

UNIVERSIDADE FEEVALE

ALEXANDRE MARTINS KLAFKE

SISTEMA DE RECOMENDAÇÃO COM USO DE MACHINE
LEARNING APLICADO AO SIMULADOR DE CASO CLÍNICO

Novo Hamburgo
2020

ALEXANDRE MARTINS KLAFKE

SISTEMA DE RECOMENDAÇÃO COM USO DE MACHINE
LEARNING APLICADO AO SIMULADOR DE CASO CLÍNICO

Trabalho de Conclusão de Curso
apresentado como requisito parcial
à obtenção do grau de Bacharel em
Ciência da Computação pela
Universidade Feevale

Orientador: Prof. Me Paulo Ricardo Muniz Barros

Novo Hamburgo
2020

AGRADECIMENTOS

Agradeço a todos que de alguma forma me auxiliaram durante todo o processo de desenvolvimento deste trabalho.

Aos meus familiares, esposa e amigos, que tiveram paciência e me deram muita força durante esse período.

E agradeço especialmente ao meu orientador Paulo Ricardo, por todo apoio e dedicação ao longo deste ano.

RESUMO

Os estudantes de nível superior na área da saúde encontram dificuldade de associar conhecimentos adquiridos na forma teórica e aplicá-los na prática. O ensino em saúde, necessita que sejam desenvolvidas algumas habilidades no futuro profissional da saúde para atuar com base numa avaliação da situação que se sustenta em racionalidade científica, facilitando dessa forma a associação entre a teoria e a prática. Com este cenário, práticas de estudo de casos clínicos são fundamentais para desenvolver algumas habilidades importantes para a formação dos alunos. Porém as pesquisas apontam que os estudantes também acabam tendo dificuldades com pacientes reais em práticas clínicas, uma vez que o paciente pode se sentir desconfortável ao saber que está sendo avaliado por um estudante de área de saúde em treinamento. Simuladores são uma maneira eficaz para integrar teoria e prática no ensino superior da área da saúde, e disponibilizam um ambiente mais próximo ao real sem os potenciais riscos à prática clínica. O Health Simulator é um simulador do tipo paciente virtual desenvolvido na Universidade Feevale que busca melhorar o raciocínio clínico, diagnóstico e o pensamento crítico do aluno da área da saúde. No Health Simulator o Educador consegue elaborar casos clínicos que serão utilizados posteriormente pelos alunos para estudo. Para este simulador foi desenvolvido um sistema de recomendação que auxilia o aluno durante a simulação, recomendando o material de estudo para aprimorar o que o aluno teve de maior dificuldade. Sistemas de recomendação fornecem aos usuários recomendações de itens, visando equilibrar elementos como novidade, precisão, dispersão e estabilidade nas recomendações, desta forma são empregadas inúmeras técnicas de recomendação que utilizam diferentes tipos de abordagens. *Machine Learning* é uma área da inteligência artificial que vem sendo estudada e utilizada para simular o aprendizado humano, permitir e obter conhecimento do mundo real a nível computacional. Dessa forma, o objetivo deste trabalho foi desenvolver um sistema de recomendação utilizando algoritmos de *Machine Learning* no Health Simulator. Os resultados se mostraram superiores frente ao sistema de recomendação já proposto.

Palavras-chave: Sistema de recomendação. Machine Learning. Simulador. Health Simulator.

ABSTRACT

Medical science students and those of other health related areas find it difficult to translate acquired theoretical knowledge into its practical application. The teaching of Health Sciences requires the development of certain skills that will enable the future health professional practitioner to guide their assessments based on scientific rationality, therefore, enabling the association between theory and practice. In this scenario, the practice of clinical study cases is of paramount importance for the development of essential skills and general education of the students. However, surveys indicate that students encounter difficulties when facing real life patients in clinical practice as the patients might also feel uncomfortable by knowing they are being assessed by a professional whose education is in progress. Simulators are an effective way to integrate theory and practice in the Health Sciences education, as they provide a safe environment for real-life practice without the potential risks of the regular clinical practice. Developed at Feevale University, the Health Simulator simulates a virtual patient whose aim is to improve medical student's clinical assessment, diagnostics, and critical thinking. Educators can use the Health Simulator to create clinical cases to be used by students for studying purposes. A system of recommendation has been developed for this simulator, which recommends the students with study materials based on student's simulation performance. Recommendation systems provide its users with recommendation of items with the purpose of balancing out elements as novelty, precision, dispersion and stability in the recommendations by applying several recommendation techniques that utilize different types of approach. Machine Learning is an arm of artificial intelligence that has been being studied and applied to simulate human learning process, to allow and obtain knowledge of the real world to a computational level. Therefore, the purpose of this paper is to develop a recommendation system utilizing Machine Learning algorithms in the Health Simulator. The results presented here were more satisfactory than the ones obtained with the current recommendation system.

Keywords: Recommendation System. Machine Learning. Simulator. Health Simulator.

LISTA DE FIGURAS

Figura 1 - Estrutura do Health Simulator.....	18
Figura 2 - Consultório de classe C.....	19
Figura 3 - Elementos de composição do cenário.....	20
Figura 4 - Modelos de enfermeiras e médicos.....	20
Figura 5 - Rede bayesiana para doação de sangue.....	22
Figura 6 - Interface administrativa.....	23
Figura 7 - Técnicas de recomendação.....	26
Figura 8 - Técnica de filtro baseada em conteúdo.....	27
Figura 9 - Modelo ER original simplificado.....	38
Figura 10 - O modelo de usuário no HS.....	39
Figura 11 - Modelo ER resultante das alterações ao BD.....	40
Figura 12 - Cabeçalho e conteúdo do arquivo ratings.csv.....	41
Figura 13 - Cabeçalho e conteúdo do arquivo movies.csv.....	42
Figura 14 - Cabeçalho e conteúdo do arquivo tags.csv.....	43
Figura 15 - Cabeçalho e conteúdo do arquivo links.csv.....	43
Figura 16 - Mapeamento entre dataset e BD.....	44
Figura 17 - Mapeamento entre BD e dataset.....	46
Figura 18 - Consulta SQL para o arquivo ratings.csv.....	46
Figura 19 - Consulta SQL para o arquivo movies.csv.....	47
Figura 20 - Dataset no método train/test.....	47
Figura 21 - Dataset ao método cross-validation.....	48
Figura 22 - Código da chamada dos algoritmos KNN.....	49
Figura 23 - Código da chamada dos algoritmos SVD e SVD++.....	50
Figura 24 - Resultado dos algoritmos nas métricas propostas usando train/test.....	51

Figura 25 - Algoritmos com a métrica MAE.....	52
Figura 26 - Algoritmos com a métrica RMSE.....	52
Figura 27 - Resultado dos algoritmos nas métricas propostas usando k-fold.....	53
Figura 28 - Resultados das métricas de acurácia da predição.....	54
Figura 29 - Comparação do SVD aplicado com intersecção.....	55
Figura 30 – Comparação do SVD aplicado com CF.....	56

LISTA DE ABREVIATURAS E SIGLAS

ML	<i>Machine Learning</i>
MF	<i>Matrix Factorization</i>
SR	Sistema de recomendação
HS	Health Simulator
BD	Banco de Dados
CF	<i>Collaborative Filtering</i>
CBF	<i>Content-Based Filtering</i>
PV	Paciente Virtual
MAE	<i>Mean Absolute Error</i>
RMSE	<i>Root Mean Square Error</i>
REST	<i>Representation Square Error</i>
KNN	<i>K Nearest Neighbor</i>
SVD	<i>Singular Value Decomposition</i>
SVD++	<i>Singular Value Decomposition plus plus</i>
SVM	<i>Support Vector Machine</i>

SUMÁRIO

1 INTRODUÇÃO.....	11
2 SIMULADORES DE ENSINO NA ÁREA DA SAÚDE.....	15
2.1 SIMULADORES DO TIPO PACIENTE VIRUTAL.....	16
2.2 HEALTH SIMULATOR.....	17
2.2.1 O front-end do Health Simulator.....	19
2.2.2 O back-end do Health Simulator.....	21
2.2.2.1 Modelagem do conhecimento.....	21
2.2.2.2 Interface de administração.....	22
2.2.2.3 Serviço web de comunicação.....	23
2.2.2.4 SR do Health Simulator.....	23
3 INTRODUÇÃO AOS ESTUDOS DE SISTEMAS DE RECOMENDAÇÃO.....	25
3.1 TÉCNICAS DE RECOMENDAÇÃO.....	25
3.1.1 Sistemas de recomendação baseados em conteúdo.....	26
3.1.2 Sistema de recomendação colaborativos.....	28
3.1.2.1 Função similaridade.....	28
4 MACHINE LEARNING.....	30
4.1 APLICAÇÕES DE MACHINE LEARNING.....	30
4.2 SISTEMAS DE RECOMENDAÇÃO COM O USO DE MACHINE LEARNING...	31
4.2.1 Algoritmo KNN.....	33
4.2.2 Algoritmo SVD.....	34
4.3 MÉTRICAS.....	35
5 TRABALHO RELACIONADO.....	37
5.1 MODELO JÁ PROPOSTO NO HS.....	38
5.1.1 Base de dados.....	40
5.2 A INSERÇÃO DOS DADOS NO HS.....	43
6 IMPLEMENTAÇÃO.....	45
6.1 RECUPERANDO BASE DE DADOS.....	46
6.2 IMPLEMENTAÇÃO DOS ALGORITMOS.....	47
6.2.1 KNN.....	48
6.2.2 SVD e SVD++.....	49
7 RESULTADOS.....	51
8 COMPARAÇÃO DO SISTEMA DE RECOMENDAÇÃO.....	54
9 CONCLUSÃO.....	57

REFERÊNCIAS BIBLIOGRÁFICAS.....	59
---------------------------------	----

1 INTRODUÇÃO

Há áreas do ensino em que há uma maior dificuldade em associar os conhecimentos adquiridos na forma teórica e aplicá-los na prática. Como pode ser observado no estudo apresentado por HIGGS et al. (2008), estudantes de nível superior na área da saúde encontram esta dificuldade, apesar da grande quantidade de conteúdos e métodos aplicados durante atividades teóricas. O ensino na saúde necessita que sejam desenvolvidas algumas habilidades no futuro profissional, para atuar com base em uma avaliação da situação que se sustenta em racionalidade científica, facilitando a associação entre a teoria e a prática, sendo que alguns autores trazem como essenciais as habilidades cognitivas do pensamento crítico, as afetivas de atenção, sensibilidade, cuidado e preocupação (FACIONE; OTHERS, 2011; PEIXOTO; PEIXOTO, 2017). Com esse cenário, práticas de estudo de casos clínicos podem colaborar para desenvolver algumas habilidades ditas como importantes para formação dos estudantes, que podem permitir a resolução de problemas reais ou em potencial que tem como base sua decisão e julgamento para tomada de decisões (HIGGS et al., 2008; LEWIS; SMITH, 1993).

Porém pesquisas apontam que estudantes também têm dificuldades com pacientes reais em práticas clínicas, uma vez que o paciente pode vir a se sentir desconfortável ao ser informado de estar sendo avaliado por alunos em treinamento. Além disso, uma grande quantidade de alunos por turma pode acabar afetando a produtividade do estudo (BEZ; BARROS; MELLO, 2018).

Na literatura encontram-se evidências que o uso de simulação computacional possui um grande conjunto de técnicas que aplicadas melhoram a forma de conduzir o ensino ao estudante de forma segura, essencialmente se há riscos ou experimentos complexos de difícil reprodução (KER; BRADLEY, 2013). Na saúde, simuladores promovem a assimilação entre a teoria e a prática, fornecendo assim um ambiente prático equivalente ao real, dando segurança ao estudante e instituição de ensino (ZIV; BEN-DAVID; ZIV, 2005).

Existem simuladores do tipo paciente virtual que simulam um ambiente real, rico em detalhes, oferecendo informações com características reais ou o mais próximo das informações que seriam ofertadas no ambiente real, permitindo a participação em cenários e situações próximas às vivenciadas no dia a dia. Ele é utilizado para treinar estudantes em ambientes como uma sala cirúrgica, emergências e formação de procedimentos (SWEET; MCDUGALL, 2008). A melhor característica desses simuladores é que eles permitem que os alunos pratiquem repetitivamente as habilidades a serem aprimoradas, sem a restrição de horário e/ou localização

(HOLZINGER et al., 2009). Com o uso do paciente virtual, tem-se o total controle sobre o ambiente e o feedback pode ser ofertado em tempo real, fazendo com que aspectos que poderiam passar despercebidos, possam ser evidenciados para uma observação, tomada de decisão ou medida, que não podem ser controladas em pacientes reais.

O Health Simulator, é um simulador de paciente virtual apresentado em forma de jogo sério (LIMA et al., 2015). Este simulador é desenvolvido em um projeto na Universidade Feevale e com a colaboração da Universidade do Vale dos Sinos. Em linhas gerais, o Health Simulator tem como objetivo desenvolver o raciocínio clínico e diagnóstico do aluno da área de saúde (BEZ et al., 2016). Neste simulador, a representação do conhecimento é feita por meio de um modelo probabilístico, apresentado na forma de uma rede bayesiana, e ela é utilizada na solução de problemas que envolvam incertezas (BEZ et al., 2017). A rede bayesiana é utilizada para representar o conhecimento de um especialista da área, que no caso da saúde faz uso de normas clínicas para isto. Com base neste modelo, é possível que o educador consiga elaborar casos clínicos no simulador, que serão utilizados posteriormente pelos alunos, como ferramenta adicional para estudo (HELWANGER; ROLIM; BARROS, 2015). Para apoiar o processo de ensino com o Health Simulator, foi desenvolvido um sistema de recomendação, para auxiliar durante a simulação, recomendando ao aluno material de estudo e casos clínicos, para aprimorar aquilo que ele mais teve dificuldade durante a simulação, ou ofertando novos casos (BEZ et al., 2016).

Sistemas de Recomendação surgiram pela primeira vez na década de 90 com intuito de minimizar os problemas com a grande quantidade de informações (MAES; SHARADANAND, 1995), que neste caso específico do Health Simulator seria relacionado a materiais de estudo e casos clínicos. A função do sistema de recomendação neste cenário é indicar ao estudante um grupo de itens dentro de uma gama maior de itens, otimizando o tempo de estudo, uma vez que há uma diminuição no tempo de busca por materiais (DE FAZIO, 2013).

Os sistemas de recomendação utilizam fontes distintas de informação para fornecer aos usuários recomendações de itens, visando equilibrar elementos como novidade, precisão, dispersão e estabilidade nas recomendações. Desta forma, são empregadas diversas técnicas de recomendação, que utilizam diferentes tipos de abordagem. Os tipos de sistemas de recomendação que podem ser implementados segundo BURKE, (2007), são divididos em quatro classes distintas: baseadas em conteúdo, filtragem colaborativa, demográfico e baseada em conhecimento. Ainda outro estudo anterior, BURKE, (2002) apresentou também outras duas técnicas baseada na utilidade e o modelo híbrido, tendo como objetivo a junção de duas

técnicas ou mais, necessitando que se utilize uma grande quantidade de dados para que a obtenção de recomendações de qualidade possa ser feita. É possível observar também alguns estudos utilizando técnicas de *Machine Learning* (ML) aplicados a sistemas de recomendação, como pode ser observado em estudos apresentados por (BRUNIALTI et al., 2015a; DEBNATH, 2008).

ML é uma área da inteligência Artificial que está sendo estudada desde o final da década de 50 (MARTENS, 1959), justamente com a emersão do campo de inteligência artificial e que usa computadores para simular o aprendizado humano e permite identificar e obter conhecimento do mundo real. Apesar dos primeiros conceitos terem aparecido na década de 50, ML foi estudada como um campo separado em 1985 (ANDERSON et al., 1985). Os humanos naturalmente aprendem com a experiência por causa da sua habilidade em argumentar logicamente, porém em contraste, computadores não aprendem fazendo uso de raciocínio, mas sim através de algoritmos computacionais (NASCIMENTO et al., 2018).

Algoritmos de ML em sistemas de recomendação são geralmente classificados em duas categorias: baseados em conteúdo (*content based*) e filtro colaborativo (*collaborative filtering*) apesar de que sistemas de recomendação mais modernos usem os dois métodos, também conhecido como modelo híbrido. Métodos baseados em conteúdo são baseados nas similaridades dos atributos dos itens, e o colaborativo calcula a similaridade das interações (KORDIK; CERNY; FRYDA, 2018).

Segundo MANOUSELIS; VUORIKARI; VAN ASSCHE, (2010), sistemas de recomendação (SR) quando aplicados na educação acabam por trazer várias vantagens ao estudante, como buscar parceiros de estudo com interesses e dificuldades similares, além de encontrar conteúdo de materiais para o aprimoramento do seu conhecimento (REIDEL et al., 2017). Como descrito anteriormente, o uso do ML em SR mostra-se bastante promissor e espera-se que aplicando essas técnicas no Health Simulator consiga-se um melhor desempenho na qualidade das recomendações para o cenário do Health Simulator. Sendo assim, a proposta apresentada aqui, tem como propósito desenvolver um Sistema de Recomendação aplicando técnicas de *Machine Learning*, para serem utilizadas no ambiente Health Simulator, e que sua aplicação possa melhorar a qualidade das recomendações atualmente obtidas no Simulador.

No capítulo 2 deste trabalho será apresentado simuladores de ensino aplicado na área da saúde, como simuladores já desenvolvidos e também o Health Simulator, que se encontra em desenvolvimento na Universidade Feevale. No capítulo 3 são destacados os conceitos de sistemas de recomendação, como as diferentes técnicas utilizadas por trabalhos acadêmicos, já

no capítulo 4 é apresentado conceitos de *Machine Learning* e suas aplicações em sistemas de recomendação, além dos principais algoritmos utilizados em estudos similares. Em seguida no capítulo 5 e 6 é apresentado o SR do trabalho relacionado e a implementação do sistema de recomendação aplicado por este trabalho. Por fim no capítulo 9 as conclusões sobre o trabalho desenvolvido.

2 SIMULADORES DE ENSINO NA ÁREA DA SAÚDE

Os estudantes da área da saúde de nível superior, no decorrer do curso ou ainda nos seus primeiros anos de práticas clínicas, encontram dificuldades de associar os conhecimentos adquiridos na forma teórica e aplicá-los na prática, mesmo com uma grande quantidade e técnicas de estudo aplicados durante atividades teóricas em sala de aula (BEZ; BARROS; MELLO, 2018). Segundo TSUJI; AGUILAR-DA-SILVA, (2010) declaram em seu estudo que há a necessidade de prática das atividades que possuem uma maior complexidade de aprendizado desde o início do curso, visando uma melhora na educação. Porém pesquisas apontam que estudantes também têm dificuldades com pacientes reais em práticas clínicas, uma vez que o paciente pode a vir se sentir desconfortável por estar sendo avaliado por alunos em treinamento. Além disso, uma grande quantidade de alunos por turma pode acabar afetando a produtividade do estudo (ZIV; BEN-DAVID; ZIV, 2005).

A simulação e suas técnicas podem ser aplicadas aos alunos de todos os níveis, desde o estudante que está no início de sua trajetória no ensino até o especialista, sendo um dos principais guias para desenvolver profissionais em saúde de forma segura e eficiente (KER; BRADLEY, 2013). O ensino com auxílio de simulação pode auxiliar o estudante a compreender relações complexas, que de outra maneira demandariam equipamentos de maior custo ou perigosos e diminui o tempo que é necessário para a obtenção das competências clínicas que seriam adquiridas com a realidade clínica. Além disso, também permite que o aluno busque novos métodos e estratégias para a resolução de problemas propostos no simulador, alterando as variáveis de entrada e analisando e entendendo as alterações que acontecem no ambiente simulado (BEZ et al., 2017).

Os simuladores têm se mostrado como uma ferramenta efetiva para minimizar os problemas descritos acima, uma vez que servem como ferramentas para os alunos articularem os estudos em sala de aula com a realidade clínica de uma maneira segura e efetiva (REIDEL et al., 2017). Os simuladores podem ser classificados em três tipos distintos: baseados em modelos físicos; os que utilizam computadores para simular uma realidade virtual e aqueles que combinam os dois modelos. Dentre esses simuladores também existe o simulador baseado em realidade virtual, realidade aumentada e a combinação desses tipos chamados de realidade mista (KNEEBONE, 2003).

Na próxima subseção serão apresentados alguns simuladores do tipo paciente virtual (PV), com o objetivo de identificar as principais funcionalidades desses sistemas e por fim será

apresentado o Health Simulator, o simulador de casos clínicos usado como modelo para este trabalho.

2.1 SIMULADORES DO TIPO PACIENTE VIRTUAL

O simulador do tipo paciente virtual pode ser definido como um software interativo baseado em tela que pode ser apresentado como: páginas HTML, realidade virtual, realidade aumentada ou até mesmo realidade mista (BEZ et al., 2017). Seu principal objetivo é simular cenários clínicos reais, com características e situações da vida real com a maior fidelidade ao ambiente simulado, em que o estudante empenha o papel do profissional da saúde realizando atividades como por exemplo a solicitação de exames e a tomada de decisões de diagnóstico e tratamento possibilitando a participação em situações e cenários semelhantes ao cotidiano (KER; BRADLEY, 2013; TIZARD; HUGHES, 2008).

Além de facilitar a associação entre a teoria e a prática, essas ferramentas têm a vantagem de trazer as características e situações da vida real, sem expor risco ao estudante ou ao paciente, e também fornecendo segurança a instituição de ensino (ZIV; BEN-DAVID; ZIV, 2005), a principal característica destes simuladores é a possibilidade de práticas clínicas que podem ser repetidas por alunos quando necessárias e ainda sem levar em conta a hora, local, e semestre do estudante, assim possibilitando que o aluno desenvolva em seu ritmo de aprendizado, favorecendo o ambiente de ensino como um todo. Segundo MARONI (2013), estudar com pacientes reais em situações não convencionais nem todos os fatores podem ser controlados e dimensionados. Contudo, com o uso de simuladores do tipo PV o educador tem a gerência total sobre o ambiente e *feedback* de forma consistente, garantindo um aprendizado em pontos que poderiam por passar despercebidos.

Nesta definição, é importante que o ecossistema do simulador, como um todo, sempre acompanhe as decisões e passos do estudante em todo seu processo de aprendizagem e ensino fornecendo um *feedback* sempre que for necessário, para que com isto o estudante sinta-se sustentado pelo ambiente e assim sendo capaz de perceber sua eficiência e deficiência durante o estudo (KER; BRADLEY, 2013).

Os simuladores e ferramentas do tipo paciente virtual tem um grande potencial para ser uma maneira inovadora e produtiva de ensino, que a partir de uma metodologia ativa, pode auxiliar na formação dos alunos. A grande aceitação das afirmativas propostas pelos alunos estudados valida o uso dessa plataforma de simulação de casos clínicos virtuais, planejando

considerar a possível inserção dessa ferramenta de ensino nas universidades (LOPES et al., 2018).

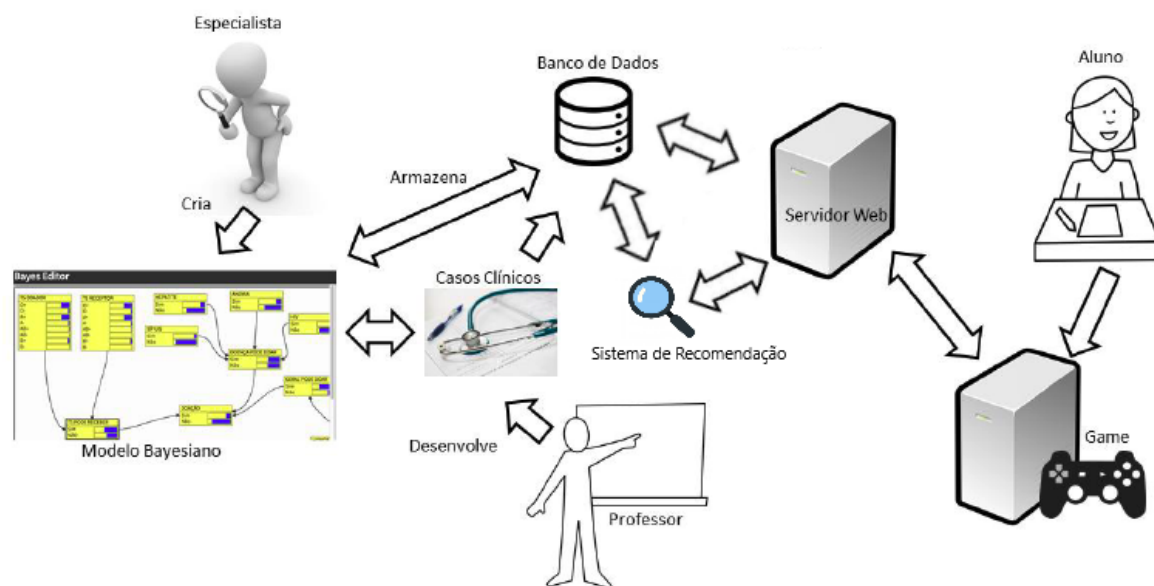
2.2 HEALTH SIMULATOR

O Health Simulator (HS) é um simulador do tipo paciente virtual na forma de jogo sério, e seu desenvolvimento ocorre em um grupo de pesquisa interdisciplinar da Universidade Feevale em parceria com a Universidade do Vale do Rio do Sinos, contando com profissionais da medicina, informática, jogos digitais, design e biomedicina (BARROS, 2020). O programa ainda conta com teses de doutorado e dissertações de mestrado e diversos trabalhos de conclusão de curso, inclusive o aqui apresentado.

A finalidade deste simulador é desenvolver o raciocínio clínico e o diagnóstico do aluno da área da saúde, acompanhando as ações e evolução de cada indivíduo que o utiliza. O seu ambiente oferece situações habituais da área da saúde, porém retirando os riscos cotidianos dessas atividades (BEZ et al., 2016). O simulador foi planejado como um recurso adicional para que os alunos apliquem seus conhecimentos e assim compreender o raciocínio clínico, com isto o aluno adquirirá uma maior precisão e confiança quando confrontado com uma situação real (LIMA et al., 2015).

Uma vez que o simulador é complexo, ele conta com uma estrutura robusta, que é apresentada na Figura 1. Ele oferece segurança e confiabilidade, tendo os pilares característicos de um software, sendo dividido em duas grandes áreas. A primeira que é responsável pela interação direta do aluno é composta pelo jogo sério, chamado de *front-end* e a segunda área é composta pelo servidor, em que é utilizado como repositório das informações necessárias para execução do jogo, chamado de *back-end*. Este, por sua vez, é dividido em outras 4 seções, são elas: Sistemas de geração da informação, no qual ficam os dados armazenados; Os casos que são gerados, quem mantém os elementos necessários para a simulação; E o modelo de apresentação do conhecimento, que é utilizado pelos professores e especialistas; Por ultimo a seção que contém o sistema de recomendação proposto pelo REIDEL et al, (2017).

Figura 1 – Estrutura do Health Simulator



Fonte: Figura adaptada

Na Figura 1, são destacadas as características do funcionamento do HS. A interação do aluno com sistema do HS, é através da uma interface web, chamada de interface de jogo, que na Figura 1 é representada pela legenda *game*. O professor possui uma interface específica em que ele pode detalhar casos clínicos para que os alunos utilizem posteriormente. Um especialista em conhecimento auxilia os professores na correta adequação das informações dos casos clínicos para que possam ser utilizadas em redes Bayesianas.

O processo de simulação, por parte de um aluno acontece com os seguintes passos, primeiro o aluno seleciona um item de uma lista de casos possíveis para a simulação, logo após o aluno interage com o PV, através da observação de sinais e sintomas. Durante esta etapa o aluno seleciona quais sinais e sintomas ele quer investigar, com base nesta seleção o ambiente analisa essas informações e a interação do aluno, para utilizar no motor pedagógico para obtenção do cálculo de desempenho do estudante com base nas suas ações percorridas. Na segunda fase o aluno seleciona um dos diagnósticos possíveis e a partir dessas informações, e com bases nas variáveis presentes no modelo de representação do conhecimento e sua influência no caso, o ambiente pode orientar novamente o aluno com alguma mensagem ou indicação de material de apoio. E na terceira e última fase, a indicação de conduta, que ocorre como no processo anterior, em que o estudante decide dentre as opções qual a melhor conduta para o caso, e o ambiente pode intervir novamente, indicando algum material de apoio, mensagem pedagógica ou indicação de um novo desafio para que o aluno possa desenvolver ou melhorar as suas habilidades (BARROS, 2020).

2.2.1 O front-end do Health Simulator

O *front-end* é apresentado para o aluno na forma de jogo sério, através de uma interface web. Nessa interface existem cenários e personagens que representam o cotidiano da área da saúde, sendo que todos os modelos tridimensionais desenvolvidos para o *front-end* incluem objetos que representam os ambientes e profissionais da saúde, criados no motor gráfico Unity. Os cenários incluem consultórios e hospitais A, B ou C, residências e um hospital do Sistema Único de Saúde (SUS) (LIMA et al., 2015). A Figura 2 representa um consultório classe C, com maca e os demais objetos desse cenário.

Figura 2 – Consultório de classe C



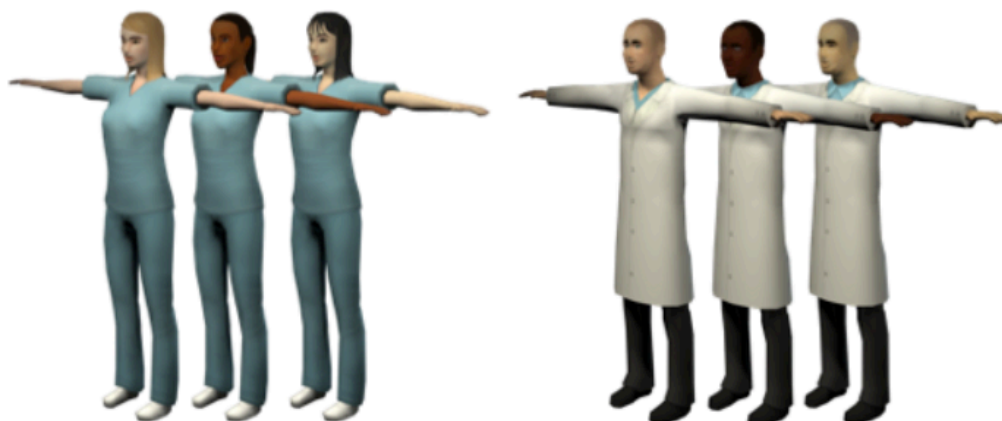
Fonte: (LIMA et al., 2015)

Além dos vários ambientes descritos acima, o simulador também conta com uma grande quantidade de elementos de arte, que permitem customizar e personalizar os ambientes. Atualmente o simulador conta com 19 cenários e com mais de 300 tipos de componentes, que quando combinados, permitem a criação de outros cenários. A Figura 3 apresenta alguns destes componentes.

Figura 3 – Elementos de composição dos cenários

Fonte: (LIMA et al., 2015)

Quanto aos personagens, foram subdivididos em uma grande variedade de combinações, com quatro tipos, de dois gêneros e nas etnias: parda, branca, negra, índia e amarela, sendo médicos, pacientes, enfermeiros e dentistas. Além disso, os pacientes possuem diferentes idades, como bebês, crianças e adolescentes, mas só para adultos e idosos que há possuem variação de biotipos como magro, normal e obeso (LIMA et al., 2015). Na Figura 4 é apresentado um exemplo.

Figura 4 – Modelos de Enfermeiras e Médicos

Fonte: (LIMA et al., 2015)

2.2.2 O *back-end* do Health Simulator

O *back-end* do Health Simulator foi desenvolvido na linguagem de C# e seu banco de dados em MSSQL (Microsoft SQL Server). A escolha desse banco de dados foi em razão das regras estabelecidas pela Universidade Feevale, pois o ambiente é hospedado nos servidores da instituição. Por se tratar de uma estrutura ampla, possui três subdivisões: componente de conhecimento, componente de administração e componente de comunicação, cada um com suas responsabilidades. Todo alicerce e coleta e manutenção dos dados para o funcionamento do simulador ocorre do *back-end* (BARROS, 2020).

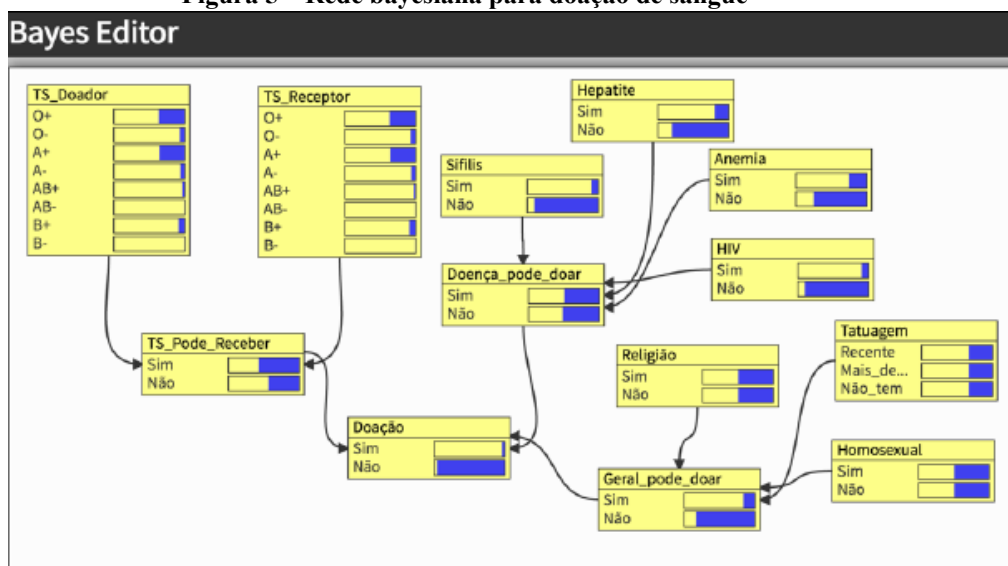
2.2.2.1 Modelagem do conhecimento

A modelagem do conhecimento é a parte responsável pela inteligência utilizada no ambiente, assim o especialista encarregado por essa fase, modela o conhecimento em um modelo estatístico, utilizando uma rede bayesiana (RB). As redes RB podem ser utilizadas para solução de problemas que envolvem incertezas (BEZ et al., 2017).

Uma RB é formada por um grafo acíclico dirigido, em que toda variável é representada por um nodo e as ligações de independência condicional são representadas pela topologia do grafo. As arestas representam relações de efeito entre as variáveis, e também possui uma tabela com a distribuição das probabilidades de condição para cada variável, considerando apenas a interferência dos pais diretos de cada nodo (LIMA et al., 2015; MARQUES; DUTRA, 2002; RUSSELL; NORVIG, 1995). O HS conta com um editor próprio para a criação de redes bayesianas, chamado de Bayes Editor (REIDEL et al., 2017). Apesar de existirem outros softwares disponíveis para criação e edição de redes bayesianas, segundo profissionais da saúde que trabalham no projeto, os mesmos tendem a ser complexos, trazendo dificuldade para o uso no ensino em saúde (MARONI, 2013).

A Figura 5 mostra uma RB para um procedimento de doação de sangue, desenvolvida por alunos do curso de biomedicina da Universidade Feevale, que descreve o problema que checa se uma pessoa pode realizar uma doação de sangue (BEZ et al., 2017).

Figura 5 – Rede bayesiana para doação de sangue



Fonte: (BEZ et al., 2017)

2.2.2.2 Interface de administração

A interface de administração é uma ferramenta web empregada para a elaboração de casos clínicos pelo professor da saúde. Todas as telas foram elaboradas levando em conta uma interface de fácil usabilidade. A Figura 6 mostra a interface, para adicionar casos de estudo. Nesta interface o professor determina quais as variáveis vão estar presentes na simulação, personagens, e o seu histórico clínico (BARROS, 2020).

Uma vez modelado o conhecimento, é possível criar um caso com os sinais e sintomas apresentados na rede bayesianas, que por consequência das probabilidades apontadas na RB, dirigirão o resultado de um caso clínico, dando início ao diagnóstico e condutas possíveis para a seleção, também é possível selecionar quais modelos 3D de personagens que participarão do jogo e adicionar exames complementares ao jogo (LIMA et al., 2015).

Figura 6 – Interface administrativa

Escolha um dos casos abaixo para editar, ou crie um novo caso, a partir do botão Novo ao lado.

Filtros de Pesquisa

Para Profissionais Universidade Associada Disciplina Associada

Nome	Tipo de Caso	Para Profissionais	Universidade Associada	Disciplina Associada	Classe Social	Cenário Associado	Personagens Associado	Descrição

Fonte: MELLO et al. (2015)

2.2.2.3 Serviço web de comunicação

Após o docente elaborar os casos clínicos utilizando a interface administração, eles são salvos no sistema de banco de dados e ficam acessíveis para o jogo através do serviço web de comunicação. Esta funcionalidade foi desenvolvida utilizando a arquitetura REST (*Representational State Transfer*), que é baseado no protocolo de comunicação HTTP, assim permitindo uma maior compatibilidade entre diferentes plataformas (HELWANGER; ROLIM; BARROS, 2015).

A comunicação entre o *front-end* e o *back-end*, deve-se ser iniciada pelo *front-end*, e é realizada em turnos em um intervalo definido de tempo, e a cada turno uma nova interação acontece, assim um método definido no *back-end* é executado (HELWANGER; ROLIM; BARROS, 2015).

2.2.2.4 SR do Health Simulator

O sistema de recomendação tem a função de indicar ao usuário ativo o grupo de itens que pode ser de seu maior interesse dentro de um grupo maior de itens recomendáveis, sendo esses itens casos clínicos e materiais para estudo que favoreçam o aprendizado do aluno que

utiliza o Health Simulator. Os resultados mostraram valores satisfatórios para o método de filtro colaborativo (CF) enquanto apontaram que alterações ainda são necessárias para que assim seja um sistema de recomendação que atenda aos requisitos e consiga entregar recomendações de qualidade (REIDEL et al., 2017).

3 INTRODUÇÃO AO ESTUDO DE SISTEMAS DE RECOMENDAÇÃO

A internet tem transformado a computação de uma forma revolucionária, produzindo e alastrando um grande volume de dados em questão de segundos em escala global. Com essa grande facilidade de acesso e abundante quantidades de informações, os usuários deparam-se com uma diversidade numerosa de opções. Esse cenário remete a uma capacidade de extrair informações, dificultando a obtenção de maior qualidade e acerto das decisões tomadas com a base de dados (COSTA; AGUIAR; MAGALHÃES, 2013).

Quando a quantidade de dados é superior ao poder de processamento de um sistema, origina o problema chamado de sobrecarga de informação (SCHICK; GORDON; HAKA, 1990), conforme alguns artigos alertam que a sobrecarga arruína a qualidade da decisão (HAHN; LAWSON; LEE, 1992), (PETERS et al., 1984). Em razão disso, surge a necessidade de procedimentos que possam proporcionar um tratamento antecipado, para favorecer a análise das informações e como resultado possa atender os objetivos do usuário. Assim com essa finalidade surgiu os primeiros sistemas de RS em 1990 (RESNICK et al., 1994), (MAES; SHARADANAND, 1995).

Os sistemas de recomendação são utilizados para conhecer usuários, e armazenar suas preferências e através dessas informações recomendar itens, que podem ser conteúdos, produtos ou até mesmo serviços de acordo com gostos e necessidades (REQUENA, 2015). Para esse objetivo, o sistema de SR utiliza dados dos usuários, como nacionalidade, idade e sexo e até mesmo dados sociais, como seguidos e seguidores em plataformas de rede sociais (BOBADILLA et al., 2013).

Um contexto ideal para aplicação de SR seria um aplicativo em que o usuário interage, sendo-lhe apresentada uma lista de itens na qual podem ser selecionados, para receber mais detalhes sobre um respectivo item ou interagir com ele (PAZZANI; BILLSUS, 2007). Seu propósito está basicamente focado em identificar a relevância de um determinado item para o usuário, e assim poder futuramente recomendá-lo para outro usuário com dados de usuários semelhantes (MARIA, 2017).

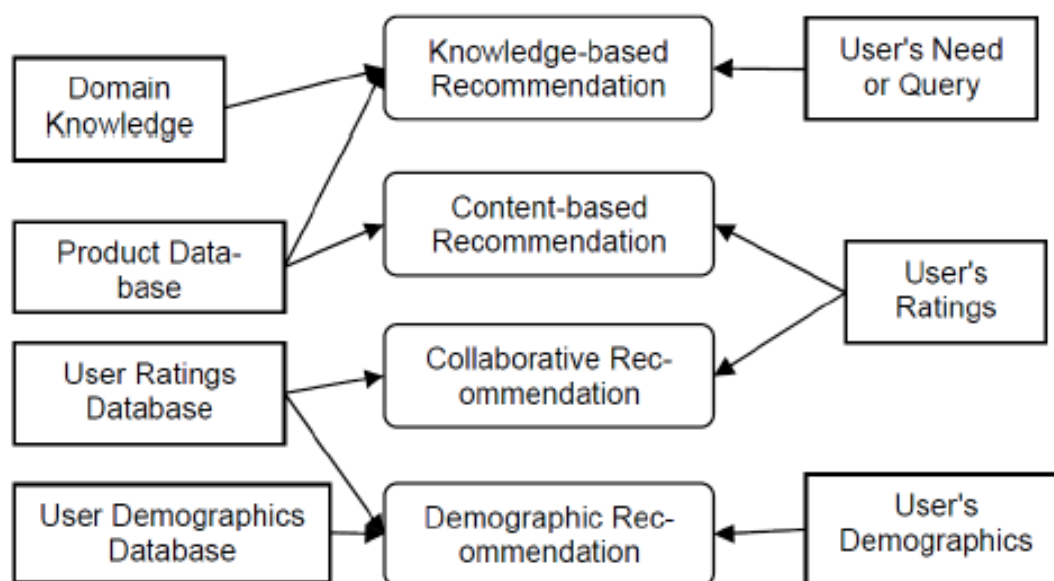
3.1 TÉCNICAS DE RECOMENDAÇÃO

Sistemas de recomendação podem utilizar diferentes fontes de informação para oferecer aos usuários recomendações de itens, buscando equilibrar fatores como novidade, gosto, acerto e estabilidade nas recomendações. Sendo assim, é necessário o uso de diferentes técnicas de

recomendação com vários tipos de abordagem, as que diferem principalmente pelo algoritmo de filtragem (REQUENA, 2015).

As técnicas de recomendação podem ser separadas com base nas suas origens de conhecimento, servindo como base para fazer recomendações, segundo BURKE (2007), propôs uma taxonomia, possuindo quatro classes diferentes, sendo elas: Baseadas em Conteúdo; Filtragem colaborativa, Demográfico e Baseadas em conhecimento, a Figura 7 exibe as divisões entre categorias e sua relação com a fonte do conhecimento.

Figura 7 – Técnicas de recomendação



Fonte: (BURKE, 2007)

Segundo BURKE (2002), também são apresentadas duas técnicas baseadas na utilidade e no modelo híbrido, que faz a junção de duas ou mais técnicas diferentes de recomendação para o usuário, assim seguindo o princípio que se utilize uma grande quantidade de dados do usuário para uma recomendação de qualidade. As próximas subseções irão detalhar cada uma das técnicas apresentadas na Figura 7.

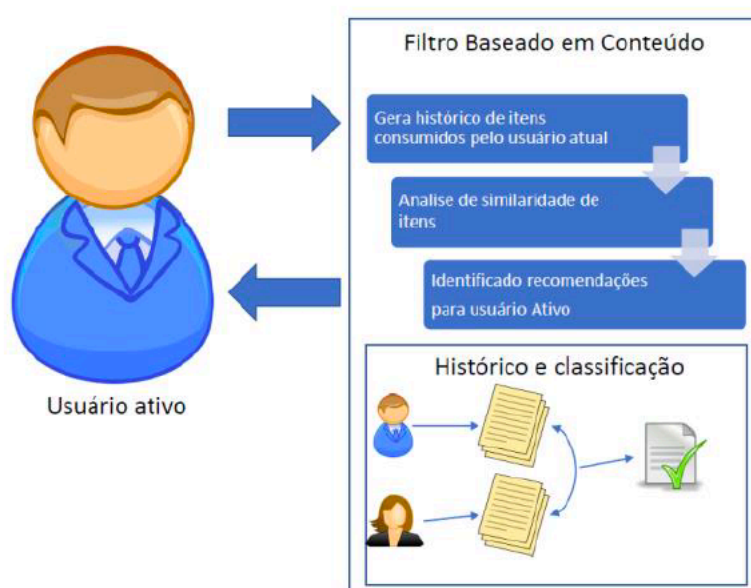
3.1.1 Sistemas de recomendação baseados em conteúdo

No SR baseado em conteúdo (BOURKOUKOU; EL BACHARI; EL ADNANI, 2017; BURKE, 2002) se busca reconhecer itens que possam ser de interesse do usuário tomando como base para a decisão itens que o usuário já tenha consumido e avaliado (AGGARWAL, 2016). Com a premissa de que um usuário tem a propensão de se interessar por itens similares, que

tenham alguma relação com os itens consumidos anteriormente e que o usuário mostrou uma boa aceitação (LOPS; DE GEMMIS; SEMERARO, 2011).

Na Figura 8 ilustra as recomendações baseadas em conteúdo, que usualmente são utilizadas informações dos itens destinados a recomendação, consequentemente determinados itens podem ser analisados, como exemplo: textos; imagens e sons, ou então relacionar as informações sobre o conteúdo. Com essas informações é possível que seja realizada a análise de similaridade entre os itens, que serviria como base para recomendação futura de itens semelhantes.

Figura 8 – Técnica de Filtro baseado em conteúdo



Fonte: (BARROS, 2020)

Geralmente para essa técnica é definido um conjunto de atributos para cada item manualmente dependendo do seu domínio. Porém em alguns casos, quando o item é um objeto textual, existem técnicas clássicas de recuperação de informação que devem ser empregadas para definir automaticamente quais atributos vão classificar o conteúdo de um item e identificar suas características. Como exemplo a técnica *TF-IDF term frequency-inverse document frequency* (REQUENA, 2015). Essa técnica consiste em identificar palavras que aparecem de forma frequente em um documento, mas que não sejam frequentes no restante do grupo de itens.

3.1.2 Sistemas de recomendação colaborativos

A técnica de filtro colaborativo otimiza a busca de itens para qual o usuário tem uma maior aceitação (BOBADILLA et al., 2013), (DASCALU et al., 2015), (GROLMAN et al., 2016), essa técnica surgiu nos anos 1990 (GOLDBERG et al., 1992), tentando reproduzir o resultado de uma recomendação de amigos ou familiares para um determinado item. Os algoritmos recomendam itens para o usuário, usando como base avaliações feitas por um grupo de usuários com gostos similares ao usuário ativo, para essa técnica não é necessário se ter informações sobre o item em questão, o que se armazena é a avaliação de usuários para cada item (LOPES, 2012). Possuem duas subcategorias que se ressaltam, quais sejam: recomendação baseada no usuário (*user based*) e baseada no item (*item based*).

Na subcategoria do tipo *user based*, é criado um agrupamento de indivíduos, com o nome de grupo dos vizinhos, que possui o perfil similar com o usuário ativo. Assumindo-se que a avaliação de um item pelo usuário ativo será parecida com ao grupo de vizinhos correspondentes, sendo assim os indivíduos com suas preferencias similares à do usuário ativo que gostam de um respectivo item, é possível dizer que irá agradar também ao usuário em questão (LOPES, 2012).

Na subcategoria do tipo *item based*, são traçados quais itens são similares entre os itens usando as avaliações de usuários, com isto quando o sistema detecta o interesse do usuário por um item, ele pode recomendar um item que é similar a este. Também é possível que o usuário possa avaliar o item positivamente ou negativamente e isto também pode ser utilizado nas decisões de recomendação (LAMPROPOULOS; TSIHRINTZIS, 2015).

3.1.2.1 Função da similaridade

Um componente da filtragem colaborativa que serve tanto para item e usuário é a função da similaridade a ser utilizada que serve para fazer a escolha dos indivíduos da agregação dos usuários ou itens em questão. Uma das funções de similaridade que geralmente são utilizadas pelos algoritmos é a similaridade do cosseno. Como pode ser observado na Eq1 que pode servir tanto para item quanto para usuários.

$$\text{cosine_sim}(u, v) = \frac{\sum_{i \in I_{uv}} r_{ui} \cdot r_{vi}}{\sqrt{\sum_{i \in I_{uv}} r_{ui}^2} \cdot \sqrt{\sum_{i \in I_{uv}} r_{vi}^2}} \quad (1)$$

O grau da similaridade entre os “itens” u , v é o resultado pelo cosseno do ângulo formado pelos vetores equivalentes, quando os valores são normalizados os vetores apresentam um módulo unitário e o cálculo de similaridade se torna igual ao produto escalar, e o grau máximo de similaridade é 1, que corresponde a vetores no mesmo sentido e direção (CATAE, 2012).

4 MACHINE LEARNING

Machine Learning (ML) ou aprendizado de máquina em português, é uma divisão da inteligência artificial que, em linhas gerais, tem como principal objetivo aprender com dados e informações do histórico utilizando o método de treinamento e tratamento de dados. Os sistemas que utilizam aprendizado de máquina, aprendem através de experimentos e, com o passar do tempo, podem ser melhorados para prever informações fundamentadas em questionamentos baseados no aprendizado alcançado (KOREN; BELL, 2015).

Segundo o estudo do Arthur Samuel (2015), ele define o termo *Machine Learning* como a área de estudo que entrega aos computadores a habilidade para aprender sem ser programado evidentemente para isso. ML possui uma grande quantidade de algoritmos que podem ser utilizados para prever informações que são de importância e através dessas informações que é definido quais algoritmos melhores a serem utilizados. Algoritmos de ML geralmente são classificados em duas subdivisões, sendo elas: O aprendizado supervisionado; E aprendizado não supervisionado.

Na divisão supervisionado as informações utilizadas no treinamento são marcadas com objetivo de aprender uma regra que vai servir para estruturar as entradas e saídas do modelo aplicado. Por outro lado, a subdivisão não supervisionada, não marca as informações, o algoritmo que é responsável por identificar automaticamente os padrões que existem no conjunto de informações analisadas (REIDEL et al., 2017). Não é sempre que os algoritmos vão servir para a solução desejada, ou seja, o gerenciamento dos modelos de ML exige uma intervenção humana para manter os algoritmos sempre atualizados, e atendendo aos requisitos e com isto gerando os resultados desejados (KOREN; BELL, 2015).

4.1 APLICAÇÕES DE MACHINE LEARNING

ML pode ser aplicada para problemas distintos e para inúmeras funcionalidades, mas é geralmente utilizado na área de desenvolvimento de software, pois pode aprender o comportamento de um usuário, e com isto posteriormente prever quais serão as ações deste usuário. O aprendizado de máquina também pode ser utilizado para detecção de *Spam*, que são e-mails indesejados que muitas as vezes lotam a caixa de entrada. Outra aplicação de ML que atualmente está em alta é o reconhecimento de voz, feito pelos assistentes virtuais da Apple e do Google (dentre outras), que usam essa técnica para aprender o comportamento e escolhas do

usuário e usam uma busca avançada baseada em computação na nuvem para analisar esses dados (MITCHELL; MICHALSKI; CARBONELL, 2013).

No mercado de ações também é amplamente utilizado ML, em algoritmos para auxiliar investidores nas suas tomadas de decisões, assim ajudando na predição de resultados baseados em dados anteriores. Outra área que utiliza amplamente algoritmos de ML é a robótica, atualmente não é difícil ver nas redes sociais ou até mesmo em sites de tecnologia tipos de robôs que imitam movimentos de animais ou seres humanos, que se adaptam ao ambiente. A maioria desses robôs utiliza aprendizado de máquina para aprender com o ambiente ao seu redor sempre se baseando em informações que são coletadas usando a fonte de predição e tomada de decisão (KOREN; BELL, 2015).

Além desses, outro exemplo apontado por KOREN; BELL, (2015) são lojas de varejo on-line que utilizam cartões de fidelidade para obter informações sobre o comportamento de seus clientes e usar essas informações posteriormente para descrever um padrão e sugerir ações. Computadores com grande poder de processamento estão ajudando o campo da medicina, como exemplo deste cenário é o Watson, um computador construído pela IBM para auxiliar médicos do mundo inteiro a diagnosticar doenças. O objetivo é que o médico ainda seja o tomador de decisão, cabendo a ele fazer o diagnóstico final do paciente, porém inúmeras vezes existem várias causas para sintomas em pacientes, e o Watson tem como objetivo ajudar os médicos a identificar de forma muito mais rápida e eficiente a causa dos sintomas.

4.2 SISTEMAS DE RECOMENDAÇÃO COM USO DE MACHINE LEARNING

Segundo BRUNIALTI et al. (2015b) sistemas de recomendação baseados em conteúdo, são uma área na qual as estratégias de Machine Learning podem ser aplicadas com sucesso, principalmente no filtro colaborativo baseado em conteúdo. Para o artigo de Lucas (2018), estudos que implementaram os classificadores para o aprendizado do perfil do usuário alcançaram bom desempenho, principalmente com a técnica de *Machine Learning Support Vector Machine* (SVM), que usando a estratégia usual de construção de classificadores em SR é utilizado para decidir se um artigo é relevante ou não para outros usuários do sistema de recomendação. No estudo compreende-se que as iniciativas na interseção dessas áreas têm produzidos resultados promissores fornecendo um incentivo para que mais pesquisas sejam empregadas nessa área.

No artigo de WANG et al. (2014) foi desenvolvido com a técnica de filtro de conteúdo híbrido para gerar uma recomendação de filmes, combinando técnicas de redução dimensional

com algoritmo de grupos. Na falta de dados para a recomendação, é usado a técnica para selecionar filmes semelhantes com base em sua avaliação, que é vital para gerar recomendações de filme de qualidade. Para isso é usado o algoritmo de ML K-means, a fim de agrupar usuários similares.

Já no artigo de ZAHIR; EL QADI (2016), é apresentado um plano de recomendação baseado na similaridade de identificação, com o objetivo de construir um modelo, que reconhecesse similaridades usando apenas *SQL statement*, para isto foram utilizadas técnicas de ML para melhorar a detecção de similaridades. A classificação baseada em regras de associação, tem um melhor potencial para fornecer uma previsão exata em comparação a SVM e Naive Bayes.

Na pesquisa de SINGHAL; SINHA; PANT (2017) é utilizado ML com uso de técnicas de Deep Learning na área de sistemas de recomendação, sendo discutidos trabalhos de pesquisa em três categorias como: sistema de recomendação baseado em conteúdo, filtragem colaborativa e sistemas híbridos. Concluiu que se obteve um melhor resultado usando Deep Learning na abordagem do tipo filtro colaborativo e também se concluiu que a maioria dos trabalhos era direcionada para indústria de entretenimento, como na recomendação de filmes e músicas.

Um algoritmo de ML amplamente utilizado é o KNN *k-nearest neighbors algorithm* também conhecido na língua portuguesa como o algoritmo dos grupos vizinhos. O centro de seu funcionamento está em descobrir o vizinho mais próximo de uma distância, na pesquisa de FERRERO (2009), foi utilizado o algoritmo KNN para classificar e selecionar o melhor método dentre eles: branqueamento, ácido ascórbico e sulfato de sódio, para o controle de escurecimento enzimático em frutas e hortaliças conservadas em temperatura ambiente, refrigeração e congelamento.

O algoritmo KNN é um algoritmo do tipo *lazy*, e seu intuito geral consiste em encontrar os k exemplos rotulados mais próximos do exemplo não classificado, e com base no rótulo desses exemplos mais próximos, é tomada a decisão relativa à classe do exemplo não rotulado. Os algoritmos do tipo KNN exigem uma pequena demanda de processamento na sua etapa de treinamento, em compensação para rotular um novo exemplo é relativamente alto, visto que no pior dos casos, esse exemplo deverá ser comparado com todos os exemplos inclusos no conjunto de exemplos de treinamento (FERRERO, 2009).

Outro algoritmo bastante utilizado em sistemas de recomendação que utilizam *Machine Learning* é o SVD (*singular value decomposition*) que é uma técnica de fatoração de matriz (real ou complexa), essa técnica pode ser aplicada em diversos cenários, desde o processamento de sinais, por exemplo simplificando imagens para facilitar seu processamento posteriormente, ou também em cenários como em estatística, em que ganhou bastante popularidade por apresentar resultado superiores a outros algoritmos (FUKS, 2013).

Segundo o estudo de SILVA; ROCHA; MOURÃO (2016), são avaliados métodos de recomendações para filmes em que apesar da popularidade de métodos de fatoração de matriz ou *Matrix Factorization* (MF) em sistemas de recomendação, não é clara a distinção prática entre eles em domínios reais. Neste contexto foi comparado diferentes métodos de recomendação baseados em MF, sendo proposto uma taxonomia para métodos de MF baseada em cinco dimensões associadas ao processo de recomendação, com esta classificação foi possível analisar as semelhanças entre técnicas existentes de MF. Posteriormente foi avaliado que técnicas baseadas em SVD se mostraram mais recomendadas com filmes com notas médias maiores em sistemas de recomendação.

Existe o algoritmo chamado de SVD++ que se refere a um algoritmo otimizado baseado no SVD, que teoricamente aumenta a precisão da previsão através do feedback explícito. A ideia é que através de banners seja produzida implicitamente uma interação com o usuário, que auxilia na taxa de predição. A inclusão do feedback implícito é feita através da inclusão de fatores implícitos que se somam aos fatores do usuário (FUKS, 2013).

4.2.1 Algoritmo KNN

O algoritmo K-nearest neighbors conhecido também como KNN e em português chamado de k-vizinhos mais próximos, é derivado da abordagem de vizinhos mais próximos. Sua fórmula pode ser observada nas Eq2 abaixo, sendo que para cada um desses vizinhos que são agregados para calcular uma estimativa que é necessariamente menor ou igual a K, sendo K o número de vizinhos para se considerar a agregação (Surprise, 2020).

$$\begin{aligned}
\hat{r}_{ui} &= \frac{\sum_{v \in N_i^k(u)} \text{sim}(u, v) \cdot r_{vi}}{\sum_{v \in N_i^k(u)} \text{sim}(u, v)} \\
\hat{r}_{ui} &= \frac{\sum_{j \in N_u^k(i)} \text{sim}(i, j) \cdot r_{uj}}{\sum_{j \in N_u^k(i)} \text{sim}(i, j)}
\end{aligned} \tag{2}$$

Na Eq2, têm duas fórmulas sendo a primeira com o a função da similaridade “u” e “v” para representar os valores baseados na similaridade de item e na segunda equação os valores de “i” e “j” para os valores de usuário.

4.2.2 Algoritmo SVD

O algoritmo SVD, utiliza o princípio da fatoração de matrizes, sua predição é definida como é mostrado na Eq3.

$$\sum_{r_{ui} \in R_{train}} (r_{ui} - \hat{r}_{ui})^2 + \lambda (b_i^2 + b_u^2 + ||q_i||^2 + ||p_u||^2) \tag{3}$$

E sua minimização utiliza fatoração de matriz, através de um gradiente descendente estocástico como a Eq4, essas etapas são executadas em todas as classificações da composição e repetidas pelo número de *epochs*. Sua linha base é inicializada com o valor de *init mean* e os fatores de itens são inicializados aleatoriamente de acordo com a distribuição normal que pode ser ajustada, que também pode ser ajustada a taxa de aprendizagem e o termo de regularização (KOREN; BELL; VOLINSKY, 2009).

$$\begin{aligned}
b_u &\leftarrow b_u + \gamma(e_{ui} - \lambda b_u) \\
b_i &\leftarrow b_i + \gamma(e_{ui} - \lambda b_i) \\
p_u &\leftarrow p_u + \gamma(e_{ui} \cdot q_i - \lambda p_u) \\
q_i &\leftarrow q_i + \gamma(e_{ui} \cdot p_u - \lambda q_i)
\end{aligned} \quad (4)$$

Uma extensão do algoritmo SVD é os algoritmos SVD++ que leva em consideração as classificações implícitas e sua predição pode ser definida pela Eq5, sendo que essa classificação implícita descreve o fato de que um usuário u avaliou um item j , independente do valor da classificação, se usuário u é desconhecido, então “ b_u ” e os fatores “ p_u ” são considerados zero. O mesmo se aplica para o item i com “ b_i ”, “ q_i ”, “ y_i ”. E da mesma forma que o SVD, os parâmetros são aprendidos usando SGD para erro quadrático regularizado (KOREN; BELL; VOLINSKY, 2009).

$$\hat{r}_{ui} = \mu + b_u + b_i + q_i^T \left(p_u + |I_u|^{-\frac{1}{2}} \sum_{j \in I_u} y_j \right) \quad (5)$$

4.3 MÉTRICAS

Conforme os estudos dos capítulos anteriores, um aspecto importante para a avaliação de um sistema de recomendação são quais as métricas utilizadas, assim possuindo uma forma de medir o seu desempenho. Existem diversas técnicas e formas de medir quão próximas à realidade estão as recomendações e predições de um sistema. Salienta-se, também que alguns autores empregam mais de uma métrica para avaliar seus estudos.

Métricas de acurácia da predição buscam demonstrar quão próximas às avaliações do usuário para um respectivo item estão as predições do sistema. Uma métrica utilizada é a *Mean Absolute Error* (MAE) que esteve presente nos estudos: (PORTUGAL; ALENCAR; COWAN, 2018), (WANG et al., 2014). A Eq6 demonstra o cálculo do valor MAE, onde N é o valor de itens avaliados, \hat{p}_i é o valor previsto pelo SR e r_i é a avaliação feita pelo usuário quanto mais perto do valor zero, mais exato é o sistema de recomendação (WANG et al., 2014).

$$MAE = \frac{\sum |\tilde{P}_{ij} - r_{ij}|}{M}$$

Exemplo da equação:

predicted rating	actual rating	error
5	3	2
4	1	3
5	4	1
1	1	0

MAE = (2+3+1+0)/4 = 1.5 (6)

Outra métrica com a finalidade de medir a acurácia da predição é chamada de *Root Mean Squared Error* (RMSE) encontrada nos estudos: (PORTUGAL; ALENCAR; COWAN, 2018), (THAI-NGHE et al., 2010)(ZAHIR; EL QADI, 2016), é mostrada na Eq7. Quando comparada à métrica MAE, possui a diferença de penalizar mais intensamente erros maiores, enquanto a métrica MAE trata todos os erros com o mesmo peso (CHAI; DRAXLER, 2014).

$$RMSE = \sqrt{\frac{\sum_{ui}(r_{ui} - \hat{r}_{ui})^2}{n}}$$

Exemplo da equação:

predicted rating	actual rating	error ²
5	3	4
4	1	9
5	4	1
1	1	0

RMSE = $\sqrt{(4 + 9 + 1 + 0)/4} = 1.87$ (7)

Também nos estudos de PORTUGAL; ALENCAR; COWAN (2018) e THAI-NGHE et al. (2010), foram utilizadas métricas para avaliar a acurácia da ordem de preferência proposta por um sistema de recomendação, denominada de TOP-N itens apresentados por um sistema, que podem ser considerados como uma lista ordenada de itens. Com isto é preciso verificar o quão parecido a ordem dos itens é em comparação à uma segunda lista, que representa os mesmos itens ordenados pelo usuário de acordo com sua preferência (BARROS, 2020).

5 TRABALHO RELACIONADO

Este capítulo faz uma breve recapitulação ao trabalho relacionado, seguindo para como foi desenvolvido a definição do *dataset* e as etapas que foram executadas pelo autor nesse conjunto de dados para que se tornassem prontos para serem utilizados na fase de aplicação. Também será apresentado todos os atributos existentes e seus significados, passando-se ao leitor as informações presentes no *dataset*.

O estudo desenvolvido por REIDEL et al. (2017), teve como objetivo desenvolver um sistema de recomendação de casos clínicos e itens de estudo que auxiliam no aprendizado do estudante que utiliza o Health Simulator, fazendo o uso de estratégias de recomendação que possam ser aplicadas na área da educação na saúde. Dessa forma, auxiliando o aluno pedagogicamente, e atuando de forma presente durante e por fim avaliando o desempenho para a recomendação de itens pertinentes ao seu aprendizado.

As técnicas de recomendação que foram utilizadas para o desenvolvimento do trabalho foram os métodos de filtros colaborativos (CF), filtros baseados em conteúdo (CBF) e os algoritmos de híbridos mistos, adaptativos e intersecção dos conjuntos. Outro ponto relevante no trabalho é da correta escolha das métricas para a avaliação das recomendações geradas pelo sistema de recomendação, na circunstância que uma escolha de métrica inadequada pode causar impacto negativo na qualidade das recomendações. Para contornar essa questão os métodos escolhidos foram avaliados em precisão de predição, quanto à acurácia de classificação.

Os resultados mostraram que o algoritmo de filtro colaborativo obteve melhores resultados, ao ser comparado ao filtro baseado em conteúdo. O método usando o CBF não conseguiu gerar uma grande quantidade de recomendação, gerando métricas que como a sensibilidade e *miss rate* sejam afetadas diretamente e apresentem resultados ruins, como consequência o método híbrido também sofreu e gerou resultados com baixa precisão. Conclui-se que é preciso fazer alterações ao método que foi proposto, para que seja gerado um maior número de recomendações e que seja pertinente ao usuário. Por outro lado, o método de CF proposto apresentou resultados que foram satisfatórios, com uma precisão próxima a 60% e sensibilidade próxima a 70% e MAE por volta de 0.85.

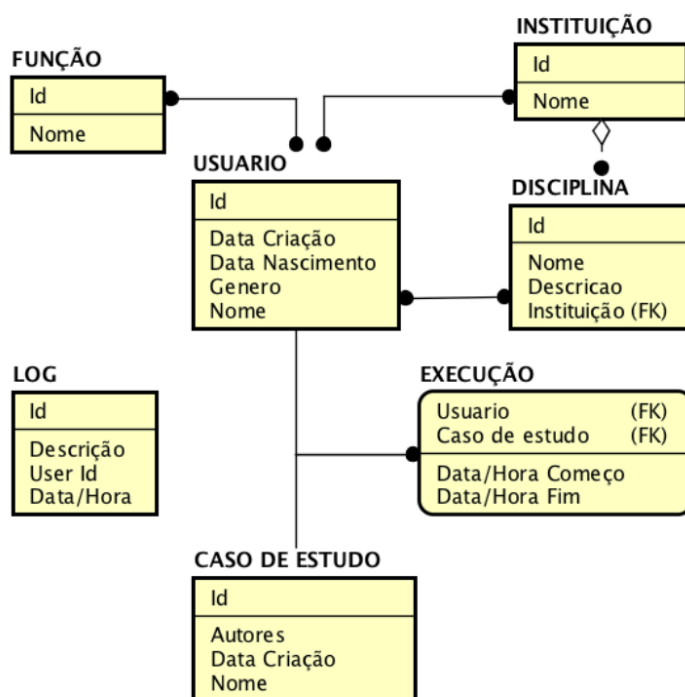
A pesquisa realizada neste trabalho apontou que os sistemas de recomendação de modelo geral não são indicados para aplicações em plataformas voltadas para educação, especialmente pelo fato de não abordar questões importantes para essa área. Desta forma não se deve encarar os algoritmos desenvolvidos como uma solução definitiva para o uso no Health

Simulador. Eles são um passo inicial para a construção de um sistema de recomendação capaz de ajudar o aluno durante a execução de casos clínicos no simulador.

5.1 MODELO JÁ PROPOSTO NO HS

Segundo REIDEL et al. (2017), o simulador antes de sua implementação já continha diversas entidades responsáveis por representar os objetivos do sistema, como (casos de estudo, disciplinas, usuários, entre outros) e o armazenamento desses dados. A Figura 9 mostra o modelo entidade-relacionamento antes das alterações propostas pelo autor, levando em consideração que por motivos de segurança e simplificação são retratadas somente as entidades e atributos que são julgados como pertinentes para o sistema de recomendação já proposto.

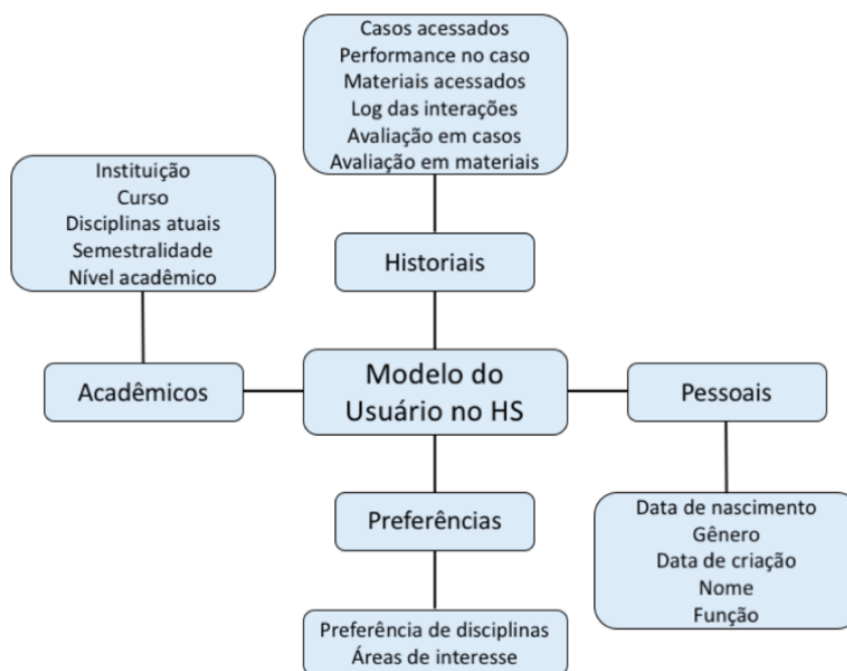
Figura 9 – Modelo ER original simplificado



Fonte: (REIDEL et al., 2017)

Seguindo a implementação de REIDEL et al. (2017), a Figura 10 apresenta um modelo de um usuário para o sistema de recomendação proposto por ele, que segundo o autor foi implementado de forma simples, levando em conta dados de maior relevância considerando o objetivo do simulador, que contém exclusivamente informações pessoais.

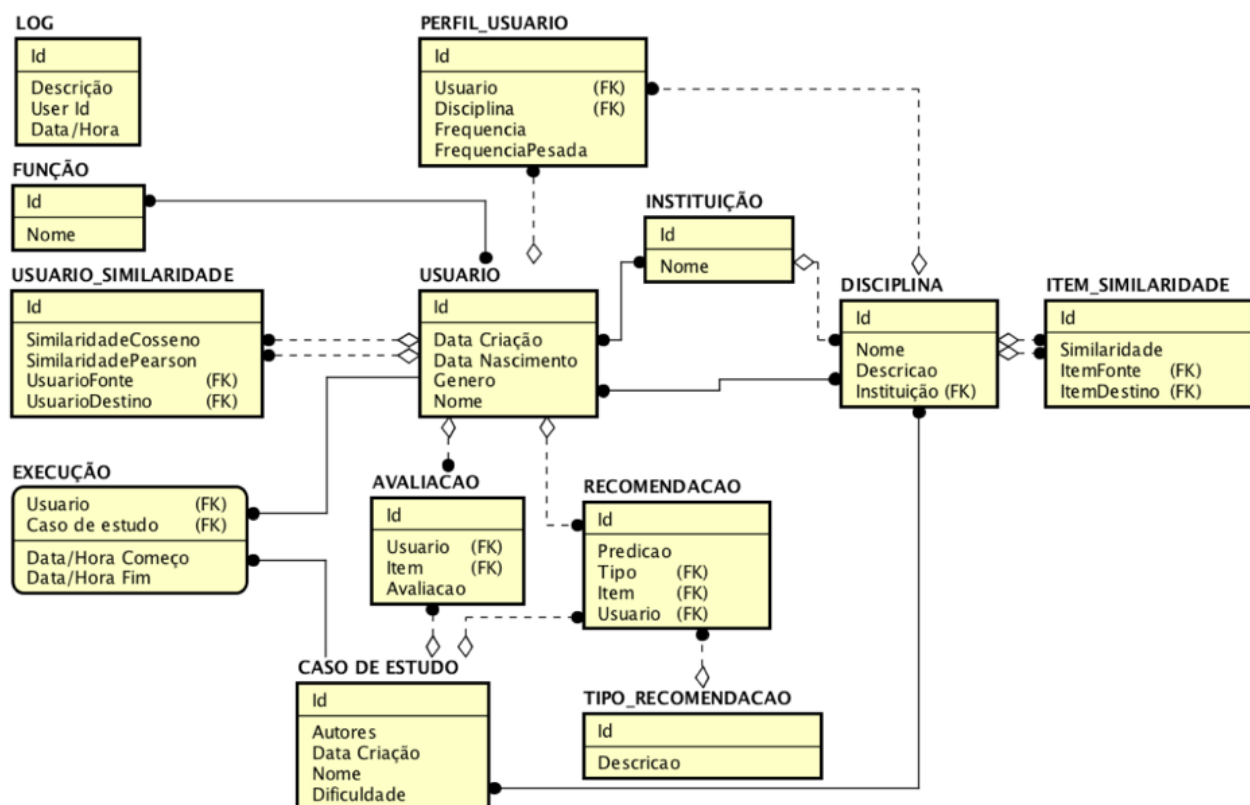
Figura 10 – O modelo de usuário no HS



Fonte: (REIDEL et al., 2017)

Através disto chega-se nas alterações propostas por REIDEL et al. (2017) mostradas na Figura 11 e implementadas no banco de dados do HS, que por sua vez utiliza o Sistema de Gerenciador de Banco de Dados (SGBD) o SQL Server.

Figura 11 – Modelo ER resultante das alterações ao BD



Fonte: (REIDEL et al., 2017)

5.1.1 Base de dados

Na época do desenvolvimento do trabalho de REIDEL et al. (2017), o HS ainda se encontrava em desenvolvimento e por isto não contava com um sistema para armazenar a interação do usuário com o simulador, além de que não existiam forma para que o usuário do HS pudesse dar um *feedback* ao conteúdo que está consumindo, por assim sendo impossível de efetuar recomendações das preferencias do usuário. Em razão disto, decidiu-se utilizar uma base de dados aberta para abastecer às entidades adicionadas ao sistema, para que se pudesse ter informações para utilizar na construção do SR.

A base de dados chamada de *dataset* escolhida foi disponibilizada pelo (GROUPLENS, 2017), sendo ela a versão menor, recomendada para desenvolvimento segundo o GROUPLENS ela conta com 100.024 avaliações realizadas por 671 usuários a 9125 filmes que referenciam 1995 *tags*. Esses dados dispõem de datas entre 9 de janeiro de 1995 a 10 de outubro de 2016 (HARPER; KONSTAN, 2015).

O *dataset* é composto por quatro arquivos CSV sendo eles, *links.csv*, *movies.csv*, *ratings.csv* e *tags.csv*. A seguir é descrito cada arquivo e seus atributos:

Estrutura do arquivo de dados de classificação (*ratings.csv*), cada linha deste arquivo após a primeira linha, definida como a linha do cabeçalho, representa uma avaliação de um filme por um determinado usuário, sendo que cada usuário avaliou no mínimo 20 filmes. Ela possui o seguinte formato visto na Figura 12:

Figura 12 – Cabeçalho e conteúdo do arquivo ratings.csv

```
userId,movieId,rating,timestamp
1,1,4.0,964982703
1,3,4.0,964981247
1,6,4.0,964982224
1,47,5.0,964983815
1,50,5.0,964982931
1,70,3.0,964982400
```

Fonte: Elaborada pelo autor

As linhas neste arquivo são ordenadas primeiro por *userId*, em seguida pelo *movieId*, as avaliações chamadas de ratings são feitas em uma escala de 5 estrelas, com incrementos de meia escala (0,5 a 5,0 “estrelas”), o *timestamp* representa horas e segundos a partir da meia noite do Tempo Universal Coordenado (UTC) de 1 de janeiro de 1970.

A estrutura do arquivo de dados de filmes (*movies.csv*) que pode ser observada na Figura 13 abaixo, possui a primeira linha sendo seu cabeçalho seguindo a mesma lógica do anterior, e que segue para os demais arquivos. Cada linha seguida do cabeçalho apresenta um filme que possui seu *movieId*, seu *title* que é inserido manualmente ou importado no Moviedb pelo MovieLens e por fim *genres* que é uma lista separada por barras verticais que são selecionadas a partir de uma lista que pode ser vista abaixo:

- *Action*;
- *Adventure*;
- *Animation*;
- *Children`s*;
- *Comedy*;

- *Crime;*
- *Documentary;*
- *Drama;*
- *Fantasy;*
- *Film-Noir;*
- *Horror;*
- *Musical;*
- *Mystery;*
- *Romance;*
- *Sci-Fi*
- *Thriller;*
- *War;*
- *Western;*
- *(no genres listed).*

Figura 13 – Cabeçalho e conteúdo do arquivo movies.csv

```
movieId,title,genres
1,Toy Story (1995),Adventure|Animation|Children|Comedy|Fantasy
2,Jumanji (1995),Adventure|Children|Fantasy
3,Grumpier Old Men (1995),Comedy|Romance
4,Waiting to Exhale (1995),Comedy|Drama|Romance
5,Father of the Bride Part II (1995),Comedy
6,Heat (1995),Action|Crime|Thriller
```

Fonte: Elaborada pelo autor

Seguindo, temos a estrutura do arquivo de dados *tags* (tags.csv) que pode ser observado na Figura 14. Cada linha após o cabeçalho representa uma *tag* aplicada a um filme por um usuário, que estão ordenadas primeiro pelo *userId* que representa qual usuário aplicou a *tag* e em seguida pelo *movieId* que representa a qual filme pertence. A *tag* é uma palavra ou frase curta determinada pelo usuário e por último o *timestamp*.

Figura 14 – Cabeçalho e conteúdo do arquivo tags.csv

```

userId,movieId,tag,timestamp
2,60756,funny,1445714994
2,60756,Highly quotable,1445714996
2,60756,will ferrell,1445714992
2,89774,Boxing story,1445715207
2,89774,MMA,1445715200
2,89774,Tom Hardy,1445715205

```

Fonte: Elaborada pelo autor

E, por último, a estrutura de *links* (links.csv), que pode ser observada na Figura 15, possui identificadores que podem ser usados para vincular à outras fontes de dados, cada linha desde arquivo após seu cabeçalho representa um filme, em que *movieId* é um identificador de filmes utilizado pelo movielens, *imdbId* é o identificador utilizado para filmes no IMDB, e por último o *tmdblId* é um identificador de filmes usado pelo TheMovieDb.

Figura 15 – Cabeçalho e conteúdo do arquivo links.csv

```

movieId,imdbId,tmdbId
1,0114709,862
2,0113497,8844
3,0113228,15602
4,0114885,31357
5,0113041,11862
6,0113277,949

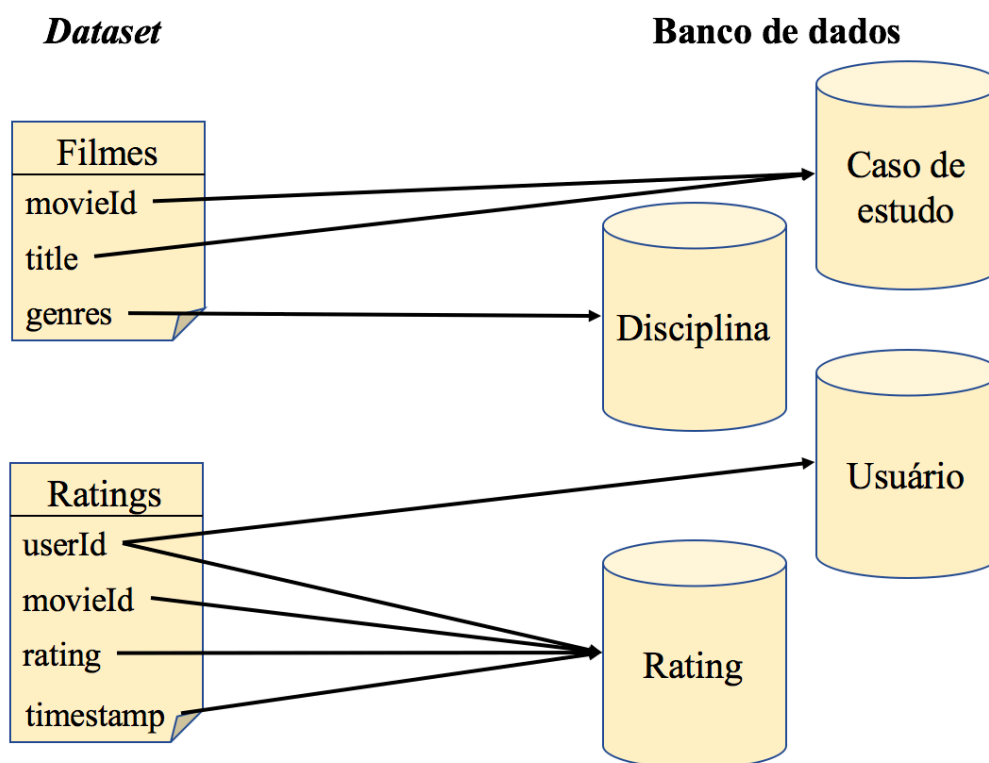
```

Fonte: Elaborada pelo autor

5.2 A INSERÇÃO DOS DADOS NO HS

Conforme REIDEL et al. (2017) foi escolhido para a inserção somente os arquivos: movies.csv e ratings.csv e iniciou-se de forma que os dados separados através de vírgulas foram traduzidos para comandos SQL adequados ao BD do HS. O conteúdo do arquivo de *movies* foi inserido na entidade Caso de Estudo, com o título de filme sendo o nome do caso de estudo. Já os gêneros foram incluídos na entidade Disciplina, mantendo a ligação com Caso de Estudo.

Figura 16 – Mapeamento entre dataset e BD



Fonte: (REIDEL et al., 2017)

Por fim, cada valor do arquivo *ratings* foi adicionado a Entidade Rating, e seu valor de *userId* também foi adicionado a entidade Usuário, para manter sua relação com a entidade Rating. A Figura 16 demonstra o mapeamento entre o *dataset* e o BD do HS conforme REIDEL et al. (2017) apresentou em seu estudo.

6 IMPLEMENTAÇÃO

Este capítulo apresenta a implementação do sistema de recomendação desenvolvido neste trabalho, como a definição das ferramentas, como os algoritmos foram implementados e quais foram as etapas executadas nesse conjunto de dados, para que se tornassem prontos para serem utilizados na fase de aplicação dos algoritmos de ML. É importante destacar que o HS se encontra em fase de desenvolvimento, então para o presente trabalho e para o fim de comparação foi utilizado a mesma estrutura do *dataset* implementada por REIDEL et al. (2017).

Para a realização deste estudo foi escolhido e definido que a linguagem de programação Python vai ser utilizada para a implementação do mesmo, por ser uma linguagem que vem sendo muito utilizada por razões de aparecerem em estudos do referencial teórico. Por ser uma linguagem de alto nível voltada para o uso geral e fácil de aprendizado, programadores com menos experiência precisam se preocupar menos com a complexidade do próprio código, podendo focar buscar soluções para o real problema.

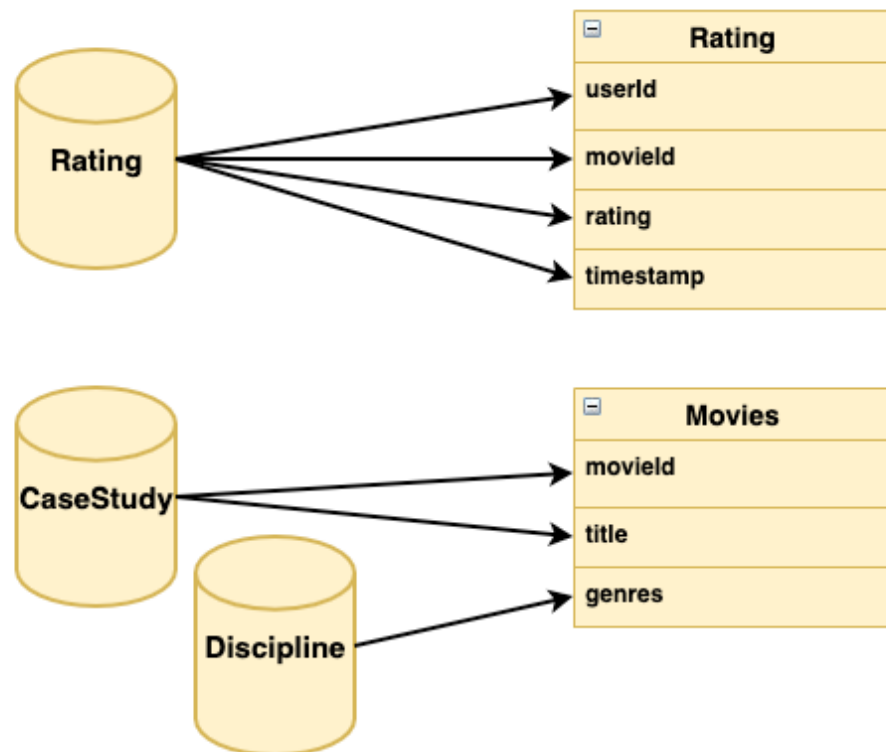
Umas das principais razões também de Python ser uma linguagem muito popular em ML e ciência de dados, são as bibliotecas, que é um grupo de código que pode ser importado para o seu ambiente de desenvolvimento e com isso estender suas funções como a biblioteca Surprise. Existe uma grande diversidade de bibliotecas para todos os aspectos do aprendizado de máquina que por sua vez abstrai e simplifica muitas funções já implementadas.

Seguindo esse raciocínio uma das bibliotecas escolhidas para esse estudo é a Surprise que fornece uma coleção de algoritmos de previsão de classificação, como também a implementação de algoritmos de métodos de vizinhança, e também baseados em fatoração de matriz como o SVD e SVD++, assim como medidas de similaridade. Além disto conta com ferramentas para avaliar, analisar e comparar o desempenho dos algoritmos de ML. O Surprise foi projetado para ser útil para pesquisadores que desejam explorar novas ideias de recomendação e também servindo como recurso de aprendizagem para usuários e alunos menos experientes em razão da sua detalhada documentação.

6.1 RECUPERANDO BASE DE DADOS

Para fins de comparação e precisão do trabalho aqui apresentado, foi utilizado a mesma base de dados que foi inserido no HS pelo REIDEL et al. (2017) como foi descrito no capítulo de inserção de dados. Com base nisto foi escolhido e definido que os dados precisariam estar no formato de csv, então foi necessário transcrever esses dados que estão no banco de dados do HS para um arquivo csv como mostra o diagrama na Figura 17.

Figura 17 – Mapeamento entre BD e dataset



Fonte: Elaborada pelo autor

As consultas SQL que quando executadas no banco de dados do HS geram os arquivos de dados csv, que podem ser observadas abaixo nas Figuras 18 e 19.

Figura 18 – Consulta SQL para o arquivo ratings.csv

```

select idUser as userId,
idCaseStudy as movieId,
format(rating, '0.0') as rating,
DATEDIFF(SECOND, '1970-01-01', timestamp) AS 'timestamp'
from Rating
  
```

Fonte: Elaborada pelo autor

Figura 19 – Consulta SQL para o arquivo movies.csv

```

create view vwMovies as
select CaseStudy.Id as movieId, CaseStudy.Name as title, Discipline.Name as genres
from DisciplineCaseStudy
inner join Discipline on Discipline.Id = DisciplineCaseStudy.Discipline_Id
inner join CaseStudy on CaseStudy.Id = DisciplineCaseStudy.CaseStudy_Id

select * from vwMovies

SELECT movieId, title, genres = STUFF(
    (SELECT '|' + genres
     FROM vwMovies t1
     WHERE t1.movieId = t2.movieId
     FOR XML PATH (''))
    , 1, 1, '') from vwMovies t2
group by movieId, title;

drop view vwMovies

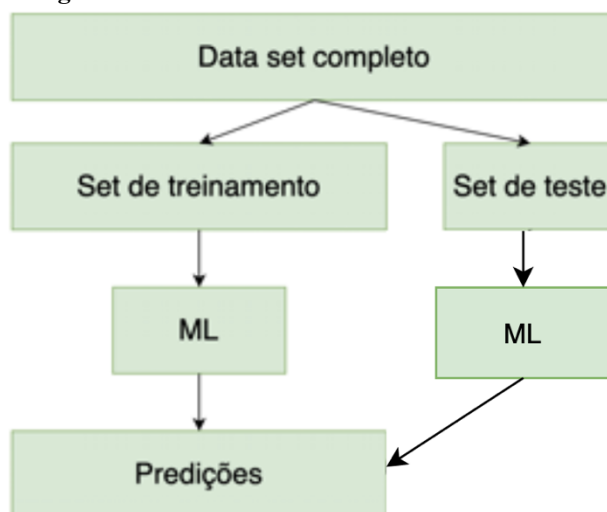
```

Fonte: Elaborada pelo autor

6.2 IMPLEMENTAÇÃO DOS ALGORITMOS

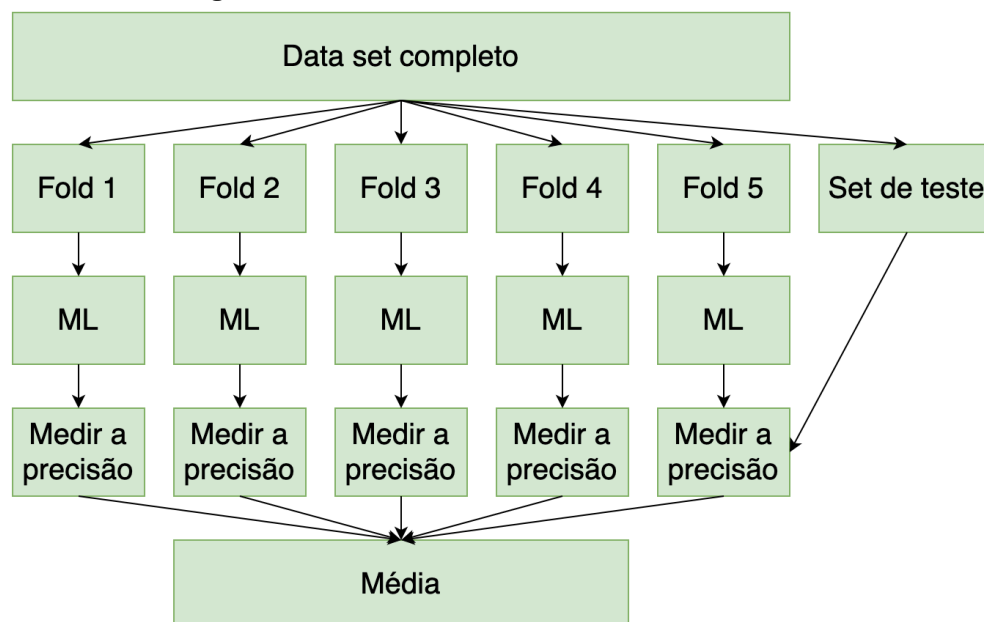
Nesta subseção será apresentado a implementação dos algoritmos de ML escolhidos para este trabalho, e também suas métricas de avaliação. Desta maneira todos os algoritmos foram implementados com IDE de Python chamada Spyder, importando funções da biblioteca Surprise que foi detalhada anteriormente.

Figura 20 – Dataset no método train/test



Fonte: Elaborada pelo autor

Figura 21 – Dataset no método cross-validation



Fonte: Elaborada pelo autor

Para todos os algoritmos descritos neste trabalho foram utilizados os dois arquivos de dados, o ratings.csv e movies.csv, que foram divididos em um conjunto de treinamento e um conjunto de teste, sendo ele 75% e 25% respectivamente chamado de *Train/Test* como ilustrado na Figura 20, e outra forma de divisão chamada *k-fold cross-validation* com respectivos 5 *folds* como pode ser observados na Figura 21. Além disso, as mesmas métricas de RMSE e MAE foram aplicadas igualmente a todos os algoritmos. Para ter uma base comparativa nas métricas de avaliação foi também implementado recomendações aleatórias.

6.2.1 KNN

Para os métodos colaborativos foi implementou-se o algoritmo chamado de KNN (*k-nearest neighbors algorithm*) em português chamado de k-vizinhos mais próximos, esse algoritmo foi utilizado tanto para o método colaborativo baseado no usuário, como para o método colaborativo baseado em item respectivamente. Foram executados dois cenários para ambos os métodos, o primeiro cenário foi executar o algoritmo com os valores de parâmetros de personalização recomendados pela biblioteca do Surprise, de outra forma para o outro cenário esses valores foram alterados para que com isso se obtivesse um melhor valor de resultado nas métricas propostas.

Figura 22 – Código da chamada dos algoritmos KNN

```
# User-based KNN.
UserKNN = KNNBasic(sim_options = {'name': 'cosine', 'user_based': True}, k=40, verbose=True, min_k=1)
evaluator.AddAlgorithm(UserKNN, "User KNN")

# Item-based KNN.
ItemKNN = KNNBasic(sim_options = {'name': 'cosine', 'user_based': False}, k=40, verbose=True, min_k=1)
evaluator.AddAlgorithm(ItemKNN, "Item KNN")
```

Fonte: Elaborada pelo autor

Como pode ser observado na Figura 22 foi implementado o algoritmo KNNBasic, para itens e usuários, conforme é definido no valor booleano que é passado na *flag* `user_based` do `sim_options`. Abaixo é mostrada a lista dos parâmetros e seus respectivos significados.

- `k(int)`: Número máximo de vizinhos a serem considerados para agregação.
- `min_k`: Número mínimo de vizinhos a serem considerados para agregação, se não houver vizinhos suficientes, a previsão é definida como a média global de todas as classificações.
- `sim_options(dict)`: Um dicionário de opções para a medida de similaridade.
 - `name`: O nome da forma da similaridade pra ser utilizada.
 - `user_based`: Valor booleano para definir o método baseado em usuário ou em item.
- `verbose`: Imprime mensagens de rastreamento.

Para o segundo cenário referente ao User-based KNN o valor do parâmetro do `k` foi alterado para 30 e o `min_k` para 2. Já para o Item-based KNN foi modificado o `k` para 50, com isto resultando em uma pequena melhora em suas métricas de ambos os algoritmos.

6.2.2 SVD e SVD++

A implementação dos métodos, de fatora  o de matriz foram o SVD (singular value decomposition) em portugu  s conhecido como decomposi  o em valores singulares e o SVD++, tamb  m se seguiu a ideia de executar esses algoritmos com os valores de par  metros recomendados pelo Surprise e posteriormente alterar esses valores para conseguir um melhor valor nas m  tricas de avalia  o. Seguindo para execu  o do SVD e SVD++, foi importado o m  dulo de SVD e SVDpp conforme como mostra a Figura 23.

Figura 23 – Código da chamada dos algoritmos SVD e SVD++

```
# SVD
SVD = SVD(n_factors=100, n_epochs=20, biased=True, init_mean=0, init_std_dev=0.1, lr_all=0.005, reg_all=0.02)
evaluator.AddAlgorithm(SVD, "SVD")

# SVD++
SVDPlusPlus = SVDpp(n_factors=20, n_epochs=20, init_mean=0, init_std_dev=0.1, lr_all=0.007, reg_all=0.02)
evaluator.AddAlgorithm(SVDPlusPlus, "SVD++")
```

Fonte: Elaborada pelo autor

A lista abaixo apresenta os significados dos parâmetros que foram utilizados nos algoritmos SVD e SVD++.

- `n_factors`: Número de fatores.
- `n_epochs`: Número de iterações do procedimento SGD.
- `biased`: Usar ou não *baselines*. (Somente no SVD).
- `init_mean`: Média da distribuição normal para inicialização de vetores de fator.
- `init_std_dev`: O desvio padrão da distribuição normal para inicialização de vetores de fator.
- `lr_all`: Taxa de aprendizado para todos os parâmetros.
- `reg_all`: Termo de regularização para todos os parâmetros.

O segundo cenário para o SVD foi alterado os parâmetros `n_factors` para 45, `n_epochs` para 29, `lr_all` alterado para 0,009 e `reg_all` sendo 0,06. Com esses valores foi possível observar uma melhora nas métricas. Por último, para o segundo cenário do algoritmo SVD++ também em busca de melhores resultados, os seus parâmetros foram alterados, para: `n_factors` para 19, `n_epochs` para 18 e `reg_all` para 0,03 em que também foi possível observar uma diminuição nas métricas de erro, obtendo-se um melhor resultado respectivamente.

7 RESULTADOS

Todos os algoritmos foram avaliados com as métricas de acurácia da predição sendo elas MAE e RMSE, conforme descrito no capítulo 4.2.1, com seus resultados mostrados nesse capítulo e também suas explicações. De modo generalizado, o algoritmo que obteve um melhor valor nas métricas de acurácia da predição foi o algoritmo SVD.

Seguindo os resultados do capítulo 6.2 de implementação dos algoritmos no segundo cenário pode ser observado na Figura 24, os valores das métricas da predição obtidas para cada um dos algoritmos implementados fazendo uso do método de Train/Test no dataset. De todos os algoritmos implementados, o que apresentou melhores resultados foi o algoritmo SVD, tendo o valor de 0,6860 na métrica MAE e 0,8891 na métrica RMSE. Os piores valores obtidos foram do algoritmo KNN baseado em Item, apresentado um valor de 0,7749 para métrica MAE e 0,9942 para RMSE respectivamente. Para fins de comparação, os resultados das métricas de acurácia para predições aleatórias apresentaram os valores de 1,1637 para MAE e 1,4532 para RMSE.

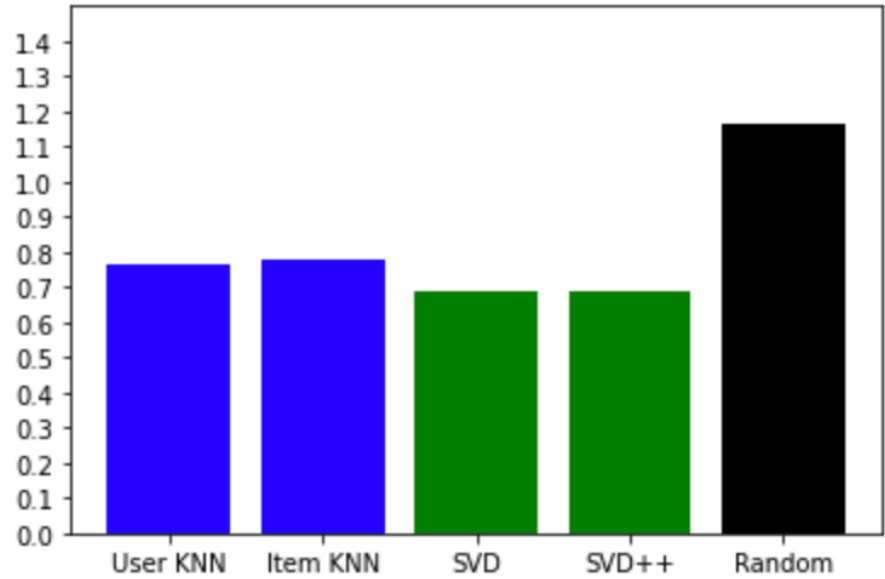
Figura 24 – Resultado dos algoritmos nas métricas propostas usando train/test

Algorithm	RMSE	MAE	Algorithm	RMSE	MAE
User KNN	0.9899	0.7669	SVD	0.8891	0.6860
Item KNN	0.9942	0.7749	SVD++	0.8909	0.6875
Random		1.4532			1.1637

Fonte: Elaborada pelo autor

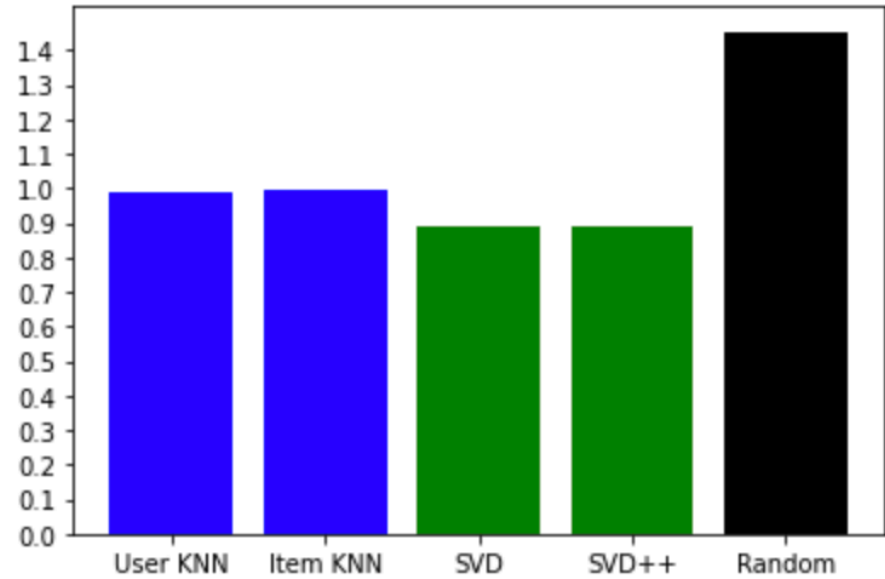
Como pode ser observado na Figura 25, os algoritmos KNN tanto para usuário quanto para item obteve-se pouca variação entre os valores da métrica da acurácia da predição MAE. O mesmo vale para os algoritmos SVD++ e SVD, resultando também para a métrica de RMSE obteve-se uma pequena variação como pode ser visto na Figura 26.

Figura 25 – Algoritmos com a métrica MAE



Fonte: Elaborado pelo autor

Figura 26 – Algoritmos com a métrica RMSE



Fonte: Elaborado pelo autor

Figura 27 – Resultados dos algoritmos nas métricas propostas usando k-fold

User KNN Evaluating RMSE, MAE of algorithm KNNBasic on 5 split(s).						
	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean
RMSE (testset)	0.9876	0.9944	0.9808	0.9834	0.9923	0.9877
MAE (testset)	0.7630	0.7674	0.7600	0.7581	0.7686	0.7634

Item KNN Evaluating RMSE, MAE of algorithm KNNBasic on 5 split(s).						
	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean
RMSE (testset)	0.9911	0.9961	0.9880	0.9868	0.9909	0.9906
MAE (testset)	0.7716	0.7715	0.7681	0.7707	0.7698	0.7704

SVD Evaluating RMSE, MAE of algorithm SVD on 5 split(s).						
	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean
RMSE (testset)	0.8893	0.8804	0.8823	0.8712	0.8839	0.8814
MAE (testset)	0.6834	0.6763	0.6753	0.6720	0.6790	0.6772

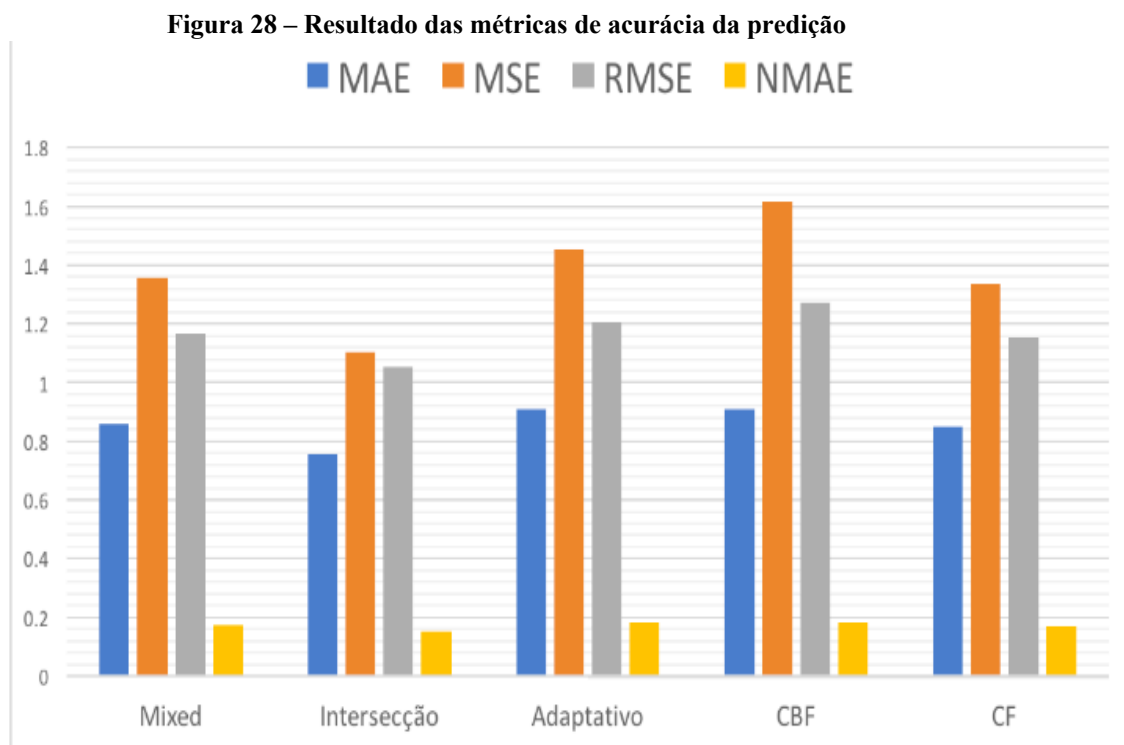
SVD++ Evaluating RMSE, MAE of algorithm SVDpp on 5 split(s).						
	Fold 1	Fold 2	Fold 3	Fold 4	Fold 5	Mean
RMSE (testset)	0.8831	0.8913	0.8880	0.8802	0.8778	0.8841
MAE (testset)	0.6801	0.6828	0.6815	0.6750	0.6769	0.6793

Fonte: Elaborada pelo autor

Para os resultados do segundo método de divisão do *dataset*, sendo ele o *cross-validation*, os resultados obtidos como pode ser observado na Figura 27, mostraram ainda um melhor resultado no algoritmo SVD com valores de 0,6772 para MAE e 0,8814 para RMSE, e para os piores valores sendo o KNN baseado em item, com MAE de 0,7704 e RMSE com o valor de 0,9906, considerando o valor da média dos Folds (*Mean*).

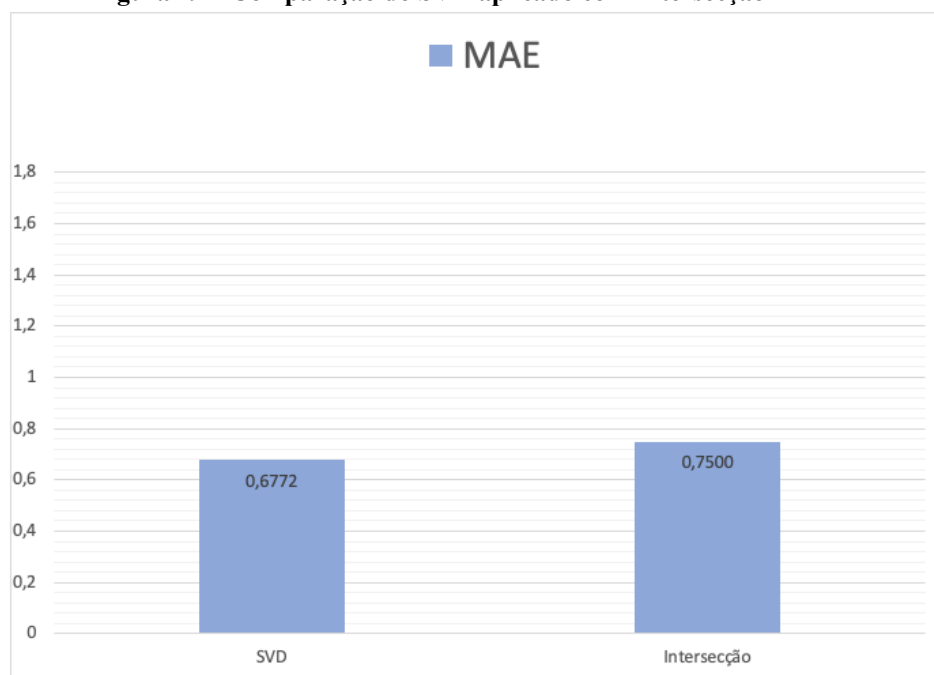
8 COMPARAÇÃO DO SISTEMA DE RECOMENDAÇÃO

Segundo os resultados apresentados no trabalho do REIDEL et al. (2017) como pode ser considerado na Figura 28, os melhores resultados nas métricas de acurácia da predição foi o método da interseção dos conjuntos, seguindo a métrica intersecção dos conjuntos com o valor de acurácia de 0,7500 MAE.



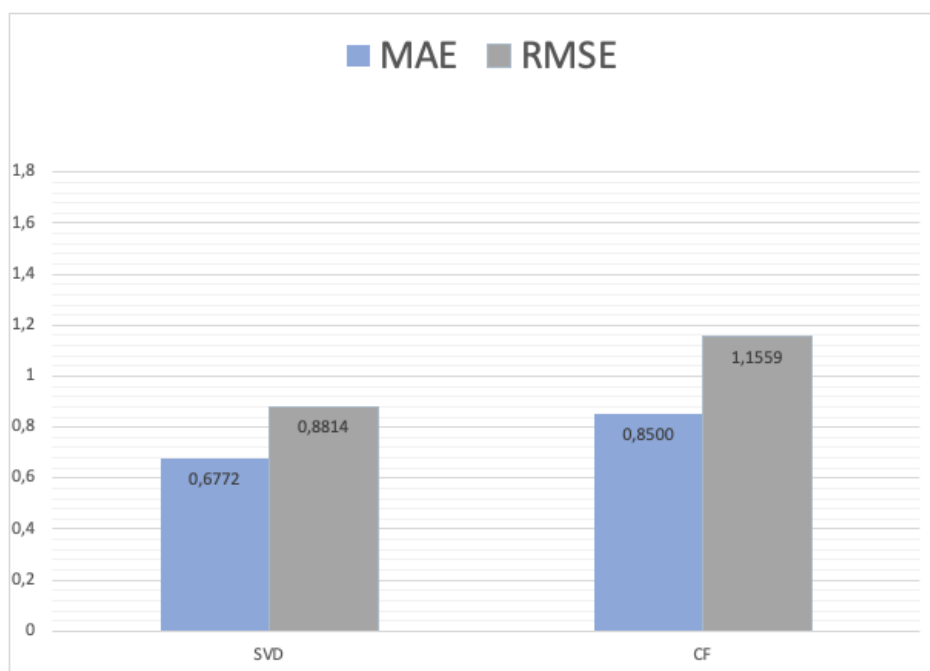
Fonte: (REIDEL et al., 2017)

Esse método ao ser comparado com o melhor algoritmo aplicado por esse trabalho nessa métrica, que foi o SVD, tendo um valor de acurácia de 0,6790 MAE, percebe-se um desempenho melhor nas recomendações com uma diminuição de aproximadamente 10% no erro, como pode ser observado no Figura 29.

Figura 29 – Comparação do SVD aplicado com Intersecção

Fonte: Elaborado pelo autor

Como pode ser observado na Figura 28, o segundo método que obteve melhores resultados nas métricas propostas pelo REIDEL et al. (2017), foi o método CF, que segundo a conclusão de seu estudo, foi o que apresentou o resultado mais satisfatório, pelo motivo de que o método da intersecção dos conjuntos gerou poucas recomendações ao ser comparado com o CF. Com isto o autor concluiu que o método CF proposto por ele, obteve uma quantidade considerável de recomendações e também obteve bons valores nas métricas de acurácia. Por esses motivos, o autor concluiu que ele possa a ser implementado ao HS como seu sistema de recomendação, mesmo sendo observado a necessidade de atualizações que busquem melhorar os resultados das métricas calculadas.

Figura 30 – Comparação do SVD aplicado com CF

Fonte: Elaborado pelo autor

Assim como foi concluído no estudo de REIDEL et al. (2017) o método que apresentou resultados mais satisfatórios no seu trabalho foi o método CF, que obteve 1,1559 e 0,8500 nas métricas de acurácia da predição RMSE e MAE respectivamente. Com base nesse cenário acima e comparando com o método SVD aplicado neste trabalho, se obteve uma melhora de aproximadamente 20,33% para métrica MAE e uma melhora próxima de 23,75% para RMSE, como pode ser observado na Figura 30.

9 CONCLUSÃO

Este trabalho estudo, teve como objetivo desenvolver um sistema de recomendação com o uso de aprendizado de máquina em inglês *Machine Learning* para recomendação de casos clínicos e materiais didáticos para o ambiente do Health Simulator da Universidade Feevale, e através de métricas de acurácia avaliar seu resultado frente ao sistema de recomendação desenvolvido e implementado no Health Simulator.

Inicialmente, no Capítulo 2, foi apresentado brevemente o assunto dos simuladores de ensino em saúde, em seguida foi apresentado o HS e evidenciado seu funcionamento, assim apresentando os principais aspectos do simulador, bem como sua estrutura de funcionamento e algumas das suas principais características, e fechando com uma breve explicação do sistema de recomendação já proposto para o HS.

Já no capítulo 3 foram expostos estudos sobre sistemas de recomendação para plataformas voltadas à educação, abordando-se algumas técnicas de recomendações comumente utilizadas para esse fim, de modo a se ter melhor entendimento dele. No Capítulo 4 foi elaborada uma definição de ML e, com isso, detalhados sistemas de recomendação que utilizam ML. Perquiriu-se, também, sobre as principais técnicas e métricas que utilizam-se em estudos da área para avaliar a qualidade das recomendações prestadas. Nos capítulos 5 e 6 foi analisado e explorado o sistema de recomendação já desenvolvido para o HS, definindo-se as métricas de acurácia e a implementação dos algoritmos de ML para o sistema de recomendação em questão. Sendo, dessa forma, possível encontrar nos capítulos 7 e 8 o desenvolvimento e comparação com o SR já proposto.

A metodologia utilizada nesse trabalho tem com característica uma pesquisa aplicada, em que se fez uso de métodos existentes de sistemas de recomendação e se utilizou algoritmos de ML existentes para a implementação do objetivo proposto para isto. O referencial teórico utilizado neste trabalho apontou que os algoritmos de recomendação voltados para sistemas de recomendação são o SVD e o KNN, desta forma, esses foram os algoritmos escolhidos e implementados para este estudo. Tendo como base o referencial teórico foi possível perceber a importância da escolha correta das métricas de avaliação para as recomendações geradas pelo SR, e também seguindo as mesmas métricas utilizadas pelo sistema já proposto para posteriormente fazer uma comparação adequada.

Os resultados revelam que o algoritmo SVD obteve os melhores resultados nas métricas escolhidas, se confrontado com o SVD++ tiveram valores aproximados, talvez isso tenha se

dado pelo motivo do algoritmo SVD++ ter um tempo de execução demasiadamente maior que os demais algoritmos utilizados para esse trabalho, e por esse motivo não foi possível testar diversos valores de parâmetros diferentes para encontrar melhores resultados com esse algoritmo. Com o algoritmo KNN também se obteve resultados similares entre as técnicas User-Based e Item-Based.

A partir disto, o algoritmo SVD foi confrontado com o método de recomendação proposto pelo (REIDEL et al., 2017), chamado de Filtro Colaborativo (CF). Sendo que o método SVD aplicado neste trabalho perante as mesmas métricas de acurácia da predição utilizadas chega-se ao resultado de uma melhora de aproximadamente 20,33% de diminuição do erro para a métrica MAE, e uma diminuição de 23,75% no erro para a métrica RMSE. Com base nesses valores foi possível evidenciar que o sistema de recomendação aqui apresentado apresentou um desempenho superior aos sistemas de recomendação implementado no HS. Ressalta-se que a base de dados utilizada para o trabalho em tela é a mesma base de dados que foi utilizada para a construção do sistema de recomendação proposto no HS. Além disso, também foram utilizadas as mesmas métricas para a avaliação das recomendações aplicadas, escolhendo-se as mais utilizadas para este cenário, qual seja, o cenário de ML para sistemas de recomendação, tendo como base estudos semelhantes.

Para a oportunidade de trabalhos futuros para esse tema, acredita-se que pode ser explorado o uso de outras métricas para a avaliação do sistema de recomendação como métricas do tipo Top-N. Além disso, também pode ser utilizado Deep Learning para criar um sistema de recomendação através do uso de redes neurais e com isso avaliar o desempenho das recomendações obtidas e comparar com os resultados aqui encontrados.

REFERÊNCIAS BIBLIOGRÁFICAS

- AGGARWAL, C. C. An introduction to recommender systems. In: **Recommender systems**. [s.l.] Springer, 2016. p. 1–28.
- ANDERSON, J. R. et al. An Artificial Intelligence Approach. **Machine Learning**, v. 1, 1985.
- BARROS, P. R. M. **Health Simulator Recommender: sistema de recomendação para apoiar o desenvolvimento do pensamento crítico no ensino em saúde**. [s.l.], 2020.
- BEZ, M. et al. Proposta de um algoritmo de recomendação usando uma rede bayesiana no health simulator. **Ingeniería e Innovación**, v. 4, n. 2, 2016.
- BEZ, M. R. et al. Bayes Editor: Desenvolvimento e validação de um editor de Redes Bayesianas. **Informática na educação: teoria & prática**, v. 20, n. 4 dez, 2017.
- BEZ, M. R.; BARROS, P. R.; MELLO, B. Proposta de uso de simuladores do tipo paciente virtual no ensino em saúde. **Desarrollo y transformación social desde escenarios educativos**, p. 59, 2018.
- BOBADILLA, J. et al. Recommender systems survey. **Knowledge-based systems**, v. 46, p. 109–132, 2013.
- BOURKOUKOU, O.; EL BACHARI, E.; EL ADNANI, M. A recommender model in e-learning environment. **Arabian Journal for Science and Engineering**, v. 42, n. 2, p. 607–617, 2017.
- BRUNIALTI, L. et al. **Aprendizado de maquina em sistemas de recomendacao baseados em conteudo textual: Uma revisao sistematica**. Anais do XI Simpósio Brasileiro de Sistemas de Informação. **Anais...2015b**
- BURKE, R. Hybrid recommender systems: Survey and experiments. **User modeling and user-adapted interaction**, v. 12, n. 4, p. 331–370, 2002.
- BURKE, R. Hybrid web recommender systems. In: **The adaptive web**. [s.l.] Springer, 2007. p. 377–408.
- CATAE, F. S. **Classificação automática de texto por meio de similaridade de palavras: um algoritmo mais eficiente**. [s.l.], 2012.
- CHAI, T.; DRAXLER, R. R. Arguments against avoiding RMSE in the literature. **Geosci. Model Dev**, v. 7, 2014.
- COSTA, E.; AGUIAR, J.; MAGALHÃES, J. Sistemas de Recomendação de Recursos Educacionais: conceitos, técnicas e aplicações. **Jornada de Atualização em Informática na Educação**, v. 1, n. 1, 2013.
- DASCALU, M.-I. et al. A recommender agent based on learning styles for better virtual collaborative learning experiences. **Computers in Human Behavior**, v. 45, p. 243–253, 2015.

DE FAZIO, M. R. **Previsão de avaliações em sistemas de recomendação para nichos de mercado.** [s.l.], 2013.

DEBNATH, S. Machine Learning Based Recommendation System. **Master's thesis, Department of Computer Science and Engineering, Indian Institute of Technology,** 2008.

FACIONE, P. A.; OTHERS. Critical thinking: What it is and why it counts. **Insight assessment**, v. 2007, n. 1, p. 1–23, 2011.

FERRERO, C. A. **Algoritmo kNN para previsão de dados temporais: funções de previsão e critérios de seleção de vizinhos próximos aplicados a variáveis ambientais em limnologia.** [s.l.], 2009.

FUKS, W. J. **Desenvolvimento de preditores para recomendação automática de produtos.** [s.l.], 2013.

GOLDBERG, D. et al. Using collaborative filtering to weave an information tapestry. **Communications of the ACM**, v. 35, n. 12, p. 61–70, 1992.

GROLMAN, E. et al. Utilizing transfer learning for in-domain collaborative filtering. **Knowledge-Based Systems**, v. 107, p. 70–82, 2016.

HAHN, M.; LAWSON, R.; LEE, Y. G. The effects of time pressure and information load on decision quality. **Psychology & Marketing**, v. 9, n. 5, p. 365–378, 1992.

HARPER, F. M.; KONSTAN, J. A. The movielens datasets: History and context. **Acm transactions on interactive intelligent systems (tiis)**, v. 5, n. 4, p. 1–19, 2015.

HELWANGER, F. A.; ROLIM, M. P.; BARROS, P. R. Arquitetura de comunicação do Health Simulator. **Gamepad VIII. Universidade Feevale, Novo Hamburgo,** 2015.

HIGGS, J. et al. **Clinical reasoning in the health professions E-book.** [s.l.] Elsevier Health Sciences, 2008.

HOLZINGER, A. et al. Learning performance with interactive simulations in medical education: Lessons learned from results of learning complex physiological models with the HAEMOdynamics SIMulator. **Computers & Education**, v. 52, n. 2, p. 292–301, 2009.

KER, J.; BRADLEY, P. Simulation in medical education. **Understanding medical education: Evidence, theory and practice**, p. 175–192, 2013.

KNEEBONE, R. Simulation in surgical training: educational issues and practical implications. **Medical education**, v. 37, n. 3, p. 267–277, 2003.

KORDIK, P.; CERNY, J.; FRYDA, T. Discovering predictive ensembles for transfer learning and meta-learning. **Machine Learning**, v. 107, n. 1, p. 177–207, 2018.

KOREN, Y.; BELL, R. Advances in collaborative filtering. In: **Recommender systems handbook.** [s.l.] Springer, 2015. p. 77–118.

KOREN, Y.; BELL, R.; VOLINSKY, C. Matrix factorization techniques for recommender systems. **Computer**, v. 42, n. 8, p. 30–37, 2009.

LAMPROPOULOS, A. S.; TSIHRINTZIS, G. A. Machine Learning Paradigms. **Applications In Recommender Systems. Switzerland: Springer International Publishing**, 2015.

LEWIS, A.; SMITH, D. Defining higher order thinking. **Theory into practice**, v. 32, n. 3, p. 131–137, 1993.

LIMA, A. et al. Projeto para desenvolvimento do Simulador Health Simulator. **Anais do Computer on the Beach**, p. 279–288, 2015.

LOPES, G. R. Avaliação e recomendação de colaborações em redes sociais acadêmicas. 2012.

LOPES, T. C. S. et al. Avaliação de método de aprendizagem ativa por caso clínico no curso de propedêutica do UniFOA. **Cadernos UniFOA**, v. 5, n. 1, p. 40, 2018.

LOPS, P.; DE GEMMIS, M.; SEMERARO, G. Content-based recommender systems: State of the art and trends. In: **Recommender systems handbook**. [s.l.] Springer, 2011. p. 73–105.

MAES, P.; SHARADANAND, M. S. Social information filtering: algorithms for automating. **Word of Mouth**", **CHI Proc**, 1995.

MANOUSELIS, N.; VUORIKARI, R.; VAN ASSCHE, F. Collaborative recommendation of e-learning resources: an experimental investigation. **Journal of Computer Assisted Learning**, v. 26, n. 4, p. 227–242, 2010.

MARIA, S. A. A. RecETC: uma funcionalidade baseada na recomendação de conteúdo para auxiliar no processo de escrita coletiva digital. 2017.

MARONI, V. **Construção de um motor de inferência para análise de desempenho em ambientes virtuais de aprendizagem aplicados ao ensino da medicina de família e comunidade**. [s.l.], 2013.

MARQUES, R. L.; DUTRA, I. Redes Bayesianas: o que são, para que servem, algoritmos e exemplos de aplicações. **Coppe Sistemas—Universidade Federal do Rio de Janeiro, Rio de Janeiro, Brasil**, 2002.

MARTENS, H. H. Two notes on machine “Learning.” **Information and Control**, v. 2, n. 4, p. 364–379, 1959.

MITCHELL, R. S.; MICHALSKI, J. G.; CARBONELL, T. M. **An artificial intelligence approach**. [s.l.] Springer, 2013.

NASCIMENTO, N. et al. **A context-aware machine learning-based approach**. Proceedings of the 28th Annual International Conference on Computer Science and Software Engineering. **Anais...2018**

PAZZANI, M. J.; BILLSUS, D. Content-based recommendation systems. In: **The adaptive web**. [s.l.] Springer, 2007. p. 325–341.

PEIXOTO, T. A. DOS S. M.; PEIXOTO, N. M. DOS S. M. Pensamento crítico dos estudantes de enfermagem em ensino clínico: Uma revisão integrativa. **Revista de Enfermagem Referência**, n. 13, p. 125–138, 2017.

PETERS, L. H. et al. The relationship between time pressure and performance: A field test of Parkinson's Law. **Journal of Occupational Behaviour**, v. 5, n. 4, p. 293–299, 1984.

PORTUGAL, I.; ALENCAR, P.; COWAN, D. The use of machine learning algorithms in recommender systems: A systematic review. **Expert Systems with Applications**, v. 97, p. 205–227, 2018.

REIDEL, D. et al. **Development of a Recommender System to the Virtual Patient Simulator Health Simulator**. Proceedings of the 23rd Brazillian Symposium on Multimedia and the Web. **Anais...2017**

REQUENA, F. O. **Incorporating Group Recommendations to Recommender Systems: Alternatives and Performance**. [s.l], 2015.

RESNICK, P. et al. **GroupLens: an open architecture for collaborative filtering of netnews**. Proceedings of the 1994 ACM conference on Computer supported cooperative work. **Anais...1994**

RUSSELL, S.; NORVIG, P. **Introduction to Artificial Intelligence** Prentice Hall, 1995.

SCHICK, A. G.; GORDON, L. A.; HAKA, S. Information overload: A temporal approach. **Accounting, organizations and society**, v. 15, n. 3, p. 199–220, 1990.

SILVA, N.; ROCHA, L.; MOURÃO, F. **Desmistificando o uso de Matrix Factorization em domínios de recomendação**. Anais do XXXV Concurso de Trabalhos de Iniciação Científica da SBC. **Anais...2016**

SINGHAL, A.; SINHA, P.; PANT, R. Use of deep learning in modern recommendation system: A summary of recent works. **arXiv preprint arXiv:1712.07525**, 2017.

SWEET, R. M.; MCDOUGALL, E. M. Simulation and computer-animated devices: the new minimally invasive skills training paradigm. **Urologic Clinics of North America**, v. 35, n. 3, p. 519–531, 2008.

THAI-NGHE, N. et al. Recommender system for predicting student performance. **Procedia Computer Science**, v. 1, n. 2, p. 2811–2819, 2010.

TIZARD, B.; HUGHES, M. **Young children learning**. [s.l.] John Wiley & Sons, 2008.

TSUJI, H.; AGUILAR-DA-SILVA, R. H. Aprender e ensinar na escola vestida de branco: do modelo biomédico ao humanístico. **São Paulo: Forte**, 2010.

WANG, Z. et al. An improved collaborative movie recommendation system using computational intelligence. **Journal of Visual Languages & Computing**, v. 25, n. 6, p. 667–675, 2014.

ZAHIR, J.; EL QADI, A. A recommendation system for execution plans using machine learning. **Mathematical and Computational Applications**, v. 21, n. 2, p. 23, 2016.

ZIV, A.; BEN-DAVID, S.; ZIV, M. Simulation based medical education: an opportunity to learn from errors. **Medical teacher**, v. 27, n. 3, p. 193–199, 2005.

GROUPLENS. *MovieLens datasets*. Disponível em <<https://grouplens.org/datasets/movielens/>>. Acesso em 12 mar. 2020.

Surprise. *A Python scikit for recommender systems*. Disponível em <<http://surpriselib.com/>>. Acesso em 17 jun. 2020.