e demonstrando o ***recommender*** em conjunto com o ***predict***:

```
library(recommenderlab)
data(Jester5k)
```

```
nratings(Jester5k)
```

```
362106
```

```
summary(rowCounts(Jester5k))
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
36.00	53.00	72.00	72.42	100.00	100.00

```
str(Jester5k)
```

```
Formal class 'realRatingMatrix' [package "recommenderlab"] with 2 slots
..@ data      :Formal class 'dgCMatrix' [package "Matrix"] with 6 slots
.. .. ..@ i      : int [1:362106] 0 1 2 3 4 5 6 7 8 9 ...
.. .. ..@ p      : int [1:101] 0 3314 6962 10300 13442 18440 22513 2751
2 32512 35685 ...
.. .. ..@ Dim     : int [1:2] 5000 100
.. .. ..@ Dimnames:List of 2
.. .. .. ..$ : chr [1:5000] "u2841" "u15547" "u15221" "u15573" ...
.. .. .. ..$ : chr [1:100] "j1" "j2" "j3" "j4" ...
.. .. ..@ x       : num [1:362106] 7.91 -3.2 -1.7 -7.38 0.1 0.83 2.91 -
2.77 -3.35 -1.99 ...
.. .. ..@ factors : list()
..@ normalize: NULL
```

Criando um *evaluationscheme* para dividir a base entre treino (90%) e teste (10%):

```
e <- evaluationScheme(Jester5k[1:500,], method = "split", train = 0.9, k=1, given=15)
```

Criando um sistema de recomendação baseado no usuário usando o *recommender* com o método UBCF, gera-se o código abaixo. Como padrão, ele utiliza o cosseno para medir a similaridade.

```
r <- Recommender(getData(e,"train"),"UBCF")
```

Utilizando o *predict* para obter ratings:

```
p <- predict(r, getData(e,"known"), type="ratings")
p
```

50 x 100 rating matrix of class 'realRatingMatrix' with 4250 ratings.

Computando o erro médio geral da previsão e o erro médio por usuário. Avaliando a lista *Top-N* gerada.

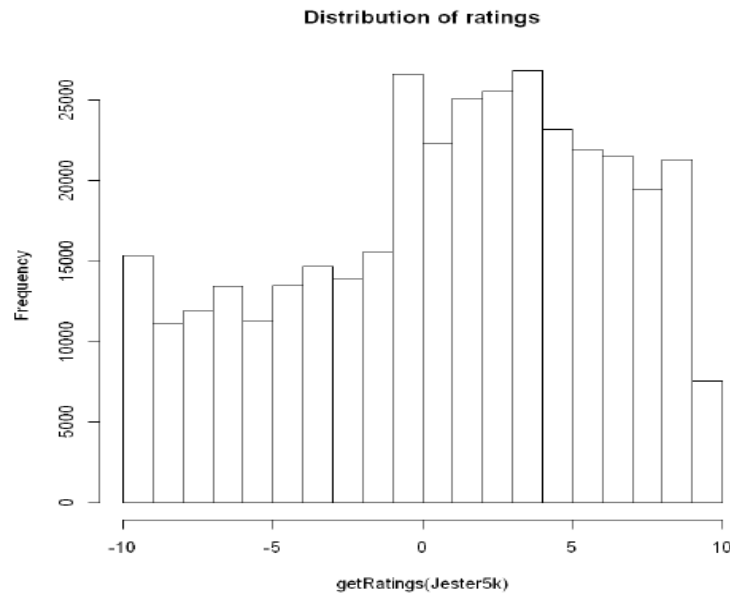
```
p <- predict(r,getData(e,"known"), type="topNList")
p
calcPredictionAccuracy(p, getData(e,"unknown"),given=15,goodRating=5)
```

Recommendations as 'topNList' with n = 10 for 50 users.

```
TP
3.26
FP
6.74
FN
10.16
TN
64.84
precision
0.326
recall
0.271417691734827
TPR
0.271417691734827
FPR
0.0910752577652364
```

Gerando o histograma da distribuição dos *ratings*:

```
hist(getRatings(Jester5k),main="Distribution of ratings")
```



Descobrimos o melhor item, com maior média de *rating*:

```
best <- which.max(colMeans(Jester5k))
best
```

j50: 50

Agora utilizando a *Jester5k* para demonstrar o método de fatoração de matriz de Simon Funk.

```
data(Jester5k)
train <- as(Jester5k[1:100], "matrix")
fsvd <- funkSVD(train, verbose=TRUE)
```

A função acima reconstrói a matriz de ratings e a decompõe em U e V (matriz de usuário e matriz de itens). A decomposição da matriz de ratings é realizada por meio da otimização do gradiente descendente popularizado por Simon Funk (por isso *funkSDV*). Calculando o erro quadrático médio com base nos ratings definidos na base de treino temos:

```
r <- tcrossprod(fsvd$U, fsvd$V)
RMSE(train, r)
```

3.20144659134123

Testando o modelo e calculando o erro na base de teste temos:

```
test <- as(Jester5k[101:105], "matrix")
p <- predict(fsvd, test, verbose = TRUE)
RMSE(test, p)
```

3.4889273935307

Agora utilizando a *MSWeb* como um exemplo de base com *ratings* binários.

```
library(recommenderlab)
data("MSWeb")
MSWeb10 <- sample(MSWeb[rowCounts(MSWeb) > 10,], 50)
MSWeb10
```

50 x 285 rating matrix of class 'binaryRatingMatrix' with 661 ratings.

```
e <- evaluationScheme(MSWeb10, method = "split", train=0.9, k=1, given=3)
e
```

Evaluation scheme with 3 items given
Method: 'split' with 1 run(s).
Training set proportion: 0.900
Good ratings: NA
Data set: 50 x 285 rating matrix of class 'binaryRatingMatrix' with 661 ratings.

Criando um sistema de recomendação de filtro colaborativo baseado em usuário usando o *recommender* e calculando predições:

```
r <- Recommender(getData(e, "train"), "UBCF")
p <- predict(r, getData(e, "known"), type="topNList", n=10)
p
calcPredictionAccuracy(p, getData(e, "unknown"), given=3)
```

Recommendations as 'topNList' with n = 10 for 5 users.

TP

5

FP

5

FN

5.4

TN

266.6

precision

0.5

recall

0.475

TPR

0.475

FPR

0.0183779399837794

Por fim, a demonstração do sistema híbrido de recomendação utilizando a base *MovieLense*. Nesse caso, escolheu-se separar as bases de treino e teste sem o *evaluationscheme*.

```
data("MovieLense")
MovieLense100 <- MovieLense[rowCounts(MovieLense) > 100,]
train <- MovieLense[1:100]
test <- MovieLense[101:103]
```

```
recom <- HybridRecommender(Recommender(train, method="POPULAR"),
                           Recommender(train, method="RANDOM"),
                           Recommender(train, method="RERECOMEND"),
                           weights = c(.6,.1,.3))

getModel(recom)

as(predict(recom, test), "list")
```

```
$recommender
$recommender[[1]]
Recommender of type 'POPULAR' for 'realRatingMatrix'
learned using 100 users.

$recommender[[2]]
Recommender of type 'RANDOM' for 'realRatingMatrix'
learned using 100 users.

$recommender[[3]]
Recommender of type 'RERECOMEND' for 'realRatingMatrix'
learned using 100 users.

$weights
[1] 0.6 0.1 0.3
```

\$'101'

'Primary Colors (1998)' 'Great Day in Harlem, A (1994)' 'Whole Wide World, The (1996)'
 'Murder, My Sweet (1944)' 'Fear of a Black Hat (1993)' 'Crooklyn (1994)'
 'Hearts and Minds (1996)' 'Pillow Book, The (1995)' 'Love! Valour! Compassion! (1997)'
 'Prefontaine (1997)'

\$'102'

'Whole Wide World, The (1996)' 'Primary Colors (1998)' 'Crooklyn (1994)'
 'Hearts and Minds (1996)' 'Murder, My Sweet (1944)' 'Great Day in Harlem, A (1994)'
 'Fear of a Black Hat (1993)' 'Brassed Off (1996)' 'Love! Valour! Compassion! (1997)'
 'Lassie (1994)'

\$'103'

'Murder, My Sweet (1944)' 'Hearts and Minds (1996)' 'Crooklyn (1994)'
 'Whole Wide World, The (1996)' 'Love! Valour! Compassion! (1997)'
 'Primary Colors (1998)' 'Great Day in Harlem, A (1994)' 'Hugo Pool (1997)'
 'Pillow Book, The (1995)' 'Fear of a Black Hat (1993)'

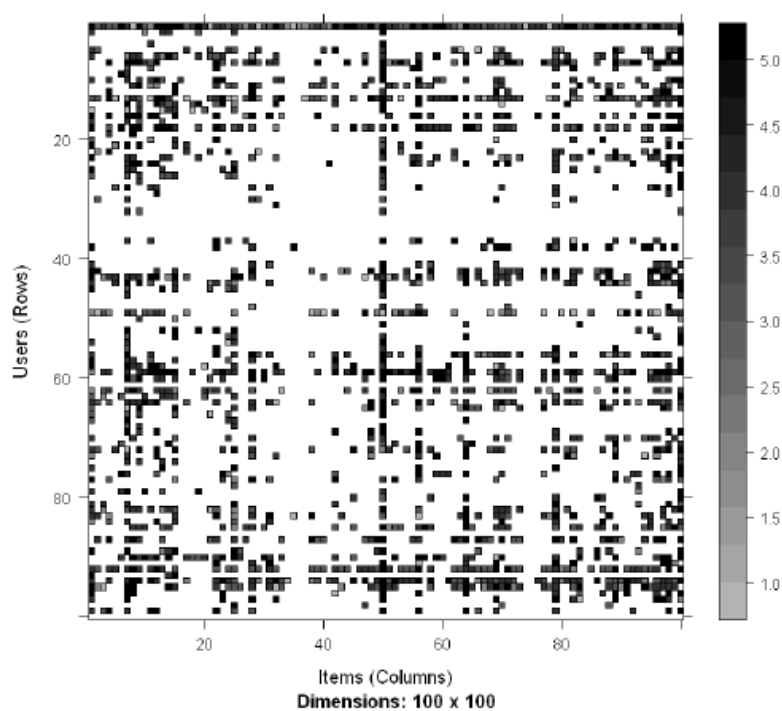
Olhando os primeiros *ratings* do primeiro usuário:

```
head(as(MovieLense[1,], "list")[[1]])
```

```
Toy Story (1995)
5
GoldenEye (1995)
3
Four Rooms (1995)
4
Get Shorty (1995)
3
Copycat (1995)
3
Shanghai Triad (Yao a yao yao dao waipo qiao) (1995)
5
```

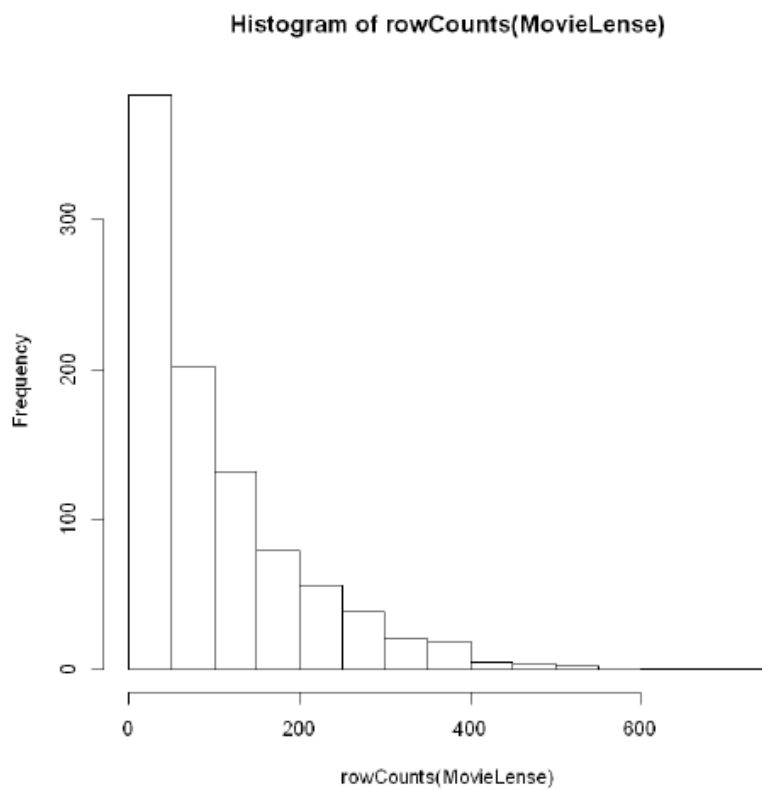
Visualizando parte da matriz:

```
image(MovieLense[1:100,1:100])
```



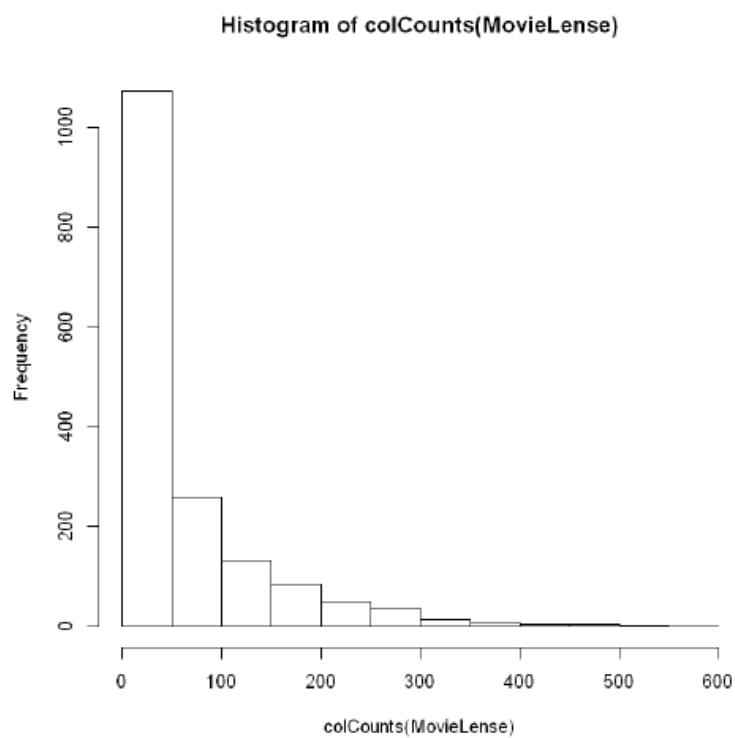
Montando um histograma de número de *ratings* por usuário:

```
hist(rowCounts(MovieLense))
```



Montando um histograma do número de *ratings* por filme:

```
hist(colCounts(MovieLense))
```



Calculando o rating médio da base:

```
mean(rowMeans(MovieLense))
```

```
3.58756455155972
```

Olhando para a base de metadados:

```
head(MovieLenseMeta)
```

title	year	url	unknown	Action	Adventure	Animation	C
Toy Story (1995)	1995	http://us.imdb.com/M/title-exact?Toy%20Story%20(1995)	0	0	0	1	
GoldenEye (1995)	1995	http://us.imdb.com/M/title-exact?GoldenEye%20(1995)	0	1	1	0	
Four Rooms (1995)	1995	http://us.imdb.com/M/title-exact?Four%20Rooms%20(1995)	0	0	0	0	
Get Shorty (1995)	1995	http://us.imdb.com/M/title-exact?Get%20Shorty%20(1995)	0	1	0	0	
Copycat (1995)	1995	http://us.imdb.com/M/title-exact?Copycat%20(1995)	0	0	0	0	
Shanghai Triad (Yao a yao yao dao waipo qiao) (1995)	1995	http://us.imdb.com/Title?Yao+a+yao+yao+dao+waipo+qiao+(1995)	0	0	0	0	

APÊNDICE G - APLICAÇÃO DETALHADA NA BASE DE AÇÕES

A base contém dados transacionais de bolsa mascarados durante um período de 252 DUs. Primeiramente ela será importada e analisada.

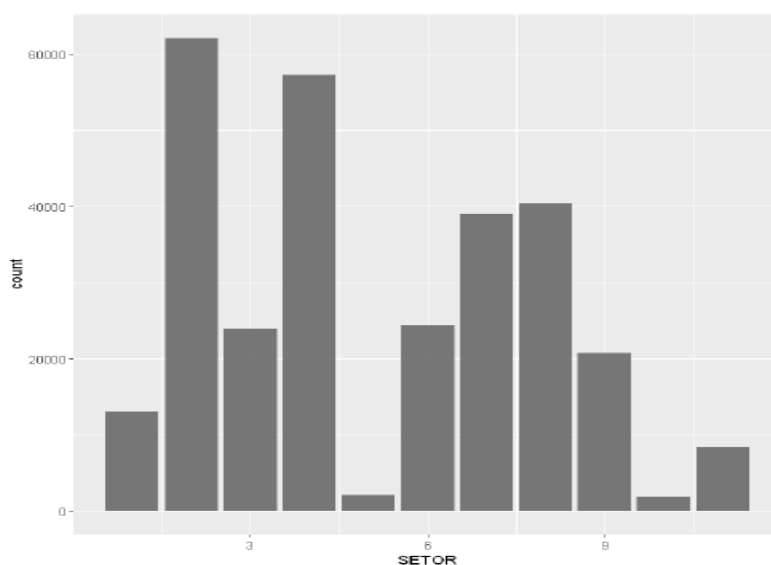
```
BASEACOES <- read.csv2("C:/Users/R/Documents/FGV/PROJETO/Nova pasta/PROJETO CIÊNCIA DE
DADOS/BASE FINAL 1.csv")
str(BASEACOES)
summary(BASEACOES)
```

```
'data.frame': 293151 obs. of 4 variables:
 $ USUARIO : int 1 1 1 1 1 3 19 19 19 19 ...
 $ ITEM : int 110 166 50 211 3 27 154 107 126 88 ...
 $ SETOR : int 8 11 7 4 3 2 4 8 2 6 ...
 $ RETORNO252: num 0.18 -0.52 0.03 -0.35 -0.01 3.3 0.71 0.09 0.43 1.58
 ...
```

USUARIO	ITEM	SETOR	RETORNO252
Min. : 1	Min. : 1.0	Min. : 1.000	Min. : -0.8600
1st Qu.: 8953	1st Qu.: 73.0	1st Qu.: 2.000	1st Qu.: 0.0000
Median :18505	Median :126.0	Median : 4.000	Median : 0.3300
Mean :18787	Mean :129.5	Mean : 5.077	Mean : 0.6888
3rd Qu.:28199	3rd Qu.:183.0	3rd Qu.: 7.000	3rd Qu.: 0.6500
Max. :40748	Max. :254.0	Max. :11.000	Max. :20.0800

Observando sua descrição é possível perceber que ela possui 293.151 observações de ratings de ações em 1 ano, com 40.748 usuários e 254 itens de 11 setores (mascarados, assim como o restante da base), com retornos que vão de -86% a 2008% no período analisado.

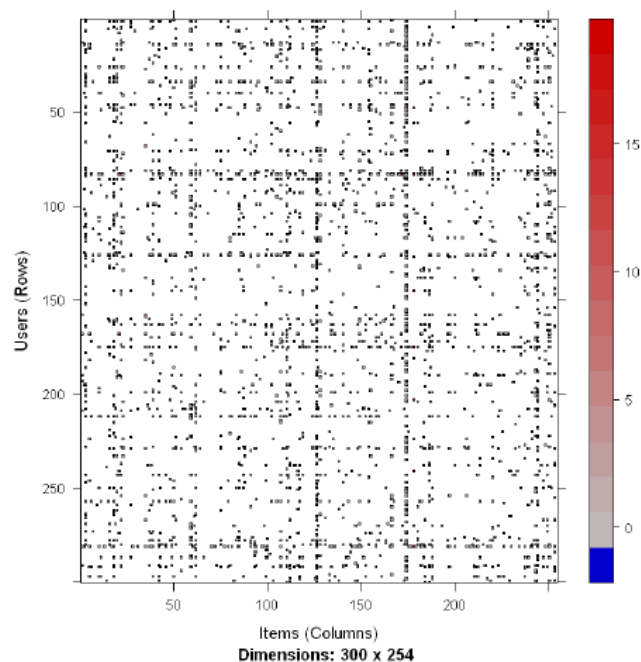
```
library(ggplot2)
ggplot(data = BASEACOES) + geom_bar(aes(SETOR, fill = ITEM), alpha = 0.8) +
  ggtitle("Resumo tipo - item")
```



A base já tratada possui user-id, item-id, o setor mascarado da ação (fonte: B3) e os retornos das ações. Para a aplicação dos algoritmos será considerado apenas o retorno como um rating. Portanto, será criada uma nova base a partir desta com user-id, item-id e retorno/rating. Tal base será transformada em uma `realRatingMatrix`, para que fique no formato adequado para aplicação dos algoritmos. Teremos, portanto, uma matriz com 40.748 linhas (usuários) e 254 colunas (itens). Na interseção de cada usuário com cada item teremos o retorno (representando o rating) do mesmo se o usuário teve contato com tal item ou N/A, caso o contato não exista.

```
library(recommenderlab)
BASEACOES_2 <- cbind(BASEACOES[,1:2],BASEACOES[,4])
BASEFINAL <- as(BASEACOES_2,"realRatingMatrix")
```

```
image(BASEFINAL[1:300,1:254])
```



Obtendo o rating médio (média das médias):

```
mean(rowMeans(BASEFINAL))
```

```
0.88221406818992
```

Verificando o número de ratings:

```
nratings(BASEFINAL)
```

```
293151
```

Obtendo o número de ratings por linha (usuário):

```
summary(rowCounts(BASEFINAL))
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1.000	2.000	4.000	7.194	9.000	166.000

Obtendo o número de ratings por coluna (item):

```
summary(colCounts(BASEFINAL))
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
3.0	130.2	541.0	1154.1	1382.5	14601.0

O `evaluationscheme` cria um esquema de avaliação, dividindo a base a ser analisada em treino e teste, de modo a permitir uma avaliação a ser executada em um número "k" de etapas em um dos seguintes métodos: - "split": divide a base em treino e teste aleatoriamente em uma proporção determinada pelo usuário (k é por default igual a 1, pois a base é dividida apenas uma vez); - "cross-validation": cria um sistema de avaliação k-folds. Os dados são divididos em k partes (por default 10 partes) e, em cada etapa k-1 partes são utilizadas para treinar o sistema e a parte remanescente é deixada para teste; - "bootstrap": realiza o treino pegando uma amostra (amostra com substituição) pelo número de usuários que há nos dados. Todos os objetos que não estiverem na base de treino são usados para teste. Dentre os argumentos, além dos dados, do método a ser utilizado e o número de etapas de validação k, há: - "given": indica o número de itens dados por observação; - "goodRating": indica o rating considerado bom para a avaliação. Criando um sistema de recomendação de filtro colaborativo baseado no usuário (UBCF - *user based collaborative filtering*).

```
r <- Recommender(getData(e2,"train"),"UBCF")
```

```
p <- predict(r, getData(e2,"known"), type="ratings")
```

```
p
```

```
4075 x 254 rating matrix of class 'realRatingMatrix' with 771548 ratings.
```

```
calcPredictionAccuracy(p, getData(e2,"unknown"),given=1)
```

RMSE

2.50000293513906

MSE

6.2500146757039

MAE

1.05183131453812

```
p <- predict(r,getData(e2,"known"), type="topNList")
p
calcPredictionAccuracy(p, getData(e2,"unknown"),given=1, goodRating = 0.06)
```

Recommendations as 'topNList' with n = 10 for 4075 users.

TP

0.296687116564417

FP

7.19533742331288

FN

4.08196319018405

TN

241.426012269939

precision

0.0396003930560105

recall

0.0464732248601477

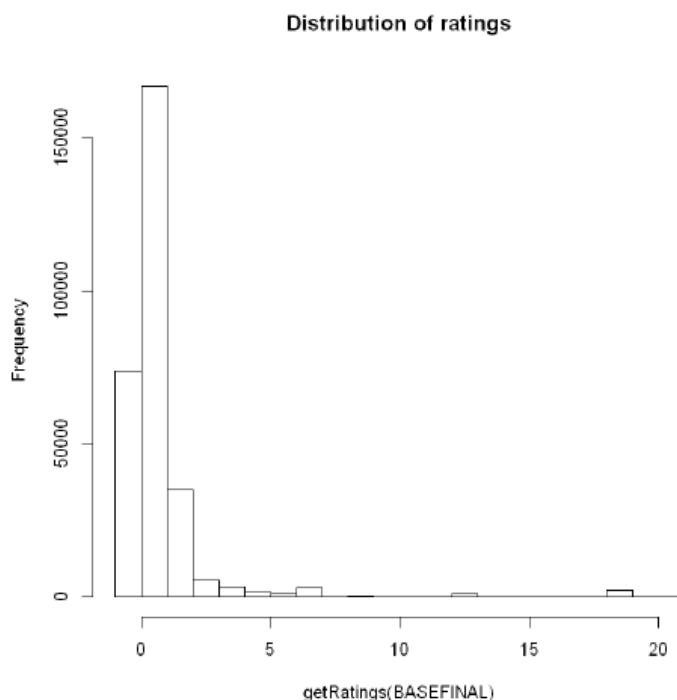
TPR

0.0464732248601477

FPR

0.0290424255734358

```
hist(getRatings(BASEFINAL),main="Distribution of ratings")
```

```
best <- which.max(colMeans(BASEFINAL))
best
```

67: 67

O item 67 é o que possui a maior média de rating. Seria o recomendado no caso de uma recomendação não personalizada. Utilizando agora o método de fatoração de matriz de Simon Funk.

```
train <- as(BASEFINAL[1:39000], "matrix")
fsvd <- funkSVD(train, verbose=TRUE)
```

A função `funkSVD` implementa a decomposição da matriz de ratings usando a otimização estocástica do gradiente descendente popularizada por Simon Funk. Seus argumentos incluem a matriz, o número de características (`k`), o termo de regularização (`gamma`), a taxa de aprendizagem (`lambda`), a melhora mínima por iteração (`min_improvement`), o número mínimo de iterações por características (`min_epochs`), o número máximo de iterações por característica (`max_epochs`) e se o processo vai ou não mostrar o progresso (`verbose = TRUE or FALSE`).

```
r <- tcrossprod(fsfd$U, fsfd$V)
RMSE(train, r)
```

0.137476245284589

```
test <- as(BASEFINAL[39001:40000], "matrix")  
p <- predict(fsvd, test, verbose = TRUE)  
RMSE(test, p)
```

0.173088985609953

APÊNDICE H - APLICAÇÃO DETALHADA NA BASE DE FUNDOS

A base contém dados transacionais de fundos de investimento mascarados durante um período de 252 DUs. Primeiramente ela será importada e analisada.

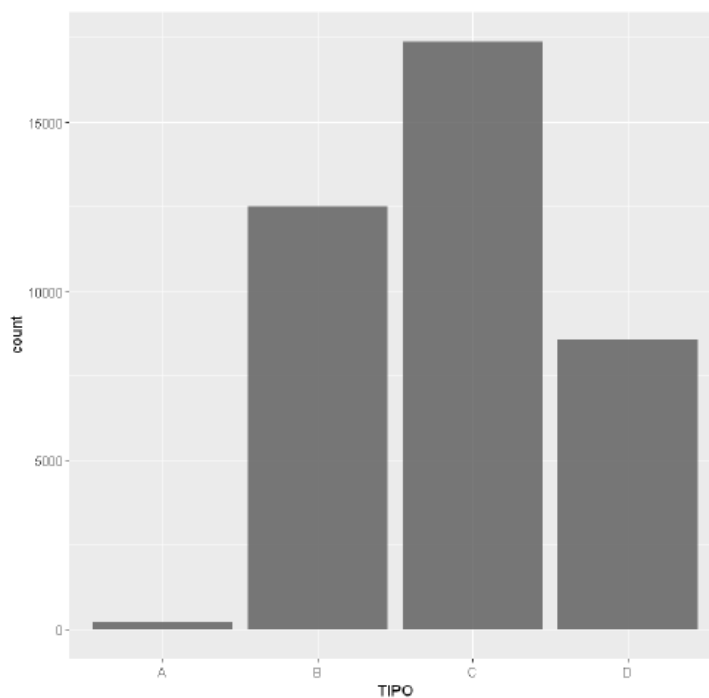
```
BASEFUNDOS <- read.csv2("C:/Users/R/Documents/FGV/PROJETO/Nova pasta/PROJETO CIÊNCIA D
E DADOS/BASE FINAL 2.csv")
str(BASEFUNDOS)
summary(BASEFUNDOS)
```

```
'data.frame': 38654 obs. of 4 variables:
 $ USUARIO : int 1 2 3 4 4 5 5 6 7 ...
 $ ITEM : int 1 2 3 4 5 2 1 4 4 4 ...
 $ TIPO : Factor w/ 4 levels "A","B","C","D": 3 3 3 3 3 3 3 3 3 3 ...
 $ RETORNO252: num 0.0729 0.1012 0.0809 0.0663 0.0952 ...
```

USUARIO	ITEM	TIPO	RETORNO252
Min. : 1	Min. : 1.00	A: 206	Min. : -0.007305
1st Qu.: 5174	1st Qu.: 9.00	B: 12491	1st Qu.: 0.072469
Median : 10782	Median : 15.00	C: 17391	Median : 0.096670
Mean : 10932	Mean : 25.63	D: 8566	Mean : 0.197163
3rd Qu.: 16550	3rd Qu.: 32.00		3rd Qu.: 0.298141
Max. : 22870	Max. : 108.00		Max. : 0.573997

Observando sua descrição é possível perceber que ela possui 38.654 observações de ratings de ações em 1 ano, com 22.870 usuários e 108 itens de 4 tipos (mascarados, assim como o restante da base), com retorno que vão de -0,73% a 57% no período analisado.

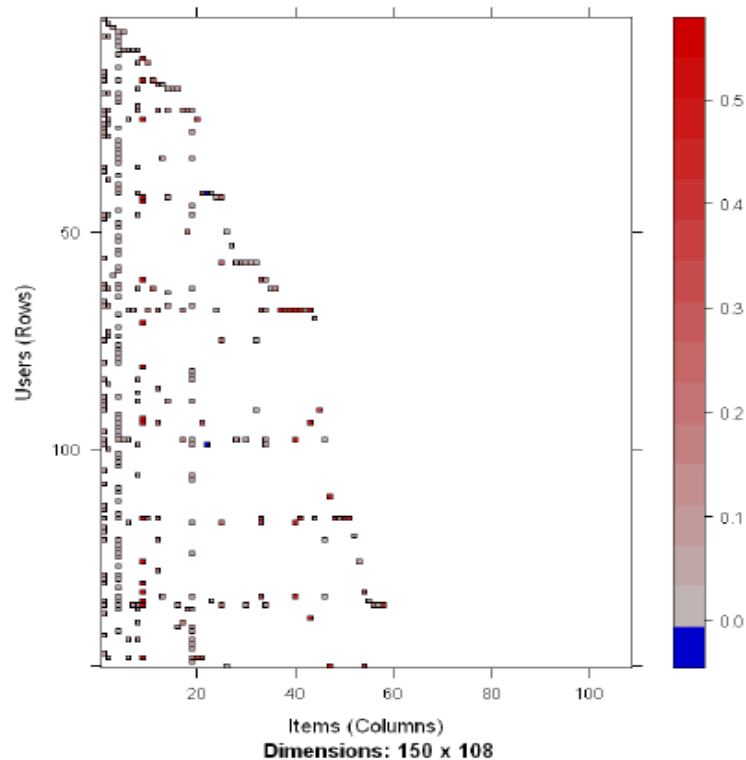
```
library(ggplot2)
ggplot(data = BASEFUNDOS) + geom_bar(aes(TIPO, fill = ITEM), alpha = 0.8) +
  ggtitle("Resumo tipo - item")
```



A base já tratada possui user-id, item-id, o tipo mascarado do fundo (FIRF,FIM,FIA, CAMBIAL) e os retornos dos mesmos. Para a aplicação dos algoritmos será considerado apenas o retorno como um rating. Portanto, será criada uma nova base a partir desta com user-id, item-id e retorno/rating. Tal base será transformada em uma `realRatingMatrix`, para que fique no formato adequado para aplicação dos algoritmos. Teremos, portanto, uma matriz com 22.870 colunas (usuários) e 112 linhas (itens). Na interseção de cada usuário com cada item teremos o retorno (representando o rating) do mesmo se o usuário teve contato com tal item ou N/A, caso o contato não exista.

```
library(recommenderlab)
BASEFUNDOS_2 <- cbind(BASEFUNDOS[,1:2],BASEFUNDOS[,4])
BASEFINAL <- as(BASEFUNDOS_2,"realRatingMatrix")
```

```
image(BASEFINAL[1:150,1:108])
```



Obtendo o rating médio (média das médias):

```
mean(rowMeans(BASEFINAL))
```

```
0.198223239147297
```

Verificando o número de ratings:

```
nratings(BASEFINAL)
```

```
38527
```

Obtendo o número de ratings por linha (usuário):

```
summary(rowCounts(BASEFINAL))
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
1.000	1.000	1.000	1.687	2.000	23.000

Obtendo o número de ratings por coluna (item):

```
summary(colCounts(BASEFINAL))
```

Min.	1st Qu.	Median	Mean	3rd Qu.	Max.
2.0	32.0	82.0	356.7	227.2	5018.0

```
e2 <- evaluationScheme(BASEFINAL, method = "split", train = 0.9, given=1, goodRating=0.06)
```

Criando um sistema de recomendação de filtro colaborativo baseado no usuário (UBCF - *user based collaborative filtering*).

```
r <- Recommender(getData(e2,"train"),"UBCF")
```

```
p <- predict(r, getData(e2,"known"), type="ratings")
```

```
p
```

2284 x 108 rating matrix of class 'realRatingMatrix' with 133732 ratings.

```
calcPredictionAccuracy(p, getData(e2,"unknown"),given=1)
```

RMSE

0.295360272559267

MSE

0.0872376906062843

MAE

0.181773492891103

```
p <- predict(r,getData(e2,"known"), type="topNList")
p
calcPredictionAccuracy(p, getData(e2,"unknown"),given=1, goodRating = 0.06)
```

Recommendations as 'topNList' with n = 10 for 2284 users.

TP

0.117775831873905

FP

5.37259194395797

FN

0.786777583187391

TN

100.722854640981

precision

0.021451355661882

recall

0.10205336286841

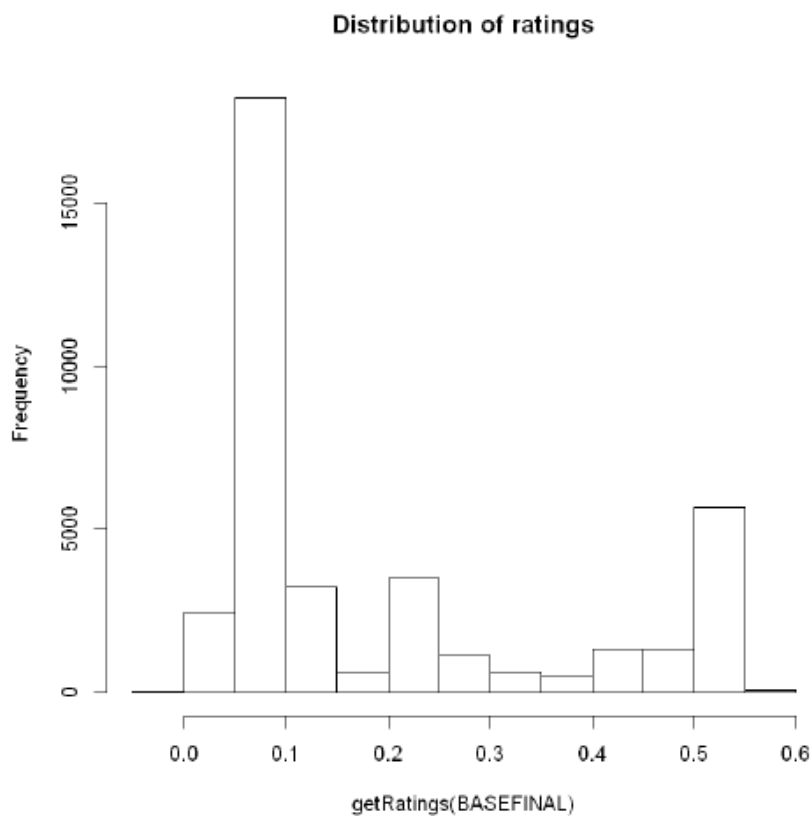
TPR

0.10205336286841

FPR

0.0506195985063815

```
hist(getRatings(BASEFINAL),main="Distribution of ratings")
```



```
best <- which.max(colMeans(BASEFINAL))  
best
```

62: 62

O item 62 é o que possui a maior média de rating. Seria o recomendado no caso de uma recomendação não personalizada. Utilizando agora o método de fatoração de matriz de Simon Funk.

```
train <- as(BASEFINAL[1:20000], "matrix")  
fsvd <- funkSVD(train, verbose=TRUE)
```

```
r <- tcrossprod(fsvd$U, fsvd$V)  
RMSE(train, r)
```

0.0324527840981732

```
test <- as(BASEFINAL[20001:22000], "matrix")  
p <- predict(fsvd, test, verbose = TRUE)  
RMSE(test, p)
```

0.0455433517576075